Diplomarbeit aus Technischer Mathematik, TU-Graz

Die Simulation von Verteilungen durch MCMC-Methoden mit adaptiver Verwerfung

Stefan Gesslbauer
September 1997

Vorgelegt der Technisch-Naturwissenschaftlichen Fakultät an der Technischen Universität Graz

Begutachter und Betreuer: Prof. Dr. Ernst Stadlober

Institut für Statistik der Technischen Universität Graz

Ich versichere, diese Arbeit selbständig verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient zu haben.

Ich möchte mich bei Herrn Prof. Dr. Ulrich Dieter für die großzügige Bereitstellung der Institutseinrichtungen, die für die Durchführung dieser Arbeit notwendig waren, bedanken.

Mein besonderer Dank gilt Herrn Prof. Dr. Ernst Stadlober für die intensive fachliche Betreuung meiner Diplomarbeit.

Danken möchte ich auch Herrn Dr. Herwig Friedl für die Unterstützung bezüglich Software und die Ratschläge im Umgang mit dem Textsystem.

Überblick

Das wichtigste Anwendungsgebiet von Markov Chain Monte Carlo (MCMC) Methoden ist die Auswertung Bayes'scher Modelle. Die dort auftretenden hochdimensionalen Integrale, die zur Ermittlung statistischer Kenngrößen berechnet werden müssen, sind im allgemeinen nicht analytisch lösbar. MCMC-Methoden bieten die Möglichkeit, diese Kenngrößen simulativ zu approximieren, indem Markovketten einer vorgegebenen Verteilung erzeugt werden, aus denen sich Schätzer der theoretischen Kenngrößen berechnen lassen. Da die Elemente der Markovkette abhängig sind, muß die Genauigkeit der Schätzung anders als bei herkömmlichen statistischen Methoden ermittelt werden. Damit die Unverzerrtheit der Schätzer gewährleistet ist, muß zusätzlich die Konvergenz der stationären Verteilung der Markovkette gegen die theoretische Verteilung untersucht werden,

Eine wegen ihrer einfachen Implementierbarkeit häufig verwendete MCMC-Methode ist der Gibbs-Generator. Für seine Anwendung wurde die adaptive Verwerfungsmethode, mit deren Hilfe Stichproben von logkonkaven Verteilungen erzeugt werden können, entwickelt. Im Rahmen dieser Arbeit wird diese Methode gemeinsam mit einer Verallgemeinerung, die die Erzeugung abhängiger Zufallszahlen beliebiger Verteilungen erlaubt, untersucht. Mit Hilfe eines dafür entwickelten Computerprogramms werden diese Methoden mit traditionellen, auf spezielle Verteilungen zugeschnittene Erzeugungsverfahren verglichen. Anhand von Graphiken kann die Funktionsweise der unterschiedlichen adaptiven Verfahren nachvollzogen werden. Zusätzlich wird die Schnelligkeit und Implementierbarkeit der Algorithmen verglichen. Dieser Arbeit ist eine Diskette beigelegt, die den Quellcode (C++-Routinen) und die ausführbaren Programme, die für diese Untersuchungen nötig waren, enthält.

Inhaltsverzeichnis

Kapitel 1

Markov Chain Monte Carlo Methoden

Ein in der statistischen Modellbildung häufig auftretendes Problem ist die Auswertung von hochdimensionalen Integralen. So litt beispielsweise die Bayes'sche Statistik lange Zeit an der Tatsache, daß die Berechnung der Aposteriori-Verteilung, die im allgemeinen eine Integration über viele Parameter erfordert, zu aufwendig für die praktische Anwendung war. Einhergehend mit der allgemeinen Verfügbarkeit von Computern wurden Verfahren entwickelt, die eine näherungsweise Berechnung der Integrale zulassen. Vor allem Markov Chain Monte Carlo (MCMC) Methoden waren in den letzten Jahren Gegenstand der Forschung und sind heute das wichtigste Instrument zur praktischen Auswertung von Bayes'schen Modellen.

MCMC-Methoden verbinden das Monte Carlo Verfahren zur numerischen Berechnung von Integralen mit der Theorie der Markovketten. Für die Monte Carlo Integration wird eine Reihe von Zufallszahlen einer gewünschten Ausgangsverteilung benötigt, mit deren Hilfe Erwartungswerte geschätzt werden können. Für komplizierte, mehrdimensionale Verteilungen sind oft keine elementaren Zufallszahlengeneratoren verfügbar. MCMC-Methoden verwenden an Stelle von unabhängigen Realisierungen der Ausgangsverteilung irreduzible Markovketten, die mit Hilfe von Verfahren, die auf dem Algorithmus von Metropolis-Hastings (Metropolis *et al.*, 1953, und Hastings, 1970) beruhen, erzeugt werden.

1.1 Bayes'sche Modellierung

Eines der Hauptanwendungsgebiete von MCMC-Methoden ist die Bayes'sche Statistik. Obwohl die zugrundeliegende Theorie schon lange bekannt ist, können entsprechende Verfahren erst heute mit Hilfe von Computern praktisch eingesetzt werden.

In der Bayes'schen Analyse geht man von einem Datenvektor $D = (D_1, \ldots, D_l)$ aus, dessen Komponenten zufällige, jedoch zu Beginn der Untersuchung bekannte Werte sind, die aus einem Experiment oder einer Erhebung stammen können. Mit Hilfe von D sollen Rückschlüsse auf den Vektor der zu schätzenden Modellparameter $\theta = (\theta_1, \ldots, \theta_k)$ gezogen werden. Die Komponenten der Vektoren D und θ sind entweder diskrete oder stetige Zufallsvariable, innerhalb eines Zufallsvektors können auch gemischt stetige und diskrete Komponenten auftreten. Im folgenden wird, wenn nicht anders angegeben, nur auf den stetigen Fall näher eingegangen. Die Ergebnisse gelten jedoch analog für diskrete oder gemischt stetige und diskrete Zufallsvektoren, gegebenenfalls ist das Integral durch das Summationszeichen zu ersetzen und statt der Dichtefunktion die Wahrscheinlichkeitsfunktion zu verwenden.

Wenn $p(D, \theta)$ die gemeinsame Dichte von D und θ ist, so läßt sie sich als Produkt zweier weiterer Dichten schreiben:

$$p(D, \theta) = p(\theta)p(D|\theta).$$

Man nennt $p(\theta)$ die Apriori-Dichte, da Vorkenntnisse über den Modellparameter θ hier durch eine spezielle Wahl der Dichte oder eine Einschränkung des Definitionsbereichs einfließen können.

Mit der sogenannten Likelihoodfunktion $p(D|\theta)$ legt man den Modelltypus fest. Sie folgt dem allgemeineren Likelihoodprinzip, das fordert, daß statistische Schlüsse ausschließlich auf den beobachteten Daten D basieren sollen. Im Fall der Likelihoodfunktion bedeutet das, daß für zwei Modelle mit identischem Datenvektor D auch die Funktion $p(D|\theta)$ identisch ist.

Wendet man nun den Satz von Bayes an, so erhält man nach der Beobachtung der Daten D die Aposteriori-Dichte

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{\int p(\theta)p(D|\theta)d\theta}.$$

Sie ist das zentrale Objekt der Bayes'schen Inferenz. Ihre Momente und Quantile können als bedingte Erwartungswerte einer Funktion $f(\theta) = f(\theta_1, \ldots, \theta_k)$, $f : \mathbb{R}^k \to \mathbb{R}$ betrachtet werden:

$$E[f(\theta)|D] = \int f(\theta)p(\theta|D)d\theta = \frac{\int f(\theta)p(\theta)p(D|\theta)d\theta}{\int p(\theta)p(D|\theta)d\theta}. \tag{1.1}$$

Eine analytische Berechnung von $E[f(\theta)|D]$ ist für höhere Dimensionen im allgemeinen nicht möglich. Alternative Ansätze schlagen numerische Verfahren oder die Laplace-Approximation vor, die aber nur in speziellen Fällen geeignet erscheinen. Für die breitere praktische Anwendung bieten sich das Monte Carlo Verfahren, insbesonders MCMC-Methoden an.

Zur Vereinfachung der Schreibweise und um die folgenden Überlegungen nicht nur auf die Bayes'sche Statistik zu beschränken, soll (1.1) zu

$$E_\pi[f(X)] = \frac{\int f(x)\pi(x)dx}{\int \pi(x)dx} \tag{1.2}$$

vereinfacht werden, wobei $X = (X_{.1}, \ldots, X_{.k})$ ein Zufallsvektor und gemäß einer k-dimensionalen Dichte verteilt ist, deren Normierungskonstante unbekannt und für die weiteren Überlegungen nicht von Bedeutung ist. $\pi(x)$ ist die genannte Dichtefunktion in der nicht normierten Form, d. h. $\int \pi(x)dx$ ist eine unbekannte Größe. Dieser Situation begegnet man häufig in Bayes'schen Analysen, wo man zwar $p(\theta|D) \propto p(\theta)p(D|\theta)$ kennt, das zur Normierung notwendige Integral $\int p(\theta)p(D|\theta)d\theta$ jedoch unbekannt und nur schwierig auszuwerten ist. Allgemein kann X eine Zufallsvektor aus einem Wahrscheinlichkeitsraum, dem ein beliebiger Ereignisraum zugrundeliegt, sein. Die folgenden Überlegungen sollen, wenn nicht anders angegeben, jedoch auf den für die Praxis wichtigsten Fall $X \in \mathbb{R}^k$ beschränkt bleiben.

1.2 Monte Carlo Integration

Monte Carlo Integration ermittelt näherungsweise den Erwartungswert $E[f(X)]$ aus (1.2) durch die Erzeugung von Zufallsvektoren $\{X_t; X_t = (X_{t.1}, \dots, X_{t.k}), t = 0, \dots, N\}$ mit dem Startwert X_0 aus der Dichte $\pi(x)$ und die Approximation durch den empirischen Erwartungswert

$$\bar{f}_N = \frac{1}{N+1} \sum_{t=0}^{N} f(X_t).$$

Wenn die Zufallsvektoren $\{X_t\}$ unabhängig sind, so sichert das Starke Gesetz der Großen Zahlen, daß der empirische Mittelwert fast sicher gegen den wahren Wert $E[f(X)]$ konvergiert. Oft hat die Dichtefunktion $\pi(x)$ eine komplizierte Gestalt, sodaß eine Erzeugung von unabhängigen $\{X_t\}$ nicht möglich ist. Für die Konvergenz ist die Unabhängigkeit der $\{X_t\}$ jedoch nicht erforderlich. Mit Verfahren, die auf dem Algorithmus von Metropolis-Hastings beruhen, können unter relativ geringem Aufwand Markovketten einer beliebigen stationären Dichte $\pi(x)$ erzeugt werden. Man nennt ein Verfahren dieser Art eine Markov Chain Monte Carlo Methode.

1.3 Der Metropolis-Hastings-Algorithmus

Umgekehrt zu der in der Markovtheorie gestellten Frage, ob zu einer gegebenen Markovkette eine stationäre Verteilung existiert und welche Gestalt sie hat, stellt sich nun das Problem, eine Markovkette zu erzeugen, deren stationäre Dichte die vorgegebene Funktion $\pi(\cdot)$ ist. Eine Lösung bietet der Algorithmus, der von Metropolis *et al.* (1953) vorgeschlagen und von Hastings (1970) verallgemeinert wurde. Weitere Erläuterungen den Metropolis-Hastings-Algorithmus betreffend finden sich in Chib und Greenberg (1995).

Eine homogene Markovkette $\{X_t, t \in \mathbb{N}_0\}$ ist eindeutig durch die Verteilung des Startwertes X_0 und die Übergangswahrscheinlichkeiten $P(x, A)$ festgelegt. Genügt eine Markovkette bestimmten Bedingungen (s. Kap. 2), so erreicht sie nach genügend langer Zeit einen stationären Zustand, der unabhängig von der Verteilung des Startwertes ist. Die Suche nach einer Markovkette mit einer bestimmten stationären Verteilungsfunktion ist deshalb äquivalent zur Bestimmung ihrer Übergangswahrscheinlichkeiten.

$P(x, A)$ gibt die Wahrscheinlichkeit an, sich vom Punkt $x \in \mathbb{R}^k$ zu einer Menge $A \in \mathcal{B}$ zu bewegen, wobei $\mathcal{B}$ die σ-Algebra auf $\mathbb{R}^k$ ist:

$$P(x, A) = \Pr[X_{n+1} \in A | X_n \in x] \qquad n \geq 0.$$

$P(x, \cdot)$ ist die Verteilungsfunktion der Markovkette nach einem Schritt ausgehend vom Punkt x, und deshalb gilt $P(x, \mathbb{R}^k) = 1$. Ein Verweilen in einem beliebigen Punkt x soll zugelassen sein, d. h. $P(x, \{x\})$ ist nicht notwendigerweise null. Mit $\Pi(\cdot)$ sei die stationäre Verteilungsfunktion bezeichnet, d. h. $\Pi(dy) = \pi(y)dy$, wobei $\Pi(dy)$ die Wahrscheinlichkeit einer infinitesimal kleinen Teilmenge $dy \in \mathbb{R}^k$ unter $\Pi(\cdot)$ ist. Allgemein erhält man eine stationäre Verteilung durch die Bedingung

$$\Pi(dy) = \int_{\mathbb{R}^k} P(x, dy)\pi(x)dx. \tag{1.3}$$

Für die Suche von geeigneten Übergangswahrscheinlichkeiten wählt man eine Funktion $p(x, y)$ so, daß für $x \in \mathbb{R}^k$ und $y \in \mathbb{R}^k$

$$P(x, dy) = p(x, y)dy + r(x)I_{(x\in dy)}$$

gilt, und $p(x, x) = 0$ ist. I ist die Indikatorfunktion, d. h. $I_{(x\in dy)} = 1$, falls $x \in dy$ und 0 sonst. Die Funktion $r(x) = 1 - \int_{\mathbb{R}^k} p(x, y)dy$ bezeichnet die Wahrscheinlichkeit, daß die Markovkette in einem Schritt im Punkt x verharrt. Erfüllt $p(x, y)$ außerdem die Umkehrbedingung

$$\pi(x)p(x, y) = \pi(y)p(y, x), \tag{1.4}$$

dann ist $\pi(\cdot)$ die stationäre Dichtefunktion von $P(x, \cdot)$. Um das zu sehen, muß man die rechte Seite von (1.3) auswerten:

$$\begin{aligned}
\int P(x, A)\pi(x)dx &= \int \Big[\int_A p(x, y)dy\Big]\pi(x)dx + \int r(x)I_{x\in A}\pi(x)dx \\
&= \int_A \Big[\int p(x, y)\pi(x)dx\Big]dy + \int_A r(x)\pi(x)dx \\
&= \int_A \Big[\int p(y, x)\pi(y)dx\Big]dy + \int_A r(x)\pi(x)dx \\
&= \int_A (1 - r(y))\pi(y)dy + \int_A r(x)\pi(x)dx \\
&= \int_A \pi(y)dy.
\end{aligned}$$

Man erhält also die hinreichende Bedingung (1.4) für $p(x, \cdot)$, sodaß $\pi(\cdot)$ eine stationäre Dichtefunktion für die Markovkette mit den Übergangswahrscheinlichkeiten $P(x, \cdot)$ ist. Es ist schwierig, ein geeignetes $p(x, \cdot)$ direkt zu finden. Der Metropolis-Hastings-Algorithmus wählt zunächst eine Dichte $q(x, y)$ mit $\int q(x, y)dy = 1$, die zu einem gegebenen Punkt x einen neuen Punkt y generiert. Im allgemeinen trifft für diese Dichte die Umkehrbedingung nicht zu, es gibt also Punkte x und y, für die ohne Beschränkung der Allgemeinheit

$$\pi(x)q(x, y) > \pi(y)q(y, x) \tag{1.5}$$

gilt. Das bedeutet, daß sich die Markovkette mit größerer Wahrscheinlichkeit von Punkt x zu Punkt y bewegt als umgekehrt. Um dieses Ungleichgewicht zu korrigieren, führt man die Akzeptanzwahrscheinlichkeit $\alpha(x, y)$ ein, die angibt, ob der vorgeschlagene Punkt y akzeptiert werden soll. $\alpha(x, y) < 1$ macht als Korrekturfaktor auf der linken Seite von (1.5) die Ungleichung zu einer Gleichung. Wenn der Punkt in einem Schritt verworfen wird, so erzeugt man mit Hilfe von $q(x,y)$ ein neues y und untersucht erneut die Akzeptanzwahrscheinlichkeit $\alpha(x,y)$. Übergänge von x zu y, $x \neq y$, ergeben sich laut

$$p_{MH}(x,y) = q(x,y)\alpha(x,y).$$

Zur Konstruktion von $\alpha(x,y)$ soll die Ungleichung (1.5) noch einmal betrachtet werden. Unter der Berücksichtigung, daß für $p_{MH}(x,y)$ die Umkehrbedingung gelten soll und die Akzeptanzwahrscheinlichkeit $\alpha(x,y)$ als obere Grenze eins hat, ergibt sich

$$\begin{aligned} \pi(x)q(x,y)\alpha(x,y) &= \pi(y)q(y,x)\alpha(y,x) \\ &= \pi(y)q(y,x). \end{aligned}$$

Man erkennt, daß $\alpha(x,y) = \pi(y)q(y,x)/\pi(x)q(x,y)$ ist. Für den Fall, daß die Ungleichung (1.5) umgedreht ist, setzt man analog $\alpha(x,y) = 1$ und berechnet $\alpha(y,x)$ wie oben. Wenn man deshalb von $p_{MH}(x,y)$ die Erfüllung der Umkehrbedingung fordert, so muß für die Akzeptanzwahrscheinlichkeit

$$\alpha(x,y) = \begin{cases} \min\left\{\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}, 1\right\}, & \text{falls } \pi(x)q(x,y) > 0 \\ 1 & \text{sonst.} \end{cases}$$

gelten. Nun muß noch die Wahrscheinlichkeit $r(x)$, daß die Markovkette in einem Schritt im gleichen Punkt verharrt, berücksichtigt werden. Wie oben erhält man

$$r(x) = 1 - \int_{\mathbb{R}^k} q(x,y)\alpha(x,y)dy.$$

Die Übergangswahrscheinlichkeiten $P_{MH}(x,dy)$ des Metropolis-Hastings- Algorithmus haben nun die Form

$$P_{MH}(x,dy) = q(x,y)\alpha(x,y)dy + \left[1 - \int_{\mathbb{R}^k} q(x,y)\alpha(x,y)dy\right] I_{(x\in dy)}.$$

Für jeden Schritt des Metropolis-Hastings-Algorithmus ist die Erzeugung eines Wertes Y als Kandidaten des nächsten Punktes und die Auswertung der Akzeptanzwahrscheinlichkeit $\alpha(x,Y)$ notwendig. Man erzeugt eine Markovkette $\{X_t, t = 0,\ldots,N\}$ mit stationärer Dichtefunktion $\pi(\cdot)$ mit Hilfe des Metropolis-Hastings-Algorithmus durch

```
Initialisiere X_0 ; setze t = 0.
Solange t < N
{
    Erzeuge Y aus q(X_t, ·)
    Erzeuge U aus Uniform(0,1)
    Falls U ≤ α(X_t, Y) setze X_{t+1} = Y
        Sonst setze X_{t+1} = X_t
    Setze t = t + 1
}.
```

Alle bisherigen Überlegungen forderten keine Voraussetzungen von der Gestalt der Dichtefunktion $q(x, y)$, die Kandidaten Y für das Element X_{t+1} der Markovkette vorschlägt. Tatsächlich funktioniert der Algorithmus für eine beliebige Wahl von $q(x, y)$. Die Konvergenzrate der Markovkette hängt jedoch entscheidend von $q(x, y)$ ab. Außerdem kann eine ungünstige Wahl die Ursache einer schlechten Mischung (s. Abschnitt 1.4) von $\{X_t\}$ sein, d. h. die Elemente der Kette sind stark abhängig oder der Definitionsbereich von $\pi(x)$ wird erst nach einer großen Anzahl von Iterationen durchlaufen. Im folgenden sollen Spezialfälle des Metropolis-Hastings-Algorithmus gezeigt werden, die sich durch eine besondere Wahl von $q(x, y)$ auszeichnen.

1.3.1 Der Metropolis-Algorithmus

Der Metropolis-Algorithmus verwendet ausschließlich symmetrische Dichten $q(x, y) = q(y, x)$. Einen Spezialfall des Metropolis-Algorithmus erhält man durch die Wahl von $q(x, y) = q(|x - y|)$, die sogenannte Irrfahrt. Die Akzeptanzwahrscheinlichkeit vereinfacht sich zu

$$\alpha(x, y) = \min\left\{\frac{\pi(y)}{\pi(x)}, 1\right\}.$$

Eine wichtige Rolle spielt der Skalierungsparameter von $q(x, y)$. Ist die Varianz klein, dann werden im allgemeinen Kandidaten Y erzeugt, die nahe dem aktuellen Element X_t liegen. Die Akzeptanzwahrscheinlichkeit ist dann zwar hoch, die Mischung der Kette jedoch schlecht. Umgekehrt führt ein zu großer Skalierungsparameter zu Kandidaten, die weit vom aktuellen Element in den Schwänzen der Dichte $\pi(x)$ liegen und deshalb häufig verworfen werden. Der Skalierungsparameter hat also großen Einfluß auf die Mischung der Markovkette.

1.3.2 Der Unabhängigkeitsgenerator

Für den Unabhängigkeitsgenerator (*Independence sampler*, Tierney, 1994) verwendet man zur Erzeugung von Kandidaten eine Dichte der Gestalt $q(x, y) = q(y)$, d. h. neue Elemente X_{t+1} der Markovkette werden unabhängig vom aktuellen Element X_t erzeugt. Die Akzeptanzwahrscheinlichkeit lautet dann

$$\alpha(x, y) = \min\left\{\frac{\omega(y)}{\omega(x)}, 1\right\},$$

wobei $\omega(x) = \pi(x)/q(x)$. Die Kandidatendichte $q(x)$ sollte im Idealfall eine gute Näherung von $\pi(x)$ sein, auf jeden Fall sollte $q(x)$ jedoch höhere Schwänze als $\pi(x)$ haben, wie folgende Überlegung zeigt: Angenommen X_t befindet sich gerade im Schwanz von $\pi(x)$. Als Kandidat wird wahrscheinlich ein Wert Y vorgeschlagen, der näher zum Zentrum der Verteilung liegt. Daraus ergibt sich, daß $\omega(X_t)$ größer als $\omega(Y)$ ist, und eine niedrige Akzeptanzwahrscheinlichkeit läßt die Kette lange in der extremen Position verweilen.

1.3.3 Der Einzelkomponenten-Metropolis-Hastings-Algorithmus

In den bisher vorgestellten MCMC-Methoden wurden für den Übergang von X_t zu X_{t+1} alle k Komponenten des Vektors X_t in einem Schritt erneuert. Oft ist es leichter, die k Komponenten in h Blöcke ($h \leq k$) einzuteilen und jeden Block einzeln zu erneuern. Nach welchen Kriterien der Vektor X effizient eingeteilt werden kann, soll später behandelt werden. Nach dieser Einteilung besteht der Zufallsvektor X aus den Blöcken $\{X_{.1}, \ldots, X_{.h}\}$, wobei ein Block wiederum ein mehrdimensionaler Zufallsvektor sein kann. Weiters sei $X_{.-i} = \{X_{.1}, \ldots, X_{.i-1}, X_{.i+1}, \ldots, X_{.h}\}$, d. h. $X_{.-i}$ entspricht X unter Wegnahme des Blocks $X_{.i}$. Eine Iteration des Einzelkomponenten-Metropolis-Hastings-Algorithmus (*Single component Metropolis Hastings*, Metropolis *et al.*, 1953) besteht aus h Erneuerungsschritten. Es sei $X_{t.i}$ der Zustand von $X_{.i}$ zum Zeitpunkt t. Um im Schritt i der Iteration $t+1$ den Block $X_{.i}$ zu erneuern, wird ein Metropolis-Hastings-Schritt angewandt. Es wird ein Kandidat $Y_{.i}$ aus einer Dichtefunktion $q_i(\{X_{t.i}, X_{t.-i}\}, Y_{.i})$ generiert, wobei $X_{t.-i}$ den Block $X_{.-i}$ nach Ausführung von Schritt $i-1$ der Iteration $t+1$ bezeichnet:

$$X_{t.-i} = \{X_{t+1.1}, \ldots, X_{t+1.i-1}, X_{t.i+1}, \ldots, X_{t.h}\}.$$

Weiters ist

$$\{X_{t.i}, X_{t.-i}\} = \{X_{t+1.1}, \ldots, X_{t+1.i-1}, X_{t.i}, X_{t.i+1}, \ldots, X_{t.h}\}.$$

Die Dichtefunktion $q_i(\cdot, \cdot)$ generiert also für Block i einen Kandidaten unter Berücksichtigung der aktuellen Werte aller Blöcke von X. Der Kandidat Y_i wird mit der Wahrscheinlichkeit $\alpha(X_{t.-i}, X_{t.i}, Y_{.i})$ akzeptiert, wobei

$$\alpha(X_{t.-i}, X_{t.i}, Y_{.i}) = \min\left\{1, \frac{\pi(Y_{.i}|X_{.-i})q_i(\{Y_{.i}, X_{.-i}\}, X_{.i})}{\pi(X_{.i}|X_{.-i})q_i(\{X_{.i}, X_{.-i}\}, Y_{.i})}\right\}. \tag{1.6}$$

Man nennt $\pi(X_{.i}|X_{.-i})$ die vollständig bedingte Verteilung von $X_{.i}$ unter der Dichte $\pi(\cdot)$:

$$\pi(X_{.i}|X_{.-i}) = \frac{\pi(X)}{\int \pi(X)dX_{.i}}. \tag{1.7}$$

Falls $Y_{.i}$ gemäß (1.6) akzeptiert wird, so setzt man $X_{t+1.i} = Y_{.i}$, ansonsten $X_{t+1.i} = X_{t.i}$. Die übrigen Blöcke bleiben im Schritt i unverändert. Daß der beschriebene Algorithmus tatsächlich eine Markovkette mit stationärer Dichtefunktion $\pi(\cdot)$ erzeugt, resultiert aus der Tatsache, daß $\pi(\cdot)$ eindeutig durch die Menge aller vollständig bedingten Verteilungen definiert ist (Besag, 1974).

1.3.4 Der Gibbs-Generator

Ein Spezialfall des Einzelkomponenten-Metropolis-Hastings-Algorithmus ist der Gibbs-Generator, der von Geman und Geman (1984) erstmals beschrieben wurde. Der Algorithmus wurde ursprünglich zur Rekonstruktion von Bildern entwickelt, heute wird er jedoch in vielen Gebieten der Statistik zur Simulation von komplexen Modellen angewandt. Das Softwarepaket `BUGS` (Spiegelhalter *et al.*, 1996) verwendet beispielsweise ausschließlich den Gibbs-Generator zur Analyse von Bayes'schen Modellen.

Die Dichtefunktion $q_i(\cdot,\cdot)$, mit deren Hilfe für den i-ten Block von X Kandidaten $Y_{.i}$ erzeugt werden, hat für den Gibbs-Generator die Gestalt

$$q_i(\{X_{.i}, X_{.-i}\}, Y_{.i}) = \pi(Y_{.i}|X_{.-i}), \tag{1.8}$$

wobei $\pi(Y_{.i}|X_{.-i})$ die vollständig bedingte Verteilung (1.7) ist. Setzt man (1.8) in (1.6) ein, so erhält man eine Akzeptanzwahrscheinlichkeit von eins, d. h. *jeder* Kandidat $Y_{.i}$ wird akzeptiert. *Gibbs sampling* ist somit die Erzeugung einer Kette von Zufallsvektoren $\{X_t, t = 1, \ldots, n\}$ aus den vollständig bedingten Verteilungen. Wenn $X_0 = \{X_{0.1}, \ldots, X_{0.k}\}$ ein beliebiger Startvektor ist, so erhält man das Schema für die erste Iteration des Gibbs-Generators durch

$$\begin{array}{lll}
\texttt{Erzeuge} & X_{1.1} & \texttt{aus} \quad \pi(X_{.1}|X_{0.-1}); \\
\texttt{Erzeuge} & X_{1.2} & \texttt{aus} \quad \pi(X_{.2}|\{X_{1.1}, X_{0.3}, \ldots, X_{0.k}\}); \\
\texttt{Erzeuge} & X_{1.3} & \texttt{aus} \quad \pi(X_{.3}|\{X_{1.1}, X_{1.2}, X_{0.4}, \ldots, X_{0.k}\}); \\
 & \vdots & \\
\texttt{Erzeuge} & X_{1.k} & \texttt{aus} \quad \pi(X_{.k}|X_{1.-k}).
\end{array}$$

1.4 Mischung und Konvergenz

Die beiden Begriffe „*Mischung*“ und „*Konvergenz*“ sind wichtige Begriffe zur Beschreibung des Langzeitverhaltens einer mit Hilfe von MCMC-Methoden erzeugten Markovkette $\{X_t, t = 0, \ldots, N\}$. Zur Schätzung von Kenngrößen der stationären Verteilung ist es notwendig, die Stichprobengröße so groß zu wählen, daß die Kette stationäres Verhalten und somit Konvergenz gegen die stationäre Verteilung aufweist. Aber selbst im Fall einer konvergenten Kette können Probleme bei der Untersuchung der stationären Verteilung auftreten, wenn die Kette eine schlechte Mischung aufweist.

Die bisher vorgestellten Metropolis-Hastings Algorithmen weisen mit Ausnahme des Gibbs-Generators eine Akzeptanzwahrscheinlichkeit $\alpha(x, y)$ auf, die echt kleiner als eins ist. Die Größe der Wahrscheinlichkeit der Akzeptanz von Kandidaten in jedem Schritt des Algorithmus hängt von der Wahl der Vorschlagsdichte $q(x, y)$ ab. Ist die Anzahl der Verwerfungen groß, so wird es vorkommen, daß sich die Kette über mehrere Iterationen nicht von einem Zustand wegbewegt, sodaß zum Beispiel $X_t = X_{t+1} = \ldots = X_{t+r}$ mit $r > 0$ gilt. Tritt dieser Fall oft auf, so spricht man von einer „schlechten Mischung“ der Markovkette. Für den Schätzer des Erwartungswertes

$$\bar{f}_N = \frac{1}{N+1} \sum_{t=0}^{N} f(X_t)$$

liefern die $r - 1$ Iterationen nach der t-ten Iteration effektiv keine neue Information mehr, d. h. um die gleiche Genauigkeit der Schätzung wie ein Algorithmus mit der idealen Akzeptanzwahrscheinlichkeit $\alpha(x, y) = 1$ zu erreichen, müßte die Samplegröße N erhöht werden.

Von schlechter Mischung spricht man auch dann, wenn zwar die Akzeptanzwahrscheinlichkeit groß ist, aufeinanderfolgende Zustände X_t und X_{t+1} der Kette jedoch stark korreliert sind. Auch das hat einen negativen Einfluß auf die Güte der Schätzung, da stark korrelierte X_t unter Umständen viele Iterationen benötigen, um den ganzen Definitionsbereich der stationären Verteilung zu erreichen. Befindet sich beispielsweise die Kette zum Zeitpunkt t_0 in einem Schwanz der Verteilung und ist die Samplegröße N nicht groß genug gewählt, daß die Kette das Zentrum der Wahrscheinlichkeitsmasse der stationären Verteilung erreichen kann, dann wird der Schätzer $\bar{f}_N$ verzerrt sein.

Nicht nur Autokorrelation zwischen Zuständen X_t und X_{t+1} kann eine schlechte Mischung verursachen, auch die Korrelation von einzelnen Komponenten $X_{t.i}$ und $X_{t.j}$ mit $i \neq j$ kann Probleme bereiten. Eine Lösung in diesem Fall kann die Reparametrisierung der Daten $X_{.1}, \ldots, X_{.k}$ oder die Verwendung eines speziellen Algorithmus (Adaptive Richtungsmethode, s. Abschnitt 6.2) bieten.

Um schlechte Mischungen zu vermeiden, ist damit sowohl auf die Wahl der Kandidatendichte $q(x, y)$ als auch auf die Parametrisierung der Ausgangsdaten zu achten.

Die Konvergenz einer Markovkette ist unabhängig von ihrer Mischung zu betrachten, obwohl eine schlechte Mischung die Konvergenzgeschwindigkeit negativ beeinflußt. In den Bereich der Konvergenzanalyse fällt unter anderem die Entscheidung, ab welcher Iteration t_0 eine Markovkette $\{X_t, t = 0, \ldots, N\}$ stationäres Verhalten zeigt und wie groß N bestimmt werden muß, sodaß die Genauigkeit des Schätzers $\bar{f}_N$ ein bestimmtes Niveau erreicht. Für diese Überlegungen stehen Analyseverfahren zu Verfügung, die in Kapitel 5 besprochen werden.

Ein besonderes Problem bezüglich der Konvergenz stellen jene stationäre Verteilungen dar, die mehrere Modi aufweisen. Von entsprechenden konvergenten Markovketten erwartet man, daß deren Zustände nicht nur um einen einzigen Modus konzentriert sind, sondern jeder Extremwert der stationären Verteilung regelmäßig von der Kette besucht wird.

Kapitel 2

Einführung in die Theorie der Markovketten

Dieses Kapitel soll einen Überblick über die Theorie der Markovketten geben, der für das Verständnis von MCMC-Methoden notwendig ist. Zuerst werden die Grundbegriffe der Markovketten mit diskretem Zustandsraum dargelegt. Ausführlicher sollen die Markovketten mit stetigem Zustandsraum behandelt werden, weil die der Kette zugrundeliegende stationäre Verteilung in der Praxis meistens ebenfalls stetig ist. Eine Zusammenfassung der vorgestellten Ergebnisse kann in Kapitel 3 und 4 von Gilks *et al.* (1995) gefunden werden. Für die Beweise der folgenden Sätze sei auf Tierney (1994) verwiesen.

Die Verteilung $\Pi(\cdot)$ sei auf der Menge E, dem sogenannten Zustandsraum, definiert. Für die folgenden Ergebnisse ist als einzige Bedingung von E zu fordern, daß die Menge aller Teilmengen $\mathcal{E}$ eine endlich erzeugte σ-Algebra sei. Alle Teilmengen von E und alle auf E definierten Funktionen seien $\mathcal{E}$-meßbar.

Markovketten müssen bestimmte Bedingungen erfüllen, damit eine stationäre Verteilung $\Pi(\cdot)$ existiert. Auf diese Bedingungen wird nur kurz eingegangen, da sich die Theorie der MCMC-Methoden mit der Konstruktion einer Markovkette, deren stationäre Verteilung eine gegebene Verteilung $\Pi(\cdot)$ sein soll, beschäftigt; die Existenz von $\Pi(\cdot)$ ist deshalb gesichert.

Für Markovketten mit diskretem Zustandsraum E bezeichnet $\pi(\cdot)$ die zu $\Pi(\cdot)$ gehörende Wahrscheinlichkeitsfunktion. Für die Betrachtungen des stetigen Zustandsraumes wird manchmal die Existenz einer Dichtefunktion vorausgesetzt. Sie wird ebenfalls mit $\pi(\cdot)$ bezeichnet. Es gilt dann $\Pi(dx) = \pi(x)dx$.

Wie bereits in Abschnitt 1.2 erwähnt, steht die Berechnung des Erwartungswertes

$$E_\Pi[f(X)] = \int f(x)\Pi(dx) = \int f(x)\pi(x)dx \tag{2.1}$$

für eine reellwertige Funktion $f(\cdot)$ im Mittelpunkt der MCMC-Methoden. Ist der Zustandsraum der Markovkette diskret, dann berechnet man (2.1) durch

$$E_\Pi[f(X)] = \int f(x)\Pi(dx) = \sum f(x)\pi(x).$$

Durch die Erzeugung einer geeigneten Markovkette wird versucht, $E_\Pi[f(X)]$ durch

$$\bar{f}_N = \frac{1}{N+1}\sum_{t=0}^{N} f(X_t) \tag{2.2}$$

näherungsweise zu bestimmen. Es wird in diesem Kapitel ein Äquivalent zum Starken Gesetz der großen Zahlen für Markovketten vorgestellt, das garantiert, daß der empirische Mittelwert (2.2) über die abhängigen $\{X_t, t \in N_0\}$ gegen den Erwartungswert $E_\Pi[f(X)]$ stochastisch konvergiert. Wie sich zeigen wird, müssen dafür nur schwache Bedingungen an die Markovkette gestellt werden. Etwas stärkere Bedingungen erlauben sogar einen Zentralen Grenzwertsatz, d. h. der Schätzer (2.2) ist normalverteilt mit dem Erwartungswert $E_\Pi[f(X)]$ und einer noch zu definierenden Varianz σ_f^2. Da die Schätzung von σ_f^2, die zur Bestimmung eines Konfidenzintervalls von $\bar{f}_N$ notwendig ist, für den Fall abhängiger X_t in der Praxis nicht immer einfach ist, ist ihr ein eigener Abschnitt gewidmet.

2.1 Markovketten mit diskretem Zustandsraum

Definition 2.1.1 *Ein stochastischer Prozeß $\{X_t, t \in N_0\}$ mit Zustandsraum E heißt* Markovkette*, falls für alle Zeitpunkte $t_0 < t_1 < \ldots < t_n$ und eine beliebige Menge A aus der σ-Algebra von E*

$$\Pr[X_{t_n} \in A | X_{t_0}, X_{t_1}, \ldots, X_{t_{n-1}}] = \Pr[X_{t_n} \in A | X_{t_{n-1}}]$$

gilt. $\Pr[\cdot|\cdot]$ *bezeichnet die bedingte Wahrscheinlichkeit auf der σ-Algebra von E.*
Die Markovkette heißt homogen*, wenn*

$$P(x, A) = \Pr[X_{t_n} \in A | X_{t_{n-1}} = x]$$

unabhängig von n ist. Man nennt $P(x, A)$ die Übergangswahrscheinlichkeit vom Punkt x zur Menge A.

Für diesen Abschnitt soll der Zustandsraum auf $E = \mathbb{N}_0$ beschränkt bleiben, d. h. $\{X_t\}$ sei eine Menge von Skalaren und ihr Zustandsraum sei diskret. Man betrachtet nun die Übergangswahrscheinlichkeiten

$$P_{ij}(t) = \Pr[X_t = j | X_0 = i], \quad i, j \in E.$$

Die Rückkehrzeit zu einem Zustand $i \in E$ wird durch

$$\tau_{ii} = \min_{t>0}\{X_t = i | X_0 = i\}$$

festgelegt. τ_{ii} ist eine diskrete Zufallsvariable mit Wahrscheinlichkeitsfunktion $f_i(t)$, wobei

$$\begin{aligned} f_i(t) &= \Pr[\tau_{ii} = t] \\ &= \Pr[X_t = i | X_r \neq i, r = 1, \ldots, t-1; X_0 = i]. \end{aligned}$$

Damit die Verteilung von $\{X_t\}$ gegen eine stationäre Verteilung konvergiert, muß die Markovkette drei Bedingungen erfüllen. Sie muß erstens *irreduzibel* sein, d. h. eine beliebige Teilmenge von E kann unabhängig vom Startpunkt X_0 mit positiver Wahrscheinlichkeit erreicht werden. Die *Aperiodizität* garantiert zweitens, daß innerhalb der Kette keine periodischen Muster von Zuständen auftreten. Und drittens sichert die *positive Rekurrenz* die Existenz einer stationären Verteilung.

Definition 2.1.2 *Die Markovkette $\{X_t, t \in \mathbb{N}_0\}$ sei homogen.*

1. *$\{X_t\}$ heißt* irreduzibel, *wenn für alle i, $j \in E$ ein $t > 0$ existiert, sodaß $P_{ij}(t) > 0$ gilt.*

2. *Die irreduzible Markovkette $\{X_t\}$ ist* aperiodisch, *wenn für mindestens ein (und deshalb für alle) $i \in E$*
$$ggT_{t>0}\{P_{ii}(t)\} = 1$$
gilt.

3. *Die irreduzible Markovkette $\{X_t\}$ heißt* rekurrent, *wenn*
$$\Pr[\tau_{ii} < \infty] = \sum_t f_i(t) = 1$$
für mindestens ein $i \in E$ (und somit für alle i) gilt. Eine nicht rekurrente irreduzible Markovkette heißt transient. *Wenn*
$$\mathrm{E}[\tau_{ii}] = \sum_t t f_i(t) < \infty$$
für mindestens ein $i \in E$ (und somit für alle i) gilt, so nennt man $\{X_t\}$ positiv rekurrent. *Eine rekurrente, jedoch nicht positiv rekurrente Markovkette heißt* nullrekurrent. *Eine weitere äquivalente Definition von positiver Rekurrenz fordert die Existenz einer* stationären Wahrscheinlichkeitsfunktion *$\pi(\cdot)$, für die*
$$\sum_i \pi(i) P_{ij}(t) = \pi(j) \qquad \forall j \in E, \quad t \geq 0 \tag{2.3}$$
gelten muß.

2.2 Markovketten mit stetigem Zustandsraum

Es sollen nun die Definitionen aus dem vorangegangenen Abschnitt so modifiziert werden, daß auch Aussagen über einen stetigen Zustandsraum E getroffen werden können. Bei der Anwendung von MCMC-Methoden tritt beispielsweise häufig der Fall $E = \mathbb{R}^k$ auf. Die folgenden Überlegungen gelten jedoch auch für den Fall, daß Zufallsvektoren $X = (X_{.1}, \ldots, X_{.k}) \in E$ sowohl stetige als auch diskrete Komponenten besitzen.

Die Definition einer Markovkette und deren Homogenität ist für stetige Zustandsräume identisch zu Definition 2.1.1. $P(x, A)$ bezeichnet wieder die Übergangswahrscheinlichkeit vom Punkt x zur Menge A. Die n-te Potenz der Übergangswahrscheinlichkeiten ergibt sich rekursiv durch

$$P^n(x,A) = \int P^{n-1}(x,dy)P(y,A) \qquad \forall x \in E, \quad A \subset E,$$

wobei $P^1(x,dy) = P(x,dy)$. Außerdem gilt $P^0(x,A) = 1$ für $x \in A$, ansonsten ist $P^0(x,A) = 0$. Für homogene Markovketten ist dann

$$\Pr[X_n \in A | X_0 = x] = P^n(x,A) \qquad n \geq 0.$$

Später wird gezeigt, unter welchen Bedingungen $P^n(x,\cdot)$ gegen $\pi(\cdot)$ konvergiert.

Die Zeit der ersten Rückkehr einer Markovkette mit einem beliebigen Startvektor $X_0 = x, x \in E$, zu einer Menge $A \subset E$ definiert man als

$$\tau_A = \inf_{n \geq 1}\{X_n \in A\},$$

wobei $\tau_A = \infty$, wenn die Markovkette nicht mehr zur Menge A zurückkehrt.

Der Begriff der Irreduzibilität verlangt für Markovketten mit stetigem Zustandsraum eine allgemeinere Definition als für den diskreten Fall.

Definition 2.2.1 *Es sei $\varphi(\cdot)$ eine Verteilung auf E.*
Eine homogene Markovkette ist φ-irreduzibel*, wenn aus $\varphi(A) > 0$ für eine Menge $A \subset E$ folgt, daß*

$$\Pr[\tau_A < \infty | X_0 = x] > 0$$

für alle $x \in E$. Eine Markovkette $\{X_t\}$ ist irreduzibel, wenn sie für eine Funktion $\varphi(\cdot)$ φ-irreduzibel ist.

Durch diese Definition können Markovketten mit diskretem Zustandsraum, die unter Definition 2.1.2 nicht als irreduzibel gelten, als φ-irreduzibel eingestuft werden. Als Beispiel soll das Problem der Irrfahrt auf dem Zustandsraum $E = \{0, 1, \ldots\}$ betrachtet werden, dessen Matrix der Übergangswahrscheinlichkeiten die Gestalt

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \ldots \\ q & 0 & p & 0 & 0 & \ldots \\ 0 & q & 0 & p & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

mit $q + p = 1$, $q > 0$ und $p > 0$ hat. Nach der Definition der Irreduzibilität für diskrete Zustandsräume ist die Markovkette reduzibel mit dem absorbierenden Zustand $\{0\}$ und den transienten Zuständen $\{1,2,\ldots\}$. Definition 2.2.1 stuft sie jedoch als irreduzibel für die Wahrscheinlichkeitsfunktion $\varphi(\cdot)$ ein, deren ganze Wahrscheinlichkeitsmasse im Ursprung liegt.

Für eine irreduzible Markovkette nach Definition 2.2.1 ist die Verteilung $\varphi(\cdot)$ im allgemeinen nicht eindeutig bestimmt. Man bezeichnet eine Verteilung $\psi(\cdot)$ als maximal, wenn sie bezüglich aller Verteilungen $\varphi(\cdot)$ absolut stetig ist, d. h. aus $\psi(A) = 0$ folgt $\varphi(A) = 0$ für eine beliebige Menge $A \subset E$. Es können mehrere maximale Verteilungen $\psi(\cdot)$ für eine irreduzible Markovkette existieren, diese sind jedoch äquivalent, d. h. sie haben die gleichen Nullmengen.

Definition 2.2.2 *Eine homogene, irreduzible Markovkette $\{X_t\}$ mit maximaler Verteilung $\psi(\cdot)$ ist* rekurrent, *wenn sie für eine Menge $A \subset E$ mit $\psi(A) > 0$ den beiden folgenden Bedingungen für $t \to \infty$ genügt:*

1. $\Pr[X_t \in A$ *unendlich oft*$|X_0 = x] > 0$ *für alle* $x \in E$.
2. $\Pr[X_t \in A$ *unendlich oft*$|X_0 = x] = 1$ *für alle* $x \in E$ *außer einer Nullmenge bezüglich* $\psi(\cdot)$.

Eine irreduzible, rekurrente Markovkette ist positiv rekurrent, *wenn eine stationäre Verteilung $\Pi(\cdot)$ existiert, d. h.*

$$\Pi(dy) = \int P(x, dy)\Pi(dx). \tag{2.4}$$

Ist die Verteilungsfunktion $\Pi(\cdot)$ stetig, dann sei $\pi(\cdot)$ die zugehörige stationäre Dichtefunktion bezüglich des Lebesgue-Maßes, d. h. $\Pi(dy) = \pi(y)dy$. Wenn der Zustandsraum diskret ist, dann bezeichnet $\pi(\cdot)$ die zu $\Pi(\cdot)$ gehörende, gemäß (2.3) gebildete Wahrscheinlichkeitsfunktion.

Satz 2.2.1 *Eine homogene Markovkette $\{X_t\}$ sei irreduzibel und hat $\Pi(\cdot)$ als stationäre Verteilung. Dann gilt:*

1. $\{X_t\}$ *ist* Π-*irreduzibel.*
2. $\Pi(\cdot)$ *ist maximal.*
3. $\Pi(\cdot)$ *ist eindeutig.*
4. $\{X_t\}$ *ist positiv rekurrent.*

Obiger Satz spielt für die MCMC-Methoden eine wichtige Rolle. Man versucht, eine Markovkette $\{X_t\}$ zu erzeugen, deren stationäre Verteilung $\Pi(\cdot)$ einer gegebenen Funktion entspricht. Kann man für $\{X_t\}$ die Irreduzibilität zeigen, so folgt daraus die positive Rekurrenz und somit die Konvergenz der Kette, wie der folgende Satz zeigt. Als Maß des Abstandes zwischen zwei Dichtefunktionen (Wahrscheinlichkeitsfunktionen) $\nu_1(\cdot)$ und $\nu_1(\cdot)$ dient die Norm

$$\|\nu_1 - \nu_2\| = 2 \sup_{A \subset E} |\nu_1(A) - \nu_2(A)|.$$

Satz 2.2.2 *$\{X_t\}$ sei eine irreduzible Markovkette mit stationärer Verteilung $\Pi(\cdot)$. Man definiert*

$$\bar{P}^N(x, A) = \frac{1}{N+1} \sum_{i=0}^{N} P^i(x, A)$$

für alle $x \in E$ und $A \subset E$. Dann gilt für $N \to \infty$

$$\|\bar{P}^N(x, \cdot) - \pi(\cdot)\| \to 0$$

für alle x ausgenommen einer Nullmenge bezüglich $\pi(\cdot)$.

Um MCMC-Methoden anwenden zu können, ist ein Äquivalent zum Starken Gesetz der großen Zahlen notwendig, durch das mit Hilfe des empirischen Mittelwertes

$$\bar{f}_N = \frac{1}{N+1}\sum_{t=0}^{N} f(X_t)$$

der abhängigen Zufallsvektoren X_t der Erwartungswert $E_\Pi[f(X)]$ approximiert werden kann.

Satz 2.2.3 *(Starkes Gesetz der großen Zahlen)*
$\{X_t\}$ sei eine irreduzible Markovkette mit Übergangswahrscheinlichkeiten $P(\cdot,\cdot)$ und stationärer Verteilung $\Pi(\cdot)$. $f(\cdot)$ sei eine reellwertige Funktion, für die $E_\Pi[|f(X)|] < \infty$ gilt. Dann folgt für $N \to \infty$

$$\Pr[\bar{f}_N \to E_\Pi[f(X)] | X_0 = x] = 1$$

für alle $x \in E$ ausgenommen einer Nullmenge bezüglich $\Pi(\cdot)$.

Für die bisherigen Ergebnisse war die Eigenschaft der Aperiodizität der Markovkette nicht nötig. Diese Eigenschaft spielt bei der Anwendung von MCMC-Methoden allgemein keine große Rolle, da nicht die Markovkette selbst im Mittelpunkt der Betrachtung steht, sondern empirische Erwartungswerte, die mit Hilfe der Kette berechnet werden. Ein gewisses Muster bei der Wiederholung von Zuständen hat deshalb keinen negativen Einfluß. In manchen Fällen ist es sogar vorteilhaft, für MCMC-Simulationen periodische Markovketten zu konstruieren.

Für stärkere Konvergenzaussagen wird ab nun die Eigenschaft der Aperiodizität einer Markovkette benötigt.

Definition 2.2.3 *Ein m-Zyklus einer irreduziblen Markovkette mit Übergangswahrscheinlichkeiten $P(\cdot|\cdot)$ ist die Menge $\{E_0, \ldots, E_{m-1}\}$ von schnittfremden Teilmengen von E, sodaß $P(x, E_j) = 1$ für $j \equiv i+1 \mod m$ für alle $x \in E$ gilt. Die Periode d ist das größte m, für das ein m-Zyklus existiert. Die Markovkette ist aperiodisch, wenn $d = 1$.*

Satz 2.2.4 *$\{X_t\}$ sei eine irreduzible, aperiodische Markovkette mit Übergangswahrscheinlichkeiten $P(\cdot,\cdot)$ und stationärer Verteilung $\Pi(\cdot)$. Dann gilt für $N \to \infty$*

$$\|P^N(x,\cdot) - \pi(\cdot)\| \to 0$$

für alle $x \in E$ ausgenommen einer Nullmenge bezüglich $\pi(\cdot)$.

Die Sätze 2.2.1 bis 2.2.4 gelten für alle Startwerte $x \in E$ ausgenommen einer Nullmenge bezüglich $\Pi(\cdot)$. Diese Ausnahme ist auf Bedingung 2 aus Definition 2.2.2 zurückzuführen, die ebenfalls eine Nullmenge ausnimmt.

Die nun folgende Definition der *Harris-Rekurrenz* weitet nun die Bedingung 2 auf ganz E aus, und man kann zeigen, daß für Markovketten mit dieser zusätzlichen Eigenschaft die genannten Sätze für alle Startwerte $x \in E$ gelten.

Definition 2.2.4 *Eine irreduzible Markovkette $\{X_t\}$ mit maximaler Dichtefunktion (Wahrscheinlichkeitsfunktion) $\psi(\cdot)$ heißt* Harris–rekurrent, *wenn für eine beliebige Menge $A \subset E$ mit $\psi(A) > 0$ für $t \to \infty$*

$$\Pr[X_t \in A \text{ unendlich oft } | X_0 = x] = 1$$

für alle $x \in E$ gilt. Sie ist positiv Harris–rekurrent, *wenn eine stationäre Verteilung* $\Pi(\cdot)$ *existiert. Irreduzible Markovketten, die aperiodisch und Harris–rekurrent sind, nennt man* ergodisch.

Für das oben formulierte Starke Gesetz der großen Zahlen war die Rekurrenz hinreichend. Um einen Zentralen Grenzwertsatz für Markovketten angeben zu können, sind strengere Bedingungen erforderlich. Die beiden folgenden Definitionen von Konvergenzraten spielen dabei eine wichtige Rolle.

Definition 2.2.5 *Eine ergodische Markovkette mit stationärer Verteilung* $\Pi(\cdot)$ *ist* geometrisch ergodisch, *wenn es eine nichtnegative, reellwertige Funktion $M(\cdot)$ mit $E_\Pi[M(X)] < \infty$ und einer positiven Konstanten $r < 1$ gibt, sodaß*

$$\|P^n(x,\cdot) - \pi(\cdot)\| \leq M(x)r^n$$

für alle x und n.

Eine geometrisch ergodische Markovkette, deren Verteilung des Startwertes X_0 die stationäre Verteilung ist, besitzt eine α-Mischung, d. h.

$$\alpha(n) = \sup_{A,B \subset E} |\Pr_\pi(X_0 \in A, X_n \in B) - \pi(A)\pi(B)| = O(r^n)$$

für $0 < r < 1$. $\Pr_\pi(\cdot)$ bezeichnet die Wahrscheinlichkeit des Eintretens des Ereignisses $\cdot$, wenn X_0 gemäß der Dichte- bzw. Wahrscheinlichkeitsfunktion $\pi(\cdot)$ verteilt ist.

Eine restriktivere Definition ist die *gleichmäßige Ergodizität*:

Definition 2.2.6 *Eine ergodische Markovkette mit stationärer Verteilung* $\Pi(\cdot)$ *ist* gleichmäßig ergodisch, *wenn es eine endliche Konstante M und einer positiven Konstante $r < 1$ gibt, sodaß*

$$\|P^n(x,\cdot) - \pi(\cdot)\| \leq Mr^n$$

für alle x und n.

Gleichmäßige Ergodizität ist äquivalent zu exponentieller ϕ-Mischung, d. h. für eine gleichmäßig ergodische Markovkette mit stationärer Verteilung $\Pi(\cdot)$ und zugehöriger Dichte- bzw. Wahrscheinlichkeitsfunktion $\pi(\cdot)$ gilt

$$\phi(n) = \sup_{A,B \subset E} |\Pr_\pi(X_n \in B | X_0 \in A) - \pi(B)| = O(r^n)$$

für $0 < r < 1$.

Wenn f eine reellwertige, bezüglich der stationären Verteilung $\Pi(\cdot)$ quadratisch integrierbare Funktion ist, dann bezeichnet man

$$\begin{aligned} \gamma_s = \gamma_{-s} &= \operatorname{Cov}\Big[f(X_i), f(X_{i+s})\Big] \\ &= E_\Pi\Big[\Big(f(X_i) - E_\Pi[f(X)]\Big)\Big(f_(X_{i+s}) - E_\Pi[f(X)]\Big)\Big] \end{aligned} \tag{2.5}$$

als die *lag s* Autokovarianz der stationären Zeitreihe $f(X_0), f(X_1), \ldots$, wenn X_0 gemäß $\Pi(\cdot)$ verteilt ist.

Man kann nun den für die Anwendung von MCMC-Methoden wichtigen Satz formulieren:

Satz 2.2.5 *(Zentraler Grenzwertsatz)*
Für die ergodische Markovkette $\{X_t\}$ mit stationärer Verteilung $\Pi(\cdot)$ und die reellwertige Funktion $f(\cdot)$ sei eine der beiden folgenden Bedingungen erfüllt:

1. *Die Kette ist geometrisch ergodisch und $E_\Pi[|f^{2+\epsilon}(X)|] < \infty$ für ein $\epsilon > 0$.*

2. *Die Kette ist gleichmäßig ergodisch und $E_\Pi[f^2(X)] < \infty$.*

Dann ist

$$\sigma_f^2 = \sum_{s=-\infty}^{+\infty} \gamma_s$$

wohldefiniert, nichtnegativ und endlich, und die Verteilung von $\sqrt{N}(\bar{f}_N - E_\Pi[f(X)])$ konvergiert gegen $N(0, \sigma_f^2)$ für $N \to \infty$.

2.3 Die Schätzung des Monte Carlo Fehlers

Der Monte Carlo Fehler beschreibt die Abweichung des Schätzers $\bar{f}_N$ vom tatsächlichen Erwartungswert $E_\Pi[f(X)]$. Durch die Angabe eines Konfidenzintervalls soll die Genauigkeit der Schätzung näherungsweise bestimmt werden. Dafür muß die Varianz σ_f^2, die im vorangegangenen Abschnitt definiert wurde, geschätzt werden. Durch die Abhängigkeit der X_t ist eine Schätzung in der Praxis problematisch. Geyer (1992) schlägt drei verschiedene Methoden zur Schätzung von σ_f^2 vor.

2.3.1 Der Fensterschätzer

Der natürliche Schätzer der *lag* s Autokovarianz γ_s ist

$$\hat{\gamma}_{N,s} = \hat{\gamma}_{N,-s} = \frac{1}{N+1} \sum_{i=0}^{N-s} [f(X_i) - \bar{f}_N][f(X_{i+s}) - \bar{f}_N].$$

Ein naiver Schätzer für σ_f^2 wäre die Summe über alle $\hat{\gamma}_{N,s}$ bezüglich s, aber wie man aus der Zeitreihentheorie weiß, ist dieser Schätzer nicht einmal konsistent. Für große s ist die Varianz von $\hat{\gamma}_{N,s}$ nämlich näherungsweise

$$\mathrm{Var}(\hat{\gamma}_{N,s}) \approx \frac{1}{N+1} \sum_{i=-\infty}^{+\infty} \gamma_i^2, \tag{2.6}$$

unter der Voraussetzung, daß $E[f(X)^4] < \infty$ und der Prozeß eine ausreichend gute Mischung aufweist. Die rechte Seite von (2.6) hängt nicht von s ab. Die Varianz des Schätzers $\hat{\gamma}_{N,s}$ geht also für $s \to \infty$ nicht gegen null, und die Varianz der Summe der $\hat{\gamma}_{N,s}$ ist von der Ordnung eins anstatt von Ordnung $1/N$.

Um einen brauchbaren Schätzer für σ_f^2 zu erhalten ist es deshalb notwendig, bei der Summierung über alle $\hat{\gamma}_{N,s}$ eine Gewichtsfunktion $\omega_N(s)$ zu verwenden, um den Einfluß der Summanden für großes s zu reduzieren:

$$\hat{\sigma}_N^2 = \sum_{s=-\infty}^{+\infty} \omega_N(s)\hat{\gamma}_{N,s}, \tag{2.7}$$

wobei $0 \leq \omega_N(s) \leq 1$ ist. Oft wählt man $\omega_N(s) = 1$ für einen Bereich von s, der von der Länge der Markovkette N abhängig sein kann, und null außerhalb dieses Bereiches. Man nennt den Bereich, für den $\omega_N(s) = 1$ ist, ein "Fenster" und den Schätzer (2.7) deshalb Fensterschätzer. Die Konsistenz des Fensterschätzers kann unter strengen Bedingungen gelten, es ist jedoch nicht klar, ob er unter den relativ schwachen Voraussetzungen des Zentralen Grenzwertsatzes konsistent ist. In der Praxis kann $\hat{\sigma}_N^2$ jedoch gute Ergebnisse liefern, auch wenn der Rechenaufwand beträchtlich ist.

2.3.2 Methode der standardisierten Zeitreihen

Die Methode der standardisierten Zeitreihen benutzt inkonsistente Schätzer für σ_f^2, verwendet jedoch die asymptotische Verteilung der Varianzschätzer zur Bestimmung von Konfidenzintervallen. Die hier vorgestellte Methode ist die einfachste Variante aus den oben genannten Schätzer.

Wenn die Länge der Markovkette N ein Vielfaches einer natürlichen Zahl m ist, dann kann man die Kette in m gleich große Teilstücke (*batches*) teilen und über jedes Teilstück den Mittelwert (*batch mean*)

$$Z_{N,r} = \frac{m}{N} \sum_{i=(r-1)N/m+1}^{rN/m} f(X_i), \qquad r = 1, \ldots, m$$

bilden. Die einzelnen $Z_{N,r}$ sind aufgrund des zentralen Grenzwertsatzes 2.2.5 unabhängige, asymptotisch normalverteilte Zufallsvariable. Ihr gemeinsamer Mittelwert ist $E_\Pi[f(X)]$. Deshalb ist eine t-Statistik, die unter Verwendung der $Z_{N,r}$ konstruiert wird, asymptotisch t-verteilt mit $m-1$ Freiheitsgraden, und sie kann zur Konstruktion eines Konfidenzintervalls von $E_\Pi[f(X)]$ dienen.

Die Methode der standardisierten Zeitreihen kann unter den schwachen Bedingungen des zentralen Grenzwertsatzes angewendet werden. Es ist jedoch darauf zu achten, daß aufgrund des asymptotischen Verhaltens des Schätzers die Länge der Markovkette N für diese Methode größer zu wählen ist als für Methoden, die die Varianz direkt schätzen. Außerdem erhält man im allgemeinen größere Konfidenzintervalle als bei der Anwendung anderer vergleichbarer Methoden.

Die vorgestellte Methode behandelt die $Z_{N,r}$ nämlich als unabhängige Zufallsvariable, was nur dann annähernd richtig ist, wenn die Länge der Teilstücke größer als die charakteristische Mischungszeit der Markovkette sind. Die Anzahl der Teilstücke m sollte deshalb so klein wie möglich, d. h. ohne das auf der t-Statistik beruhende Konfidenzintervall zu sehr zu vergrößern, sein. In der Praxis hat sich eine Wahl von m im Bereich von 10–30 als brauchbar erwiesen. Damit m nicht zu klein gewählt wird, sollte aber vor Anwendung der Methode auf jeden Fall die Autokovarianzstruktur von $f(X_0), f(X_1), \ldots$ untersucht werden.

2.3.3 Spezielle Schätzer für Markovketten

Herkömmliche Methoden zur Analyse von Simulationsergebnisse wurden nicht speziell für Markovketten entwickelt. Unter Berücksichtigung der spezifischen Eigenschaften von Markovketten können deshalb eventuell bessere Ergebnisse erzielt werden.

Betrachtet man beispielsweise die Autokovarianzen γ_s einer reversiblen Markovkette, so stellt man fest, daß γ_s für gerade s positiv ist, für ungerade s trifft das im allgemeinen nicht zu. Der folgende Satz über die Summe von "benachbarten" Autokovarianzen kann zur Konstruktion dreier verschiedener Schätzer für σ_f^2 angewendet werden.

Satz 2.3.1 *Für eine reversible, irreduzible, stationäre Markovkette, deren Autokovarianzen γ_s gemäß (2.5) definiert sind, ist $\Gamma_s = \gamma_{2s} + \gamma_{2s+1}$ die Summe zweier "benachbarter" Autokovarianzen.*

Γ_s ist dann eine streng positive, monoton fallende und konvexe Funktion in s.

Der Beweis kann in Geyer (1992) nachgelesen werden.

Die in Satz 2.3.1 genannten Eigenschaften von Γ_s können zur Konstruktion von Fensterschätzern (s. Abschnitt 2.3.1) benutzt werden. Wie bereits erwähnt, ist die Summation über alle $\hat{\gamma}_{N,s}$ in (2.6) nicht sinnvoll, da die Varianz von $\hat{\gamma}_{N,s}$ für $s \to \infty$ nicht gegen 0 geht.

Betrachtet man die empirischen Autokovarianzen einer Markovkette mit den Eigenschaften aus Satz 2.3.1, so kann man einerseits den für zunehmende s abklingenden Einfluß der theoretischen Autokovarianz γ_s und andererseits ein "Grundrauschen", das durch die unabhängig von s gleichbleibende Varianz von $\hat{\gamma}_{N,s}$ verursacht wird, feststellen. Zur Schätzung der Varianz σ_f^2 von $\bar{f}_N$ macht es Sinn, nur über jene empirische Autokovarianzen $\hat{\gamma}_{N,s}$ zu summieren, in denen der Einfluß von γ_s jenen des Grundrauschens überwiegt. Satz 2.3.1 gibt nun darüber Auskunft, wie groß das Fenster zu wählen ist.

Es ist sicher nicht sinnvoll, die Summation auf solche Terme auszuweiten, für die "benachbarte" empirische Autokovarianzen negativ sind. Man bildet deshalb nur die Schätzer für Γ_s, nämlich

$$\hat{\Gamma}_{N,s} = \hat{\gamma}_{N,2s} + \hat{\gamma}_{N,2s+1},$$

für die $\hat{\Gamma}_{N,s}$ positiv ist.

Als erster intuitiver Schätzer für σ_f^2 kann nun

$$\begin{aligned} \hat{\sigma}^2_{\mathrm{pos,N}} &= \hat{\gamma}_{N,0} + 2\sum_{i=1}^{2s+1} \hat{\gamma}_{N,i} \\ &= -\hat{\gamma}_{N,0} + 2\sum_{i=0}^{s} \hat{\Gamma}_{N,i} \end{aligned} \tag{2.8}$$

verwendet werden, wobei s die größte natürliche Zahl ist, für die

$$\hat{\Gamma}_{N,i} > 0, \qquad i = 1, \ldots, s$$

gilt.

Der Schätzer $\hat{\sigma}^2_{\mathrm{pos,N}}$ ist für praktische Zwecke meistens gut verwendbar. Es kann jedoch vorkommen, daß für zunehmendes s der Einfluß des Grundrauschens bereits überwiegt, die Schätzer $\hat{\Gamma}_{N,s}$ jedoch positiv sind und die Wahl des Fensters deshalb zu groß vorgenommen wird. In diesem Fall können aus Satz 2.3.1 zusätzliche Kriterien zur Bestimmung der Fenstergröße abgeleitet werden.

Γ_s ist eine streng monoton fallende Funktion bezüglich s. Man kann nun einen weiteren Schätzer $\hat{\sigma}^2_{\mathrm{mon,N}}$ ableiten, der gleich wie $\hat{\sigma}^2_{\mathrm{pos,N}}$ in (2.8) zu ermitteln ist, nur wird s so bestimmt, daß $\hat{\Gamma}_{N,0} > \ldots > \hat{\Gamma}_{N,s}$ gilt. Ein dritter Schätzer $\hat{\sigma}^2_{\mathrm{konv,N}}$ ergibt sich aus der in Satz 2.3.1 beschriebenen Eigenschaft der Konvexität von Γ_s bezüglich s. Die Fenstergröße s legt man nun so fest, daß $\hat{\Gamma}_{N,0}, \ldots, \hat{\Gamma}_{N,s}, 0$ eine konvexe Folge darstellt.

Es ist nicht bekannt, ob $\hat{\sigma}^2_{\mathrm{pos,N}}$, $\hat{\sigma}^2_{\mathrm{mon,N}}$ und $\hat{\sigma}^2_{\mathrm{konv,N}}$ konsistent sind, wenn man nur die Summierbarkeit der Autokovarianzen voraussetzt. Daß die drei genannten Schätzer in der Praxis trotzdem eingesetzt werden können, garantiert der folgende Satz:

Satz 2.3.2 *Für fast alle durch MCMC-Methoden erzeugte Markovketten gilt*

$$\liminf_{N\to\infty} \hat{\sigma}^2_N \geq \sigma^2_f,$$

wobei $\hat{\sigma}^2_N$ einer der drei genannten Schätzer für σ^2_f ist.

Durch die Tatsache, daß σ^2_f fast immer überschätzt wird, kann garantiert werden, daß die mit Hilfe von $\hat{\sigma}^2_{\mathrm{pos,N}}$, $\hat{\sigma}^2_{\mathrm{mon,N}}$ und $\hat{\sigma}^2_{\mathrm{konv,N}}$ geschätzten Konfidenzintervalle nie kleiner als die theoretischen sind.

Kapitel 3

Die adaptive Verwerfungsmethode

Ein zur Erzeugung von Markovketten häufig eingesetztes Instrument ist der Gibbs-Generator. Um ihn effizient einsetzen zu können, wurde mit der adaptiven Verwerfungsmethode ein Universalverfahren zur Erzeugung von Zufallszahlen entwickelt.

Wie in Abschnitt 1.3.4 beschrieben wurde, ist der Gibbs-Generator ein Einzelkomponenten-Algorithmus, der eine Markovkette $\{X_t; X_t = (X_{t.1}, \ldots, X_{t.k}), t \in \mathbb{N}_0\}$ mit stationärer, k-dimensionaler Dichte $\pi(\cdot)$ dadurch erzeugt, daß im i-ten Schritt der $t+1$-ten Iteration der Block $X_{t.i}$ erneuert wird. Die Zufallsgröße $X_{.i}$ wird aus der vollständig bedingten Verteilung

$$\pi(X_{.i}|X_{.-i}) = \frac{\pi(X)}{\int \pi(X) dX_{.i}}. \tag{3.1}$$

gewonnen. Eine Iteration des Algorithmus besteht aus k Schritten, d. h. um den Übergang von X_{t_0} zu X_{t_0+1} für ein $t_0 \in \mathbb{N}_0$ zu vollziehen, ist die Erneuerung aller k Blöcke und somit die Generierung von Zufallsgrößen aus k unterschiedlichen Dichten durchzuführen. Für eine praktische Anwendung ist daher eine effiziente Methode zur Generierung von $X_{.i}$ nötig, um die Rechenzeit in akzeptablen Grenzen zu halten.

Es sollen nun Verfahren vorgestellt werden, die in der Praxis gute Ergebnisse erzielt haben. Für den Fall, daß die Dichte $\pi(x)$ *logkonkav* bezüglich x ist, können Verfahren mit adaptiver Verwerfung verwendet werden. Diese Verfahren stellen eine Verbesserung der herkömmlichen Verwerfungsmethode insofern dar, als die Hutfunktion bei jeder Verwerfung so modifiziert wird, daß die Wahrscheinlichkeit einer weiteren Verwerfung reduziert wird. Zur Konstruktion der Hutfunktion werden die Tangentenmethode (Gilks und Wild, 1992) und die Sekantenmethode vorgestellt (Gilks, 1992). Da es sich bei beiden Verfahren um Verwerfungsmethoden handelt, sind die erzeugten Zufallsgrößen unabhängig.

Eine Erweiterung der adaptiven Verwerfungsmethode durch einen Metropolis-Hastings-Schritt erlaubt auch die Anwendung auf nicht-logkonkave Dichtefunktionen (Gilks *et al.*, 1995). Wie beim herkömmlichen Metropolis-Hastings-Algorithmus sind hier die erzeugten Zufallsgrößen aber abhängig.

Zur Herleitung der adaptiven Verwerfungsmethode seien die zu erzeugenden Zufallsgrößen X Skalare mit Dichtefunktion $p : D \to \mathbb{R}$, deren Definitionsbereich $D \subseteq \mathbb{R}$ ist. Später wird gezeigt, wie die Methode für die Erzeugung von Zufallsvektoren $X = (X_{.1}, \ldots, X_{.k})$ genutzt werden kann.

3.1 Die Verwerfungsmethode

Zunächst soll kurz die herkömmliche, nicht-adaptive Verwerfungsmethode beschrieben werden, die auf von Neumann (1951) zurückgeht. Deren Ziel ist es, unabhängige Zufallsvariable X aus einer gegebenen Dichtefunktion $p(\cdot)$ zu erzeugen. Die Dichte muß nicht normiert sein, d. h. die Verwerfungsmethode kann auf eine Funktion $g(\cdot) = Cp(\cdot)$ angewandt werden, wobei $C = \int_D g(x)dx$ nicht bekannt sein muß.

Man definiert nun die sogenannte Hutfunktion $h(\cdot)$, für die $h(x) \geq g(x)$ für alle $x \in D$ gilt. Das Integral $\int_D h(x)dx$ sollte leicht berechenbar sein, da aus der Dichtefunktion $h(\cdot)/\int_D h(x)dx$ Zufallszahlen generiert werden müssen.

Wenn die Auswertung der Funktion $g(\cdot)$ einen hohen Rechenaufwand erfordert, empfiehlt es sich, zusätzlich eine leichter berechenbare Squeezefunktion $s(\cdot)$ zu wählen, sodaß $s(x) \leq g(x)$ für alle $x \in D$ gilt. Für die Erzeugung von unabhängigen Zufallsvariablen $X_1, \ldots, X_N$ gemäß der Dichtefunktion $p(\cdot)$ hat der Algorithmus der Verwerfungsmethode die Form:

Setze $t = 1$
Solange $t \leq N$
{
  Erzeuge Y aus $h(\cdot)/\int_D h(x)dx$
  Erzeuge U aus $Uniform(0,1)$
  Falls $U \leq s(Y)/h(Y)$ setze $X_t = Y$ und $t = t + 1$
    Sonst falls $U \leq g(Y)/h(Y)$ setze $X_t = Y$ und $t = t + 1$
}.

Die Verwerfungsmethode ist nur dann sinnvoll, wenn die Erzeugung der Zufallszahlen von $h(\cdot)/\int_D h(x)dx$ einfacher ist als von der Ausgangsdichte $p(\cdot)$. Die Anzahl der Verwerfungen bei der Ausführung des Algorithmus hängt von der Güte der Anpassung der Hutfunktion $h(\cdot)$ an die Funktion $g(\cdot)$, die durch die Effizienz α mit

$$\alpha = \frac{\int_D h(u)du}{\int_D g(v)dv} \tag{3.2}$$

gemessen wird, ab. Für die Effizienz gilt $\alpha \geq 1$, und je näher sich α eins nähert, umso weniger Verwerfungen sind zu erwarten.

3.2 Die adaptive Verwerfungsmethode

Die Effizienz der Verwerfungsmethode kann durch die sukzessive Modifikation der Hutfunktion während der Ausführung des adaptiven Verwerfungsalgorithmus gesteigert werden. Durch die Forderung der Logkonkavität an die Dichtefunktion $p(\cdot)$ ist es nicht nötig, zur Bestimmung der Hutfunktion den Modus von $p(\cdot)$ zu berechnen. Außerdem wird mit jeder Verwerfung und der einhergehenden Adaption der Hutfunktion die Wahrscheinlichkeit einer nochmaligen Verwerfung immer kleiner.

Der Definitionsbereich $D \subseteq \mathbb{R}$ von $p(\cdot)$ sei zusammenhängend und $g(\cdot)$ differenzierbar auf ganz D. Die Funktion $l(x) = \log g(x)$ sei logkonkav auf D, d. h. $l'(x) = \mathrm{d}l(x)/\mathrm{d}x$ ist monoton fallend. Diese Definition läßt sowohl Geradensegmente für $l(x)$ als auch Unstetigkeitsstellen für $l'(x)$ zu. Um den Verwerfungsschritt durchführen zu können, muß eine Hutfunktion gefunden werden. Zu deren Konstruktion stehen für die adaptive Verwerfungsmethode die folgenden zwei Verfahren zur Verfügung:

3.2.1 Die Hutfunktion nach der Tangentenmethode

Den Ausgangspunkt der Konstruktion stellt die Menge $T_r = \{x_i \in D : i = 1, \ldots, r\}$ von r Stützpunkten dar, für die $x_1 \leq x_2 \leq \ldots \leq x_r$ gilt. In diesen Punkten werden die Funktionen $l(x)$ und $l'(x)$ ausgewertet. Damit die Tangentenmethode angewendet werden kann, muß die Menge T_r eingangs *mindestens zwei Stützpunkte* enthalten. Man definiert dann eine Hutfunktion auf T_r durch $\exp[h_r(x)]$, wobei $h_r(x)$ eine stückweise lineare Funktion ist. $h_r(x)$ wird durch die Tangenten an die Funktion $l(x)$ in den Punkten von T_r festgelegt. Die Schnittpunkte z_j der Tangenten an x_j und x_{j+1} für $j = 1, \ldots, r-1$ erhält man aus

$$l(x_j) + l'(x_j)(z_j - x_j) = l(x_{j+1}) + l'(x_{j+1})(z_j - x_{j+1}),$$

und damit

$$z_j = \frac{l(x_{j+1}) - l(x_j) - x_{j+1}l'(x_{j+1}) + x_j l'(x_j)}{l'(x_j) - l'(x_{j+1})}.$$

Es seien z_0 die untere Schranke von D (oder $-\infty$, falls D nach unten unbeschränkt ist) und z_r die obere Schranke von D (oder ∞, falls D nach oben unbeschränkt ist). Für $x \in [z_{j-1}, z_j]$ und $j = 1, \ldots, r$ definiert man die auf dem ganzen Definitionsbereich D stetige Hutfunktion durch

$$h_r(x) = l(x_j) + (x - x_j)l'(x_j).$$

Wegen der Konkavität von $l(x)$ gilt $l(x) \leq h_r(x)$ für alle $x \in D$.

In Abbildung 3.1 ist die Hutfunktion, die gemäß der Tangentenmethode konstruiert wurde, gemeinsam mit der logarithmierten Dichte der verallgemeinerten Inversen Gaußverteilung (GIG-Verteilung) (s. Abschnitt 4.1) dargestellt. Die Graphik wurde mit dem Programm `ARS` , das ausführlicher in Abschnitt 4.2 vorstellt wird, erzeugt. Abbildung 3.2 zeigt die Dichte und die Hutfunktion nach der Rücktransformation durch die Exponentialverteilung.

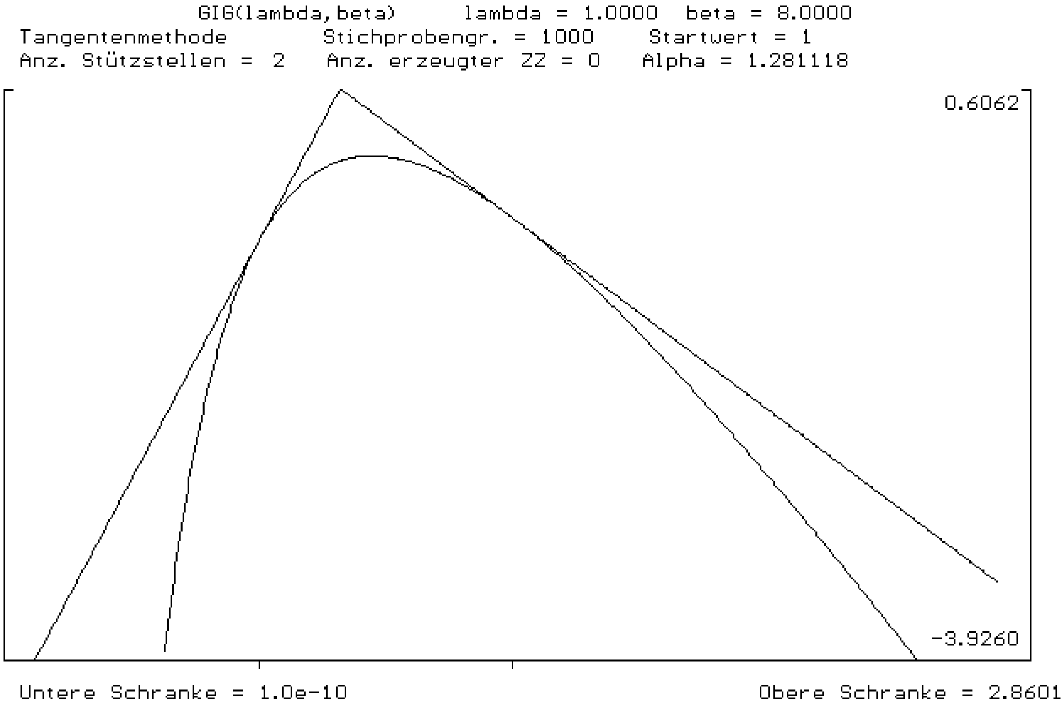


Abbildung 3.1: *Darstellung der logarithmierten Dichte und der Hutfunktion gemäß der Tangentenmethode mit* $r = 2$ *Stützstellen. Die Positionen der Stützstellen sind in der Graphik durch Marker an der Abszissenachse gekennzeichnet.*

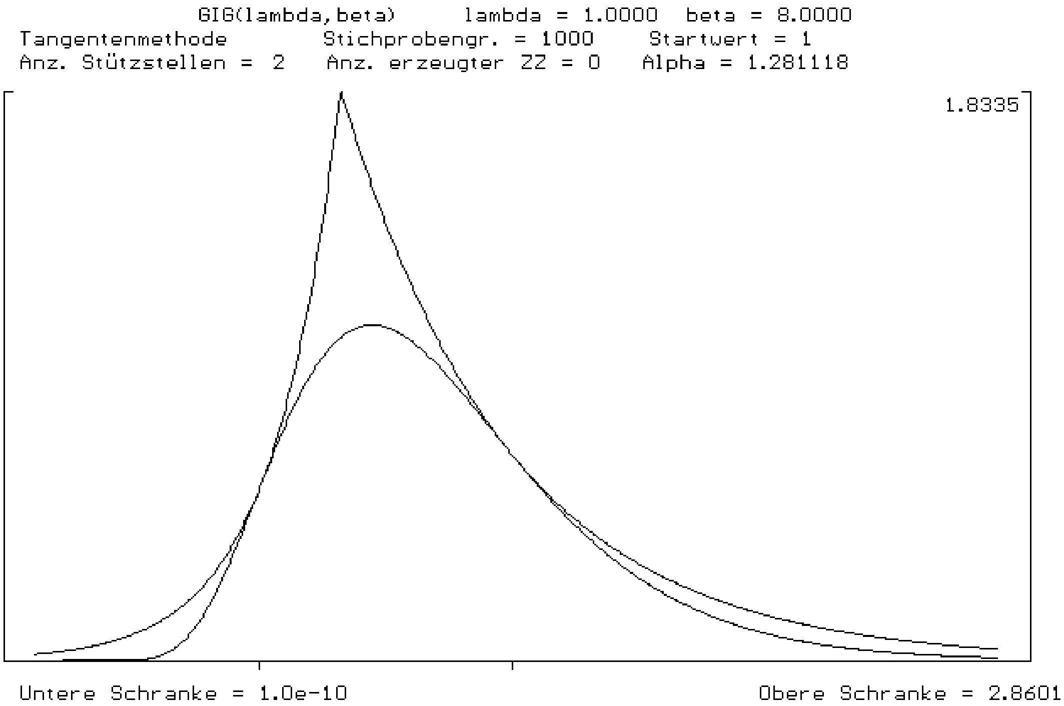


Abbildung 3.2: *Darstellung der Dichte und der Hutfunktion aus Abbildung 3.1 nach der Rücktransformation mit der Exponentialfunktion.*

3.2.2 Die Hutfunktion nach der Sekantenmethode

Auch hier geht man von einer Menge $T_r = \{x_i \in D : i = 1, \ldots, r\}$ mit $x_1 \leq x_2 \leq \ldots \leq x_r$ aus, auf der die Funktion $l(x)$ ausgewertet wird. Im Gegensatz zur Tangentenmethode muß die Menge T_r zu Beginn *mindestens drei Stützpunkte* enthalten. Zusätzlich zur Menge T_r legt man z_0 und z_r mit den unteren und oberen, nicht notwendigerweise endlichen Grenzen von D fest. Um die Berechnung der Ableitungen $l'(x)$ auf T_r zu vermeiden, konstruiert man die Hutfunktion mit Hilfe von Sekanten, die die Funktion $l(x)$ über den Punkten der Menge T_r schneiden. Die lineare Funktion $L_{ij}(x; T_r)$ sei für $1 \leq i \leq j \leq m$ die Gerade durch die Punkte $[x_i, l(x_i)]$ und $[x_j, l(x_j)]$. Für alle anderen Kombinationen von (i, j) sei $L_{ij}(x; T_r)$ nicht definiert. Man erhält dann eine stückweise lineare Hutfunktion

$$h_r(x) = \min\{L_{i-1,i}(x; T_r), L_{i+1,i+2}(x; T_r)\}, \qquad x_i \leq x \leq x_{i+1} \tag{3.3}$$

für $i = 0, \ldots, r$. Für das Minimum in (3.3) soll gelten:

$$\min\{a, b\} = \min\{b, a\} = a,$$

wenn der Ausdruck b nicht definiert ist.

Aufgrund der Logkonkavität von $p(\cdot)$ läßt sich leicht verifizieren, daß $h_r(x)$ eine Hutfunktion von $l(x)$ ist, d. h. es gilt $h_r(x) \geq l(x)$ für alle $x \in D$. Man beachte, daß die Funktion $h_r(x)$ in den Punkten x_1 und x_r nicht stetig ist.

In Abbildung 3.3 ist die Dichte der GIG-Verteilung gemeinsam mit der Hutfunktion gemäß der Sekantenmethode für $r = 4$ Stützstellen dargestellt.

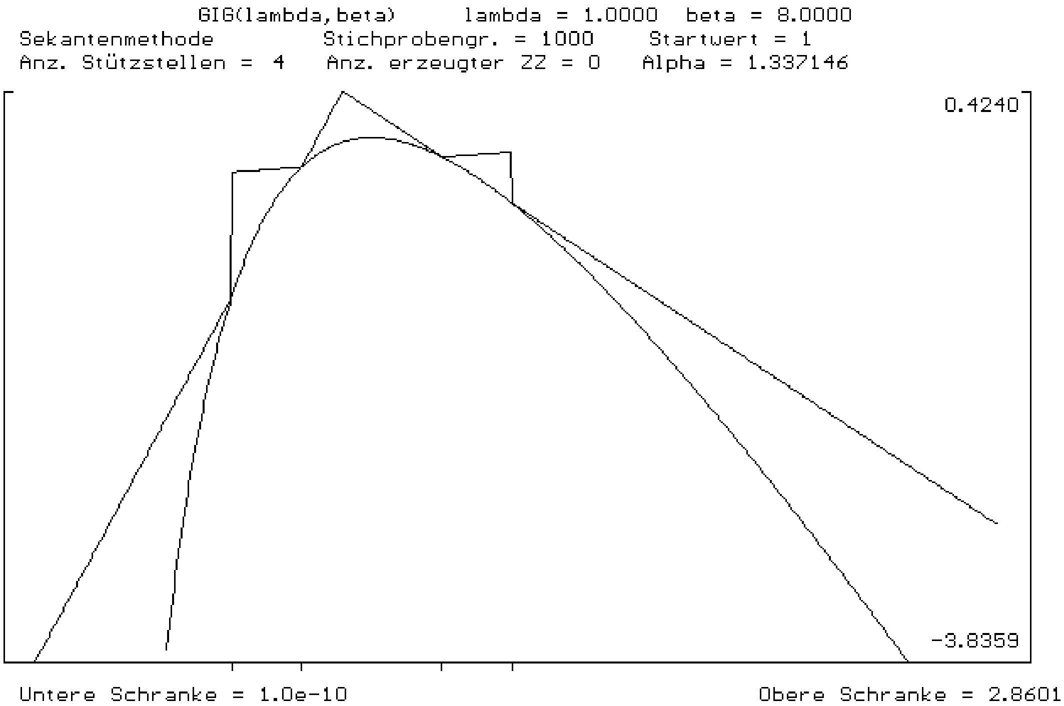


Abbildung 3.3: *Darstellung der logarithmierten Dichte und der Hutfunktion gemäß der Sekantentenmethode. An den jeweils kleinsten und größten Stützstellen ist die Hutfunktion nicht stetig.*

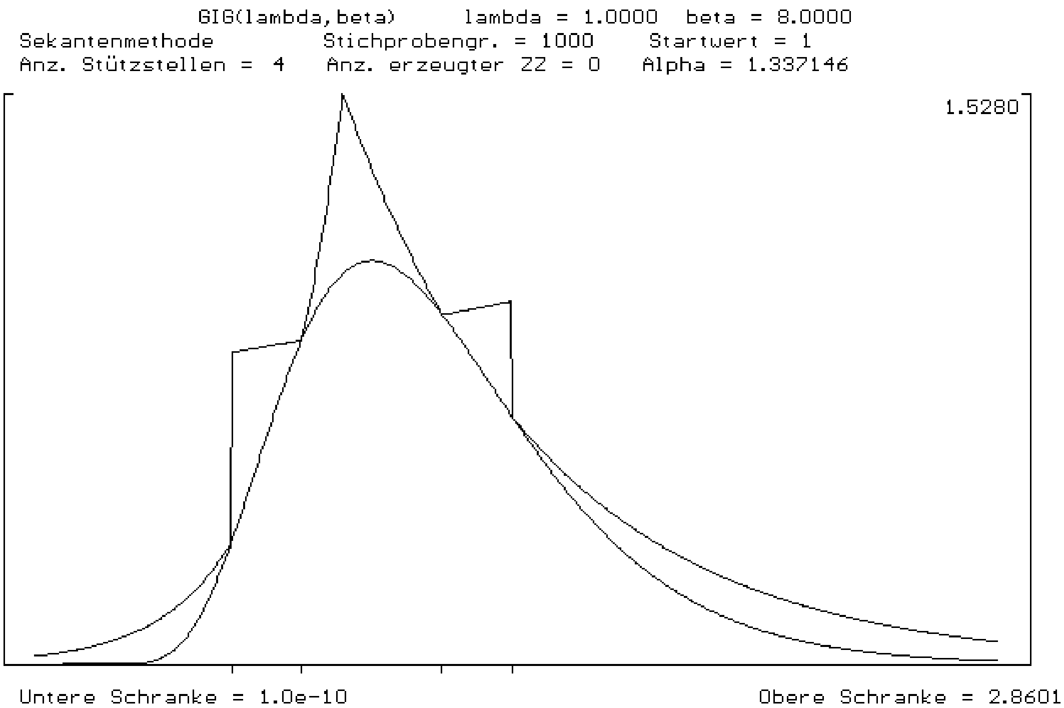


Abbildung 3.4: *Darstellung der Dichte und der Hutfunktion aus Abbildung 3.3 nach der Rücktransformation mit der Exponentialfunktion.*

3.2.3 Der Algorithmus

Vor der Ausführung der adaptiven Verwerfungsmethode wird die Menge T_r festgelegt. Sie muß, wie bereits erwähnt, bei Anwendung der Tangentenmethode mindestens zwei, bei Benutzung der Sekantenmethode mindestens drei Punkte enthalten. Es ist zu beachten, daß bei beiden Methoden jeweils ein Punkt links bzw. rechts des Modus von $l(x)$ gewählt werden muß. Eine exakte Bestimmung des Modus ist aufgrund der Eigenschaft der Log-konkavität von $p(\cdot)$ jedoch nicht notwendig. Es genügt, die zwei Randpunkte x_1 und x_r so zu bestimmen, daß bei Benutzung der Tangentenmethode $l'(x_1) > 0$ und $l'(x_r) < 0$ gilt und bei Anwendung der Sekantenmethode die Steigung der Gerade $L_{1,2}(x;T_r)$ strikt positiv und die Steigung von $L_{r-1,r}(x;T_r)$ strikt negativ ist.

Um beim Verwerfungsschritt die eventuell aufwendige Auswertung von $l(x)$ zu vermeiden, kann man eine Squeezefunktion $s_r(x)$ definieren, indem man die Funktionswerte der Punkte aus T_r durch Gerade verbindet. Für $x \in [x_j, x_{j+1}]$ ist

$$s_r(x) = \frac{(x_{j+1} - x)l(x_j) + (x - x_j)l(x_{j+1})}{x_{j+1} - x_j}$$

für $j = 1, \ldots, r-1$. Für $x < x_1$ oder $x > x_r$ sei $s(x) = -\infty$. Die Squeezefunktion entspricht also den verbundenen Sekanten, die die Funktion $l(x)$ über den Punkten der Menge T_r schneiden.

Die Dichtefunktion, von der Kandidaten zur Durchführung des Verwerfungsschrittes erzeugt werden, erhält man durch Normierung der Hutfunktion:

$$u_r(x) = \frac{\exp[h_r(x)]}{\int_D \exp[h_r(v)]dv}.$$

$u_r(x)$ ist eine stückweise exponentielle Funktion, aus der die Generierung von Zufallszahlen immer möglich ist.

Die adaptive Verwerfungsmethode unter Verwendung der Tangentenmethode kann nun zum Algorithmus ARS (*Adaptive Rejection Sampling*) zusammengefaßt werden, der aus drei Schritten besteht. In eckigen Klammern [...] sind die Modifikationen, die bei Verwendung der Sekantenmethode durchzuführen sind, angegeben.

1. *Initialisierung*: Zunächst wird die Menge T_r festgelegt, wobei $r \geq 2$ $[r \geq 3]$. Falls D nach unten unbeschränkt ist, dann ist x_1 so zu wählen, daß $l'(x_1) > 0$ $[\frac{\mathrm{d}L_{1,2}(x;T_r)}{\mathrm{d}x} > 0]$. Falls D nach oben unbeschränkt ist, dann ist x_r so zu wählen, daß $l'(x_r) < 0$ $[\frac{\mathrm{d}L_{r-1,r}(x;T_r)}{\mathrm{d}x} < 0]$ ist. Für die r Punkte der Menge T_r sind die Funktionen $h_r(x)$, $u_r(x)$ und $s_r(x)$ auszuwerten.

2. *Verwerfungsschritt*:

$$\begin{array}{l} \texttt{Erzeuge}\ \ Y\ \texttt{aus}\ u_r(\cdot) \\ \texttt{Erzeuge}\ \ U\ \texttt{aus}\ Uniform(0,1) \\ \texttt{Falls}\ \ U \leq \exp[s_r(Y) - h_r(Y)] \quad \texttt{akzeptiere}\ Y \\ \quad \texttt{Sonst falls}\quad U \leq \exp[l(Y) - h_r(Y)] \quad \texttt{akzeptiere}\ Y \\ \qquad \texttt{Sonst verwerfe}\ \ Y. \end{array} \tag{3.4}$$

3. *Adaption der Hutfunktion*: Falls $l(Y)$ und $l'(Y)$ $[l(Y)]$ im Zuge des Verwerfungsschrittes ermittelt wurden, wird Y in T_{r+1} aufgenommen. Die Elemente von T_{r+1} werden in aufsteigender Reihenfolge geordnet, die Funktionen $h_{r+1}(x)$, $u_{r+1}(x)$ und $s_{r+1}(x)$ ermittelt und r um eins erhöht. Sollen noch weitere unabhängige Zufallszahlen gemäß der Dichte $p(\cdot)$ erzeugt werden, dann ist mit dem Verwerfungsschritt fortzufahren.

Im dritten Schritt des angegebenen Algorithmus wird nach jeder Verwerfung die Menge der Stützpunkte um ein Element vergrößert. Die neu berechnete Hutfunktion $h_{r+1}(x)$ approximiert die logarithmierte Dichtefunktion $l(x)$ besser als $h_r(x)$. Dadurch wird die Wahrscheinlichkeit einer neuerlichen Verwerfung im nächsten Schritt reduziert.

Die Startwerte der Stützpunkte für die Menge T_r müssen nicht fest sein. Wenn $g(\cdot)$ die vollständig bedingte Verteilung ist, so kann beispielsweise bei Verwendung der Sekantenmethode die Hutfunktion $\exp[h_r(\cdot)]$ aus der vorangegangenen Iteration des Gibbs-Erzeugungsprozesses zur näherungsweisen Bestimmung der 5%-, 50%- und 95%-Quantile der Dichtefunktion $p(\cdot)$ herangezogen werden. Diese können als Startwerte der Menge T_r fungieren.

3.2.4 Die adaptive Verwerfungsmethode in Bayes-Modellen

In der Bayes'schen Analyse werden komplexe Datenmengen oft durch n Submodelle beschrieben, wobei das i-te Submodell, $i = 1, \ldots, k$, durch eine bedingte Dichtefunktion $\pi(X_{.i}|\{X_{.j}; j \in S_i\})$ bezüglich der eindimensionalen Modellparameter $X_{.1}, \ldots, X_{.k}$ beschrieben wird. Die Menge S_i enthält die für das Submodell i benötigten Parameter.

Wenn es sich um ein hierarchisches Modell handelt, so erhält man die vollständig bedingten Verteilungen aus den bedingten Dichtefunktionen der Submodelle (Spiegelhalter *et al.*, 1996):

$$\begin{aligned} \pi(X_{.i}|X_{.-i}) &\propto \pi(X_{.i}, X_{.-i}) = \pi(X) \\ &\propto \pi(X_{.i}|\{X_{.j}; j \in S_i\}) \prod_{\{l:i \in S_l\}} \pi(X_{.l}|\{X_j; j \in S_l\}). \end{aligned} \tag{3.5}$$

Mit Hilfe des Gibbs-Generators können nun Stichproben aus der gemeinsamen Dichtefunktion aller Parameter $\pi(X_{.1}, \ldots, X_{.k})$ oder aus bestimmten Randverteilungen, die von Interesse sind, erzeugt werden.

Um die adaptive Verwerfungsmethode anwenden zu können, ist für die Dichte, von der Zufallsgrößen erzeugt werden sollen, deren Logkonkavität vorauszusetzen. Um diese Eigenschaften zu garantieren, können während der Modellbildung bestimmte Kombinationen von Verteilungen für die Submodelle gewählt werden (Spiegelhalter *et al.*, 1996). Wenn keine Logkonkavität vorliegt, kann diese Eigenschaft eventuell durch eine geeignete Transformation des Parameters der Dichtefunktion erreicht werden.

Logarithmiert man die vollständig bedingten Verteilungen (3.5) des hierarchischen Modells, so erhält man

$$l(X_{.i}) = \log \pi(X_{.i}|\{X_{.j}; j \in S_i\}) + \sum_{\{l:i \in S_l\}} \log \pi(X_{.l}|\{X_j; j \in S_l\}). \tag{3.6}$$

Wenn alle Terme auf der rechten Seite von (3.6) konkav bezüglich $X_{.i}$ sind, so ist dies auch $l(X_{.i})$ als Summe von konkaven Funktionen.

Der Gibbs-Erzeugungsprozeß besteht in der blockweisen Simulation von vollständig bedingten Verteilungen der Form (3.1). Oft haben die Blöcke $X_{.i}$ des Zufallsvektors $X = (X_{.1}, \ldots, X_{.k})$ in (3.5) und (3.6) eine Dimension größer als eins. Die adaptive Verwerfungsmethode wurde bis jetzt nur für eindimensionale Zufallsvariable $X_{.i}$ beschrieben. Sie kann jedoch auch auf den Fall, daß $X_{.i}$ multivariat ist, angewendet werden, wie die folgende Überlegung zeigt: Ist $X_{.ij}$ eine Komponente von Block $X_{.i}$, so gilt für die eindimensionale vollständig bedingte Verteilung

$$\pi(X_{.ij}|X_{.-ij}) \propto \pi(X_{.i}|X_{.-i}).$$

Wenn die mehrdimensionale vollständig bedingte Verteilung $\pi(X_{.i}|X_{.-i})$ logkonkav bezüglich $X_{.i}$ ist, so ist es auch $\pi(X_{.ij}|X_{.-ij})$ bezüglich $X_{.ij}$. Der Gibbs-Generator kann nun der Reihe nach auf alle Komponenten $X_{.ij}$ unter Verwendung der adaptiven Verwerfungsmethode angewandt werden.

3.3 Die adaptive Verwerfungsmethode mit Metropolis-Hastings Schritt

Die adaptive Verwerfungsmethode kann nicht direkt angewendet werden, wenn Zufallsgrößen von einer nicht-logkonkaven Dichte $p(\cdot)$ erzeugt werden sollen. In diesem Fall kann man auf den Algorithmus von Metropolis-Hastings zurückgreifen, der abhängige Zufallsgrößen von einer beliebigen Dichte generieren kann. Um eine kleine Verwerfungswahrscheinlichkeit und damit eine schnellere Konvergenz der Markovkette gegen die stationäre Verteilung zu garantieren, sollte die Vorschlagsdichte $q(x, \cdot)$ eine gute Näherung der Ausgangsdichte $p(\cdot)$ sein.

Die in den Abschnitten 3.2.1 und 3.2.2 vorgestellten Verfahren können für eine beliebige Dichtefunktion $p(\cdot)$ zwar im allgemeinen keine Hutfunktion liefern, aufgrund der Tatsache, daß sie mit sukzessiver Vergrößerung der Menge T_r der Stützpunkte die Dichtefunktion $p(\cdot)$ zusehends besser approximieren, können die genannten Methoden zur Bestimmung der Kandidatendichte $q(x, \cdot)$ herangezogen werden.

Wie in den vorangegangenen Abschnitten muß die Dichte $p(\cdot)$ nicht normiert sein, d. h. es existiert eine Funktion $g(\cdot)$, sodaß $g(x) = Cp(x)$ für alle $x \in D$ gilt. In der Praxis von MCMC-Methoden entspricht die Dichte $g(\cdot)$ der i-ten vollständig bedingten Verteilung $\pi(\cdot|X_{.-i})$, die im allgemeinen ebenfalls nicht normiert ist.

In diesem Abschnitt seien die Blöcke des Zufallsvektor $X = (X_{.1}, \ldots, X_{.k})$ nur eindimensional, d. h. $X_{.i} \in \mathbb{R}$, $i = 1, \ldots, k$. Dann ist $g(\cdot)$ eine Funktion $g : \mathbb{R} \to \mathbb{R}$. Somit ist für die adaptive Verwerfungsmethode mit Metropolis-Hastings-Schritt bei Ausführung eines Schrittes die Erneuerung nur einer einzelnen Komponente $X_{.i} \in \mathbb{R}$ möglich. Im allgemeinen gilt diese Einschränkung für den Gibbs-Generator (s. Abschnitt 1.3.4) nicht.

Daß $g(\cdot)$ von den Modellparametern $X_{.-i}$ abhängig ist, wird in dieser Notation nicht berücksichtigt, da diese Abhängigkeit für die Beschreibung eines Schrittes des Algorithmus nicht erheblich ist. Es bezeichne $X_{t.i}$ den Zustand des i-ten Blocks zum Zeitpunkt t. Ziel des Algorithmus ist es, eine neue Zufallsgröße $X_{t+1.i}$ zu finden, sodaß die stationäre Dichtefunktion der $X_{.i}$ die i-te vollständig bedingte Verteilung ist.

Man definiert in Anlehnung an die Funktion (3.3) gemäß der Sekantenmethode für eine Menge T_r von Stützpunkten eine Funktion

$$h_r(x) = \max\Big\{L_{i,i+1}(x; T_r), \min\{L_{i-1,i}(x; T_r), L_{i+1,i+2}(x; T_r)\}\Big\}, \qquad x_i \leq x \leq x_{i+1}, \quad (3.7)$$

die für ein genügend großes r eine beliebig gute Näherung an die Funktion $\log g(x)$ darstellt. Für das Minimum und Maximum in Gleichung (3.7) soll

$$\min\{a, b\} = \min\{b, a\} = \max\{a, b\} = \max\{b, a\} = a,$$

wenn der Ausdruck b nicht definiert ist. Für nicht-logkonkave Dichtefunktionen ist $h_r(x)$ im allgemeinen keine Hutfunktion. Wenn $p(x)$ logkonkav ist, reduziert sich $h_r(x)$ jedoch zur Hutfunktion $h_r(x)$ nach der Sekantenmethode, wie dies in Abschnitt 3.2.2 beschrieben wurde.

In Abbildung 3.5 ist die Funktion $h_r(\cdot)$ mit der logarithmierten Dichte der Burr $V(r,k)$-Verteilung (Burr, 1942) dargestellt. Die Burr V-Verteilung hat für die Wahl von r und

k in Abbildung 3.5 zwei Modi, weshalb sie nicht logkonkav ist. Man erkennt, daß $h_r(\cdot)$ keine Majorante der logarithmierten Dichtefunktion darstellt. Die Abbildung wurde mit dem Programm `ARMS` erstellt, in dem die adaptive Verwerfungsmethode mit Metropolis-Hastings-Schritt implementiert wurde. Dieses Programm wird später in Abschnitt 4.2 vorgestellt.

Als Dichte, aus der Kandidaten für den adaptiven Verwerfungsschritt generiert werden sollen, definiert man

$$u_r(x) = \frac{\exp[h_r(x)]}{\int_D \exp[h_r(v)]dv}.$$

Diese Funktion ist stückweise exponentiell und stellt für die Erzeugung von Zufallsgrößen kein Problem dar.

Der Algorithmus ARMS (*Adaptive Rejection Metropolis Sampling*) zur Erzeugung von $X_{t+1.i}$ aus $X_{t.i}$ besteht nun aus zwei Schritten:

1. *Adaptiver Verwerfungsschritt*:
   Initialisierung von r und T_r unabhängig von $X_{t.i}$.

   `Erzeuge` $Y_{.i}$ `aus` $u_r(\cdot)$
   `Erzeuge` U `aus` $Uniform(0,1)$
   `Solange` $U > g(Y_{.i})/\exp h_r(Y_{.i})$
   `{`
       `Setze` $T_{r+1} = T_r \cup \{Y_{.i}\}$
       `Ordne die Elemente von` T_{r+1} `in aufsteigender Reihenfolge`
       `Setze` $r = r + 1$
       `Berechne` $h_r(\cdot)$ `und` $u_r(\cdot)$ (3.8)
       `Erzeuge` $Y_{.i}$ `aus` $u_r(\cdot)$
       `Erzeuge` U `aus` $Uniform(0,1)$
   `}.`

2. *Metropolis-Hastings-Schritt*:

   `Erzeuge` U `aus` $Uniform(0,1)$
   `Falls` $U < \min\left\{1, \frac{g(Y_{.i})\min\{g(X_{t.i}),\exp h_r(X_{t.i})\}}{g(X_{t.i})\min\{g(Y_{.i}),\exp h_r(Y_{.i})\}}\right\}$
       `Setze` $X_{t+1.i} = Y_{.i}$
   `Sonst setze` $X_{t+1.i} = X_{t.i}$

Für den Zufallsvektor $X = (X_{.1}, \ldots, X_{.k})$ ist der Algorithmus ARMS für eine Iteration k-mal durchzuführen. Die Markovkette bewegt sich damit vom Zustand X_{t_0} zu X_{t_0+1} für ein $t_0 \in \mathbb{N}_0$.

Es ist zu beachten, daß die Wahl der Anfangselemente der Menge T_r unabhängig von $X_{t.i}$, dem Element des i-ten Blocks aus der vorangegangenen Iteration, erfolgen muß. Ist die Unabhängigkeit nicht gegeben, kann die Konvergenz gegen die gewünschte stationäre Verteilung, nämlich der des Gibbs-Generators, nicht garantiert werden.

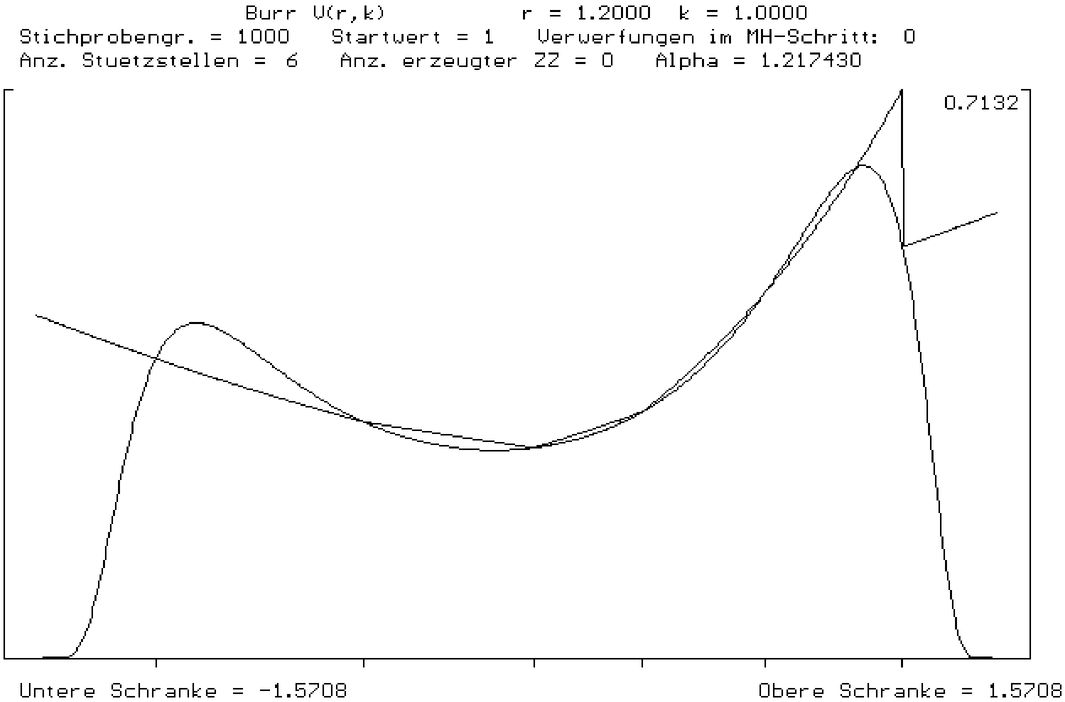


Abbildung 3.5: *Darstellung der logarithmierten Dichte der Burr V-Verteilung und der Hutfunktion gemäß (3.7) für $r = 6$. An den jeweils kleinsten und größten Stützstellen kann die Hutfunktion Sprünge aufweisen (hier ist sie stetig in x_1, aber unstetig in x_6).*

Durch eine geschickte Wahl der Anfangswerte von T_r kann die Wahrscheinlichkeit einer Verwerfung im Metropolis-Hastings-Schritt reduziert werden. Eine gute Wahl von Stützpunkten ist jene im Abschnitt 3.2.3 beschriebene, die auf der Funktion $h_r(\cdot)$ der vorangegangenen Iteration t basiert. Diese Vorgangsweise ist nur dann zulässig, wenn gezeigt werden kann, daß diese Wahl der Stützstellen unabhängig von $X_{t.i}$ ist (ansonsten kann, wie oben beschrieben, die Konvergenz gegen die stationäre Verteilung nicht garantiert werden). Daß die Unabhängigkeit gegeben ist, sollen folgende Überlegungen zeigen: Es sei $r_{t.i}$ der realisierte Wert von r nach dem i-ten Schritt der t-ten Iteration. Es kann gezeigt werden, daß $h_{r_{t.i}}(\cdot)$ und $X_{t.i}$ unabhängig sind (Gilks *et al.*, 1995). Daraus folgt, daß zu Beginn der $t+1$-ten Iteration auch $X_{t.i}$ und T_r unabhängig sind, wenn die Stützstellen mit $T_r = T_{r_{t.i}}$ initialisiert werden.

Die Dichte zur Generierung von Kandidaten $Y_{.i}$ für den Metropolis-Hastings-Schritt ergibt sich aus dem adaptiven Verwerfungsschritt von ARMS. Sie hat die Form

$$q_i(\{X_{t.i}, X_{t.-i}\}, Y_{.i}) = q_i(X_{t.i}, Y_{.i}) \propto \min\{g(Y_{.i}), \exp[h_r(Y_{.i})]\}. \tag{3.9}$$

Die rechte Seite von (3.9) ist aufgrund der Wahl von T_r unabhängig von $X_{t.i}$ und deshalb ist $q(X_{t.i}, Y_{.i}) = q(Y_{.i})$. Die adaptive Verwerfungsmethode mit Metropolis-Hastings-Schritt ist somit ein Beispiel für einen Unabhängigkeitsgenerator (s. Abschnitt 1.3.2).

Um die Wahrscheinlichkeit, daß $X_{.i}$ nach einem Schritt des Algorithmus im gleichen Punkt verharrt, d. h. $X_{t-1.i} = X_{t.i}$, zu reduzieren, kann die i-te Ausführung von ARMS innerhalb einer Iteration n-mal wiederholt werden. n muß konstant sein und darf während verschiedener Iterationen nicht verändert werden. Es ist folgende Vorgangsweise einzuhalten:

Nachdem ARMS für den Block $X_{t.i}$ zum erstenmal durchlaufen wurde, setzt man $r = r_{t.i}$ und $T_r = T_{r_{t.i}}$. Der Block $X_{t.i}$ bleibt unverändert, falls $Y_{.i}$ im Metropolis-Hastings-Schritt verworfen wurde, ansonsten setzt man $X_{t.i} = Y_{.i}$.

Der Algorithmus ARMS ist effizient, wenn die Verwerfungsrate im Metropolis-Hastings-Schritt sehr klein ist. Das gilt vor allem für Dichtefunktionen, deren Form fast logkonkav ist. Ist die Dichte $p(\cdot)$ logkonkav, so wird im zweiten Schritt von ARMS der Kandidat $Y_{.i}$ nie verworfen. In diesem Fall entspricht ARMS der adaptiven Verwerfungsmethode. ARMS kann also Abweichungen von der Logkonkavität bei gleichzeitiger Beibehaltung der Effizienz der adaptiven Verwerfungsmethode berücksichtigen. Ein Nachteil des Algorithmus ist die Tatsache, daß in einem Schritt nur eine Komponente des Zufallsvektor X erneuert werden kann. Oft kann eine Erneuerung von mehr als einer Komponente in einem Schritt bei geschickter Wahl der Kandidatendichte $q(\cdot,\cdot)$ eine schnelle Mischung der resultierenden Markovkette zur Folge haben.

Kapitel 4

Die adaptive Verwerfungsmethode in der Praxis

Im vorangegangenen Kapitel wurde die Theorie der adaptiven Verwerfungsmethode vorgestellt. Der Benutzer des Verfahrens kann zwischen der Tangentenmethode (s. Abschnitt 3.2.1) und der Sekantenmethode (s. Abschnitt 3.2.2) wählen, die beide Universalverfahren zur Erzeugung von unabhängigen Stichproben beliebiger logkonkaven Verteilungen sind. Die Wahl zwischen den beiden Methoden wird im allgemeinen davon abhängen, ob die Ableitung der logarithmierten Dichtefunktion, die für die Tangentenmethode notwendig ist, bekannt und leicht berechenbar ist. Wenn dies nicht der Fall ist, können mit Hilfe der Sekantenmethode Zufallszahlen der entsprechenden Verteilung erzeugt werden, wobei nur die logarithmierte Dichtefunktion spezifiziert werden muß.

Durch die Erweiterung des Algorithmus um einen Metropolis-Hastings-Schritt (s. Abschnitt 3.3) können auch Stichproben von Verteilungen, die nicht logkonkav sind, mit Hilfe eines adaptiven Verwerfungsverfahrens erzeugt werden. Die Realisierungen der Stichprobe sind dann aber nicht unabhängig.

Die besprochenen Verfahren wurden ursprünglich für die Simulation von Markovketten durch den Gibbs-Generator entwickelt, und in dieser Funktion finden sie heute häufig Verwendung. Da jedoch sowohl die Tangentenmethode als auch die Sekantenmethode unabhängige Stichproben für beliebige logkonkave Verteilungen erzeugen, kann ihre Verwendung auch außerhalb von MCMC-Methoden interessant sein. Die adaptive Verwerfungsmethode ist ein Universalverfahren und kann deshalb mit Generatoren von Zufallszahlen, die auf bestimmte Verteilungen zugeschnitten sind, bezüglich der Effizienz im allgemeinen nicht konkurrieren. Da für ihre Implementation lediglich die Spezifikation der logarithmierten Dichtefunktion oder eventuell zusätzlich deren Ableitung nötig ist, kann sie für bestimmte Problemstellungen eine attraktive Alternative sein.

In diesem Kapitel soll die Leistungsfähigkeit der adaptiven Verwerfungsmethode anhand der Generierung von Zufallszahlen aus der Verallgemeinerten Inversen Gaußverteilung untersucht werden. Zu diesem Zweck wurden die Computerprogramme `ARS`, in dem der Algorithmus ARS (3.4) mit der Tangenten- und der Sekantenmethode implementiert wurde, und `ARMS` zum Test des Algorithmus ARMS (3.8) der adaptiven Verwerfungsmethode mit Metropolis-Hastings-Schritt, entwickelt. Die Programme sind so aufgebaut, daß sie leicht zur Erzeugung von Zufallszahlen aus beliebigen Verteilungen erweitert werden können. Details zur Implementierung und der Benutzung der Programme für weitere Verteilungen können im Anhang A nachgelesen werden.

4.1 Die Verallgemeinerte Inverse Gaußverteilung

Die verallgemeinerte inverse Gaußverteilung (GIG-Verteilung) wurde von Barndorff-Nielsen (1977) eingeführt. Für die folgenden Untersuchungen wurde die Darstellung der GIG-Verteilung als Zweiparameterfamilie mit den Parametern λ und β, $\beta \geq 0$, gewählt. Die zugehörige Dichtefunktion hat für diese Parametrisierung die Gestalt

$$f_X(x) = \frac{x^{\lambda-1} e^{-\frac{1}{2}\beta(x^{-1}+x)}}{2K_\lambda(\beta)}, \quad x > 0,$$

wobei K_λ die modifizierte Besselfunktion dritter Art mit Index λ ist und durch

$$K_\lambda(\beta) = \frac{1}{2}\int_{-\infty}^{\infty} \cosh(\lambda u) e^{-\beta \cosh(u)} du$$

definiert ist. Für die Implementation der adaptiven Verwerfungsmethode wurde für die Berechnung von K_λ auf die in der Diplomarbeit von Lehner (1989) zur Verfügung gestellten Funktion zurückgegriffen.

Aufgrund der Eigenschaft der GIG(λ,β)-Verteilung, daß die Zufallsvariable $1/X$ zur selben Verteilungsfamilie mit $-\lambda$ statt λ gehört, genügt es, die Fälle $\lambda \geq 0$ zu betrachten.

Für die Anwendung von adaptiven Verwerfungsverfahren ist die Logkonkavität der Dichtefunktion $f(x)$, von der unabhängige Stichproben erzeugt werden sollen, vorauszusetzen. Daß die GIG(λ,β)-Verteilung diese Eigenschaft für $\lambda \geq 1$ erfüllt, erkennt man, wenn man die logarithmierte Dichtefunktion bezüglich x ableitet:

$$\begin{aligned} \log f(x) &= (\lambda - 1)\log x - \tfrac{1}{2}\beta(\tfrac{1}{x} + x) \\ \frac{d}{dx}\log f(x) &= \tfrac{\lambda-1}{x} - \tfrac{1}{2}\beta(-\tfrac{1}{x^2} + 1) = \tfrac{\lambda-1}{x} + \tfrac{1}{2}\beta(\tfrac{1}{x^2} - 1) \end{aligned}$$

Die erste Ableitung ist für $x > 0$, $\beta \geq 0$ und $\lambda \geq 1$ als Summe zweier monoton fallender Funktionen ebenfalls monoton fallend, womit die Bedingung aus Abschnitt 3.2 für logkonkave Verteilungen erfüllt ist.

Der für die Wahl der Stützpunkte wichtige Modus m kann explizit durch die zwei Parameter λ und β ausgedrückt werden:

$$m = m(\lambda, \beta) = \frac{\lambda - 1 + \sqrt{(\lambda-1)^2 + \beta^2}}{\beta}.$$

Zusätzlich erhält man die Momente der GIG-Verteilung durch

$$\alpha_k = E[X^k] = \frac{K_{\lambda+k}(\beta)}{K_\lambda(\beta)}, \quad k \in \mathbb{N},$$

mit deren Hilfe man die zentralen Momente

$$\mu_k = E\Big[(X - E[X])^k\Big],$$

berechnen kann. Die Verteilungskennzahlen der GIG-Verteilung können schließlich durch

$$\begin{aligned} \text{Erwartungswert } \mu &= \alpha_1 \\ \text{Standardabweichung } \sigma &= \sqrt{\mu_2} \\ \text{Schiefe } \gamma_1 &= \frac{\mu_3}{\mu_2^{3/2}} \\ \text{Kurtosis } \gamma_2 &= \frac{\mu_4}{\mu_2^2} - 3. \end{aligned} \tag{4.1}$$

bestimmt werden.

In den nächsten Abschnitten werden die Ergebnisse, die bei der Generierung von Stichproben der GIG(λ,β)-Verteilung mit Hilfe der adaptiven Verwerfungsmethode erzielt wurden, präsentiert.

4.2 Implementierung der adaptiven Verwerfungsmethode

In den beiden Programmen `ARS` und `ARMS` wurden die in Kapitel 3 vorgestellten Algorithmen der adaptiven Verwerfungsmethode zur Untersuchung der Leistungsfähigkeit bei der Erzeugung von Stichproben der GIG(λ,β)-Verteilung implementiert. Durch einfache Modifikationen der Programme (s. Anhang A) können auch andere Verteilungen simuliert werden.

`ARS` und `ARMS` haben die gleiche graphische Oberfläche. Über ein Hauptmenü können Eingabe- und Ausgabefunktionen angesprochen werden. Die Eingabe erfolgt über Dialogboxen (für numerische Werte) und über Submenüs (für die Auswahl der Verteilung oder der gewünschten Methode).

Für die Ausgabe stehen mehrere Graphiken zur Auswahl. Die erzeugte Stichprobe wird in einem Histogramm (s. Abbildung 4.1) klassifiziert und die theoretischen Momente (Erwartungswert, Standardabweichung, Schiefe, Kurtosis) mit den empirischen verglichen. Sollten für die gewünschte Verteilung keine theoretischen Momente existieren, werden in diesem Ausgabefenster die theoretischen Quantile $(q_{0.10}, q_{0.25}, q_{0.50}, q_{0.75}, q_{0.90})$ deren Schätzungen gegenübergestellt. Neben dem Histogramm kann in anderen Fenstern die Hutfunktion entweder in ihrer ursprünglichen Form $h(\cdot)$ (s. Abbildung 4.2), oder nach der Rücktransformation in der Form $\exp[h(\cdot)]$ jeweils im Vergleich mit der theoretischen Dichtefunktion betrachtet werden. In den Darstellungen der Hutfunktion wird gleich wie in der Histogrammausgabe eine Stichprobe der in der Eingabe spezifizierten Größe erzeugt, sodaß die Funktion $h(\cdot)$ genau jener entspricht, die im Histogrammodus verwendet wird. Damit die Adaption von $h(\cdot)$ verfolgt werden kann, werden die Graphiken nach jeder Verwerfung angehalten.

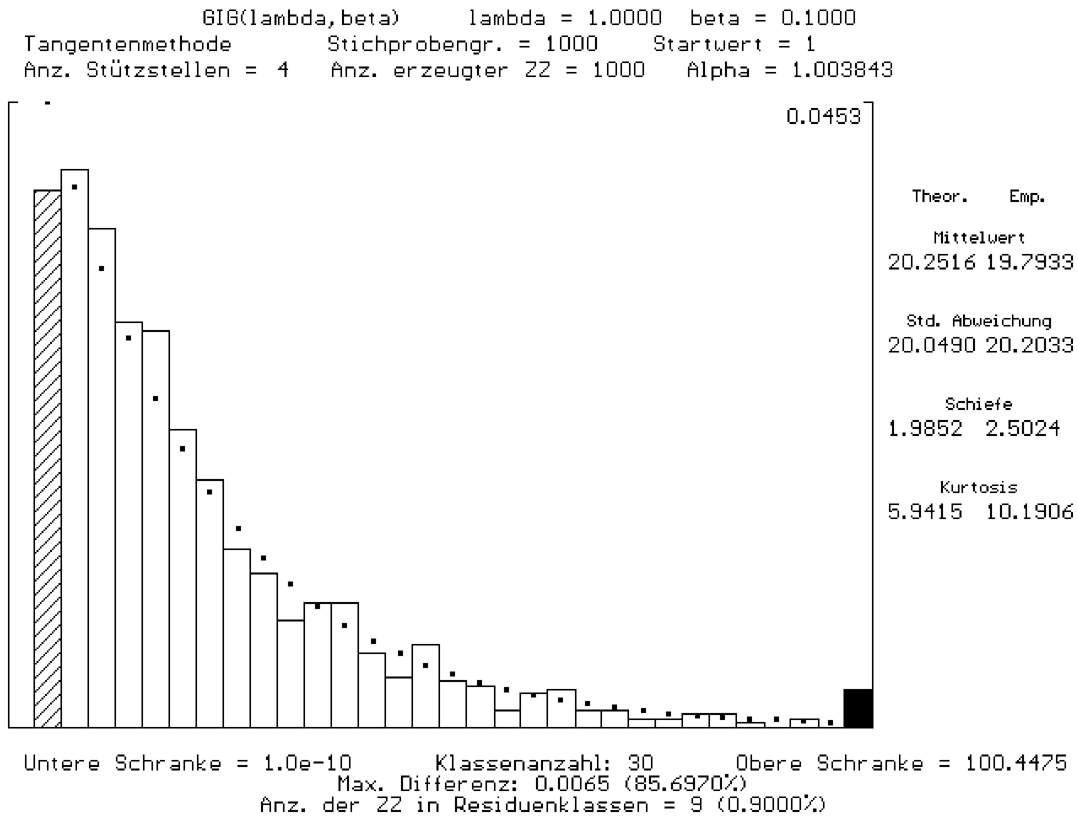


Abbildung 4.1: *Histogrammdarstellung mit dem Programm* `ARS` .

Folgende Auswahlmöglichkeiten können im Eingabebereich getroffen werden bzw. werden während und nach einer Simulation im Ausgabebereich angezeigt:

1. **Eingabebereich:**

   (a) *Verteilung:* `ARS` und `ARMS` können Stichproben beliebiger Verteilungen erzeugen. Der Benutzer kann noch weitere Verteilungen hinzufügen (s. Anhang A).

   (b) *Methode:* Im Programm `ARS` kann zur Erzeugung der Stichprobe zwischen der Tangenten- und der Sekantenmethode gewählt werden.

   (c) *Parameter:* Für jede Verteilung können bis zu zwei Gestaltparameter festgelegt werden.

   (d) *Stichprobengröße:* Durch diesen Wert wird festgelegt, wie viele Zufallszahlen der gewünschten Verteilung erzeugt werden sollen. Die graphische Darstellung der Hut- und Squeezefunktionen, die für jede Adaption neu gezeichnet werden, erfolgt solange, bis die gewählte Stichprobengröße erreicht ist.

   (e) *Startwert:* Der Startwert des Gleichverteilungsgenerators kann vom Benutzer festgelegt werden.

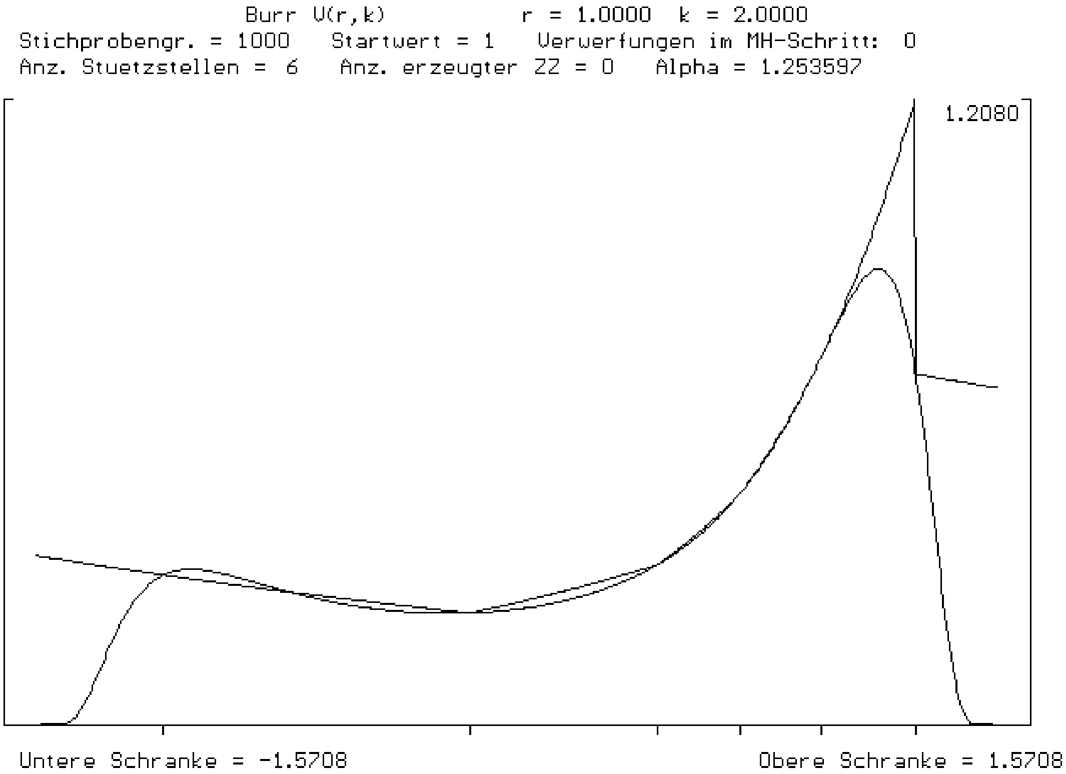


Abbildung 4.2: *Graphik des Programms* **ARMS** *der Kandidatendichte $h(\cdot)$, die für die Dichtefunktion $g(\cdot)$ der Burr V-Verteilung gemäß der adaptiven Verwerfungsmethode mit Metropolis-Hastings-Schritt konstruiert wurde.*

2. Ausgabereich:

(a) *Effizienz α:* Die Effizienz α von Verwerfungsmethoden (s. a. (3.2)), die durch

$$\alpha = \frac{\int_D h(u)du}{\int_D g(v)dv}$$

definiert ist ($g(\cdot)$ ist die nicht-normierte Dichtefunktion, von der eine Stichprobe erzeugt werden soll), gibt die Güte der Anpassung der Hutfunktion an die Dichtefunktion an.

(b) *Anzahl der Stützstellen:* Es wird die Anzahl r der Stützstellen angegeben, durch die die Funktion $h_r(\cdot)$ zum Zeitpunkt der Ausgabe der Graphik definiert ist.

(c) *Verwerfungen im MH-Schritt* (nur für `ARMS`): Die Verwerfungen, die im Metropolis-Hastings-Schritt erfolgen, werden gezählt und angezeigt.

(d) *Anzahl der bisher erzeugten Zufallszahlen:* Die Erzeugung der Zufallszahlen kann schrittweise bis zur jeweils nächsten Verwerfung durchgeführt werden. Im Graphikfenster der Hut- und Squeezefunktion kann dadurch die Adaption, die nach jeder Verwerfung durchgeführt wird, beobachtet werden. Eine Simulation ist dann beendet, wenn die Anzahl der bisher erzeugten Zufallszahlen gleich der Stichprobengröße ist.

(e) *Momente:* Die theoretischen Momente werden gemäß (4.1) berechnet. Zum Vergleich werden die empirischen Momente der Stichprobe von der ersten bis zur zuletzt erzeugten Zufallszahl angezeigt. Wenn die theoretischen Momente nicht existieren, werden stattdessen die theoretischen und empirischen Quantile angezeigt.

(f) *Anzahl der Klassen:* Die Anzahl der Klassen wird nur im Histogrammfenster angegeben.

(g) *Obere und untere Schranke:* Diese Werte sind die Grenzen, innerhalb derer das Histogramm bzw. die Hut- und Squeezefunktionen gezeichnet werden.

(h) *Maximale Differenz:* Damit die Histogrammklassen mit der theoretischen Dichtefunktion verglichen werden können, ist der Wert der theoretischen Dichte im Klassenmittelpunkt durch einen Punkt skizziert. Die Klasse, die am meisten vom theoretischen Wert abweicht, ist in der Graphik schraffiert dargestellt. Außerdem ist der Ordinatenwert der Klasse mit maximaler Differenz absolut und relativ (in Prozent) zum theoretischen Wert angegeben.

(i) *Anzahl der Zufallszahlen in den Residuenklassen:* Liegen Zufallszahlen außerhalb der oberen bzw. unteren Schranke, dann werden sie im Histogramm in der oberen bzw. unteren Restklasse zusammengefaßt. Die beiden Restklassen sind, um sie von den anderen Klassen zu unterscheiden, gefüllt gezeichnet. Die Anzahl der Werte in Restklassen wird sowohl als Absolutwert als auch prozentuell zur bisher erzeugten Anzahl von Zufallszahlen angegeben.

Die Analyse der konkurrierenden Verfahren der adaptiven Verwerfungsmethode wird für unterschiedliche Kombinationen der Gestaltparameter λ und β der GIG-Verteilung durchgeführt. Es ist zu beachten, daß $\lambda \geq 1$ gelten muß, damit die GIG(λ,β)-Verteilung logkonkav ist. Für β sind nur positive Werte zugelassen. Um das Adaptionsverhalten der unterschiedlichen Methoden zu studieren, wurden zur Untersuchung Parameterkombinationen gewählt, für die die GIG-Verteilung möglichst unterschiedliche Gestalten aufweist. Für die Kombination $\lambda = 1$, $\beta = 0.1$ ist die Verteilung beispielsweise extrem rechtsschief, während sie für $\lambda = 8.0$, $\beta = 8.0$ fast symmetrisch ist.

4.3 Analyse der Tangentenmethode

4.3.1 Initialisierung der Stützstellen

Wie in Abschnitt 3.2.1 erklärt wurde, müssen zur Initialisierung der Tangentenmethode $r = 2$ Stützpunkte der Menge T_2 bestimmt werden. Die beiden Punkte x_1 und x_2 müssen strikt positive bzw. negative Steigung der Tangente an die Dichtefunktion aufweisen. Da die zu betrachtende Verteilung logkonkav ist, ist dies gleichbedeutend mit der Bestimmung von je einem Punkt links und rechts vom Modus m. Um die Anzahl der Verwerfungen gering zu halten, sollten diese Punkte so gewählt werden, daß die Effizienz α möglichst nahe bei eins liegt. In der Praxis von MCMC-Methoden stellt sich diese Frage im allgemeinen nicht, da der Modus in vielen Problemstellungen nicht bekannt ist. In diesen

		$\lambda=1$			$\lambda=2$			$\lambda=8$		
		$\beta=0.1$	$\beta=1$	$\beta=8$	$\beta=0.1$	$\beta=2$	$\beta=8$	$\beta=0.1$	$\beta=2$	$\beta=8$
$x_1^{(1)}$	$x_2^{(1)}$	1.0101	1.1847	2.0607	1.3372	1.4567	2.0707	1.8940	1.9274	1.9581
	$x_2^{(2)}$	1.0083	1.1565	2.4921	1.2216	1.4141	2.5086	2.1198	2.1904	2.2589
	$x_2^{(3)}$	1.0086	1.1800	3.3440	1.2706	1.5506	3.4184	2.8598	2.9954	3.1328
$x_1^{(2)}$	$x_2^{(1)}$	1.0100	1.1602	1.4666	1.3135	1.3569	1.4739	1.5056	1.5066	1.5080
	$x_2^{(2)}$	*1.0081*	*1.1123*	1.3558	1.1607	*1.2236*	1.3481	*1.3216*	*1.3225*	*1.3217*
	$x_2^{(3)}$	1.0084	1.1232	1.4735	1.1692	1.2611	1.4606	1.4104	1.4113	1.4084
$x_1^{(3)}$	$x_2^{(1)}$	1.0102	1.1802	1.5025	1.3291	1.4009	1.5231	1.5815	1.5861	1.6215
	$x_2^{(2)}$	1.0082	1.1205	*1.3020*	1.1576	1.2291	*1.3119*	1.3295	1.3341	1.3694
	$x_2^{(3)}$	1.0084	1.1234	1.3106	*1.1453*	1.2243	1.3178	1.3252	1.3298	1.3655

Tabelle 4.1: *Die Werte der Effizienz α für verschiedene Parameterkombinationen der GIG(λ,β)-Verteilung. Die Stützstellen wurden durch $x_1^{(i)} = \max\{\frac{i}{4}m, m - \frac{4-i}{2}\sigma\}$, $i = 1,2,3$ und $x_2^{(j)} = m + \frac{j}{2}\sigma$, $j = 1,2,3$ bestimmt. Die kleinsten α-Werte für jede Parameterkombination sind kursiv dargestellt.*

Fällen werden über Suchalgorithmen zwei Abszissen bestimmt, in denen die Steigung der Tangente an die Dichtefunktion für einen Punkt negativ, für den anderen positiv ist.

Für die GIG(λ,β)-Verteilung kann der Modus explizit angegeben werden. Die Überlegung, welche Startwerte für T_r zu wählen sind, beschränkt sich auf die Frage, wie weit sie vom Modus entfernt sein sollen. Gilks (1992) schlägt für symmetrische Verteilungen aufgrund von Simulationsergebnissen vor, die beiden Punkte mit jeweils dem Abstand einer Standardabweichung σ vom Modus zu wählen. Die GIG(λ,β)-Verteilung ist für kleine Werte der Parameter λ und β stark rechtsschief. Bei einer Initialisierung des linken Stützpunktes durch $x_1 = m - \sigma$ würde der Punkt außerhalb des Definitionsbereichs, der nur positive Werte zuläßt, liegen. Deshalb ist folgende Wahl vorzuziehen:

$$x_1 = \max\left\{\tfrac{1}{2}m, m - \sigma\right\} \tag{4.2}$$

Da der Definitionsbereich der Verteilung $(0,\infty)$ ist, kann der rechte Stützpunkt immer durch

$$x_2 = m + \sigma \tag{4.3}$$

initialisiert werden.

In Tabelle 4.1 wurde die Effizienz α für verschiedene Initialisierungen von T_r und unterschiedliche Parameterkombinationen berechnet. Man erkennt, daß die genannte Wahl für x_1 und x_2 gemäß (4.2) und (4.3) annähernd optimal ist. Man beachte, daß asymmetrische Initialisierungen, in denen ein Punkt näher, der andere weiter vom Modus liegt, ebenfalls keine Verringerung von α zur Folge haben.

Tabelle 4.1 zeigt zudem, daß das Verhalten der adaptiven Verwerf¡ungsmethode stark von der Gestalt der Verteilung abhängt. Sind die Parameter der GIG(λ,β)-Verteilung klein und die Verteilung deshalb stark rechtsschief, können mit zwei Stützstellen im allgemeinen kleinere Werte für α erzielt werden.

Zu einer genaueren Lokalisierung des Optimums wurden weitere Initialisierungen untersucht, die sich um die Punkte gemäß (4.2) und (4.3) konzentrieren. Die in Tabelle 4.2

		$\lambda=1$			$\lambda=2$			$\lambda=8$		
		$\beta=0.1$	$\beta=1$	$\beta=8$	$\beta=0.1$	$\beta=2$	$\beta=8$	$\beta=0.1$	$\beta=2$	$\beta=8$
$x_1^{(1)}$	$x_2^{(1)}$	1.0083	1.1215	1.4632	1.1941	1.2702	1.4485	1.3976	1.3982	1.3910
	$x_2^{(2)}$	1.0082	1.1201	1.4986	1.1751	1.2674	1.4781	1.4038	1.4043	1.3951
	$x_2^{(3)}$	1.0082	1.1247	1.5658	1.1760	1.2860	1.5410	1.4483	1.4488	1.4374
$x_1^{(2)}$	$x_2^{(1)}$	1.0083	1.1160	1.3497	1.1845	1.2375	1.3463	1.3368	1.3377	1.3377
	$x_2^{(2)}$	*1.0081*	1.1123	1.3558	1.1607	1.2236	1.3481	1.3216	1.3225	1.3217
	$x_2^{(3)}$	1.0082	1.1150	1.3909	1.1566	1.2307	1.3802	1.3415	1.3424	1.3408
$x_1^{(3)}$	$x_2^{(1)}$	1.0083	1.1177	1.2949	1.1824	1.2315	1.2999	1.3171	1.3188	1.3272
	$x_2^{(2)}$	*1.0081*	*1.1122*	*1.2811*	1.1548	1.2096	*1.2827*	*1.2868*	*1.2885*	*1.2967*
	$x_2^{(3)}$	1.0082	1.1132	1.2946	*1.1470*	*1.2085*	1.2940	1.2895	1.2912	1.2992

Tabelle 4.2: *Die Werte der Effizienz α für verschiedene Parameterkombinationen der GIG(λ,β)-Verteilung. Die Stützstellen wurden durch $x_1^{(i)} = \max\{\frac{3+i}{10}m, m - \frac{14-2i}{10}\sigma\}$, $i = 1,2,3$ und $x_2^{(j)} = m + \frac{6+2j}{10}\sigma$, $j = 1,2,3$ bestimmt. Die kleinsten α-Werte für jede Parameterkombination sind kursiv dargestellt.*

dargestellten Ergebnisse zeigen, daß der rechte Punkt x_2 mit der Initialisierung (4.3) nahe am Optimum liegt, während der linke Punkt tendenziell etwas näher beim Modus liegen sollte. Für die weiteren Simulationen der GIG-Verteilung wurde deshalb statt (4.2) die Initialisierung

$$x_1 = \max\left\{\tfrac{6}{10}m, m - \tfrac{8}{10}\sigma\right\}$$

gewählt.

4.3.2 Adaptionshäufigkeit

Nachdem die Stützstellen für T_2 bestimmt wurden, kann die Tangentenmethode zur Erzeugung unabhängiger Zufallszahlen gestartet werden. Mit zwei Startstützpunkten ist die Effizienz α nur für bestimmte Parameterkombinationen in der Nähe von 1. Es ist deshalb anzunehmen, daß im allgemeinen nur wenige Zahlen ohne weitere Adaption der Hutfunktion generiert werden können. Genauere Auskunft darüber gibt Tabelle 4.3, in der für verschiedene Kombinationen von λ und β eine Stichprobe einer vorgegebenen Größe erzeugt wurde. Man erkennt, daß die Adaptionshäufigkeiten von der Parameterkombination und damit von der Gestalt der Verteilung abhängen. Wie bereits früher beschrieben wurde, ist die Effizienz α für Verteilungen mit großer Schiefe nahe bei eins. Ein kleines α (s. Tabelle 4.2) zieht eine geringe Anzahl von Verwerfungen und damit weniger Adaptionen der Hutfunktion nach sich, wie Tabelle 4.3 zeigt.

4.3.3 Erzeugungszeit

Mit der adaptiven Verwerfungsmethode steht dem Benutzer ein Instrument zur Verfügung, das es erlaubt, Hutfunktionen beliebig nah an die Dichtefunktion, von der eine Stichprobe erzeugt werden soll, anzunähern. Daß eine beliebig gute Approximation jedoch nicht

	λ=1											
	$\beta = 0.1$				$\beta = 1$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.01	0	0	0	0.05	0	0	0	0.40	0	0	2
10	0.08	0	0	1	0.88	1	0	2	2.16	2	1	4
100	0.62	0	0	2	4.71	5	3	7	6.02	6	4	8
1000	4.48	4	3	7	12.62	13	10	15	16.03	16	13	18
10000	13.40	13	10	17	28.93	29	25	33	37.33	37	33	42

	λ=2											
	$\beta = 0.1$				$\beta = 2$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.16	0	0	1	0.33	0	0	1	0.39	0	0	2
10	1.33	1	0	3	1.88	2	0	3	2.03	2	0	4
100	5.54	5	4	8	6.21	6	4	9	6.10	6	4	8
1000	14.65	15	11	18	15.39	15	12	18	16.25	16	13	19
10000	33.78	33	30	38	35.52	35	31	41	37.42	37	34	42

	λ=8											
	$\beta = 0.1$				$\beta = 2$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.43	0	0	2	0.34	0	0	2	0.47	0	0	2
10	2.09	2	1	3	2.24	2	1	4	2.32	2	1	4
100	6.51	6	5	9	6.60	7	5	9	6.50	6	4	9
1000	16.56	17	13	19	16.52	16	13	20	16.70	17	14	20
10000	37.82	38	33	42	37.94	38	33	42	38.31	38	34	43

Tabelle 4.3: *Untersuchung der Adaptionshäufigkeiten der Tangentenmethode angewendet auf die GIG(λ,β)-Verteilung für mehrere Parameterkombinationen und je 100 Simulationsläufe. Die mittlere Anzahl der Adaptionen $\bar{R}$ ist gemeinsam mit dem Median $q_{0.5}$ und den Quartilen $q_{0.25}$ und $q_{0.75}$ der empirischen Verteilung der Adaptionen angegeben.*

nur Vorteile aufweist, sollen folgende Überlegungen zeigen: Mit jeder Adaption wird eine zusätzliche Stützstelle x_i, an der die Tangente an die Dichtefunktion berechnet wird, in die Menge T_r aufgenommen. Die logarithmierte Dichte $l(\cdot)$ wird dadurch sukzessive durch eine stückweise lineare Funktion approximiert (s. Abbildung 4.3). Wenn die Anzahl der Stützstellen gleich r ist, besteht die Hutfunktion $h(\cdot)$ aus r linearen Funktionen. Den Funktionswert der Hutfunktion an der Stelle x für $x \in [z_{i-1}, z_i)$, wobei die z_i die Schnittpunkte der Tangenten sind, erhält man durch

$$h(x) = l(x_i) + l'(x_i)(x - x_i).$$

Durch die Rücktransformation $\exp[h(\cdot)]$ wird die Hutfunktion zu einer stückweisen Exponentialfunktion (s. Abbildung 4.4), aus der Kandidaten für den Verwerfungsschritt generiert werden müssen. Dazu muß die kumulierte Dichtefunktion berechnet werden, die aufgrund der stückweisen Definition der Hutfunktion nicht geschlossen invertierbar ist. Eine Anwendung der herkömmlichen Inversionsmethode zur Erzeugung der Kandidaten ist deshalb nicht möglich. Es kann jedoch jedes Stück der kumulierten Exponentialfunk-

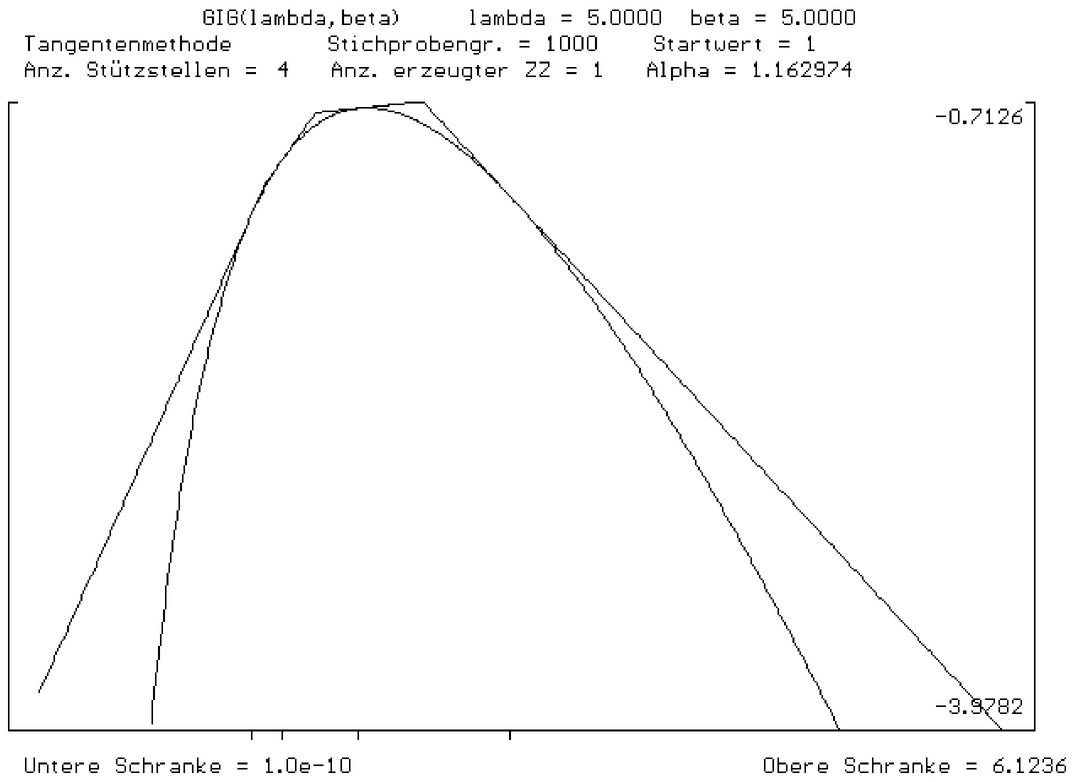


Abbildung 4.3: *Bei der Ausführung der Tangentenmethode wird die Dichte $l(\cdot)$ durch Tangenten in r Stützstellen (hier für $r = 4$) approximiert. Die entstehende Hutfunktion $h(\cdot)$ paßt sich mit steigendem r der Dichte an.*

tion getrennt invertiert werden, sodaß bei r Stützstellen r Funktionen angegeben werden können. Man definiert dafür

$$I_j(x) = \int_{z_{j-1}}^{x} \exp[h(v)]dv \tag{4.4}$$

mit $j = 1, \ldots, r$ und $x \in [z_{j-1}, z_j)$. Die Funktion $I_j(x)$ kann invertiert werden, da $h(\cdot)$ eine lineare Funktion ist. Die invertierte Funktion wird im folgenden mit $I_j^{-1}(\cdot)$ bezeichnet. Mit folgenden vier Schritten kann nun ein Kandidat Y für den Verwerfungsschritt der Tangentenmethode erzeugt werden:

1. Erzeugung einer gleichverteilten Zufallszahl U aus dem Intervall $(0, 1)$.

2. Berechnung der Flächenstücke unter den Segmenten der Hutfunktion:

   $$I_j := \int_{z_{j-1}}^{z_j} \exp[h(v)]dv, \quad j = 1, \ldots, r.$$

   Nun kann man mit $I_{total} = \sum_{j=1}^{r} I_j = \int_D \exp[h(v)]dv$ die Gesamtfläche unter der Hutfunktion definieren.

3. Bestimmung des Intervalls i mit $i \in \{1, \ldots, r\}$, für das

   $$\exists x \in [z_{i-1}, z_i) : \sum_{j=1}^{i-1} I_j + I_i(x) = I_{total}U$$

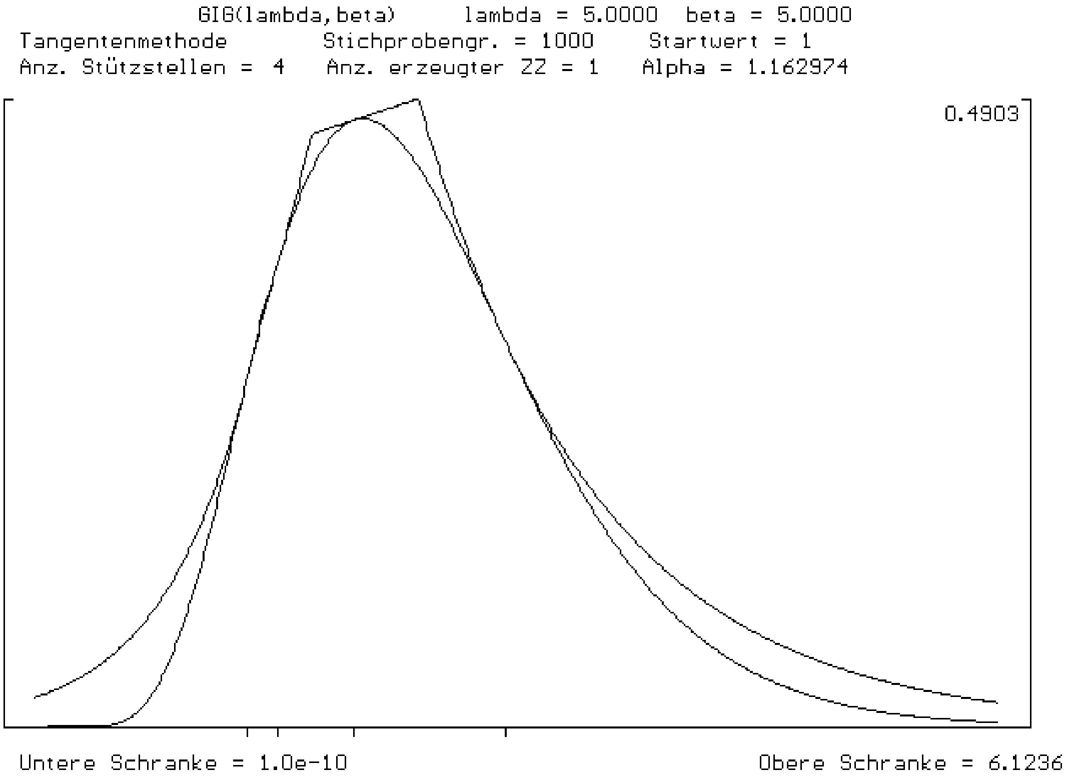


Abbildung 4.4: *Nach der Rücktransformation ist die Hutfunktion $h(\cdot)$ eine stückweise exponentielle Funktion.*

gilt.

4. Der Kandidat Y kann nun durch

$$Y = I_i^{-1}(I_{total}U - \sum_{j=1}^{i-1} I_j)$$

ermittelt werden.

Anmerkung: Es empfiehlt sich, die in Schritt 2 zu berechnenden Flächenstücke für größere Stichproben nur jeweils nach einer Adaption der Hutfunktion zu berechnen und sie dann für die weitere Ausführung des Algorithmus zu speichern. Eine Neuberechnung für jede Kandidatenerzeugung verlängert die Ausführungsgeschwindigkeit.

Die in den Punkten 1 bis 4 beschriebene Erzeugung von Kandidaten Y, die für den Verwerfungsschritt benötigt werden, ist abhängig von der Anzahl der Stützstellen r. So wird beispielsweise in Punkt 3 der Index eines Intervalls durch eine Suche bestimmt, deren Ausführungszeit von der Größe der zu durchsuchenden Menge $\{[z_0, z_1), \ldots, [z_r, z_{r+1})\}$ ist. (Für diese Suche könnte aus Geschwindigkeitsgründen ein Algorithmus verwendet werden, der auf die Gestalt der Verteilung Rücksicht nimmt und in jenen Intervallen die Suche startet, die die größte Wahrscheinlichkeitsmasse aufweisen.) Es ist daher anzunehmen, daß für den Fall, daß r eine gewisse Schranke überschreitet, der Algorithmus zur

Kandidatenerzeugung zu aufwendig wird, sodaß die Verbesserung, die durch eine weitere Adaption der Hutfunktion gewonnen wird, nicht ins Gewicht fällt.

In Tabelle 4.4 sind die mittleren Erzeugungszeiten einer Zufallszahl für verschiedene Kombinationen von λ und β in Abhängigkeit von der Anzahl der Adaptionen angegeben. Die Tabelle wurde erstellt, indem für jede Parameterkombination ein Simulationslauf ausgeführt wurde, wobei nach jeder Adaption 100.000 Zufallszahlen erzeugt wurden, ohne eine weitere Adaption durchzuführen. Danach wurden solange weitere Zahlen generiert, bis eine Verwerfung auftrat. Die Hutfunktion wurde modifiziert und 100.000 neue Zahlen ohne weitere Adaption erzeugt. Dieses Vorgangsweise wurde für jede Parameterkombination $10\times$ wiederholt, für jede Stichprobenerzeugung die Zeit gemessen und diese gemittelt. Diese und die folgenden Zeitmessungen wurden auf einem IBM kompatiblen Rechner (CPU: 80586/200MHz) durchgeführt.

Tabelle 4.4 zeigt, daß für Hutfunktionen, die eine Effizienz von etwa $\alpha = 1.05$ aufweisen, weitere Adaptionen keine positive Auswirkung auf die mittlere Erzeugungszeit einer Zufallszahl haben. Es ist deshalb empfehlenswert, für die Erzeugung einer großen Anzahl von Zufallszahlen den Adaptionsschritt nicht mehr durchzuführen, wenn α eine gewisse Schranke unterschritten hat, und weitere Kandidaten aus einer gleichbleibenden Hutfunktion zu erzeugen.

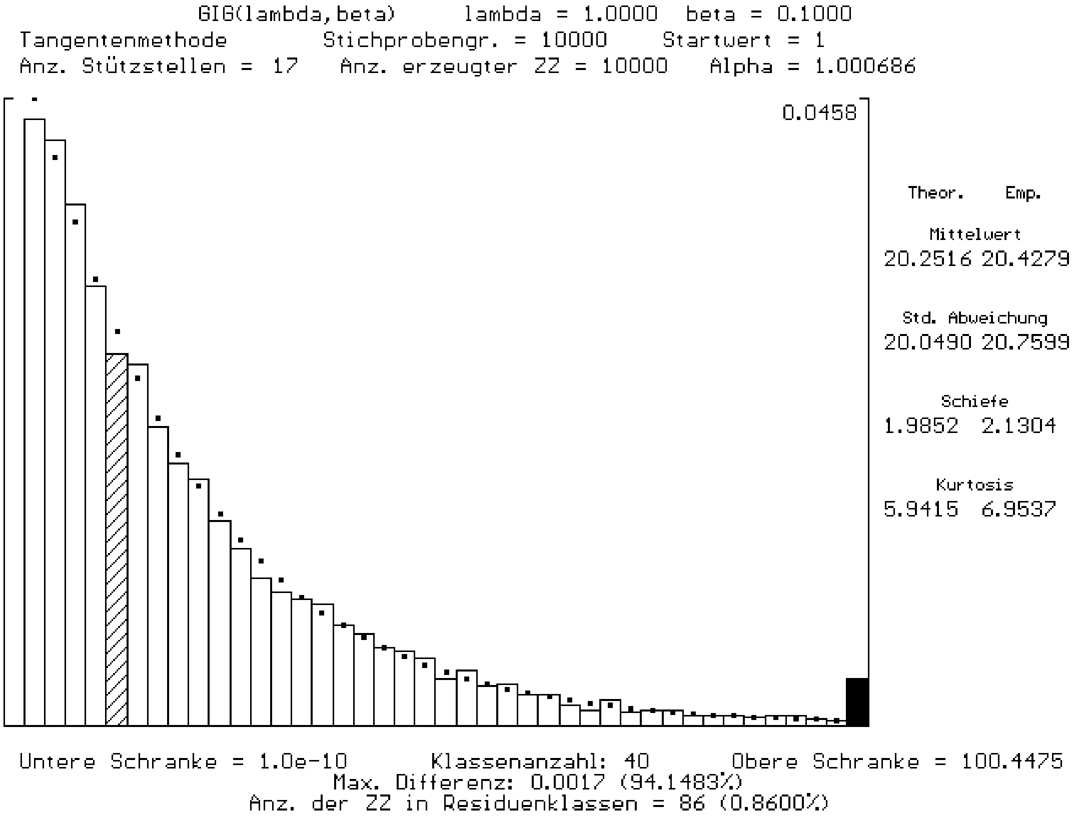
Abschließend wurde ein Vergleich der Schnelligkeit der Tangentenmethode mit herkömmlichen Verfahren zur Erzeugung von GIG(λ,β)-verteilter Zufallszahlen durchgeführt. Die Ergebnisse sind zusammen mit denen der Sekantenmethode weiter unten in Abschnitt 4.5 zu finden.

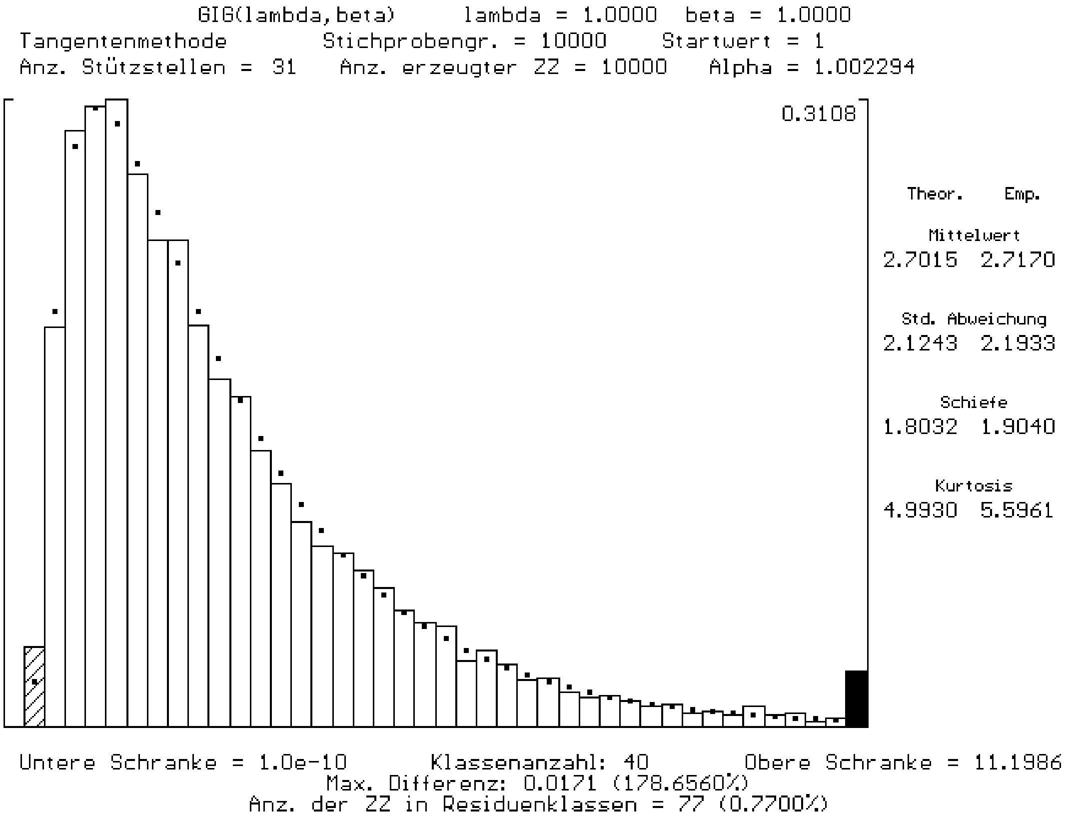
	λ=1					
	$\beta = 0.1$		$\beta = 1$		$\beta = 8$	
r	α	Erzeugungszeit	α	Erzeugungszeit	α	Erzeugungszeit
2	1.0081	*22.5*	1.1122	26.9	1.2811	32.9
3	1.0062	23.6	1.0965	25.8	1.1704	29.1
4	1.0050	24.2	1.0582	25.9	1.1456	28.6
5	1.0040	24.7	1.0251	*24.1*	1.0802	*25.8*
6	1.0033	25.9	1.0212	24.7	1.0758	26.4
7	1.0027	26.3	1.0164	25.3	1.0572	*25.8*
8	1.0023	27.5	1.0157	25.8	1.0362	26.9
9	1.0019	26.9	1.0125	26.9	1.0226	27.5
10	1.0019	27.0	1.0122	28.0	1.0155	28.5
11	1.0013	28.0	1.0084	29.2	1.0148	28.0

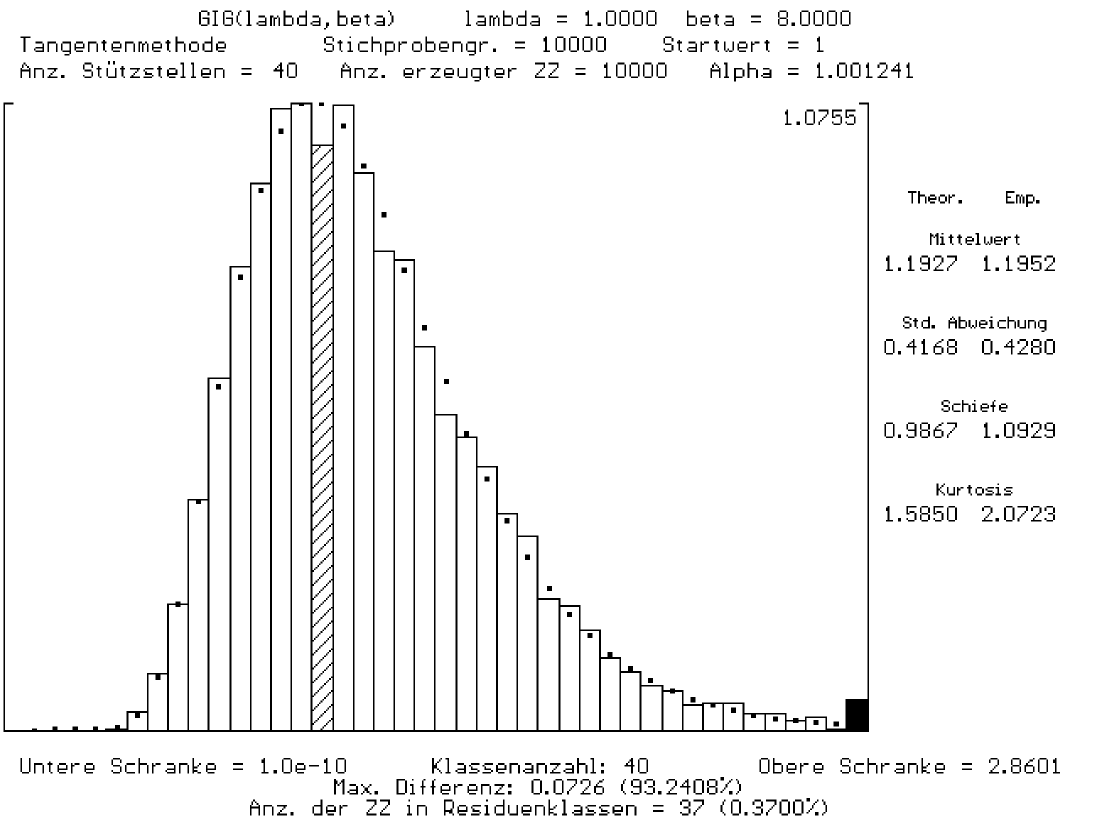
	λ=2					
	$\beta = 0.1$		$\beta = 2$		$\beta = 8$	
r	α	Erzeugungszeit	α	Erzeugungszeit	α	Erzeugungszeit
2	1.1548	27.5	1.2096	29.1	1.2827	32.9
3	1.0989	25.8	1.1535	26.9	1.1375	28.0
4	1.0610	24.7	1.0961	25.9	1.0486	26.4
5	1.0575	*24.2*	1.0441	*24.7*	1.0437	26.4
6	1.0401	25.3	1.0276	25.2	1.0304	26.3
7	1.0230	25.2	1.0224	25.3	1.0265	*25.8*
8	1.0203	26.4	1.0195	26.9	1.0202	27.0
9	1.0187	26.9	1.0155	26.9	1.0171	26.3
10	1.0133	27.5	1.0146	27.0	1.0114	26.9
11	1.0108	26.9	1.0133	27.4	1.0089	27.5

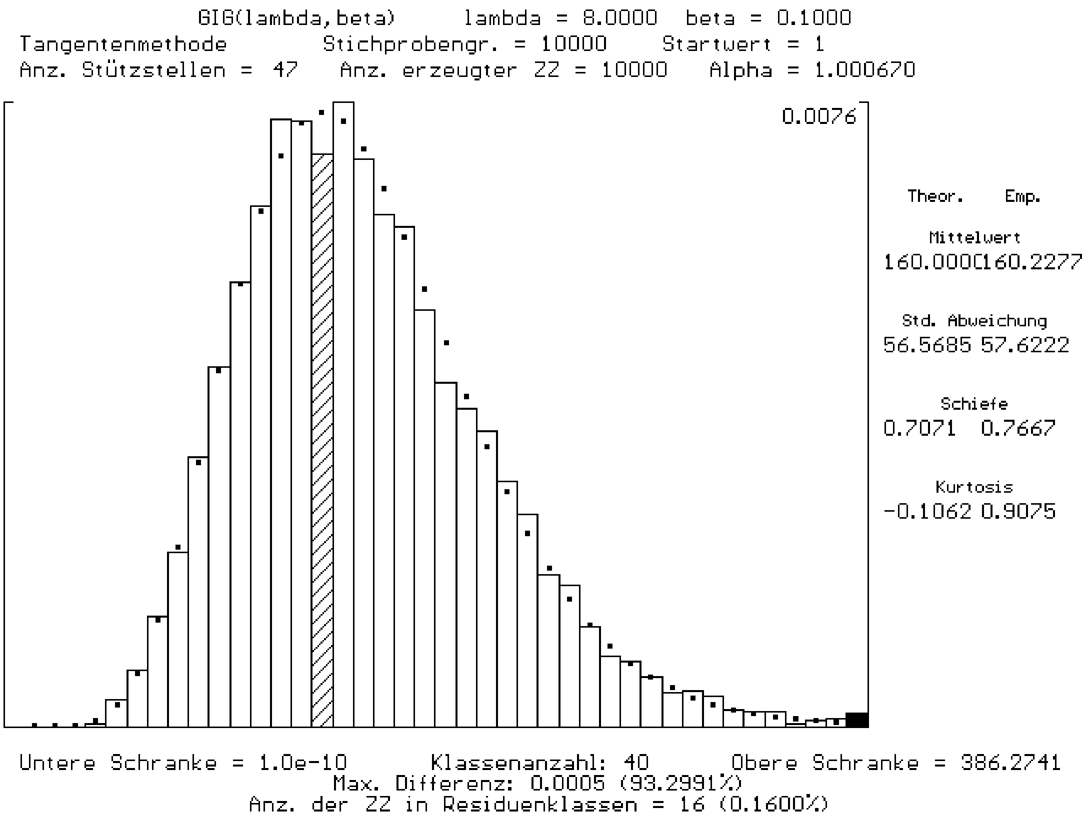
	λ=8					
	$\beta = 0.1$		$\beta = 2$		$\beta = 8$	
r	α	Erzeugungszeit	α	Erzeugungszeit	α	Erzeugungszeit
2	1.2868	32.4	1.2885	32.4	1.2967	32.4
3	1.1928	28.6	1.1912	28.5	1.2340	32.4
4	1.1542	29.6	1.1407	28.6	1.1353	28.1
5	1.1110	28.6	1.1363	29.7	1.0816	26.9
6	1.0731	26.4	1.0550	*26.3*	1.0502	*25.2*
7	1.0521	26.4	1.0430	26.4	1.0409	25.9
8	1.0304	*26.3*	1.0412	*26.3*	1.0307	25.8
8	1.0220	26.4	1.0346	27.0	1.0210	26.3
10	1.0182	26.4	1.0262	26.9	1.0187	27.5
11	1.0142	27.4	1.0245	28.0	1.0151	27.5

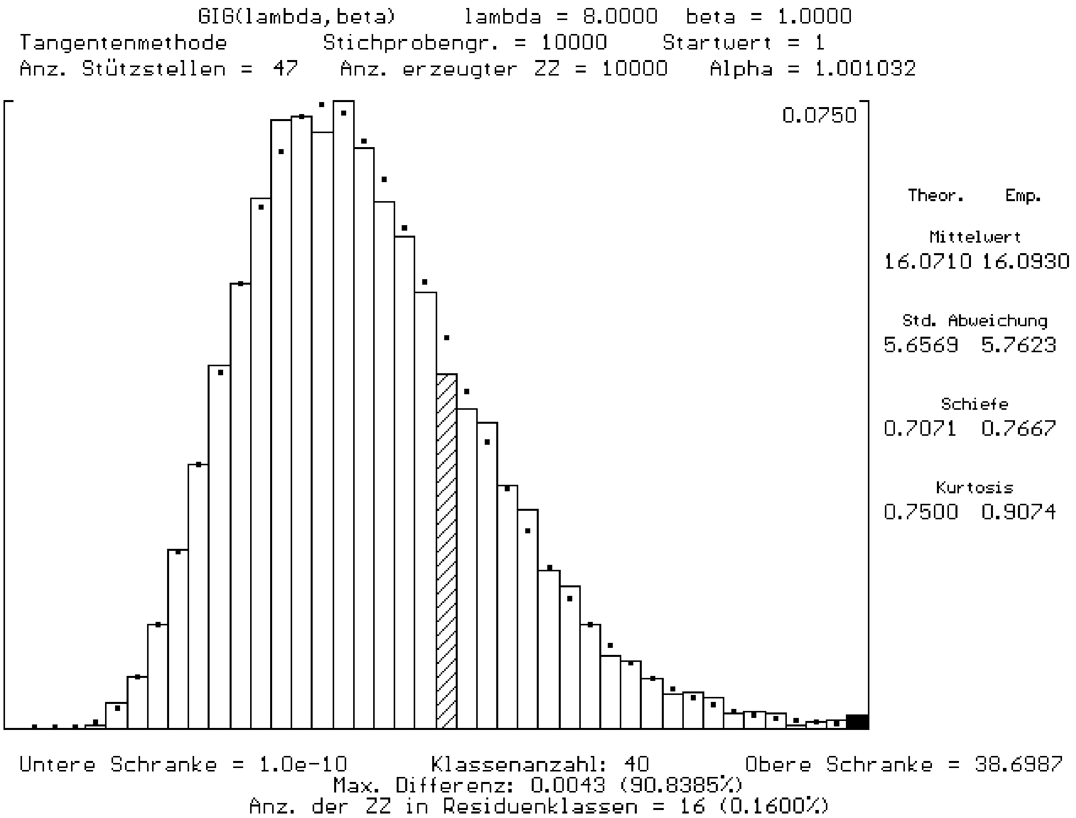
Tabelle 4.4: *Darstellung der mittleren Erzeugungzeit (in μsek) einer GIG(λ,β)-verteilten Zufallszahl abhängig von der Anzahl der Stützstellen r der Hutfunktion, die mit der Tangentenmethode konstruiert wurde.*

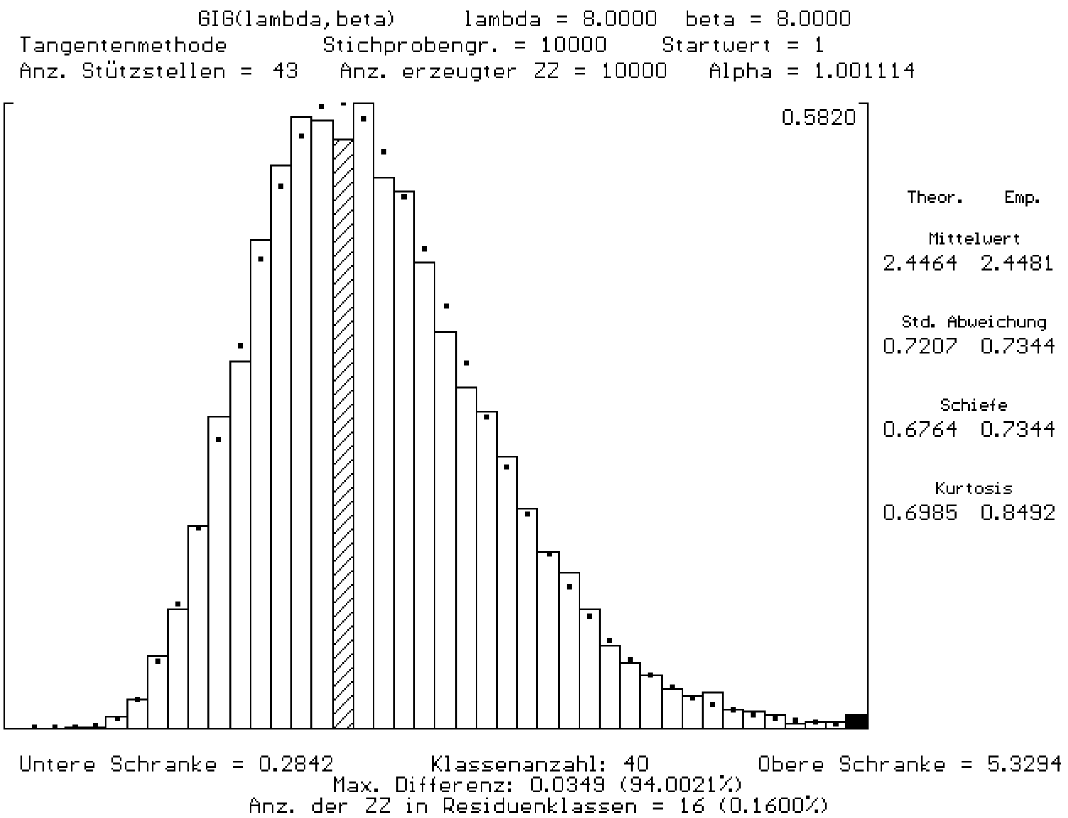

GIG(lambda,beta) lambda = 1.0000 beta = 0.1000
Tangentenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 17 Anz. erzeugter ZZ = 10000 Alpha = 1.000686
0.0458
Theor. Emp.
Mittelwert
20.2516 20.4279
Std. Abweichung
20.0490 20.7599
Schiefe
1.9852 2.1304
Kurtosis
5.9415 6.9537
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 100.4475
Max. Differenz: 0.0017 (94.1483%)
Anz. der ZZ in Residuenklassen = 86 (0.8600%)


GIG(lambda,beta) lambda = 1.0000 beta = 1.0000
Tangentenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 31 Anz. erzeugter ZZ = 10000 Alpha = 1.002294
0.3108
Theor. Emp.
Mittelwert
2.7015 2.7170
Std. Abweichung
2.1243 2.1933
Schiefe
1.8032 1.9040
Kurtosis
4.9930 5.5961
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 11.1986
Max. Differenz: 0.0171 (178.6560%)
Anz. der ZZ in Residuenklassen = 77 (0.7700%)


GIG(lambda,beta) lambda = 1.0000 beta = 8.0000
Tangentenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 40 Anz. erzeugter ZZ = 10000 Alpha = 1.001241
1.0755
Theor. Emp.
Mittelwert
1.1927 1.1952
Std. Abweichung
0.4168 0.4280
Schiefe
0.9867 1.0929
Kurtosis
1.5850 2.0723
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 2.8601
Max. Differenz: 0.0726 (93.2408%)
Anz. der ZZ in Residuenklassen = 37 (0.3700%)


GIG(lambda,beta) lambda = 8.0000 beta = 0.1000
Tangentenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 47 Anz. erzeugter ZZ = 10000 Alpha = 1.000670
0.0076
Theor. Emp.
Mittelwert
160.0000160.2277
Std. Abweichung
56.5685 57.6222
Schiefe
0.7071 0.7667
Kurtosis
-0.1062 0.9075
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 386.2741
Max. Differenz: 0.0005 (93.2991%)
Anz. der ZZ in Residuenklassen = 16 (0.1600%)


GIG(lambda,beta) lambda = 8.0000 beta = 1.0000
Tangentenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 47 Anz. erzeugter ZZ = 10000 Alpha = 1.001032
0.0750
Theor. Emp.
Mittelwert
16.0710 16.0930
Std. Abweichung
5.6569 5.7623
Schiefe
0.7071 0.7667
Kurtosis
0.7500 0.9074
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 38.6987
Max. Differenz: 0.0043 (90.8385%)
Anz. der ZZ in Residuenklassen = 16 (0.1600%)


GIG(lambda,beta) lambda = 8.0000 beta = 8.0000
Tangentenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 43 Anz. erzeugter ZZ = 10000 Alpha = 1.001114
0.5820
Theor. Emp.
Mittelwert
2.4464 2.4481
Std. Abweichung
0.7207 0.7344
Schiefe
0.6764 0.7344
Kurtosis
0.6985 0.8492
Untere Schranke = 0.2842 Klassenanzahl: 40 Obere Schranke = 5.3294
Max. Differenz: 0.0349 (94.0021%)
Anz. der ZZ in Residuenklassen = 16 (0.1600%)

4.4 Analyse der Sekantenmethode

4.4.1 Initialisierung der Stützstellen

Im Gegensatz zu der Tangentenmethode wird mit der Sekantenmethode die Hutfunktion durch Sekanten bestimmt, die die Dichtefunktion in r Stützstellen, die in der Menge T_r zusammengefaßt sind, schneiden. Weil für die Sekantenmethode die erste Ableitung der Dichtefunktion nicht angegeben werden muß, eignet sie sich vor allem für Verteilungen, deren Ableitung nicht oder nur schwer berechenbar ist.

Zum Start des Algorithmus sind *mindestens drei* Stützstellen zu bestimmen, von denen eine links und eine rechts vom Modus liegen muß. Analog zur Tangentenmethode wird untersucht, wie die Stützstellen gewählt werden sollen, damit die Effizienz α beim Start des Verfahrens möglichst nahe bei eins liegt.

Empirische Untersuchungen mit dem Programm `ARS` zeigen, daß drei Stützstellen im allgemeinen nicht genügen, um eine gute Approximation der Hutfunktion gemäß der Sekantenmethode an die Dichtefunktion zu gewährleisten. Bei der Initialisierung von T_3 wurde zunächst so vorgegangen: Zuerst wurden die beiden äußeren Punkte x_1 und x_3 bestimmt, für die $x_1 < m$ und $m < x_3$ gilt. Beide Punkte wurden ähnlich wie in der Tangentenmethode gewählt, d. h. $x_1 = \max\{\frac{m}{2}, m - \sigma\}$ und $x_3 = m + \sigma$. Da mit der GIG(λ,β)-Verteilung eine rechtsschiefe Verteilung vorliegt, ist der Punkt x_2 rechts vom Modus zu wählen. Zunächst wurde untersucht, welche Auswirkung eine Wahl von x_2 im Zentrum der Verteilung auf α hat. Abbildung 4.5 zeigt, daß man mit dieser Wahl weder für eine schiefe noch für eine annähernd symmetrische Verteilung mit $\alpha \approx 2$ akzeptable Approximationen erhält. Nun wurde versucht, x_2 weiter im rechten Schwanz der Verteilung zu wählen. In Abbildung 4.6 erkennt man, daß diese Wahl für den schiefen Verteilungstyp mit $\alpha \approx 1.2$ ein klar besseres Ergebnis liefert. Für den annähernd symmetrischen Verteilungstyp entsteht jedoch zwischen x_1 und x_2 ein so großer Raum zwischen der asymmetrischen Hutfunktion und der Dichte, daß bei einem Start des Algorithmus mit dieser Konstellation mit hoher Wahrscheinlichkeit in diesem Bereich noch vor Erzeugung der ersten Zufallszahl eine Adaption der Hutfunktion vorgenommen würde.

Aufgrund der Tatsache, daß die GIG-Verteilung für unterschiedliche Parameterwerte entweder stark rechtsschief oder fast symmetrisch sein kann, scheint es ratsam, die Anzahl der Stützstellen zu Beginn mit $r = 4$ festzusetzen. Man könnte nun versuchen, durch empirische Versuche Strategien zur Verteilung der Stützstellen auf der x-Achse zu finden, die für die unterschiedlichen Gestalten der GIG-Verteilung die besten Resultate liefern. Es zeigt sich, daß eine gleichmäßige Streuung der Stützstellen um das Zentrum der Verteilung gute Ergebnisse liefert. Setzt man beispielsweise

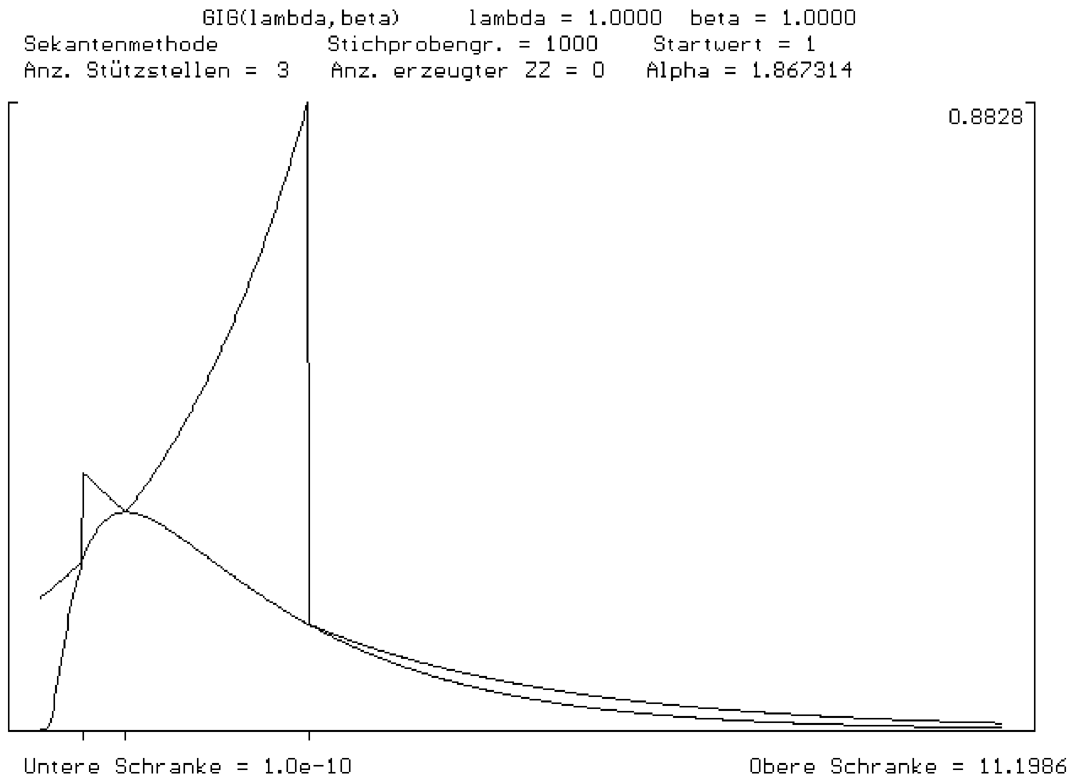
$$\begin{aligned} x_1 &= \max\left\{\tfrac{1}{3}m, m - \sigma\right\} \\ x_2 &= \max\left\{\tfrac{2}{3}m, m - \tfrac{1}{2}\sigma\right\} \\ x_3 &= m + \tfrac{1}{2}\sigma \\ x_4 &= m + \sigma, \end{aligned} \tag{4.5}$$

erhält man die in Tabelle 4.5 aufgeführten Werte für α. Wenn man berücksichtigt, daß eine

	λ=1			λ=2			λ=8		
	$\beta = 0.1$	$\beta = 1$	$\beta = 8$	$\beta = 0.1$	$\beta = 2$	$\beta = 8$	$\beta = 0.1$	$\beta = 2$	$\beta = 8$
α	1.0152	1.1469	1.3371	1.2182	1.2829	1.3421	1.3594	1.3613	1.3711

Tabelle 4.5: *Effizienzwerte α der Sekantenmethode mit vier Stützstellen zur Konstruktion der Hutfunktion für die GIG(λ,β)-Verteilung mit unterschiedlichen Werten für λ und β. Die Stützstellen wurden gemäß (4.5) gewählt.*

Approximation durch Sekanten bei gleicher Anzahl von Stützstellen nicht so gut sein kann wie durch Tangenten, dann sind die in Tabelle 4.5 dargestellten Effizienzen für die praktische Verwendung durchaus akzeptabel. Für die weitere Analyse der Sekantenmethode wird deshalb die durch (4.5) bestimmte Initialisierung von T_4 verwendet (s. Abbildung 4.7).



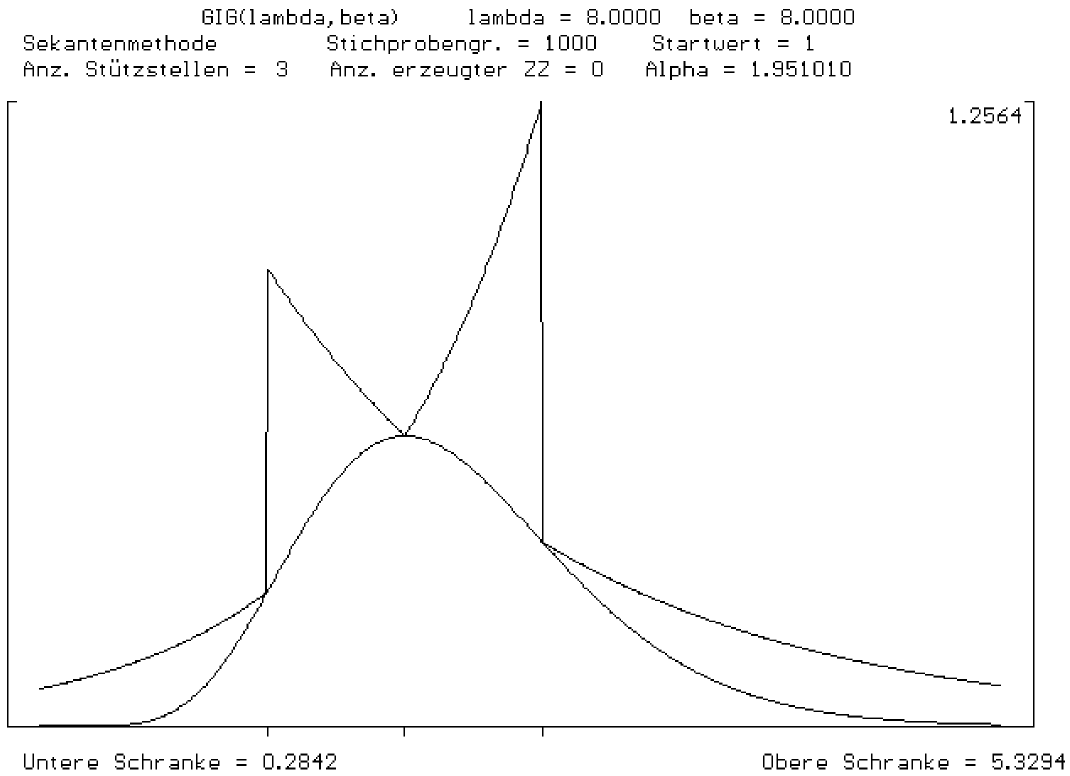


Abbildung 4.5: *Initialisierung der Sekantenmethode mit drei Stützstellen. Mit Effizienzen von $\alpha \approx 2$ für den schiefen und für den annähernd symmetrischen Verteilungstyp sollte diese Initialisierungsvariante nicht gewählt werden.*

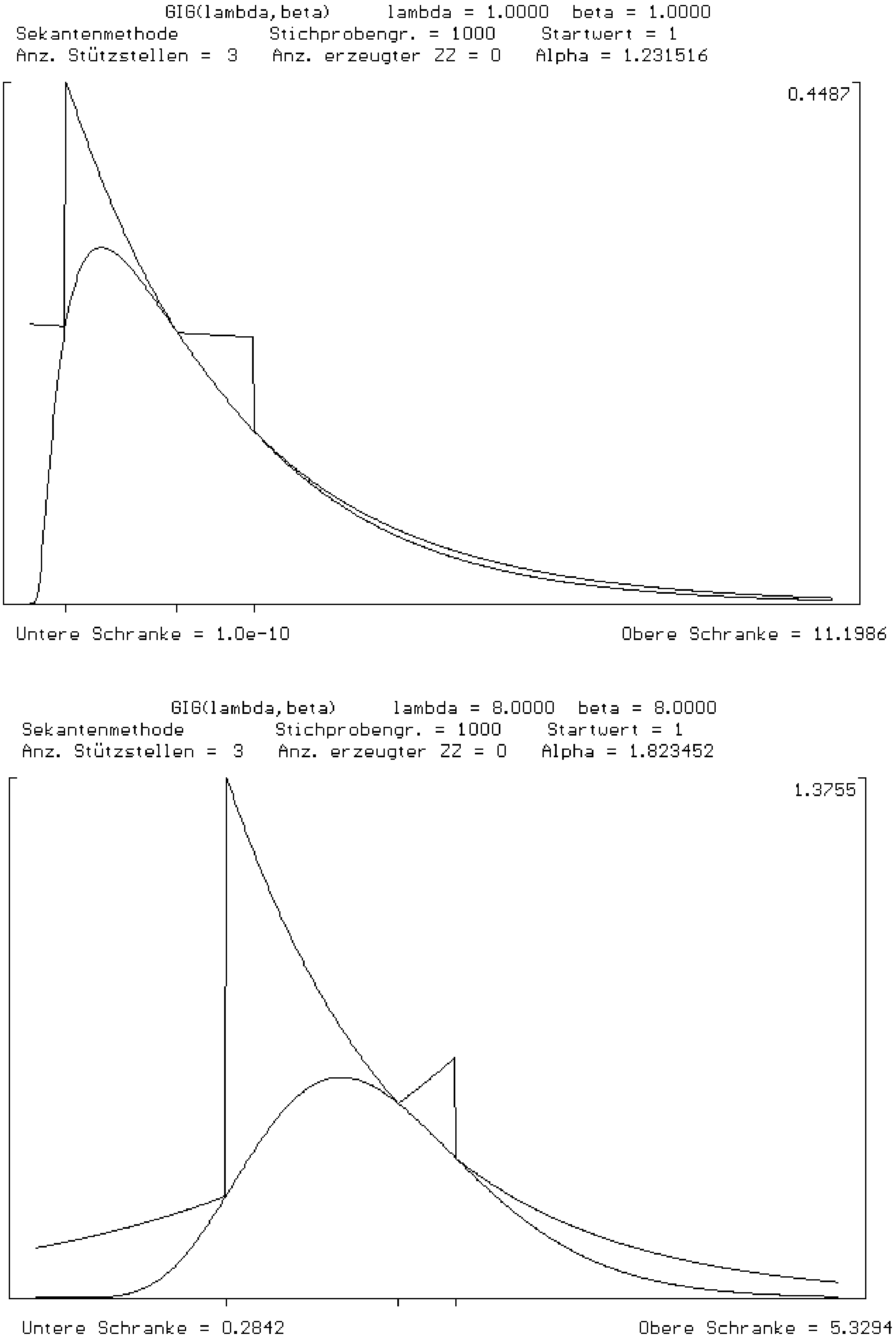


Abbildung 4.6: *Diese Initialisierung für $r = 3$ ist nur für den schiefen Verteilungstyp geeignet.*

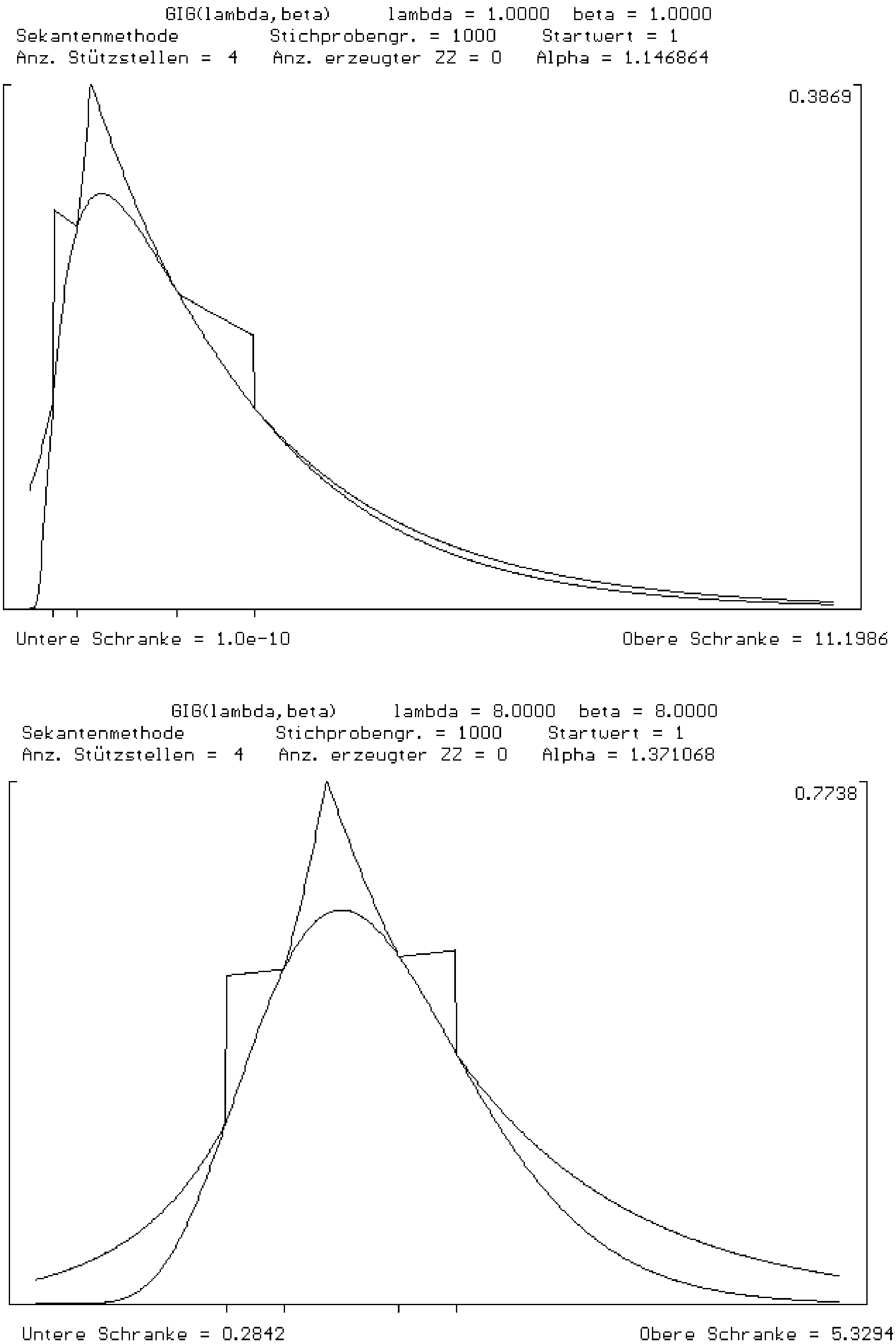


Abbildung 4.7: *Mit vier Stützstellen kann die Dichte gut durch die Sekantenmethode approximiert werden.*

	λ=1											
	$\beta = 0.1$				$\beta = 1$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.01	0	0	0	0.09	0	0	1	0.22	0	0	1
10	0.15	0	0	1	1.13	1	0	3	2.41	2	1	4
100	1.20	1	0	3	6.21	6	4	9	8.98	9	6	12
1000	6.02	6	4	9	17.78	17	14	21	23.17	23	19	27
10000	18.98	19	15	22	41.88	41	37	48	53.76	53	48	60

	λ=2											
	$\beta = 0.1$				$\beta = 2$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.14	0	0	1	0.31	0	0	1	0.32	0	0	1
10	1.71	2	0	4	2.16	2	0	4	2.53	2	1	5
100	7.03	7	5	9	8.27	8	6	11	9.28	9	7	12
1000	20.16	20	17	23	22.01	22	19	25	23.43	23	20	27
10000	48.77	49	43	55	51.19	51	46	58	54.03	54	49	60

	λ=8											
	$\beta = 0.1$				$\beta = 2$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.53	0	0	2	0.47	0	0	2	0.45	0	0	2
10	2.54	3	0	5	2.65	3	0	5	2.71	3	0	5
100	9.38	9	7	12	9.49	9	7	12	9.64	10	7	12
1000	23.86	24	19	28	23.77	24	19	27	24.06	24	19	28
10000	55.57	55	50	62	55.41	55	49	63	55.64	55	49	62

Tabelle 4.6: *Untersuchung der Adaptionshäufigkeiten der Sekantenmethode angewendet auf die GIG(λ,β)-Verteilung für mehrere Parameterkombinationen, für die je 100 Simulationsläufe durchgeführt wurden. Die mittlere Anzahl der Adaptionen $\bar{R}$ ist gemeinsam mit dem Median $q_{0.5}$ und den Quartilen $q_{0.25}$ und $q_{0.75}$ der empirischen Verteilung der Adaptionen angegeben.*

4.4.2 Adaptionshäufigkeit

Als erste Analyse der Sekantenmethode wurde für verschiedene Parameterkonstellationen der GIG(λ,β)-Verteilung untersucht, wieviele Adaptionen der Hutfunktion für die Erzeugung einer Stichprobe mit vorgegebener Größe benötigt werden (s. Tabelle 4.6). Wie bereits angemerkt wurde, sind die Effizienzen der Hutfunktion nach der Sekantenmethode bei gleicher Anzahl von Stützstellen deutlich höher als die der Hutfunktion nach der Tangentenmethode. Daraus resultiert bei der Erzeugung einer gleich großen Stichprobe eine größere Verwerfungsrate und somit eine häufigere Adaption der Hutfunktion. Diese Tatsache läßt sich auch durch einen Vergleich der Tabelle 4.3 aus dem vorigen Abschnitt mit Tabelle 4.6 bestätigen.

4.4.3 Erzeugungszeit

Für die Sekantenmethode gelten bezüglich des Zusammenhangs zwischen der Anzahl der Stützstellen r und der Erzeugungszeit einer Zufallszahl ähnliche Überlegungen wie für die Tangentenmethode (s. Abschnitt (4.3.3)): Es ist nicht vernünftig, für eine große Stichprobe den Adaptionsschritt so oft wie möglich durchzuführen, sondern die Hutfunktion, nachdem α eine gewisse Schranke erreicht hat, für die restlichen Iterationen unverändert zu lassen.

Die Hutfunktion gemäß der Tangentenmethode besteht, wenn die Anzahl der Stützstellen gleich r ist, aus r linearen Funktionen, die jeweils zwischen den Schnittpunkten der Tangenten $z_0, \ldots, z_r$ definiert sind. Um einen Kandidaten Y für den Verwerfungsschritt zu erzeugen, ist es nötig, eine Suche aus r Elementen durchzuführen, die das Verfahren für große r deutlich verlangsamt.

Die Hutfunktion der Sekantenmethode setzt sich für r Stützstellen aus $2(r-1)$ linearen Funktionen zusammen. Für die Erzeugung des Kandidaten Y gilt ähnliches wie für die Tangentenmethode: Auch hier ist für jede Erzeugung eine Suche in einer Menge, deren Anzahl von Elementen nun jedoch $2(r-1)$ beträgt, notwendig, weshalb der Adaptionsschritt nach noch weniger Verwerfungen unwirtschaftlich wird.

Für Tabelle 4.7 wurde für unterschiedliche Parameterkombinationen der GIG-Verteilung die mittlere Erzeugungszeit einer Zufallszahl abhängig von der Anzahl der Adaptionen der Hutfunktion ermittelt. Man erkennt, daß, nachdem sich die Erzeugungszeit für ungefähr fünf Adaptionen tendenziell verkürzt, für weitere Adaptionen wieder ansteigende Zeiten zu verzeichnen sind. Es scheint deshalb ratsam, den Adaptionsschritt für Hutfunktionen, die eine Effizienz von $\alpha \approx 1.08$ aufweisen, nicht mehr auszuführen. Vergleicht man Tabelle 4.7 mit Tabelle 4.4 aus Abschnitt 4.3.3, so erkennt man nicht nur, daß die Reduktion von α pro Adaption der Hutfunktion für die Sekantenmethode kleiner ist als für die Tangentenmethode, sondern daß auch die mittlere Erzeugungszeit für die Sekantenmethode im Durchschnitt fast um 50% größer ist. Die beiden Methoden werden im folgenden Abschnitt eingehender verglichen.

4.5 Vergleich der Tangenten- und Sekantenmethode mit herkömmlichen Verfahren

In diesem Abschnitt soll die Leistungsfähigkeit der adaptiven Verwerfungsmethode mit der eines herkömmlichen, auf die GIG-Verteilung zugeschnittenen Verfahrens, verglichen werden. Für mehrere Parameterkombinationen der GIG(λ,β)-Verteilung wurden über einen langen Simulationslauf die mittlere Erzeugungszeit einer Zufallszahl berechnet. Für die Tangenten- und Sekantenmethode wurden die Überlegungen aus den Abschnitten 4.3 und 4.4 zur Initialisierung der Stützstellen und den Abbruch des Adaptionsvorganges nach dem Erreichen eines bestimmten α-Niveaus berücksichtigt.

Die Hutfunktion der Tangentenmethode wurde mit zwei Stützstellen gemäß

$$\begin{aligned} x_1 &= \max\left\{\tfrac{6}{10}m, m - \tfrac{8}{10}\sigma\right\} \\ x_2 &= m + \sigma \end{aligned}$$

	λ=1					
	$\beta = 0.1$		$\beta = 1$		$\beta = 8$	
r	α	Erzeugungszeit	α	Erzeugungszeit	α	Erzeugungszeit
4	1.0152	*30.2*	1.1469	34.6	1.3371	41.2
5	1.0077	30.8	1.1220	36.2	1.1445	36.8
6	1.0070	33.0	1.0720	*34.1*	1.1060	36.8
7	1.0060	35.1	1.0648	35.1	1.0881	36.8
8	1.0049	36.3	1.0542	35.2	1.0595	*35.2*
9	1.0041	38.4	1.0512	34.6	1.0512	37.9
10	1.0035	38.4	1.0435	36.2	1.0427	39.0
11	1.0031	40.1	1.0277	39.0	1.0347	40.6
12	1.0028	42.3	1.0224	40.7	1.0311	40.1
13	1.0023	44.0	1.0177	42.8	1.0260	40.1

	λ=2					
	$\beta = 0.1$		$\beta = 2$		$\beta = 8$	
r	α	Erzeugungszeit	α	Erzeugungszeit	α	Erzeugungszeit
4	1.2182	36.8	1.2829	39.5	1.3421	41.2
5	1.1788	36.8	1.2104	36.8	1.2679	39.5
6	1.1610	39.0	1.1497	36.8	1.2177	39.6
7	1.0707	*35.2*	1.0917	*34.1*	1.1308	*35.1*
8	1.0627	36.2	1.0577	35.7	1.1200	36.3
9	1.0514	35.7	1.0415	35.7	1.0668	36.2
10	1.0475	36.8	1.0370	35.7	1.0537	36.8
11	1.0469	39.0	1.0326	36.8	1.0479	36.8
12	1.0322	37.9	1.0284	38.4	1.0325	38.5
13	1.0280	40.1	1.0248	39.5	1.0284	39.5

	λ=2					
	$\beta = 0.1$		$\beta = 2$		$\beta = 8$	
r	α	Erzeugungszeit	α	Erzeugungszeit	α	Erzeugungszeit
4	1.3594	41.2	1.3613	40.6	1.3711	41.8
5	1.2352	39.0	1.2782	38.5	1.3104	40.1
6	1.1713	35.7	1.2521	41.7	1.2216	36.8
7	1.1432	*34.6*	1.1676	37.9	1.1938	*35.1*
8	1.0941	35.2	1.1604	*36.8*	1.1553	37.9
9	1.0668	36.2	1.1161	*36.8*	1.1098	36.3
10	1.0501	37.9	1.1076	37.3	1.0794	37.9
11	1.0369	38.5	1.0709	37.9	1.0634	39.0
12	1.0320	40.1	1.0621	39.0	1.0489	39.0
13	1.0297	41.2	1.0515	38.5	1.0413	39.0

Tabelle 4.7: *Mittlere Erzeugungszeiten (in μsek) einer GIG(λ,β)-verteilten Zufallszahl abhängig von der Anzahl der Stützstellen r der Hutfunktion, die mit der Sekantenmethode erzeugt wurde.*

		Erzeugungszeit (in μsek)		
λ	β	Tangentenmethode	Sekantenmethode	Quotientenmethode
1	0.1	22.0	29.6	14.2 (29.8)
	1	25.2	36.2	13.2 (31.3)
	8	24.2	37.4	10.8 (48.0)
2	0.1	24.2	37.4	12.0 (46.7)
	2	24.0	36.2	13.2 (45.6)
	8	25.2	37.2	11.2 (48.1)
8	0.1	27.6	37.4	13.0 (46.4)
	2	26.4	36.2	10.8 (49.1)
	8	25.4	36.2	13.2 (46.7)

Tabelle 4.8: *Vergleich der mittleren Erzeugungszeit einer GIG(λ,β)-verteilten Zufallszahl für unterschiedliche Methoden. Die für die Quotientenmethode in Klammer angegebenen Werte sind die Zeiten, die für das Setup benötigt werden.*

initialisiert und die Adaption der Hutfunktion $h(\cdot)$ nach dem Erreichen von $\alpha < 1.05$ abgebrochen. Für die Erzeugung der restlichen Zufallszahlen wurde $h(\cdot)$ unverändert gelassen.

Die Menge der Anfangsstützstellen T_4 wurde für die Sekantenmethode dabei gemäß (4.5) initialisiert und kein Adaptionsschritt durchgeführt, nachdem $\alpha < 1.08$ erreicht wurde.

Zum Vergleich der adaptiven Verwerfungsmethode mit einem herkömmlichen Verfahren wurde die Quotientenmethode mit Verschiebung um den Modus m (Lehner, 1989) herangezogen. Diese Methode erlaubt für alle Gestalten der GIG(λ,β)-Verteilung (also nicht nur für den logkonkaven Fall) eine schnelle Erzeugung von Zufallszahlen. Die Quotientenmethode ist in dem Programm `c_rand` (Niederl, 1993) implementiert, mit dem auch die Zeitmessungen vorgenommen wurden.

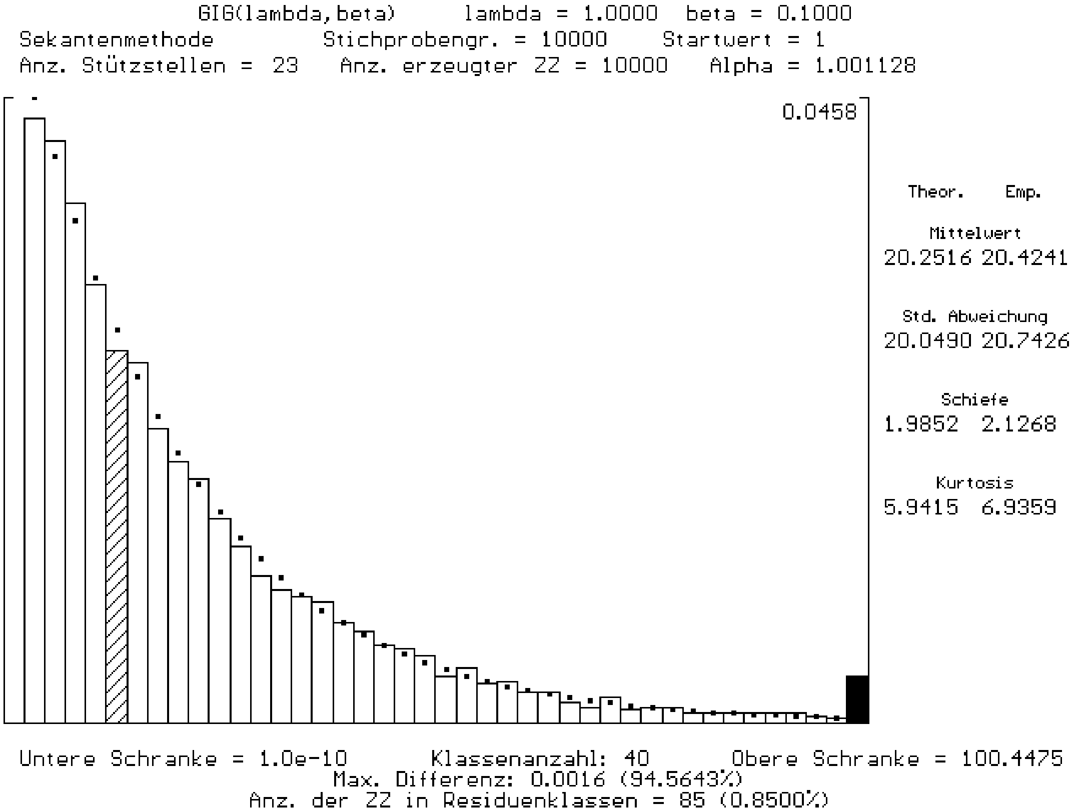
Tabelle 4.8 zeigt die mittlere Erzeugungszeit für eine gemäß der GIG(λ,β)-verteilte Zufallszahl. Für jede Parameterkombination und jedes Verfahren wurden 50.000 Zufallzahlen erzeugt.

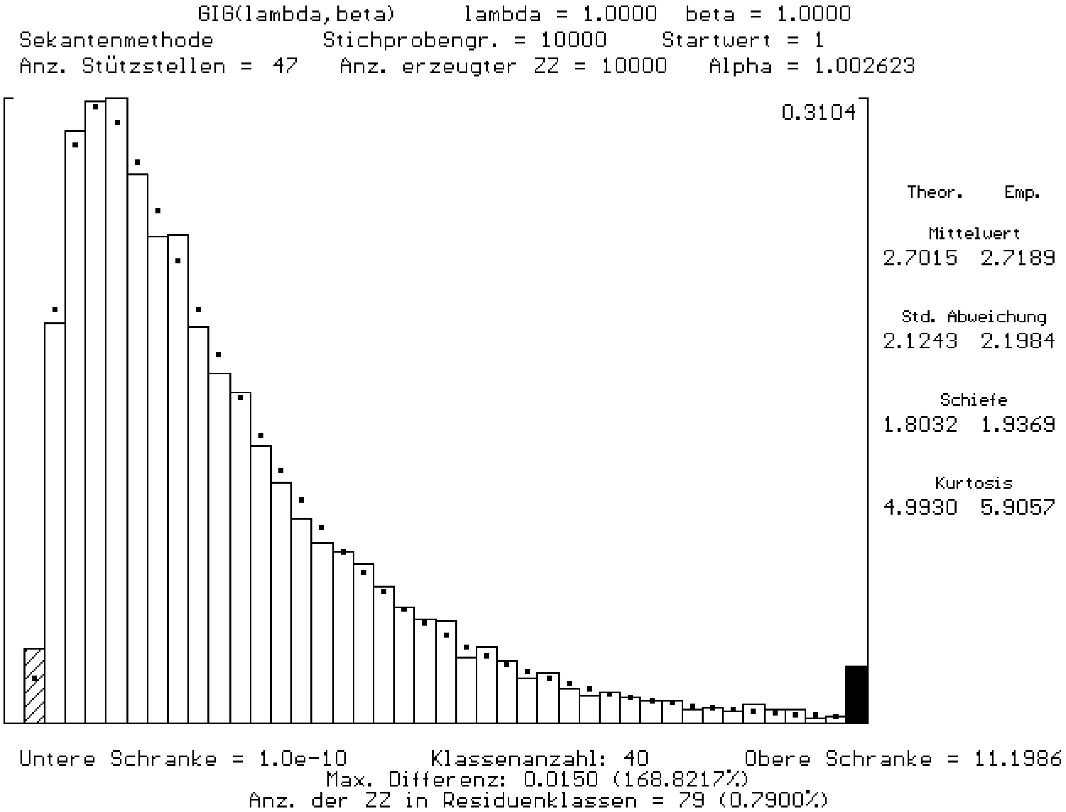
Der Vergleich der mittleren Erzeugungszeiten bestätigt, daß die adaptive Verwerfungsmethode mit dem auf die GIG-Verteilung zugeschnittenen Verfahren zur Erzeugung einer großen Stichprobe nicht konkurrenzfähig ist. Die Tangentenmethode benötigt für die Generierung einer Zufallszahl für die betrachteten Parameterkombinationen etwa doppelt so lang wie die Quotientenmethode. Die Werte der Sekantenmethode weisen diese als die deutlich langsamste Methode aus.

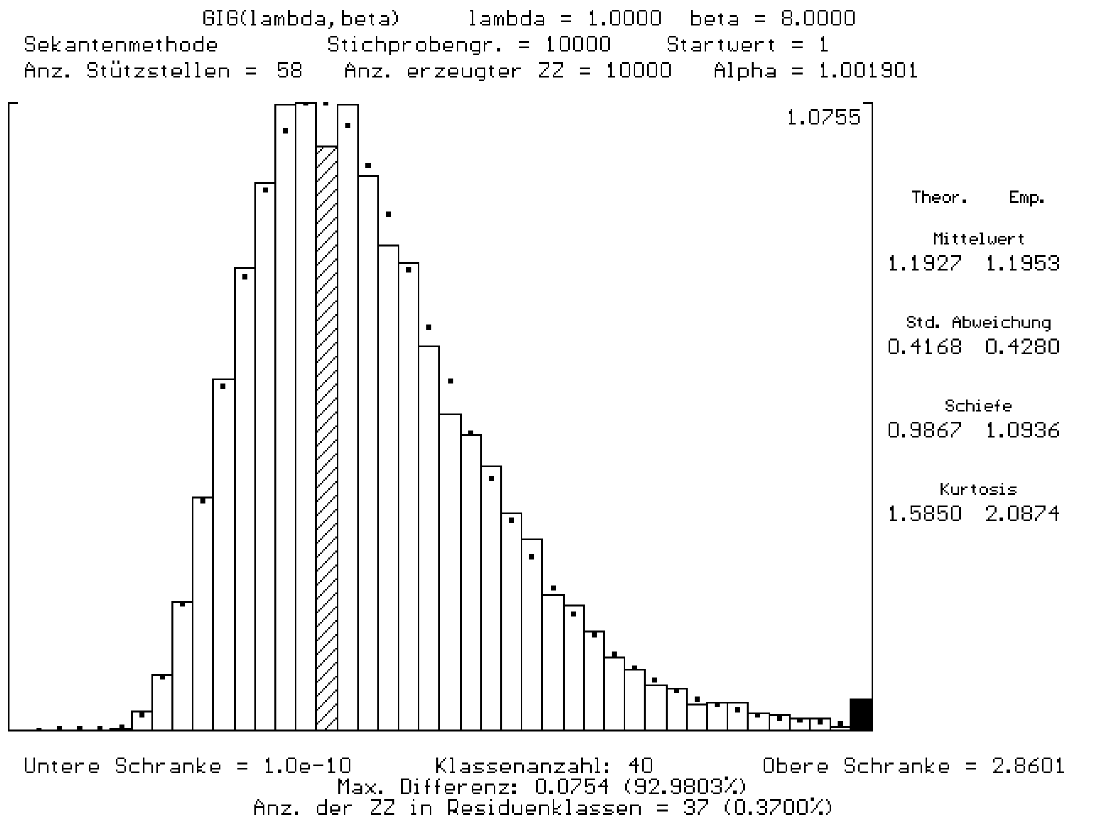
Auch wenn die Anwendung der adaptiven Verwerfungsmethode zur Erzeugung von Zufallszahlen einer bestimmten Verteilung nicht ratsam zu sein scheint, ist ihre Anwendung im Gibbs-Generator, für den sie ursprünglich entwickelt wurde, sinnvoll. Ein Gibbs-Erzeugungsprozeß besteht aus der Generierung jeweils einer Zufallszahl aus sich *ständig ändernden* vollständig bedingten Verteilungen (s. Abschnitt 1.3.4). Für diese spezielle Anwendung ist deshalb eine Universalmethode zu verwenden, die neben relativ kurzen Erzeugungszeiten eine einfache Initialisierung aufweist.

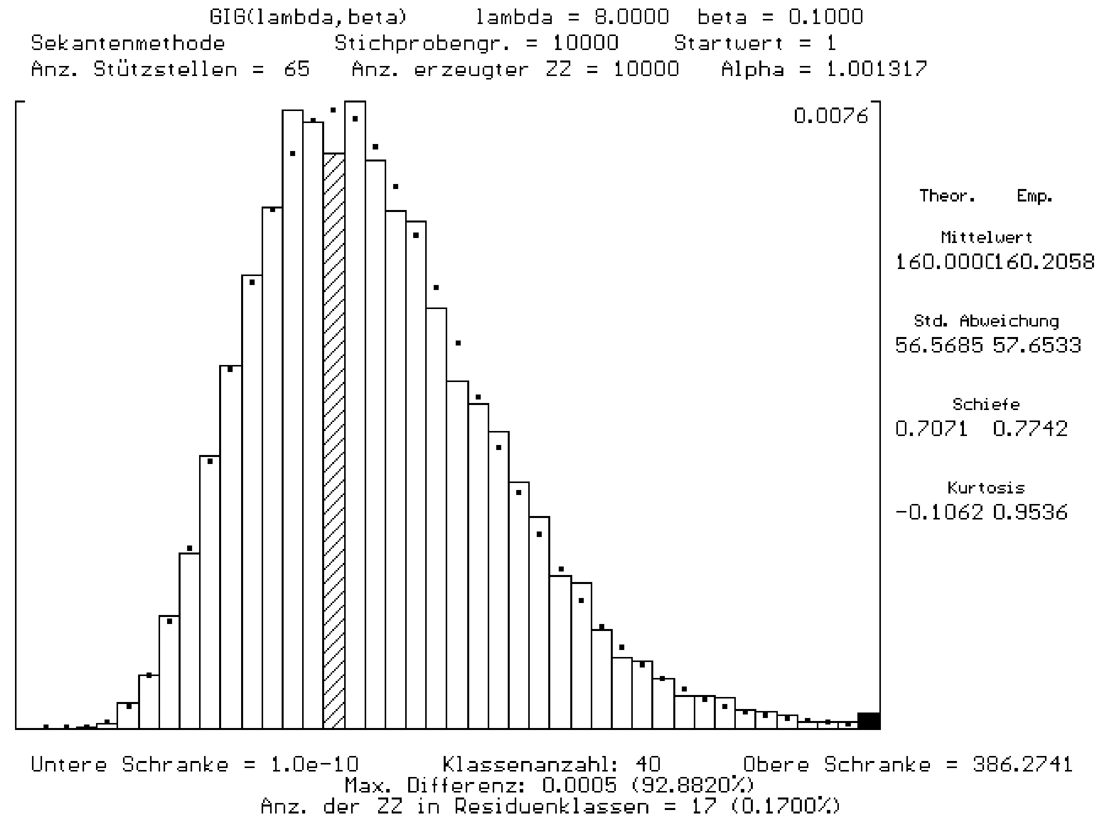
Der Initialisierungs- und Erzeugungsprozeß kann bei den adaptiven Methoden nicht vollständig getrennt werden, da Adaptionen der Hutfunktion nach jeder Verwerfung stattfinden. Die in Tabelle 4.8 dargestellten Werte für die Tangenten- und die Sekantenmethode umfassen damit neben der reinen Erzeugungszeit auch die Dauer, die zur Konstruktion der Hutfunktion nötig ist. Bei herkömmlichen Verfahren wie der Quotientenmethode ist noch zusätzlich der Aufwand zu berücksichtigen, der zur Initialisierung bei jeder Parameteränderung notwendig ist.

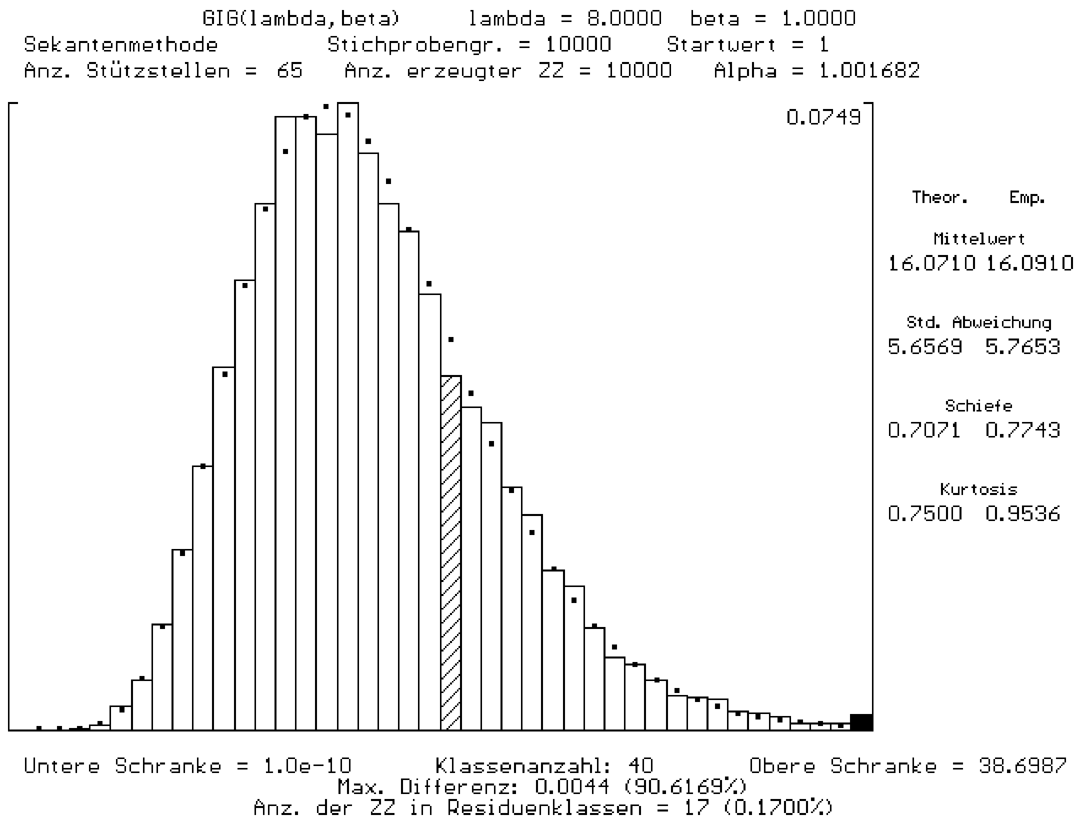
Die in Tabelle 4.8 dargestellten Zeiten zeigen, daß die Sekantenmethode aufgrund ihrer vergleichsweise komplizierten Hutfunktion (s. Abschnitt 4.4.3) für die praktische Anwendung nicht so geeignet ist wie die Tangentenmethode, der auch dann der Vorzug zu geben ist, wenn die Berechnung der Ableitung der Dichtefunktion aufwendig ist. Bei der Anwendung des Gibbs-Generators in Bayes-Modellen kann auf diese Situation Rücksicht genommen werden, indem bereits bei der Modellierung (Spiegelhalter *et al.*, 1996) die Wahl bestimmter Verteilungen vermieden und dadurch die Geschwindigkeit der Tangentenmethode erhöht wird.

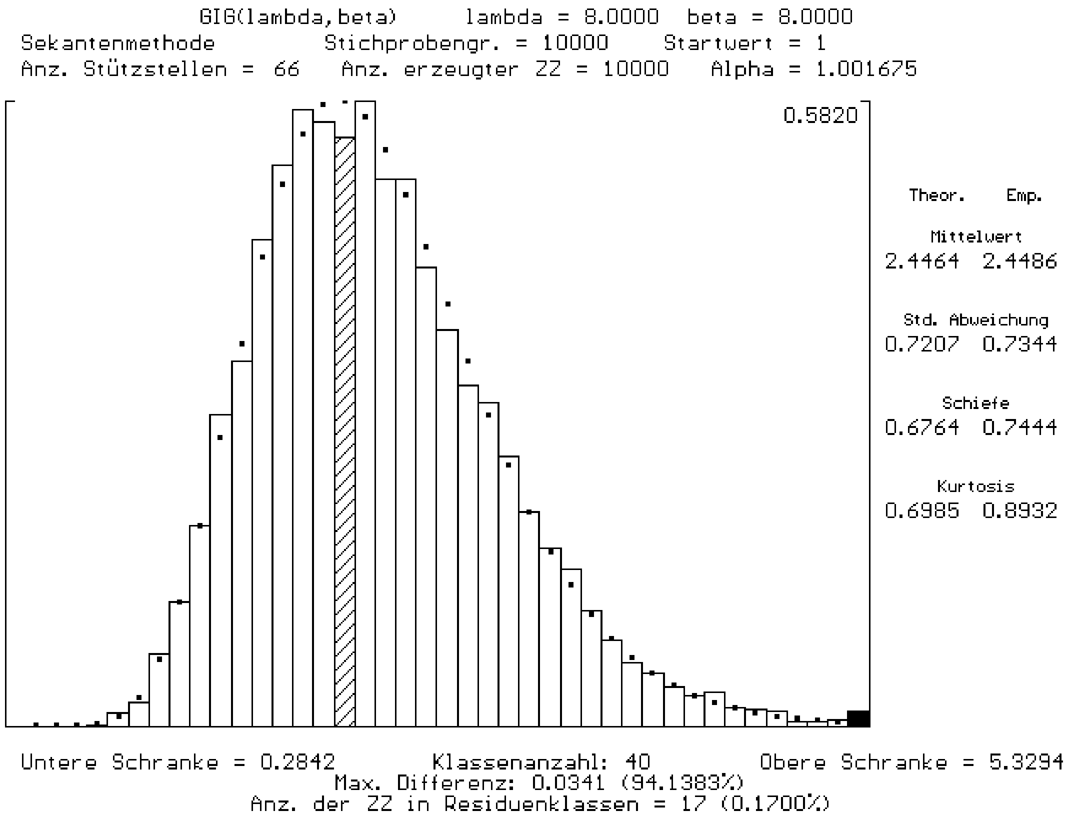

GIG(lambda,beta) lambda = 1.0000 beta = 0.1000
Sekantenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 23 Anz. erzeugter ZZ = 10000 Alpha = 1.001128
0.0458
Theor. Emp.
Mittelwert
20.2516 20.4241
Std. Abweichung
20.0490 20.7426
Schiefe
1.9852 2.1268
Kurtosis
5.9415 6.9359
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 100.4475
Max. Differenz: 0.0016 (94.5643%)
Anz. der ZZ in Residuenklassen = 85 (0.8500%)


GIG(lambda,beta) lambda = 1.0000 beta = 1.0000
Sekantenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 47 Anz. erzeugter ZZ = 10000 Alpha = 1.002623
0.3104
Theor. Emp.
Mittelwert
2.7015 2.7189
Std. Abweichung
2.1243 2.1984
Schiefe
1.8032 1.9369
Kurtosis
4.9930 5.9057
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 11.1986
Max. Differenz: 0.0150 (168.8217%)
Anz. der ZZ in Residuenklassen = 79 (0.7900%)


GIG(lambda,beta) lambda = 1.0000 beta = 8.0000
Sekantenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 58 Anz. erzeugter ZZ = 10000 Alpha = 1.001901
1.0755
Theor. Emp.
Mittelwert
1.1927 1.1953
Std. Abweichung
0.4168 0.4280
Schiefe
0.9867 1.0936
Kurtosis
1.5850 2.0874
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 2.8601
Max. Differenz: 0.0754 (92.9803%)
Anz. der ZZ in Residuenklassen = 37 (0.3700%)


GIG(lambda,beta) lambda = 8.0000 beta = 0.1000
Sekantenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 65 Anz. erzeugter ZZ = 10000 Alpha = 1.001317
0.0076
Theor. Emp.
Mittelwert
160.0000160.2058
Std. Abweichung
56.5685 57.6533
Schiefe
0.7071 0.7742
Kurtosis
-0.1062 0.9536
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 386.2741
Max. Differenz: 0.0005 (92.8820%)
Anz. der ZZ in Residuenklassen = 17 (0.1700%)


GIG(lambda,beta) lambda = 8.0000 beta = 1.0000
Sekantenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 65 Anz. erzeugter ZZ = 10000 Alpha = 1.001682
0.0749
Theor. Emp.
Mittelwert
16.0710 16.0910
Std. Abweichung
5.6569 5.7653
Schiefe
0.7071 0.7743
Kurtosis
0.7500 0.9536
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 38.6987
Max. Differenz: 0.0044 (90.6169%)
Anz. der ZZ in Residuenklassen = 17 (0.1700%)


GIG(lambda,beta) lambda = 8.0000 beta = 8.0000
Sekantenmethode Stichprobengr. = 10000 Startwert = 1
Anz. Stützstellen = 66 Anz. erzeugter ZZ = 10000 Alpha = 1.001675
0.5820
Theor. Emp.
Mittelwert
2.4464 2.4486
Std. Abweichung
0.7207 0.7344
Schiefe
0.6764 0.7444
Kurtosis
0.6985 0.8932
Untere Schranke = 0.2842 Klassenanzahl: 40 Obere Schranke = 5.3294
Max. Differenz: 0.0341 (94.1383%)
Anz. der ZZ in Residuenklassen = 17 (0.1700%)

4.6 Analyse der adaptiven Verwerfungsmethode mit Metropolis-Hastings-Schritt

Die in Abschnitt 3.3 vorgestellte adaptive Verwerfungsmethode mit Metropolis-Hastings-Schritt (ARMS) ist im Gegensatz zur herkömmlichen adaptiven Verwerfungsmethode auch auf nicht-logkonkave Verteilungen anwendbar. Die Dichte $g(\cdot)$, von der Zufallszahlen erzeugt werden, wird durch eine Funktion $h_r(\cdot)$, die ähnlich wie bei der Sekantenmethode durch Sekanten in r Stützstellen konstruiert wird, approximiert. Die Funktion $h_r(\cdot)$ ist für nicht-logkonkave Verteilungen zwar im allgemeinen keine Hutfunktion, sie kann aber zur Ausführung eines Metropolis-Hastings-Schrittes verwendet werden. Die mit ARMS erzeugten Zufallszahlen sind deshalb nicht unabhängig.

Die GIG(λ,β)-Verteilung, mit deren Hilfe in den vorangegangenen Abschnitten die Tangenten- und Sekantenmethode untersucht wurde, ist für $\lambda \geq 1$ logkonkav. Diese Bedingung muß nun für ARMS nicht berücksichtigt werden. ARMS ist für logkonkave Verteilungen identisch mit der Sekantenmethode, weshalb es genügt, die Zufallszahlenerzeugung aus der GIG(λ,β)-Verteilung für den Parameterbereich $0 \leq \lambda < 1$ zu betrachten. Sie ist für diese Wahl von α aus $[0, 1)$ aber „fast logkonkav“, d. h. die Abweichung vom logkonkaven Verteilungstyp ist so gering, daß die Funktion $h_r(\cdot)$ nur in kleinen Intervallen des Definitionsbereichs D keine Majorante der Dichte ist und das Verhalten von ARMS bei dieser Parameterwahl sehr ähnlich der Sekantenmethode ist.

Um die Funktionsweise von ARMS für nicht-logkonkave Verteilungen zu studieren, wurden zusätzlich die Burr $V(r,k)$-Verteilung (Burr, 1942) betrachtet. Diese Verteilung weicht sehr stark vom logkonkaven Verteilungstyp ab, und sie weist für bestimmte Kombinationen der Gestaltparameter r und k zwei Modi auf. Mit der Inversionsmethode steht ein sehr schnelles Standardverfahren zur Erzeugung einer Stichprobe der Burr $V(r,k)$-Verteilung zur Verfügung.

4.6.1 Initialisierung der Stützstellen

Bei der Frage, wie die Stützstellen für die Tangenten- und Sekantenmethode am besten zu wählen sind, wurde in den Abschnitten 4.3.1 und 4.4.1 die Effizienz α als Maß für die Güte der Anpassung der Hutfunktion $h_r(\cdot)$ an die nicht-normierte Dichte $g(\cdot)$ verwendet. Da für ARMS im allgemeinen nicht $h_r(x) \geq g(x) \ \forall x \in D$ gilt, kann $\alpha = \int_D h_r(u)du / \int_D g(v)dv$ grundsätzlich nichts über die Güte der Anpassung von $h_r(\cdot)$ an die Dichtefunktion aussagen. Im Programm `ARMS` wird α trotzdem für jede Verteilung angezeigt, da α vor allem für fast logkonkave Verteilungen einen Anhaltspunkt über die Güte der Anpassung bietet. Für Verteilungen, die sehr stark vom logkonkaven Verteilungstyp abweichen, kann α in Zusammenhang mit der Graphik von $h_r(\cdot)$ einen Eindruck von der Größenordnung des Unterschiedes von $h_r(\cdot)$ und der Dichte bieten.

Als Maß der Abweichung wäre etwa $I_\Delta = \int_D |g(x) - h_r(x)|dx$ geeignet, aber der Aufwand einer Integralberechnung wie für I_Δ zur Optimierung der Initialisierung der Stützstellen scheint nicht gerechtfertigt, da die Adaptionshäufigkeit von ARMS für eine kleine Anzahl r an Startstützstellen (etwa $r = 3$) in jedem Fall zu hoch ist.

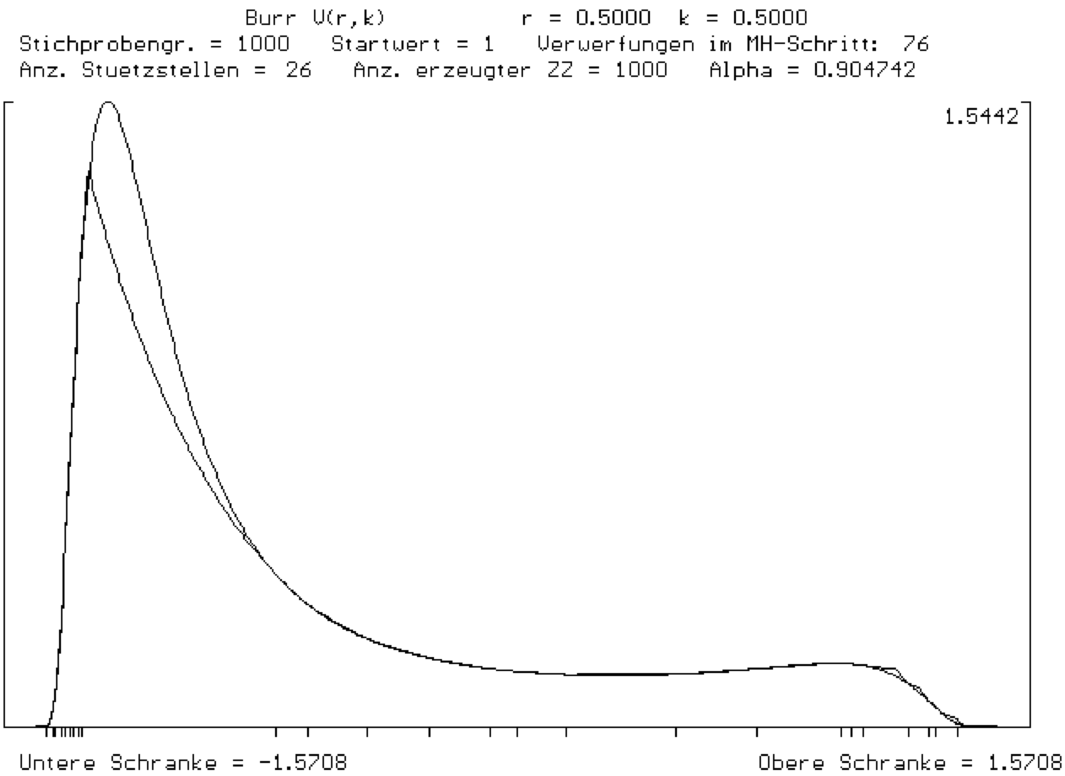


Abbildung 4.8: *Nach 1000 Iterationen wird die Burr V(0.5,0.5)-Verteilung für eine schlechte Intialisierung der Menge T_r (die $r = 3$ Stützstellen wurden rechts vom Modus gewählt) nur schlecht durch die Funktion $h_r(\cdot)$ angenähert. Im Bereich des linken Modus ist $h_r(\cdot)$ deutlich kleiner als die zu approximierende Dichtefunktion.*

Wenn r zu klein gewählt wird, kann zudem das Problem auftreten, daß die Funktion $h_r(\cdot)$ die Dichte $g(\cdot)$ auch nach vielen Adaptionen nur schlecht approximiert, wie folgende Überlegung zeigt: Eine Adaption wird vorgenommen, wenn die Bedingung

$$U > \frac{g(Y)}{h_r(Y)} \tag{4.6}$$

erfüllt ist, wobei U eine gleichverteilte Zufallszahl aus dem Intervall $(0, 1)$ ist und Y ein aus der Dichte $h_r(\cdot)/\int_D h_r(u)du$ erzeugter Kandidat für den Metropolis-Hastings-Schritt (s. Definition des Algorithmus ARMS aus Abschnitt 3.3). Wenn Y aus einem Intervall $[a, b]$ stammt, für das $g(Y) > h_r(Y)$ gilt, wird $h_r(\cdot)$ in diesem Intervall nicht adaptiert. Die Funktion $h_r(\cdot)$ unterliegt zwar aufgrund ihrer Konstruktion mit

$$h_r(x) = \max\Big\{L_{i,i+1}(x; T_r), \min\{L_{i-1,i}(x; T_r), L_{i+1,i+2}(x; T_r)\}\Big\} \tag{4.7}$$

dem Einfluß von Adaptionen, die knapp außerhalb von $[a, b]$ vorgenommen werden. Enthält das Intervall $[a, b]$ den Modus der Dichte $g(\cdot)$, kann die in Abbildung 4.8 dargestellte Situation eintreten.

Nach 1000 erzeugten Zufallszahlen und 26 Adaptionen von $h_r(\cdot)$ wird $g(\cdot)$ im Bereich des Modus nur sehr schlecht von $h_r(\cdot)$ approximiert. ARMS erzeugt zwar auch in dieser Situation Zufallszahlen, die gemäß $g(\cdot)$ verteilt sind, im Metropolis-Hastings-Schritt werden

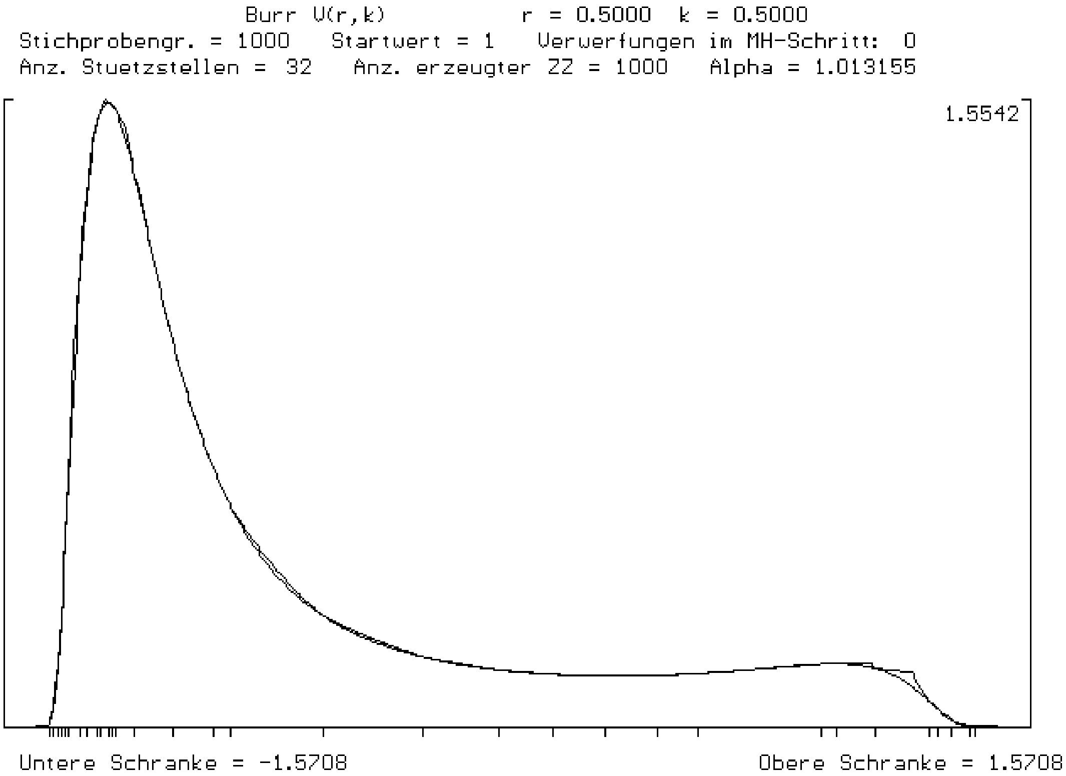


Abbildung 4.9: *Nach 1000 Iterationen approximiert die Funktion $h_r(\cdot)$, nachdem die 6 Stützstellen mit Quantilen der Burr V-Verteilung initialisiert wurden, sehr gut.*

jedoch vergleichsweise viele Kandidaten Y verworfen und die Mischung der resultierenden Markovkette ist schlecht.

Um dem vorzubeugen, schlagen Gilks *et al.* (1995) vor, für Verteilungen, die stark vom logkonkaven Typ abweichen, für den Gibbs-Erzeugungsprozeß zu Beginn die Anzahl der Stützstellen mit $r = 6$ festzusetzen und die Stützstellen mit den 5%-, 30%-, 45%-, 55%-, 70%- und 95%-Prozent-Quantilen zu initialisieren. Diese Wahl scheint auch für die Burr V-Verteilung gut geeignet. Außerdem können die Quantile aufgrund der Invertierbarkeit der Verteilungsfunktion leicht bestimmt werden. Diese Wahl bringt eine deutlich bessere Anpassung von $h(\cdot)$ an die Dichte, wie Abbildung 4.9 verdeutlicht.

Die GIG-Verteilung ist für $0 \leq \lambda < 1$ fast logkonkav, sodaß es genügt, $r = 4$ Stützstellen wie für die Sekantenmethode (4.5) zu bestimmen (s. Abbildung 4.10).

4.6.2 Adaptionshäufigkeit

Mit den im vorangegangenen Abschnitt erläuterten Initialisierungen der Stützstellen wurden nun für mehrere Parameterkombinationen der GIG- und Burr V-Verteilung Simulationen durchgeführt, um die mittlere Adaptionshäufigkeiten von ARMS für verschiedene Verteilungstypen zu untersuchen. Tabelle 4.9 und 4.10 zeigen, daß für die GIG-Verteilung im Durchschnitt deutlich weniger Adaptionen der Hutfunktion für eine gleich große Anzahl an erzeugten Zufallszahlen nötig sind als für die Burr V-Verteilung.

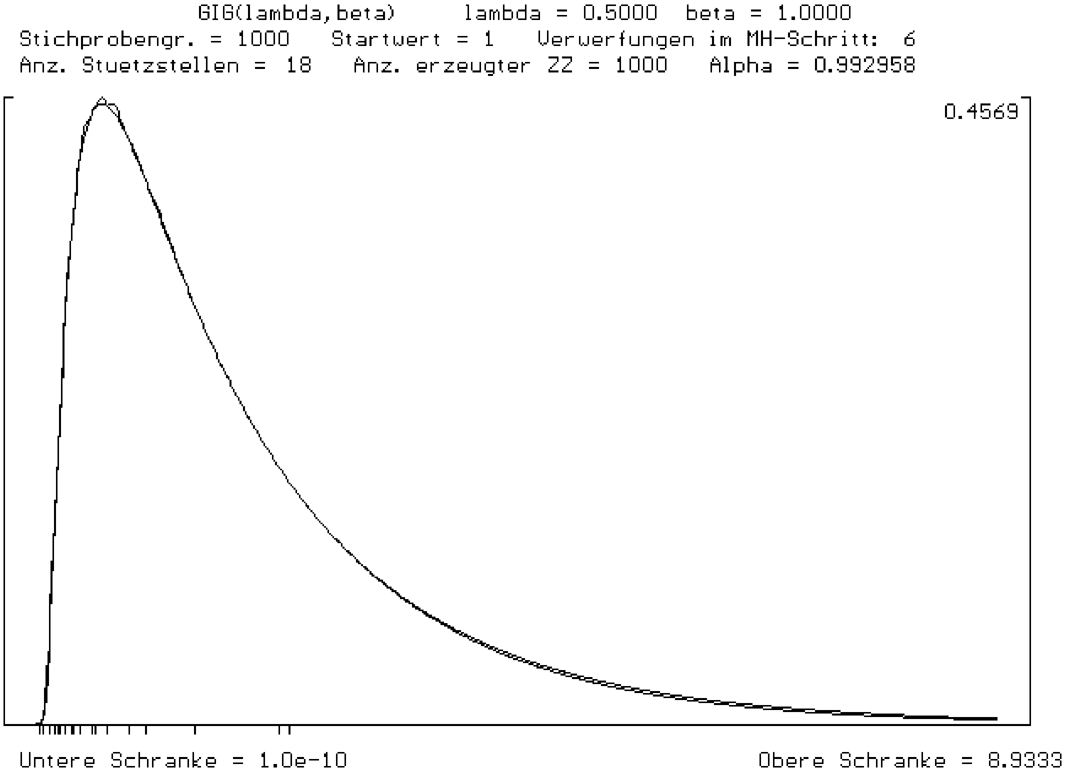


Abbildung 4.10: *Weil die GIG(λ,β)-Verteilung fast logkonkav ist, können die Stützstellen wie in der Sekantenmethode initialisiert werden. Die Abbildung zeigt, daß die Dichte nach 1000 erzeugten Zufallzahlen durch $h(\cdot)$ sehr gut angenähert wird.*

4.6.3 Erzeugungszeit

An die Dichtefunktion $g(\cdot)$, von denen mit Hilfe von ARMS Zufallszahlen erzeugt werden sollen, wird nicht die Bedingung der Logkonkavität gestellt. Diese Tatsache hat großen Einfluß auf die Ausführungszeit der Methode. Es ist beispielsweise nicht möglich, für ARMS eine Squeezefunktion zu definieren, die es erlaubt, im Verwerfungsschritt die Auswertung der Dichtefunktion im Kandidaten Y zu vermeiden, was folgende Überlegung zeigt: Wie in Abschnitt 3.3 erläutert, sind die im Verwerfungsschritt von ARMS erzeugten Kandidaten gemäß der Dichte

$$f(Y) = \min\{g(Y), h_r(Y)\} \tag{4.8}$$

verteilt, wobei $h_r(\cdot)$ eine die Dichte $g(\cdot)$ approximierende lineare Funktion ist. $h_r(\cdot)$ wird aus den Sekanten durch die r Stützstellen definiert. Durch die Minimalbedingung in (4.8) gibt es keine zur Zeit des Verwerfungsschrittes bekannte lineare Funktion, die durch $f(\cdot)$ majorisiert würde. In Abbildung 4.11 ist diese Situation für die Burr V(1,1)-Verteilung dargestellt, die für diese Parameterkonstellation zwei Modi besitzt. Damit ergibt sich keine Squeezefunktion direkt aus der Konstruktion der Methode, wie es etwa bei der Tangenten- oder Sekantenmethode der Fall war. Daß das Fehlen einer Squeezefunktion sich deutlich auf die Erzeugungsgeschwindigkeit auswirkt, ist besonders für die Burr V-Verteilung zu beobachten, deren Auswertung der Dichte relativ aufwendig ist (bei der Implementierung der logarithmierten Burr V-Dichte mußten zusätzlich noch Abfragen eingebaut werden,

	$\lambda = 0$											
	$\beta = 0.1$				$\beta = 1$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.64	0	0	2	0.13	0	0	1	0.25	0	0	1
10	2.30	2	1	4	0.84	1	0	2	2.31	2	0	4
100	6.72	7	4	10	4.33	4	3	6	8.85	9	7	12
1000	18.41	18	15	22	12.07	12	9	15	23.22	23	19	27
10000	42.89	43	37	48	30.95	31	26	36	53.43	53	48	59

	$\lambda = 0.5$											
	$\beta = 0.1$				$\beta = 1$				$\beta = 8$			
N	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	$\bar{R}$	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.29	0	0	1	0.11	0	0	1	0.25	0	0	1
10	1.51	1	0	3	0.87	1	0	2	2.31	2	1	4
100	4.50	4	2	7	4.48	4	2	6	9.01	9	6	12
1000	12.98	13	10	17	13.34	13	11	17	23.33	23	20	27
10000	32.46	32	28	37	31.62	32	27	37	53.49	53	48	59

Tabelle 4.9: *Untersuchung der Adaptionshäufigkeiten von ARMS angewendet auf die GIG(λ,β)-Verteilung für mehrere Parameterkombinationen, für die je 100 Simulationsläufe durchgeführt wurden. Die mittlere Anzahl der Adaptionen $\bar{R}$ ist gemeinsam mit dem Median $q_{0.5}$ und den Quartilen $q_{0.25}$ und $q_{0.75}$ der empirischen Verteilung der Adaptionen angegeben.*

um Überläufe während der Exekution des Programms zu verhindern; kritische Situationen ergaben sich für Stützstellen am Rand des Definitionsbereiches $(-\pi/2, \pi/2)$ der Burr V-Verteilung, wo die logarithmierte Dichtefunktion sehr große negative Werte aufweist.)

Bei den empirischen Versuchen mit den Testverteilungen wurde klar, daß der größte negative Einfluß auf die Erzeugungszeit auf der Tatsache beruht, daß die Funktion $h_r(\cdot)$, anders als bei der Tangenten- und Sekantenmethode, nicht geschlossen implementiert werden konnte. Um dies zu veranschaulichen, genügt ein Vergleich der Definitionen von $h_r(\cdot)$ für die unterschiedlichen Methoden. Während für die Sekantenmethode die Definition der Funktion

$$h_r(x) = \min\{L_{i-1,i}(x;T_r), L_{i+1,i+2}(x;T_r)\}, \qquad x_i \le x \le x_{i+1},$$

nur eine Minimalbedingung enthält, wird die Definition von $h_r(\cdot)$ noch durch eine Maximalbedingung ergänzt, um auch auf nicht-logkonkave Verteilungen anwendbar zu sein:

$$h_r(x) = \max\Big\{ L_{i,i+1}(x;T_r), \min\{L_{i-1,i}(x;T_r), L_{i+1,i+2}(x;T_r)\}\Big\}, \qquad x_i \le x \le x_{i+1}.$$

Für die Sekantenmethode konnte $h_r(\cdot)$ aufgrund der einfacheren Konstruktion so im Programm `ARS` implementiert werden, daß für eine Funktionsauswertung kein Minimumvergleich notwendig war. Es konnte ein Algorithmus gefunden werden, der für jeden Punkt x des Definitionsbereiches die lineare Funktion $L(x;T_r)$, durch die $h_r(\cdot)$ im Punkt x definiert ist, direkt bestimmt. Für ARMS steht kein derartiger Algorithmus zur Verfügung, weshalb für jede Funktionsauswertung von $h_r(\cdot)$ eine Minimal- und Maximalabfrage durchgeführt wird. Es ist zu berücksichtigen, daß diese Abfragen nicht nur bei Funktionsauswertungen, sondern auch bei der Kandidatenerzeugung auftreten.

	$r=1$							
	$k=1$				$k=2$			
N	R	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	R	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.24	0	0	1	0.32	0	0	1
10	2.03	2	0	4	2.	2	0	5
100	7.64	8	5	11	8.79	9	6	12
1000	21.80	22	18	27	24.52	24	20	30
10000	54.43	54	47	61	57.81	57	52	65

	$r=1$							
	$k=1$				$k=2$			
N	R	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$	R	$q_{0.5}$	$q_{0.25}$	$q_{0.75}$
1	0.21	0	0	1	0.22	0	0	1
10	1.92	2	0	4	2.03	2	0	4
100	8.67	8	6	12	8.60	8	6	12
1000	23.29	23	20	27	24.79	25	20	30
10000	53.80	53	49	62	59.02	59	52	66

Tabelle 4.10: *ARMS weist für die Burr V(r,k)-Verteilung eine deutlich größere mittlere Adaptionshäufigkeit als für die fast logkonkave GIG-Verteilung (s. Tabelle 4.9) auf.*

Wie stark sich die beschriebenen negativen Einflüsse auf die Erzeugungszeiten in der Praxis auswirken, zeigt Tabelle 4.11 Zum Vergleich wurden die mittleren Erzeugungszeiten, die mit herkömmlichen Methoden erzielt wurden, angegeben. Für die GIG-Verteilung wurde die Quotientenmethode, für die Burr V-Verteilung die Inversionsmethode verwendet. Die Werte in Klammern sind die Initialisierungszeiten. Die Zeitmessungen für diese Methoden wurden mit dem Programm `c_rand` (Niederl, 1993) durchgeführt.

Mit `ARMS` wurden für die GIG- und die Burr V-Verteilung verschieden große Stichproben für mehrere Parameterkombinationen erzeugt, aus denen die mittlere Erzeugungszeit pro Zufallszahl berechnet wurde. Analog zur Tangenten- und Sekantenmethode verlangsamt sich auch ARMS für eine zunehmende Anzahl r an Stützstellen. Es sollte deshalb auch für ARMS ein Abbruchkriterium, das den Adaptionsprozeß nach dem Erreichen einer gewissen Güte der Anpassung von $g(\cdot)$ an die Dichte stoppt, verwendet werden. Wie im vorangegangenen Abschnitt erklärt wurde, kann α nicht als Vergleichsmaß dienen, da $h_r(\cdot)$ keine Hutfunktion ist. Die Adaptionen nach einer bestimmten Schranke für r zu beenden, ist ebenfalls problematisch, da die Schnelligkeit der Anpassung für verschiedene Verteilungen sehr unterschiedlich sein kann. Die mittleren Erzeugungszeiten in Tabelle 4.11 zeigen, daß eine Verwendung eines Abbruchkriteriums auch für die beiden Testverteilungen empfehlenswert ist, da die Zeiten stark vom Stichprobenumfang (und somit von der Anzahl der Stützstellen) abhängen.

Ein weiterer Punkt, der bei der Verwendung von ARMS berücksichtigt werden muß, ist die Untersuchung der Konvergenz und Mischung der Markovkette, die von den erzeugten Zufallszahlen aufgrund des Metropolis-Hastings-Schrittes gebildet wird. Die aus den Simulationen mit `ARMS` resultierenden Markovketten sind schnell konvergent, denn sie liefern schon für kleine Stichprobengrößen (z. B. für 100 Zufallszahlen) und beide Testverteilungen gute Näherungen der theoretischen statistischen Kenngrößen. Die Anzahl der

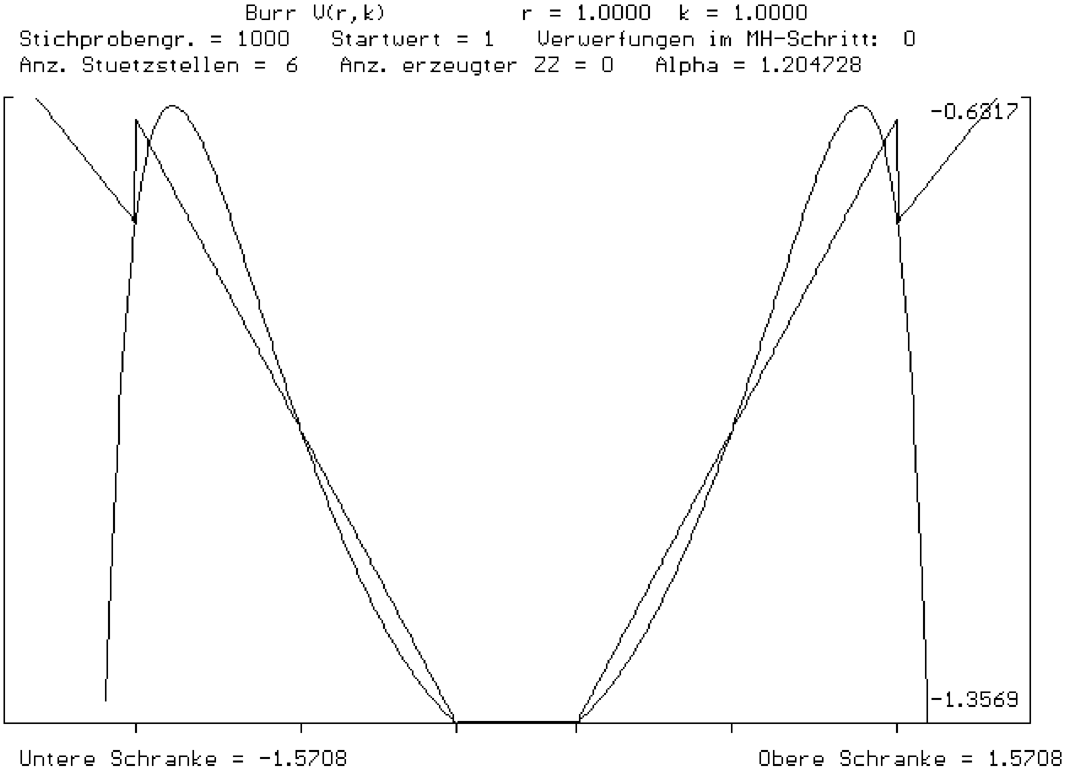


Abbildung 4.11: *Die Burr V-Verteilung wird nicht durch $h(\cdot)$ majorisiert. Aus den Sekanten durch die Stützstellen kann deshalb keine Squeezefunktion konstruiert werden.*

Verwerfungen im Metropolis-Hastings-Schritt übersteigt nicht 8% der Stichprobengröße, sodaß auf eine gute Mischung der Markovkette geschlossen werden kann.

4.7 Zusammenfassung

Wie die vorangegangenen Abschnitte gezeigt haben, können die adaptiven Verwerfungsmethoden eine interessante Alternative zu herkömmlichen Methoden der Generierung von Zufallszahlen darstellen. Sie können zwar mit Verfahren, die für bestimmte Verteilungen entwickelt wurden, im Hinblick auf die Geschwindigkeit nicht konkurrieren; aufgrund ihrer universalen Einsetzbarkeit sind sie aber für gewisse praktische Anwendungen wie beispielsweise MCMC-Methoden sehr nützlich.

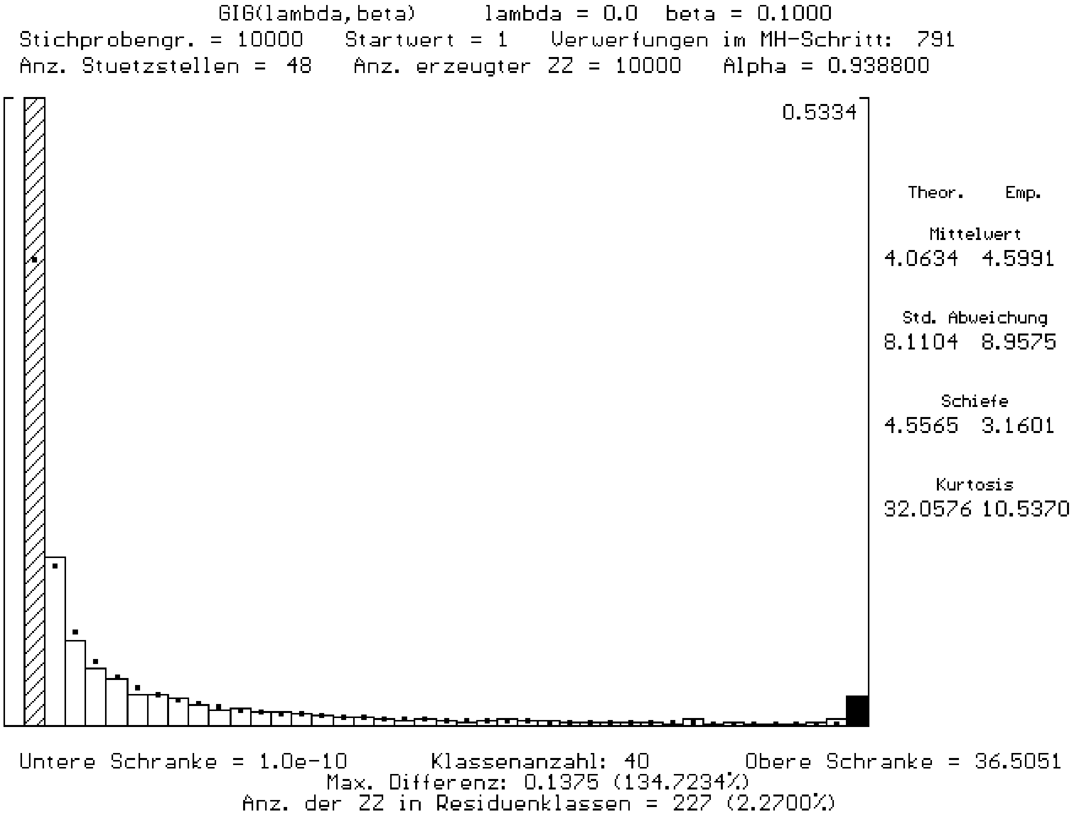
In den empirischen Untersuchungen konnten mit der Tangentenmethode für die untersuchten Testverteilungen wegen ihrer einfachen Implementierbarkeit und ihrer Schnelligkeit die besten Resultate erzielt werden. Eine Anwendung der Sekantenmethode scheint im – in der Praxis wahrscheinlich selten auftretenden – Fall, daß die Ableitung der logarithmierten Dichtefunktion nicht berechenbar ist, empfehlenswert. Gegenüber der Tangentenmethode müssen relativ große Einbußen in bezug auf die Anpassungs- und Erzeugungsgeschwindigkeit hingenommen werden. Die Ergebnisse, die mit der adaptiven Verwerfungsmethode mit Metropolis-Hastings-Schritt erzielt wurden, sind, was die Erzeugungszeit betrifft, im Vergleich zu den anderen Methoden nicht befriedigend. Die im Programm `ARMS` implementierte Form kann sicher noch optimiert werden. Daß dieses Verfahren auch zur Er-

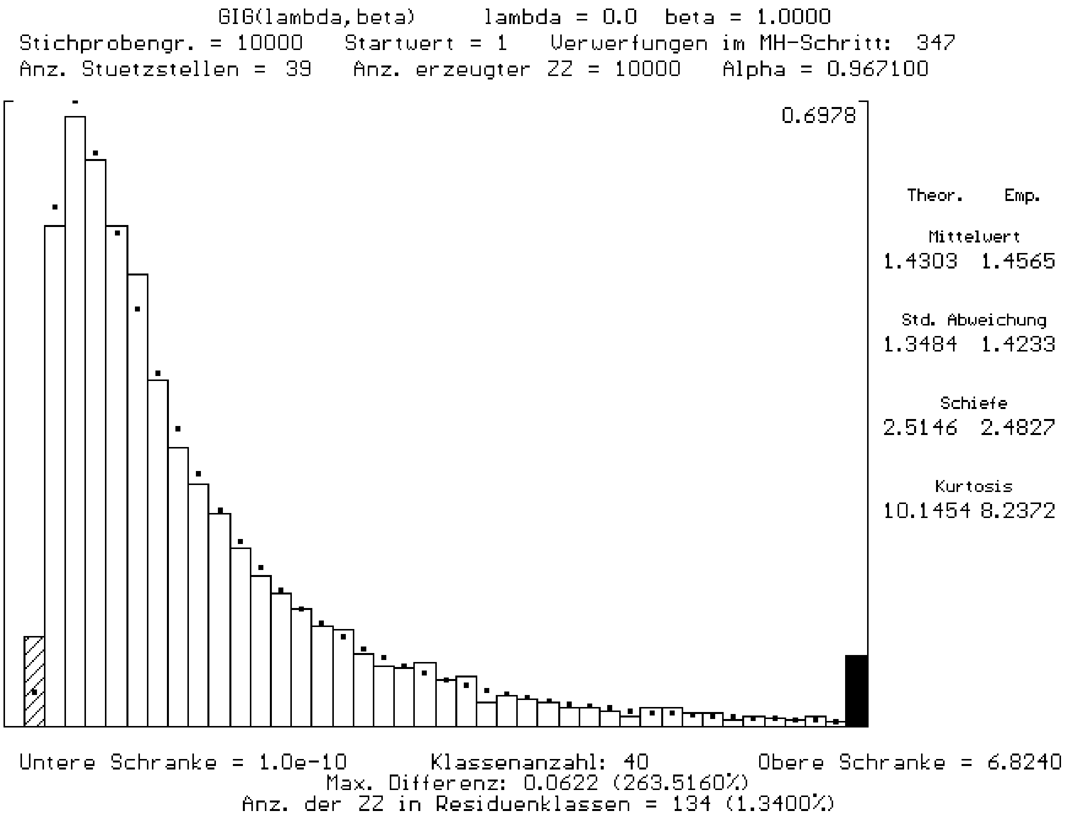
GIG(λ,β)			ARMS		
λ	β	Quotientenmeth.	$N = 5.000$	$N = 10.000$	$N = 50.000$
0	0.1	27.2 (17.0)	110.5	126.7	182.4
0	1	11.3 (33.1)	110.6	126.3	183.3
0	8	11.9 (49.5)	144.1	149.0	202.2
0.5	0.1	22.8 (14.3)	98.3	115.9	160.4
0.5	1	10.4 (39.7)	98.5	121.1	183.6
0.5	8	16.1 (41.0)	132	154.2	206.4

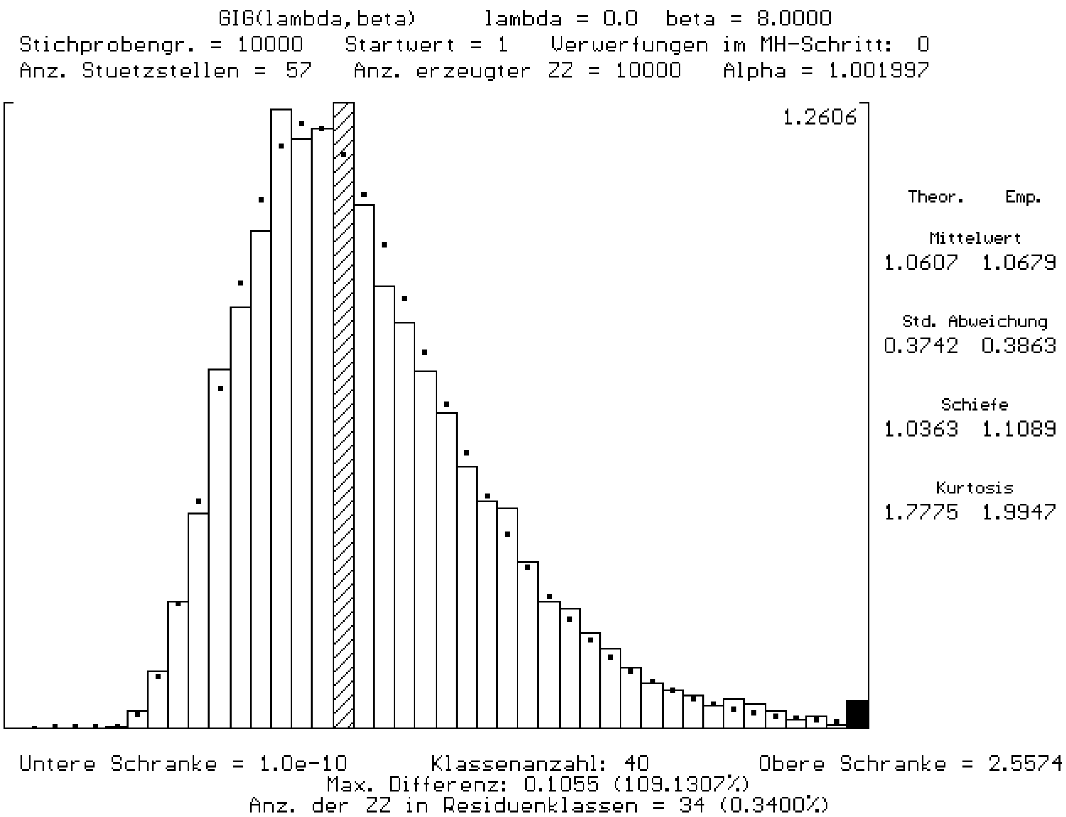
Burr $V(r,k)$			ARMS		
r	k	Inversionsmeth.	$N = 5.000$	$N = 10.000$	$N = 50.000$
1	1	10.3	154.2	175.8	241.8
1	2	17.2	164.7	182.1	236.2
2	1	16.4	166.0	175.3	216.4
2	2	17.5	176.9	198.0	262.4

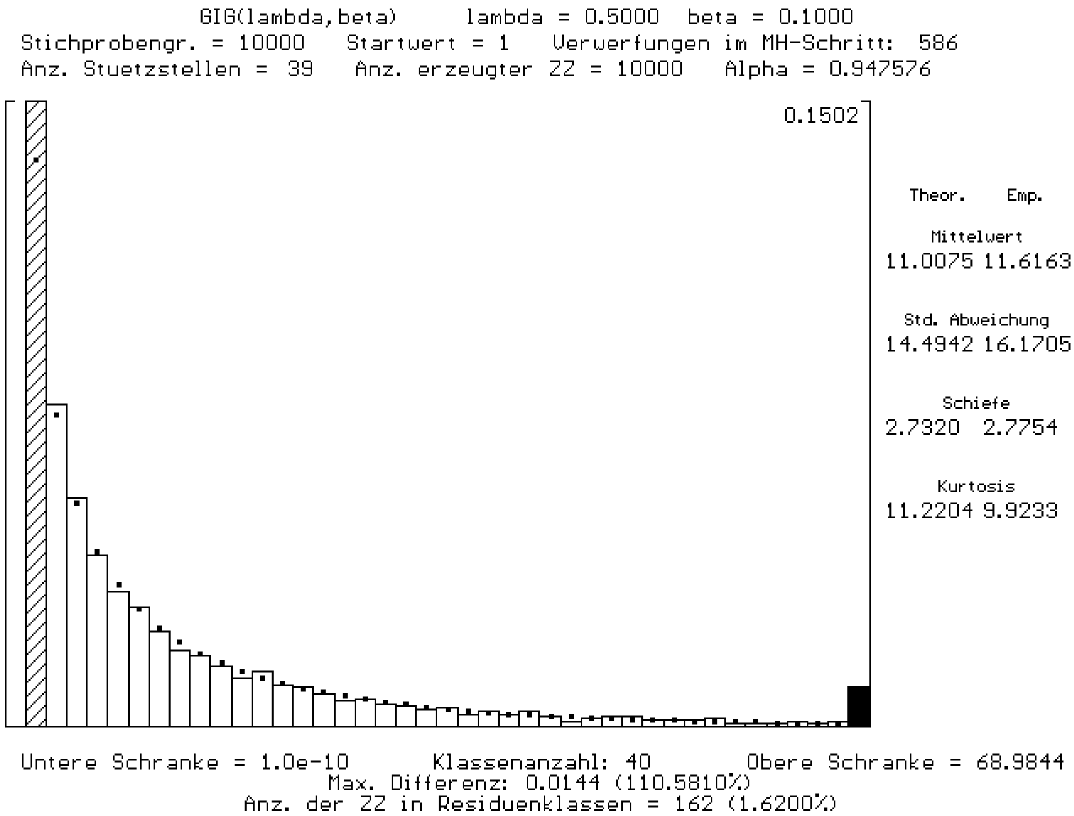
Tabelle 4.11: *Vergleich der mittleren Erzeugungszeit (in μsek) einer Zufallszahl für ARMS mit einer herkömmlichen Methode. Es wurden jeweils Stichproben der Größe N generiert. Die Zeitmessung erfolgte für die Erzeugung der letzten 5.000 Zahlen.*

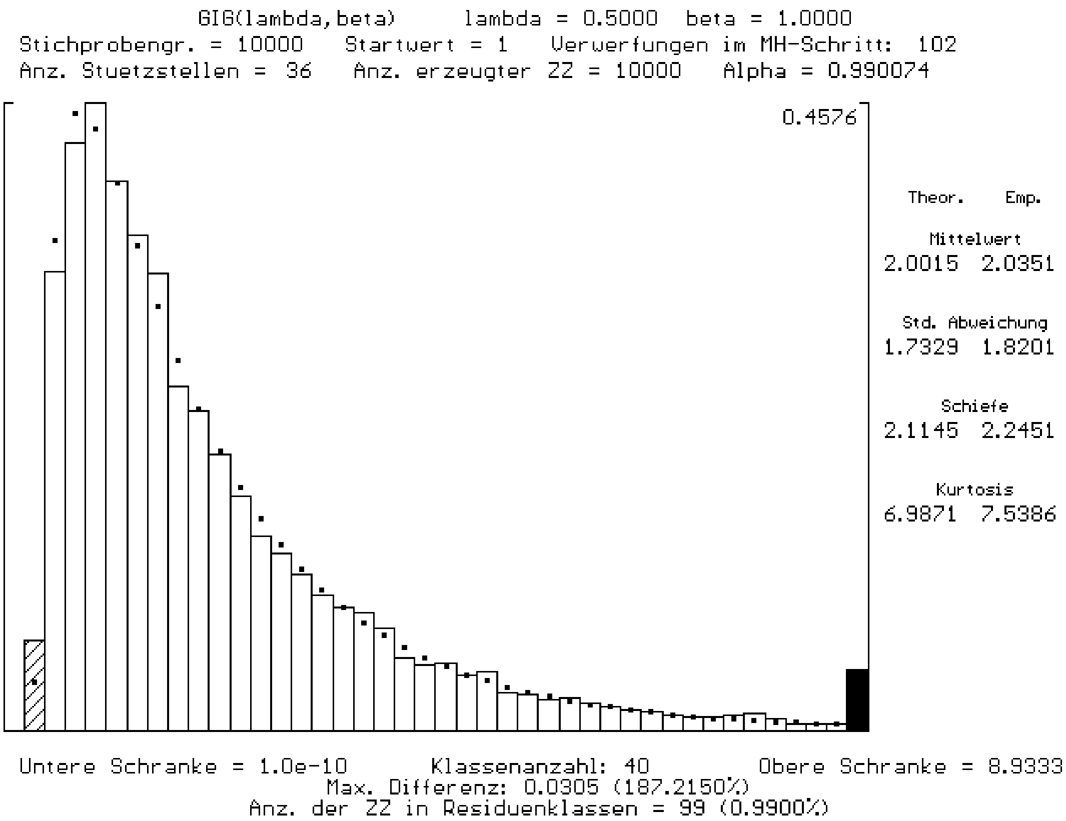
zeugung von Zufallszahlen aus nicht-logkonkaven Verteilungen verwendet werden kann, wurde durch die empirischen Untersuchungen jedenfalls bestätigt.

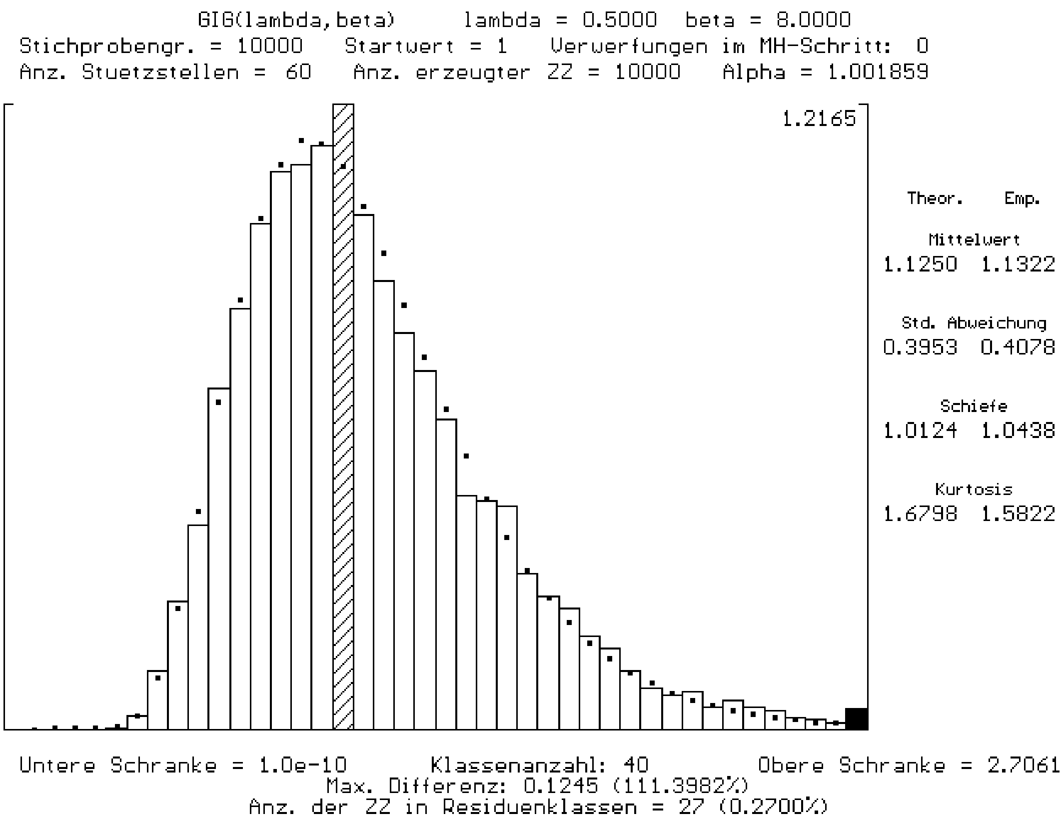

GIG(lambda,beta) lambda = 0.0 beta = 0.1000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 791
Anz. Stuetzstellen = 48 Anz. erzeugter ZZ = 10000 Alpha = 0.938800
0.5334
Theor. Emp.
Mittelwert
4.0634 4.5991
Std. Abweichung
8.1104 8.9575
Schiefe
4.5565 3.1601
Kurtosis
32.0576 10.5370
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 36.5051
Max. Differenz: 0.1375 (134.7234%)
Anz. der ZZ in Residuenklassen = 227 (2.2700%)

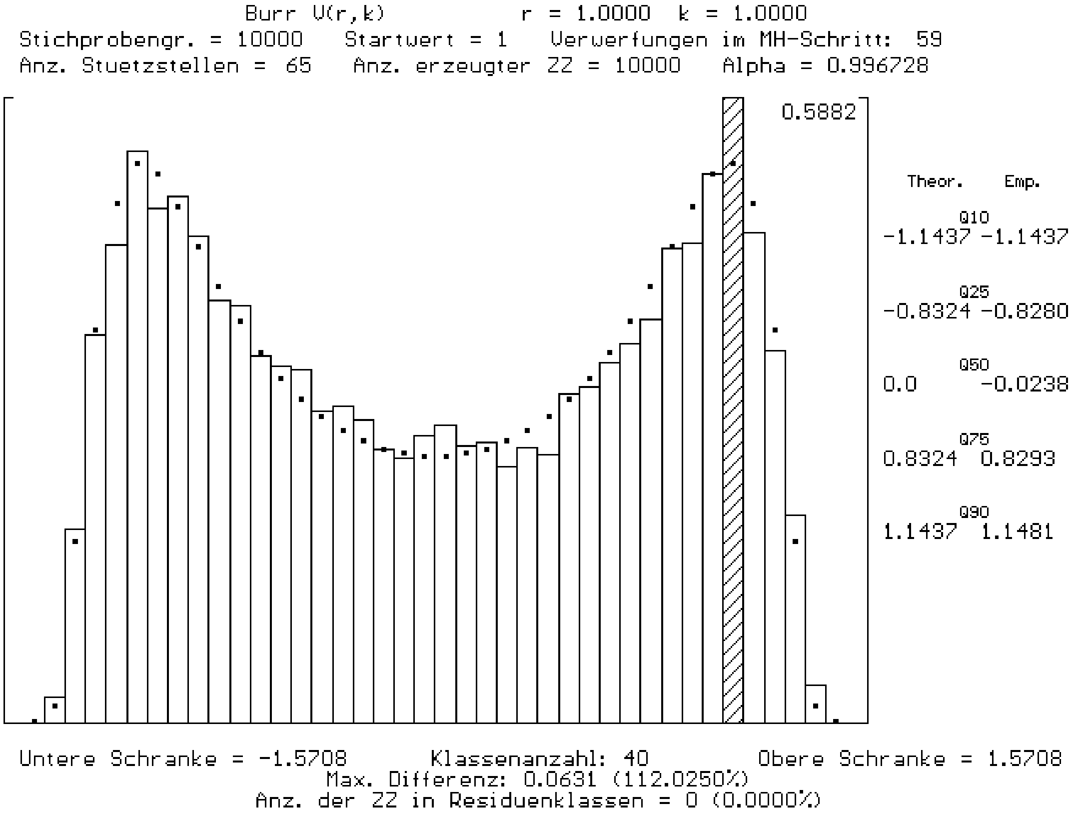

GIG(lambda,beta) lambda = 0.0 beta = 1.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 347
Anz. Stuetzstellen = 39 Anz. erzeugter ZZ = 10000 Alpha = 0.967100
0.6978
Theor. Emp.
Mittelwert
1.4303 1.4565
Std. Abweichung
1.3484 1.4233
Schiefe
2.5146 2.4827
Kurtosis
10.1454 8.2372
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 6.8240
Max. Differenz: 0.0622 (263.5160%)
Anz. der ZZ in Residuenklassen = 134 (1.3400%)

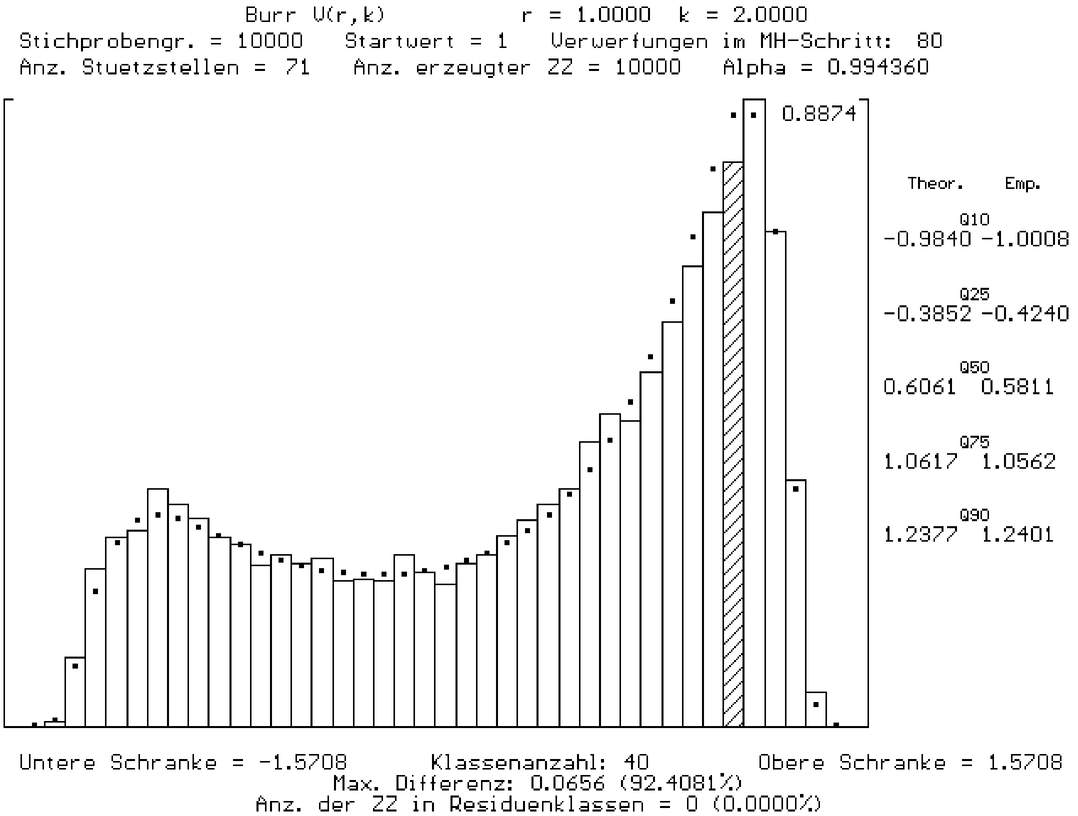

GIG(lambda,beta) lambda = 0.0 beta = 8.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 0
Anz. Stuetzstellen = 57 Anz. erzeugter ZZ = 10000 Alpha = 1.001997
1.2606
Theor. Emp.
Mittelwert
1.0607 1.0679
Std. Abweichung
0.3742 0.3863
Schiefe
1.0363 1.1089
Kurtosis
1.7775 1.9947
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 2.5574
Max. Differenz: 0.1055 (109.1307%)
Anz. der ZZ in Residuenklassen = 34 (0.3400%)

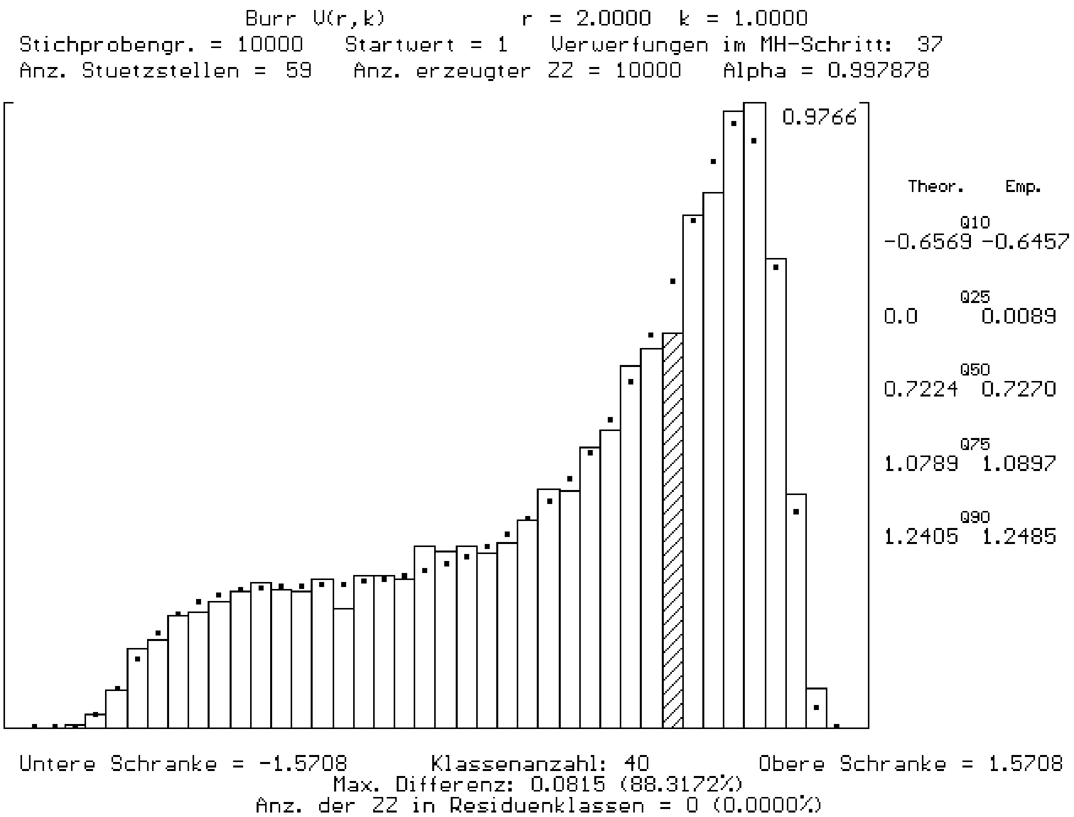

GIG(lambda,beta) lambda = 0.5000 beta = 0.1000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 586
Anz. Stuetzstellen = 39 Anz. erzeugter ZZ = 10000 Alpha = 0.947576
0.1502
Theor. Emp.
Mittelwert
11.0075 11.6163
Std. Abweichung
14.4942 16.1705
Schiefe
2.7320 2.7754
Kurtosis
11.2204 9.9233
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 68.9844
Max. Differenz: 0.0144 (110.5810%)
Anz. der ZZ in Residuenklassen = 162 (1.6200%)

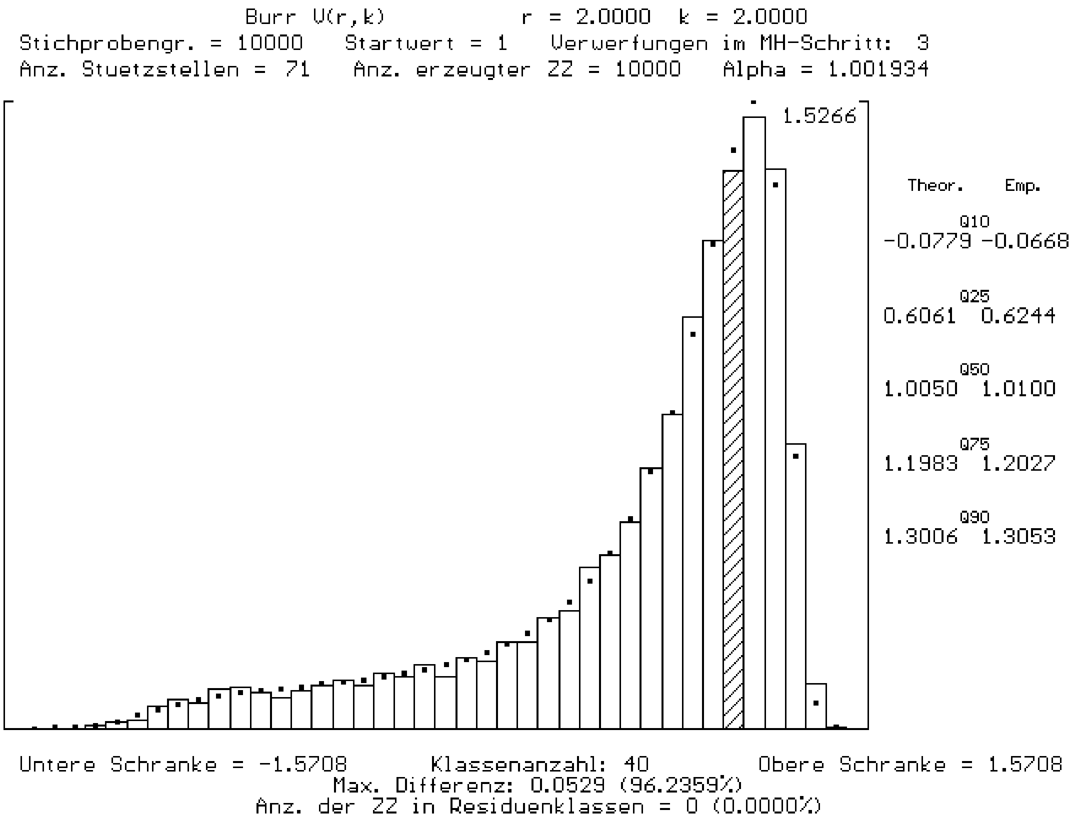

GIG(lambda,beta) lambda = 0.5000 beta = 1.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 102
Anz. Stuetzstellen = 36 Anz. erzeugter ZZ = 10000 Alpha = 0.990074
0.4576
Theor. Emp.
Mittelwert
2.0015 2.0351
Std. Abweichung
1.7329 1.8201
Schiefe
2.1145 2.2451
Kurtosis
6.9871 7.5386
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 8.9333
Max. Differenz: 0.0305 (187.2150%)
Anz. der ZZ in Residuenklassen = 99 (0.9900%)


GIG(lambda,beta) lambda = 0.5000 beta = 8.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 0
Anz. Stuetzstellen = 60 Anz. erzeugter ZZ = 10000 Alpha = 1.001859
1.2165
Theor. Emp.
Mittelwert
1.1250 1.1322
Std. Abweichung
0.3953 0.4078
Schiefe
1.0124 1.0438
Kurtosis
1.6798 1.5822
Untere Schranke = 1.0e-10 Klassenanzahl: 40 Obere Schranke = 2.7061
Max. Differenz: 0.1245 (111.3982%)
Anz. der ZZ in Residuenklassen = 27 (0.2700%)


Burr V(r,k) r = 1.0000 k = 1.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 59
Anz. Stuetzstellen = 65 Anz. erzeugter ZZ = 10000 Alpha = 0.996728
0.5882
Theor. Emp.
Q10
-1.1437 -1.1437
Q25
-0.8324 -0.8280
Q50
0.0 -0.0238
Q75
0.8324 0.8293
Q90
1.1437 1.1481
Untere Schranke = -1.5708 Klassenanzahl: 40 Obere Schranke = 1.5708
Max. Differenz: 0.0631 (112.0250%)
Anz. der ZZ in Residuenklassen = 0 (0.0000%)


Burr V(r,k) r = 1.0000 k = 2.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 80
Anz. Stuetzstellen = 71 Anz. erzeugter ZZ = 10000 Alpha = 0.994360
0.8874
Theor. Emp.
Q10
-0.9840 -1.0008
Q25
-0.3852 -0.4240
Q50
0.6061 0.5811
Q75
1.0617 1.0562
Q90
1.2377 1.2401
Untere Schranke = -1.5708 Klassenanzahl: 40 Obere Schranke = 1.5708
Max. Differenz: 0.0656 (92.4081%)
Anz. der ZZ in Residuenklassen = 0 (0.0000%)


Burr V(r,k) r = 2.0000 k = 1.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 37
Anz. Stuetzstellen = 59 Anz. erzeugter ZZ = 10000 Alpha = 0.997878
0.9766
Theor. Emp.
Q10
-0.6569 -0.6457
Q25
0.0 0.0089
Q50
0.7224 0.7270
Q75
1.0789 1.0897
Q90
1.2405 1.2485
Untere Schranke = -1.5708 Klassenanzahl: 40 Obere Schranke = 1.5708
Max. Differenz: 0.0815 (88.3172%)
Anz. der ZZ in Residuenklassen = 0 (0.0000%)


Burr V(r,k) r = 2.0000 k = 2.0000
Stichprobengr. = 10000 Startwert = 1 Verwerfungen im MH-Schritt: 3
Anz. Stuetzstellen = 71 Anz. erzeugter ZZ = 10000 Alpha = 1.001934
1.5266
Theor. Emp.
Q10
-0.0779 -0.0668
Q25
0.6061 0.6244
Q50
1.0050 1.0100
Q75
1.1983 1.2027
Q90
1.3006 1.3053
Untere Schranke = -1.5708 Klassenanzahl: 40 Obere Schranke = 1.5708
Max. Differenz: 0.0529 (96.2359%)
Anz. der ZZ in Residuenklassen = 0 (0.0000%)

Kapitel 5

Konvergenzanalyse

In den vorangegangenen Kapiteln wurde gezeigt, daß mit Hilfe des Metropolis-Hastings-Algorithmus eine Markovkette, deren stationäre Dichtefunktion eine beliebig vorgegebene Funktion $\pi(x) = \pi(x_1, \ldots, x_k)$ ist, erzeugt werden kann. Mit Hilfe von empirischen Erwartungswerten können dann bestimmte Charakteristiken von $\pi(\cdot)$ geschätzt werden. Damit sind die theoretischen Hilfsmittel zur Anwendung von MCMC-Methoden gegeben.

Es soll nun das für die Praxis wichtige Thema der Konvergenzanalyse behandelt werden. In Kapitel 2 wurde festgestellt, daß die n-te Potenz der Übergangswahrscheinlichkeiten der Markovkette $\{X_t\}$ unter bestimmten Voraussetzungen gegen die stationäre Dichtefunktion unabhängig vom Startwert X_0 konvergiert. In der praktischen Anwendung von MCMC Methoden ist es oft schwierig zu entscheiden, ab welcher Iteration ein stationäres Verhalten der Markovkette zu beobachten ist. Zu Beginn der Simulation wird im allgemeinen der Einfluß des Startwertes X_0 dann groß sein, wenn X_0 nicht aus der stationären Verteilung erzeugt wurde. Die Markovkette befindet sich dann eventuell in einem Teil des Definitionsbereiches von $\pi(\cdot)$ mit niedriger Wahrscheinlichkeitsmasse. Die Zustände $X_0, X_1, \ldots, X_n$, die benötigt werden, bis die Markovkette einen stationären Zustand erreicht hat, werden zur Berechnung des empirischen Erwartungswertes nicht berücksichtigt:

$$\bar{f}_N = \frac{1}{N-n} \sum_{i=n+1}^{N} f(X_i) \tag{5.1}$$

Die Konvergenzanalyse hat nun folgende Aufgaben:

1. Ermittlung der Anzahl der ersten Iterationen der Simulation, die zum Erreichen der Stationarität erforderlich sind. Sie werden nicht zur Berechnung von $\bar{f}_N$ herangezogen.

2. Feststellen, wie groß N gewählt werden soll, damit der Schätzer $\bar{f}_N$ einer bestimmten Genauigkeit genügt.

Zwar ist die stationäre Verteilung theoretisch unabhängig von X_0, aber die Wahl des Startwertes hat im allgemeinen einen großen Einfluß auf die Schnelligkeit der Konvergenz. Probleme können vor allem dann auftreten, wenn Kenngrößen nur von einzelnen

Komponenten des Zufallsvektors X_t betrachtet werden und die Dimension von X_t groß ist. Ist die zu untersuchende Dichtefunktion $\pi(\cdot)$ unimodal, so wird eine gute Wahl von X_0 in der Nähe des Modus liegen. Oft ist es jedoch aufwendig, den Modus oder auch nur einen Bereich des Definitionsbereiches von $\pi(\cdot)$, der eine hohe Wahrscheinlichkeitsmasse aufweist, zu bestimmen. In diesem Fall sowie bei multimodalen Dichtefunktionen ist es unerläßlich, mehrere Simulationsläufe mit unterschiedlichen Startwerten auszuführen. Die Diskussion, ob zur Feststellung der Konvergenz ein langer oder mehrere kurze Simulationsläufe betrachtet werden sollen, ist noch nicht beendet (Gelman und Rubin, 1992).

Zur Entscheidung, wann die Markovkette einen stationären Zustand erreicht hat, stehen verschiedene Verfahren zur Verfügung: Man unterscheidet *quantitative Methoden*, mit deren Hilfe charakteristische Größen bezüglich der Konvergenz angegeben werden können. *Qualitative Verfahren* treffen nur die Aussage, ob stationäres Verhalten eingetreten ist oder nicht. Graphische Analysen können zu Beginn der Untersuchung einen Überblick über die Simulationsläufe bieten. Sie sind insbesondere nützlich zur Ermittlung des Index der Markovkette, ab der Stationarität zu beobachten ist.

Die quantitativen Methoden zur Konvergenzanalyse ermitteln statistische Kenngrößen anhand eines einzelnen Simulationslaufes, die Rückschlüsse auf die Stationarität zulassen. Sie versuchen, einen Bias in den Schätzgrößen festzustellen, der nahelegt, daß die zugrundeliegende Markovkette noch nicht die Stationarität erreicht hat. Nicht alle Verfahren schätzen auch die Korrelation innerhalb der Markovkette, um festzustellen, wieviele Iterationen durchgeführt werden müssen, damit die Schätzgrößen eine bestimmte Genauigkeit erreichen.

Für manche Methoden müssen zur Konvergenzanalyse mehrere Simulationen ausgeführt werden. Alle Methoden haben jedoch gemeinsam, daß sie keine absolute Sicherheit bezüglich der Stationarität der Markovkette geben können. Da die einzelnen Verfahren sehr unterschiedliche Zugänge zur Konvergenzanalyse wählen, wird empfohlen, nicht eine einzelne Methode zu verwenden, sondern mit Hilfe von unterschiedlichen Verfahren die Markovkette auf Stationarität zu untersuchen.

In dem Programmpaket `CODA` (Best *et al.*, 1995) wurden mehrere Verfahren implementiert, mit denen die Konvergenz von Markovketten untersucht werden können. Ursprünglich zur Diagnose des Outputs der Software `BUGS` (Spiegelhalter *et al.*, 1996), in der Gibbs-Erzeugungsmethoden implementiert wurden, entwickelt, können nun mit Hilfe von `CODA` auch beliebige Markovketten analysiert werden.

Ein eingehender Vergleich der gängigsten Methoden zur Konvergenzanalyse ist in Cowles und Carlin (1996) zu finden. Im folgenden werden die Methode, die für `CODA` verwendet wurden, vorgestellt. Dabei sei $\{X_t, t = 1, \ldots, N\}$ die bei einem Simulationslauf erzeugte Markovkette mit $X_t = (X_{t.1}, \ldots, X_{t.k})$ und $X_{t.i} \in \mathbb{R}$, $i = 1, \ldots, k$.

5.1 Konvergenzanalyse nach Gelman und Rubin

Gelman und Rubin (1992) verwenden zur Untersuchung der Konvergenzanalyse mehrere (mindestens zwei) unabhängige Simulationsläufe mit unterschiedlichen Startwerten. In der Praxis hat sich gezeigt, daß für Dichten mit einem dominierenden Modus ungefähr zehn Simulationsläufe zielführend sind; für multimodale Dichten sind entsprechend mehr Läufe anzusetzen. Ein Vergleich der Varianzen innerhalb der Markovketten und zwischen den einzelnen Simulationsläufen, basierend auf der klassischen Varianzanalyse, soll Rückschlüsse auf die Stationarität der Kette geben. Der Varianzvergleich dient zur Schätzung des Faktors, mit dem die Skalierungsparameter der Randverteilungen der Dichtefunktion $\pi(x_1, \ldots, x_k)$ reduziert werden können, wenn die Simulation unendlich lang fortgeführt würde.

Die Methode erzielt gute Resultate, wenn die Komponenten von X, nämlich die Variablen $X_{.i}$, annähernd normalverteilt sind. Deshalb werden rein positive $X_{.i}$ logarithmiert, und Komponenten, die aus dem Intervall (0,1) stammen, vor der Analyse mit der logit-Funktion, die durch

$$\operatorname{logit}(p) = \log \frac{p}{1-p} \quad p \in (0,1)$$

definiert ist, transformiert.

Für die unterschiedlichen Simulationsläufe sollen Startwerte X_0 verwendet werden, die von einer Dichte stammen, deren Zentren ungefähr mit denen der stationären Dichtefunktion $\pi(\cdot)$ übereinstimmen, deren Streuung jedoch größer als die von $\pi(\cdot)$ sein soll. Dadurch werden manche, jedoch nicht alle Startwerte im Bereich der größten Wahrscheinlichkeitsmasse der stationären Verteilung liegen. Zur Konstruktion der Dichtefunktion für die Startwerte wird $\pi(\cdot)$ approximiert. Man geht dabei wie folgt vor: Mit Hilfe einer numerischen Methode werden die Modi von $\pi(\cdot)$ ermittelt, und für jeden Modus wird die Matrix der zweiten Ableitung nach allen Komponenten $X_{.1}, \ldots, X_{.k}$ geschätzt. Die Dichte $\pi(\cdot)$ kann nun durch eine Normalverteilung, deren Mittelwerte den approximierten Modi entsprechen und deren Kovarianzmatrix mit Hilfe der zweiten Ableitungen ermittelt wurden, genähert werden. Man erzeugt Zufallsvektoren aus dieser Normalverteilung und dividiert die Komponenten durch χ^2_μ-verteilte Zufallsvariable. Die resultierenden t-verteilten Zufallsvektoren dienen als Startvektoren X_0 für die Simulationsläufe. Die Freiheitsgrade μ bestimmen die Streuung der Startwerte und sind je nach Gestalt von $\pi(\cdot)$ zu wählen.

Ausgehend von m unabhängigen Startwerten werden mit Hilfe einer beliebigen MCMC Methode in $N = 2n$ Iterationen parallele Markovketten erzeugt. Man verwendet die letzten $X_{n+1.i}, \ldots, X_{N.i}$ zur Approximation der Verteilung der skalaren Zufallsgröße $X_{.i}$ für $i = 1, \ldots, k$ durch eine t-Verteilung, deren Skalierungsparameter sowohl die Varianzen innerhalb als auch zwischen den einzelnen Markovketten berücksichtigt. Die Konvergenz wird mit Hilfe des Faktors, mit dem der Skalierungsparameter durch einen unendlich langen Simulationslauf reduziert werden könnte gemessen. Die Schätzung dieses Faktors erhält man durch

$$\sqrt{\widehat{R}} = \sqrt{\left(\frac{n-1}{n} + \frac{m+1}{mn}\frac{B}{W}\right)\frac{\nu}{\nu-2}},$$

wobei B die Varianz zwischen den Mittelwerten der m parallelen Markovketten, W der Mittelwert der Varianzen der m Simulationsläufe und ν die Freiheitsgrade der t-Verteilung, die die i-te Randverteilung $\pi(x_1, \ldots, x_k)$ approximieren soll, sind.

Weist die Markovkette eine schlechte Mischung auf, so ist B im allgemeinen für kleine N größer als W, denn die Startwerte X_0 sind im Vergleich zur stationären Dichte $\pi(\cdot)$ weiter gestreut. Konvergenz im Sinne von Gelman und Rubin ist dann gegeben, wenn $\sqrt{\widehat{R}}$ für alle $X_{.i}$, $i = 1, \ldots, k$ nahe eins ist. Das bedeutet, daß die gepoolte Varianz zwischen den einzelnen Ketten die Varianz der Mittelwerte dominiert und somit die m Markovketten nicht mehr unter dem Einfluß der Startwerte stehen.

Daß eine Anwendung der Methode von Gelman und Rubin nicht unproblematisch ist, sollen folgende Überlegungen zeigen. Der Benutzer muß zu Beginn eine approximierende Funktion für die Dichte $\pi(\cdot)$ mit größerer Streuung als $\pi(\cdot)$ finden. Eine Aufgabe, die angesichts der Tatsache, daß man in der Praxis nur über wenig Information von $\pi(\cdot)$ verfügt, im allgemeinen schwierig ist. Die Konvergenzanalyse beruht auf Näherungen mit Hilfe von t-Verteilungen. MCMC Methoden werden aber häufig dann eingesetzt, wenn die zu untersuchende Dichtefunktion $\pi(\cdot)$ eine kompliziertere Gestalt aufweist. Außerdem ist das Verfahren nicht effizient, da nur die Hälfte aller Iterationen verwendet werden.

Gelman und Rubin betrachten zur Bestimmung der Konvergenz nur einzelne Komponenten $X_{.i}$. Oft ist aber das Verhalten der Markovkette unter Berücksichtigung aller Komponenten von Interesse. Die besprochene Methode kann in diesem Fall nicht direkt verwendet werden. Es besteht jedoch die Möglichkeit, anstatt der Markovkette $\{X_t, t = 1, \ldots, N\}$ deren Transformierte $\{-2 \log \pi(X_{t.1}, \ldots, X_{t.k}), t = 1, \ldots, N\}$ zu untersuchen. Die mit Hilfe der Aposteriori-Verteilung $\pi(\cdot)$ transformierte Kette ist univariat, und auf diese kann das Verfahren von Gelman und Rubin angewendet werden.

5.2 Konvergenzanalyse nach Raftery und Lewis

Die Methode von Raftery und Lewis (1992b) wird zur Untersuchung einer einzelnen Markovkette verwendet. Sie dient sowohl zur Analyse der Konvergenz als auch zur Beschränkung der Varianz für Quantilschätzer von Verteilungen einzelner Zufallsvariablen $X_{.i}$.

Nach Durchführung des Simulationslaufes werden

(i) das Niveau q des zu untersuchenden Quantils x_q (z. B. $q = 0.25$),

(ii) die Genauigkeit r der Schätzung (die halbe Länge des Konfidenzintervalls, z. B. $r = \pm 0.005$) und

(iii) die Wahrscheinlichkeit s, diese Genauigkeit zu erreichen (das Konfidenzniveau, z. B. $s = 0.95$)

festgelegt. Die Methode von Raftery und Lewis liefert dann `Nmin`, die minimale Anzahl von Iterationen, die nötig ist, um das Quantil mit der festgelegten Genauigkeit zu schätzen, falls die Iterationen unabhängig sind. Dieser theoretische Wert basiert auf der binomialen Varianz und gibt eine untere Schranke für die Länge des Simulationslaufes an. `Nmin` wird größer, wenn die Genauigkeit der Schätzung und das Konfidenzniveau gesteigert werden. Dadurch, daß die Formel für `Nmin` auf der binomialen Varianz beruht, erhält man das auf den ersten Blick überraschende Resultat, daß bei gleicher Genauigkeit für die Schätzung der Quantile nahe dem Median mehr Iterationen als für extreme Quantile notwendig sind.

Ist die Anzahl der tatsächlich ausgeführten Iterationen größer als `Nmin`, dann liefert die Methode von Raftery und Lewis die Kenngrößen `N` (die Gesamtanzahl von Iterationen, die für jede Variable $X_{.i}$ ausgeführt werden soll), `M` (die Anzahl von Iterationen, die zur Erreichung der Stationarität notwendig ist) und `k` (wenn nur jede `k`-te Iteration zur Untersuchung des angegebenen Quantils verwendet werden soll). Große Werte von `M` lassen auf langsame Konvergenz der Markovkette schließen, während ein `N`, das viel größer als `Nmin` ist, oder ein `k` größer als eins, auf starke Korrelationen zwischen den Iterationen hindeuten. So empfehlen Raftery und Lewis (1992a) für den Fall, daß `I` $=$ `N/Nmin` > 5, die Variablen $X_{.i}$ neu zu parametrisieren, da mangelnde Konvergenz vermutet werden muß.

5.3 Konvergenzanalyse nach Geweke

Geweke (1992) schlägt eine Diagnosemethode vor, die ursprünglich zur Untersuchung von Zeitreihen diente. Gewekes Verfahren kann nur angewendet werden, wenn der Mittelwert einer Funktion $f(\cdot)$ in einer oder mehrerer Komponenten $X_{.i}$ betrachtet werden soll. Wenn $f(X_t)$, $t = 1, 2 \ldots$, nach jeder Iteration ausgewertet wird, läßt sich diese Folge von reellen Zahlen als Zeitreihe auffassen.

Die Methode von Geweke beruht auf der Annahme, daß aus den Eigenschaften des MCMC Prozesses und der Funktion $f(\cdot)$ die Existenz einer Spektraldichte $S_f(\omega)$ folgt, die für die Frequenz $\omega = 0$ stetig ist. Wenn diese Annahme zutrifft, dann hat der Schätzer von $E[f(X)]$,

$$\bar{f}_n = \frac{1}{n} \sum_{t=1}^{n} f(X_{t.i}),$$

die asymptotische Varianz $S_f(0)/n$.

Es wird ein einzelner Simulationslauf der Länge N untersucht, indem aus der Menge aller Iterationen $X_1, \ldots, X_N$ zwei Teilketten der Länge N_A und N_B entnommen und miteinander verglichen werden. In der Literatur werden diese beiden Teilketten "Fenster" genannt. Man wählt beispielsweise die ersten 10% ($N_A = N/10$) und die letzten 50% ($N_B = N/2$) aller erzeugten X_t. Man bildet nun die Schätzer der Erwartungswerte $\bar{f}_N^A$ und $\bar{f}_N^B$ aus den Iterationen der beiden Fenster. Wenn N_A/N und N_B/N für $N \to \infty$ konstant bleiben und $N_A + N_B < N$ gilt, so ist Gewekes Teststatitistik $Z = \bar{f}_N^A - \bar{f}_N^B$ asymptotisch Standard-normalverteilt.

Geweke schlägt vor, seine Methode zunächst zur Bestimmung der ersten Iterationen, für die die Markovkette noch nicht stationär ist, zu verwenden. Nach dem Ausscheiden dieser Iterationen kann der asymptotische Standardfehler des Mittelwertes $\sqrt{S_f(0)/n}$ geschätzt werden. Die Methode reagiert allerdings sehr sensitiv auf die Wahl der Fenster, weshalb die Anzahl der ersten Iterationen nicht zu klein gewählt werden sollte.

Analog zum Verfahren von Gelman und Rubin (s. Abschnitt 5.1) kann jenes von Geweke nur auf einzelne Komponenten $X_{.i}$ angewendet werden. Zur Untersuchung der Konvergenz der Markovkette unter Berücksichtigung aller Komponenten sollte die Kette ebenfalls durch $-2 \log \pi(X_{t.1}, \ldots, X_{t.i})$ transformiert werden. Gewekes Verfahren ist direkt auf die transformierte Kette anwendbar. Es läßt dann Rückschlüsse auf das stationäre Verhalten aller Komponenten $X_{.1}, \ldots, X_{.k}$ zu.

5.4 Konvergenzanalyse nach Heidelberger und Welch

Heidelberger und Welch (1983) entwickelten eine Methode, mit der entschieden werden kann, ob der Startwert einer Kette von diskreten Zuständen transient ist, was bedeuten würde, daß die ersten Iterationen der Kette noch kein stationäres Verhalten zeigen. Wie sich gezeigt hat, eignet sich das Verfahren auch zur Analyse der Konvergenz für den Output eines Gibbs-Generators oder einer beliebigen MCMC Methode. Es umfaßt nicht nur eine Stationaritätsuntersuchung, sondern überprüft auch, ab welcher Iteration der Mittelwert einer Komponente $X_{.i}$ mit einer vorher festgelegten Genauigkeit geschätzt werden soll. Für diese Analyse genügt die Ausführung eines einzigen Simulationslaufes.

Das Verfahren basiert auf der Theorie der Brown'schen Brücke und verwendet die Cramér-von Mises Statistik zum Test der Nullhypothese, daß die vorliegenden $X_{1.i}, \ldots, X_{N.i}$ aus einem stationären Prozeß stammen, der eine ϕ-Mischung aufweist. Für Markovketten ist das äquivalent zu gleichförmiger Ergodizität (s. Definition 2.2.6). Man kann jedoch zeigen, daß diese Überlegungen auch für nur geometrisch ergodische Ketten gelten. Diese Eigenschaft erfüllen viele konvergente Gibbs und Metropolis-Hastings-Algorithmen auf allgemeinen Zustandsräumen.

Es soll nun die Kette $X_{1.i}, \ldots, X_{N.i}$ mit der Spektraldichte $S_i(\omega)$ für ein beliebiges $i \in \{1, \ldots, k\}$ betrachtet werden. $\lfloor a \rfloor$ bezeichnet die größte ganze Zahl kleiner gleich a und es seien weiters

$$dT_0 = 0,$$

$$T_r = \sum_{t=1}^{r} X_{t.i}, \qquad r \geq 1,$$

$$\bar{X}_{.i} = \frac{1}{N} \sum_{t=1}^{N} X_{t.i},$$

$$B_N(x) = \frac{T_{\lfloor Nx \rfloor} - \lfloor Nx \rfloor \bar{X}_{.i}}{\sqrt{N S_i(0)}}, \qquad 0 \leq x \leq 1.$$

Unter der Nullhypothese ist $B_N = \{B_N(x), 0 \leq x \leq 1\}$ für große N annähernd wie eine Brown'sche Brücke verteilt. Deshalb kann die Cramér-von Mises Statistik

$$\int_0^1 B_N^2(x) dx$$

zum Test der Hypothese verwendet werden. Da $S_i(\omega)$ an der Stelle $\omega = 0$ nicht bekannt ist, muß $S_i(0)$ aus den Daten geschätzt und in $B_N(x)$ verwendet werden.

Heidelberger und Welch verwenden diesen Test zum Aufspüren und Ausscheiden eines transienten Startwertes, zur Bestimmung eines Konfidenzintervalls für $E_\pi[X_{.i}]$ und zur Kontrolle, ob die Anzahl der Iterationen der Kette genügend groß ist. Es wird nun der ursprünglich von Heidelberger und Welch vorgeschlagene Algorithmus vorgestellt. Die im Softwarepaket `CODA` verwendete Methode ist eine vereinfachte Version.

Der Benutzer legt mit $Nmax$ die maximale Anzahl von möglichen Iterationen und mit ϵ die gewünschte Genauigkeit der halben Länge des genannten Konfidenzintervalls fest.

Es werden nun $N_1 = Nmax/10$ Iterationen der gewählten MCMC-Methode ausgeführt. Wenn der Startwert dieses Simulationslaufes transient ist, so ist ein Schätzer für $S_i(0)$ basierend auf der vollständigen Stichprobe $X_{1.i}, \ldots, X_{N_1.i}$ im allgemeinen zu groß. Das würde die Teststatistik von Cramér-von Mises verkleinern und somit die Macht des Tests reduzieren. Deshalb schlagen Heidelberger und Welch vor, für die Schätzung von $S_i(0)$ nur die zweite Hälfte der Stichprobe zu verwenden. Wird die Nullhypothese verworfen, so werden die ersten 10% der $X_{t.i}$ ausgeschieden. Der Prozeß wird solange fortgesetzt, bis für einen Teil des Outputs des ersten Simulationslaufes, der länger als $N_1/2$ ist, der Test nicht verwirft oder 50% der Stichprobe ausgeschieden wurden und der Test weiterhin nicht angenommen wird.

Im ersten Fall wird $S_i(0)$ mit Hilfe aller nicht ausgeschiedenen $X_{t.i}$ neu geschätzt und der Standardfehler des Mittelwertes von $X_{.i}$ durch $\sqrt{\widehat{S}_i(0)/N_p}$ approximiert, wobei N_p die Anzahl der nicht ausgeschiedenen $X_{.i}$ ist. Das 95%-Konfidenzintervall des Mittelwertes erhält man durch

$$\frac{1}{N_p} \sum_{t=N_1-N_p+1}^{N_1} X_{t.i} \quad \pm \quad 1.96\sqrt{\widehat{S}_i(0)/N_p}.$$

Die Schätzung des Mittelwertes der $X_{.i}$ gilt dann als genau genug, wenn die halbe Länge des Konfidenzintervalls kleiner als ϵ mal dem empirischen Mittelwert der nicht ausgeschiedenen Iterationen ist. Heidelberger und Welch stufen dann den Prozeß als stationär ein und die Konvergenzanalyse schließt mit der Ausgabe der Ergebnisse.

Im Fall, daß die Hypothese der Stationarität mit Hilfe der Cramér-von Mises Statistik nicht angenommen oder das Konfidenzintervall nicht genau genug angegeben werden konnte, wird ein neuer Simulationslauf der Länge $N_2 = 1.5N_1$ ausgeführt. Es wird nun die gleiche Analyse – wie bereits beschrieben – für den nun längeren Simulationslauf durchgeführt. Sollte auch diese Kette nicht die gewünschten Eigenschaften aufweisen, dann werden sukzessive längere Ketten der Länge $N_{j+1} = \min\{1.5N_j, Nmax\}$ untersucht.

Die Betrachtung von sukzessive längeren Simulationsläufen garantiert, daß eine eventuelle Stationarität zum frühest möglichen Zeitpunkt entdeckt wird. Das Verfahren wurde für `CODA` in der Weise vereinfacht, daß zu Beginn $N_1 = Nmax$ gesetzt wird und somit nicht schrittweise längere Ketten betrachtet werden, sondern in einem Schritt der ganze vorliegende Simulationslauf auf Stationarität untersucht und die Genauigkeit der Mittelwertschätzung überprüft wird.

Kapitel 6

Optimierung von MCMC-Methoden

Nicht jede MCMC-Methode eignet sich in gleicher Weise zur Simulation einer gegebenen stationären Verteilung. Wie in Abschnitt 1.4 erwähnt, kann die Mischung der resultierenden Markovkette $\{X_t, t = 0, \dots, N\}$ von der Wahl des Algorithmus abhängen. Durch Reparametrisierung der Komponenten $X_{.1}, \dots, X_{.k}$ können noch vor Beginn der Simulation Verbesserungen im Endresultat erzielt werden. Bestimmte MCMC-Algorithmen sind auf Probleme zugeschnitten, die sich aufgrund einer multimodalen, stationären Dichtefunktion ergeben. Sie bieten im allgemeinen eine höhere Wahrscheinlichkeit, daß die erzeugte Markovkette den gesamten Definitionsbereich dieser stationären Dichtefunktion erreicht.

In diesem Kapitel werden Vorgehensweisen vorgestellt, die für bestimmte Formen von stationären Verteilungen zu besseren Ergebnissen führen können. Dies muß aber nicht immer der Fall sein. Sehr häufig ist die Erfahrung des Benutzers der MCMC-Methoden gefordert, um für eine bestimmte Problemstellung jenes Verfahren oder eine Kombination von Methoden zu bestimmen, die eine Markovkette erzeugen, aus der sich hinreichend genaue Ergebnisse ableiten lassen.

6.1 Optimierung durch Reparametrisierung

In diesem Abschnitt wird gezeigt, wie sowohl der Gibbs-Generator als auch der Metropolis-Hastings-Algorithmus bei schlechter Parametrisierung reagieren (Gilks *et al.*, 1995b, Kapitel 6). Man kann beispielsweise eine zweidimensionale stationäre Dichtefunktion $\pi(x_{.1}, x_{.2})$ betrachten, deren Konturen als Ellipsen, die entlang der Geraden $x_{.1} = x_{.2}$ ausgerichtet sind, in Abbildung 6.1 dargestellt sind. Die zur Dichte $\pi(x_{.1}, x_{.2})$ gehörigen Zufallsvariablen $X_{.1}$ und $X_{.2}$ weisen eine starke Abhängigkeit auf.

Eine Stichprobe von der stationären Dichtefunktion $\pi(x_{.1}, x_{.2})$ kann mit dem Gibbs-Generator (s. Abschnitt 1.3.4), der in einem Schritt eine Komponente von $x = (x_{.1}, x_{.2})$ erneuert, erzeugt werden. Für die vorliegende Dichtefunktion besteht eine Iteration des Gibbs-Generators aus zwei Schritten, nämlich der Erneuerung von $x_{.1}$ und $x_{.2}$, die jeweils eine Bewegung parallel zu einer Koordinatenachse darstellt.

In Abbildung 6.1 ist ein möglicher Pfad des Gibbs-Generators ausgehend vom Startpunkt x_0 dargestellt. Eine Iteration, nämlich der Übergang von x_t zu x_{t+1}, besteht aus zwei

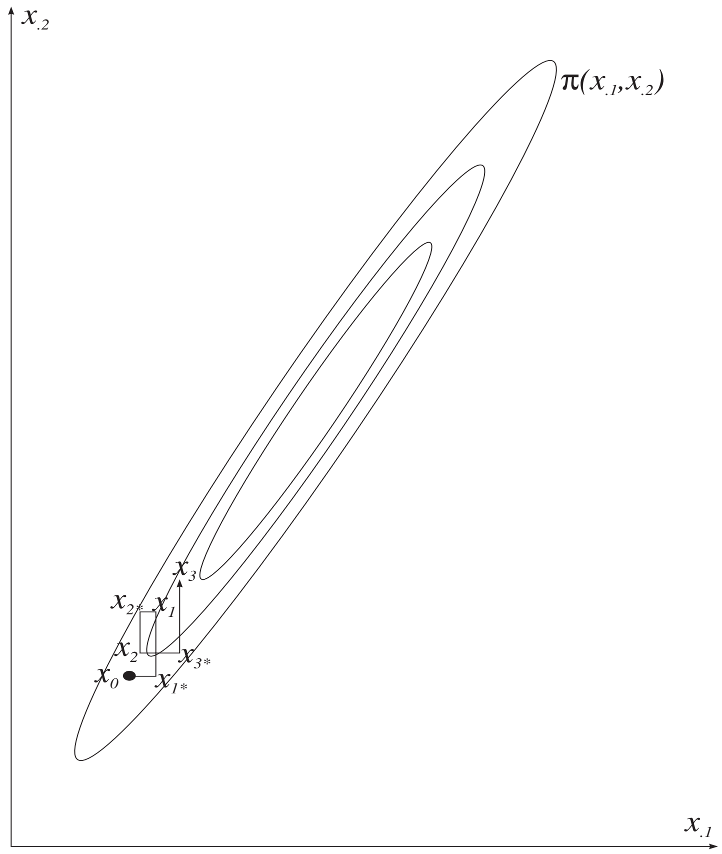


Abbildung 6.1: *Iterationen des Gibbs-Generators für eine stationäre Dichtefunktion* $\pi(x_{.1}, x_{.2})$.

Schritten. Der Punkt, der nach der Erneuerung von $x_{t.1}$ erreicht wird, ist mit x_{t^*} für $t = 0, 1, \ldots,$ gekennzeichnet.

Ein Problem der Mischung ergibt sich aus der speziellen Form der Dichtefunktion $\pi(x_{.1}, x_{.2})$. Weil $\pi(\cdot)$ stark um die Gerade $x_{.1} = x_{.2}$ konzentriert ist und der Gibbs-Generator nur eine abwechselnde Bewegung in Richtung von $x_{.1}$ und $x_{.2}$ zuläßt, können die Sprünge von x_t zu x_{t+1^*} bzw. x_{t+1^*} zu x_{t+1} nur klein sein. Es werden viele Iterationen benötigt, um den ganzen Definitionsbereich von $\pi(\cdot)$ abzudecken, und die Mischung der erzeugten Markovkette ist schlecht.

Der Grund für die kleinen Sprünge liegt in der Funktionsweise des Gibbs-Generators, der neue Punkte aus den vollständig bedingten Verteilungen erzeugt. Beispielsweise stammt x_{t+1^*} aus der Dichte $\pi(\cdot|x_{t.2})$ für $t = 0, 1, \ldots$.

Diese bedingten Dichten sind Schnitte durch die gemeinsame Dichte $\pi(\cdot)$ parallel zu den Koordinatenachsen. Aufgrund der Gestalt von $\pi(\cdot)$ in Abbildung 6.1 ist der Bereich der bedingten Dichten sehr klein, die erzeugten Punkte liegen deshalb eng nebeneinander.

Wendet man für die Erzeugung einer Markovkette mit stationärer Dichtefunktion einen Metropolis-Hastings-Algorithmus an, so kann bei starker Abhängigkeit zwischen den Komponenten von x ebenfalls eine schlechte Mischung auftreten, wenn die Kandidatendichte $q(\cdot, \cdot)$ nicht in Abstimmung zu $\pi(\cdot)$ gewählt wird. Wenn $q(\cdot, \cdot)$ beispielsweise eine Gleichverteilung auf einer Kreisscheibe ist, so erhält man die in Abbildung 6.2 dargestellte Situation. Die Ellipsen stellen wieder die Konturen von $\pi(\cdot)$ dar; in der Kreisscheibe sind die Kandidaten für x_{t+1} in der t-ten Iteration gleichverteilt.

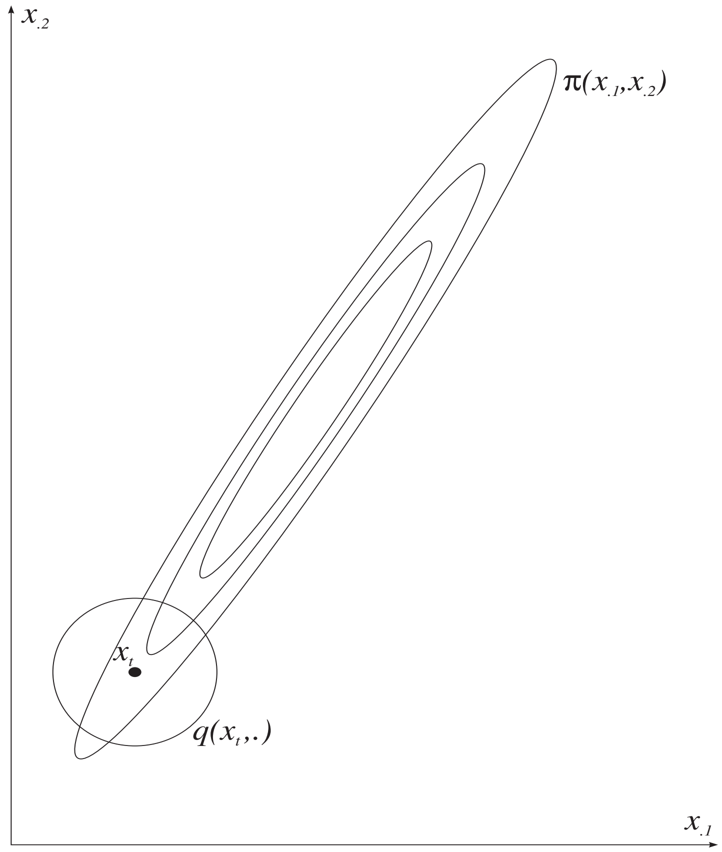


Abbildung 6.2: *Metropolis-Hastings-Algorithmus mit kreisförmiger Gleichverteilung als Kandidatendichte* $q(\cdot,\cdot)$.

Man erkennt in Abbildung 6.2, daß die Gleichverteilung $q(\cdot,\cdot)$ den Bereich von $\pi(\cdot)$ mit großer Wahrscheinlichkeitsmasse nur im Zentrum um den Punkt x_t schneidet. Für Kandidatenpunkte x', die nicht in diesem Schnittbereich liegen, gilt $\pi(x') < \pi(x_t)$. Da man die Akzeptanzwahrscheinlichkeit von x' durch

$$\alpha(x_t, x') = \min\left\{1, \frac{\pi(x')}{\pi(x_t)}\right\}$$

erhält (s. Abschnitt 1.3), werden diese Kandidatenpunkte mit großer Wahrscheinlichkeit verworfen. Je kleiner die Schnittmenge von $q(X_t,\cdot)$ und $\pi(\cdot)$ ist, desto öfter können Verwerfungen vorkommen und umso schlechter ist die Mischung der Markovkette.

Ist eine Anpassung der Kandidatendichte an die stationäre Dichtefunktion $\pi(x_{.1}, x_{.2})$ nicht möglich, so kann $\pi(x_{.1}, x_{.2})$ bezüglich $x_{.1}$ und $x_{.2}$ so transformiert werden, daß die Konturen keine Ellipsen, sondern annähernd Kreise sind, d. h. die zu $\pi(x_{.1}, x_{.2})$ gehörigen Zufallsvariablen $X_{.1}$ und $X_{.2}$ sind annähernd unabhängig. Ist die Kandidatendichte $q(x_t,\cdot)$

wieder eine Gleichverteilung auf einer Kreisscheibe, so ist der Schnittbereich von $q(x_t, \cdot)$ und $\pi(\cdot)$ größer, wodurch eine bessere Akzeptanzwahrscheinlichkeit erreicht werden kann (s. Abbildung 6.3).

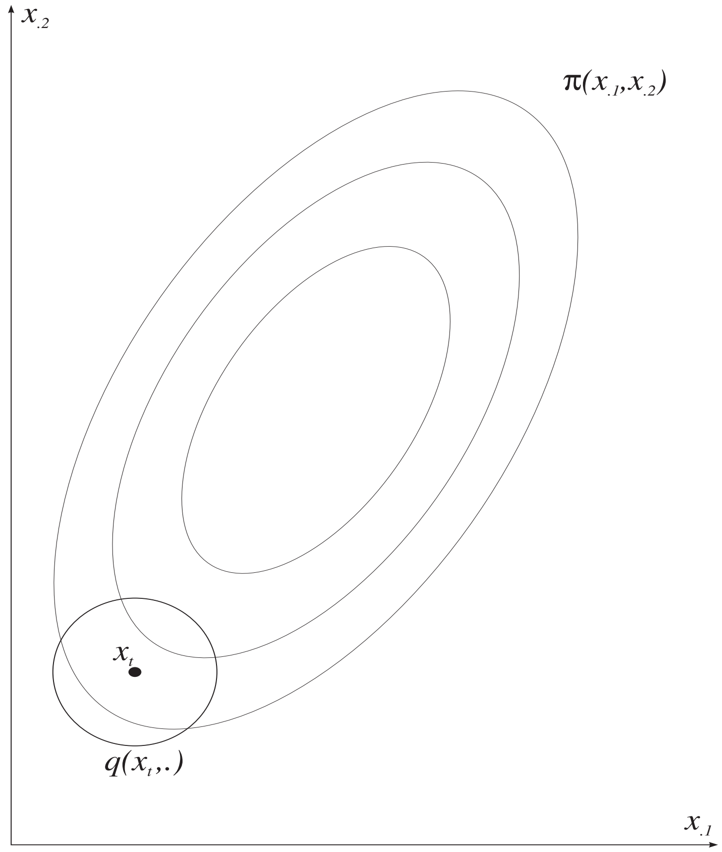


Abbildung 6.3: *Reparametrisierte stationäre Dichtefunktion* $\pi(x_{.1}, x_{.2})$.

Für den Gibbs-Generator lassen sich durch Reparametrisierung ebenfalls bessere Ergebnisse erzielen. Wie zuvor beschrieben, kann der Grund für kleine Sprünge des Gibbs-Generators in einem engen Bereich der vollständig bedingten Verteilungen, die von $\pi(\cdot)$ abgeleitet werden, liegen. Durch die Reparametrisierung kann die Konzentration der Wahrscheinlichkeitsmasse um die Gerade $x_{.1} = x_{.2}$ verringert werden, wodurch sich auch der Bereich der vollständig bedingten Verteilungen vergrößert.

Die Reparametrisierung kann aufgrund der Kenntnis bestimmter Eigenschaften der Daten, wenn beispielsweise Abhängigkeiten durch das zugrundeliegende Modell bedingt sind, vorgenommen werden. Ist die stationäre Dichtefunktion $\pi(\cdot)$ nicht zu kompliziert, kann sie auf Korrelationen einzelner Komponenten untersucht werden. In der Praxis lassen sich nicht immer alle Abhängigkeiten feststellen, weshalb es wünschenswert ist, über Algorithmen zu verfügen, für die keine Reparametrisierung notwendig ist. Im folgenden Abschnitt werden Verfahren vorgestellt, die die Bewegungsrichtung adaptiv in jeder Iteration bestimmen.

6.2 Adaptives Richtungssampling

Der Gibbs-Generator bewegt sich parallel zu den Koordinatenachsen, was, wie im vorangegangenen Abschnitt gezeigt wurde, zu einer schlechten Mischung der erzeugten Markovkette führen kann. Lineare Reparametrisierung, die bei Mischungsproblemen Verbesserungen bringen kann, entspricht einer Änderung der Bewegungsrichtung des Gibbs-Generators. Es ist deshalb naheliegend, Algorithmen zu entwickeln, bei denen die Bewegungsrichtung bei jeder Iteration neu bestimmt wird. Im adaptiven Richtungssampling wird diese Idee aufgegriffen und zu einem allgemeinen Algorithmus zusammengefaßt, von dem sich spezielle Methoden ableiten lassen, die für gewisse Problemstellungen nützlich sein können.

Zur Erzeugung einer Markovkette mit stationärer Dichtefunktion $\pi(x_{.1}, \ldots, x_{.k})$ wird beim adaptiven Richtungssampling in der t-ten Iteration eine Menge $S_t = \{x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(m)}\}$ von m erzeugten Punkten aus $\mathbb{R}^k$ gespeichert. Bei jeder Iteration wird aus der Menge S_t ein Punkt $x_t^{(c)}$ zufällig ausgewählt und erneuert. Der neue Punkt $x_{t+1}^{(c)}$ liegt auf einer Gerade, die durch den Punkt $x_t^{(c)}$ geht, und die die zufällige Richtung e_t hat, die von weiteren Punkten aus der Menge S_t abhängen kann:

$$x_{t+1}^{(c)} = x_t^{(c)} + r_t e_t, \quad \text{wobei} \quad r \in \mathbb{R}.$$

Jede Iteration des adaptiven Richtungssampling besteht somit aus:

> `Wähle` $x_t^{(c)}$ `zufällig aus` S_t
> `Wähle` e_t `aus` $\mathbb{R}^k$
> `Erzeuge` r_t `aus einer Dichte` $f(r)$
> `Setze` $x_{t+1}^{(c)} = x_t^{(c)} + r_t e_t$
> `Setze` $x_{t+1}^{(i)} = x_t^{(i)}$ `für` $i \neq c$

Wie die Dichte $f(r)$ aus dem obigen Algorithmus zu bestimmen ist, wird in den folgenden Abschnitten erklärt.

6.2.1 Der Snooker-Algorithmus

Der Snooker-Algorithmus (Gilks *et al.*, 1994) ist ein adaptives Richtungssamplingverfahren, bei dem die Richtung $e_t = x_t^{(i)} - x_t^{(c)}$ gesetzt wird, wobei $x_t^{(i)}$ ebenfalls ein zufällig gewählter Punkt aus der Menge S_t ist. Der in der t-ten Iteration erzeugte Punkt liegt somit auf der Geraden

$$l_t(r) = x_t^{(c)} + r(x_t^{(i)} - x_t^{(c)}).$$

Die Dichte, aus der r erzeugt werden muß, hat die Form

$$f(r) \propto \pi(x_t^{(c)} + re_t)|1 - r|^{k-1}.$$

Es ist dies die vollständig bedingte Verteilung entlang der Geraden $l_t(r)$, multipliziert mit dem Term $|1 - r|^{k-1}$, der abhängig von k, der Dimension des Zustandsraumes, ist. Der multiplikative Term ergibt sich aus der Tatsache, daß die Bewegungsrichtung e_t abhängig von $x_t^{(c)}$ ist (Gilks *et al.*, 1994).

In Abbildung 6.4 ist die Ausgangsposition des Snooker-Algorithmus für eine zweidimensionale stationäre Dichtefunktion $\pi(x_{.1}, x_{.2})$ vor der t-ten Iteration dargestellt. Die Menge S_t enthält drei Punkte.

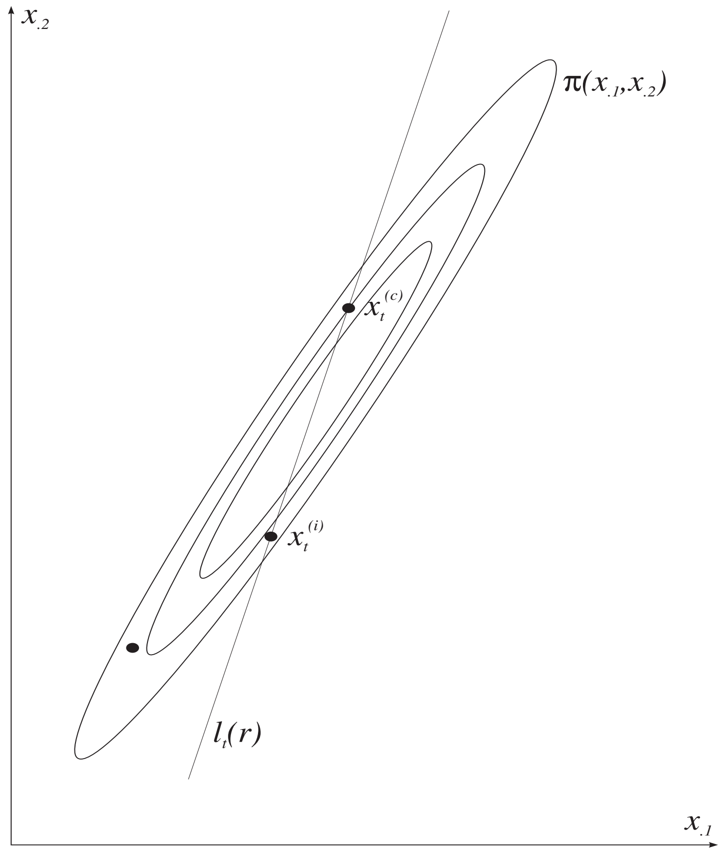


Abbildung 6.4: *Der Snooker-Algorithmus: Ausgangsposition vor der t-ten Iteration.*

Mit Fortschreiten der Iterationen wird es immer wahrscheinlicher, daß die Gerade l_t Bereiche mit hoher Wahrscheinlichkeitsmasse von $\pi(\cdot)$ passiert, weil sowohl $x_t^{(c)}$ als auch $x_t^{(i)}$ mit immer größerer Wahrscheinlichkeit in solchen Bereichen liegen.

Abbildung 6.5 zeigt die Situation nach der t-ten Iteration.

6.2.2 Der Hit-and-run-Algorithmus

Der Hit and run-Algorithmus (Schmeiser und Chen, 1991) ist der einfachste Spezialfall einer adaptiven Richtungssamplingmethode mit $m = 1$; die Menge S_t besitzt somit nur ein Element x_t.

Der die Bewegungsrichtung angebende Vektor $e_t \in \mathbb{R}^k$ wird zufällig bestimmt. e_t kann beispielsweise gleichverteilt im k-dimensionalen Raum sein. Dazu erzeugt man k stan-

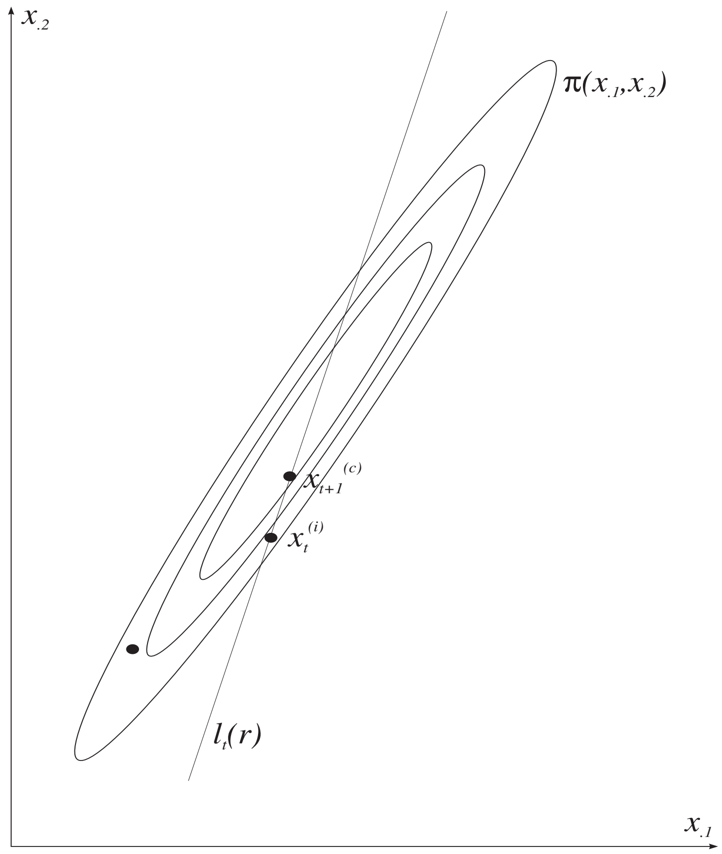


Abbildung 6.5: *Der Snooker-Algorithmus: In der t-ten Iteration wird der Punkt $x_{t+1}^{(c)}$ erzeugt.*

dardnormalverteilte Zufallsvariable $\{z_i; i = 1, \ldots, k\}$ und setzt für $i = 1, \ldots, k$

$$e_{t.i} = \frac{z_i}{\sqrt{\sum_j z_j^2}}.$$

Da der Richtungsvektor e_t unabhängig von $x_t^{(c)}$ ist, entspricht die Dichte $f(r)$, aus der der Skalar r erzeugt wird, der vollständig bedingten Verteilung:

$$f(r) \propto \pi(x_t + re_t).$$

6.3 Diskussion

In diesem Kapitel wurden nur einige Möglichkeiten zur Optimierung von MCMC-Methoden präsentiert. Spezielle Verteilungstypen fordern zur Simulation mit MCMC-Methoden weitere Strategien, um schnell konvergente Markovketten zu erhalten. So können die hier vorgestellten Optimierungen beispielsweise keine gute Simulation von multimodalen Verteilungen sicherstellen. Für eine richtige Wahl der verwendeten Simulationsmethode ist deshalb auch eine ungefähre Kenntnis der Gestalt der Verteilung erforderlich.

Die Anwendung von MCMC-Methoden kann nicht nach bestimmten, gleichbleibenden Schemata erfolgen, sondern verlangt vom Benutzer große Flexibilität und Erfahrung im Umgang mit bestehenden Verfahren. Die Forschung beschäftigt sich heute intensiv mit der Weiterentwicklung von MCMC-Methoden. Ob sich ihre Anwendung in Bayes'schen Modellen als Alternative zur herkömmlichen statischen Analyse außerhalb des akademischen Bereiches etablieren wird, ist aufgrund der weitreichenden mathematischen Kenntnisse, die eine Verwendung der Methoden voraussetzt, fraglich.

Literaturverzeichnis

Barndorff-Nielsen, O. (1977), *Exponentially decreasing distributions for the logarithm of particle size*, Proceedings of the Royal Society of London, A, **353**, 401–419.

Besag, J. (1974), *Spatial interaction and the statistical analysis of lattice systems (with discussion)*, J. R. Statist. Soc. B **36**, 192–236.

Best, N. G., Cowles, M. K. und Vines, K. (1995), `CODA`: *Convergence diagnosis and output analysis software for Gibbs sampling output, Version 0.30 (Manual)*, Unversity of Cambridge, MRC Biostatistics Unit.
Handbuch und Programmpaket `CODA` im Internet erhältlich unter der Adresse `http://www.mrc-bsu.cam.ac.uk`

Burr, I. W. (1942), *Cumulative frequency functions*, Ann. Math. Statist. **13**, 215–232.

Chib, S. und Greenberg, E. (1995), *Understanding the Metropolis-Hastings algorithm*, The Am. Statistician **49**, 327–335.

Cowles, M. K. und Carlin, B. P. (1996), *Markov chain Monte Carlo convergence diagnostics: A comparative review*, J. Amer. Statist. Assoc. **91**, 893–904.

Geman, S. und Geman, D. (1984), *Stochastic relaxation, Gibbs distributions and the Baysian restoration of images*, IEEE Trans. Pattn. Anal. Mach. Intel., **6**, 721–741.

Geweke, J. (1992), *Evaluating the accuracy of sampling-based approaches to calculating posterior moments*, Bayesian Statistics 4 (Hrgb. J. M. Bernardo, J. O. Berger, A. P. Dawid und A. F. M. Smith), Oxford: Oxford University Press.

Geyer, C. J. (1992), *Practical Markov chain Monte Carlo*, Statist. Sci. **7**, 473–511.

Gilks, W. R., Best, N. G. und Tan, K. K. C. (1995), *Adaptive rejection Metropolis sampling within Gibbs sampling*, Appl. Statist. **44**, 455–472.

Gilks, W. R. (1992), *Derivative-free adaptive rejection sampling for Gibbs sampling*, Bayesian Statistics 4 (Hrgb. J. M. Bernardo, J. O. Berger, A. P. Dawid und A. F. M. Smith), Oxford: Oxford University Press, 641–649.

Gilks, W. R. und Wild, P. (1992), *Adaptive rejection sampling for Gibbs sampling*, Appl. Statist. **41**, 337–348.

Gilks, W. R., Richardson, S. und Spiegelhalter, D. J. (Hg.) (1995), *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London.

Gilks, W. R., Roberts, G. O. und George, E. I. (1994), *Adaptive direction sampling*, The Statistician **43**, 179–189.

Gelman, A. und Rubin, D. B. (1992), *Inference from iterative simulation using multiple sequences*, Statistical Science **7**, 457–472.

Hastings, W. K. (1970), *Monte Carlo sampling methods using Markov chains and systems*, Biometrika **57**, 97–109.

Heidelberger, P. und Welch, P. (1983), *Simulation run length control in the presence of an initial transient*, Operations Research **31**, 1109–1144.

Lehner, K. (1989), *Erzeugung von Zufallszahlen für zwei exotische Verteilungen*, Diplomarbeit, 107 pp., Technische Universität Graz.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., und Teller, E. (1953), *Equations of state calculations by fast computing machine*, J. Chem. Phys. **21**, 1087–1091.

v. Neumann, J. (1951), *Various techniques used in connection with random digits*, National Bureau of Standards Series, Vol. **12**, 36–38.

Niederl, F. (1993), *Computergerechte Erzeugung und Visualisierung von Zufallszahlen mit C–Rand*, Diplomarbeit, Technische Universität Graz.

Raftery, A. E. und Lewis, S. (1992a), *Comment: One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo*, Statistical Science **7**, 493–497.

Raftery, A. E. und Lewis, S. (1992b), *How many iterations in the Gibbs sampler?*, Bayesian Statistics 4 (Hrgb. J. M. Bernardo, J. O. Berger, A. P. Dawid und A. F. M. Smith), Oxford: Oxford University Press, 763–773.

Schmeiser, B. und Chen, M.-H. (1991), *General hit-and-run Monte Carlo sampling for evaluating multidimensional integrals*, Technical report, School of Industrial Engineering, Purdue University.

Spiegelhalter, D., Thomas, A., Best, N. G. und Gilks, W. R. (1996), `BUGS 0.5`: *Bayesian inference using Gibbs sampling – Manual (Version ii)*, University of Cambridge, MRC Biostatistics Unit.
Handbuch und Programmpaket `BUGS` im Internet erhältlich unter der Adresse `http://www.mrc-bsu.cam.ac.uk`

Tierney, L. (1994), *Markov chains for exploring posterior distributions (with discussion)*, Ann. Statist. **22**, 1701–1762.

Anhang A

Die Programme `ARS` und `ARMS`

Zur Analyse der adaptiven Verwerfungsmethode wurden zwei Programme entwickelt: Mit `ARS` können Stichproben mit Hilfe der Tangenten- oder der Sekantenmethode erzeugt werden; in `ARMS` wurde die adaptive Verwerfungsmethode mit Metropolis-Hastings-Schritt implementiert. Die Programme wurden in Borland C++ geschrieben und können von einem IBM kompatiblem Personalcomputer mit einem 80x86 Prozessor ausgeführt werden. `ARS` und `ARMS` haben eine graphische Oberfläche, für deren Verwendung wenigstens eine EGA-Karte Voraussetzung ist.

Die Programme wurden ursprünglich zur Untersuchung der Erzeugung von Zufallszahlen der GIG(λ,β)-Verteilung entworfen. Der objektorientierte Programmcode ist aber so erweiterbar, daß der Benutzer damit auch andere Verteilungen simulieren kann. Für eine allgemeine Anwendung müssen neue Klassen von bestehenden Basisklassen abgeleitet werden und einige globale Variable erweitert werden.

Es gibt zwei Möglichkeiten, den bestehenden Programmcode weiterzuverwenden:

1. Ableitung von den Basisklassen `C_ARS` und `C_ARMS` zur Simulation von beliebigen Verteilungen. Die beiden Klassen sind unabhängig von einer graphischen Oberfläche und können in jeder C++-Applikation eingesetzt werden (s. Abschnitt A.1).

2. Erweiterung der Programme `ARS` und `ARMS` durch Hinzufügen beliebiger Testverteilungen, für die die bestehende Oberfläche mit Visualisierung der Hutfunktionen und Darstellung der Stichproben in Histogrammen genutzt werden kann (s. Abschnitt A.2).

Als Ausgangspunkt einer Erweiterung der besprochenen Klassen und Programme können die mit dem übrigen Quellcode mitgelieferten Dateien `template.h` und `template.cpp` dienen. Sie enthalten die für die Integration einer neuen Verteilung notwendigen Klassen, die vom Benutzer noch mit den charakteristischen Funktionen und Werten der Verteilung ergänzt werden müssen.

Der Programmcode von `ARS` und `ARMS` beruht auf den Algorithmen von Gilks (1992) und Gilks *et al.* (1995), die in den Abschnitten 3.2 und 3.3 beschrieben sind.

A.1 Die Basisklassen

Abhängig vom Typ der Verteilung, aus der Zufallszahlen erzeugt werden sollen, ist zur Simulation von unterschiedlichen Basisklassen abzuleiten. Wenn die Verteilung logkonkav ist, sollte die Klasse `C_ARS` gewählt werden, ansonsten ist die Klasse `C_ARMS` zu verwenden.

Für Details, die über die Erläuterungen der folgenden Abschnitte bezüglich der Basisklassen hinausgehen, sei auf die Kommentare in den Dateien `ars.h` und `arms.h` verwiesen.

A.1.1 Die Klasse C_ARS

Die Klasse `C_ARS` ist eine rein virtuelle Basisklasse, die die Funktionalität, welche Tangenten- und Sekantenmethode teilen, enthält. Von ihr sind die beiden ebenfalls rein virtuellen Klassen `C_ARS_Tang`, in der die Tangentenmethode, und `C_ARS_Sec`, in der die Sekantenmethode implementiert sind, abgeleitet. Als rein virtuelle Klassen können von `C_ARS_Tang` und `C_ARS_Sec` nur Instanzen abgeleiteter Klassen angelegt werden. Die Deklarationen und Definitionen der drei genannten Klassen finden sich in den Dateien `ars.h` und `ars.cpp`.

Um funktionsfähige Klassen zu konstruieren, müssen folgende Funktionen neu definiert werden (für alle abgeleiteten Klassen dieses Kapitels gilt: Konstruktor und Destruktor enthalten keine Anweisungen, wenn sie nicht ausdrücklich für die entsprechende Klasse beschrieben werden):

- *Konstruktor:* Enthält den Aufruf der Funktion `InitAbscissae()` (s. nächster Punkt). Dem Konstruktor können optional bis zu zwei reelle Parameter übergeben werden. Sie sind die Gestaltparameter der Verteilung, von der die Stichprobe erzeugt werden soll.

- `void InitAbscissae()`: In dieser Funktion werden die Anzahl r der Stützstellen mit `m_nNoOfAbsc` und die Stützstellen selbst durch `x[1]`,...,`x[r]` festgelegt (zur Wahl der Startstützstellen s. Abschnitt 4.3.1). Mit `x[0]` und `x[r+1]` werden die untere bzw. obere Schranke des Definitionsbereichs spezifiziert. Am Ende der Funktion muß die Funktion `void InitARS()` der Basisklasse aufgerufen werden. In ihr werden Initialisierungen wie die Berechnung der Schnittpunkte der Tangenten/Sekanten vorgenommen.

- `double LogDens(double x)`: Definition der logarithmierten Dichtefunktion $l(\cdot)$, von der Zufallszahlen erzeugt werden sollen. Der Eingabeparameter ist die Stelle `x` des Definitionsbereiches, an der die logarithmierte Dichte ausgewertet wird, Rückgabewert ist der berechnete Funktionswert.

- Nur für `C_ARS_Tang`: `double DiffLogDens(double)`: Definition der abgeleiteten logarithmierten Dichtefunktion. Eingabeparameter und Rückgabewert entsprechen denen von `double LogDens(double x)`.

Nach Implementation obiger Funktionen kann eine Instanz der abgeleiteten Klasse angelegt werden, und mit dem Aufruf der Memberfunktion `double Generate()` wird eine

Zufallszahl der spezifizierten Dichte erzeugt. Der Adaptionsprozeß erfolgt sukzessive mit der Generierung von neuen Zufallszahlen. Neue Stützstellen (max. 200) werden automatisch von der Basisklasse gespeichert.

Der Startwert des Gleichverteilungsgenerators `ULONG DRand()` (s. Niederl, 1993), der zur Erzeugung von Kandidaten aus der Hutfunktion notwendig ist, kann mit der öffentlichen Funktion `SetSeed(ULONG newseed)` der Basisklasse festgelegt werden.

A.1.2 Die Klasse C_ARMS

Die Ableitung einer Klasse von `C_ARMS` zur Erzeugung von Zufallszahlen gemäß der adaptiven Verwerfungsmethode mit Metropolis-Hastings-Schritt erfolgt gleich wie die von `C_ARS_Sec` aus dem vorangegangenen Abschnitt. Vor Beginn des Erzeugungsprozesses muß mit der Funktion `void SetZeroState(double newstate)` der Ausgangszustand der Markovkette, die die Gesamtheit der generierten Zahlen bildet, festgelegt werden.

Der Programmcode zu `C_ARMS` ist in den Dateien `arms.h` und `arms.cpp` enthalten.

A.2 Erweiterung der Programme ARS und ARMS

Zur Auswertung von Stichproben, die mit den Klassen `C_ARS` und `C_ARMS` erzeugt wurden, kann die graphische Oberfläche von `ARS` und `ARMS` genutzt werden. Dazu müssen im Programmcode einige Erweiterungen vorgenommen werden. Die Modifikationen sind für `ARS` und `ARMS` identisch und müssen für jede neue Verteilung in gleicher Weise ausgeführt werden.

C++-Applikationen werden im allgemeinen in Projekten verwaltet. Es wird empfohlen, die im folgenden beschriebenen Änderungen (außer denen, die am Hauptmodul in `main.cpp` vorgenommen werden) für jede Verteilung in einer Header- (`*.h`) und Klassendefinitionsdatei (`*.cpp`) zu speichern und diese Dateien für den Kompilations- und Linkvorgang im Projekt zu integrieren.

Nachdem die von `C_ARS_Tang` und `C_ARS_Sec` bzw. `C_ARS` abgeleiteten Klassen gemäß Abschnitt A.1 implementiert wurden, müssen noch folgende Erweiterungen durchgeführt werden.

- Ableitung von der Klasse `C_Stat`: Mit ihr werden die theoretischen statistischen Kenngrößen berechnet. Folgende Funktionen sind zu diesem Zweck zu definieren:
  - `void CalcTheoretDistrVals()`: Hier werden wahlweise entweder die theoretischen Momente (Mittelwert, Standardabweichung, Schiefe, Kurtosis) oder die theoretischen Quantile (für das 10%-, 25%-, 50%-, 75%- und 90%-Niveau) berechnet.
  - `double GetLowerBound()`: Berechnung der unteren Schranke des Definitionsbereichs.
  - `double GetUpperBound()`: Berechnung der oberen Schranke des Definitionsbereichs.

- Ableitung von der Klasse `C_ParamDlg`: Mit ihrer Hilfe wird die Eingabe der Gestaltparameter realisiert. Abhängig von der Anzahl der Parameter können bis zu zwei Instanzen von den abgeleiteten Klassen angelegt werden. Hat die zu betrachtende Verteilung keine Gestaltparameter, muß diese Klasse nicht definiert werden.
  - *Konstruktor*: Die Klasse `C_ParamDlg` ist eine Template-Klasse und kann für reelle und natürliche Parameter genutzt werden. Da Gestaltparameter von Verteilungen im allgemeinen reelle Zahlen sind, soll im Konstruktor der Typ `double` gewählt werden. Weiters sind im Konstruktor keine Anweisungen nötig.
  - `void CheckParam(PARAMSTR ErrStr, double param)`: Mit dieser virtuellen Funktion wird der Bereich eines Gestaltparameters der Verteilung geprüft. Liegt eine Eingabe außerhalb des zulässigen Bereichs, muß der übergebene String mit einer entsprechenden Fehlermeldung belegt werden.

Um die in den vorangegangenen Punkten besprochenen Ergänzungen durchzuführen, wird empfohlen, die mit dem Quellcode zur Verfügung gestellten Dateien `template.h` und `template.cpp` zu verwenden. Sie enthalten alle genannten Klassen und Funktionen. Nach den notwendigen Anpassungen können die Dateien umbenannt und ins Projekt integriert werden.

Folgende Adaptionen sind im Modul `main.cpp` vorzunehmen:

- `distribution`: In diesem globalen Feld werden Charakteristika der hinzugefügten Verteilung gespeichert (Name, Anzahl der Gestaltparameter, Vergleich der Stichprobe mit theoretischen Momenten oder Quantilen).
- `param`: Dieses globale Feld enthält die Spezifikation der Gestaltparameter (Name, Defaultwert, obere und untere Schranke des zulässigen Bereichs).
- `menu`: Mit `AppendItem(const char *`*Verteilungsname*`)` wird in der Funktion `InitMenus()` im Menü aller Verteilungen ein Eintrag für die neue Verteilung hinzugefügt.
- Schließlich müssen in der Funktion `OnDistrSel()` dynamisch Instanzen aller neu definierten Klassen angelegt werden.

A.3 Dateien von ARS und ARMS

Für einen fehlerfreien Kompilations- und Linkvorgang der Programme ARS und ARMS sind die Dateien aus Tabelle A.1 nötig. In der C++-Entwicklungsumgebung müssen die unter *Definitionen* und *Graphikroutinen* aufgeführten Dateien in das zu erstellende Projekt integriert werden.

Projekt ARS	Projekt ARMS
Deklarationen:	*Deklarationen:*
ars.h	arms.h
main.h	main.h
screen.h	screen.h
menu.h	menu.h
paramdlg.h	paramdlg.h
globals.h	globals.h
txt_cnst.h	txt_cnst.h
gig.h	gig.h
normal.h	burrv.h
Definitionen:	*Definitionen:*
ars.cpp	arms.cpp
main.cpp	main.cpp
screen.cpp	screen.cpp
menu.cpp	menu.cpp
paramdlg.cpp	paramdlg.cpp
gig.cpp	gig.cpp
bessel.cpp	bessel.cpp
normal.cpp	burrv.cpp
Graphikroutinen:	*Graphikroutinen:*
egavga.obj	egavga.obj
litt.obj	litt.obj
trip.obj	trip.obj

Tabelle A.1: *Dateiliste der Programm ARS und ARMS .*

Alle Dateien aus Tabelle A.1 sind auf der dieser Diplomarbeit beiliegenden Diskette zu finden. Neben den Quelltextdateien stehen noch die ausführbaren Programme `ars.exe` und `arms.exe`, die zu den Programmen gehörenden Projektdateien (`*.prj`) und die Template-Dateien zur Integration neuer Verteilungen zur Verfügung.