On the Numerics of Estimating Generalized Hyperbolic Distributions

A Master Thesis Presented

by

Wang, Congcong

(189010)

to

Prof. Dr. Wolfgang Härdle

CASE - Center of Applied Statistics and Economics

Humboldt-Universität zu Berlin

in partial fulfillment of the requirements

for the degree of

Master of Science

Berlin, September 15, 2005
DECLARATION OF AUTHORSHIP

I hereby confirm that I have authored this master thesis independently and without use of others than the indicated resources. All passages, which are literally or in general matter taken out of publications or other resources, are marked as such.

Wang, Congcong

Berlin, September 15, 2005
ACKNOWLEDGMENT

I would like to thank Professor Dr. Wolfgang Härdle for giving me the opportunity and motivation to write this thesis.

I’m especially indebted to Ying Chen for her excellent guidance all the time. Furthermore, I’m also grateful to my family and friends, without their support it would be impossible to finish this work.
ABSTRACT

An important empirical fact in financial market is that return distributions are often skewed and heavy-tailed. This paper employs maximum likelihood estimation to estimate the five parameters of generalized hyperbolic distribution, a highly flexible heavy-tailed distribution. The estimation utilizes Powell’s methods in multidimensions and the performance of estimation is measured by simulation studies. Application to the financial market provides us with estimates of return distribution of some financial assets.

Key words and Phrases: Generalized Hyperbolic distribution, Maximum Likelihood Estimation, Powell’s Methods in Multidimensions.
## CONTENTS

1. **Introduction** .......................................................... 12

2. **Generalized Hyperbolic Distributions** .......................... 15
   2.1 **Definition and Parameterization** .......................... 16
       2.1.1 Probability Density Function and Parameterization .. 16
       2.1.2 Representation as A Normal Variance-mean Mixture . 20
   2.2 **Properties** ...................................................... 23
       2.2.1 Moment Generating Function .............................. 23
       2.2.2 Mean and Variance ........................................... 25
       2.2.3 Characteristic Function ................................... 26
   2.3 **Subclasses and Limiting Distributions** .................... 26
       2.3.1 Hyperbolic Distributions ................................. 27
       2.3.2 Normal Inverse Gaussian Distributions ................. 28
       2.3.3 Limiting Distributions .................................... 28
   2.4 **Tail-Behavior** .................................................. 31

3. **Methods of Estimation** ............................................. 33
   3.1 **Maximum-Likelihood Estimation** ............................ 33
   3.2 **Numerical Algorithms** ......................................... 34
       3.2.1 Golden Section Search in One Dimension ............... 34
       3.2.2 Parabolic Interpolation and Brent’s Method in One Dimension .................................................. 37
       3.2.3 Powell’s Methods in Multidimensions ................. 38
   3.3 **Local and Global Maximum** ................................... 40
4. Simulation Studies ............................................. 41
   4.1 Procedure ............................................. 41
   4.2 Estimation with Fractional $\lambda$ .................. 41
   4.3 Estimation with integer $\lambda$ ....................... 53

5. Application to Financial Market ............................. 61
   5.1 Data Description ..................................... 61
   5.2 Data Transformation ................................. 61
   5.3 Estimation .......................................... 64

6. Summary and Outlooks ...................................... 68

7. Appendix .................................................. 69
   7.1 C Codes ............................................. 69
      7.1.1 mlgh.c ........................................ 69
      7.1.2 mlghint.c .................................... 81
   7.2 XploRe Codes ....................................... 93
      7.2.1 mlgh.xpl ...................................... 93
      7.2.2 mlghint.xpl .................................. 93
      7.2.3 ghmv.xpl ..................................... 94
**LIST OF FIGURES**

1.1 Comparison between the pdf curves of a standard Gaussian (blue) and a Cauchy distribution (red) with location parameter 0 and scale parameter 1 ........................................... 13

2.1 Comparison between pdf curves of GH(-0.5,1,0,1,0) and normal(0,1) distribution ........................................... 15

2.2 Pdf curves of three GH distributions with $\lambda = 1.3$, $\alpha = 1$, $\beta = 0$, $\delta = 1$ and $\mu = 0$ (black), $\delta = 2$ (blue), $\delta = 3$ (red dotted) ........................................... 17

2.3 Pdf curves of three GH distributions with $\lambda = 1.3$, $\alpha = 1$, $\beta = 1$, $\delta = 1$ and $\mu = 0$ (black), $\delta = 2$ (blue), $\delta = 3$ (red dotted) ........................................... 18

2.4 Pdf curves of three GH distributions with $\lambda = 1.3$, $\alpha = 1$, $\beta = 1$, $\delta = 1$ and $\mu = 0$, $\beta = -0.25$ (blue), $\beta = 0.55$ (red dotted) ........................................... 19

2.5 Pdf curves of three GH distributions with $\lambda = 1.3$, $\alpha = 1$, $\beta = 0$, $\delta = 1$ and $\mu = 0$, $\alpha = 2$ (blue), $\alpha = 0.7$ (red dotted) ........................................... 20

2.6 Pdf curves of three GH distributions with $\lambda = 1.3$, $\alpha = 0$, $\beta = 1$, $\delta = 1$ and $\mu = 0$, $\alpha = 1.5$, $\delta = 1$ (blue), $\alpha = 2.25$, $\delta = 0.8$ (red dotted curve) ........................................... 21

2.7 The density function of the GIG distribution: $\lambda = 0$, $\chi = 1$ (black), $\lambda = 1$, $\chi = 1$ (blue) and $\lambda = 20$, $\chi = 10$ (red) ........................................... 22

2.8 Pdf curves of three GH distributions with $\lambda = 1.3$, $\alpha = 1$, $\beta = 0$, $\delta = 1$ (black), $\delta = 5$ (blue), $\delta = 15$ (red) ........................................... 25

2.9 Pdf curves of three GH distributions with $\alpha = 1$, $\beta = 0.1$, $\delta = 1$, $\mu = 0$, $\lambda = 6$ (blue), $\lambda = 10$ (red) ........................................... 26

2.10 Comparison between pdf curves of GH(0.5,1,615,0,1,0), NIG(-0.5,1,0,1,0) and HYP(1,1.875,0,1,0) distributions ........................................... 27

2.11 Limiting distribution: normal distribution ........................................... 29

2.12 Limiting distribution: t distribution ........................................... 29
2.13 Limiting distribution: Cauchy distribution . . . . . . . . . . . 30
2.14 Tail comparison between GH distributions, pdf (left) and ap-
proximation (right): . . . . . . . . . . . . . . . . . . . . . . . 31
2.15 Tail comparison between GH(-0.5,1,0,1,0) distribution (black)
and its limiting distributions: normal(red), Student-t (blue)
and Cauchy (green) distributions. . . . . . . . . . . . . . . . . 32
3.1 Illustration of successive bracketing of a maximum by golden
section search in one dimension. . . . . . . . . . . . . . . . . 35
3.2 Illustration of successive bracketing of inverse parabolic in-
terpolation in one dimension. . . . . . . . . . . . . . . . . . . 37
3.3 Local and global maximum. . . . . . . . . . . . . . . . . . . 40
4.1 Boxplot of Example (-1,1,0,1,0) . . . . . . . . . . . . . . . . 43
4.2 Comparison between original (blue) and estimated (red) pdf
curves with parameters (-3,1,0,1,0) and fractional $\lambda$ . . . . . . 46
4.3 Comparison between original (blue) and estimated (red) pdf
curves with parameters (-1,1,0,1,0) and fractional $\lambda$ . . . . . . 46
4.4 Comparison between original (blue) and 6th estimated (red)
pdf curves with parameters (-1,1,0,1,0) and fractional $\lambda$ . . . . . 47
4.5 Comparison between original (blue) and 11th estimated (red)
pdf curves with parameters (-1,1,0,1,0) and fractional $\lambda$ . . . . . 47
4.6 Comparison between original (blue) and estimated (red) pdf
curves with parameters (-0.5,1,0,1,0) and fractional $\lambda$ . . . . . 48
4.7 Comparison between original (blue) and estimated (red) pdf
curves with parameters (0.49,1,0,1,0) and fractional $\lambda$ . . . . . 48
4.8 Comparison between original (blue) and estimated (red) pdf
curves with parameters (1,1,0,1,0) and fractional $\lambda$ . . . . . . 49
4.9 Comparison between original (blue) and estimated (red) pdf
curves with parameters (3,1,0,1,0) and fractional $\lambda$ . . . . . . 49
4.10 Comparison between original (blue) and estimated (red) pdf
curves with parameters (-0.5,2,0,1,0) and fractional $\lambda$ . . . . . 50
4.11 Comparison between original (blue) and estimated (red) pdf
curves with parameters (-0.5,4.5,0,1,0) and fractional $\lambda$ . . . . . 50
4.12 Comparison between original (blue) and estimated (red) pdf
curves with parameters (1,2,0,1,0) and fractional $\lambda$ . . . . . . 51
4.13 Comparison between original (blue) and estimated (red) pdf curves with parameters (1,4.5,0,1,0) and fractional $\lambda$ 51
4.14 Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1.5,0.5,1,0) and fractional $\lambda$ 52
4.15 Comparison between original (blue) and estimated (red) pdf curves with parameters (1,1.5,0.5,1,0) and fractional $\lambda$ 52
4.16 Comparison between original (blue) and estimated (red) pdf curves with parameters (-1,1,0,1,0) and integer $\lambda$ 55
4.17 Comparison between original (blue) and estimated (red) pdf curves with parameters (-1,1,0,1,0) and integer $\lambda$ 55
4.18 Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1,0,1,0) and integer $\lambda$ 56
4.19 Comparison between original (blue) and estimated (red) pdf curves with parameters (0.49,1,0,1,0) and integer $\lambda$ 56
4.20 Comparison between original (blue) and estimated (red) pdf curves with parameters (1,1,0,1,0) and integer $\lambda$ 57
4.21 Comparison between original (blue) and estimated (red) pdf curves with parameters (3,1,0,1,0) and integer $\lambda$ 57
4.22 Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,2,0,1,0) and integer $\lambda$ 58
4.23 Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,4.5,0,1,0) and integer $\lambda$ 58
4.24 Comparison between original (blue) and estimated (red) pdf curves with parameters (1,2,0,1,0) and integer $\lambda$ 59
4.25 Comparison between original (blue) and estimated (red) pdf curves with parameters (1,4.5,0,1,0) and integer $\lambda$ 59
4.26 Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1.5,0.5,1,0) and integer $\lambda$ 60
4.27 Comparison between original (blue) and estimated (red) pdf curves with parameters (1,1.5,0.5,1,0) and integer $\lambda$ 60
5.1 Comparison between kernel density estimation (blue) and GH estimation with fractional $\lambda$ (red). Left - BMW; Right - THY 65
5.2 Comparison between kernel density estimation (blue) and GH estimation with fractional $\lambda$ (red). Left - DMUSD; Right - BPUSD 66
5.3 Comparison between kernel density estimation (blue) and GH estimation with integer $\lambda$ (red). Left - BMW; Right - THY . . 66

5.4 Comparison between kernel density estimation (blue) and GH estimation with integer $\lambda$ (red). Left - DMUSD; Right - BPUSD 67
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Limiting cases of GH distributions</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Example: Results of Estimation</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Results of estimation with fractional $\lambda$, original parameters $\lambda,1,0,1,0$</td>
<td>45</td>
</tr>
<tr>
<td>4.3</td>
<td>Results of estimation with fractional $\lambda$, original parameters $-0.5,\alpha,0,1,0$</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>Results of estimation with fractional $\lambda$, original parameters $1,\alpha,0,1,0$</td>
<td>45</td>
</tr>
<tr>
<td>4.5</td>
<td>Results of estimation with fractional $\lambda$, original parameters $\lambda,1.5,0.5,1,0$</td>
<td>45</td>
</tr>
<tr>
<td>4.6</td>
<td>Results of estimation with integer $\lambda$, original parameters $(\lambda,1,0,1,0)$</td>
<td>54</td>
</tr>
<tr>
<td>4.7</td>
<td>Results of estimation with integer $\lambda$, original parameters $-0.5,\alpha,0,1,0$</td>
<td>54</td>
</tr>
<tr>
<td>4.8</td>
<td>Results of estimation with integer $\lambda$, original parameters $(1,\alpha,0,1,0)$</td>
<td>54</td>
</tr>
<tr>
<td>4.9</td>
<td>Results of estimation with integer $\lambda$, original parameters $(\lambda,1.5,0.5,1,0)$</td>
<td>54</td>
</tr>
<tr>
<td>5.1</td>
<td>KPSS test for stock prices and exchange rates with reference point $T = 8$</td>
<td>63</td>
</tr>
<tr>
<td>5.2</td>
<td>KPSS test for log-returns of stocks and currencies with reference point $T = 8$</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>Parameter estimates of GARCH(1,1)</td>
<td>64</td>
</tr>
<tr>
<td>5.4</td>
<td>Parameter estimates of GH distributions with fractional $\lambda$</td>
<td>65</td>
</tr>
<tr>
<td>5.5</td>
<td>Parameter estimates of GH distributions with integer $\lambda$</td>
<td>65</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This thesis focuses on a particular heavy-tailed distribution: generalized hyperbolic (GH) distribution. The aim is to overview GH distributions and to estimate the five parameters of GH distributions in the financial environment.

Heavy-tailed distributions were first introduced by the Italian-born Swiss economist Pareto (1896) and extensively studied by Paul Lévy. Although then these distributions were mainly studied theoretically, nowadays they have found many applications in areas as diverse as finance, medicine, seismology, structural engineering.

A distribution is called heavy-tailed if it has higher probability density in its tail areas compared with a normal distribution with the same mean \( \mu \) and variance \( \sigma^2 \). Figure 1.1 demonstrates the differences of the pdf curves of a standard Gaussian distribution and a Cauchy distribution with location parameter \( \mu = 0 \) and scale parameter \( \sigma = 1 \). The graphic shows that the probability density of the Cauchy distribution is much higher than that of the Gaussian in the tail part, while in the area around the center, the probability density of the Cauchy distribution is much lower.

In terms of kurtosis, a heavy-tailed distribution has kurtosis greater than 3, which is called leptokurtic, in contrast to mesokurtic distribution (kurtosis \( = 3 \)) and platykurtic distribution (kurtosis \( < 3 \)).

An important empirical fact in financial market is that return distributions are often skewed and have heavier tails than the normal distribution. Risk management based on normal assumptions may therefore lead to underestimation of the risk. Researchers have tried to fix this by offering other classes of distributions, first the stable Paretian class and more recently the generalized hyperbolic class.

One of the reasons, which make the GH distributions so popular, is that its five parameters are flexible enough to fit many different data sets well and make GH distributions potentially useful in many different contexts. As a striking feature of GH distribution, it embraces many subclasses and limiting distributions, e.g. hyperbolic, normal inverse Gaussian, Student-t and normal distributions. All of them have been used to model financial
returns.

In recent years normal inverse Gaussian (NIG) distribution, a subclass of GH distribution, has been successfully fitted to returns in financial time series by many researchers; see Eberlein and Keller (1995), Prause (1997), Barndorff-Nielsen (1997), Prause (1999), Barndorff-Nielsen and Shephard (2001). This has opened an area, in which NIG distributions are used as building blocks to model the time dynamics of financial markets.

Since NIG distribution is a special case of GH distribution, in which one of the parameters of GH distribution, namely $\lambda$, is fixed to -1/2, estimation of NIG distribution is actually a four-parameter estimation. It is rather interesting to further exploit GH distributions, which means to include the parameter $\lambda$ in the estimation, a step from four-parameter to five-parameter estimation.

Fig. 1.1: Comparison between the pdf curves of a standard Gaussian (blue) and a Cauchy distribution (red) with location parameter 0 and scale parameter 1.

The work is divided into four parts. The first section starts with definition of GH distributions, the subclasses and limiting distributions of GH distributions, as well as the normal variance-mean mixture presentation. Their properties are also examined in the first part: moment generating function and characteristic function. The second section presents the maximum likelihood estimation and numerical algorithms used to estimate parameters, including Golden Section search and parabolic interpolation, on which
Powell’s method in multi-dimensions are based. In the third part, various original parameter sets are employed in simulation studies to measure the performance of estimation under different situations. Finally, part 4 uses real data to estimate GH distribution density and compares the results with those of nonparametric method.

All results and generated codes using XploRe and C language are gathered at the appendix. The bibliography contains classical references on GH distributions, where deeper computational and mathematical treatment can be found.
2. GENERALIZED HYPERBOLIC DISTRIBUTIONS

The generalized hyperbolic distribution was introduced by Barndorff-Nielsen (1977) for modeling grain size distributions of wind blown sands. The original paper focused on the special case of the hyperbolic distribution. The name of the distributions is derived from the fact that its log-density forms a hyperbola, while the log-density of the normal distribution is a parabola.

GH distributions embrace many subclasses and limiting distributions, e.g. hyperbolic, normal inverse Gaussian, Student-t and normal distributions, which will be discussed in detail in Section 2.3. Being a normal variance-mean mixture, GH distributions possess heavy tails, i.e. the kurtosis is higher than that of normal distribution. Hence it provides the possibility of modeling the well-known heavy tails of return distributions for most financial assets. Figure 2.1 compares the pdf curves of a GH distribution (black) and a normal distribution (red dotted). Both distributions have mean 0 and variance 1.

Fig. 2.1: Comparison between pdf curves of GH(-0.5,1,0,1,0) and normal(0,1) distribution.

GHvsN.png
2. Generalized Hyperbolic Distributions

2.1 Definition and Parameterization

2.1.1 Probability Density Function and Parameterization

We denote the one-dimensional generalized hyperbolic (GH) distribution by \( GH(x; \lambda, \alpha, \beta, \delta, \mu) \) for \( x \in \mathbb{R} \). It then can be characterized via its probability density:

\[
f_{GH}(x; \lambda, \alpha, \beta, \delta, \mu) = a(\lambda, \alpha, \beta, \delta, \mu) \left\{ \delta^2 + (x - \mu)^2 \right\}^{(\lambda - \frac{1}{2})/2} \exp\{\beta(x - \mu)\} \times K_{\lambda-\frac{1}{2}}(\alpha \sqrt{\delta^2 + (x - \mu)^2}),
\]

where

\[
a(\lambda, \alpha, \beta, \delta, \mu) = \frac{(\alpha^2 - \beta^2)^{\frac{\lambda}{2}}}{\sqrt{2\pi} \alpha^{\lambda - \frac{1}{2}} \delta^\lambda K_{\lambda}(\delta \sqrt{\alpha^2 - \beta^2})},
\]

and \( K_{\lambda} \) is a modified Bessel function of the third kind with index \( \lambda \):

\[
K_{\lambda}(x) = \frac{1}{2} \int_{0}^{\infty} y^{\lambda - 1} \exp\left\{ -\frac{1}{2} (x + y^{-1}) \right\} dy,
\]

which shows the strict positivity of \( K_{\lambda} \) on \( \mathbb{R} > 0 \). The substitution \( x := y^{-1} \) immediately gives \( K_{-\lambda} = K_{\lambda} \). Furthermore, \( K_{\lambda}(x) \) is obviously monotonically decreasing in \( x \) on \( \mathbb{R} > 0 \).

Besides \( \mu \in \mathbb{R} \), the values which parameters can take are:

\[
\begin{align*}
\delta &\geq 0, |\beta| < \alpha, & \text{if} & \lambda > 0 \\
\delta &> 0, |\beta| < \alpha, & \text{if} & \lambda = 0 \\
\delta &> 0, |\beta| \leq \alpha, & \text{if} & \lambda < 0.
\end{align*}
\]

\( \alpha \) and \( \beta \) are sometimes replaced by alternative parameterizations in the literature:

\[
\begin{align*}
\rho &= \frac{\beta}{\alpha}, & \zeta &= \delta \sqrt{\alpha^2 - \beta^2} \\
\chi &= \rho \xi, & \xi &= \frac{1}{\sqrt{1 + \zeta}} \\
\bar{\alpha} &= \delta \alpha, & \bar{\beta} &= \delta \beta
\end{align*}
\]
For symmetric distributions $\beta = \bar{\beta} = \rho = \chi = 0$ holds.

Roughly speaking, $\mu$ is a location parameter, $\delta$ serves for scaling, $\beta$ determines the skewness and $\alpha$ the shape. Increasing $\xi$ or decreasing $\zeta$ or $\delta \alpha$ reflects an increase in the kurtosis. $\lambda$ characterizes certain subclasses and considerably influences the size of the tails.

Figure 2.2 shows that the effect of location parameter $\mu$ is very clear: an increase in $\mu$ moves the pdf curves of GH distribution rightward horizontally.

The role played by scale parameter $\delta$ is demonstrated in Figure 2.3. With an increase in the value of $\delta$, the pdf curve becomes flatter. At the same time, a raise on $\delta$ with $\alpha$ remaining constant decreases the kurtosis of the GH distributions. That’s why we mentioned "roughly speaking" concerning the effects of the parameters in GH distributions, because they are multifold: one parameter can have an impact on different moments.

The versatility of parameter $\beta$ is even more obvious in Figure 2.4. The main effect on the probability density is, that the pdf curve skews left when $\beta < 0$ and skews right when $\beta > 0$; the density is symmetric when $\beta = 0$. With larger value of absolute value of $\beta$, the skewness is more obvious. Besides the main effect, $\beta$ also moves the pdf curve horizontally, which means it changes the mean. As demonstrated in Figure 2.4, the pdf curve moves rightward...
Fig. 2.3: Pdf curves of three GH distributions with $\lambda = 1.3, \alpha = 1, \beta = 0, \mu = 0$ and $\delta = 1$ (black), $\delta = 2$ (blue), $\delta = 3$ (red dotted) when $\beta$ takes a positive value and moves leftward when $\beta$ is negative. We will go to details when we introduce the moments of GH distributions.

Figure 2.5 illustrates the impact of parameter $\alpha$. A decrease in $\alpha$ results in an increase in kurtosis, which peaks the pdf curve. At the same time, other parameters remaining constant, a decrease in $\alpha$ also forces the variance to increase, which in contrast flattens the curve, vice versa.

Finally let’s take a look at the combined effects of $\alpha \delta$. As indicated before, an increase $\delta \alpha$ reflects a decrease in the kurtosis. In Figure 2.6, we gradually increase the value of $\alpha$ from 1 to 2.25, while decreasing the value of $\delta$ from 1.2 to 0.8, so that the products of $\alpha$ and $\delta$ increases from 1.2 to 1.5, and further to 1.8. The effects are obvious: the red dotted pdf curve, which has the largest $\alpha \delta$ value, has the fastest decaying speed in tail areas, implying it has the smallest kurtosis while the black curve has the slowest decaying speed.

With a different way parameterization, the role of the scale and location parameters $\delta$ respectively $\mu$ become more obvious. Barndorff-Nielsen and Stelzer (2005) represent the pdf of GH distributions as:
2. Generalized Hyperbolic Distributions

Fig. 2.4: Pdf curves of three GH distributions with \( \lambda = 1.3, \alpha = 1, \delta = 1, \mu = 0 \) and \( \beta = 0 \) (black), \( \beta = -0.25 \) (blue), \( \beta = 0.55 \) (red dotted) 

\[
f_{GH}(x; \lambda, \alpha, \beta, \delta, \mu) = \frac{\bar{\gamma}^{\lambda/2-1/4}}{\sqrt{2\pi}\delta K_{\lambda}(\bar{\gamma})} \left\{ 1 + \frac{(x - \mu)^2}{\delta^2} \right\}^{\lambda/2-1/4} \times K_{\lambda-1/2} \left( \bar{\alpha} \sqrt{1 + \frac{(x - \mu)^2}{\delta^2}} \right) \exp\{\beta(x - \mu)\} \tag{2.8}
\]

where

\[
\gamma = \sqrt{\alpha^2 - \beta^2}, \quad \bar{\alpha} = \delta \alpha, \\
\bar{\beta} = \delta \beta, \quad \bar{\gamma} = \delta \gamma.
\tag{2.9}
\]

For those, who are not familiar with details of GH distributions, the term \( (\frac{x - \mu}{\delta}) \) is similar to normalization.

Suppose \( X \sim GH(\lambda, \alpha, \beta, \delta, \mu) \), Blasild (1981) proved that a linear transformation \( Y = aX + b \) is again GH-distribution with parameters \( \lambda^+ = \lambda, \alpha^+ = \alpha/|a|, \beta^+ = \beta/|a|, \delta^+ = \delta|a| \) and \( \mu^+ = a\mu + b \), which means

\[
\tilde{X} = aX + b \sim GH(\lambda, \alpha/|a|, \beta/|a|, \delta|a|, a\mu + b).
\tag{2.10}
\]
Since $\alpha^+\delta^+ = \alpha \delta$ and $\beta^+\delta^+ = \beta \delta$ holds, the term $\lambda$, $\bar{\alpha}$ and $\bar{\beta}$ are scale- and location-invariant parameters of the univariate GH distributions. The same holds for the other parameterizations ($\zeta, \rho$) and ($\xi, \chi$).

### 2.1.2 Representation as A Normal Variance-mean Mixture

When we work with GH distributions, it is sometimes more convenient to represent them in other forms. For example, GH distributions can be represented as a normal variance-mean mixture with normal distribution of mean $\xi = \mu + \beta \sigma^2$ and variance $\sigma^2$. The mixture includes generalized inverse Gaussian (GIG) distributions. The representation as mixture is very helpful in studying GH distributions, since GIG is often used to generate GH random variables. See `rndgh.xpl`.

Mixture modeling means to modeling a statistical distribution by a mixture (or weighted sum) of different distributions. With unrestricted choices of component density functions, it can approximate any continuous density to arbitrary accuracy, given sufficiently large number of component. The pdf
of a mixture which consists of \( n \) distributions can be written as

\[
f(x) = \sum_{l=1}^{n} w_l p_l(x),
\]

under the constraints:

\[
0 \leq w_l \leq 1, \quad \sum_{l=1}^{n} w_l = 1, \quad \int p_l(x)dx = 1,
\]

where \( p_l(x) \) is the pdf of \( l \)’th distribution or \( l \)’th component density and \( w_l \) is called coefficient, which can be viewed as weight of \( l \)’th distribution in the mixture.

If we denote the generalized inverse Gaussian as \( GIG(x; \lambda, \chi, \psi) \), GH distributions can be represented as:

\[
f_{GH}(x; \lambda, \alpha, \beta, \delta, \mu) = \int_{0}^{\infty} N(x; \mu + \beta \omega, \omega)GIG(\omega; \lambda, \delta^2, \alpha^2 - \beta^2)d\omega.
\]
where $N$ is the normal density function with respect to mean $\mu + \beta \omega$ and variance $\omega$. $GIG(x; \lambda, \chi, \psi)$ has the following probability density function:

$$f_{GIG}(x; \lambda, \chi, \psi) = \frac{(\psi/\chi)^{\lambda/2}}{2K_\lambda(\sqrt{\psi\chi})} e(x; \lambda, \chi, \psi),$$

with

$$e(x; \lambda, \chi, \psi) = x^{\lambda-1} \exp\{- (1/2)(\chi x^{-1} + \psi x)\}, x > 0.$$  \hspace{1cm} (2.14)

See Prause (1999). The domain of variation for the parameters is

- $\chi > 0$, $\psi \geq 0$ if $\lambda < 0$,
- $\chi > 0$, $\psi > 0$ if $\lambda = 0$,
- $\chi \geq 0$, $\psi > 0$ if $\lambda > 0$.

Fig. 2.7: The density function of the GIG distribution: $\lambda = 0, \chi = 1$ (black), $\lambda = 1, \chi = 1$ (blue) and $\lambda = 20, \chi = 10$ (red).

With the parameterization of GH distributions given in equation 2.9, Barndorff-Nielsen and Stelzer (2005) write the GIG density function as:
\[
\begin{align*}
    f_{\text{GIG}}(x; \lambda, \delta, \gamma) &= \frac{(\gamma/\delta)^\lambda}{2K_\lambda(\delta \gamma)} x^{\lambda-1} \exp\left\{-\frac{1}{2}(\delta^2 x^{-1} + \gamma^2 x)\right\} \\
    &= \frac{\gamma^\lambda}{2K_\lambda(\gamma)} \delta^{-2\lambda} x^{\lambda-1} \exp\left\{-\frac{1}{2}(\delta^2 x^{-1} + \gamma^2 \delta^{-2} x)\right\}. 
\end{align*}
\]

The density can also be reparameterized by setting \( \xi = \sqrt{\chi \psi} \), a shape parameter, and \( \eta = \sqrt{\chi / \psi} \), a scale parameter. When \( \eta = 1 \), \( \xi = \chi = \psi \). See Atkinson (1982). Figure 2.7 demonstrates three members of generalized inverse Gaussian distribution with the same scale parameter \( \eta = 1 \) but different shape parameters.

2.2 Properties

2.2.1 Moment Generating Function

We assume that \( \mu = 0 \) for simplicity. Since

\[
\int f_{\text{GH}}(x; \lambda, \alpha, \beta, \delta, 0) = 1,
\]

\[
\int \left\{ \delta^2 + x^2 \right\}^{(\lambda-\frac{1}{2})/2} \exp\{\beta x\} K_{\lambda-\frac{1}{2}}(\alpha \sqrt{\delta^2 + x^2}) dx = \frac{1}{a(\lambda, \alpha, \beta, \delta, 0)},
\]

the moment generating function of GH distributions with \( |\beta + u| < \alpha \) is simply the ratio of the norming constants \( a \) defined in equation 2.2 corresponding to the parameters \( (\lambda, \alpha, \beta, \delta) \) and \( (\lambda, \alpha, \beta + \mu, \delta) \). See Prause (1999).

\[
M_{\text{GH}}(u) = \int \exp\{ux\} f_{\text{GH}}(x; \lambda, \alpha, \beta, \delta, 0) dx
\]

\[
= a(\lambda, \alpha, \beta, \delta) \int \exp\{ux\} (\delta^2 + x^2)^{\frac{1}{2}(\lambda-\frac{1}{2})} K_{\lambda-\frac{1}{2}}(\alpha \sqrt{\delta^2 + x^2}) \exp\{\beta x\} dx
\]

\[
= \frac{a(\lambda, \alpha, \beta, \delta)}{a(\lambda, \alpha, \beta + \mu, \delta)}
\]

\[
= \frac{(\alpha^2 - \beta^2)^{\lambda/2}}{\sqrt{2\pi \delta^\lambda} K_\lambda(\delta \sqrt{\alpha^2 - (\beta + u)^2})} \frac{\sqrt{2\pi \delta^\lambda} \alpha^{\lambda-1/2} K_\lambda(\delta \sqrt{\alpha^2 - (\beta + u)^2})}{\left\{\alpha^2 - (\beta + u)^2\right\}^{\lambda/2}}
\]

\[
= \left\{\frac{\alpha^2 - \beta^2}{\alpha^2 - (\beta + u)^2}\right\}^{\lambda/2} \frac{K_\lambda(\delta \sqrt{\alpha^2 - (\beta + u)^2})}{K_\lambda(\delta \sqrt{\alpha^2 - \beta^2})}. 
\]
The moment generating function of the generalized hyperbolic distribution is given by:

\[ M_{GH}(u) = \exp\{u\mu\} \left\{ \frac{\alpha^2 - \beta^2}{\alpha^2 - (\beta + u)^2} \right\}^{\lambda/2} K_\lambda(\delta \sqrt{\alpha^2 - (\beta + u)^2}) \frac{\lambda}{2} K_{\lambda+1} \left( \delta \sqrt{\alpha^2 - \beta^2} \right), \quad |\beta + u| < \alpha \] (2.17)

in which the restriction $|\beta + u| < \alpha$ comes from the domain of variation of the parameters in equations 2.4.

If $X \sim GH(x; \lambda, \alpha, \beta, \delta, \mu)$, Gut (1995) proved that all moments of $X$ exist. Generalized hyperbolic distributions therefore possess moments of arbitrary order. In particular we take the first two derivatives of $M_{GH}(u)$ to find the mean and variance.

We assume $\mu = 0$ without loss of generality. Since $K_\lambda'(x) = -K_{\lambda+1}(x) + \frac{\lambda}{x} K_\lambda(x)$, Prause (1999) shows that,

\[ M_{GH}'(u) = \frac{(\alpha^2 - \beta^2)^{\lambda/2}}{K_\lambda(\delta \sqrt{\alpha^2 - \beta^2})} \left\{ K_\lambda(\delta \sqrt{\alpha^2 - (\beta + u)^2}) \left\{ \alpha^2 - (\beta + u)^2 \right\}^{-\lambda/2} \right\}' \]

\[ = \frac{(\alpha^2 - \beta^2)^{\lambda/2}}{K_\lambda(\delta \sqrt{\alpha^2 - \beta^2})} K_{\lambda+1}(\delta \sqrt{\alpha^2 - (\beta + u)^2}) \left\{ \alpha^2 - (\beta + u)^2 \right\}^{(\lambda+1)/2}. \] (2.18)

If we insert 0 for $u$ we obtain

\[ M_{GH}'(0) = \beta \frac{\delta K_{\lambda+1}(\delta \sqrt{\alpha^2 - \beta^2})}{\sqrt{\alpha^2 - \beta^2} K_\lambda(\delta \sqrt{\alpha^2 - \beta^2})}. \] (2.19)

Applying the definition of $\gamma$ and $\bar{\gamma}$ in 2.9 gives the mean of generalized hyperbolic distribution:

\[ E[X] = \mu + \beta \frac{\delta K_{\lambda+1}(\gamma)}{\gamma K_\lambda(\gamma)}. \] (2.20)

Taking the second derivative gives the variance:

\[ \text{Var}[X] = \delta^2 \left[ \frac{K_{\lambda+1}(\gamma)}{\gamma K_\lambda(\gamma)} + \frac{\beta^2}{\gamma^2} \left\{ \frac{K_{\lambda+2}(\gamma)}{K_\lambda(\gamma)} - \left( \frac{K_{\lambda+1}(\gamma)}{K_\lambda(\gamma)} \right)^2 \right\} \right]. \] (2.21)
2. Generalized Hyperbolic Distributions

2.2.2 Mean and Variance

Higher order moments can also be calculated because of the existence of all moments, but the expressions become more and more complicated. Clearly, both formulae for mean and variance are less complicated in the symmetric case, e.g. when we assume $\beta = 0$, recalling that $\beta$ roughly speaking describes the skewness. The mean is simply $\mu$ under this condition.

The expression of mean of GH distributions in equation 2.20 shows that $\lambda, \alpha, \beta, \delta$ and $\mu$ all have effects on the mean. The impacts of $\mu$ and $\beta$ have been illustrated by Figure 2.2 and Figure 2.4 respectively. With the parameter set in Figure 2.4, it is easy to obtain, that $E[X] = 0$ for the black curve, $E[X] = -0.8385$ for the blue curve and $E[X] = 2.3971$ for the red dotted curve. See Appendix 7.2.3 ghmv.xpl. However the effect of $\alpha, \delta$ and $\lambda$ is not developed, since $\beta$ takes the value of 0 in Figure 2.3 and Figure 2.5 therefore the second term of equation 2.20 does not influence the mean. By setting $\beta = 0.1$, Figure 2.8 and 2.9 demonstrate the impacts of $\delta$ and $\lambda$ on the first moment of GH distribution. It is easy to discern, that the mean of GH distribution increases in $\delta$ and also in $\lambda$ under given parameters setting. In the graphics, the curves move rightward with the increase of $\delta$ and $\lambda$. 

![Fig. 2.8: Pdf curves of three GH distributions with $\lambda = 1.3, \alpha = 1, \beta = 0.1, \mu = 0$ and $\delta = 1$ (black), $\delta = 5$ (blue), $\delta = 15$ (red).]
2. Generalized Hyperbolic Distributions

2.2.3 Characteristic Function

Prause (1999) showed, that the radius of convergence of the moment generating function $M_{GH}$ around zero is $\alpha - \beta$, and with Gut (1995) Theorem III 3.3 the moment generating function $M_{GH}$ is a real analytic, i.e. it can be expanded in a power series around zero.

The characteristic function of the generalized hyperbolic distribution is given by

$$
\varphi_{GH}(u) = \exp\{i\mu u\} \left\{ \frac{\alpha^2 - \beta^2}{\alpha^2 - (\beta + iu)^2} \right\}^{\lambda/2} \frac{K_\lambda(\delta \sqrt{\alpha^2 - (\beta + iu)^2})}{K_\lambda(\delta \sqrt{\alpha^2 - \beta^2})}. \quad (2.22)
$$

2.3 Subclasses and Limiting Distributions

With specific values of $\lambda$, different subclasses are identified. For $\lambda = 1$ we obtain hyperbolic (HYP) distributions and for $\lambda = -1/2$ we get the normal inverse Gaussian (NIG) distributions.
When $\lambda \in 1/2\mathbb{Z}$, the Bessel function $K_{\lambda}$ can be simplified to

$$K_{n+1/2}(x) = \sqrt{\frac{\pi}{2}} x^{-1/2} \exp\{-x\} \left(1 + \sum_{i=1}^{n} \frac{(n+i)!}{(n-i)!i!} (2x)^{-i}\right), \quad (2.23)$$

for $\lambda = n+1/2, n = 0, 1, 2, \ldots$. Since $K_{\lambda}(x) = K_{-\lambda}(x)$, we obtain $K_{1/2}(x) = K_{-1/2}(x) = \sqrt{\pi/2} x^{-1/2} \exp\{-x\}$. This allows simpler expressions for probability density functions of HYP distributions and NIG distributions.

### 2.3.1 Hyperbolic Distributions

The HYP distribution has the following pdf:

$$f_{\text{HYP}}(x; \alpha, \beta, \delta, \mu) = \frac{\sqrt{\alpha^2 - \beta^2}}{2\alpha \delta K_1(\delta \sqrt{\alpha^2 - \beta^2})} \exp\{-\alpha \sqrt{\delta^2 + (x - \mu)^2 + \beta(x - \mu)}\} \quad (2.24)$$

The focus of original paper by Barndorff-Nielsen (1977) was on the subclass of the HYP distributions. Barndorff-Nielsen (1982) then employed the three-dimensional HYP distributions in relativistic statistical Physics.
The application in particle size distributions of sand is further discussed by Barndorff-Nielsen, Blæsild, Jensen and Sørensen (1983), Barndorff-Nielsen, Blæsild, Jensen and Sørensen (1985), Barndorff-Nielsen and Christiansen (1988), Hartmann and Bowman (1993), Sutherland and Lee (1994) applied the distributions to coastal sediments. In Xu, Durst and Tropea (1993) HYP distributions found application in fluid sprays. Other areas, where HYP distributions have been employed, include biology (e.g. Blæsild (1981)) and primary magnetization of lava flows (Kristjansson and McDougall (1982)). Furthermore, in Barndorff-Nielsen, Jensen and Sørensen (1989) the HYP distribution is employed to model wind shear data of landing aircrafts parsimoniously.

2.3.2 Normal Inverse Gaussian Distributions

The NIG distribution has the following pdf:

$$f_{NIG}(x; \alpha, \beta, \delta, \mu) = \frac{\alpha \delta}{\pi} \frac{K_1(\alpha \sqrt{\delta^2 + (x - \mu)^2})}{\sqrt{\delta^2 + (x - \mu)^2}} \exp\{\delta \sqrt{\alpha^2 - \beta^2 + \beta(x - \mu)}\}$$

(2.25)


In recent years many authors have successfully fitted NIG distribution to returns in financial time series; see Eberlein and Keller (1995), Prause (1997), Barndorff-Nielsen (1997), Prause (1999), Barndorff-Nielsen and Shephard (2001). This has, in particular, led to modeling the time dynamics of financial markets by stochastic processes using NIG distributions as building blocks.

Figure 2.10 illustrates the comparison between pdf curves of GH distribution and its two subclasses introduced above. The three distributions have mean 0 and variance 1. NIG distribution (-0.5,1,0,1,0) is identified in the graphic as black curve, GH (0.5,1.615,0,1,0) the blue curve and HYP (1,1.875,0,1,0) the red dotted curve, with parameter order \((\lambda, \alpha, \beta, \delta, \mu)\).

2.3.3 Limiting Distributions

An important aspect is that GH distributions cover many special cases, including limiting distributions of normal, Student-t and Cauchy distributions.
The normal distributions are obtained as a limiting case of the GH distributions.

Fig. 2.11: Limiting distribution: normal distribution.

Fig. 2.12: Limiting distribution: t distribution.
2. Generalized Hyperbolic Distributions

Fig. 2.13: Limiting distribution: Cauchy distribution. 

Integrations for $\delta \to \infty$ and $\delta/\alpha \to \sigma^2$. See Barndorff-Nielsen (1978). In Figure 2.11 we set the parameters of GH distribution to $(0.5, 26, 0, 26, 0)$, where $\alpha = \delta = 26$, a relatively large value to approximate infinity. The pdf curve of GH$(0.5, 26, 0, 26, 0)$ (red dots) laps over the pdf curve of standard normal distribution (black). The blue curve of GH$(0.5, 1, 0, 1, 0)$ is placed here as a comparison.

The Student-t distribution results from a mixture of normal and inverse gamma distributions. We have a Student-t distributions as a limit of GH distributions for $\lambda < 0$ and $\alpha = \beta = \mu = 0$. See Barndorff-Nielsen (1978). In Figure 2.12 red dots of GH$(-2, 1.0 \times 10^{-10}, 0, 2, 0)$ lap over the curve of Student-t distribution with degrees of freedom 4, since $\lambda = -n/2$, $\delta = \sqrt{n}$, denoting the degrees of freedom by n. $\alpha$ here is set to a small number to approximate 0. The blue curve stands for GH$(1, 1, 0, 1, 0)$ in the graphic.

Cauchy distribution can be obtained from limiting case of GH distributions with $\lambda = -1/2$, $\alpha = \beta = 0$ and $\delta = 1$. See Blaesild (1999). The pdf curve of GH$(-0.5, 1.0 \times 10^{-10}, 0, 1, 0)$ is presented as the red dots in Figure 2.13. The black curve is the pdf curve of standard Cauchy distribution, which is lapped over by the red dots. The blue curve stands for GH$(0.5, 1, 0, 1, 0)$ in the graphic.

As a summary, the limiting distributions discussed here are listed in Table 2.1.
2. Generalized Hyperbolic Distributions

Parameter Description

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal($\mu, \delta^2$)</td>
<td>$\delta \to \infty$ and $\delta/\alpha \to \sigma^2$.</td>
</tr>
<tr>
<td>Student-t($n$)</td>
<td>$\lambda &lt; 0$ and $\alpha = \beta = \mu = 0$. $\lambda = -n/2$, $\delta = \sqrt{n}$.</td>
</tr>
<tr>
<td>Cauchy(0,1)</td>
<td>$\lambda = -1/2$, $\alpha = \beta = 0$ and $\delta = 1$.</td>
</tr>
</tbody>
</table>

Tab. 2.1: Limiting cases of GH distributions

Fig. 2.14: Tail comparison between GH distributions, pdf (left) and approximation (right).

2.4 Tail-Behavior

In the final analysis, it is the heavy tail that makes GH distributions so popular in modelling the time dynamics of financial markets, since it is much closer to the empirical density of financial time series. Generally the GH distributions have an exponential decaying speed

$$f_{GH}(x; \lambda, \alpha, \beta, \delta, \mu = 0) \sim x^{\lambda-1} \exp\{- (\alpha - \beta) x\} \quad \text{as} \quad x \to \infty. \quad (2.26)$$

Figure 2.14 illustrates the tail behavior of GH distributions with different value of $\lambda$ with $\alpha = 1, \beta = 0, \delta = 1, \mu = 0$. The left panel contains part of pdf curves of GH distributions and the right panel demonstrates the approximation by the function mentioned above. It is clear that among the four distributions, GH with $\lambda = 1.5$ has the lowest decaying speed, while NIG decays fastest.

In Figure 2.15 the tail behavior of GH distributions and the limiting distributions is demonstrated. All distributions have mean 0 and variance 1 except for Student-t, since its variance $\sigma^2 = n/(n - 2)$, where $n$ is denoted as the degrees of freedom, and when $n \to \infty$, Student-t approaches normal distribution. The variance of t distribution in the graphic is 1.11 (from $n = 20$).
Fig. 2.15: Tail comparison between \( \text{GH}(-0.5,1,0,1,0) \) distribution (black) and its limiting distributions: normal(red), Student-t (blue) and Cauchy (green) distributions.

The green curve represents tail of Cauchy distribution, black that of GH distribution, blue Student-t and red normal distribution. Cauchy, GH and Student-t distributions decay more slowly than normal distribution.
3. METHODS OF ESTIMATION

3.1 Maximum-Likelihood Estimation

Assuming the independence of observations $x_i$, $i = 1, ..., n$, we maximize the log-likelihood function:

$$L_{GH}(\lambda, \alpha, \beta, \delta, \mu) = n \log \{a(\lambda, \alpha, \beta, \delta)\} + \left(\frac{\lambda}{2} - \frac{1}{4}\right) \sum_{i=1}^{n} \log \{\delta^2 + (x_i - \mu)^2\}$$

$$+ \sum_{i=1}^{n} \left[\log K_{\lambda-\frac{1}{2}}(\alpha \sqrt{\delta^2 + (x_i - \mu)^2} + \beta (x_i - \mu))\right] \quad (3.1)$$

For HYP ($\lambda = 1$) or NIG ($\lambda = -1/2$) distributions the algorithm uses the simpler expressions of the log-likelihood function.

Taking the first derivatives of the log-likelihood function respect to the five parameters, we obtain the following expressions, in which the log-likelihood function is denoted by $L$. See Prause (1999).

$$\frac{d}{d\lambda} L = n \left\{\frac{1}{2} \ln \left(\frac{\alpha^2 - \beta^2}{\alpha \delta}\right) - \frac{k_\lambda(\delta \sqrt{\alpha^2 - \beta^2})}{K_{\lambda}(\delta \sqrt{\alpha^2 - \beta^2})}\right\}$$

$$+ \sum_{i=1}^{n} \left\{\frac{1}{2} \ln \{\delta^2 + (x_i - \mu)^2\} + \frac{k_{\lambda-1/2}(\alpha \sqrt{\delta^2 + (x_i - \mu)^2})}{K_{\lambda-1/2}(\alpha \sqrt{\delta^2 + (x_i - \mu)^2})}\right\} \quad (3.2)$$

$$\frac{d}{d\alpha} L = n \frac{\delta \alpha}{\sqrt{\alpha^2 - \beta^2}} R_\lambda(\delta \sqrt{\alpha^2 - \beta^2})$$

$$- \sum_{i=1}^{n} \sqrt{\delta^2 + (x_i - \mu)^2} R_{\lambda-1/2}(\alpha \sqrt{\delta^2 + (x_i - \mu)^2}) \quad (3.3)$$
$$\frac{d}{d\beta}L = n\left\{ -\frac{\delta \beta}{\sqrt{\alpha^2 - \beta^2}} R_{\lambda}(\delta \sqrt{\alpha^2 - \beta^2}) - \mu \right\} + \sum_{i=1}^{n} x_i \quad (3.4)$$

$$\frac{d}{d\delta}L = n\left\{ -\frac{2\lambda}{\delta} + \sqrt{\alpha^2 - \beta^2} R_{\lambda}(\delta \sqrt{\alpha^2 - \beta^2}) \right\}$$

$$+ \sum_{i=1}^{n} \left\{ \frac{(2\lambda - 1)\delta}{\delta^2 + (x_i - \mu)^2} - \frac{\alpha \delta R_{\lambda}(\alpha \sqrt{\delta^2 + (x_i - \mu)^2})}{\sqrt{\delta^2 + (x_i - \mu)^2}} \right\} \quad (3.5)$$

$$\frac{d}{d\mu}L = -n\beta + \sum_{i=1}^{n} \frac{x_i - \mu}{\sqrt{\delta^2 + (x_i - \mu)^2}}$$

$$\times \left\{ \frac{2\lambda - 1}{\sqrt{\delta^2 + (x_i - \mu)^2}} - \alpha R_{\lambda-1/2}(\alpha \sqrt{\delta^2 + (x_i - \mu)^2}) \right\} \quad (3.6)$$

where

$$k_{\lambda}(x) = \frac{dK_{\lambda}(x)}{d\lambda}$$

$$R_{\lambda}(x) = \frac{K_{\lambda+1}(x)}{K_{\lambda}(x)}$$

Set them to zero, we obtain a complicated nonlinear equation system. Theoretically, there is a solution to a system with five equations and five unknown parameters. However, in practice, the solution is very difficult to be acquired.

Algorithms without using derivatives are therefore utilized to solve the problem of maximizing a function in five-dimensional space.

### 3.2 Numerical Algorithms

The algorithms of multi-dimensional maximization require algorithms of one-dimensional search for maximum value. The following algorithms are presented by Press, Teukolsky, Vetterling and Flannery (2002).

#### 3.2.1 Golden Section Search in One Dimension

The golden section or golden mean has its root back to the ancient Pythagoreans. The fraction of 0.38197 or 0.61803 is considered to have some aesthetic properties and it is rather helpful to search for the extremes.
In Figure 3.1 we have a triplet of points: \( \{a, f(a)\}, \{b, f(b)\} \) and \( \{c, f(c)\} \) with \( a < b < c \). Together they bracket a maximum. Among the ordinates, \( f(b) \) is then the largest one. Now we choose a new point \( x \), either between \( a \) and \( b \) or \( b \) and \( c \). Suppose we choose the latter. Then evaluating of \( f(x) \) is crucial for the choice of the next bracketing points. If \( f(x) < f(b) \), then the new bracketing triplet of points is \( {a, b, x} \); if \( f(x) > f(b) \), then the new bracketing triplet is \( {b, x, c} \). The principle is: the middle point of the new triplet must be the abscissa whose ordinate is the best maximum achieved so far. We continue the process of bracketing until the distance between the two outer points of the triplet is tolerably small. The optimal bracketing interval \( {a, b, c} \) has its middle point \( b \) a fractional distance 0.38197 from one end, and 0.61803 from the other end. This optimal method of function minimization is thus called **golden section search**.

Golden section search can be summarized as follows:

At each stage, given a bracketing triplet of points, the next point to be tried is that which is a fraction 0.38197 into the larger of the two intervals (measuring from the center point of the triplet). Even we are in a situation that the segments of starting bracketing triplet are not in the golden ratios, the procedure of choosing successive points at the golden mean point of the larger segment will quickly converge us to the proper ratios. By using golden section search, it is guaranteed that we will bracket the maximum to
an interval just 0.61803 times the size of the preceding interval.

Press et al. (2002) demonstrate why the golden section is the optimal fraction when searching for the maximum. Suppose that \( b \) is a fraction \( w \) of the way between \( a \) and \( c \), i.e.

\[
\frac{b - a}{c - a} = w \quad \frac{c - b}{c - a} = 1 - w.
\]  

(3.7)

Also suppose that our next trial point \( x \) is an additional fraction beyond \( b \),

\[
\frac{x - b}{c - a} = z.
\]  

(3.8)

Then the length of the next bracketing segment will either be \( w + z \) or \( 1 - w \), relative to the current one. In order to minimize the worst case possibility, we choose \( z \) to make these equal, namely

\[
z = 1 - 2w.
\]  

(3.9)

It is easy to find out that the equation [3.9] implies that the point \( x \) lies in the larger of the two segments, since \( z \) is positive only if \( w < 1/2 \). And the new point is symmetric point to \( b \) in the original interval, namely with \(|b - a|\) equal to \(|x - c|\).

Suppose we apply the same strategy at each stage, then if \( z \) is chosen to be optimal, so was \( w \) before it. This scale similarity implies that \( x \) should be the same fraction of the way from \( b \) to \( c \), if that is the larger segment, as was \( b \) from \( a \) to \( c \), in other words,

\[
\frac{z}{1 - w} = w.
\]  

(3.10)

Equation [3.9] and [3.10] give the quadratic equation

\[
w^2 - 3w + 1 = 0,
\]  

(3.11)

which yields:

\[
w = \frac{3 - \sqrt{5}}{2} \approx 0.38197
\]  

(3.12)
3. Methods of Estimation

3.2.2 Parabolic Interpolation and Brent’s Method in One Dimension

Another one-dimensional algorithm presented by Press et al. (2002) to search for a extreme is parabolic interpolation. Before we go on to this algorithm, let’s take a closer look at golden section search.

The introduction to golden section search and the deduction of golden mean makes it clear that golden section search is designed to deal with the worst case of function maximization. Although it is a sure process, it is very slow. If the function is nicely parabolic near to the maximum, then we do not need to slowly crawl through the whole function, the parabola fitted through any three points ought to take us in a single leap to the maximum, or at least very near to it (see Figure 3.2). The procedure is technically called inverse parabolic interpolation, since we want to find an abscissa rather than an ordinate.

The formula for the abscissa $x$ that is the maximum of a parabola through three points $\{a, f(a)\}, \{b, f(b)\}$ and $\{c, f(c)\}$ is:

$$x = b - \frac{1}{2} \frac{(b - a)^2\{f(b) - f(c)\} - (b - c)^2\{f(b) - f(a)\}}{(b - a)\{f(b) - f(c)\} - (b - c)\{f(b) - f(a)\}}. \quad (3.13)$$

Since this formula involves a fraction, it should be noticed that the formula
fails if the denominator is zero, which means the three points are collinear. However no maximization scheme that depends solely on \(3.13\) is likely to succeed. We should combine the strengths of two approaches in order to succeed. The exacting task is to invent a scheme that relies on a sure-but-slow technique, like golden section search mentioned above, when the function is not cooperative, but that switches over to a quick search, like parabolic interpolation introduced in this section, when the function allows. Brent’s method is implemented as the scheme. The general principles are: the parabolic step must fall within the bounding interval \((a, b)\), and imply a movement from the best current value \(x\) that is less than half the movement of the step before last. This second criterion insures that the parabolic steps are actually converging to something, rather than, for example, bouncing around in some nonconvergent limit cycle. Press et al. (2002) give an empirical reason for comparing to the step before last: Experience shows that it is better not to ”punish” the algorithm for a single bad step if it can make it up on the next one.

### 3.2.3 Powell’s Methods in Multidimensions

The fundament of multidimensional search for extreme is still the algorithms of one-dimension. Suppose we start at a point \(P\) in \(N\)-dimensional space, and proceed from there in some vector direction \(n\), then any function of \(N\) variables \(f(P)\) can be maximized along the line \(n\) by our one-dimensional methods. When the maximum is achieved along \(n\), a new direction is chosen, and the maximizing process is repeated along the new direction. The multidimensional maximization therefore consists of successive sequences of such line maximization. Different methods will differ only by how, at each stage, they choose the next direction \(n\) to try.

As demonstrated by Press et al. (2002) in their book, a good direction searching method requires that maximization along one direction is not undermined by the subsequent maximization along another. This kind of directions is called ”non-interfering” direction, or more conventionally conjugate directions.

Suppose that we are going to move along some new direction \(v\) after having moved along some direction \(u\) to a maximum. The condition that motion along \(v\) not spoil our maximization along \(u\) is just that the gradient stay perpendicular to \(u\). Two vectors \(u\) and \(v\) are said to be conjugate, when the relation mentioned above holds for them. And a set of vectors is called a conjugate set, when the relation holds pairwise for all members of the set of vectors. If we do successive line maximization of a function along a conjugate set of directions, then we don’t need to redo any of those directions.
Powell first discovered a direction set method that does produce $N$ mutually conjugate directions. The basic procedure is given as follows:

First, initialize the set of direction $\mathbf{u}_i$ to the basis vectors,

$$
\mathbf{u}_i = \mathbf{e}_i \quad i = 0, ..., N - 1.
$$

(3.14)

Now repeat the following sequence of steps until the function stops decreasing:

- Save the starting position as $\mathbf{P}_0$
- For $i = 0, ..., N - 1$, move $\mathbf{P}_i$ to the minimum along direction $\mathbf{u}_i$ and call this point $\mathbf{P}_{i+1}$.
- For $i = 0, ..., N - 2$, set $\mathbf{u}_i \leftarrow \mathbf{u}_{i+1}$.
- Set $\mathbf{u}_{N-1} \leftarrow \mathbf{P}_N - \mathbf{P}_0$.
- Move $\mathbf{P}_N$ to the minimum along direction $\mathbf{u}_{N-1}$ and call this point $\mathbf{P}_0$.

The basic procedure is given to search for a minimum, an analog of search for maximum.

Powell, in 1964, showed that, for a quadratic form like Taylor series, $k$ iterations of the above basic procedure produce a set of directions $\mathbf{u}_i$ whose last $k$ members are mutually conjugate. Therefore, $N$ iterations of basic procedure, amounting to $N(N + 1)$ line maximizations in all, will exactly minimize a quadratic form.

When we implement Powell’s quadratically convergent algorithm, a problem emerges. The procedure of throwing away, at each stage, $\mathbf{u}_0$ in favor of $\mathbf{P}_N - \mathbf{P}_0$ tend to produce sets of directions that fold up on each other and become linearly dependent. Once this happens, then the procedure finds the minimum of the function $f$ only over a subspace of the full $N$-dimensional case.

In order to fix up the problem of linear dependence in Powell’s algorithm, we implement a method which tries to find a few good directions instead of $N$ necessary conjugate directions.

The basic idea of our modified Powell’s method is still to take $\mathbf{P}_N - \mathbf{P}_0$ as a new direction; it is, after all, the average direction moved after trying all $N$ possible directions. The change is to discard the old direction along which the function $f$ made its largest decrease. This seems paradoxical, since that
direction was the best of the previous iteration. However, it is also likely to be a major component of the new direction that we are adding, so dropping it gives us the best chance of avoiding a buildup of linear dependence.

3.3 Local and Global Maximum

When maximizing a function, the problem of local and global maximum arises. If, for instance, a two-dimensional function forms a surface as illustrated in Figure 3.3. Besides the global maximum in the middle, there are many local maximums (identified as many small peaks in the graphic). Suppose we start our process from the point 2, and then probably we will reach the global maximum. It is the optimal case. If we start from point 3, we will probably achieve a local maximum near the global one. But that is not the worst case; the difference between the value of global maximum and local maximum marked in the graphic besides point 3 is not so large. The fall between the global maximum and the local maximum besides point 1 is quite large. If this is the surface of a likelihood function and we start from point 1, then we will get estimated parameters far from genuine ones.

Without knowledge of the structure of the probability density, it is rather difficult to optimize the choice of starting position. A partial solution will be choosing different starting points and comparing the resulting maximums. In the following chapter we will discuss on the robustness of starting position.
4. SIMULATION STUDIES

4.1 Procedure

We now utilize simulations to assess our results of estimation. The procedure is described as follows:

- Generate a set of artificial observations which is GH distributed. See rndgh.xpl.
- Estimate the parameters by the 5-parameter estimation methods: mlgh.xpl or mlghint.xpl. See 7.2.
- Repeat the first two steps for many times to obtain a batch of estimated parameters.
- Compare the estimated parameters with the original ones.

The details of implementation of estimation can be found in Appendix 7.1. Here, to be specific, we generate 2000 artificial observations and repeat the first two steps for 200 times to obtain a batch of estimated parameter with dimension $200 \times 5$.

Table 4.1 shows 15 results of estimation with the original parameters $(-1,1,0,1,0)$ (We use the parameter order $(\lambda, \alpha, \beta, \delta, \mu)$). The last column of the table records the log-likelihood value.

4.2 Estimation with Fractional $\lambda$

We start our assessment of estimation with original parameters $(-1,1,0,1,0)$. 200 repetitions provide us with results demonstrated in Table 4.1. The first row is the original parameters used to generate artificial data. The row labeled ”mean” records mean values of all 200 sets of estimated parameters. The last row gives standard deviations of the estimated parameters.

Table 4.1 indicates, that the means of the estimated parameters are very close to the original parameters. Figure 4.3 corroborates the conclusion.
The left panel of the graphic compares the original (blue) with estimated (blue) pdf curves, while the right panel concentrates on the left tail area of the distributions. In either panel, the red and blue curves overlap, indicating that the results of estimation are truly satisfactory.

However, Figure 4.1 shows that there are many outliers for estimated $\lambda$ and $\alpha$. A closer look at Table 4.1 reveals more information. Among the 15 examples of estimated parameters, 6th and 11th are striking. The estimated parameters deviate relatively far from the mean. From the values of the table, it seems that their pdf curves have little chance to be adjacent to the original one, but the flexibility of GH distributions allows the effects of one parameter to be compensated by other parameters. The results are illustrated in Figure 4.4 and Figure 4.5. The graphics show that their pdf curves almost overlap with the original ones. Their tails seem to be far from the original ones in the graphics, but the scale of y-axis clarifies that they are actually very close to the original tails. The results are helpful to explain the standard deviations in Table 4.1 and the many outliers of $\lambda$ and $\alpha$ in Figure 4.1.

Now we gradually increase the value of $\lambda$ and measure the results of estimation. Table 4.2 lists the results when $\lambda$ takes four different values, and
Figure 4.6 - 4.9 compare their estimated pdf curves with the original ones. When \( \lambda \) takes the value of -0.5 or 1, GH distribution is identified as its two subclasses. See Chapter 2.3.

Either the table or the figures indicate, that the results of estimation are desirable, when original \( \lambda \) takes the value of -3, -1, -0.5, 0.49 and 1. When original \( \lambda = -3 \), all parameters fit quite well. The largest deviation comes from \( \lambda \). There is less than 0.11 difference between the original and estimated \( \lambda \). Figure 4.2 demonstrates two overlapped curves, implying a perfect match, and the right panel shows that the difference between two tails is measured by the magnitude of \( 10^{-4} \). When original \( \lambda = -0.5 \) or \( \lambda = 0.49 \), either the values of estimated parameters or the pdf curves are very close to the original ones. In these cases, there is nothing to carp at. For the original \( \lambda = 1 \), estimated \( \lambda \) deviates a little from 1, while other parameters are still very close to the original ones. As illustrated by Figure 4.8, a small gap appears between the red and blue curves, indicating a deficient value of estimated \( \lambda \).

When original \( \lambda \) takes a large positive value, in our case, \( \lambda = 3 \), the estimated and original pdf curves separate, the estimated pdf curve is more peaked in the middle than the original curve. The estimated value of \( \lambda \) is much smaller. It has a value around 1.03, only a third of the original one. Although the value of \( \delta \) increases and takes a value more than 2.2, it is not large enough to offset the effects of \( \lambda \). A change of estimation order of parameters does not improve the results.

The influences of \( \alpha \) on the results of estimation can also be evaluated in the same way. Here we choose original \( \lambda = -0.5 \) and \( \lambda = 1 \), so that our results
are more comparable with other literature. We increase the value of $\alpha$ to 2 and then to a large one, 4.5. Table 4.3-4.4 and Figure 4.10-4.13 illustrate the results.

Table 4.3 shows rather large deviation of $\lambda$ and $\alpha$, while other parameters are well estimated. When original $\alpha = 2$, estimated $\alpha$ is a little smaller than the original one, and estimated $\lambda$ take a much smaller negative value than -0.5. But the results are satisfactory. As displayed in Figure 4.10, most parts of the curves overlaps, and the only gap at the peak is also quite small. The two tails are very close to each other. The effects of deviations offset each other, providing us a nice result. When original $\alpha = 4.5$, estimated $\lambda$ goes even deeper into a small negative value, far from the original, while estimated $\alpha$ takes a value only more than a half of 4.5. The combined results are illustrated in Figure 4.11. The gap at the peak is larger than that with $\alpha = 2$ and the vertical difference between tails is larger than 0.005.

Table 4.4 displays a similar way of deviation as Table 4.3. The performance of estimation is worse in the case of $\lambda = 1$ than in the case of $\lambda = -0.5$, as illustrated by Figure 4.12 and Figure 4.13. However the differences of tails are relatively small. They are measured with a magnitude of $10^{-3}$.

In Table 4.5, we use a positive $\beta$ to examine the performance of estimation of a skewed GH distribution. 200 repetitions provide us with perfect results of estimation when original $\lambda = -0.5$. The estimated parameters are very close to the original ones. Figure 4.14 verifies this with two overlapped curves. The results are not so perfect when $\lambda = 1$, as Figure 4.15 indicates. There are gaps between the estimated and original curves, although the left tails are still very close to each other.

From the simulation studies, it can be concluded that the estimation of GH distributions works very well, when the parameters of GH distributions are not very large. However, when the original parameters, particularly $\lambda$ and $\alpha$, take large positive values, the performances of estimation are not so desirable.
### Tab. 4.2: Results of estimation with fractional $\lambda$, original parameters ($\lambda, 1, 0, 1, 0$).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>mean($\lambda = -3$)</td>
<td>-2.894749</td>
<td>1.067013</td>
<td>-0.026335</td>
<td>0.989045</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.684444</td>
<td>0.713504</td>
<td>0.143434</td>
<td>0.110281</td>
</tr>
<tr>
<td>mean($\lambda = -0.5$)</td>
<td>-0.480997</td>
<td>1.013325</td>
<td>-0.006658</td>
<td>0.991011</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.505728</td>
<td>0.217238</td>
<td>0.041726</td>
<td>0.207311</td>
</tr>
<tr>
<td>mean($\lambda = 0.49$)</td>
<td>0.546592</td>
<td>0.977767</td>
<td>-0.004014</td>
<td>0.943336</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.351150</td>
<td>0.096753</td>
<td>0.032694</td>
<td>0.265398</td>
</tr>
<tr>
<td>mean($\lambda = 0$)</td>
<td>0.421271</td>
<td>0.976144</td>
<td>-0.002571</td>
<td>0.988370</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.561137</td>
<td>0.121657</td>
<td>0.043518</td>
<td>0.542232</td>
</tr>
</tbody>
</table>

### Tab. 4.3: Results of estimation with fractional $\lambda$, original parameters (-0.5, $\alpha$, 0, 1, 0).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>-0.5</td>
<td>-</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>mean($\alpha = 2$)</td>
<td>-1.330474</td>
<td>1.585631</td>
<td>-0.012431</td>
<td>1.147776</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.905404</td>
<td>0.429256</td>
<td>0.081769</td>
<td>0.263837</td>
</tr>
<tr>
<td>mean($\alpha = 4.5$)</td>
<td>-3.878362</td>
<td>2.434933</td>
<td>-0.007591</td>
<td>1.173874</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.745779</td>
<td>0.951038</td>
<td>0.257702</td>
<td>0.173874</td>
</tr>
</tbody>
</table>

### Tab. 4.4: Results of estimation with fractional $\lambda$, original parameters (1, $\alpha$, 0, 1, 0).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>mean($\alpha = 2$)</td>
<td>-1.171250</td>
<td>1.435458</td>
<td>-0.002609</td>
<td>1.365879</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.547099</td>
<td>0.262497</td>
<td>0.080305</td>
<td>0.194302</td>
</tr>
<tr>
<td>mean($\alpha = 4.5$)</td>
<td>-3.960707</td>
<td>3.009104</td>
<td>-0.001979</td>
<td>1.343088</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.660589</td>
<td>0.931344</td>
<td>0.277702</td>
<td>0.169194</td>
</tr>
</tbody>
</table>

### Tab. 4.5: Results of estimation with fractional $\lambda$, original parameters ($\lambda, 1.5, 0.5, 1, 0$).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>-</td>
<td>1.5</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>mean($\lambda = -0.5$)</td>
<td>-0.642239</td>
<td>1.455295</td>
<td>0.504557</td>
<td>1.038165</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.548510</td>
<td>0.245575</td>
<td>0.083440</td>
<td>0.188165</td>
</tr>
<tr>
<td>mean($\lambda = 1$)</td>
<td>0.181915</td>
<td>1.377126</td>
<td>0.510125</td>
<td>1.116492</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.514431</td>
<td>0.157081</td>
<td>0.080562</td>
<td>0.285200</td>
</tr>
</tbody>
</table>
4. Simulation Studies

Fig. 4.2: Comparison between original (blue) and estimated (red) pdf curves with parameters (-3,1,0,1,0) and fractional $\lambda$.

Fig. 4.3: Comparison between original (blue) and estimated (red) pdf curves with parameters (-1,1,0,1,0) and fractional $\lambda$. 
Fig. 4.4: Comparison between original (blue) and 6th estimated (red) pdf curves with parameters (-1,1,0,1,0) and fractional $\lambda$.

Fig. 4.5: Comparison between original (blue) and 11th estimated (red) pdf curves with parameters (-1,1,0,1,0) and fractional $\lambda$. 
Fig. 4.6: Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1,0,1,0) and fractional $\lambda$.  

Fig. 4.7: Comparison between original (blue) and estimated (red) pdf curves with parameters (0.49,1,0,1,0) and fractional $\lambda$.  

---

4. Simulation Studies

---
Fig. 4.8: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,1,0,1,0) and fractional $\lambda$.

Fig. 4.9: Comparison between original (blue) and estimated (red) pdf curves with parameters (3,1,0,1,0) and fractional $\lambda$. 
Fig. 4.10: Comparison between original (blue) and estimated (red) pdf curves with parameters \((-0.5,2,0,1,0)\) and fractional \(\lambda\).

Fig. 4.11: Comparison between original (blue) and estimated (red) pdf curves with parameters \((-0.5,4.5,0,1,0)\) and fractional \(\lambda\).
Fig. 4.12: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,2,0,1,0) and fractional $\lambda$.

Fig. 4.13: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,4.5,0,1,0) and fractional $\lambda$. 
Fig. 4.14: Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1.5,0.5,1,0) and fractional $\lambda$.

Fig. 4.15: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,1.5,0.5,1,0) and fractional $\lambda$. 

\(\text{simcmp10.xpl}\)

\(\text{simcmp11.xpl}\)
4.3 Estimation with integer $\lambda$

Estimation of GH distributions is time consuming. The main reason is the complexity to compute modified Bessel function of fractional order. If we use a modified Bessel function of integer order, the costs of estimation will be substantially reduced. This means that we use an integer $\lambda$ to replace the real $\lambda$ which we used above. The details of implementation can be found in Appendix 7.1.

The assessment of estimation with integer $\lambda$ uses the same batch of original parameters as in the last section.

Table 4.6 and Figure 4.16 - 4.21 demonstrate the results of estimation with different original $\lambda$s. One feature of Table 4.6 is really striking: except the case when original $\lambda = 3$ the estimated $\lambda$s all take the value of 1. In the only exception, the estimated $\lambda = 1.92$, the mean value of 16 $\lambda$s which take the value of 1 and 184 $\lambda$s which take the value of 2 among the 200 simulations. On the other hand, estimated $\alpha$s and $\delta$s deviate far from the original values.

However, the total results are quite desirable when original $\lambda$ takes the value of -3, -1, -0.5, 0.49 and 1, as illustrated in Figure 4.16 - 4.20. In these cases, most parts of the estimated and original pdf curves overlap and their magnified tail parts are also very close to each other. The tiny gaps at the peaks and relatively larger distance between estimated and original tails in the Figure 4.16 - 4.19 indicate that estimation with integer $\lambda$ performs slightly worse than that of fractional $\lambda$. When original $\lambda = 1$, the gap at the peak occurs as before. From the estimated parameters, we can find out that $\delta$ is a little smaller than the original value while other parameters are well estimated. The gap is created solely by the deficient $\delta$ estimates. When $\lambda$ takes a large positive value, as in Figure 4.21 $\lambda = 3$, the results are not acceptable as well.

Table 4.7 - 4.9 and Figure 4.22 - 4.27 show the outcome of estimation with different original $\alpha$s and positive $\beta$. The tables show that while $\lambda$ always takes the value of 1, estimated $\alpha$s and $\delta$s deviate from the original values to make the estimated pdf curves adjacent to the original ones.

For original parameter $\lambda = -0.5$, we get satisfactory results for $\alpha = 2$ and $\alpha = 4.5$, while the performance is not so desirable for the case $\lambda = 1$. When original $\beta$ takes a positive value, estimated $\beta$s are very close to the original one. Figure 4.22 - 4.27 illustrates, that the outcome we obtain about $\alpha$ and $\beta$ by estimation with integer $\lambda$ is similar to that with fractional $\lambda$.

Generally speaking, the estimation with fractional $\lambda$ performs better than that with integer $\lambda$, but the latter takes much less time to compute.
### Tab. 4.6: Results of estimation with integer $\lambda$, original parameters ($\lambda, 1, 0, 1, 0$).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mean($\lambda = -3$)</td>
<td>1.000000</td>
<td>4.019720</td>
<td>-0.002461</td>
<td>0.493896</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.318593</td>
<td>0.159888</td>
<td>0.086667</td>
</tr>
<tr>
<td>mean($\lambda = -1$)</td>
<td>1.000000</td>
<td>1.930011</td>
<td>-0.006522</td>
<td>0.418007</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.090105</td>
<td>0.066943</td>
<td>0.084313</td>
</tr>
<tr>
<td>mean($\lambda = -0.5$)</td>
<td>1.000000</td>
<td>1.589057</td>
<td>-0.002052</td>
<td>0.452279</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.072576</td>
<td>0.033982</td>
<td>0.092661</td>
</tr>
<tr>
<td>mean($\lambda = 0.49$)</td>
<td>1.000000</td>
<td>1.109968</td>
<td>0.000763</td>
<td>0.681548</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.054870</td>
<td>0.033982</td>
<td>0.154060</td>
</tr>
<tr>
<td>mean($\lambda = 1$)</td>
<td>1.000000</td>
<td>1.238800</td>
<td>-0.002404</td>
<td>1.797543</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.271974</td>
<td>0.098074</td>
<td>0.042012</td>
<td>0.549621</td>
</tr>
</tbody>
</table>

### Tab. 4.7: Results of estimation with integer $\lambda$, original parameters (-0.5, $\alpha$, 0, 1, 0).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>-0.5</td>
<td>-</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>mean($\alpha = 2$)</td>
<td>1.000000</td>
<td>2.589086</td>
<td>0.001074</td>
<td>0.665730</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.186796</td>
<td>0.092427</td>
<td>0.116603</td>
</tr>
<tr>
<td>mean($\alpha = 4.5$)</td>
<td>1.000000</td>
<td>5.322449</td>
<td>0.015300</td>
<td>0.711570</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.587208</td>
<td>0.254687</td>
<td>0.137863</td>
</tr>
</tbody>
</table>

### Tab. 4.8: Results of estimation with integer $\lambda$, original parameters (1, $\alpha$, 0, 1, 0).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>mean($\alpha = 2$)</td>
<td>1.000000</td>
<td>2.213086</td>
<td>-0.001887</td>
<td>0.840281</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.152440</td>
<td>0.082170</td>
<td>0.141899</td>
</tr>
<tr>
<td>mean($\alpha = 4.5$)</td>
<td>1.000000</td>
<td>5.615335</td>
<td>0.003632</td>
<td>0.927466</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.796738</td>
<td>0.281883</td>
<td>0.199529</td>
</tr>
</tbody>
</table>

### Tab. 4.9: Results of estimation with integer $\lambda$, original parameters ($\lambda, 1.5, 0.5, 1, 0$).

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>-</td>
<td>1.5</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>mean($\lambda = -0.5$)</td>
<td>1.000000</td>
<td>2.076111</td>
<td>0.527648</td>
<td>0.562929</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.144388</td>
<td>0.100483</td>
<td>0.106043</td>
</tr>
<tr>
<td>mean($\lambda = 1$)</td>
<td>1.000000</td>
<td>1.597358</td>
<td>0.521914</td>
<td>0.798836</td>
</tr>
<tr>
<td>s.d.</td>
<td>0.000000</td>
<td>0.117911</td>
<td>0.084639</td>
<td>0.141009</td>
</tr>
</tbody>
</table>
4. Simulation Studies

Fig. 4.16: Comparison between original (blue) and estimated (red) pdf curves with parameters \((-1,1,0,1,0)\) and integer \(\lambda\).

Fig. 4.17: Comparison between original (blue) and estimated (red) pdf curves with parameters \((-1,1,0,1,0)\) and integer \(\lambda\).
Fig. 4.18: Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1,0,1,0) and integer $\lambda$.

Fig. 4.19: Comparison between original (blue) and estimated (red) pdf curves with parameters (0.49,1,0,1,0) and integer $\lambda$. 
Fig. 4.20: Comparison between original (blue) and estimated (red) pdf curves with parameters $(1,1,0,1,0)$ and integer $\lambda$.

Fig. 4.21: Comparison between original (blue) and estimated (red) pdf curves with parameters $(3,1,0,1,0)$ and integer $\lambda$. 
Fig. 4.22: Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,2,0,1,0) and integer $\lambda$.

Fig. 4.23: Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,4.5,0,1,0) and integer $\lambda$. 
Fig. 4.24: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,2,0,1,0) and integer $\lambda$.

Fig. 4.25: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,4.5,0,1,0) and integer $\lambda$. 
Fig. 4.26: Comparison between original (blue) and estimated (red) pdf curves with parameters (-0.5,1.5,0.5,1,0) and integer \( \lambda \).

Fig. 4.27: Comparison between original (blue) and estimated (red) pdf curves with parameters (1,1.5,0.5,1,0) and integer \( \lambda \).
5. APPLICATION TO FINANCIAL MARKET

5.1 Data Description

As the first step into application to financial market, we choose three sets of data from http://www.quantlet.org/mdbase: BMW stock price, Thyssen stock price and foreign exchange rate.

The dataset of BMW stock price (BMW) contains opening prices, highest prices, lowest prices and closing prices from Jan.02,1990 to Dec.30,1992 in Federal Republic of Germany. Including the date, we have a $747 \times 5$ dataset.

Similarly, the dataset of Thyssen stock price (THY) contains opening prices, highest prices, lowest prices and closing prices from Jan.02,1990 to Dec.30,1992 in Federal Republic of Germany. Together with the date, we have also a $747 \times 5$ dataset. We use closing prices to calculate returns.

The data of foreign exchange rate studied here contain

- foreign exchange rate German Mark to US Dollar (DMUSD)
- foreign exchange rate British Pound to US Dollar (BPUSD)

They cover daily observations from Dec.01,1979 to Apr.01,1994 in Federal Republic of Germany. The dimension of the data is therefore $3720 \times 2$.

5.2 Data Transformation

As presented before, an important empirical fact in financial market is that returns of financial assets are often heavy-tail distributed. We assume the return is GH distributed and estimate the five parameters.

In order to get a stationary process, log-returns of exchange rates are used. If we denote $r_t$ as log-return at time $t$,
\begin{equation}
 r_t = \log(p_t) - \log(p_{t-1}) = \log\left(\frac{p_t}{p_{t-1}}\right) \tag{5.1}
\end{equation}
where $p_t$ is the observed price of financial asset or exchange rate of currency at time $t$. In the case of exchange rate, $r_t$ represents the logarithm of the financial return at time $t$ of holding a unit of the currency.

The KPSS test from Kwiatkowski, Philipps, Schmidt and Shin is then employed to test for stationarity. The regression model with a time trend has the form

$$X_t = c + \mu t + k \sum_{i=1}^{t} \xi_i + \eta_t$$

(5.2)

with stationary $\eta_t$ and i.i.d. $\xi_t$ with an expected value 0 and variance 1. For $k = 0$ the process is trend stationary, while it is an integrated process for $k \neq 0$. The null hypothesis is $H_0 : k = 0$, and the alternative hypothesis is $H_1 : k \neq 0$.

The KPSS test statistic is

$$KPSS_T = \frac{\sum_{t=1}^{n} S_t^2}{n^2 \hat{\omega}_T^2},$$

(5.3)

with

$$S_t = \sum_{i=1}^{t} \hat{\eta}_i,$$

$$\hat{\omega}_T^2 = \hat{\sigma}_\eta^2 + 2 \sum_{\tau=1}^{T} \left(1 - \frac{\tau}{T-1}\right) \hat{\gamma}_\tau,$$

where $\hat{\sigma}_\eta^2$ is the variance estimator of $\eta_t$ and $\hat{\gamma}_\tau = 1/n \sum_{t=\tau+1}^{n} \hat{\eta}_t \hat{\eta}_{t-\tau}$ is the covariance estimator.

The results of Table 5.1 indicates that the stock prices and the exchange rates are not stationary or trend stationary, since in every case the null hypothesis at a significance level of 1% is rejected. On the other hand, as Table 5.2 exhibits, all the log-returns are stationary even at the significance level of 10%. When tested with time trend, the null hypothesis is accepted at the level of 10% for the case of BMW and THY, while the process of log-return of DMUSD and BPUSD is trend stationary at the level of 1% and 5% respectively.

Now we have a stationary process, a further transformation leads us to the problem of volatility. As indicated by Franke, Härdle and Hafner (2004), volatility plays an important role in modeling financial systems and time series. Although the volatility is not observable, it can be estimated from
the data. The problem is to find an appropriate model for volatility. ARCH models are the most important class of models in this area.

ARCH models (autoregressive conditional heteroscedasticity) can efficiently model the typical empirical findings in financial time series, the conditional heteroscedasticity. After the collapse of the currency system in Bretton Woods and the following time period of flexible exchange rates in the seventies, such models were increasingly necessary for researchers and practitioners. The ARCH model can be generalized by extending it with autoregressive terms of the volatility. The resulting model is called Generalized ARCH model or GARCH model, which is appropriate for modeling returns of financial assets.

The process \( (\varepsilon_t), t \in \mathbb{Z} \), is GARCH\((p,q)\), if \( E[\varepsilon_t | F_{t-1}] = 0 \),

\[
\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_i \sigma_{t-j}^2, \tag{5.4}
\]
We assume that the log-returns follow GARCH(1,1) process. With the help of `garchest.xpl`, the processes can be estimated. The parameter estimates are listed in Table 5.3.

Together with parameters, the volatilities of the processes are also estimated by `garchest.xpl`. To rule out the influence of volatility, the log-returns are divided by volatility estimates, which means $Z_t = \varepsilon_t / \sigma_t$ is assumed to be GH distributed.

### 5.3 Estimation

Table 5.4 and Figure 5.1 - 5.2 demonstrate the results of GH distribution estimation with fractional $\lambda$. The left panel of Figure 5.1 illustrates the results of the case BMW, while the right panel shows the results of the case THY. Similarly, the left and right panel of Figure 5.2 display the results of the case DMUSD and of the case BPUSD respectively. The red curves in the graphics represent the pdf curves of estimated GH distributions. The blue curves show the outcome of kernel density estimation (a nonparametric method), which is utilized here as a comparison. From the graphic, we find that two methods of estimation give rather similar pdf curves in all cases. However, from the gaps between the red and blue curves we conclude that the results are more desirable in the cases of DMUSD and BPUSD than in the cases of BMW and THY. It is probably because the former have larger sample size than the latter.

Table 5.5 and Figure 5.3 - 5.4 display the results of GH distribution estimation with integer $\lambda$. Again, we have more satisfactory results in DMUSD and BPUSD than in BMW and THY. Either in the left or in the right panel of Figure 5.4, the red curve and blue curve almost overlap, while there are relatively larger gaps around the peak areas in Figure 5.3. The comparison between Table 5.4 and 5.5 reveals that the log-likelihood values of fractional and integer cases are very close to each other. The estimation with fractional $\lambda$ performs slightly better than that with integer $\lambda$ in all cases, since the former has a higher log-likelihood values.
Tab. 5.4: Parameter estimates of GH distributions with fractional $\lambda$

<table>
<thead>
<tr>
<th></th>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
<th>LogLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW</td>
<td>-1.454790</td>
<td>0.441041</td>
<td>-0.030650</td>
<td>1.199405</td>
<td>0.009788</td>
<td>-994.275246</td>
</tr>
<tr>
<td>THY</td>
<td>-2.051983</td>
<td>0.149017</td>
<td>-0.116073</td>
<td>1.433777</td>
<td>0.065313</td>
<td>-991.483863</td>
</tr>
<tr>
<td>DMUSD</td>
<td>1.628804</td>
<td>1.796164</td>
<td>-0.024493</td>
<td>0.000000</td>
<td>0.032792</td>
<td>-5166.568932</td>
</tr>
<tr>
<td>BPUSD</td>
<td>1.711117</td>
<td>1.846472</td>
<td>0.079376</td>
<td>0.000000</td>
<td>-0.060799</td>
<td>-5172.000888</td>
</tr>
</tbody>
</table>

Tab. 5.5: Parameter estimates of GH distributions with integer $\lambda$

<table>
<thead>
<tr>
<th></th>
<th>$\lambda$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\delta$</th>
<th>$\mu$</th>
<th>LogLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW</td>
<td>1.000000</td>
<td>1.519522</td>
<td>-0.019417</td>
<td>0.236442</td>
<td>-0.002284</td>
<td>-996.149029</td>
</tr>
<tr>
<td>THY</td>
<td>1.000000</td>
<td>1.591577</td>
<td>-0.079240</td>
<td>0.354969</td>
<td>0.027018</td>
<td>-996.386270</td>
</tr>
<tr>
<td>DMUSD</td>
<td>1.000000</td>
<td>1.613605</td>
<td>-0.025404</td>
<td>0.564662</td>
<td>0.033675</td>
<td>-5168.951861</td>
</tr>
<tr>
<td>BPUSD</td>
<td>1.000000</td>
<td>1.646890</td>
<td>0.074733</td>
<td>0.618643</td>
<td>-0.056103</td>
<td>-5174.957993</td>
</tr>
</tbody>
</table>

Fig. 5.1: Comparison between kernel density estimation (blue) and GH estimation with fractional $\lambda$ (red). Left - BMW; Right - THY
Fig. 5.2: Comparison between kernel density estimation (blue) and GH estimation with fractional $\lambda$ (red). Left - DMUSD; Right - BPUSD

Fig. 5.3: Comparison between kernel density estimation (blue) and GH estimation with integer $\lambda$ (red). Left - BMW; Right - THY
Fig. 5.4: Comparison between kernel density estimation (blue) and GH estimation with integer $\lambda$ (red). Left - DMUSD; Right - BPUSD. 

\texttt{afmint.xpl}
6. SUMMARY AND OUTLOOKS

This thesis focuses on a particular heavy-tailed distribution: GH distributions. We have looked through the major features of GH distributions: parameters, moment generating function, characteristic function and so on. Some of the subclasses and limiting distributions of GH distributions have also been examined.

The emphasis of the thesis is using maximum likelihood estimation to estimate the five parameters of GH distributions. Several numerical algorithms of searching for extreme, including Golden Section search and parabolic interpolation, are introduced and Powell’s methods in multidimensions, which are based on the algorithms introduced, are utilized in our case.

The outcome of the estimation is assessed by simulation studies. Different original parameter sets are employed to measure the performance of estimation. The results of estimation with fractional $\lambda$ and its simplified version, estimation with integer $\lambda$, are presented. Estimation with fractional $\lambda$ performs better, while estimation with integer $\lambda$ takes much less time to process.

When the estimation is exercised in the financial environment with real data, kernel density estimation is used as comparison. The results of application are rather desirable, since in every case KDE curve is very similar to GH estimated curve.

While being exceedingly well when the original parameters of GH distributions are not very large, the performances of estimation are not so desirable if the original parameters take large positive values. A useful extension of the current work would therefore involve closer examination of the reasons behind the defection of the GH estimation and possible solutions to it. Another extension would incorporate a scheme which improve the performance of estimation with integer $\lambda$ or reduce the costs of estimation with fractional $\lambda$. 
7. APPENDIX

7.1 C Codes

7.1.1 mlgh.c

Return the five parameter estimates of GH distribution and log-likelihood value with fractional $\lambda$.

***Some of the functions used here are from Prause (1999) and the book Numerical Recipes in C++ written by Press et al. (2002).***

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PI 3.141592653589793116
#define ACC 40.0
#define TINY 1.0e-20
#define ITMAX 3000
#define MAXIT 10000

#define TOL 2.0e-4
#define NR_END 1
#define FREE_ARG char *
#define CGOLD 0.3819660
#define ZEPS 1.0e-10
#define SHFT (a,b,c,d) (a)=(b);(b)=(c);(c)=(d);
#define SIGN (a,b) ((b) >= 0.0 ? fabs (a) : -fabs (a))
#define GOLD 1.618034
#define GLIMIT 100.0
#define EPS 1.0e-10
#define XMIN 2.0
#define FPMIN 1.0e-30

static double maxarg1,maxarg2;
#define FMAX(a,b) (maxarg1=(a),maxarg2=(b),(maxarg1) > (maxarg2) ?\  (maxarg1) : (maxarg2))

static double sqrarg;
```
#define SQR(a) \{(sqrarg=(a)) == 0.0 ? 0.0 : sqrarg*sqrarg\)

void mlgh(double *lambda, double *alpha, double *beta, double *delta, double *mu, double *ml);
double logligh(double *p);

void bessik(double x, double xnu, double *ri, double *rk, double *rip, double *rkp);
void beschb(double x, double *gam1, double *gam2, double *gampl, double *gammi);
double chebev(double a, double b, double c[], int m, double x);

double *vector(long nl, long nh);
FILE *DATA;
double *X;
int H;

void main()
{
    int n,i;
    double *r, lambda, alpha, beta, delta, mu, ml;
    r = readdata(&n);
    H = n;
    X = malloc(sizeof(double)*(n+1));
    for (i = 1; i <= H; i++) *(X+i) = *(r+i);
    mlgh(&lambda, &alpha, &beta, &delta, &mu, &ml);
    printf("%9.16f\n%9.16f\n%9.16f\n%9.16f\n%9.16f\n%9.16f\n%9.16f\n%9.16f\n%9.16f\n%9.16f\n\n", lambda, alpha, beta, delta, mu, ml);
}

double *readdata(int *n)
{
    int i;
    double *r, tmp;
    DATA = fopen("data.txt", "r");
    fscanf(DATA, "%lf", &tmp);
7. Appendix

71

```c
75    *n = (int)tmp;
76    r = malloc(sizeof(double)*(*(n)+1));
77    *r = 0.0;
78    for ( i = 1 ; i <= (*n) ; i ++ ) {
79        fscanf(DATA, "%lf", &tmp);
80        *(r+i) = tmp;
81    }
82    fclose(DATA);
83    return r;
84 }
85
86
87    void mlgh(double *lambda, double *alpha, double *beta, double *delta, double *mu, double *ml)
88    {
89        int iter;
90        long i;
91        double *p, **xi, ftol=1.0e-6, fret, psi, chi, t1, t2, ri,
92                      rk, rip, tmp, dif, a1, a2;
93        p = vector(1,5);
94        xi = matrix(1,5,1,5);
95        p[1] = 0.0; p[2] = 0.0; p[3] = 0.0; p[4] = 0.0; p[5] = 0.0;
96        xi[1][1] = 1.0; xi[1][2] = 0.0; xi[1][3] = 0.0; xi[1][4] = 0.0; xi[1][5] = 0.0;
97        xi[2][1] = 0.0; xi[2][2] = 1.0; xi[2][3] = 0.0; xi[2][4] = 0.0; xi[2][5] = 0.0;
98        xi[3][1] = 0.0; xi[3][2] = 0.0; xi[3][3] = 1.0; xi[3][4] = 0.0; xi[3][5] = 0.0;
99        xi[4][1] = 0.0; xi[4][2] = 0.0; xi[4][3] = 0.0; xi[4][4] = 1.0; xi[4][5] = 0.0;
100       xi[5][1] = 0.0; xi[5][2] = 0.0; xi[5][3] = 0.0; xi[5][4] = 0.0; xi[5][5] = 1.0;
101    powell(p, xi, 5, ftol, &iter, &fret, &logligh);
102        *lambda = p[1];
103        *alpha = sqrt(p[3]*p[3]+exp(p[2]));
104        *beta = p[3];
105        *delta = exp(0.5*p[4]);
106        *mu = p[5];
107        psi = p[2];
108        chi = p[4];
109        t1=0;
110        t2=0;
111    for ( i = 1 ; i <= H ; i ++ ){
112        dif = *(X+i)-*mu;
113        a1 = exp(chi)+dif*dif;
114        t1 = t1+log(a1);
115        a2 = sqrt(((1.0+delta)*exp(psi))*a1);
116    }
```

bessik(a2, fabs(*lambda-0.5), &ri, &rk, &rip, &rkp);
t2 = t2+log(rk)+*beta*dif;
}
tmp = exp(0.5*(psi+chi));
bessik(tmp, fabs(*lambda), &ri, &rk, &rip, &rkp);
tmp = log(rk);
tmp = 0.5*(lambda)*psi-0.5*log(2*PI)-(0.5*(lambda)
-0.25)*log(*beta*(beta)+exp(psi))-0.5*(lambda)*chi-tmp
;
tmp = H*tmp+(0.5*(lambda)-0.25)*t1+t2;
*ml = tmp;
free_vector(p,1,5);
free_matrix(xi,1,5,1,5);
}

double logligh(double p[])
{
    long i;
double a1, a2, dif, lambda, psi, beta, chi, mu, t1, t2,
tmp, ri, rk, rip, rkp;
lambda = p[1];
psi = p[2];
beta = p[3];
chi = p[4];
mu = p[5];
t1=0;
t2=0;
for ( i = 1 ; i <= H ; i ++ ){
    dif = *(X+i)-mu;
a1 = exp(chi)+dif*dif;
t1 = t1+log(a1);
a2 = sqrt((beta*beta+exp(psi))*a1);
bessik(a2, fabs(lambda-0.5), &ri, &rk, &rip, &rkp);
t2 = t2+log(rk)+beta*dif;
}
tmp = exp(0.5*(psi+chi));
bessik(tmp, fabs(lambda), &ri, &rk, &rip, &rkp);
tmp = log(rk);
tmp = 0.5*lambda*psi-0.5*log(2*PI)-(0.5*lambda-0.25)*log(
    beta*beta+exp(psi))-0.5*lambda*chi-tmp;
tmp = H*tmp+(0.5*lambda-0.25)*t1+t2;
tmp = -tmp;
return tmp;
}

void bessik (double x, double xnu, double *ri, double *rk,
double *rip, double *rkp)
{
    double a1,b,c,d,del,del1,delb,dels,e,f,fact,fact2,ff,
    gam1,gam2,gammi,gampl,h,p,pimu,q,q1,q2,qnew,riel,
ril1,rimu,ripl,ripl1,ritemp,rk1,rkmu,rktemp,s,
    sum,sum1,x2,xi,xi2,xmu,xmu2;
int i,l,nl;

if (x <= 0.0 || xnu < 0.0) {
    printf("bad arguments in bessik");
    exit(EXIT_FAILURE);
}

nl = (int)(xnu+0.5);
xmu = xnu-nl;
xmu2 = xmu*xmu;
xi = 1.0/x;
xi2 = 2.0*xi;
h = xnu*xi;
if (h < FPMIN) h = FPMIN;
b = xi2*xnu;
d = 0.0;
c = h;
for (i=0;i<MAXIT;i++) {
    b += xi2;
d = 1.0/(b+d);
c = b+1.0/c;
del = c*d;
h = del*h;
    if (fabs(del-1.0) <= EPS) break;
}
if (i >= MAXIT) {
    printf("x too large in bessik; try asymptotic 
        expansion");
    exit(EXIT_FAILURE);
}

ril = FPMIN;
ripl = h*ril;
ril1 = ril;
ripl1 = ripl;
fact = xnu*xi;
for (l = nl-1; l >= 0; l--) {
    ritemp = fact*ril+ripl;
    fact -= xi;
    ripl = fact*ritemp + ril;
    ril = ritemp;
}

f = ripl/ril;
if (x < XMIN) {
    x2 = 0.5+x;
pimu = PI*xmu;
    fact = (fabs(pimu) < EPS ? 1.0 : pimu/sin(pimu));
d = -log(x2);
e = xmu*d;
    fact2 = (fabs(e) < EPS ? 1.0 : sinh(e)/e);
beschb(xmu,&gam1,&gam2,&gaml,&gamm);
\[ ff = \text{fact}*(\text{gam1}*\cosh(e) + \text{gam2} \times \text{fact2} \times d); \]

\[ \text{sum} = ff; \]
\[ e = \exp(e); \]
\[ p = 0.5*e/\text{gaml}; \]
\[ q = 0.5/(e*\text{gamm}); \]
\[ c = 1.0; \]
\[ d = x2 \times x2; \]
\[ \text{sum1} = p; \]
\[ \text{for} (i=1; i<=\text{MAXIT}; i++) \{ \]
\[ \quad \text{ff} = (i*ff+p+q)/(i*1-xmu2); \]
\[ \quad c *= (d/i); \]
\[ \quad p /= (i - xmu); \]
\[ \quad q /= (i + xmu); \]
\[ \quad \text{del} = c*ff; \]
\[ \quad \text{sum} += \text{del}; \]
\[ \quad \text{del1} = c*(p-i*ff); \]
\[ \quad \text{sum1} += \text{del1}; \]
\[ \quad \text{if} (\text{fabs} (\text{del}) < \text{fabs}(\text{sum})*\text{EPS}) \text{ break}; \]
\[ \}
\[ \text{if} \ (i > \text{MAXIT}) \]
\[ \{
\quad \text{printf}("\text{bessik series failed to converge}");
\quad \text{exit} (\text{EXIT\_FAILURE});
\}
\[ \text{rkmu} = \text{sum}; \]
\[ \text{rk1} = \text{sum1} \times x2; \]
\[ \}
\[ \text{else} \{
\quad b = 2.0*(1.0+x);
\quad d = 1.0/b;
\quad h = \text{delh} = d;
\quad q1 = 0.0;
\quad q2 = 1.0;
\quad a1 = 0.25 - xmu2;
\quad q = c = a1;
\quad a = -a1;
\quad s = 1.0+q*\text{delh}; \]
\[ \text{for} (i=1; i<\text{MAXIT}; i++) \{ \]
\[ \quad a -= 2*i; \]
\[ \quad c = -a*c/(i+1.0); \]
\[ \quad \text{qnew} = (q1-b*q2)/a; \]
\[ \quad q1 = q2; \]
\[ \quad q2 = \text{qnew}; \]
\[ \quad q += c*\text{qnew}; \]
\[ \quad b += 2.0; \]
\[ \quad d = 1.0/(b+a*d); \]
\[ \quad \text{delh} = (b*1-d)*\text{delh}; \]
\[ \quad h *= \text{delh}; \]
\[ \quad \text{dels} = q*\text{delh}; \]
\[ \quad s *= \text{dels}; \]
\[ \quad \text{if} (\text{fabs}(\text{dels/s})<=\text{EPS}) \text{ break}; \]
\[ \}
\[ \text{if} \ (i >= \text{MAXIT}) \]
\[ \{
\]
7. Appendix

```c
printf("bessik: failure to converge in cf2");
exit(EXIT_FAILURE);
}

h = a1*h;
rmu = sqrt(PI/(2.0*x)*exp(-x)/s);
rk1 = rmu*(xmu+x+0.5-h)*xi;
"

rkmu = xmu*xi*rmu-rk1;
rimu = xi/(f*rmu-rmup);
*ri = (rimu*ril)/ril;
*rip = (rimu*ripl)/ril;
for (i=1;i <= nl;i++) {
    rktemp = (xmu+i)*xi2*rk1+rmu;
    rmu = rk1;
    rk1 = rktemp;
}

*rk = rmu;
*rkp = xnu*xi*rmu-rk1;
"

void beschb (double x, double *gam1, double *gam2, double *gampl, double *gammi)
{
    int NUSE1 = 7, NUSE2 = 8;
    double xx, *c1, *c2;
    c1 = vector(1,7);
    c2 = vector (1,8);
    c1[1] = -1.142022680371168e0;
    c1[2] = 6.5165112670737e-3;
    c1[3] = 3.08709173086e-4;
    c1[5] = 6.9437664e-9;
    c1[7] = -1.356e-13;
    c2[1] = 1.84374058730371168e0;
    c2[2] = -7.68528408447867e-2;
    c2[3] = 1.2719271366546e-3;
    c2[5] = -3.1261198e-8;
    c2[6] = 2.423096e-10;
    c2[7] = -1.702e-13;
    c2[8] = -1.49e-15;
    xx = 8.0*x*x-1.0;
    *gam1 = chebev(-1.0,1.0,c1,NUSE1,xx);
    *gam2 = chebev(-1.0,1.0,c2,NUSE2,xx);
    *gampl = *gam2-x*(**gam1);
    *gammi = *gam2-x*(**gam1);
    free_vector(c1,1,7);
    free_vector(c2,1,8);
    }
```
double chebev(double a, double b, double c[], int m, double x) {
    double d = 0.0, dd = 0.0, sv, y, y2;
    int j;
    if ((x-a)*(x-b) > 0.0) {
        printf("x not in routine chebev");
        exit(EXIT_FAILURE);
    }
    y2 = 2.0*(y = (2.0*x-a-b)/(b-a));
    for (j=m;j>1;j--) {
        sv = d;
        d = y2*d-dd+c[j];
        dd = sv;
    }
    return y*d-dd+0.5*c[1];
}

double *vector(long nl, long nh) {
    double *v;
    v = (double *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(double)));
    return v-nl+NR_END;
}

double **matrix(long nrl, long nrh, long ncl, long nch) {
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    double **m;
    m=(double **) malloc((size_t)((nrow+NR_END)*sizeof(double *)));
    m += NR_END;
    m -= nrl;
    m[nrl]=(double **) malloc((size_t)((nrow*ncol+NR_END)*
                     sizeof(double)));
    m[nrl] += NR_END;
    m[nrl] -= ncl;
    for (i=nrl+1;i<=nrh;i++) m[i]=m[i-1]+ncol;
    return m;
}

void free_vector(double *v, long nl, long nh) {
    free((FREE_ARG) (v+nl-NR_END));
}
void free_matrix(double **m, long nrl, long nrh, long ncl, long nch)
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
    free((FREE_ARG) (m+nrl-NR_END));
}

void powell(double p[], double **xi, int n, double ftol, int *iter, double *fret, double (*func)(double[]))
{
    int i, ibig, j;
    double del, fp, fptt, t,*pt,*ptt,*xit;
    pt=vector(1,n);
    ptt=vector(1,n);
    xit=vector(1,n);
    *fret=(*func)(p);
    for (j=1;j<=n;j++) pt[j]=p[j];
    for (*iter=1;;++(*iter)) {
        fp=(*fret);
        ibig=0;
        for (i=1;i<=n;i++) {
            for (j=1;j<=n;j++) xit[j]=xi[j][i];
            fptt=(*fret);
            linmin(p,xit,n,fret,func);
            if (fptt-(*fret) > del) {
                del=fptt-(*fret);
                ibig=i;
            }
        }
        if (2.0*(fp-(*fret)) <= ftol*(fabs(fp)+fabs(*fret))+TINY) {
            free_vector(xit,1,n);
            free_vector(ptt,1,n);
            free_vector(pt,1,n);
            return;
        }
        if (*iter == ITMAX) {
            printf("powell exceeding maximum iterations.");
            exit(EXIT_FAILURE);
        }
        for (j=1;j<=n;j++) {
            ptt[j]=2.0*p[j]-pt[j];
            xit[j]=p[j]-pt[j];
            pt[j]=p[j];
        }
        fptt=(*func)(ptt);
        if (fptt < fp) {
            t=2.0*(fp-2.0*(fptt)+fptt)*SQR(fp-(*fret)-del)-del*SQR(fp-fptt);
            if (t < 0.0) {
                linmin(p,xit,n,fret,func);
                for (j=1;j<=n;j++) {
                
            }


```c
7. Appendix 78

    xi[j][ibig]=xi[j][n];
    xi[j][n]=xit[j];

    }
    }
    }
    }
    }
}
}
}
}
}

int ncom;
double *pcom, *xicom, (*nrfunc)(double []);

void linmin(double p[], double xi[], int n, double *fret,
            double (*func)(double []))
{
    int j;
    double xx, xmin, fx, fb, fa, bx, ax;

    ncom=n;
    pcom=vector(1,n);
    xicom=vector(1,n);
    nrfunc=func;
    for (j=1;j<=n;j++) {
        pcom[j]=p[j];
        xicom[j]=xi[j];
    }
    ax=0.0;
    xx=1.0;
    mnbrak(&ax,&xx,&bx,&fa,&fx,&fb,f1dim);
    *fret=brent(ax,xx,bx,f1dim,TOL,&xmin);
    for (j=1;j<=n;j++) {
        xi[j] *= xmin;
        p[j] += xi[j];
    }
    free_vector(xicom,1,n);
    free_vector(pcom,1,n);
}

double f1dim(double x)
{
    int j;
    double f,*xt;
    xt=vector(1,ncom);
    for (j=1;j<=ncom;j++) xt[j]=pcom[j]+x*xicom[j];
    f=(*nrfunc)(xt);
    free_vector(xt,1,ncom);
    return f;
}

double brent(double ax, double bx, double cx, double (*f)(
              double), double tol1, double *xmin)
{
    int iter;
    double a,b,d,etemp, fu, fv, fw, px, x,tol1, tol2, u,v,w,x,xm
```
7. Appendix

:  

double e=0.0;

a=(ax < cx ? ax : cx);
b=(ax > cx ? ax : cx);
x=w=v=bx;
fw=fv=fx=(*f)(x);
for (iter=1; iter<=ITMAX; iter++) {
  xm=0.5*(a+b);
tol2=2.0*(tol1=tol*fabs(x)+ZEPS);
  if (fabs(x-xm) <= (tol2-0.5*(b-a))) {
    *xmin=x;
    return fx;
  }
  if (fabs(e) > tol1) {
    r=(x-w)*(fx-fv);
    q=(x-v)*(fx-fw);
    p=(x-v)*q-(x-w)*r;
    q=2.0*(q-r);
    if (q > 0.0) p = -p;
    q=fabs(q);
    etemp=e;
    e=d;
    if (fabs(p) >= fabs(0.5*q*etemp) || p <= q*(a-x) || p >= q*(b-x))
      d=CGOLD*(e=(x >= xm ? a-x : b-x));
    else {
      d=p/q;
      u=x+d;
      if (u-a < tol2 || b-u < tol2)
        d=SIGN(tol1,xm-x);
    }
  }
  else {
    d=CGOLD*{e=(x >= xm ? a-x : b-x)};
  }
  u=(fabs(d) >= tol1 ? x+d : x+SIGN(tol1,d));
  fu=(*f)(u);
  if (fu <= fx) {
    if (u >= x) a=x; else b=x;
    SHFT(v,w,x,u)
    SHFT(fv,fw,fx,fu)
  }
  else {
    if (u < x) a=u; else b=u;
    if (fu <= fw || w == x) {
      v=w;
      u=w;
      fv=fw;
      fw=fu;
    }
    else if (fu <= fv || v == x || v == w) {
      v=u;
      fv=fu;
    }
  }
}
printf("Too many iterations in brent");
exit(EXIT_FAILURE);
}

void mnbrak(double *ax, double *bx, double *cx, double *fa, double *fb, double *fc, double (*func)(double))
{
    double ulim, u, r, q, fu, dum;
    *fa = (*func)(*ax);
    *fb = (*func)(*bx);
    if (*fb > *fa) {
        SHFT(dum, *ax, *bx, dum)
        SHFT(dum, *fb, *fa, dum)
    }
    *cx = (*bx) + GOLD * (*bx - *ax);
    *fc = (*func)(*cx);
    while (*fb > *fc) {
        r = (*bx - *ax) * (*fb - *fc);
        q = (*bx - *cx) * (*fb - *fa);
        u = (*bx) - (((*bx - *cx) * q - (*bx - *ax) * r) / (2.0 * SIGN(FMAX(fabs(q - r), TINY), q - r)));
        ulim = (*bx) + GLIMIT * (*cx - *bx);
        if (((*bx - u) * (u - *cx) > 0.0) {
            fu = (*func)(u);
            if (fu < *fc) {
                *ax = (*bx);
                *bx = u;
                *fa = (*fb);
                *fb = fu;
                return;
            } else if (fu > *fc) {
                *cx = u;
                *fc = fu;
                return;
            }
        u = (*cx) + GOLD * (*cx - *bx);
        fu = (*func)(u);
    }
    else if (((*cx - u) * (u - ulim) > 0.0) {
        fu = (*func)(u);
    if (fu < *fc) {
        SHFT(*bx, *cx, u, *cx + GOLD * (*cx - *bx))
        SHFT(*fb, *fc, fu, (*func)(u))
    }
    }
    else if ((u - ulim) * (ulim - *cx) >= 0.0) {
        u = ulim;
        fu = (*func)(u);
    }
Return the five parameter estimates of GH distribution and log-likelihood value with integer \( \lambda \).

***Some of the functions used here are from Prause (1999) and the book *Numerical Recipes in C++* written by Press et al. (2002).***

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PI 3.141592653589793116
#define ACC 40.0
#define TINY 1.0e-20
#define ITMAX 3000
#define MAXIT 10000
#define TOL 2.0e-4
#define NR_END 1
#define FREE_ARG char *
define GOLD 1.618034
#define GLIMIT 100.0
#define EPS 1.0e-10
#define XMIN 2.0
#define FPMIN 1.0e-30

static double maxarg1, maxarg2;
define FMAX (a, b) (maxarg1=(a), maxarg2=(b), (maxarg1) > (maxarg2) ? \( (maxarg1) : (maxarg2) \)

static double sqrarg;
define SQR(a) ((sqrarg=(a)) == 0.0 ? 0.0 : sqrarg*sqrarg)

void mlgh(int *lambda, double *alpha, double *beta, double *delta, double *mu, double *ml);
```
double logligh(double *p);
double besskh(int n, double x);
double bessk(int n, double x);
double bessi0(double x);
double bessk0(double x);
double bessi1(double x);
double bessk1(double x);
int ff(double x);
double *vector(long nl, long nh);
double **matrix(long nrl, long nrh, long ncl, long nch);
void free_vector(double *v, long nl, long nh);
void free_matrix(double **m, long nrl, long nrh, long ncl, long nch);
void powell(double p[], double **xi, int n, double ftol, int *iter, double *fret, double (**func)(double []));
void linmin(double p[], double xi[], int n, double *fret, double (**func)(double []));
double f1dim(double x);
double brent(double ax, double bx, double cx, double (*f)(double), double tol, double *xmin);
void mnbrak(double *ax, double *bx, double *cx, double *fa, double *fb, double *fc, double (**func)(double));
double *readdata(int *n);
FILE *DATA;
FILE *RES;
double *X;
int H;
void main()
{
    int n,i,lambda;
    double *r, alpha, beta, delta, mu, ml;
    r = readdata(&n);
    H = n;
    X = malloc(sizeof(double)*(n+1));
    for (i = 1; i <= H; i++) *(X+i)=*(r+i);
    mlgh(&lambda, &alpha, &beta, &delta, &mu, &ml);
    printf("%d
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
", lambda, alpha, beta, delta, mu, ml);
    RES = fopen("result.txt","w");
fprintf(RES, "%d
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
%9.4f
", lambda, alpha, beta, delta, mu, ml);
fclose(RES);
}

double *readdata(int *n)
{
    int i;
    double *r, tmp;
DATA = fopen("data.txt", "r");
fscanf(DATA, "%lf", &tmp);
*n = (int) tmp;
r = malloc(sizeof(double)*(*n+1));
*r = 0.0;
for ( i = 1 ; i <= (*n) ; i ++ ) {
    fscanf(DATA, "%lf", &tmp);
    *(r+i) = tmp;
}
fclose(DATA);
return r;
}

void mlgh(int *lambda, double *alpha, double *beta, double *delta, double *mu, double *ml)
{
    int iter;
    long i;
    double *p, **xi, ftol=1.0e-6, fret, psi, chi, t1, t2, tmp,
           *lambda = ff(p[1]);
           *beta = p[3];
           *delta = exp(0.5*p[4]);
           *mu = p[5];
           psi = p[2];
           chi = p[4];
           t1=0;
           t2=0;
    for ( i = 1 ; i <= H ; i ++ ){
        dif = *(X+i)-*mu;
        a1 = exp(chi)+dif*dif;
        }
\[ t_1 = t_1 + \log(a_1); \]
\[ a_2 = \sqrt{(*)\beta(*) + \exp(\psi) + a_1}; \]
\[ t_2 = t_2 + \log(besskh(\lambda_2 - 1, a_2)) + \beta * \text{dif}; \]
\]
\[ \text{tmp} = \exp(0.5 * (\psi + \chi)); \]
\[ \text{tmp} = \log(bessk(\lambda, \text{tmp})); \]
\[ \text{tmp} = 0.5 * (\lambda + -0.5 * \log(2\pi) + 0.5 * (\lambda) * -0.25 * \log(\beta * \text{exp(\psi)} - 0.5 * (\lambda) * \chi - \text{tmp}; \]
\[ \text{tmp} = H * \text{tmp} + (0.5 * (\lambda) - 0.25) * t_1 + t_2; \]
\[ \text{tmp} = 0.5 * (\lambda) - 0.25 * \text{dif} \]
\[ \text{free_vector}(p, 1, 5); \]
\[ \text{free_matrix}(x_1, 1, 5, 1, 5); \]
\]
\[ \text{double logligh(double p[])} \]
\[
\{ 
\]
\[ \text{int lambda}; \]
\[ \text{long i}; \]
\[ \text{double a1, a2, dif, psi, beta, chi, mu, t1, t2, tmp}; \]
\[ \text{lambda} = \text{ff}(p[1]); \]
\[ \text{psi} = p[2]; \]
\[ \text{beta} = p[3]; \]
\[ \text{chi} = p[4]; \]
\[ \text{mu} = p[5]; \]
\[ t1 = 0; \]
\[ t2 = 0; \]
\[ \text{for (i = 1; i <= H; i ++ )} \}
\[ \text{tmp} = \exp(0.5 * (\psi + \chi)); \]
\[ \text{tmp} = \log(bessk(\lambda, \text{tmp})); \]
\[ \text{tmp} = 0.5 * \lambda - 0.5 * \log(2\pi) - 0.5 * \lambda - 0.25 * \log(\beta * \text{exp(\psi)} - 0.5 * \lambda + \chi - \text{tmp; \]
\[ \text{tmp} = H * \text{tmp} + (0.5 * \lambda - 0.25) * t_1 + t_2; \]
\[ \text{tmp} = -\text{tmp}; \]
\[ \text{return} \text{tmp}; \]
\]
\[ \text{double besskh(int n, double x)} \]
\[
\{ 
\]
\[ \text{int i, j}; \]
\[ \text{double sum1, sum2, sum3, tmp}; \]
\[ \text{if (n <= -2) n = (int) \text{fabs(n)} - 1; \]
\[ \text{if (n == -1 || n == 0)} \}
\[ \text{sqrt(PI/2) * exp(-0.5 * log(x) - x); \]
\[ \text{return} \text{tmp}; \]
\begin{verbatim}
7. Appendix

double bessk(int n, double x) {
    int j;
    double bk, bkm, bkp, tox;
    n = abs(n);
    tox=2.0/x;
    bkm=bessk0(x);
    bk=bessk1(x);
    for (j = 1; j < n; j++) {
        bkp=bkm+j*tox*bk;
        bkm=bk;
        bk=bkp;
    }
    return bk;
}

double bessi0(double x) {
    double ax, ans;
    double y;
    if ((ax=fabs(x)) < 3.75) {
        y=x/3.75;
        y*=y;
        ans=1.0+y*(3.5156229+y*(3.0899424+y*(1.2067492+y*(0.2659732+y*(0.360768e-1+y*0.45813e-2)))))
    } else {
        y=3.75/ax;
        //
    }
}
\end{verbatim}
ans=(exp(ax)/sqrt(ax))*(0.39894228*y*(0.1328592e-1 + y*(-0.1575656e-2 + y*(0.196281e-2) + y*(0.2635537e-1 + y*(-0.1647633e-1 + y*(0.392737e-2))))));
return ans;
}
}
double bessk0(double x)
{
double y, ans;
if (x <= 2.0) {
    y=x*x/4.0;
    ans=(-log(x/2.0)*bessi0(x)) + (-0.57721566 + y*(0.42278420 + y*(-0.7832358e-1 + y*(-0.1062446e-1 + y*(-0.587872e-2))));
} else {
    y=2.0/x;
    ans=(exp(-x)/sqrt(x))*(1.25331414 + y*(-0.7832358e-1 + y*(0.587872e-2)));
}
return ans;
}

double bessi1(double x)
{
double ax, ans;
    double y;
    if ((ax=fabs(x)) < 3.75) {
        y=x/3.75;
        y=y;
        ans=ax*(0.5+y*(0.87890594+y*(0.51498869+y*(0.301532e-2+y*0.32411e-3))));
    } else {
        y=3.75/ax;
        ans=0.228296761+y*(-0.2895312e-1+y*((0.1787654e-1 -y*0.420059e-2));
        ans=0.39894228+y*(-0.3988024e-1+y*(-0.362018e-2 + y*0.1031556e-1+y*ans));
        ans *= (exp(ax)/sqrt(ax));
    }
return x < 0.0 ? -ans : ans;
}

double bessk1(double x)
{
    double bessi1(double x);
    double y, ans;
    if (x <= 2.0) {
        y=x*x/4.0;
        ans=(log(x/2.0)*bessi1(x)) + (1.0+y*(-0.15443144 + y*(-0.18156897+y*(-0.1919402e-1 + y*(-0.110404e-2+y*(-0.46886e-4))))));
    }
\begin{verbatim}
7. Appendix

\begin{verbatim}
} else {
    y = 2.0 / x;
    ans = (exp(-x) / sqrt(x)) * (1.25331414 + y * (0.23498619 + y * (-0.3655620 e-1 + y * (-0.780353 e-2 + y * (0.325614 e-2 + y * (-0.68245 e-3))));
}
return ans;
\}
\}
int ff(double x){
    int y;
    if (x - (int)x >= 0.5) y = (int)x + 1;
    else y = (int)x;
    return y;
}
\}
double *vector(long nl, long nh)
{
    double *v;
    v = (double *)malloc((size_t)((nh-nl+1)*NR_END)*sizeof(double));
    return v-nl+NR_END;
}
\}
double **matrix(long nrl, long nrh, long ncl, long nch)
{
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    double **m;
    m = (double **)malloc((size_t)((nrow*NR_END)*sizeof(double *)));
    m += NR_END;
    m -= nrl;
    m[nrl]=(double *)malloc((size_t)((nrow*ncol+NR_END)*sizeof(double)));
    m[nrl] += NR_END;
    m[nrl] -= ncl;
    for (i=nrl+1; i<=nrh; i++) m[i]=m[i-1]+ncol;
    return m;
}
\}
void free_vector(double *v, long nl, long nh)
{
    free((FREE_ARG) (v+nl-NR_END));
}
\}
void free_matrix(double **m, long nrl, long nrh, long ncl, long nch)
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
\}
\end{verbatim}
\end{verbatim}
\end{document}

void powell(double p[], double **xi, int n, double ftol, int *iter, double *fret, double (*func)(double []))
{
    int i, ibig, j;
    double del, fp, fptt, t, *pt, *ptt, *xit;
    pt = vector(1, n);
    ptt = vector(1, n);
    xit = vector(1, n);
    *fret = (*func)(p);
    for (j=1; j<=n; j++) pt[j] = p[j];
    for (*iter = 1; (*iter)++) {
        fp = (*fret);
        ibig = 0;
        del = 0.0;
        for (i=1; i<=n; i++) {
            for (j=1; j<=n; j++) xit[j] = xi[j][i];
            fptt = (*fret);
            linmin(p, xit, n, fret, func);
            if (fptt > (*fret)) {
                del = fptt - (*fret);
                ibig = i;
            }
        }
        if (2.0*(fp - (*fret)) <= ftol*(fabs(fp) + fabs(*fret)) + TINY) {
            free_vector(xit, 1, n);
            free_vector(ptt, 1, n);
            free_vector(pt, 1, n);
            return;
        }
        if (*iter == ITMAX) {
            printf("powell exceeding maximum iterations.");
            exit(EXIT_FAILURE);
        }
        for (j=1; j<=n; j++) {
            ptt[j] = 2.0*p[j] - pt[j];
            xit[j] = p[j] - pt[j];
            pt[j] = p[j];
        }
        fptt = (*func)(ptt);
        if (fptt < fp) {
            t = 2.0*(fp - 2.0*(fptt) + fptt)*SQR(fp - (*fret) - del) -
                del*SQR(fp - fptt);
            if (t < 0.0) {
                linmin(p, xit, n, fret, func);
                for (j=1; j<=n; j++) {
                    xi[j][ibig] = xi[j][n];
                    xi[j][n] = xi[j];
                }
            }
        }
    }
}
7. Appendix

```c
    int ncom;
    double *pcom, *xicom, (*nrfunc)(double []);

    void linmin(double p[], double xi[], int n, double *fret, double (*func)(double []))
    {
        int j;
        double xx, xmin, fx, fb, fa, bx, ax;
        ncom=n;
        pcom=vector(1,n);
        xicom=vector(1,n);
        nrfunc=func;
        for (j=1; j<=n; j++) {
            pcom[j]=p[j];
            xicom[j]=xi[j];
        }
        ax=0.0;
        xx=1.0;
        mnbrak(&ax, &xx, &bx, &fa, &fx, &fb, f1dim);
        *fret=brent(ax, xx, bx, f1dim, TOL, &xmin);
        for (j=1; j<=n; j++) {
            xi[j] *= xmin;
            p[j] += xi[j];
        }
        free_vector(xicom,1,n);
        free_vector(pcom,1,n);
    }

    double f1dim(double x)
    {
        int j;
        double f, *xt;
        xt=vector(1,ncom);
        for (j=1; j<=ncom; j++) xt[j]=pcom[j]+x*xicom[j];
        f = (*nrfunc)(xt);
        free_vector(xt, 1, ncom);
        return f;
    }

    double brent(double ax, double bx, double cx, double (*f)(double), double tol, double *xmin)
    {
        int iter;
        double a, b, d, etemp, fu, fv, fw, fx, p, r, tol1, tol2, u, v, w, x, xm;
        double e = 0.0;
        a = (ax < cx ? ax : cx);
```
7. Appendix

b=(ax > cx ? ax : cx);
x=x=v=bx;
fw=fv=fx=(*f)(x);
for (iter=1; iter<=ITMAX; iter++) {
    xm=0.5*(a+b);
    tol2=2.0*(tol1=tol*fabs(x)+ZEPS);
    if (fabs(x-xm) <= (tol2-0.5*(b-a))) {
        *xmin=x;
        return fx;
    }
    if (fabs(e) > tol1) {
        r=(x-w)*(fx-fv);
        q=(x-v)*(fx-fw);
        p=(x-v)*q-(x-w)*r;
        q=2.0*(q-r);
        if (q > 0.0) p = -p;
        q=fabs(q);
        etemp=e;
        e=d;
        if (fabs(p) >= fabs(0.5*q*etemp) || p <= q*(a-x) || p >= q*(b-x))
            d=CGOLD*(e=(x >= xm ? a-x : b-x));
        else {
            d=p/q;
            u=x+d;
            if (u-a < tol2 || b-u < tol2)
                d=SIGN(tol1,xm-x);
        }
    } else {
        d=CGOLD*(e=(x >= xm ? a-x : b-x));
    }
    u=(fabs(d) >= tol1 ? x+d : x+SIGN(tol1,d));
    fu=(*f)(u);
    if (fu <= fx) {
        if (u >= x) a=x; else b=x;
        SHFT(v,w,x,u)
        SHFT(fv,fw,fx,fu)
    } else {
        if (u < x) a=u; else b=u;
        if (fu <= fw || w == x) {
            v=w;
            w=u;
            fw=fw;
        }
    }
}
printf("Too many iterations in brent");
7. Appendix

```c
exit(EXIT_FAILURE);
}

void mnbrak(double *ax, double *bx, double *cx, double *fa,
            double *fb, double *fc, double (*func)(double))
{
    double ulim, u, r, q, fu, dum;
    *fa = (*func)(*ax);
    *fb = (*func)(*bx);
    if (*fb > *fa) {
        SHFT(dum, *ax, *bx, dum)
        SHFT(dum, *fb, *fa, dum)
    }
    *cx = (*bx) + GOLD * (*bx - *ax);
    *fc = (*func)(*cx);
    while (*fb > *fc) {
        r = (*bx - *ax) * (*fb - *fc);
        q = (*bx - *cx) * (*fb - *fa);
        u = (*bx) - ((*bx - *cx) * q - (*bx - *ax) * r) / (2.0 * SIGN(FMAX(fabs(q - r), TINY), q - r));
        ulim = (*bx) + GLIMIT * (*cx - *bx);
        if ((*bx - u) * (u - *cx) > 0.0) {
            fu = (*func)(u);
            if (fu < *fc) {
                *ax = (*bx);
                *bx = u;
                *fa = (*fb);
                *fb = fu;
                return;
            } else if (fu > *fb) {
                *cx = u;
                *fc = fu;
                return;
            }
            u = (*cx) + GOLD * (*cx - *bx);
            fu = (*func)(u);
        } else if (((*cx - u) * (u - ulim)) > 0.0) {
            fu = (*func)(u);
            if (fu < *fc) {
                SHFT(*bx, *cx, u, *cx + GOLD * (*cx - *bx))
                SHFT(*fb, *fc, fu, (*func)(u))
            }
        } else if (((u - ulim) * (ulim - *cx)) >= 0.0) {
            u = ulim;
            fu = (*func)(u);
        } else {
            u = (*cx) + GOLD * (*cx - *bx);
            fu = (*func)(u);
        }
    }
```
\texttt{SHFT(*ax,*bx,*cx,u)}
\texttt{SHFT(*fa,*fb,*fc,fu)}
\texttt{\}}
7. XploRe Codes

7.2 mlgh.xpl

Call dll file to use function "mlgh" in XploRe environment.

```plaintext
proc(lambda, alpha, beta, delta, mu, ml) = mlgh(data)
; -------------------------------------------
; Macro mlgh
; -------------------------------------------
; Description call dll file to use function "mlgh"
; in XploRe environment.
; estimate the five parameters of GH
distribution with fractional lambda.
; -------------------------------------------
; Usage {lambda, alpha, beta, delta, mu, ml} = mlgh(data)
; Input
; Parameter data
; Definition numeric, data to be estimated
; Output
; Parameter lambda
; Definition scalar, estimated value of parameter lambda
; Parameter alpha
; Definition scalar, estimated value of parameter alpha
; Parameter beta
; Definition scalar, estimated value of parameter beta
; Parameter delta
; Definition scalar, estimated value of parameter delta
; Parameter mu
; Definition scalar, estimated value of parameter mu
; Parameter ml
; Definition scalar, value of log-likelihood
; -------------------------------------------

lambda = 0
alpha = 0
delta = 0
beta = 0
mu = 0
ml = 0
length = rows(data)

h = dlopen("Project2.dll")
dcall(h,"mlgh",data,length,lambda,alpha,beta,delta,mu,ml)
dclose()
endp
```

7.2.2 mlghint.xpl

Call dll file to use function "mlghint" in XploRe environment.
7. Appendix

7.2.3 ghmv.xpl

Calculate the mean and variance of a given GH distribution.

```plaintext
proc(M,V)=ghmv(l,a,b,d,m)
; -----------------------------------------------
; Library distribs
; -----------------------------------------------
; Macro ghmv
; -----------------------------------------------
```

```plaintext
lambda = 0
alpha = 0
delta = 0
beta = 0
mu = 0
ml = 0
length = rows(data)
```

```plaintext
h = dlopen("Project3.dll")
dlcall(h,"mlghint",data,length,lambda,alpha,beta,delta,mu,ml)
dlclose()
endp
```
Description: calculate the mean and variance of a given generalized hyperbolic distribution.

Usage: \( \{M, V\} = \text{ghmv}(l, a, b, d, m) \)

Input:
- Parameter \( l \):
  - Definition: scalar, parameter lambda of the GH distribution
- Parameter \( a \):
  - Definition: scalar, parameter alpha of the GH distribution
- Parameter \( b \):
  - Definition: scalar, parameter beta of the GH distribution
- Parameter \( d \):
  - Definition: scalar, parameter delta of the GH distribution
- Parameter \( m \):
  - Definition: scalar, parameter mu of the GH distribution

Output:
- Parameter \( M \):
  - Definition: scalar, mean of the GH distribution
- Parameter \( V \):
  - Definition: scalar, Variance of the GH distribution

Example:
\[
x = \text{ghmv}(-0.5, 1, 0, 1, 0)
x.M
x.V
\]

Result:
- Contents of \( M \):
  - \([1,] 0\)
- Contents of \( V \):
  - \([1,] 1\)

Author: Congcong Wang

\[
g = \sqrt{a^2 - b^2}
\]
\[
M = m + b*d/g*mbessel3(l+1,d*g)/mbessel3(l,d*g)
\]
\[
V = mbessel3(l+2,d*g)/mbessel3(l,d*g)
\]
\[
V = V - (mbessel3(l+1,d*g)/mbessel3(l,d*g))^2
\]
\[
V = V * b^2/g^2
\]
\[
V = V + mbessel3(l+1,d*g)/(d*g*mbessel3(l,d*g))
\]
\[
V = V * d^2
\]

endp
BIBLIOGRAPHY


Barndorff-Nielsen, O. E., Blæsild, P. and Schmiegel, J. (2004). A parsimo-
nious and universal description of turbulent velocity increments, *The
European Physical Journal B* 41: 345–363.

The fascination of sand - with three appendices by r.a.bagnold, *Depart-
ment of Theoretical Statistics, Institute of Mathematics, University of
Århus, Denmark.*

The fascination of sand, in a. c. atkinson & s. e. fienberg (eds), *A
celebration of statistics - The ISI centenary volume, Springer, New York*
pp. 57–87.


distributions, with an application to Johannsen’s bean data, *Biometrika*
68: 251–263.

distributions., *Working Paper, University of Århus.*

noulli* 1: 281–299.

Eriksson, A., Forsberg, L. and Ghysels, E. (2004). Approximating the prob-
ability distribution of functions of random variables: A new approach,
*Série Scientifique* 21.


distribution - a case study: Pattern of sediment sorting in a small
tidal-inlet - het zwin, the netherlands, *Journal of Coastal Research*
9: 1044–1053.

geomagnetic field in iceland, *The Geophysical Journal of the Royal As-

Pareto, V. (1896). *Cours d’économie politique professé à l’université de
Lausanne.*


