Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author’s right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage. [http://researchspace.auckland.ac.nz/feedback](http://researchspace.auckland.ac.nz/feedback)

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the [Library Thesis Consent Form](#) and [Deposit Licence](#).

Note: Masters Theses

The digital copy of a masters thesis is as submitted for examination and contains no corrections. The print copy, usually available in the University Library, may contain corrections made by hand, which have been requested by the supervisor.
Some Problems Concerning the Generalized Hyperbolic and Related Distributions

by

Thanh Tam, TRAN

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Auckland
Abstract

This thesis presents our attempts to deal with some significant problems concerning the generalized hyperbolic and generalized inverse Gaussian distribution.

Firstly, the presence of the modified Bessel function of the second kind $K_{\lambda}(z)$ in the density function of these distributions has been known to be one of the challenges for research effort. Here, the approach is to obtain a reliable algorithm which, for a given $|\lambda| \in [0, 90]$, can identify a sufficiently large value of $z$ to obtain accurate approximate value of $K_{\lambda}(z)$ with a definite number of terms. Consequently, computational and mathematical difficulties which are due to $K_{\lambda}(z)$ can be dealt with in a much simpler manner replacing the special function by a finite series. For completeness, a reliable routine to approximate the function when $z$ is small was also implemented.

The second problem concerns the fitting of the univariate hyperbolic and univariate generalized inverse Gaussian distribution to data using a numeric optimization algorithm. It is well-known that the log-likelihood functions of these distributions are flat and that they need “good” starting values to converge to an optimal value. These problems were reported in the literature even when sample sizes of 500 observations were used. Here, the approaches include obtaining parameter estimates by a symbolic method and reducing the parameter space. The latter is to relax the assumption which must be met for the former to be applicable. Important benefits of the symbolic method are it eliminates the need for starting values and it works stably for sample size of less than 100, or even 20, observations.

Thirdly, routines to fit subclasses of the multivariate generalized hyperbolic distribution using the Expectation Maximization (EM) algorithm have been derived in the literature. This research suggests an EM based fitting routine for three subclasses of the multivariate generalized hyperbolic, which significantly improves the
algorithm convergence speed while maintaining its simplicity. The approach can also be extended to other subclasses.

The modified Bessel function of the second kind is one of the important special functions. Research in the literature has identified the evaluation of the incomplete Bessel function as challenging. Thus, significant effort has been spent on obtaining a numerical method to evaluate it. Here, the problem is approached from both analytical and computational directions and involving three different integral representations of the incomplete Bessel function.

Lastly, calculating tail probabilities of the generalized inverse Gaussian has attracted major research effort. This research utilizes the result obtained from the evaluation of the incomplete Bessel function to derive analytical and numerical approaches to calculate these probabilities.
Acknowledgements

I wish to thank Associate-Professor David Scott and Dr Thomas Yee of the Department of Statistics who supervised this thesis in terms of the content and presentation respectively. I am also grateful to Dr Garry Tee of the Department of Mathematics for his time and willingness to proofread my work.
## Contents

1 Introduction  
1.1 The generalized inverse Gaussian distribution 19  
1.2 The univariate generalized hyperbolic distribution 21  
1.2.1 The hyperbolic distribution 24  
1.3 The multivariate generalized hyperbolic distributions 26  
1.4 Parameter estimation of the subclasses of the GH distributions 29  
1.5 Thesis outline 30  

2 Asymptotic approximations of the modified Bessel function of the second kind 34  
2.1 Introduction 34  
2.1.1 Modified Bessel function of the second kind 35  
2.1.2 Asymptotic approximations of $K_{\lambda}(z)$ in the literature 35  
2.1.3 Applications of asymptotic approximations of $K_{\lambda}(z)$ in statistics 38  
2.2 Asymptotic expansion series of $K_{\lambda}(z)$ 39  
2.2.1 Derivation 40  
2.2.2 Features of the remainder 42  
2.2.3 Properties of the asymptotic approximations 43  
2.3 An algorithm to approximate the Bessel function when $|\lambda|$ is not half of an odd integer 44  
2.3.1 Identifying sufficiently large value of $z$ 45
2.3.2 Approximation procedure for $K_\lambda(z)$ for $\lambda \in [-90, 90]$ ............................. 50
2.4 Approximations when $z$ values are small ................................................................. 54
2.5 Numerical studies ........................................................................................................... 55
  2.5.1 Asymptotic approximation versus “exact” evaluation ........................................... 56
  2.5.2 Asymptotic approximation versus approximation using Debye polynomials ...................... 57
  2.5.3 Asymptotic versus other approximation method ...................................................... 60
2.6 Summary ......................................................................................................................... 62

3 Parameter estimation of the hyperbolic and subclasses of the generalized inverse Gaussian 63
  3.1 Introduction ..................................................................................................................... 63
  3.2 Maximum-likelihood estimation ..................................................................................... 65
    3.2.1 Calculus steps .......................................................................................................... 65
    3.2.2 Computational tools for symbolic maximum-likelihood estimation ......................... 67
  3.3 Symbolic maximum likelihood estimators of the shape parameters of the hyperbolic distribution 68
    3.3.1 Approximate log-likelihood function of the hyperbolic distribution ................................ 68
    3.3.2 Evaluating symbolic maximum-likelihood estimators of the approximate log-likelihood 71
    3.3.3 Numerical examination ......................................................................................... 73
    3.3.4 Symbolic versus numeric fitting of the shape parameters .................................... 75
  3.4 Numerical approach for fitting the standardized hyperbolic distribution 78
    3.4.1 Standardization of the generalized hyperbolic distributions ..................................... 79
    3.4.2 Steps to fit standardized hyperbolic distribution .................................................... 81
    3.4.3 Numerical examination ......................................................................................... 83
    3.4.4 Numerical examination ......................................................................................... 83
3.4.5 Fitting the standardized hyperbolic distribution to real financial data ............................................. 86
3.5 Maximum-likelihood estimation of three subclasses of the generalized inverse Gaussian distributions ................................. 88
3.5.1 Symbolic maximum likelihood estimators ........................................ 89
3.5.2 Evaluation of the maximum-likelihood estimators ................................ 91
3.5.3 Numerical examination ........................................................................ 92
3.5.4 Symbolic versus numeric fitting of the subclasses of the GIG ........... 93
3.6 Conclusions ......................................................................................... 97

4 Calibration of subclasses of the multivariate generalized hyperbolic distributions ......................................................... 99
4.1 Introduction ......................................................................................... 99
4.2 Expectation Maximization algorithm ..................................................... 100
4.2.1 Variants of the EM algorithm ............................................................. 102
4.3 Parameter estimations .......................................................................... 103
4.3.1 Multivariate generalized hyperbolic distribution .............................. 104
4.3.2 Symbolically fitting subclasses of the multivariate generalized hyperbolic ................................................................. 107
4.4 Numerical examination ......................................................................... 107
4.4.1 Technical details ................................................................................ 107
4.4.2 Comparison of fitting outputs ............................................................ 109
4.4.3 Comparison of computational speed ................................................... 109
4.4.4 Fitting real multivariate data ............................................................... 110
4.5 Conclusion .......................................................................................... 111

5 Evaluation of the incomplete Bessel function ........................................... 113
5.1 Introduction .......................................................................................... 113
5.2 Evaluation of the incomplete Bessel function defined by (5.1) using
the Whittaker function .................................................. 116
5.2.1 Relationship between the Whittaker function and the Bessel
function in finite range .................................................. 116
5.2.2 Closed-form formulae of the incomplete Bessel function when λ
equals half of a positive odd integer ............................ 119
5.3 Evaluation of the Laplace integral in finite ranges .................... 123
5.3.1 Evaluation steps ..................................................... 125
5.4 Analytical evaluation of the incomplete Bessel function defined by (5.2)
when λ = ±\frac{1}{2} .......................................................... 129
5.4.1 Formulae for the lower incomplete Bessel function .............. 129
5.4.2 Formulae for the upper incomplete Bessel function .............. 129
5.5 Numerical approach to approximate incomplete Bessel function given
by (5.3) ................................................................. 130
5.5.1 Description of the algorithm ...................................... 131
5.6 Relationships between the incomplete Bessel function defined by (5.2),
(5.3) and the generalized inverse Gaussian distribution ............ 133
5.6.1 Relationships between (5.2) and the density function of the
generalized inverse Gaussian distribution (5.50) .................. 133
5.6.2 Relationships between (5.3) and the density function of the
generalized inverse Gaussian distribution (5.50) .................. 135
5.7 Numerical studies ..................................................... 136
5.7.1 Accuracy of the analytical methods ............................ 136
5.7.2 Accuracy of the numerical method ............................ 140
5.7.3 Difficulties with the numerical algorithm ...................... 141
5.7.4 Numerical versus analytical incomplete Bessel functions .... 146
5.8 Summary ............................................................. 149
6 Distribution function of the generalized inverse Gaussian distribution

6.1 Introduction .................................................. 150

6.2 Analytical distribution functions for the subclasses of the generalized inverse Gaussian .................................................. 152

6.2.1 Cumulative distribution function ................................ 152

6.2.2 Complementary cumulative distribution function ............ 152

6.3 Numerical cumulative and complement cumulative distribution function of the generalized inverse Gaussian .................................................. 153

6.4 Numerical studies ............................................... 154

6.4.1 When \( \lambda = -\frac{1}{2} \) ........................................ 154

6.4.2 When \( \lambda = \frac{1}{2} \) ........................................ 157

6.4.3 Numerical discussion ........................................... 159

7 Goodness-of-fit test for the hyperbolic distribution 160

7.1 Introduction .................................................. 160

7.1.1 Cramér-von Mises goodness-of-fit test for the hyperbolic distribution .................................................. 163

7.2 Current and recommended approaches ................................ 166

7.2.1 Look-up table ............................................. 166

7.2.2 Interpolation ............................................. 168

7.2.3 Multivariate adaptive polynomial spline regression ............ 168

7.2.4 Recommended approaches ................................... 169

7.3 Numeric distribution function of the test statistic for the hyperbolic distribution .................................................. 169

7.3.1 Extended table of critical values for the test ................ 170

7.3.2 Multivariate adaptive polynomial spline fitting .............. 174

7.4 Moran log spacing goodness-of-fit test for the hyperbolic distribution 179

7.5 Numerical study ............................................... 180
7.5.1 Tests using skew normal distribution . . . . . . . . . . . . . . 180
7.5.2 Goodness-of-fit test using real financial data . . . . . . . . . . 181
7.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182

8 Conclusions and future research 184

A Main functions of Chapter 2 191
A.1 Important testing information . . . . . . . . . . . . . . . . . . . . . . 191
A.2 Function to identify sufficient large argument value of $K_\lambda(z)$ . . . 193
A.3 Function to approximate $K_\lambda(z)$ given $z$ is sufficiently large . . . 194
A.4 Function to approximate $K_1(z)$ when $z$ is small . . . . . . . . . . 195

B Main functions of Chapter 3 196
B.1 Notes on the Q–Q plots of the hyperbolic distribution . . . . . . . . 196
B.2 Important testing information . . . . . . . . . . . . . . . . . . . . . . 197
B.2.1 Relating to the fitting functions in Sections 3.3 . . . . . . . . . 197
B.2.2 Relating to the fitting function in Section 3.4 . . . . . . . . . 198
B.2.3 Relating to the fitting functions in Sections 3.5 . . . . . . . . . 199
B.2.4 Critical effects of different parameterization conventions . . . 200
B.2.5 Intended sample sizes for testing . . . . . . . . . . . . . . . . . . 201
B.3 Symbolic maximum-likelihood of the shape parameters of the hyper-
bolic distribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 202
B.3.1 Numeric maximum-likelihood of the shape parameters of the
hyperbolic distribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . 204
B.3.2 Finding starting values for the numerical fitting process using
method of moments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
B.3.3 Comparing numeric with symbolic maximum-likelihood esti-
mation of the shape parameters of the hyperbolic distribution 206
B.4 Maximum-likelihood of the standardized hyperbolic distribution . . 209
B.5 Symbolic fitting of subclasses of the generalized inverse Gaussian distributions ........................................ 212
B.5.1 Numeric fitting of the subclasses of the generalized inverse Gaussian distributions ........................................ 214
B.5.2 Finding start values for the numerical fitting function using method of moments ........................................ 215
B.5.3 Comparing numeric with symbolic maximum-likelihood estimation of the subclasses of the generalized inverse Gaussian distributions ........................................ 216

C Main functions of Chapter 4 ........................................ 219
C.1 Important testing information ........................................ 219
C.2 Function to calibrate special cases of the multivariate hyperbolic distributions ........................................ 220
C.3 Parameter values used for the computational speed comparison in Section 4.4 ........................................ 222

D Main functions of Chapter 5 ........................................ 224
D.1 Important testing information ........................................ 224
D.1.1 Form one ........................................ 224
D.1.2 Form two ........................................ 224
D.2 Function to calculate the incomplete Bessel function given by the integral (5.1) ........................................ 225
D.3 Maple procedure to check the accuracy of the function in Appendix D ........................................ 226
D.4 Function to calculate the incomplete Bessel function given by the integral (5.2) ........................................ 228
D.5 Maple procedure to check the accuracy of the function in Appendix D ........................................ 229
D.6 Function to calculate the incomplete Bessel function given by the integral (5.3) ........................................ 230
D.7 Maple procedure to check the accuracy of numeIBF() 232
D.8 Function to compare analytical and numerical methods for $\lambda = \pm 1/2$ 234

E Main functions of Chapter 6 237
E.1 Important testing information 237
E.2 Function to calculate tail probabilities of the generalized inverse Gaussian distributions 239
E.3 Maple procedure to calculate tail probabilities of the generalized inverse Gaussian distributions 241

F Main functions of Chapter 7 243
F.1 Conducting Cramér-von Mises test for the hyperbolic distribution . 243
F.2 Moran test for the hyperbolic distribution . 245
F.3 Function to examine the power of the tests . 246
List of Figures
1.1

Plot of the log-probability densities function of the hyperbolic distribution on the shape triangle. . . . . . . . . . . . . . . . . . . . . . . . 25

2.1

Approximation accuracy vs number of terms for λ = 0. The dashed
lines represent the “exact” value K0 (z) given by function besselK (z, λ).
The solid lines indicate the value of the asymptotic approximations of
function besselKseriesAppro (z, λ) as s → 84. Plots (a), (b), (c) show
the approximation error when 1 ≤ s ≤ 5. Plots (d), (e), (f) show the
approximation error when 1 ≤ s ≤ 84. . . . . . . . . . . . . . . . . . . 46

2.2

Approximation accuracy vs number of terms for λ = 1. The dashed
lines represent the “exact” value K1 (z) given by function besselK (z, λ).
The solid lines indicate the value of the asymptotic approximations of
function besselKseriesAppro (z, λ) as s → 84. Plots (a), (b), (c) show
the approximation error when 1 ≤ s ≤ 5. Plots (d), (e), (f) show the
approximation error when 1 ≤ s ≤ 84. . . . . . . . . . . . . . . . . . . 47

2.3

Approximation accuracy vs number of terms for λ = 2. The dashed
lines represent the “exact” value K2 (z) given by function besselK (z, λ).
The solid lines indicate the value of the asymptotic approximation of
function besselKseriesAppro (z, λ) as s → 84. Plots (a), (b), (c) show
the approximation error when 1 ≤ s ≤ 5. Plots (d), (e), (f) show the
approximation error when 1 ≤ s ≤ 84. . . . . . . . . . . . . . . . . . . 48
9


3.1 Goodness-of-fit plots for the fitting of the shape parameters $\rho$ and $\zeta$. PIT data: Probability-integral-transformed data. 74

3.2 Histograms of differences between estimates and true parameter values. $\hat{\varepsilon} = \hat{\rho} - \rho$ and $\hat{\kappa} = \hat{\zeta} - \zeta$ where $\rho = 1.5$, $\zeta = 23$. First row: $n = 20$; second row: $n = 30$; third row: $n = 100$. 75

3.3 Goodness-of-fit plots for the fitting of parameters $\rho$ and $\zeta$ using standardized hyperbolic distribution. PIT data: Probability-integral-transformed data. 84

3.4 Histograms of differences between estimates and true parameter values. $\hat{\varepsilon} = \hat{\rho} - \rho$ and $\hat{\kappa} = \hat{\zeta} - \zeta$ where $\rho = 0$, $\zeta = 0.235$, $\delta = 1$ and $\mu = 10$. Column A: differences obtained by $\text{standHyperbFit()}$. Column B: differences obtained by $\text{hyperbFit()}$. 85

3.5 Histograms of differences between estimates and true parameter values. $\hat{\varepsilon} = \hat{\rho} - \rho$ and $\hat{\kappa} = \hat{\zeta} - \zeta$ where $\rho = -0.444$, $\zeta = 0.235$, $\delta = 1$ and $\mu = 5$. Column A: differences obtained by $\text{standHyperbFit()}$. Column B: differences obtained by $\text{hyperbFit()}$. 86

3.6 Plots of fitting standardized hyperbolic distribution to real financial return data. 88

3.7 Goodness-of-fit plots for the fitting of parameters $\chi$ and $\psi$. PIT: Probability-integral-transformed data. 93

3.8 Histograms of differences between estimates and true parameter values. $\lambda = -\frac{1}{2}$, $\hat{\varepsilon} = \hat{\chi} - \chi$ and $\hat{\kappa} = \hat{\psi} - \psi$ where $\chi = 5$, $\psi = 3$. First row $n = 20$, second row $n = 50$, third row $n = 100$. 94

3.9 Histograms of differences between estimates and true parameter values. $\lambda = \frac{3}{2}$, $\hat{\varepsilon} = \hat{\chi} - \chi$ and $\hat{\kappa} = \hat{\psi} - \psi$ where $\chi = 9$, $\psi = 1$. First row $n = 20$, second row $n = 50$, third row $n = 100$. 95
3.10 Histograms of differences between estimates and true parameter values. $\lambda = \frac{5}{2}$, $\hat{\varepsilon} = \hat{\chi} - \chi$ and $\hat{\kappa} = \hat{\psi} - \psi$ where $\chi = 5$, $\psi = 2$. First row $n = 20$, second row $n = 50$, third row $n = 100$.

4.1 Computational speed comparisons (Time is in seconds). Box 1: \texttt{symFitmNH()} (EM). Box 2: \texttt{symFitmNH()} (MCECM). Box 3: \texttt{fit.mNH()} (MCECM).

5.1 Integral of function $f(z)$ round a contour

5.2 Integral of function $W_{k,m}$ round a contour

7.1 Trellis plot $\xi$ vs $W_{\alpha}^2$ conditioned on 7 levels of $\alpha$ in Table 7.2

7.2 Trellis plot $\chi$ vs $W_{\alpha}^2$ conditioned on 7 levels of $\alpha$ in Table 7.2

7.3 Distribution of polymars() regression fitting residuals

B.1 Histogram and Q–Q plot of generated hyperbolic data versus their own theoretical quantiles
List of Tables

1.1 The relations between four parameterizations of the GIG .................. 20
1.2 The relations between four parameterizations of the generalized hyperbolic distributions. ............................................................... 23
1.3 The relations between four parameterizations of the hyperbolic distribution. ................................................................. 27
2.1 The .Machine numbers used by function besselKseriesAppr(z, \lambda) .... 49
2.2 Underlined numbers indicate the decimal places where “exact” and approximated value agree, and \( s \) is the number of terms required by \(^\ast K_\lambda(z)\). ................................................................. 57
2.3 Underlined numbers indicate the decimal places where “exact” and approximated evaluation agree. The evaluation was on an \( e^z \) scale to avoid an underflow problem, and \( s \) is the number of terms required by \(^\ast K_\lambda(z)\). ................................................................. 57
2.4 Coloured and underlined numbers indicate the decimal places where \(^d K_\lambda(z)\) and \(^\ast K_\lambda(z)\) agree with “exact” value \( K_\lambda(z) \) respectively. NaNs indicate the routine failed, and \( s \) is the number of terms required by \(^\ast K_\lambda(z)\). ................................................................. 59
2.5 Coloured and underlined numbers indicate the decimal places where \(^\ast K_1(z)\) and \(^\ast K_1(z)\) agree with “exact value” \( K_1(z) \) respectively, and \( s \) is the number of terms required by \(^\ast K_1(z)\). ................................................................. 61
3.1 Symbols ◦ indicate failure to start an optimization process. Parameter vector $\theta$ follows Parameterization 1 of package HyperbolicDist.

3.2 Symbols ◦ indicate failure to start an optimization process.

4.1 Fitting of 2-DHYP. Starting values $\frac{3}{2}, 10, 5$, and $n = 30$.

4.2 Fitting of 4-DHYP. Starting values $\frac{3}{2}, 15, 5$, and $n = 50$.

4.3 Fitting of 3 dimensions MNIG. Starting values $-\frac{1}{2}, 1, 5$, and $n = 30$.

5.1 Analytical evaluation incomplete Bessel function (5.1) versus numerical integration using Maple 15 and Mathematica. Values of $\lambda$ are to be divided by 2. Underlined numbers indicate the decimal places where the numbers agree.

5.2 Accuracy comparison between analytic lower incomplete Bessel function evaluated in $\mathbb{R}$ and numerical integration using Maple 15. Underlined numbers indicate the decimal places where the numbers agree.

5.3 Underlined numbers indicate the decimal places where the numbers in two columns agree.

5.4 Underlined numbers indicate the decimal places where the results given by numeIBF () and numeIBFproc () (correct values) agree. Symbols − indicate repeated values.

5.5 Underlined numbers indicate the decimal places where the results given by numeIBF () and numeIBFproc () (correct values) agree.

5.6 None of the results given by numeIBF () match the “exact evaluation”. Symbols − indicate repeated values.

5.7 Accuracy comparison between analytic and numeric evaluation of the lower incomplete Bessel function when $\lambda = 1/2$.

5.8 Accuracy comparison between analytic and numeric evaluation of the lower incomplete Bessel function when $\lambda = -1/2$. 
6.1 Accuracy comparison between different methods to calculate $P(W \leq w)$ for $\lambda = -\frac{1}{2}$. Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CDF. Symbol $-$ indicates repeated values. .................................................. 155

6.2 Accuracy comparison between different methods to evaluate $P(W > w)$ for $\lambda = -\frac{1}{2}$. Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CCDF. Symbol $-$ indicate repeated values. .................................................. 156

6.3 Accuracy comparison between different methods to evaluate $P(W \leq w)$ for $\lambda = \frac{1}{2}$. Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CDF. Symbol $-$ indicate repeated values. .................................................. 157

6.4 Accuracy comparison between different methods to evaluate $P(W > w)$ for $\lambda = \frac{1}{2}$. Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CCDF. Symbol $\ast$ and $-$ indicate greater than 1 probabilities and repeated values respectively. .................................................. 158

7.1 Short title ................................................................. 165

7.1 A reproduction of table of Table 5 by Puig and Stephens ............. 166

7.2 Short title ................................................................. 171

7.2 Columns with underlined values of $\chi$ and also the rows with $\alpha = 0.075, 0.04, 0.15$ are not available in Table 7.1 (e.g., Table 5 by Puig and Stephens.) .................................................. 172

7.3 $\beta$ is the shape parameters. Sample size $n = 500$. Level of significance $\alpha = 0.05$. .................................................. 181

7.4 $\beta$ is the shape parameters. Sample size $n = 500$. Level of significance $\alpha = 0.01$. .................................................. 181
List of Abbreviations

$d$-DHYP \hspace{1cm} d$-dimensional hyperbolic distributions.

CCDF \hspace{1cm} Complementary cumulative distribution function.

CDF \hspace{1cm} Cumulative distribution function.

EM \hspace{1cm} Expectation Maximization algorithm.

GH \hspace{1cm} Generalized hyperbolic distribution.

GIG \hspace{1cm} Generalized inverse Gaussian distribution.

HYP \hspace{1cm} Hyperbolic distribution.

IG \hspace{1cm} Inverse Gaussian distribution.

MHYP \hspace{1cm} Multivariate hyperbolic distribution.

MLE \hspace{1cm} Maximum likelihood estimator.

mle \hspace{1cm} Maximum likelihood estimate.

MNIG \hspace{1cm} Multivariate normal inverse Gaussian distribution.

NIG \hspace{1cm} Normal inverse Gaussian distribution.

PIT \hspace{1cm} Probability-integral-transformation.

SGH \hspace{1cm} Standardized generalized hyperbolic distribution.
SHYP  Standardized hyperbolic distribution.

SN    Skew normal distribution.
Chapter 1

Introduction

This thesis mainly consists of computational research to serve the purpose of creating efficient and reliable R routines to work with real non-Gaussian data. To this end, the computational stability and accuracy of these routines were extensively tested using at least 2 of these software packages: R, Maple and Mathematica. In addition, they are also clearly documented.

The introduction of the family of hyperbolic (HYP) distributions can be seen as one of the first major attempts to find more suitable distributions to work with heavy-tailed real data than the Gaussian distribution. The statistical foundation of the HYP is based on the groundbreaking work of [Bagnold (1954)], who conducted intriguing research into the transportation, sorting and sedimentation of sand particles by wind. He discovered that the size distribution of blown sand exhibits distinctive patterns. These random regularities, if plotted on the double logarithmic scale, will strongly indicate a smooth curve with two phases: increasing and then decreasing linearly after a smooth transition. This work was later formalized to be a family of the hyperbolic distributions in [Barndorff-Nielsen (1977)]. Important statistical and mathematical features of the HYP were discussed in [Barndorff-Nielsen and Halgreen (1977), Barndorff-Nielsen (1978), Barndorff-Nielsen (1979), Halgreen (1979), Barndorff-Nielsen and Blaesild (1981), Blaesild (1981), Blaesild and Jensen (1981)].
and Barndorff-Nielsen et al. (1985). They were termed hyperbolic distributions, because their logarithmic density plots are hyperbola and hyperboloid in one and several dimensions, respectively. The corresponding descriptions of the Gaussian distributions, however, are parabola and paraboloid. This is why the tails of the hyperbolic distributions decrease more slowly than those of the Gaussian distributions. Thus, they are more suitable to describe data with significant mass in the tails and greater kurtosis than a normal distribution.

The HYP is one of the subclasses of generalized hyperbolic (GH) distributions, which is created by a normal mean-variance mixture. Several attractive distributional and mathematical properties of the GH have been discovered in the literature. The GH distributions have a very flexible tail regime, spanning from Gaussian tails via exponential tails to power tails. Eberlein and Keller (1995) found that the hyperbolic distribution fitted well to German stock returns, and suggested the hyperbolic Lévy process for the evolution of stock prices. They concluded that the Lévy process describes stock returns better than the Brownian motion process. Prause (1999) devoted his PhD thesis to the study of the GH distribution. He proposed a new model for the pricing of financial options, to remedy distributional shortfalls of the Black-Scholes model\footnote{The Black-Scholes is a well-known model for financial options pricing. The main assumption is stock returns follow Gaussian distribution.}. In the multivariate context, McNeil et al. (2005) gave a comprehensive account of the fitting of the subclasses of the multivariate generalized hyperbolic distributions utilizing the Expectation Maximization (EM) algorithm.

Despite the practical and analytical importance of the GH family of distributions, computing routines which implement them for real data analysis appear not to have been fully investigated. The hyp program was the only computer program for fitting the hyperbolic distributions up to three dimensions, see Blæsild and Sørensen (1992). This program has become outdated as it was designed to accommodate the computer power of the early 1990s. Until recently, interest in computational and applied issues of the GH family has increased many-fold. This is because of the
exponentially increasing computer power, and widespread popularity of software such as R. The second reason is the discomfort of using Gaussian related models to describe non-Gaussian data. The work that is available in the form of R packages currently includes **GeneralizedHyperbolic** (Scott 2009), **fBasics** (Wüertz et al., 2010), **ghyp** (Breymann and Luethi, 2009) and **QRMlib** (McNeil and Ulman, 2010).

### 1.1 The generalized inverse Gaussian distribution

The generalized inverse Gaussian (GIG) distribution was proposed in 1946 by Étienne Halphen in research on hydrology. One of the most well-cited references for the properties of this distribution is Jørgensen (1982). The probability density distribution of the GIG has 3 parameters, and it is written as

\[
\text{GIG}(w|\lambda, \chi, \psi) = \frac{(\psi/\chi)^{\lambda/2}}{2K_\lambda(\sqrt{\chi\psi})} w^{\lambda-1} e^{-\{(1/2)(\chi w^{-1} + \psi w)\}}, \quad w > 0, \tag{1.1}
\]

which can occur in following cases

\[
\begin{align*}
\lambda &\in \mathbb{R}, \quad \chi > 0, \quad \psi > 0 \quad \text{(normal case)}, \\
\lambda &> 0, \quad \chi = 0, \quad \psi > 0 \quad \text{(Gamma distribution)}, \\
\lambda &< 0, \quad \chi > 0, \quad \psi = 0 \quad \text{(Inverse Gamma distribution)}
\end{align*}
\]

where \( K_\lambda(z) \) is the modified Bessel function of the second kind order \( \lambda \) and argument \( z = \sqrt{\chi\psi} \). It is very important to note that \( K_\lambda(z) \) is also called the modified Bessel function of the third kind in the literature. This thesis only concerns the normal case of the GIG. For these cases, the inverse Gaussian distribution \( (\lambda = -\frac{1}{2}) \) appears to be the only subclass that has been studied extensively in the literature. Two references, among others, are Chhikara and Folks (1974), and Jørgensen (1982).
Parameterization

<table>
<thead>
<tr>
<th>From: λ,α,β</th>
<th>To: Parameterization</th>
<th>χ/β = ψ</th>
<th>l/β = δ</th>
<th>ω = γ</th>
<th>0 &lt; l,0 &lt; ψ</th>
<th>ω,γ,β</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ,ω,η</td>
<td>Parameterization</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>0 &lt; l,0 &lt; ψ</td>
<td>l,δ,γ,ψ</td>
</tr>
<tr>
<td>λ,δ,γ</td>
<td>Parameterization</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>0 &lt; l,0 &lt; ψ</td>
<td>l,δ,γ,ψ</td>
</tr>
<tr>
<td>λ,χ,ψ</td>
<td>Parameterization</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>0 &lt; l,0 &lt; ψ</td>
<td>l,δ,γ,ψ</td>
</tr>
<tr>
<td>λ,α,β</td>
<td>Parameterization</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>l/β = δ</td>
<td>0 &lt; l,0 &lt; ψ</td>
<td>l,δ,γ,ψ</td>
</tr>
</tbody>
</table>

Table 1.1: The relations between four parameterizations of the GIG.
1.2 The univariate generalized hyperbolic distribution

The GIG is not a special case of the generalized hyperbolic distribution. The GIG was called the mixing distribution in the normal mean-variance to form the GH in Barndorff-Nielsen (1977). If a random variable \( W \) follows GIG(\( \lambda, \chi, \psi \)) with \( \lambda \) and \( \mu \in \mathbb{R}, \alpha \) and \( \delta \geq 0, \beta \in [-\alpha, \alpha] \) such that \( (\lambda, \delta, \sqrt{\alpha^2 - \beta^2}) \) are in the parameter space of the GIG, then the generalized hyperbolic distribution is given by

\[
\text{GH}(x; \lambda, \alpha, \beta, \delta, \mu) = \int_{0}^{\infty} f_N(x; \mu + \beta w, w) f_{\text{GIG}}(w; \lambda, \delta, \sqrt{\alpha^2 - \beta^2}) \, dw.
\]

After tedious algebra (cf. p.315–316 of Paolella 2007), the density function of the GH is obtainable

\[
\text{GH} (x; \lambda, \alpha, \beta, \delta, \mu) = a(\lambda, \alpha, \beta, \delta, \mu) \big( \delta^2 + (x - \mu)^2 \big)^{(\lambda-1)/2} \times K_{\lambda-1/2}(\alpha \sqrt{\delta^2 + (x - \mu)^2}) \exp \{ \beta (x - \mu) \}, \quad (1.2)
\]

where

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

The parameter domains of (1.2) are

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

The parameters \( \alpha \) and \( \beta \) are the shape parameters and \( \delta \in \mathbb{R}^+ \) and \( \mu \in \mathbb{R} \) are the scale and location parameters, respectively. The parameterization used in (1.2)
is called Parameterization 1. Other parameterizations which can be found in the
literature include

\begin{align}
2\text{nd} & \quad \zeta = \delta \sqrt{\alpha^2 - \beta^2}, \quad \rho = \frac{\beta}{\alpha}, \\
3\text{rd} & \quad \xi = (1 - \zeta)^{-1/2}, \quad \chi = \xi \rho, \\
4\text{th} & \quad \alpha = \alpha \delta, \quad \beta = \beta \delta.
\end{align}

The relationships between these parameterizations are presented in Table 1.2. The GH family is closed under affine transformation (with the parameter \( \lambda \) being invariant). The tail behaviour of the GH is generally considered to be semi-heavy. Table 9.3 of Paolella (2007) gives comprehensive connections between the GIG and the GH. When \( \lambda = -\frac{1}{2} \), the negative inverse Gaussian (NIG) distribution is obtainable. The variance-gamma (VG) distribution is a limiting case when \( \lambda = \frac{1}{2} \). The skewed generalized hyperbolic Student-\( t \) (SGHT) distribution is also an important limiting class. Standard references for these special cases of the GH include Madan et al. (1990), Madan et al. (1998), Prause (1999), Wenbo (2005) and Aas and Hobæk Haff (2006).
Table 1.2: The relations between four parameterizations of the generalized hyperbolic distributions.

<table>
<thead>
<tr>
<th>From: Parameterization</th>
<th>To: Parameterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda, \alpha, \beta, \delta, \mu ), ((\alpha &gt; 0, \delta &gt; 0))</td>
<td>( \lambda, \alpha, \beta, \delta, \mu ), ((\alpha &gt; 0, \delta &gt; 0))</td>
</tr>
<tr>
<td>( \lambda, \rho, \zeta, \delta, \mu ), ((\zeta &gt; 0, \delta &gt; 0))</td>
<td>( \lambda, \beta, \delta, \mu ), ((\beta &gt; 0, \delta &gt; 0))</td>
</tr>
<tr>
<td>( \lambda, \xi, \delta, \mu ), ((\xi &gt; 0, \delta &gt; 0))</td>
<td>( \lambda, \xi, \beta, \delta, \mu ), ((\alpha &gt; 0, \delta &gt; 0))</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\alpha &= \delta \sqrt{1 - \rho^2} \\
\beta &= \rho \alpha \\
\zeta &= \frac{\beta}{\delta} \\
\rho &= \frac{\beta}{\alpha} \\
\chi &= \frac{\beta}{\alpha} \sqrt{1 + \sqrt{1 + \delta^2 \alpha^2 - \beta^2}} \\
\nu &= \frac{\beta}{\alpha} \\
\end{align*}
\]
1.2.1 The hyperbolic distribution

Barndorff-Nielsen (1977) established the density of the hyperbolic distribution analytically by modelling the shape of the plot of mass-size distribution of sand transportation. However, the density of the hyperbolic distribution can also be obtained as a special case of the GH.

**Definition 1.** For \( \lambda = 1 \), a subclass of the GH (1.2), called the hyperbolic distribution, is obtainable as

\[
\text{HYP}(x; \alpha, \beta, \delta, \mu) = \frac{\sqrt{\alpha^2 - \beta^2}}{2\delta \alpha K_1\left(\frac{\delta}{\sqrt{\alpha^2 - \beta^2}}\right)} \exp\left\{-\alpha\sqrt{\delta^2 + (x - \mu)^2} + \beta(x - \mu)\right\},
\]

where \( x \in \mathbb{R}, \mu \in \mathbb{R}, \delta \geq 0 \) and \( \alpha > |\beta| \). Using (1.3), the density function (1.4) can be rewritten as

\[
\text{HYP}(x; \rho, \zeta, \delta, \mu) = a(\rho, \zeta) \exp\left\{-\zeta \left[\sqrt{1 + \rho^2} \sqrt{1 + ((x - \mu)/\delta)^2 - \rho(x - \mu)/\delta}\right]\right\}
\]

where

\[
a(\rho, \zeta) = \left[2 \delta (1 + \rho^2) K_1(\zeta)\right]^{-1}.
\]

Table 1.3 shows the relationship between the available parameterizations. Generally, Parameterization 2 offers the simplest form of the exponent. Parameterizations 2 and 3 show that the hyperbolic distributions belong to a regular exponential family of distributions. For parameter estimation purposes, the \((\rho, \zeta)\) parameterization appears to be preferred because the restriction of being positive is only placed on \( \zeta \). However, the \((\alpha, \beta)\) parameterization was also used for this purpose in Eberlein and Keller (1995). Furthermore, the joint domain of variation of the shape parameters \( 0 \leq |\chi| < \xi < 1 \) in Parameterization 4 implies a feature called the shape triangle in Figure 1.1. This plot reproduces Figure 6 of Eberlein and Keller (1995). Here, N: normal distribution, GIG(1): GIG with \( \lambda = 1 \), L: skewed and
symmetrical Laplace distribution, \( E \): exponential distribution.

Figure 1.1: Plot of the log-probability densities function of the hyperbolic distribution on the shape triangle.
1.3 The multivariate generalized hyperbolic distributions

Similar to the univariate hyperbolic family of distributions, the multivariate generalized hyperbolic (MGH) distribution is obtainable via normal mean-variance mixture of the multivariate \(d\)-dimensional normal distributions and the GIG as the mixing distribution, see Barndorff-Nielsen (1977). The random vector \(X\) is said to have a multivariate normal mean-variance mixture if

\[
X \overset{d}{=} \mu + W\gamma + \sqrt{W}AZ,
\]

where

1. \(Z \sim N_k(0, I_k)\),

2. \(W \geq 0\) is a non-negative, scalar-valued random variable independent of \(Z\),

3. \(A \in \mathbb{R}^{d \times k}\) is a matrix, and

4. \(\mu\) and \(\gamma\) are parameter vectors in \(\mathbb{R}^d\).

Alternatively, it can be written as

\[
X|W = w \sim N_d(\mu + W\gamma, W\Sigma), \quad \Sigma = AA'.
\]

The joint density of \(d\)-dimensional generalized hyperbolic distributions in the non-singular case (\(\Sigma\) has rank \(d\)) is given by

\[
\text{MGH}_d(x; \lambda, \chi, \psi, \mu, \Sigma, \gamma) = \int_0^\infty \frac{1}{(2\pi |\Sigma| w)^{d/2}} \times \\
\exp \left\{ -\frac{(x - \mu - w\gamma)'(w\Sigma)^{-1}(x - \mu - w\gamma)}{2} \right\} h(w) \, dw
\]

(1.6)
To: Parameterization

<table>
<thead>
<tr>
<th>From: Parameterization</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho, \zeta, \delta, \mu$</td>
<td>$\alpha, \beta, \delta, \mu$</td>
<td>$\phi, \gamma, \delta, \mu$</td>
<td>$\xi, \chi, \delta, \mu$</td>
<td></td>
</tr>
<tr>
<td>$\rho, \zeta, \delta, \mu$</td>
<td>$\alpha = \frac{\zeta \sqrt{1 + \rho^2}}{\delta}$</td>
<td>$\phi = \frac{\zeta}{\delta \sqrt{1 + \rho^2 + \rho}}$</td>
<td>$\xi = \frac{1}{\sqrt{1 + \zeta}}$</td>
<td></td>
</tr>
<tr>
<td>$(\zeta &gt; 0, \delta &gt; 0)$</td>
<td>$\beta = \frac{\zeta}{\delta}$</td>
<td>$\gamma = \frac{\zeta}{\delta \sqrt{1 + \rho^2 - \rho}}$</td>
<td>$\chi = \frac{\rho}{\sqrt{(1 + \zeta)(1 + \rho^2)}}$</td>
<td></td>
</tr>
<tr>
<td>$\alpha, \beta, \delta, \mu$</td>
<td>$\phi = \alpha + \beta$</td>
<td>$\gamma = \alpha - \beta$</td>
<td>$\xi = \frac{\beta}{\sqrt{1 + \delta \sqrt{\alpha^2 - \beta^2}}}$</td>
<td></td>
</tr>
<tr>
<td>$(\alpha &gt; 0, \delta &gt; 0)$</td>
<td>$\zeta = \delta \sqrt{\alpha^2 - \beta^2}$</td>
<td>$\chi = \frac{\beta}{\alpha \sqrt{1 + \delta \sqrt{\alpha^2 - \beta^2}}}$</td>
<td>$\zeta = 1$</td>
<td></td>
</tr>
<tr>
<td>$\phi, \gamma, \delta, \mu$</td>
<td>$\rho = \frac{\phi - \gamma}{\sqrt{\phi^2}}$</td>
<td>$\alpha = \frac{\phi + \gamma}{2}$</td>
<td>$\xi = \frac{1}{\sqrt{1 + \delta \sqrt{\phi^2}}}$</td>
<td></td>
</tr>
<tr>
<td>$(\phi &gt; 0, \gamma &gt; 0, \delta &gt; 0)$</td>
<td>$\zeta = \delta \sqrt{\phi^2}$</td>
<td>$\beta = \frac{\phi - \gamma}{2}$</td>
<td>$\chi = \frac{\phi - \gamma}{(\phi + \gamma) \sqrt{1 + \delta \sqrt{\phi^2}}}$</td>
<td></td>
</tr>
<tr>
<td>$\xi, \chi, \delta, \mu$</td>
<td>$\rho = \frac{\chi}{\sqrt{\xi^2 - \chi^2}}$</td>
<td>$\alpha = \frac{1 - \xi^2}{\delta \xi^2 \sqrt{\xi^2 - \chi^2}}$</td>
<td>$\phi = \frac{(1 - \xi^2)(\xi + \chi)}{\delta \xi^2 \sqrt{\xi^2 - \chi^2}}$</td>
<td></td>
</tr>
<tr>
<td>$(\xi &gt; 0, \delta &gt; 0)$</td>
<td>$\zeta = \frac{1 - \xi^2}{\xi^2}$</td>
<td>$\beta = \frac{\chi(1 - \xi^2)}{\delta \xi^2 \sqrt{\xi^2 - \chi^2}}$</td>
<td>$\gamma = \frac{(1 - \xi^2)(\xi - \chi)}{\delta \xi^2 \sqrt{\xi^2 - \chi^2}}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.3: The relations between four parameterizations of the hyperbolic distribution.
where \( h(w) \) is the density of \( W \sim \text{GIG}(w; \lambda, \chi, \psi) \). Applying (1.6), Wenbo (2005) gave

\[
\text{MGH}_d(x; \lambda, \chi, \psi, \mu, \Sigma, \gamma) = \frac{\sqrt{\chi \psi}^{-\lambda} \psi^\lambda (\psi + \gamma' \Sigma^{-1} \gamma)^{d/2-\lambda}}{(2\pi)^{d/2} |\Sigma|^{1/2} K_\lambda(\sqrt{\chi \psi})} \exp\left\{ (x - \mu)' \Sigma^{-1} \gamma \right\} \times \\
\frac{K_{\lambda-d/2} \left( \sqrt{[\chi + (x - \mu)' \Sigma^{-1} (x - \mu)] [\psi + \gamma' \Sigma^{-1} \gamma]} \right)}{\left( \sqrt{[\chi + (x - \mu)' \Sigma^{-1} (x - \mu)] [\psi + \gamma' \Sigma^{-1} \gamma]} \right)^{d/2-\lambda}}.
\]

(1.7)

The MGH appears in the literature with numerous parameterizations, but the one presented in Barndorff-Nielsen (1977) is most commonly found. This is because it is the parametrization implemented in the \texttt{hyp} program and it is consistent with the description of the GH in Barndorff-Nielsen and Blæsild (1981). The multivariate generalized hyperbolic family is extremely flexible and it contains many subclasses known by different names (cf. p.80 of McNeil et al., 2005). This thesis focusses on the following:

1. If \( \lambda = \frac{d+1}{2} \) then the word “generalized” is dropped and the MGH is referred to as a \( d \)-dimensional hyperbolic distribution (\( d \)-DHYP). Note that univariate margins of this distributions have \( \lambda = \frac{d+1}{2} \) and are not one-dimensional hyperbolic distributions. The routine to fit this subclasses of the MGH include \texttt{fit.hypmv()} (in \texttt{ghyp}). If \( \lambda = 1 \) then a subclass of the MGH called multivariate hyperbolic distribution (MHYP) is formed. In this case, the univariate margins is one-dimensional hyperbolic distributions. The function to fit the MHYP include \texttt{fit.mNH()} (in \texttt{QRMlib}).

2. When \( \lambda = -\frac{1}{2} \) the MGH is known as the multivariate normal inverse Gaussian (MNIG) distribution. The functions \texttt{fit.NIGmv()} (in \texttt{ghyp}) and \texttt{fit.mNH()} fit the MNIG.
1.4 Parameter estimation of the subclasses of the GH distributions

This thesis only concerns parameter estimation of the HYP, $d$-DHYP and MNIG, which are all subclasses of the GH and MGH. A question to be raised is why not estimate $\lambda$ like other parameters of the GH and MGH rather than fixing $\lambda$ to a specific value for parameter estimation purposes? In other words, why not obtain parameter estimation of all parameters of the GH and MGH rather than those of their subclasses? Regarding to the parameter estimation results of this thesis, one of the referees commented that “This is fitting only a couple of very particular cases of the GH, with $\lambda$ known”. Theoretically, maximum-likelihood estimation of $\lambda$ is obtainable like those of other parameters of the GH (cf. p.11 of Prause 1999) or the MGH (cf. p.82 of McNeil et al. 2005). However, Prause (1999) pointed out on p.8 that “The variation in the likelihood function of the GH distribution is small for a wide range of parameters. Consequently, the generalized hyperbolic distribution applied as a model for financial data leads to overfitting.” This observation is confirmed by Figure 2.7 of Wenbo (2005), which plotted calibrated maximum log-likelihood against the value of $\lambda$. The plot shows the maximum log-likelihood value changes from $-1041$ to $-1047$ as $\lambda$ ranges from $-10$ to 10. These studies illustrate that fitting the GH rather than its subclasses is likely to worsen a well-known problem in parameter calibration for these family of distributions, viz. the flatness of the log-likelihood function. This problem is even troublesome when fitting the HYP and it was noticed on p.29 of Barndorff-Nielsen and Blæsild (1981). From practical point of view, it was suggested on p.9 of Prause (1999) that “For numerical reasons it is useful to find a suitable subclass of the GH distributions” when fitting the GH to financial returns data. These may explain the reason why theoretical steps for fitting the GH and MGH (i.e., $\lambda$ unknown) have been well-established but McNeil and Ulman (2010) only implemented the fitting of the subclasses (i.e., $\lambda$ known).
of these distributions. Specifically, the R package \texttt{QRMlib} includes \texttt{fit.NH()} to fit HYP, NIG and \texttt{fit.mNH()} for MHYP, MNIG. Following the McNeil’s approach in package \texttt{QRMlib} version 1.4.5, this thesis focuses on the fitting of the HYP, MHYP and MNIG. Note that the S-plus\textsuperscript{2} version of \texttt{QRMlib} also includes parameter estimation of the subclasses of the GH and MGH only. However, it is noted that the package \texttt{ghyp} (version 1.5.4) include the functions \texttt{fit.ghypmv()} and \texttt{fit.ghypuv()} to fit the MGH and GH respectively. The extension of this thesis results to fit the MGH and GH can be found in Chapter 8.

1.5 Thesis outline

This thesis consists of eight chapters:

- Chapter 1 is a literature review.

- Chapter 2 concerns the asymptotic approximation of the modified Bessel function of the second kind $K_\lambda(z)$, for sufficiently large $z \in \mathbb{R}^+$ and $\lambda \in [-90, 90]$. This chapter derives a robust algorithm to answer the questions in the literature about the number of terms required and the magnitude of $z$ for the asymptotic approximation to give a specific level of accuracy. For $0 \leq |\lambda| < 34$, the approximation accuracy is $\text{Machine\$double\.eps}^{0.95}$. In R, $\text{Machine\$double\.eps}$ is the smallest positive floating-point number $x$ such that $1+x \neq 1$; for a 32-bit machine it is normally $2.220446 \times 10^{-16}$. The algorithm can be straightforwardly coded in other software packages such as Maple. In these cases, the approximation is even more accurate because Maple has a significantly smaller machine accuracy number than the number in R. The approximation of $K_1(z)$ for small values of $z$ is also addressed by implementing the work in \cite{Harris2009}. The accuracy of the approximation routines of this chapter was compared against the “correct” evaluation of $K_\lambda(z)$ using Maple. For the tested

\textsuperscript{2}S-plus is a commercial statistical software.
values of $\lambda$ and $z$, the approximate and “correct” values agree up 15 decimal places. This level of accuracy even applies to cases where $K_\lambda(z)$ assumes extreme values. For $K_1(z)$ with $z \geq 20$, the approximation routine which is based on the asymptotic expansion of this chapter is more accurate than the approximation proposed in [Harris (2009)].

- Chapter 3 is reserved for parameter estimation of the hyperbolic and 3 subclasses of the generalized inverse Gaussian distributions by symbolically deriving explicit formulae for the maximum-likelihood estimators. For sample sizes less than 100 observations, the existing numerical approaches in the literature are unstable because of difficulties in obtaining “correct” starting values for it to converge. However, the symbolic fitting routines are stable even for sample size of 20 observations because they do not require starting values. However, the symbolic fitting of the HYP is restricted by a set of assumptions. The numerical fitting of the so-called standardized hyperbolic distribution was implemented to work with cases where these assumptions are relaxed. The computational speed of this function is faster than the existing fitting function for non-standardized hyperbolic distributions.

- Chapter 4 concerns the fitting of the $d$-DHYP ($d = 2$ and 4) and multivariate normal inverse Gaussian distributions with any arbitrary dimensions using the EM algorithm. The obtained routines converge more quickly and are more stable than the fitting routines in package QRMIlib version 1.4.5. These improvements are obtainable because the maximization of the mixing distribution (i.e., the GIG) is carried out symbolically whereas this task has only been carried out numerically in the literature.

- Chapter 5 proposes three different methods to evaluate three integral representations of the incomplete Bessel functions. Two of these methods are analytic which allow the derivation of closed-formed formulae for the incom-
plete Bessel functions. The accuracy of these formulae was then numerically examined using Maple, Mathematica and R, the results were accurate up to machine accuracy. The chapter also implements an algorithm by Slevinsky and Safouhi (2010) to numerically approximate an incomplete Bessel function for any value of $\lambda \in \mathbb{R}$. After being tested extensively, the routine was accurate for a range of $\lambda$. However, tests also showed that it has serious computational problem which had not been discussed by the authors. This chapter proposes a “diagnosis” tool for these issues and a Maple procedure to provide accurate value of the incomplete Bessel function.

- Chapter 6 derives explicit distribution functions of two special cases of the GIG for $\lambda = \pm \frac{1}{2}$. The distribution function (CDF) of the GIG when $\lambda = -\frac{1}{2}$ has been derived by three different authors. However, the method proposed in this thesis is completely different with theirs, in the sense that it involves direct analytical evaluation of the incomplete Bessel function. Explicit distribution function of the GIG for $\lambda = \frac{1}{2}$ and complementary distribution functions (CCDF) of the GIG when $\lambda = \pm \frac{1}{2}$ have never been obtained previously. A numerical approach for approximating distribution functions of the GIG for any value of $\lambda \in \mathbb{R}$ was also implemented as an R function. This function works for a range of $\lambda$ but it is not reliable because of the computational problems it inherits from the numerical approach to evaluate the incomplete Bessel function.

- Chapter 7 examines goodness-of-fit tests for the hyperbolic distribution using the Cramér-von Mises and Moran log-spacing tests. The chapter extends the table of precalculated value of the Cramér-von Mises test statistic. A fitting routine based on multivariate splines was obtained using this table. This routine is then used to approximate accurate $p$-value for the test. The Moran goodness-of-fit test was also implemented. However, numerical tests
show that the power of the Moran goodness-of-fit test is low compared to the Cramér-von Mises test.

- Chapter 8 gives conclusions and future research proposals.

The notation $\mathbb{Z}^+$ and $\mathbb{Z}^*$ indicate the set of positive and non-negative integers, while $\mathbb{Z}$ is the set of all integers. All R routines in this thesis were tested on a PC which ran a 32-bit Windows Vista operating system. The processor was an Intel dual-core with speed 2.26 GHz and a 4 GB RAM.
Chapter 2

Asymptotic approximations of the modified Bessel function of the second kind

2.1 Introduction

Asymptotic approximation is an important topic in applied analysis and its applications dominate many fields in science and engineering. Major references include Dingle (1973), Wong (1989) and Olver (1997). Asymptotic expansion of the modified Bessel function of the second kind is well-known. Some of the standard references are Whittaker (1902), Gray and Mathews (1931), Jeffreys (1962) and Abramowitz and Stegun (1970). Most recently, the function `besselK.nuAsym(z, λ)` in the R package `Bessel`, see Maechler (2011), is to approximate $K_\lambda(z)$ using the so-called Debye polynomials.

This chapter has two purposes. Firstly, for a given value of $\lambda$, it derives a routine which approximates the sufficiently large value of $z$ so that the accuracy of the approximation of $K_\lambda(z)$ improves when more terms are added to the asymptotic approximation series. Given the sufficiently large value of $z$, the second purpose is to
obtain a routine which answers the questions in the literature about approximation accuracy and the number of terms required. For completeness, the method in [Harris (2009)] was implemented to approximate $K_1(z)$ when $z$ is small. Following the convention on p.519 of [Olver (1997)], the terms asymptotic approximations and expansions are used exchangeably to refer to the topic concerned in this chapter.

2.1.1 Modified Bessel function of the second kind

This Bessel function is the solution of a differential equation given by

$$z^2 \frac{d^2 w}{dz^2} + z \frac{dw}{dz} + (z^2 - u^2) w = 0.$$  

(2.1)

[Gray and Mathews (1931)] proved (2.1) to be a special case of Laplace’s equation

$$(a z + a') \frac{d^2 w}{dz^2} + (b z + b') \frac{dw}{dz} + (c z + c') = 0,$$

and they solved the equation by letting

$$a = 1, \quad a' = 0, \quad b = 1, \quad b' = 0, \quad c = 1, \quad c' = 0.$$  

One of the solutions is the modified Bessel function of the second kind, whose integral representation, among others, is given by

$$K_\lambda(z) = \sqrt{\frac{\pi}{2z}} \frac{1}{\Gamma(\lambda + \frac{1}{2})} e^{-z} \int_0^\infty e^{-\xi} \xi^{\lambda-1/2} \left(1 + \frac{\xi}{2z}\right)^{-\lambda-1/2} d\xi$$  

(2.2)

where $z > 0$, $\lambda \in \mathbb{R}$ and $\lambda + \frac{1}{2} > 0$. This thesis concerns $K_\lambda(z)$ only.

2.1.2 Asymptotic approximations of $K_\lambda(z)$ in the literature

Before a precise definition of asymptotic expansion is given, the following example and related discussion are taken from [Olver (1997)] to illustrate the concept.
Consider the integral
\[ I(x) = \int_0^\infty \frac{e^{-xt}}{1+t} \, dt \]  
(2.3)
which can be written as
\[ I(x) = \int_0^\infty e^{-xt} (1 - t + t^2 - \cdots) \, dt \]
\[ = \frac{1}{x} - \frac{1}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots. \]  
(2.4)
This series diverges for all finite values of \( x \) and therefore appears to be meaningless.
However, if a heuristic approach is adopted to sum the series (2.4) for a particular value of \( x \), say \( x = 10 \). The first four terms are given by
\[ 0.1000 - 0.0100 + 0.0020 - 0.0006 = 0.0914, \]  
(2.5)
which is very close to the correct value \( G(10) = 0.09156333\ldots \) obtained by numerically integrating (2.3) using Mathematica.

Consider the difference \( R_s(x) \) between \( I(x) \) and the \( s \)th partial sum of (2.3), given by
\[ R_s(x) = I(x) - g_s(x), \]
where
\[ g_s(x) = \frac{1}{x} - \frac{1}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots + (-1)^{s-1} \frac{(s-1)!}{x^s}. \]
Here \( s \in \mathbb{Z}^+ \) (\( s = 1, 2, 3, \ldots \)), and \( R_s(x) \) is called the remainder of the partial series. Since
\[ \frac{1}{1+t} = 1 - t + t^2 - t^3 + \cdots + (-1)^{s-1} t^{s-1} + \frac{(-1)^s t^s}{1+t}, \]
substitution in (2.3) yields
\[ R_s(x) = (-1)^s \int_0^\infty \frac{t^s e^{-xt}}{1+t} \, dt. \]
Clearly,

$$|R_s(x)| < \int_0^\infty t^s e^{-xt} \, dt = \frac{s!}{x^{s+1}}.$$  

In other words, the partial sum of (2.4) approximates the function $I(x)$ with an error that is numerically smaller than the first neglected term of the series. Because the next term in (2.5) is 0.00024, it fully explains the closeness of the value 0.0914 of $g_4(10)$ to that of $G(10)$. Given functions $f(z)$ and $g(z)$, where $g(z_0) \neq 0$, the notation $f(z) \sim g(z)$ is to denote $f(z)/g(z)$ approaches unity as $z \to z_0$.

**Definition 2.1.1** (Poincaré). A power series

$$\sum_{s=0}^{\infty} \alpha_s z^{-s},$$

convergent or divergent, is said to be an asymptotic expansion of a defined function $f(z)$ if the expression

$$\varepsilon_s(z) = z^s [f(z) - G_s(z)]$$

$$= z^s R_s(z)$$

satisfies

$$\lim_{|z| \to \infty} \varepsilon_s(z) = 0$$

(2.6)

where

$$G_s(z) = \sum_{s=0}^{s-1} \alpha_s z^{-s}.$$  

In this case, $G_s(z)$ and $R_s(z)$ are called the asymptotic expansion and the remainder of the asymptotic approximations of function $f(z)$ respectively. Notationally, it is written as

$$f(z) \sim \sum_{s=0}^{\infty} \alpha_s z^{-s}. $$

(2.7)

Asymptotic expansion of the Bessel function is well-known in the literature.
Entry 9.7.2 of [Abramowitz and Stegun (1970)] gives

\[ K_\lambda(z) \sim \sqrt{\frac{\pi}{2z}} e^{-z} \left\{ 1 + \frac{4\lambda^2 - 1^2}{1! 8z} + \frac{(4\lambda^2 - 1^2)(4\lambda^2 - 3^2)}{2! (8z)^2} + \frac{(4\lambda^2 - 1^2)(4\lambda^2 - 3^2)(4\lambda^2 - 5^2)}{3! (8z)^3} + \cdots \right\}, \quad z \in \mathbb{R}^+, \quad (2.8) \]

upon the condition that \( z \) is large. Because of this condition, a question to be raised is what value of \( z \) should be considered as large? Moreover, it was stated on p.55 of [Gray and Mathews (1931)] that “the series (2.8), if regarded as an infinite series, is divergent. However, by making \( z \) large enough the remainder can be made indefinitely small; so that, for large values of \( z \) a finite number terms will give an approximate value of \( K_\lambda(z) \)”. In addition, it was also pointed out on p.721 of [Kreyszig (2006)] that “series (2.8) is asymptotic in the sense that for large enough \( z \), \( K_\lambda(z) \) may be approximated to any fixed degree of accuracy with a small number of terms”.

For a given value of \( \lambda \), these statements raise two questions. Firstly, what is this finite number of terms? Secondly what is the approximation error?

### 2.1.3 Applications of asymptotic approximations of \( K_\lambda(z) \) in statistics

Asymptotic approximations of the Bessel function have important applications in statistics because \( K_\lambda(z) \) appears in the density function of the GH and its related distributions such as the HYP or the NIG. [Eberlein and von Hammerstein (2004)] examined the asymptotic properties of \( K_\lambda(z) \) to obtain the limit distributions when the parameters of the GH tend to or lie on the boundaries of the shape triangle 1.1. [Eberlein and Keller (1995)] studied the asymptotic features of the modified Bessel function with applications in stochastic processes to value financial options. [Barndorff-Nielsen (1997)] studied the Cauchy distribution as a special case of the NIG by applying the asymptotic approximation of the Bessel function as the parameters
of the former tend to infinity. Applications of the asymptotic approximation of the
Bessel function in relation to the multivariate hyperbolic distribution can be found
in Schmidt et al. (2003) and Barndorff-Nielsen and Blæsild (1981). In these studies,
the form of asymptotic approximation used is

$$K_{\lambda}(z) \sim \sqrt{\frac{\pi}{2z}} e^{-z}, \quad z \to \infty.$$ 

As can be seen, this approximation does not give an approximation error nor the
values of the parameters of the distribution for which it can be used.

This chapter can be outlined as follows. Section 2.2 derives the asymptotic
expansion of $K_{\lambda}(z)$ when $z$ is large. The remainder term of the asymptotic expansion
derived in this chapter is more readily shows important features of the asymptotic
expansion of $K_{\lambda}(z)$ than the remainder appears on p.55 of Gray and Mathews (1931).
Section 2.3 is to derive a robust algorithm and implement it as an R function to
approximate sufficiently large value of $z$, the number of terms required and the
approximation accuracy of $K_{\lambda}(z)$ for $\lambda \in [-90, 90]$. Section 2.4 implements the
approximation $K_{1}(z)$ when $z$ is small by applying the results presented in Harris
(2009). Section 2.5 examines the accuracy of the obtained routine by comparing the
approximation results with those returned by Maple 15, besselK.nuAsym($z, \lambda$) and
the method in Harris (2009). Section 2.6 summarizes the chapter.

### 2.2 Asymptotic expansion series of $K_{\lambda}(z)$

This section is to obtain an asymptotic expansion of the Bessel function which agrees
with the already derived expansion in the literature. However, the expression of the
remainder here is simpler than the one given on p.55 of Gray and Mathews (1931),
and it readily shows the important features of asymptotic approximation of $K_{\lambda}(z)$.
2.2.1 Derivation

Letting \( h = \lambda - \frac{1}{2} \), equation (2.2) can be rewritten as

\[
K_\lambda(z) = \sqrt{\frac{\pi}{2z}} \frac{1}{\Gamma(\lambda + \frac{1}{2})} e^{-z} f(h) \tag{2.9}
\]

where

\[
f(h) = \int_0^\infty e^{-\xi} \xi^h \left(1 + \frac{\xi}{2z}\right)^h d\xi. \tag{2.10}
\]

Using Whittaker (1903), the binomial expansions of the term in parentheses is

\[
\left(1 + \frac{\xi}{2z}\right)^h = 1 + \frac{h \xi}{2z} + \frac{h(h-1)}{2!} \left(\frac{\xi}{2z}\right)^2 + \cdots + \frac{h(h-1) \cdots (h-s+1)}{s!} \left(\frac{\xi}{2z}\right)^s + \\
\frac{h(h-1) \cdots (h-s)}{s!} \int_0^{\xi/2z} \left(\frac{\xi}{2z} - t\right)^s (1 + t)^{h-s-1} dt. \tag{2.11}
\]

Inserting (2.11) into (2.10) to have

\[
f(h) = \int_0^\infty e^{-\xi} \xi^h d\xi + \sum_{r=1}^{s-1} \left\{ \frac{h(h-1) \cdots (h-r+1)}{r!} \left(\frac{1}{2z}\right)^r \int_0^\infty e^{-\xi} \xi^{h+r} d\xi \right\} + \\
\frac{h(h-1) \cdots (h-s)}{s!} \int_0^\infty e^{-\xi} \xi^h \left[ \int_0^{\xi/2z} \left(\frac{\xi}{2z} - t\right)^s (1 + t)^{h-s-1} dt \right] d\xi
\]

\[
= \sum_{r=0}^{s-1} \left\{ \frac{\Gamma(h+1)}{\Gamma(h-r+1)} \left(\frac{1}{r!}\right) \left(\frac{1}{2z}\right)^r \int_0^\infty e^{-\xi} \xi^{h+r} d\xi \right\} + R_s, \tag{2.12}
\]

Here, the remainder

\[
R_s = \frac{h(h-1) \cdots (h-s)}{s!} \int_0^\infty e^{-\xi} \xi^h \left\{ \int_0^{\xi/2z} \left(\frac{\xi}{2z} - t\right)^s (1 + t)^{h-s-1} dt \right\} d\xi.
\]

Equation (2.12) implies that \( f(h) \) can be written as a summation of \( G_s \) and the remainder \( R_s \). Letting \( t = v (2z)^{-1} \) and change of variable give

\[
R_s = \frac{h(h-1) \cdots (h-s)}{s!} \int_0^\infty e^{-\xi} \xi^h \left[ \int_0^\xi (\xi - v)^s (1 + \frac{v}{2z})^{h-s-1} dv \right] d\xi.
\]

40
In order to show that $G_s$ is the asymptotic expansion of $f(h)$, the critical point is to demonstrate that $R_s \to 0$ as $z \to \infty$.

**Proof.** As $z$ becomes large, $s$ having any definite finite integer value, the remainder term $R_s$ (cf. p.293 of Whittaker, 1902) is given by

$$R_s = \frac{h(h-1)\cdots(h-s)}{s!} \left( \frac{1}{2z} \right)^{s+1} \int_0^\infty e^{-\xi} \xi^h d\xi \int_0^\xi (\xi - v)^s dv$$

$$= \frac{h(h-1)\cdots(h-s)}{(s+1)!} \left( \frac{1}{2z} \right)^{s+1} \int_0^\infty e^{-\xi} \xi^{h+s+1} d\xi$$

$$= \frac{h(h-1)\cdots(h-s) \Gamma(h+s+2)}{(s+1)!} \left( \frac{1}{2z} \right)^{s+1}.$$  \hspace{1cm} (2.13)

It then follows that $\lim_{z \to \infty} z^s R_s = 0$ as required by (2.6).

This means that the series $G_s$ on the RHS of (2.12) is the asymptotic expansion of $f(h)$. The remaining step is to obtain the asymptotic expansion of $K_{\lambda}(z)$ by substituting (2.12) into (2.9), which gives

$$K_{\lambda}(z) = \sqrt{\frac{\pi}{2z}} e^{-z} \frac{1}{\Gamma \left( \lambda + \frac{1}{2} \right)} \left\{ \sum_{r=0}^{s-1} \left[ \frac{\Gamma(h+1)}{\Gamma(h-r+1)} \left( \frac{1}{2} \right)^r \left( \frac{1}{2z} \right)^r \int_0^\infty e^{-\xi} \xi^{h+r} d\xi \right] + R_s \right\}.$$ \hspace{1cm} (2.14)

Replacing $h = \lambda - \frac{1}{2}$ and evaluating the integral in (2.14) applying

$$\int_0^\infty e^{-\xi} \xi^{\lambda+r-1/2} d\xi = \Gamma \left( \lambda + r + \frac{1}{2} \right),$$

which is the gamma function (cf. p.255 of Abramowitz and Stegun, 1970) to obtain

$$K_{\lambda}(z) = \sqrt{\frac{\pi}{2z}} e^{-z} \sum_{r=0}^{s-1} \left\{ \frac{\Gamma(h+1)}{\Gamma(h-r+1)} \left( \frac{1}{2} \right)^r \left( \frac{1}{2z} \right)^r \right\} + R_s$$

$$= \sqrt{\frac{\pi}{2z}} e^{-z} \left\{ 1 + \frac{4\lambda^2 - 1}{118z} + \frac{(4\lambda^2 - 12)(4\lambda^2 - 32)}{2! (8z)^2} + \cdots + \frac{(4\lambda^2 - 12)(4\lambda^2 - 32)\cdots(4\lambda^2 - (2s-3)^2)}{(s-1)! (8z)^{s-1}} \right\} + R_s,$$ \hspace{1cm} (2.15)
where

\[
R_s = \sqrt{\frac{\pi}{2z}} e^{-z} \left\{ \left( \lambda - \frac{1}{2} \right) \left( \lambda - \frac{3}{2} \right) \ldots \left( \lambda - \frac{1}{2} - s \right) \Gamma \left( \lambda + \frac{3}{2} + s \right) \left( \frac{1}{2z} \right)^{s+1} \right\}. \tag{2.16}
\]

Since \( \lim_{z \to \infty} z^s R_s = 0 \) the asymptotic expansion of \( K_{\lambda}(z) \) is given by

\[
K_{\lambda}(z) \sim \sqrt{\frac{\pi}{2z}} e^{-z} \left\{ 1 + \frac{4\lambda^2 - 1^2}{1! 8z} + \frac{(4\lambda^2 - 1^2)(4\lambda^2 - 3^2)}{2! (8z)^2} + \cdots + \frac{(4\lambda^2 - 1^2)(4\lambda^2 - 3^2) \ldots (4\lambda^2 - (2s - 3)^2)}{(s-1)! (8z)^{s-1}} \right\}, \tag{2.17}
\]

which agrees with the asymptotic expansion series of \( K_{\lambda}(z) \) on p.55 of Gray and Mathews (1931). Note that \( R_s \) is obtainable by substituting \( h = \lambda - \frac{1}{2} \) into \( R_s \).

In order to address a referee’s request for stating theoretical error bounds of the expansion (2.17) and the conditions on which these bounds apply, the statement on p.378 of Abramowitz and Stegun (1970) is used: “If \( z \) is sufficiently large and \( \lambda \leq s + \frac{1}{2} \) then the remainder after \( s \) terms in the expansion (2.17) does not exceed the \( (s + 1) \)th term in absolute value and of the same size.” When series (2.17) is written down explicitly for \( s = 4 \), it agrees with entry 9.7.2 of Abramowitz and Stegun (1970).

2.2.2 Features of the remainder

From (2.16) there are two readily observed important features.

1. Given a value of \( j \in \mathbb{Z}^* \) \((j = 0, 1, 2, 3, \ldots)\), if \(|\lambda| = j + \frac{1}{2}\) then the remainder \( R_s \) (2.16) equals 0 because its numerator equals 0. Consequently, the sign “=” replaces “∼” in (2.17). For these values of \( \lambda \), if \( h = \lambda - \frac{1}{2} \) is substituted into (2.13) then \( R_s \) is also 0 because the numerator of (2.13) equals 0.

2. Because \( \lim_{z \to \infty} z^s R_s = 0 \) regardless of the sign of \( \lambda \), (2.16) holds when \( \lambda \) is negative. This means that the results of this chapter are applicable to any
cases where $\lambda \in \mathbb{R}$ and $z$ is sufficiently large. This feature implies the benefit of using asymptotic approximations over the Debye polynomials because the latter are only applicable for $\lambda \in \mathbb{R}^+$.

Note that separate proofs are required to show each of these features if the remainder on p.55 of Gray and Mathews (1931) is used.

### 2.2.3 Properties of the asymptotic approximations

The properties of asymptotic expansion (2.17) include subtraction, addition, division, multiplication, integration. Moreover, “An important property of the Poincaré asymptotic expansion is that the expansion, if it exists, is unique.” (cf. p.21 of Wong, 1989). This has a very important implication in statistics because it ensures that if $K_\lambda(z)$ in the density function of a statistical distribution is replaced by its asymptotic expansion (2.17) with a finite number of terms then the approximate density function is unique. It is noted that two different functions can have the same asymptotic expansion but that the uniqueness feature is not necessarily true for the remainder. This can be illustrated by the existence of the remainder in the asymptotic expansion of Taylor series, which can be found in different forms in the literature.

Differentiation is not always legitimate for asymptotic expansions. An example is if $f(x) = e^{-x} \sin(e^x)$ and $x$ is real and positive then

$$f(x) \sim 0 + \frac{0}{x} + \frac{0}{x^2} + \cdots, \quad x \to \infty,$$

(2.18)

but $f'(x) = \cos(e^x) - e^{-x} \sin(e^x)$ oscillates as $x \to \infty$ and therefore has no asymptotic approximation of the form (2.18) (cf. p.21 of Olver, 1997).

Differentiation of the asymptotic approximation of the Bessel function in (2.17) is valid because of Theorem 4.2 on p.9 and further discussion on p.21 of Olver (1997), which allows for (2.17) to be differentiated any number of times. The first
derivative of (2.17) is given by entry 9.7.4 of [Abramowitz and Stegun (1970)].

The uniqueness and differentiation features allow for the use of (2.17) with a finite number of terms to form the approximate log-likelihood function for the purpose of finding maximum-likelihood estimates in the next chapter.

2.3 An algorithm to approximate the Bessel function when $|\lambda|$ is not half of an odd integer

This section derives an algorithm to address the questions raised in Section (2.1.2) when $|\lambda|$ is not equal half of an odd integer and $\lambda \in [-90, 90]$. Letting

$$^sK_\lambda(z) = \sqrt{\frac{\pi}{2z}} e^{-z} \left\{ 1 + \frac{4\lambda^2 - 1^2}{1!8z} + \frac{(4\lambda^2 - 1^2)(4\lambda^2 - 3^2)}{2!(8z)^2} + \cdots + \frac{(4\lambda^2 - 1^2)(4\lambda^2 - 3^2) \cdots \{4\lambda^2 - (2s - 3)^2\}}{(s-1)!(8z)^{s-1}} \right\}$$

$$\approx K_\lambda(z)$$

(2.19)

This expression serves to mean that the Bessel function $^sK_\lambda(z)$ given by the series of $s$ terms in the braces is used to approximate ($\approx$) the Bessel function $K_\lambda(z)$. The RHS of (2.19) is coded as the R function `besselKseriesAppro(z, \lambda)`, (Appendix A.3). The evaluation of $K_\lambda(z)$ is readily obtainable by using the function `besselK(z, \lambda)` in base\(^1\). The absolute approximation error is given by

$$| ^sK_\lambda(z) - K_\lambda(z) | = \tau.$$  

(2.20)

As discussed in Section 2.2.2, when $|\lambda|$ equals half of an odd integer the value of $^sK_\lambda(z)$ and $K_\lambda(z)$ returned by `besselKseriesAppro(z, \lambda)` and `besselK(z, \lambda)` respectively are equal so the word “approximation” is dropped.

\(^1\)The base package in R
2.3.1 Identifying sufficiently large value of $z$

Here, the method is to increase the number of terms in the summation $G_s$ for a given value of $\lambda$ and $z$, which is the same with the one used by Olver (1997) in Section 2.1.2. However, the summation is not carried out manually.

Firstly, the behavior of (2.19) when $z = 5, 10, 20$ was studied by adding 84 terms successively to the approximation series. The solid and dashed line in Figures 2.1, 2.2 and 2.3 represent the values of $sK_\lambda(z)$ and $K_\lambda(z)$ for $\lambda = 0, 1$ and 2 respectively. For $z = 5$ and 10, by comparing the scales of Plots (a), (b) with those of (d) and (e) it is obvious that the approximation accuracy deteriorates as $s \to 84$. The deviation between the dashed and solid line indicates the accumulation of approximation errors as $s$ increases. Eventually, approximation errors explodes when $s = 84$. However, when $z = 20$, there is no change in the scale nor deviation between the dashed and solid line. This indicates that for all 3 values of $\lambda$, the approximation error remained unchanged as $s \to 84$. 
Figure 2.1: Approximation accuracy vs number of terms for $\lambda = 0$. The dashed lines represent the “exact” value $K_0(z)$ given by function \texttt{besseliK}(z, \lambda). The solid lines indicate the value of the asymptotic approximations of function \texttt{besseliKseriesAp}(z, \lambda) as $s \to 84$. Plots (a), (b), (c) show the approximation error when $1 \leq s \leq 5$. Plots (d), (e), (f) show the approximation error when $1 \leq s \leq 84$. 
Figure 2.2: Approximation accuracy vs number of terms for $\lambda = 1$. The dashed lines represent the “exact” value $K_1(z)$ given by function `besselK(z, \lambda)`. The solid lines indicate the value of the asymptotic approximations of function `besselKseriesAppro(z, \lambda)` as $s \to 84$. Plots (a), (b), (c) show the approximation error when $1 \leq s \leq 5$. Plots (d), (e), (f) show the approximation error when $1 \leq s \leq 84$. 
Figure 2.3: Approximation accuracy vs number of terms for $\lambda = 2$. The dashed lines represent the “exact” value $K_2(z)$ given by function $\text{besselK}(z, \lambda)$. The solid lines indicate the value of the asymptotic approximation of function $\text{besselKseriesApprox}(z, \lambda)$ as $s \to 84$. Plots (a), (b), (c) show the approximation error when $1 \leq s \leq 5$. Plots (d), (e), (f) show the approximation error when $1 \leq s \leq 84$. 
Figures 2.1–2.3 illustrate the fact that asymptotic expansions of $K_\lambda(z)$ is a divergent series and the approximation accuracy depends on the magnitude of $z$. If $z$ is not sufficiently large then adding more terms results in the accumulation of approximation errors (cf. p.55 of [Gray and Mathews, 1931]). By applying this feature, the function `largeVal(z, \lambda)` (in Appendix A.2) was written to identify whether $z$ is sufficiently large. Here, the routine iteratively adds as many terms to (2.19) as computationally allowed. By this it is meant that for a given $z$, adding an extra $s^{th}$ term is only computationally meaningful if the denominator of that term satisfies

$$(s - 1)! (8z)^{s-1} < \Psi$$

where $\Psi = .Machine$double.xmax,

which is the largest normalized floating-point number. Table 2.1 shows three of the

<table>
<thead>
<tr>
<th>Notation</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi$</td>
<td><code>.Machine$double.xmax</code></td>
<td>$1.797693 \times 10^{308}$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td><code>.Machine$double.xmin</code></td>
<td>$2.225074 \times 10^{-308}$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td><code>.Machine$double.eps</code></td>
<td>$2.220446 \times 10^{-16}$</td>
</tr>
</tbody>
</table>

Table 2.1: The `.Machine` numbers used by function `besselKseriesAppro(z, \lambda)`.

`.Machine` numbers of the R software on the 32 bit machine used to run the function `besselKseriesAppro(z, \lambda)`. The value of $z = 705$ was chosen because this is the minimum value of $z$ where the so-called underflow problem occurs. In other words, because $K_\lambda(z) = 0, \forall z > 705$ and $\lambda \in \mathbb{R}$ so `besselKseriesAppro(z, \lambda)` automatically switches to exponential scaling when $z > 705$ to avoid returning an approximate value as 0 when in fact the value is less than $\epsilon$ i.e., underflow occurs. Note that the function does inform the user when this change in scale is required. On machines with different architecture, e.g. word length, those constants will change and hence the corresponding bands will also change.
The stopping criterion of $\text{largeVal}(z, \lambda)$ may be something like

$$(s - 1)! (8z)^ {s-1} \geq \Psi^{0.98}. \quad (2.21)$$

Let $\tau_s$ be the value of $\tau$ when (2.21) is met, if $z$ is not sufficiently large then $\tau_s$ assumes a very large value. Hence, a value of $z$ is considered to be insufficiently large if

$$\frac{1}{\tau_s} \leq \gamma^{0.95} \quad \text{where} \quad \gamma = \text{Machine\$double\_eps.}, \quad \text{say}, \quad (2.22)$$

and $\tau$ is given by (2.20). The routine $\text{largeVal}(z, \lambda)$ outputs 0 if (2.22) is satisfied else 1.

### 2.3.2 Approximation procedure for $K_\lambda(z)$ for $\lambda \in [-90, 90]$

This section describes the asymptotic approximation steps. Firstly, there are two computational issues need solving.

1. Section 9.6 of Abramowitz and Stegun (1970) states “$K_\lambda(z)$ tends to zero as $|z| \to \infty$.” Consequently, $K_\lambda(z)$ can take values which are less than the smallest non-zero normalized floating-point number of a software. This issue is commonly referred to as underflow problem. The function $\text{besselK}(z, \lambda)$ deals with this problem by allowing the user to specify an additional argument, $\text{expon.scaled} = \text{TRUE}$ so that the value equals $e^z K_\lambda(z)$ is computed. For the purpose of the function $\text{besselKseriesAppro}(z, \lambda)$, underflow occurs when $z \in [705, \Psi]$. The approach used by $\text{besselK}(z, \lambda)$ to deal with underflow problem is applied by $\text{besselKseriesAppro}(z, \lambda)$. In other words, when $z \in [705, \Psi]$ the term $e^{-z}$ in (2.19) is cancelled computationally.

2. The definition of the absolute approximation error in (2.20) is mathematically correct. However, care must be taken when it is used as a stopping criterion for
the iterations by the function \texttt{besselKseriesAppro}(z, \lambda). This is because \(K_\lambda(z)\) can assume values which are less than \(\gamma\).

The approximation process includes the following 2 steps.

- **Step 1**: If the provided value of \(z < 705\) then \texttt{besselKseriesAppro}(z, \lambda) calls the function \texttt{largeVal}(z, \lambda) to identify if \(z\) is sufficiently large for the provided value of \(\lambda\).
  - If \texttt{largeVal}(z, \lambda) returns 0 then \texttt{besselKseriesAppro}(z, \lambda) calculates the sufficiently large value of \(z\), denoted as \(z'\), which ensures there is no accumulation of approximation error \(\forall z \geq z'\). The value of \(z'\) is then output so that a value of \(z \geq z'\) is supplied by the user for Step 2.
  - If \texttt{largeVal}(z, \lambda) returns 1 then \(z'\) assumes the value of \(z\) and goes to Step 2.

- **Step 2**: Apply the approximation error which is defined as

\[
\tau^+ = |'K_\lambda(z') \sigma - K_\lambda(z') \sigma| \quad \forall z' > 0 \tag{2.23}
\]

where

\[
\sigma = 1 \quad \text{if} \quad K_\lambda(z') > \gamma, \\
\sigma = \frac{1}{K_\lambda(z')} \quad \text{if} \quad K_\lambda(z') \leq \gamma \tag{2.24}
\]

to calculate the number of terms required and the corresponding approximation error.
For $0 < z' < 705$, the routines stops when one of the followings is met:

\[
\tau^* \leq \gamma^{0.95} \quad \text{if} \quad 0 \leq |\lambda| < 34, \\
\tau^* \leq \gamma^{0.7} \quad \text{if} \quad 34 \leq |\lambda| < 39, \\
\tau^* \leq \gamma^{0.5} \quad \text{if} \quad 39 \leq |\lambda| < 45, \\
\tau^* \leq \gamma^{0.3} \quad \text{if} \quad 45 \leq |\lambda| < 50, \\
\tau^* \leq \gamma^{0.2} \quad \text{if} \quad 50 \leq |\lambda| \leq 90.
\]

For $z' \geq 705$, the routine stops when

\[
\tau^* \leq \gamma^{0.95}.
\]

As can be seen from (2.20) and (2.23), $\tau$ and $\tau^*$ are equal when $\sigma = 1$. However, if $\sigma \neq 1$ then the use of $\tau$ as a stopping criterion is computationally meaningless because $K_\lambda(z') \leq \gamma$. Substituting (2.24) into (2.23) to obtain

\[
\tau^* = \left| \frac{sK_\lambda(z')}{K_\lambda(z')} - 1 \right|.
\]

In these cases, the use of $\tau^*$ is both computationally and mathematically sound. The ratio between the approximated and exact Bessel functions has a critical role for the stability and accuracy of this algorithm. This ratio approaches 1 as $s$ increases, which means $\tau^*$ approaches one of the above stopping criterions for all $z' > 0$ as required.

**Limitations of the algorithm**

For $\lambda < 705$, the limitation of this algorithm is that it becomes less accurate when $\lambda$ increases. This is because $sK_\lambda(z)$ and $K_\lambda(z)$ on the RHS of (2.23) are increasing functions of $\lambda$. Thus, when $z < 705$, the use of different stopping criterions for 5 subranges of $\lambda \in [-90, 90]$ is necessary to improve computational stability of the
function `besselKseriesAppro(z, λ)`.

Here are some example outputs

```r
> besselKseriesAppro(1, 1.2, details = TRUE, digits = 5)
For absolute value of lambda = 1.2, z should be >= 15
> besselKseriesAppro(15, 1.2, details = TRUE, digits = 5)
Approximated value : 1.0287e-07
Approximation error : 8.1438e-16
Number of terms required : 6
```

This output means that for \( \lambda = 1.2 \) the value of \( z' \) is 15, the approximated value of \( K_1(15) \) is \( 1.0287 \times 10^{-07} \), the absolute approximation error is \( 8.1438 \times 10^{-16} \) and the number of terms required is \( s = 6 \).

```r
> besselKseriesAppro(1, 89.1, details = TRUE, digits = 5)
For absolute value of lambda = 89.1, z should be >= 101
> besselKseriesAppro(103, 89.1, details = TRUE, digits = 5)
Approximated value : 1.4777e-30
Approximation error : 0.00028757
Number of terms required : 50
```

This means that for \( \lambda = 89.1 \) the value of \( z' \) is 101, the approximated value of \( K_{89.1}(103) \) is \( 1.4777 \times 10^{-30} \), the absolute approximation error is \( 0.00028757 \) and the number of terms required is \( s = 50 \).

```r
> besselKseriesAppro(1, 50, details = TRUE, digits = 5)
For absolute value of lambda = 50, z should be >= 21
> besselKseriesAppro(21, 50, details = TRUE, digits = 5)
Approximated value : 2.9377e+10
Approximation error : 0.00011826
Number of terms required : 54
```

This means that for \( \lambda = 50 \) the value of \( z' \) is 21, the approximated value of \( K_{50}(21) \) is \( 2.9377 \times 10^{+10} \), the absolute approximation error is \( 0.00011826 \) and the number of terms required is \( s = 54 \).
> besselKseriesAppro(0.001, 1.5, details = TRUE, digits = 5)
lambda : 1.5
Exact evaluation : 39633
Number of terms : 2

This indicates that $K_{1.5}(0.001) = 39633$ is not an approximate value because $\lambda$ is half of the odd integer 3. The number of terms $s$ is 2.

> besselKseriesAppro(1000, 0.9, details = TRUE, digits = 5)
Approximated value is on expon.scale
Approximated value : 0.039644
Approximation error : 1.3878e-17
Number of terms required : 4

This indicates that the approximation of $K_{0.9}(1000)$ is on exponential scale and the approximated value is 0.039644. The number of terms required is 4.

2.4 Approximations when $z$ values are small

This section addresses the concern of a referee about what approach should be used to approximate $K_{\lambda}(z)$ when $z$ is not sufficiently large. Entry 9.6.9 of [Abramowitz and Stegun (1970)] gives

$$K_{\lambda}(z) \approx \left( \frac{z}{2} \right)^{\lambda} \frac{1}{\Gamma(\lambda + 1)}, \quad z \to 0. \quad (2.25)$$

However, expression (2.25) does not specify what value of $z$ should be considered to be small nor approximation accuracy. [Harris (2009)] appears to be the only reference which answers those questions for $\lambda = 0$ and 1. This section implements the latter to obtain the function smallValAppro$(z, 1)$ for approximating $K_1(z)$ when $z$ is small to an absolute error of $\pm 10^{-15}$ or better. The formulae are given for four ranges of $z$, and in each formula range there are subranges of $z$ for different values of coefficients. For testing purposes, the coefficient values are only presented in cases where $z \geq 20$. 

54
This function is documented in Appendix A.4.

For \( z < 1.35 \)

\[
K_1(z) \approx \frac{1}{z} + z \left[ \log \left( \frac{z}{2} \right) \right] \left( \frac{1}{2} + \sum_{j=1}^{4} a_j z^{2j} \right) - z \sum_{j=0}^{4} b_j z^{2j},
\]

there are three sets of values of coefficients \( a_j \) and \( b_j \) corresponding to three ranges of \( z < 0.6, 0.6 \leq z < 1 \) and \( 1 \leq z < 1.35 \). Note that the summation \( \sum_{j=0}^{4} b_j z^{2j} \) was mistyped as \( \sum_{z=0}^{4} b_j z^{2j} \) on p.1735 of Harris (2009).

For \( 1.35 \leq z < 2.4 \)

\[
K_1(z) \approx z^{1/2} e^{-z} \sum_{j=1}^{9} c_j z^{-j},
\]

there are two sets of coefficients \( c_j \) for \( 1.35 \leq z < 1.75 \) and \( 1.75 \leq z < 2.4 \).

For \( 2.4 \leq z < 20 \)

\[
K_1 \approx z^{1/2} e^{-z} \sum_{j=1}^{7} c_j z^{-j},
\]

and there are 5 subranges for the values of coefficient \( c_j \): \( 2.4 \leq z < 2.8 \), \( 2.8 \leq z < 3.4 \), \( 3.4 \leq z < 4.25 \) and \( 4.25 \leq z < 6 \).

For \( z \geq 20 \)

\[
K_1(z) \approx z^{1/2} e^{-z} \sum_{j=1}^{3} c_j z^{-j}, \tag{2.26}
\]

with

\[
c_1 = 1.253325590451110162, \quad c_2 = 0.469253888870098903,
\]

\[
c_3 = -0.130663020947967846.
\]

### 2.5 Numerical studies

This section examines the accuracy of the function \( \text{besselKseriesAppro}(z, \lambda), {}^v K_\lambda(z) \), by comparing the function approximated values against the values returned by the functions \( \text{smallValAppro}(z, 1), {}^o K_1(z) \) and \( \text{besselK.nuAsym}(z, \lambda), {}^d K_\lambda(z) \). For this
purpose, the value of $K_{\lambda}(z)$ was calculated using command $\text{BesselK}(\lambda, z)$ in Maple and it was considered as “exact” value. The number of terms $s$ in $sK_{\lambda}(z)$ returned by $\text{besselKseriesAppro}(z, \lambda)$ indicates how many terms were required to obtain a specific accuracy level by the asymptotic approximations method. The number of terms $d$ in $dK_{\lambda}(z)$ was set to $d = 4$, which is the maximum numbers of terms allowed by this function. Note that the order of $z$ and $\lambda$ in the syntax of $\text{besselKseriesAppro}(z, \lambda)$ is the same with these of the R function $\text{besselK}(z, \lambda)$.

2.5.1 Asymptotic approximation versus “exact” evaluation

The functions $\text{largeVal}(z, \lambda)$ and $\text{besselKseriesAppro}(z, \lambda)$ are documented in Appendices A.2 and A.3 respectively. This section compares the accuracy of the latter against the “exact” values. The approximation results in Tables 2.2 and 2.3 are for $z < 705$ and $z \geq 705$ (i.e., approximating results when underflow occurred) respectively. They can be reproduced using

$$\text{besselKseriesAppro}(z, \lambda, \text{details = TRUE}, \text{digits = 14})$$

where the values of $z$ and $\lambda$ are specified in the first and second column of these tables.

The tables show that the approximated values of $sK_{\lambda}(z)$ agree with the “exact” values 13 decimal places or better (underlined numbers). The approximated and “exact” values even agree in cases where the values of the Bessel function of the second kind is much smaller than R machine accuracy number.
Table 2.2: Underlined numbers indicate the decimal places where “exact” and approximated value agree, and $s$ is the number of terms required by $\hat{K}_\lambda(z)$.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$\lambda$</th>
<th>$K_\lambda(z)$</th>
<th>$\hat{K}_\lambda(z)$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25.20</td>
<td>0.13285041773309</td>
<td>0.13285041773309</td>
<td>34</td>
</tr>
<tr>
<td>18</td>
<td>29.02</td>
<td>2.10645205202137</td>
<td>2.10645205202137</td>
<td>40</td>
</tr>
<tr>
<td>73</td>
<td>9.12</td>
<td>$5.10031745196595 \times 10^{-33}$</td>
<td>$5.10031745196595 \times 10^{-33}$</td>
<td>11</td>
</tr>
<tr>
<td>90</td>
<td>12.30</td>
<td>$2.49062966088729 \times 10^{-40}$</td>
<td>$2.49062966088729 \times 10^{-40}$</td>
<td>12</td>
</tr>
<tr>
<td>35</td>
<td>5.12</td>
<td>$1.92446021294562 \times 10^{-16}$</td>
<td>$1.92446021294563 \times 10^{-16}$</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 2.3: Underlined numbers indicate the decimal places where “exact” and approximated evaluation agree. The evaluation was on an $e^z$ scale to avoid an underflow problem, and $s$ is the number of terms required by $\hat{K}_\lambda(z)$.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$\lambda$</th>
<th>$K_\lambda(z)e^z$</th>
<th>$\hat{K}_\lambda(z)e^z$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>750</td>
<td>$10^{-05}$</td>
<td>0.04575693992889</td>
<td>0.04575693992889</td>
<td>4</td>
</tr>
<tr>
<td>950</td>
<td>$10^{-10}$</td>
<td>0.04065753288127</td>
<td>0.04065753288127</td>
<td>4</td>
</tr>
<tr>
<td>10^9</td>
<td>$10^{-10}$</td>
<td>3.96332729711059</td>
<td>3.96332729711059</td>
<td>1</td>
</tr>
<tr>
<td>1050</td>
<td>81</td>
<td>0.87691685030327</td>
<td>0.87691685030328</td>
<td>25</td>
</tr>
<tr>
<td>5050</td>
<td>55</td>
<td>0.02379373866414</td>
<td>0.02379373866414</td>
<td>10</td>
</tr>
</tbody>
</table>

2.5.2 Asymptotic approximation versus approximation using Debye polynomials

This section compares approximation accuracy of the asymptotic expansion series (2.17) against those of the Debye polynomials which are specified by entries 9.3.9–10 of Abramowitz and Stegun (1970). The former and the latter were implemented in the function `besselKseriesAppro` $(z, \lambda)$ and `besselK.nuAsym` $(z, \lambda)$ respectively. It is important to note that `besselK.nuAsym` $(z, \lambda)$ only works for $\lambda > 0$. It was stated in the documentation of the routine `besselK.nuAsym` $(z, \lambda)$ that the function is “for large $\lambda$ and possibly large $z$" without specifying what value of $z$ should be considered as large. Thus, the sufficiently large value of $z$ returned by `besselKseriesAppro` $(z, \lambda)$ was used for the testing purpose. As can be seen from Table 2.4, for these tested value of $\lambda$ and $z$, the function `besselKseriesAppro` $(z, \lambda)$ is
more accurate than the function \texttt{besselK.nuAsym}(z, \lambda). The values on Table 2.4 can be reproduced using

\begin{align*}
  ^aK_\lambda(z) & \quad \texttt{besselKseriesAppro}(z, \text{lambda}, \text{details} = \text{TRUE}, \text{digits} = 15) \\
  ^dK_\lambda(z) & \quad \texttt{require(Bessel)} \\
                  & \quad \texttt{besselK.nuAsym}(x = z, \text{nu} = \text{lambda}, \text{k.max} = 4, \\
                  & \quad \quad \texttt{expon.scaled} = \text{FALSE}) \\
  K_\lambda(z) & \quad \texttt{besselK}(x = z, \text{nu} = \text{lambda})
\end{align*}

the value of \( z \) and \( \lambda \) are specified in the first and second column.
Table 2.4: Coloured and underlined numbers indicate the decimal places where $\delta K_\lambda(z)$ and $\kappa K_\lambda(z)$ agree with “exact” value $K_\lambda(z)$ respectively. NaNs indicate the routine failed, and $s$ is the number of terms required by $\kappa K_\lambda(z)$.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$\lambda$</th>
<th>$\delta K_\lambda(z)$</th>
<th>$K_\lambda(z)$</th>
<th>$\kappa K_\lambda(z)$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>-1</td>
<td>NaN</td>
<td>1.01417298634117 $\times 10^{-007}$</td>
<td>1.014172927541249 $\times 10^{-007}$</td>
<td>6</td>
</tr>
<tr>
<td>50</td>
<td>-10</td>
<td>NaN</td>
<td>9.15098820998803 $\times 10^{-023}$</td>
<td>9.15098820998799 $\times 10^{-023}$</td>
<td>13</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>44410222 9074 $\times 10^{-023}$</td>
<td>3.44410222671756 $\times 10^{-023}$</td>
<td>3.44410222671756 $\times 10^{-023}$</td>
<td>9</td>
</tr>
<tr>
<td>90</td>
<td>2</td>
<td>1051801100 89 $\times 10^{-040}$</td>
<td>1.10518011004842 $\times 10^{-040}$</td>
<td>1.10518011004842 $\times 10^{-040}$</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>273256113 399 $\times 10^{-045}$</td>
<td>5.27325611329295 $\times 10^{-045}$</td>
<td>5.27325611329295 $\times 10^{-045}$</td>
<td>8</td>
</tr>
</tbody>
</table>
2.5.3 Asymptotic versus other approximation method

For $\lambda = 1$, this section compares approximation accuracy of $\text{smallValAppro}(z, 1)$ in cases where (2.26) applies, denoted as $^c K_1(z)$, with those of $\text{besselKseriesAppro}(z, \lambda)$ for $z \geq 20$. The evaluation of $z = 991, 1995, 5900, \text{ and } 10^9$ is on $e^z$ scale (i.e., $^c K_1(z) e^z$, $K_1(z) e^z$ and $^s K_1(z) e^z$. In Table 2.5, the coloured and underlined numbers indicate the decimal places where the approximate values by $\text{smallValAppro}(z, 1)$ and those given by the asymptotic approximation function $\text{besselKseriesAppro}(z, \lambda)$ agree with the “exact” values respectively. As can be seen, for these tested $z$ values the asymptotic expansion method is significantly more accurate than the approach derived in Harris (2009). Columns $^c K_1(z)$ and $^s K_1(z)$ can be reproduced using

\begin{verbatim}
smallValAppro(z = z, lambda = 1, digits = 13)
\end{verbatim}

and

\begin{verbatim}
besselKseriesAppro(z, lambda = 1, details = TRUE, digits = 13)
\end{verbatim}

respectively.
Table 2.5: Coloured and underlined numbers indicate the decimal places where $^\ast K_1(z)$ and $^\ast K_1(z)$ agree with “exact value” $K_1(z)$ respectively, and $s$ is the number of terms required by $^\ast K_1(z)$.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$^\ast K_1(z)$</th>
<th>$K_1(z)$</th>
<th>$^\ast K_1(z)$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>991</td>
<td>0.03982823697671</td>
<td>0.03982789632124</td>
<td>0.03982789632124</td>
<td>4</td>
</tr>
<tr>
<td>5900</td>
<td>0.01631794477539</td>
<td>0.01631779729269</td>
<td>0.01631779729269</td>
<td>3</td>
</tr>
<tr>
<td>1995</td>
<td>0.02806557493703</td>
<td>0.02806532671795</td>
<td>0.02806532671795</td>
<td>3</td>
</tr>
<tr>
<td>351</td>
<td>2.44630071945896 × 10^{-154}</td>
<td>2.44628224230588 × 10^{-154}</td>
<td>2.44628224230588 × 10^{-154}</td>
<td>5</td>
</tr>
<tr>
<td>$10^9$</td>
<td>3.96336351708480 × 10^{-005}</td>
<td>3.96332729909226 × 10^{-005}</td>
<td>3.96332729909226 × 10^{-005}</td>
<td>1</td>
</tr>
</tbody>
</table>
2.6 Summary

In this chapter, the asymptotic approximation topic proposed in the 18th century was examined using modern computing power. Like asymptotic approximation of other functions, the key point for approximation accuracy of $K_\lambda(z)$ is what the magnitude of $z$ is. The highest level of accuracy and the number of terms added depends on the computational limits of the software used. The obtained algorithm answers the questions in the literature about the finite number of terms required and the magnitude of $z$ to obtain a specific approximation error for $\lambda \in [-90, 90]$. The algorithm is robust and accurate. However, for $z < 705$, its accuracy decreases when $\lambda$ increases. The algorithm can be straightforwardly implemented in other programming languages such as Maple. In this case, the mentioned accuracy issue can be improved because Maple has a significantly larger and smaller value of the largest and smallest normalized floating-point number respectively than what R has.

For the tested parameter values the asymptotic approximations method is more accurate than the Debye approximation approach. This result has an important application for the study of the GH and GIG family of distributions because it allows the replacement of the Bessel function of the second kind in the density function of these distributions by an expression of elementary functions. Specifically, since the density function of the GH involves a ratio of $K_\lambda(z)$, the application of this chapter’s results to approximate the GH density functions can greatly simplify the expression of their density functions. The routine also has important applications in other research areas which involves the modified Bessel function of the second kind because it gives accurate approximation with a definite number of terms. For $\lambda = 1$, the approach is more accurate than the method proposed in Harris (2009).
Chapter 3

Parameter estimation of the hyperbolic and subclasses of the generalized inverse Gaussian

3.1 Introduction

Maximum-likelihood estimation is one of the most widely-used statistical methods for parameter estimation using sample data. Section 1.4 explains why fitting subclasses of the GH and MGH has attracted major research effort in the literature. This chapter concerns parameter estimation of the HYP (i.e., subclass of the GH with $\lambda = 1$) and 3 subclasses of the GIG with $\lambda = -\frac{1}{2}, \frac{3}{2}$ and $\frac{5}{2}$. The HYP and its related distribution are considered to be better models for heavy-tailed data. Consequently, a major part of research effort has been spent on parameter estimation. Commonly cited references include Barndorff-Nielsen (1978), Barndorff-Nielsen (1979), Eberlein and Keller (1995), Prause (1999) and Bibby and Sørensen (2003). The GIG is closely related to the GH family of distributions because it is the mixing distribution in the normal mean-variance mixture to form the GH distribution. Subsequently, parameter estimation of the GIG has been addressed in
the context of maximum-likelihood estimate of the subclasses of the GH using the EM algorithm. Thus, the fitting results of the mentioned subclasses of the GIG are foundational for multivariate parameter calibration of the \( d \)-DHYP and MNIG in the next chapter. Detailed discussion on this topic can be found in Protassov (2004), McNeil et al. (2005) and Wenbo (2005).

The terms maximum-likelihood estimator and maximum-likelihood estimate are commonly used in the literature. This chapter follows the convention on p.312 of Casella and Berger (2002), who used \( MLE \) and \( mle \) to denote the former and the latter, respectively. The authors also stated that “there is one distinction that must be made clear, the difference between an estimate and an estimator. An estimator is a function of the sample, while an estimate is the realized value of an estimator (that is, a number) that is obtained when a sample is actually taken.”

This chapter has two objectives. The first one involves parameter estimation of the HYP by deriving symbolic MLE of the unknown shape parameters \( \rho \) and \( \zeta \). This is based on the assumptions that the location \( \mu \) and scale \( \delta \) parameter known and the data is from the HYP with parameter \( \zeta \geq 22 \). To relax these assumptions, the second approach utilizes the results of Scott et al. (2009) to obtain a numerical approach to find the mle of the unknown shape parameters of the so-called standardized hyperbolic (SHYP) distributions. This approach treats \( \delta \) and \( \mu \) as nuisance parameters and their estimates are obtained by a non-likelihood approach. From the numerical test results, it appears that the fitting of SHYP is significantly faster and it gives better estimates of the shape parameters than the method of fitting non-standardized hyperbolic distribution, where estimates of all 4 parameters are obtained using maximum-likelihood method.

The second purpose is to propose a symbolic approach for deriving explicit MLE of 3 subclasses of GIG distribution, where parameter \( \lambda \) assumes \( \lambda = -\frac{1}{2}, \frac{3}{2}, \frac{5}{2} \). These cases are of interest, because the results will be used for fitting important subclasses of the multivariate generalized hyperbolic distributions in the next chapter. Note
that the proposed method can be applied to derive explicit MLE of the GIG when \( \lambda \) assumes other half of an odd integer values.

An important feature of the symbolic maximum-likelihood method is that it does not require starting values to initialize an optimization process to find mle of the parameter of interest. As a result, the symbolic method offers stable and fast maximum-likelihood fitting routines even for small sample sizes. This is an important improvement because a numerical approach requires “good” starting values to converge successfully to an extremum of the log-likelihood. However, finding those “good” starting values can be difficult. A specific reference regarding the fitting of the HYP or subclasses of the GIG by a symbolic method cannot be located at the time this thesis is written.

The chapter is structured as follows. Section 3.2 gives theoretical foundation for the derivation of the maximum likelihood estimator of a log-likelihood function of 2 unknown parameters. Section 3.3 derives MLE of the shape parameters of the HYP. Section 3.4 involves fitting the SHYP numerically. Section 3.5 derives explicit MLE for 3 subclasses of the GIG. Section 3.6 summarizes the chapter.

### 3.2 Maximum-likelihood estimation

This section briefly describes the calculus steps and the Maple commands used to obtain MLE of the log-likelihood with 2 unknown parameters. It lays theoretical and computational base for obtaining MLE of the shape parameters of the HYP and three subclasses of the GIG.

#### 3.2.1 Calculus steps

Suppose \( \mathbf{x} = (x_1, \ldots, x_n)' \) are i.i.d. observations from a population of the distribution for which MLE of 2 unknown parameters \( \mathbf{\theta} = (\theta_1, \theta_2)' \) are to be found. The probability density function can be written as \( f(\mathbf{x} | \mathbf{\theta}, \mathbf{\bar{\theta}}) \), where \( \mathbf{\bar{\theta}} \) is the vector of
known parameters if the distribution has more than 2 parameters. Accordingly, the log-likelihood can be written as

\[ L(\theta, \bar{\theta} | x) = \sum_{i=1}^{n} f(x_i | \theta, \bar{\theta}) = L(\theta). \]

Assuming that \( L(\theta) \) is a smooth function with interior maximum. When using bivariate calculus to prove that \( L(\theta) \) attains a local maximum at \( \hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2)' \), it was stated on p.322 of \textit{Casella and Berger (2002)} that the following 3 conditions need to hold:

1. The first order partial derivatives are 0,

\[ \frac{\partial}{\partial \theta_1} L(\theta) \bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} = 0 \quad \text{and} \quad \frac{\partial}{\partial \theta_2} L(\theta) \bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} = 0. \] (3.1)

2. At least one second-order partial derivatives is negative,

\[ \frac{\partial^2}{\partial \theta_1^2} L(\theta) \bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} < 0 \quad \text{or} \quad \frac{\partial^2}{\partial \theta_2^2} L(\theta) \bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} < 0. \]

3. The determinant of the Hessian \( H \) is positive,

\[ |H| = \begin{vmatrix} \frac{\partial^2}{\partial \theta_1^2} L(\theta) & \frac{\partial^2}{\partial \theta_1 \partial \theta_2} L(\theta) \\ \frac{\partial^2}{\partial \theta_1 \partial \theta_2} L(\theta) & \frac{\partial^2}{\partial \theta_2^2} L(\theta) \end{vmatrix}_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} > 0. \] (3.2)

Conditions 2 and 3 ensure that \( H \) is negative-definite by having both eigenvalues negative, which means that \( L(\theta) \) attains a local maximum at \( \theta_1 = \hat{\theta}_1, \theta_2 = \hat{\theta}_2 \).

The differences between a parameter true value and its estimates are of interest for the purpose of examining the accuracy of the fitting routines in this chapter by
defining
\[ \hat{\epsilon} = \hat{\theta}_1 - \theta_1, \]
\[ \hat{\kappa} = \hat{\theta}_2 - \theta_2. \]

3.2.2 Computational tools for symbolic maximum-likelihood estimation

In this version of the thesis, the obtained MLE are not called “analytic” MLE as it was pointed out by one of the referees that “analytic” is not an appropriate term. It was commented that “I find the claim that he has found analytic mles a bit of a stretch.” As the approach here is to obtain the MLE by symbolically solving the score functions for the unknown parameters, the appropriate term to be used is symbolic MLE. The reliance on a symbolic computer package such as Maple to find the MLE also did not appear to have been favoured either as it was stated that “…relies heavily on Maple for the computation.” In fact, symbolic computation has been widely used in likelihood research by statisticians. Stafford et al. (1994) described a set of procedures that automate many algebraic calculations to study of likelihood and likelihood-type functions. The purposes of the authors were to provide a practical alternative to difficult manual algebraic computations, and to obtain quick and error-free results. In Stafford and Andrews (1993), the authors proposed a broad method for using Maple, Mathematica and other commercial software to work with likelihood problems. Rose and Smith (2000) used Mathematica to obtain MLE of some common distributions. This chapter utilizes Maple to obtain the MLE for the HYP and GIG with two unknown parameters. The obtained MLE were then coded as R functions to find mle from a given data set. Specifically, the command solve( ) in Maple was used to solve the system of equations in (3.1) symbolically and possible solutions of the system are stored in the form of algebraic expressions consisting of polynomials RootOf(exp_Z). Lastly, the R function polyroot( ) is used to solve for
the solutions of RootOf(exp(Z)). Possible solutions returned by polyroot( ) are tested
for conditions (3.1)–(3.2).

3.3 Symbolic maximum likelihood estimators of the shape parameters of the hyperbolic dis-
tribution

This section derives symbolic MLE of the shape parameter \( \rho \) and \( \zeta \) by replac-
ing \( K_1(\zeta) \) in (1.5) using the first two terms of its asymptotic approximation

\[
\tilde{K}_1(\zeta) = \sqrt{\frac{\pi}{2\zeta}} e^{-\zeta} \left(1 + \frac{3}{8\zeta}\right),
\]

(3.3)

the objective for such replacement is it allows the calculus described in Section 3.2.1
to be performed symbolically.

3.3.1 Approximate log-likelihood function of the hyperbolic
distribution

Definition 2. If \( x = (x_1, \ldots, x_n)' \) are i.i.d. observations from a hyperbolic distri-
bution with parameters \( \delta, \mu \) known and \( \zeta \geq 22 \) then the approximate log-likelihood
function is defined as

\[
\tilde{L}(\rho, \zeta) = -2n \delta \log (1 + \rho^2) - n \log \left[\tilde{K}_1(\zeta) - \zeta \sqrt{(1 + \rho^2)} \Delta + \zeta \rho \Lambda \right],
\]

(3.4)

where

\[
\Delta = \sum_{i=1}^{n} \left[1 + (x_i - \mu)^2/\delta^2\right]^{1/2}, \quad \Lambda = \sum_{i=1}^{n} (x_i - \mu)/\delta.
\]

(3.5)

These assumptions will be relaxed later by fitting the SHYP to standardized

68
data.

**Lower bound of the approximation error of the log-likelihood**

In order to use (3.4), it is very important to specify the lower bound of the approximation error by using the function `besselKseriesAppro(z, \lambda)` obtained in Chapter 2. For \( \zeta = 22 \), the approximation error is given by

\[
> \text{besselKseriesAppro}(z = 22, \text{lambda} = 1, \text{details} = \text{TRUE}, \text{digits} = 3)
\]

**Approximated value : 7.58e-11**  
**Approximation error : 6.76e-16**  
**Number of terms required : 2**

This output indicates that the approximated value of \( K_1(22) \) is \( 7.58 \times 10^{-11} \) and the approximation error equals \( 6.76 \times 10^{-16} \). As it was demonstrated in Chapter 2 that the approximation becomes even more accurate when \( \zeta > 22 \). This output specifies that the numerical lower bound of the approximate log-likelihood (3.4) using (3.3) is \( 6.76 \times 10^{-16} \) which is as small as the `Machine$double.eps` of the machine used to run R for the purposes for this thesis.

**Obtaining maximum likelihood estimators**

Firstly, the score functions were found by taking partial derivatives with respect to unknown parameters \( \rho \) and \( \zeta \) using the Maple command `diff()` . Next, `solve()` was used to solve the system of score functions symbolically. However, Maple did not return possible solutions of the system of score functions directly. Instead, these solutions were stored as solutions of equations expressed as `RootOf(exp_Z)`. It is stated in the documentation of the software that `RootOf(exp_Z)` is a placeholder for storing all roots of an equation in one variable \( Z \). If \( M \) denotes the MLE of a fitted parameter then MLE of the shape parameter \( \zeta \) is given by

\[
M_\zeta = \text{RootOf} \left( \sum_{j=0}^{8} Z^j c_j \right), \quad (3.6)
\]

69
with

\[ c_0 = 18225 n^4, \]
\[ c_1 = 129600 n^4, \]
\[ c_2 = 180 n^2 (2486 n^2 - 45 \Delta^2 + 54\Lambda^2) , \]
\[ c_3 = 720 n^2 (1336 n^2 + 15 (-7 \Delta^2 + 8\Lambda^2)) , \]
\[ c_4 = 48 (28987 n^4 + 27\Lambda^2 (-\Delta^2 + \Lambda^2) + n^2 (-5835 \Delta^2 + 6486\Lambda^2)) , \quad (3.7) \]
\[ c_5 = 1536 (897 n^4 - 9 \Delta^2\Lambda^2 + 9\Lambda^4 + n^2 (-345 \Delta^2 + 394\Lambda^2)) , \]
\[ c_6 = 2048 (443 n^4 + 27\Lambda^2 (-\Delta^2 + \Lambda^2) + 3 n^2 (-89 \Delta^2 + 114\Lambda^2)) , \]
\[ c_7 = 32768 (11 n^4 - 3 \Delta^2\Lambda^2 + 3\Lambda^4 + n^2 (-9 \Delta^2 + 14\Lambda^2)) , \]
\[ c_8 = 65536 (n^2 + \Lambda^2) (n^2 - \Delta^2 + \Lambda^2) . \]

For a given sample of \( n \) i.i.d. observations, all quantities in (3.5) and (3.7) are easily computed and the \texttt{R} function \texttt{polyroot()} is used to solve equation (3.6). These roots are considered possible maximum-likelihood estimates of parameter \( \zeta \) denoted as \( \hat{\zeta} \).

Because (3.6) is a polynomial of order 8, \texttt{polyroot()} returns a vector of 8 roots in the form of 8 complex numbers. Notationally, \( \hat{\zeta}_j = a_j + ib_j \) and \( \hat{\zeta} = (\hat{\zeta}_1, \ldots, \hat{\zeta}_8)' \).

For MLE of parameter \( \rho \), Maple gives

\[
M_\rho = -\left\{ n\Lambda \left( (9 + 2\hat{\zeta}(7 + 8\hat{\zeta}))^2 (15 + 2\hat{\zeta}(15 + 8\hat{\zeta})) n^2 + 4\hat{\zeta}^2(3 + 8\hat{\zeta})^2 \times \left( -4(6 + \hat{\zeta}(11 + 8\hat{\zeta})) \Delta^2 + (9 + 2\hat{\zeta}(7 + 8\hat{\zeta}))\Lambda^2 \right) \right) \right\} \times \left\{ 2\hat{\zeta}(3 + 8\hat{\zeta}) \left( 4\hat{\zeta}^2(3 + 8\hat{\zeta})^2\Lambda^2 (-\Delta^2 + \Lambda^2) + (9 + 2\hat{\zeta}(7 + 8\hat{\zeta})) \times (15 + 2\hat{\zeta}(15 + 8\hat{\zeta})) n^2 (\Delta^2 + \Lambda^2) \right) \right\}^{-1} . \quad (3.8)
\]

The evaluation of estimates of (3.8) gives \( \hat{\rho} = (\hat{\rho}_1, \ldots, \hat{\rho}_8)' \) with \( \hat{\rho}_j = d_j + ie_j \).
3.3.2 Evaluating symbolic maximum-likelihood estimators of the approximate log-likelihood

This section describes the process to evaluate $M_\zeta$ and $M_\rho$ to obtain the mle of the shape parameters by the function \texttt{symFitShapeHyp()} in Appendix B.3.

Firstly, for each $\hat{\rho}$ and $\hat{\zeta}$ which is returned by \texttt{polyroot()}, the real and imaginary parts are separated using functions \texttt{Re()} and \texttt{Im()}, respectively. Since $\hat{\rho}$ and $\hat{\zeta}$ are subject to sampling variation, the critical idea for obtaining mle of these parameters is to choose the real part $a_j$ of $\hat{\zeta}_j$ and $d_j$ of $\hat{\rho}_j$ which satisfy the three conditions in Section 3.2.1. For each $a_j$ and $d_j$, the first condition is considered to be met if

$$
\left| \frac{\partial}{\partial \rho} \tilde{L}(\theta) \right|_{\theta_1=d_j, \theta_2=a_j} < \epsilon^* \quad \text{and} \quad \left| \frac{\partial}{\partial \zeta} \tilde{L}(\theta) \right|_{\theta_1=d_j, \theta_2=a_j} < \epsilon^*, \quad j = 1, \ldots, 8.
$$

(3.9)

The second-derivative test is considered to be met if both eigenvalues of the Hessian matrix in (3.2) are negative. The condition that $a_j$ and $d_j$ are real numbers are satisfied if

$$
b_j < \epsilon^* \quad \text{and} \quad e_j < \epsilon^*.
$$

Here, $\epsilon^* = 10^{-12}$ is an arbitrary number. The checking for the above conditions is repeatedly carried out for all $a_j$ and $d_j$ ($j = 1, \ldots, 8$). Finally, the uniqueness condition is checked by calculating the length of the vector of $a_j$ and $d_j$, which satisfies previously tested conditions. If this length is 1 then $\hat{\rho}$ and $\hat{\zeta}$ are the unique maximum-likelihood estimates of $\rho$ and $\zeta$.

Computational stability

Firstly, it is extremely important to note that \texttt{symFitShapeHyp()} is meant to be added to the package \texttt{HyperbolicDist}, see Appendix B.2.1, so it works under Parameterization 1 of this package like all other functions of \texttt{HyperbolicDist}. Computational stability of the function \texttt{symFitShapeHyp()} was studied by generat-
ing 5000 pseudo hyperbolic data sets of different sizes from 20 to 50. The testing shape parameters cover a range of the possible shapes of the hyperbolic distributions. The test shows that function \texttt{symFitShapeHyp()} performs very stably for all sample sizes and testing parameter values. It is observed that

- the first-derivative test is met because the LHS of the inequalities in (3.9) is less than $\varepsilon^* = 10^{-12}$,

- the second-derivative test is always satisfied,

- the imaginary parts of the estimates, which already met the first and second-derivative tests are less than $\varepsilon^* = 10^{-12}$,

- the uniqueness condition is always satisfied.

A question of interest is whether the maximum of the log-likelihood obtained at the value of $\hat{\rho}$ and $\hat{\zeta}$ is the global maximum? To obtain a firmer answer to this question, it must firstly be proven that $\hat{\rho} = (\hat{\rho}_1, \ldots, \hat{\rho}_8)'$ and $\hat{\zeta} = (\hat{\zeta}_1, \ldots, \hat{\zeta}_8)'$ are all roots of the log-likelihood function (3.4), and secondly, the log-likelihood does not have maximum at infinity. Obviously, the first task is not workable because even solving the system of score functions is not possible if done manually. In addition, proving the second condition to be true is a formidable task as it was stated on p.323 of Casella and Berger (2002). However, given that Maple is one of the most powerful and popular symbolic software package available, it is reasonable to believe that $\hat{\rho}$ and $\hat{\zeta}$ contains all roots of the log-likelihood function (3.4). Secondly, extensive numeric test for the function \texttt{symFitShapeHyp()} showed that the maximum log-likelihood function has only one real solution for each of the unknown parameters. It appears that the obtained maximum log-likelihood value is a global one. Similar conclusion was also reached on p.8 of Prause (1999). The author fitted the HYP to financial data using a numerical method, he stated that “Although we have no theoretically assured convergence of our algorithm, tests with different starting
values reveal that for financial data the use of reasonable starting values results in the convergence to a global extremum.”

3.3.3 Numerical examination

Appendix B.2.1 gives technical details for the numerical examination of this function. Figure 3.1 presents the plots for graphical assessment of the goodness-of-fit of the fitting of \( \rho \) and \( \zeta \) by generating 50 observations using parameter vector \( \theta = (5, 23, 1, 5)' \). These plots were obtainable using \texttt{symFitShapeHyp(..., plots = TRUE)}. As can be seen, the fitted curves (i.e., density and log-density) fit data (i.e., histogram and log-histogram) very well which is confirmed by the Q-Q and P-P plots.
Figure 3.1: Goodness-of-fit plots for the fitting of the shape parameters $\rho$ and $\zeta$. PIT data: Probability-integral-transformed data.

Figure 3.2 depicts the distribution of the differences between true and estimated parameter values by generating 300 samples using $\theta = (1.5, 23, 5, 1)'$. As can be seen, $\hat{\varepsilon}$ varies less than $\hat{\kappa}$ and the variation of both quantities decreases when sample size $n$ increases, which is consistent with the maximum-likelihood theory. More importantly, the distribution of $\hat{\varepsilon}$ and $\hat{\kappa}$ are centered around 0 even for small sample sizes $n = 20$ and 30.
Figure 3.2: Histograms of differences between estimates and true parameter values. \( \hat{\varepsilon} = \hat{\rho} - \rho \) and \( \hat{\kappa} = \hat{\zeta} - \zeta \) where \( \rho = 1.5, \zeta = 23 \). First row: \( n = 20 \); second row: \( n = 30 \); third row: \( n = 100 \).

### 3.3.4 Symbolic versus numeric fitting of the shape parameters

Appendix B.3.1 describes the R function `numeFitShapeHyp()` for fitting the shape parameters of the HYP using `optim()` and `nlm()`. These are commonly used routines for finding mle of the HYP and its related distributions in the literature. The functions `symFitShapeHyp()` and `numeFitShapeHyp()` are entirely comparable in all respects but the method to find optimal value of the log-likelihood given by (3.4).
Here, the purpose is to demonstrate that computational stability of the former is significantly better because it does not require starting values to find the maximum-likelihood estimates.

A parameter vector was used to generate a “population”, $H_1$, of 5000 observations of HYP data. Each row in Table 3.1 gives the output from the fitting of a sample of size $n$ drawn from $H_1$. The symbol “$\circ$” indicates cases where the numeric fitting functions failed to start an optimization process. Starting values $\rho_{sv}$ and $\zeta_{sv}$ were used by `optim(..., method = “BFGS”)` and `nlm()` only. These tests show that computational stability of `numeFitShapeHyp()` was susceptible to changes in starting values which happened even for relatively large sample $n = 300$. The subscript SYM means the estimates are obtained by `symFitShapeHyp()`. The subscripts BFGS and NLM indicate the mle are obtained by `optim(..., method = “BFGS”)` and `nlm()` respectively. The subscript SV stands for starting values of $\rho$ and $\zeta$ which are only required by `numeFitShapeHyp()`. Significantly more extensive tests are described in Appendix B.3.3.
<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$n$</th>
<th>$\hat{\rho}_{\text{SYM}}$</th>
<th>$\hat{\zeta}_{\text{SYM}}$</th>
<th>$\hat{\rho}_{\text{BFGS}}$</th>
<th>$\hat{\zeta}_{\text{BFGS}}$</th>
<th>$\hat{\rho}_{\text{NLM}}$</th>
<th>$\hat{\zeta}_{\text{NLM}}$</th>
<th>$\rho_{\text{SV}}$</th>
<th>$\zeta_{\text{SV}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1.5, 23, 10, -5)'$</td>
<td>30</td>
<td>1.12</td>
<td>21.32</td>
<td>1.12</td>
<td>21.32</td>
<td>1.12</td>
<td>21.32</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.12</td>
<td>21.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.12</td>
<td>21.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.12</td>
<td>21.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>$(−11, 25, 1, 10)'$</td>
<td>300</td>
<td>−10.56</td>
<td>26.81</td>
<td>−10.56</td>
<td>26.81</td>
<td>−10.56</td>
<td>26.81</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−10.56</td>
<td>26.81</td>
<td>−10.56</td>
<td>26.81</td>
<td>−10.56</td>
<td>26.81</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−10.56</td>
<td>26.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>−10.56</td>
<td>26.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 3.1: Symbols $\circ$ indicate failure to start an optimization process. Parameter vector $\theta$ follows Parameterization 1 of package HyperbolicDist.
3.4 Numerical approach for fitting the standardized hyperbolic distribution

Section 3.3 details the symbolic approach for fitting the HYP, which is based on a set of 3 assumptions: $\delta$, $\mu$ known and $\zeta \geq 22$. In order to relax these assumptions, this section proposes the routine `standHyperbFit()`, which is described in Appendix B.4, for numerically fitting the so-called standardized hyperbolic distribution (SHYP). This is a subclass of the standardized generalized hyperbolic distribution (SGH) proposed in Scott et al. (2009).

Obtaining parameter estimates of distributions with multiple parameters has been known to be a challenging task. Pawitan (2001) reviewed different approaches to obtain estimates of the parameters of interest only and the remaining ones were treated as nuisance parameters. Estimated-likelihood, as it was called on p.292 Pawitan (2001), among others, is a well-researched approach for dealing with nuisance parameters. This method was also termed pseudo-likelihood in Gong and Samaniego (1981). Asymptotic features of the estimates of the pseudo-likelihood were shown to be consistent with those of standard maximum-likelihood approach in Parke (1986). Further discussion on this topic can be found in Pawitan (2001). The underlying idea of these work is that nuisance parameters are to be estimated by a method other than maximum-likelihood. They are then inserted in the log-likelihood function for maximum-likelihood estimation of the parameter of interest. The term pseudo-likelihood distinguishes the cases where the nuisance parameters are known (e.g., $\delta$ and $\mu$ known) with those where they are to be estimated from a data set. The latter accounts for the uncertainty of not knowing the value of nuisance parameters.

Computationally, the need for dividing the parameter space of the subclasses of the GH (e.g., HYP) into the parameters of interest and nuisance ones is important. This is because besides common challenges of multi dimensional likelihood estimation, the log-likelihood of the HYP has been known to be flat (cf. p.29 of
Barndorff-Nielsen and Blæsild [1981]. Practically, the rationale for this is that only the estimates of the shape parameters are useful for modelling real data, see Bibby and Sørensen (2003) and that these shape parameters are unchanged under the location and scale transformation, see Prause (1999). Thus, for the purpose of this section, parameters $\mu$ and $\delta$ are treated as nuisance parameters and their estimates are to serve the purpose of obtaining the mle of the shape parameters only.

### 3.4.1 Standardization of the generalized hyperbolic distributions

This section reproduces the proof and discussion in Scott et al. (2009) for standardizing the GH to create standardized generalized hyperbolic distributions. The SHYP is a special case of the SGH with $\lambda = 1$.

**Lemma 1.** Suppose $X \sim \text{GH}(\lambda, \alpha^*, \beta^*, \delta^*, \mu^*) = \text{GH}(\lambda, \zeta, \rho)$. If we define

$$
\alpha^* \equiv \alpha(\zeta, \rho) = \left\{ \frac{\zeta}{1 - \rho^2} \kappa_\lambda(\zeta) \left(1 + \frac{\rho^2 \zeta^2}{1 - \rho^2} \Delta \kappa_\lambda(\zeta) \right) \right\}^{1/2},
$$

$$
\beta^* \equiv \beta(\zeta, \rho) = \alpha(\zeta, \rho) \rho,
$$

$$
\delta^* \equiv \delta(\zeta, \rho) = \frac{1}{\alpha(\zeta, \rho)} \frac{\zeta}{\sqrt{1 - \rho^2}},
$$

$$
\mu^* \equiv \mu(\zeta, \rho) = -\frac{1}{\mu(\zeta, \rho)} \frac{\rho \zeta^2}{1 - \rho^2} \kappa_\lambda(\zeta),
$$

with

$$
\kappa_\lambda(\zeta) = \frac{K_{\lambda+1}(\zeta)}{\zeta K_\lambda(\zeta)} \quad \text{and} \quad \Delta \kappa_\lambda(\zeta) = \kappa_{\lambda+1}(\zeta) - \kappa_\lambda(\zeta),
$$

then we have

$$
\mathsf{E}X = 0 \quad \text{and} \quad \text{Var}X = 1.
$$

This can be proven in a straightforward way, by computing the mean and variance from the general formulae for the expectation values.
Proof.

\[ \text{EX} = \mu^* + \frac{\beta^* \delta^*}{\sqrt{\alpha^* - \beta^*}} \frac{K_{\lambda+1}(\zeta)}{K_{\lambda}(\zeta)} \]
\[ = -\frac{1}{\alpha^*} \frac{\rho \zeta^2}{1 - \rho^2} \kappa_{\lambda}(\zeta) + \frac{\alpha^* \rho \delta^* \zeta}{\sqrt{\alpha^* - \beta^*}} \kappa_{\lambda}(\zeta) = 0 \]

as required, and

\[ \text{Var} X = \delta^2 \left\{ \frac{K_{\lambda+1}(\zeta)}{\zeta K_{\lambda}(\zeta)} + \frac{\beta^*}{\alpha^* - \beta^*} \left[ \frac{K_{\lambda+2}(\zeta)}{K_{\lambda}(\zeta)} - \left( \frac{K_{\lambda+1}(\zeta)}{K_{\lambda}(\zeta)} \right)^2 \right] \right\} \]
\[ = \delta^2 \left\{ \kappa_{\lambda}(\zeta) + \alpha^* \rho^2 \delta^2 \frac{\kappa_{\lambda+1}(\zeta) \kappa_{\lambda}(\zeta) - \kappa_{\lambda}(\zeta)^2}{\kappa_{\lambda+1}(\zeta) \kappa_{\lambda+1}(\zeta)} \right\} = 1, \]

completing the proof. \( \square \)

Note that if \( \rho = 0 \) then GH distribution is symmetric. In this case the GH parameters simplify to

\[ \alpha^* \equiv \alpha(\zeta) = \zeta \frac{K_{\lambda+1}(\zeta)}{K_{\lambda}(\zeta)}; \quad \beta^* = 0, \]
\[ \delta^* \equiv \delta(\zeta) = \frac{K_{\lambda}(\zeta)}{K_{\lambda+1}(\zeta)}; \quad \mu^* = 0. \]

Another way to view Lemma 1 is that it allows the introduction of a fifth parameterization called the \((\alpha^*, \beta^*)\) parameterization. This parameterization is most closely related to the second or \((\rho, \zeta)\) parameterization

\[ \alpha^* = \left\{ \frac{\zeta}{1 - \rho^2} \kappa_{\lambda}(\zeta) \left( 1 + \frac{\rho^2 \zeta^2}{1 - \rho^2} \Delta \kappa_{\lambda}(\zeta) \right) \right\}^{1/2}, \]
\[ \beta^* = \alpha(\zeta, \rho) \rho, \]
\[ \delta = \delta, \]
\[ \mu = \mu. \] (3.11)

The \( \delta \) and \( \mu \) are the location and scale parameters with \( \mu \) able to take any real number, and \( \delta \) is a positive value. From Lemma 1 we see that if \( X \sim \text{GH}(\lambda, \alpha^*, \beta^*, \delta, \mu) \)
then setting

\[
\delta^* = \frac{1}{\alpha(\zeta, \rho)} \frac{\zeta}{\sqrt{1 - \rho^2}},
\]

\[
\mu^* = -\frac{1}{\mu(\zeta, \rho)} \frac{\rho \zeta^2}{1 - \rho^2} \kappa(\zeta)
\]

(3.12)

gives a GH distribution with the same shape but having mean zero and unit standard deviation.

**Definition 3.** The standardized generalized hyperbolic distribution is defined by

\[
\text{SGH}(x; \lambda, \rho, \zeta) = \text{GH}(\lambda, \alpha^*, \beta^*, \delta^*, \mu^*)
\]

where \(\alpha^*\) and \(\beta^*\) are given in (3.11) and \(\delta^*\) and \(\mu^*\) are given in (3.12). The SHYP is a special case of the SGH with \(\lambda = 1\).

Because the GH is used to model heavy-tailed data, a referee raised the question of why not use the absolute deviation from the median used for standardization instead of the mean, as the former is not affected by outliers? An answer is that although the GH distributions are described as heavy-tailed, all moments always exist as does the moment generating function. Thus, there is no problem concerning the existence of the mean and variance. In addition, standardizing formulae are available for the mean and variance, but not for the median and inter-quartile range.

### 3.4.2 Steps to fit standardized hyperbolic distribution

Given \(x = (x_1, \ldots, x_n)'\) are i.i.d. observations from a hyperbolic distribution with all parameters unknown, the following exact log-likelihood of the SHYP is to be optimized by a numerical method.

\[
L_{\text{hyp}}(\rho, \zeta) = -2n \delta \log (1 + \rho^2) - n \log [K_1(\zeta)] - \zeta \sqrt{(1 + \rho^2)} \Delta + \zeta \rho \Lambda. \quad (3.13)
\]
The formulae to calculate $\Delta$, $\Lambda$ are given by (3.5) but $\delta$ and $\mu$ are replaced by $\hat{\delta}^*$, $\hat{\mu}^*$ which are explained below. The following are steps to find the mle of standardized hyperbolic distributions described in Scott et al. (2009).

- **Step 1:** Standardize the data with

\[
\tilde{x} = \frac{x - \overline{x}}{s}
\]

where $\overline{x}$ and $s$ are the sample mean and standard deviation respectively.

- **Step 2:** Use function `hyperbFitStart(\ldots, method = “BN”)` and standardized data to find the estimates of parameters in $(\rho, \zeta, \delta, \mu)$ parameterization. The obtained estimates are considered as “starting values” for the next steps. The function `hyperbFitStart()` offers 4 different methods to find starting values. The method used in this chapter was described in Barndorff-Nielsen (1977).

- **Step 3:** Let $\tilde{\rho}$ and $\tilde{\zeta}$ be the estimates of $\rho$ and $\zeta$ in Step 2. These values of $\tilde{\rho}$ and $\tilde{\zeta}$ are then substituted into (3.10) to obtain the corresponding shape parameters under the $(\alpha^*, \beta^*, \delta^*, \mu^*)$ parameterization.

- **Step 4:** Let $(\tilde{\alpha}^*, \tilde{\beta}^*, \tilde{\delta}^*, \tilde{\mu}^*)$ be the estimates from Step 3. These estimates are then converted to the $(\rho, \zeta)$ parameterization of (3.13). Denoting these estimates as $(\tilde{\rho}^*, \tilde{\zeta}^*, \tilde{\delta}^*, \tilde{\mu}^*)$. The rationale here is that the SHYP under $(\alpha^*, \beta^*)$ has the same shape as the HYP $(\rho, \zeta)$ parameterization but has mean zero and unit standard deviation. Moreover, since parameters $\rho$ and $\zeta$ are unchanged under location and scale transformation, the estimates of $\tilde{\delta}^*$, $\tilde{\mu}^*$ and standardized data ($\tilde{x}$) are inserted into equation (3.5) to obtain $\Delta$ and $\Lambda$. These are the quantities required for evaluating the MLE of $\rho$ and $\zeta$ in the next step.

- **Step 5:** Use the numerical optimization function `optim()` to obtain the mle of $\rho$ and $\zeta$. These estimates are denoted $\hat{\rho}$ and $\hat{\zeta}$. 

82
• **Step 6:** Normalize the location and scale parameters using \( \hat{\mu} = \bar{X} + s\tilde{\mu}^* \) and \( \hat{\delta} = s\tilde{\delta}^* \). Note that \( \hat{\mu} \) and \( \hat{\delta} \) are not maximum likelihood estimates of \( \mu \) and \( \delta \) and are considered as nuisance parameters.

### 3.4.3 Numerical examination

### 3.4.4 Numerical examination

Appendices B.2.2 and B.4 gives critical technical information of \texttt{standHyperbFit()}.

Figure 3.3 presents the plots for graphical assessment of the goodness-of-fit of the standardized hyperbolic distributions to a pseudo data set of 500 observations which was generated using parameter vector \( \theta = (5, 0.45, 3, 5)' \). These plots were obtainable using \texttt{standHyperbFit(..., plots = TRUE)}. As can be seen, the data is strongly positively skewed. It appears that the fitted curves fit the data well examining the upper two panels of Figure 3.3 (see Appendix B.1). Note that \( \hat{\theta} \) which was in Parameterization 1 was changed to \( \tilde{\theta} \) in Parameterization 2 for plotting purposes by the function \texttt{plot.standHyperbFit()}. This is because the parameterization used by the function \texttt{dhyperb()} is Parameterization 2. The function \texttt{plot.standHyperbFit()} was automatically called by \texttt{standHyperbFit(..., plots = TRUE)}. More example can be found in Appendix B.4.
Figure 3.3: Goodness-of-fit plots for the fitting of parameters $\rho$ and $\zeta$ using standardized hyperbolic distribution. PIT data: Probability-integral-transformed data.
Figures 3.4 and 3.5 show histograms of the differences between the obtained estimates of the shape parameters and their true values by generating 200 samples of size $n = 500$. The function \texttt{standHyperbFit()} and \texttt{hyperbFit()} were then used to fit the same data sets. Overall, the distribution of $\hat{\varepsilon}$ and $\hat{\kappa}$ given by \texttt{standHyperbFit()} appears to centre about 0 more than those given by \texttt{hyperbFit()}. This indicates that, for these tested parameters, \texttt{standHyperbFit()} gave better estimates than \texttt{hyperbFit()}. 

![Histograms](image)

**Figure 3.4:** Histograms of differences between estimates and true parameter values. $\hat{\varepsilon} = \hat{\rho} - \rho$ and $\hat{\kappa} = \hat{\zeta} - \zeta$ where $\rho = 0$, $\zeta = 0.235$, $\delta = 1$ and $\mu = 10$. Column A: differences obtained by \texttt{standHyperbFit()}. Column B: differences obtained by \texttt{hyperbFit()}. 

85
Figure 3.5: Histograms of differences between estimates and true parameter values. \( \hat{\varepsilon} = \hat{\rho} - \rho \) and \( \hat{\kappa} = \hat{\zeta} - \zeta \) where \( \rho = -0.444 \), \( \zeta = 0.235 \), \( \delta = 1 \) and \( \mu = 5 \). Column A: differences obtained by \texttt{standHyperbFit()} . Column B: differences obtained by \texttt{hyperbFit()} .

### 3.4.5 Fitting the standardized hyperbolic distribution to real financial data

The following code gave access to real return data of Disney stock on the Dow Jones index. This data was used to obtain the estimates of the shape parameters of the standardized hyperbolic distribution.

```r
require(QRMlib)
data(DJ)
Ret.DJ <- mk.returns(DJ)
window1.start <- timeDate("01/01/1993", format="%m/%d/%Y")
window1.end <- timeDate("12/31/2000", format="%m/%d/%Y")
```

86
sample1 <- (seriesPositions(Ret.DJ) > window1.start & seriesPositions(Ret.DJ) < window1.end)
DJ30daily <- Ret.DJ[sample1,]
DJ30daily <- 100 * seriesData(DJ30daily)
rseries <- DJ30daily[, "DIS"]
par(mfrow = c(2, 2))
standHyperbFit(rseries, optMethod = "BFGS", plots = TRUE)

The fitting outputs are

Data: rseries
Estimates:

<table>
<thead>
<tr>
<th>mu</th>
<th>delta</th>
<th>rho</th>
<th>zeta</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.062</td>
<td>0.140</td>
<td>-0.008</td>
<td>0.099</td>
</tr>
</tbody>
</table>

maxLike: -2752
Method: BFGS
Convergence code: 0
Iterations: 28
Sd Error Estimates:

<table>
<thead>
<tr>
<th>sdRhoHat</th>
<th>sdZetaHat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.016</td>
<td>0.022</td>
</tr>
</tbody>
</table>

These indicate the fitted distribution is slightly negatively skewed and the kurtosis is not significantly greater than that of the normal distribution. In addition, the function optim(..., method = "BFGS") successfully converged to give the vector of estimated parameters \( \hat{\theta} = (0.062, 0.139, -0.009, 0.104)' \) and the maximum log-likelihood of \(-2751.6\) after 28 iterations. Note that \( \hat{\mu} \) and \( \hat{\delta} \) were considered as nuisance parameters in the fitting of the standardized hyperbolic distribution.

Figure 3.6 shows that the fitted model describes the symmetry of the data very well. Note that the univariate normal inverse Gaussian was used to fit this data using fitNH() in package QRMlib.
Figure 3.6: Plots of fitting standardized hyperbolic distribution to real financial return data.

3.5 Maximum-likelihood estimation of three subclasses of the generalized inverse Gaussian distributions

This section proposes a symbolic method to find the MLE of any subclass of the GIG with $\lambda = \pm (j + \frac{1}{2})$, $j = 0, 1, 2, 3, \ldots$. However, for the purpose of fitting the multivariate generalised hyperbolic distribution in Chapter 4 only the MLE of the
cases, where \( \lambda = -\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \) were derived. These values of \( \lambda \) cover the important subclasses of the multi variate hyperbolic distributions. They are the multivariate normal inverse Gaussian distribution \( (\lambda = -\frac{1}{2}) \), the two and four dimensional hyperbolic distributions \( (\lambda = \frac{3}{2} \text{ and } \frac{5}{2}) \).

If \( \mathbf{w} = w_1, \ldots, w_n \) is a sample of i.i.d. observations from the GIG with density function \( (1.1) \) and parameter \( \lambda = \pm(j + \frac{1}{2}) \), then the exact log-likelihood function can be written as

\[
L(\chi, \psi) = \frac{n\lambda}{2} \log \left( \frac{\psi}{\chi} \right) - n \log(2K_\lambda(z)) + (\lambda - 1) \sum_{i=1}^{n} \log(w_i) - \frac{(\chi \Lambda^* + \psi \Delta^*)}{2}, \quad (3.14)
\]

where

\[
\Delta^* = \sum_{i=1}^{n} w_i, \quad \Lambda^* = \sum_{i=1}^{n} w_i^{-1}, \quad z = \sqrt{\psi \chi}.
\]

### 3.5.1 Symbolic maximum likelihood estimators

When \( \lambda = -\frac{1}{2} \)

For this value of \( \lambda \), the MLEs of \( \chi \) and \( \psi \) are scalars which are given by

\[
M_{\chi} = \frac{n^3}{(-n^2 + \Lambda^* \Delta^*) \Lambda^*}, \quad M_{\psi} = \frac{\Lambda^* n}{-n^2 + \Lambda^* \Delta^*}.
\]

When \( \lambda = \frac{3}{2} \)

The MLE of parameter \( \chi \) is given by

\[
M_{\psi} = -\frac{n^2 \Delta^* \Lambda^* \mathbf{Z} - 3n^2 \mathbf{Z} - 3n^2 + \Delta^* \Lambda^*}{(-n^2 + \Delta^* \Lambda^*) \Delta^* (\mathbf{Z} + 1)}
\]

and the MLE of \( \psi \) is

\[
M_{\chi} = \frac{\Delta^* (\mathbf{Z} + 1)}{n}
\]

89
where \( \mathbf{Z} \) is the solution vector of the polynomial

\[
\mathbf{Z} = \text{RootOf} \left( c_0 + c_1 Z + c_2 Z^2 \right)
\]

with coefficients

\[
\begin{align*}
c_0 &= -3 n^2 + \Delta^* \Lambda^*, \\
c_1 &= -3 n^2 + 2 \Delta^* \Lambda^*, \\
c_2 &= -n^2 + \Delta^* \Lambda^*.
\end{align*}
\]

**When** \( \lambda = \frac{5}{2} \)

The symbolic MLE of parameter \( \chi \) is given by

\[
M_\chi = \frac{n \mathbf{Z}^2 (\mathbf{Z} + 1)}{\Delta^* (3 \mathbf{Z} + \mathbf{Z}^2 + 3)}
\]

and the MLE for parameter \( \psi \) is

\[
M_\psi = \frac{\mathbf{Z}^2}{M_\chi}
\]

where \( \mathbf{Z} \) is the vector solution of the following polynomial

\[
\mathbf{Z} = \text{RootOf} \left( \sum_{j=0}^{4} c_j Z^j \right)
\]

with coefficients

\[
\begin{align*}
c_0 &= 9 \Lambda^* \Delta^* - 15n^2, \\
c_1 &= 18 \Lambda^* \Delta^* - 30n^2, \\
c_2 &= 15 \Lambda^* \Delta^* - 21n^2, \\
c_3 &= 6 \Lambda^* \Delta^* - 7n^2, \\
c_4 &= \Lambda^* \Delta^* - n^2.
\end{align*}
\]
3.5.2 Evaluation of the maximum-likelihood estimators

The function `symFitSubGIG()` , which is described in Appendix B.5, finds mle of these subclasses of the GIG by evaluating the above MLEs. The function is numerically tested by generating 3000 data sets of sizes \( n = 20, 50 \) using a range of testing parameters. It is observed that:

- the first-derivative test is met because the LHS of the inequalities in (3.9) is less than \( \epsilon^* = 10^{-12} \),
- the second-derivative test is always satisfied, and the eigenvalues of the Hessian were evaluated,
- the imaginary parts of the estimates, which already met the first and second-derivative tests are less than \( \epsilon^* = 10^{-12} \),
- the uniqueness condition is always satisfied.

However, it is not possible to analytically prove that the obtained extremum is a global maximum of the log-likelihood. From numerical examinations, it appears that the global maximum is reached. Using the function `symFitSubGIG()` to fit a pseudo data set of \( n = 50 \) generated using \( \theta = (\frac{3}{2}, 5, 3)' \) the below output was obtainable.

```
Data: data
Estimates:
  chi  psi
  8.12 4.16
maxLike : -132
1st deriv test:
  dchi  dpsi
  1.28e-15 -6.03e-15
2rd deriv test:
  -0.187 -8
```

This output indicates the estimates are \( \hat{\chi} = 8.12 \) and \( \hat{\psi} = 4.16 \). The first derivative test (3.1) is satisfied because both \( 1.28 \times 10^{-15} \) and \( -6.03 \times 10^{-15} \) are less than \( \epsilon^* \).
Both eigenvalues of the Hessian matrix evaluated at these values of mle are negative which indicate the optimum is a maximum.

### 3.5.3 Numerical examination

Figure 3.7 presents the plots for graphical assessment of the goodness-of-fit of the fitting the GIG using parameter vector $\theta = (3/2, 5, 25)'$ to generate a sample of 50 observations. As can be seen, the data is slightly positively skewed. Even with this sample size the fitted curves (i.e., density and log-density) appear to fit the data (i.e., histogram and log-histogram) well. The fitting of other subclasses of the GIG to the data is described in Appendix B.5.

Figures 3.8–3.10 show histograms of the differences between the obtained estimates of parameter $\chi$ and $\psi$ and their true values by generating 300 samples of sizes $n = 20, 50$ and 100 using $\theta = (-1/2, 5, 3)'$, $\theta = (3/2, 9, 1)'$ and $\theta = (5/2, 5, 2)'$. As can be seen, the variation of the difference between estimates of a parameter and its true values decreases when sample size increases, which is consistent with the maximum-likelihood theory. However, $\hat{\varepsilon}$ and $\hat{\kappa}$ are centred around 0 even for sample size $n = 20$. `symFitSubGIG()` is going to be added to the package `HyperbolicDist` so it works under Parameterization 1 of this package, see Appendices B.2.3 and B.2.4.
Figure 3.7: Goodness-of-fit plots for the fitting of parameters $\chi$ and $\psi$. PIT: Probability-integral-transformed data.

3.5.4 Symbolic versus numeric fitting of the subclasses of the GIG

Computational stability of `symFitSubGIG()` and that of `nume.subgigFit()` are compared using the function `subgigFitFuncs.test()` in Appendix B.5.3. Both of the fitting functions find the mle by optimizing log-likelihood (3.14) using the same data set. The only difference is `symFitSubGIG()` obtains mle symbolically by evaluating the derived MLEs, while `nume.subgigFit()` uses R numeric optimization routines `optim()`. The symbolic method does not require starting values to initialize the optimization.
process, while the numeric does. The optimization methods implemented by `optim()` include the BFGS and Nelder–Mead algorithm. The subscript BFGS and N–M denote the mle obtained by the former and latter respectively. The objective here is to show that computational stability of the function `nume.subgigFit()` is susceptible to changes in starting values and the optimization method (i.e., BFGS and Nelder–Mead). The subscript SV stands for starting values of parameters \( \chi \) and \( \psi \). Starting values \( \chi_{SV} \) and \( \psi_{SV} \) were used by `optim(..., method = “BFGS”)` and `optim(..., method = “Nelder – Mead”)` only.
Figure 3.9: Histograms of differences between estimates and true parameter values. \( \lambda = \frac{3}{2}, \hat{\varepsilon} = \hat{\chi} - \chi \) and \( \hat{\kappa} = \hat{\psi} - \psi \) where \( \chi = 9, \psi = 1 \). First row \( n = 20 \), second row \( n = 50 \), third row \( n = 100 \).

For each value of \( \lambda \), a pseudo data set was generated using randomly sampled values of \( \chi \) and \( \psi \). The data was then used to find the mle, where the starting values for the numeric approach were also randomly chosen. Table 3.2 shows that the symbolic approach was stable and gave consistent estimates. However, the numerical function `nume.subgigFit()` was unstable even for a small changes in starting values. Significantly more extensive tests are described in Appendix B.5.3.
Table 3.2: Symbols o indicate failure to start an optimization process.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>3.33</td>
<td>8.82</td>
<td>3.33</td>
<td>8.82</td>
<td>3.33</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3.33</td>
<td>8.82</td>
<td>3.33</td>
<td>8.82</td>
<td>3.33</td>
</tr>
</tbody>
</table>

\[ \frac{\theta}{\theta} = \left( \frac{3.33 \cdot 3.33}{8.82} \right) = \theta \]

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>2</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>1.94</td>
<td>4.66</td>
<td>1.94</td>
<td>4.66</td>
<td>1.94</td>
</tr>
<tr>
<td>2</td>
<td>1.94</td>
<td>4.66</td>
<td>1.94</td>
<td>4.66</td>
<td>1.94</td>
<td>4.66</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1.94</td>
<td>4.66</td>
<td>1.94</td>
<td>4.66</td>
<td>1.94</td>
</tr>
</tbody>
</table>

\[ \frac{\theta}{\theta} = \left( \frac{1.94 \cdot 1.94}{4.66} \right) = \theta \]

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>4</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>5.83</td>
<td>0.10</td>
<td>5.83</td>
<td>0.10</td>
<td>5.83</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>5.83</td>
<td>0.10</td>
<td>5.83</td>
<td>0.10</td>
<td>5.83</td>
</tr>
</tbody>
</table>

\[ \frac{\theta}{\theta} = \left( \frac{5.83 \cdot 5.83}{0.10} \right) = \theta \]
Figure 3.10: Histograms of differences between estimates and true parameter values. \( \lambda = \frac{5}{12}, \hat{\varepsilon} = \hat{\chi} - \chi \) and \( \hat{\kappa} = \hat{\psi} - \psi \) where \( \chi = 5, \psi = 2 \). First row \( n = 20 \), second row \( n = 50 \), third row \( n = 100 \).

### 3.6 Conclusions

This chapter presents a symbolic approach to find maximum likelihood estimates of the shape parameter of the hyperbolic distribution. The strengths of this method are it does not depend on starting values and it gives consistent mle even for small sample sizes. However, its use is restricted by the assumptions: \( \delta, \mu \) known and \( \zeta \geq 22 \). The numeric approach to fit the SHYP allows these assumptions to be relaxed. However, the fitting routine is unstable when sample sizes \( n < 500 \) and the fitted parame-
ters depended critically on starting values given by \texttt{hyperbFitStart()}. Secondly, the chapter derives MLEs of three subclasses of the GIG for \( \lambda = -\frac{1}{2}, \frac{3}{2} \) and \( \frac{5}{2} \). After a series of tests, it appeared that the symbolic fitting functions were stable and gave good estimates of parameter \( \chi \) and \( \psi \) even for small sample sizes. These MLEs are to be used in the next chapter to obtain mle of the subclasses of the multivariate hyperbolic and multivariate normal inverse Gaussian distribution.
Chapter 4

Calibration of subclasses of the multivariate generalized hyperbolic distributions

4.1 Introduction

Section 1.3 shows the MGH was formed by the normal mean-variance mixture in which the GIG was called the mixing distribution. This fact allows the application of the EM algorithm in [Dempster et al. (1977)] to find mle of the family of MGH distributions. [Protassov (2004)] proposed a method to find maximum-likelihood estimates of the MGH with arbitrary dimension and fixed $\lambda$. [Wenbo (2005)] advanced the algorithm in Protassov (2004) and suggested that his algorithm solved the stability problem reported by Protassov. [McNeil et al. (2005)] appears to be the most well-cited reference.

This chapter applies the symbolically derived MLEs of the subclasses of the GIG in Section 3.5 to propose an EM-based function, $\text{symFitmNH()}$, for obtaining mle of the $d$-dimensional hyperbolic distributions ($d$-DHYP) with $d = 2, 4$ (2-DHYP, 4-DHYP) and the MNIG for any dimensions. Note that the proposed approach can
be extended to fit the $d$-DHYP with higher dimensions. Comparing the functions used to fit subclasses of the MGH, such as \texttt{fit.mNH()} (in R package \texttt{QRMlib}) or \texttt{fit.hypmv()} and \texttt{fit.NIGmv()} (in R package \texttt{ghyp}), the function \texttt{symFitmNH()} has faster computational speed and stability.

This chapter can be structured as follows. Section 4.2 gives brief descriptions of the EM algorithm, its variant called MCECM and their application for fitting the MGH. Section 4.3 describes the approach of using the derived MLEs in Section 3.5 to obtain mle of the mentioned subclasses of the MGH. Section 4.4 is for numerical examination using both pseudo and real data. Section 4.5 summarizes the chapter.

### 4.2 Expectation Maximization algorithm

Each iteration of the EM algorithm involves two steps, which are called the Expectation (E-step) and Maximization (M-step). It was pointed out on p.139 of \cite{Gentle2004} that the EM algorithm allows the creation of a parameter estimation of the same parameters with “augmented data”, $Y_{\text{aug}}$, from which it is possible to work out the mle in a simpler way. The authors termed the augmented and observed, $Y_{\text{obs}}$, data “complete” and “incomplete” data respectively. Likewise, the corresponding likelihoods were termed “complete-data likelihood” and the “incomplete-data likelihood”, respectively. The term “data augmentation” was also used extensively in \cite{Meng1997} who mentioned that “the key feature for the popularity of the EM algorithm as a powerful yet simple tool is the algorithm formulates a complete-likelihood and then exploits its simplicity to find mle of the incomplete-likelihood”. The augmented part of the data could be called “missing data” which is also a widely-used term in the EM literature, see \cite{Dempster1977} and \cite{Liu1994}. Moreover, it was noted on p.139 of \cite{Gentle2004} that “The missing data need not necessarily be missing in the practical sense of the word. It
may just be a conceptually convenient technical device.”

As an iteration-based algorithm, the EM requires initial value $\theta^{(0)} \in \Theta$, it then finds $\theta^*$ (i.e., a stationary point of $L(\theta|Y_{\text{obs}})$ following the below steps ($t = 0, 1, \ldots$):

- **E-step**: imputes the augmented data log-likelihood $L(\theta|Y_{\text{aug}})$ by

  $$Q(\theta|\theta^{[t]}) = \mathbb{E}[L(\theta|Y_{\text{aug}})|Y_{\text{obs}}, \theta^{[t]}].$$

- **M-step**: determines $\theta^{[t+1]}$ by maximizing the imputed log-likelihood $Q(\theta|\theta^{[t]})$

  $$Q(\theta^{[t+1]}|\theta^{[t]}) \geq Q(\theta|\theta^{[t]}).$$

The use of $t$ to index the iteration was adopted from the convention used on p.515 of Meng and Van Dyk (1997). This should not be confused with the $t$-variable or $t$-model. Computationally, the EM algorithm also requires a stopping criteria to cease the iterations. The standard stopping rule in the literature is

$$\left| \frac{L^{[t]} - L^{[t+1]}}{L^{[t]}} \right| < \text{tol}_{\text{LL}} \quad (4.1)$$

where $L^{[t]}$ is the value of the log-likelihood after $t$ iterations and $\text{tol}$ is the tolerance level.

**A new stopping rule based on changes in parameter estimates of the mixing distributions**

The use of (4.1) as a stopping criterion for the EM algorithm is a common practice in fitting uni- and multi-variate generalized hyperbolic distributions e.g. function `fit.hypuv()`, `fit.ghypmv()` in package `ghyp` or `fit.NH()` and `fit.mNH()` in package `QRMlib`. However, it was pointed out by Dr Murray Jorgensen, an expert in the EM algorithm (Pers. Comm Jorgensen., M, November, 2011), that this practice may be criticized because the log-likelihood depends on an arbitrary constant. Thus,
function `symFitmNH()` of this chapter allows the user to halt the fitting algorithm using either the criterion based on log-likelihood (4.1) or changes in parameter estimates of the mixing distribution. Let \( \beta = (\hat{\chi}, \hat{\psi})' \) be the vector of the estimates of these parameters, the second stopping rule after \( t \) iterations is given by

\[
\sum |\beta^{[t]} - \beta^{[t+1]}| < 10^{-4}, \quad \text{say.}
\]

The function `symFitmNH()` was tested extensively using both criterions. Results showed that it worked stably and the maximum log-likelihood values obtained by different stopping methods were not very different.

### 4.2.1 Variants of the EM algorithm

The E-step and M-step described above give a description of a complete standard EM algorithm. There has been a significant effort to speed up the algorithm while maintaining its very appealing simplicity and monotone convergence properties. [Meng and Van Dyk (1997)] stated “The convergence rate of the algorithm is determined by the fraction of missing information and the data augmentation scheme that one uses for constructing the augmented data likelihood determines the speed of the algorithm. It has been well-recognized since [Dempster et al. (1977)] that by augmenting less, one can develop a faster algorithm”. To achieve this, [Meng and Rubin (1993)] suggested an extension of the EM algorithm called Expectation Conditional Maximization (ECM) whose idea was to replace the M-step of each EM iteration with a sequence of \( S > 1 \) conditional or constrained maximization (CM) steps. Each of these CM steps maximizes the expected complete-data log-likelihood found in the preceding E-step subject to constraints on \( \theta \). The constraints are to ensure that the maximization is done over the full parameter space of \( \theta \). That is, each CM-step maximizes the expected complete-data log-likelihood over some function of \( \theta \), say \( \theta_s = \theta_s(\theta)(s = 1, ..., S) \). The rationale is because the CM maximization are over
smaller-dimensional spaces, often they are simpler and faster than the correspond-
ing full maximization required by the M-step of standard EM. Moreover, [Meng and
Van Dyk (1997)] discussed what they termed model reduction where the parameters \( \theta \) of the complete-likelihood is divided into \( \theta = (\theta_1, \theta_2)' \) such that \( L(\theta | Y_{\text{obs}}) \) is easy to maximize for \( \theta_1 \) given \( \theta_2 \), and vice versa.

Multi-cycle, expectation, conditional maximization algorithm

[Meng and Rubin (1993)] proposed an extension of the ECM algorithm called multi-
cycle, expectation, conditional maximization (MCECM), which inserts an E-step
before some of the CM-steps. Computationally, the MCECM is believed to con-
verge faster than the standard EM algorithm because it combines the strength of
the ECM (i.e., augmenting less) with model reduction. The MCECM algorithm is
the only algorithm which was implemented in multivariate fitting functions of the
package QRMlib and ghyp. But both EM and MCECM were implemented in the
function symFitmNH(). It will be showed latter that there is no significant differ-
ence in computational speed between the EM and MCECM algorithm when they
were used to fit the 2-DHYP, 4-DHYP and MNIG where the mixing distribution
was maximized symbolically. This illustrates that the MCECM algorithm does not
always converge faster than the EM algorithm which is consistent with a statement
on p.524 of [Meng and Van Dyk (1997)].

4.3 Parameter estimations

This section uses the derived MLEs of the subclasses of the GIG to find mle of the
three subclasses of the MGH.
4.3.1 Multivariate generalized hyperbolic distribution

Let $x$ be an i.i.d. data set drawn from the MGH distribution with $\theta = (\lambda, \chi, \psi, \mu, \Sigma, \gamma)'$. Since the MGH distribution is formed by normal mean-variance given by (1.6), the vector of parameters can be divided to give $\theta = (\theta_1, \theta_2)'$ (i.e., model reduction) where

$\theta_1 = (\mu, \Sigma, \gamma)'$,

$\theta_2 = (\lambda, \chi, \psi)'$.

The MGH log-likelihood can be written as

$$L(\theta|x_1 \ldots x_n, w_1 \ldots w_n) = \sum_{i=1}^{n} \log f_{x|w}(x_i|w_i; \theta_1) + \sum_{i=1}^{n} \log g_w(w_i; \theta_2). \tag{4.2}$$

The E-step is applied to calculate the conditional expectation of the augmented log-likelihood (4.2) using data $x_1, \ldots, x_n$ and parameter values $\theta^{[t]}$. The objective function is

$$Q(\theta; \theta^{[t]}) = E \left[ \log L(\theta; x_1, \ldots, x_n, w_1, \ldots, w_n|x_1, \ldots, x_n; \theta^{[t]}) \right].$$

The main point for utilizing the EM algorithm is to overcome the latency of data $w_1, \ldots, w_n$ so, in practice, performing the E-step is equivalent to replacing any function $g(w_i)$ of the latent mixing variables which arise in (4.2) by the quantities $E(g(w)|x_i, \theta^{[t]})$. These quantities can be calculated by recognizing that the conditional density of $w_i$ given $x_i$ satisfies $f_{w|x}(w|x_i; \theta) \propto f_{w,x}(w, x_i; \theta)$ (The likelihood of the mixing distribution given the observed data is proportionate to the complete data likelihood.), up to some constant of proportionality. Thus, it can be deduced from (1.7) that $w_i|x_i \sim \text{GIG}(\tilde{\lambda}, \tilde{\chi}, \tilde{\psi})$ where

$$\tilde{\lambda} = \lambda - \frac{1}{2} d, \quad \tilde{\chi} = (x_i - \mu)' \Sigma^{-1} (x_i - \mu) + \chi, \quad \tilde{\psi} = \psi + \gamma' \Sigma^{-1} \gamma.$$
The E-step now becomes the calculation of the expected values of the GIG using well-known formula

\[ E(W^\alpha) = \left( \frac{\tilde{\chi}}{\tilde{\psi}} \right)^{\alpha/2} K_{\lambda+\alpha} \left( \frac{\tilde{\chi}}{\tilde{\psi}} \right)^{1/2} / K_{\lambda} \left( \frac{\tilde{\chi}}{\tilde{\psi}} \right)^{1/2}. \]

For \( \alpha = \pm 1 \), the expected values of the latent variable \( w \) at iteration \( t \) are given by

\[ \hat{\Delta}_i^{[t]} = E[w_i|x_i; \theta^{[t]}], \quad \hat{\Lambda}_i^{[t]} = E(w_i^{-1}|x_i; \theta^{[t]}) \]

which give

\[ \Delta^* = \sum_{i=1}^{n} \hat{\Delta}_i^{[t]}, \quad \Lambda^* = \sum_{i=1}^{n} \hat{\Lambda}_i^{[t]} \]

(4.3)

The M-step has two quantities to maximize which come from the 2 terms in the RHS of (4.2). These quantities are denoted \( Q_1 (\mu, \Sigma, \gamma; \theta^{[t]}) \) and \( Q_2 (\lambda, \chi, \psi; \theta^{[t]}) \).

- Quantity \( Q_1 \) is optimized symbolically using the formulae on p.83 of [McNeil et al. (2005)] to obtain mle of \( \hat{\mu}, \hat{\Sigma} \) and \( \hat{\gamma} \). Note that the distributions \( \text{MGH}_d (\lambda, \chi k^{-1}, k \psi, k \Sigma, k \gamma) \) and \( \text{MGH}_d (\lambda, \chi, \psi, \mu, \Sigma, \gamma) \) are identical for any \( k > 0 \), which causes a well-known issue called identifiability problem. This can be solved by constraining the determinant \( |\Sigma| \) to be a particular value. In [Blæsild (1981)] and [Protassov (2004)] the value of \( |\Sigma| \) is set to unity. Following [McNeil et al. (2005)] the determinant of sample covariance matrix \( |S| \) is used to set

\[ |\Sigma|^{t+1} = \frac{|S|^{1/d} \Sigma^{[t+1]} |\Sigma^{[t]}|^{1/d}}{\Sigma^{[t+1]}|^{1/d}}. \]

- Using the density function of the GIG given by (1.1), the optimization of \( Q_2 \)
is to maximize

$$(\lambda - 1) \sum_{i=1}^{n} \xi_i^{[t]} - \frac{1}{2} \psi \sum_{i=1}^{n} \Delta_i^{[t]} - \frac{1}{2} \chi \sum_{i=1}^{n} \Lambda_i^{[t]} = \frac{1}{2} n \lambda \log(\chi) + \frac{1}{2} n \lambda \log(\psi) - n \log(2 K_\lambda(\sqrt{\chi \psi})) \quad (4.4)$$

where

$$\hat{\xi}_i^{[t]} = E(\log(w_i) | x_i; \theta^{[t]}) \quad (4.5)$$

Note that the calculation of (4.5) involves derivatives of a Bessel function of the second kind.

**Obtaining mle of the MGH using MCECM algorithm**

The above section describes a complete EM algorithm for fitting the MGH distributions. MCECM is the only algorithm which was implemented for fitting the MGH in the literature because it is hoped that the MCECM converges faster than the standard EM algorithm. However, both the EM and MCECM algorithms were implemented here so that their computing speed can be compared.

The MCECM algorithm for obtaining mle of the MGH can be described as follows: Assuming the parameters $\mu$, $\Sigma$ and $\gamma$ have been updated first in iteration $t$, the value of $\theta$ can be defined as

$$\theta^{[t,2]} = (\lambda^{[t]}, \chi^{[t]}, \psi^{[t]}, \mu^{[t+1]}, \Sigma^{[t+1]}, \gamma^{t+1})',$$

recalculate the quantities in (4.3) and (4.5) and then maximize $Q_2(\lambda, \chi, \psi; \theta^{[t,2]})$ in (4.4) (cf. pp.82–3 of McNeil et al. 2005).
4.3.2 Symbolically fitting subclasses of the multivariate generalized hyperbolic

Parameter estimations of the subclasses of the MGH can be implemented straightforwardly by fixing $\lambda$ and optimizing (4.4) with respect to the remaining parameters. The fitting of $Q_1$ remains unchanged. In these cases, the calculation of (4.5) is not required because the first derivative of the log-likelihood (4.4) with respect to $\lambda$ equals zero. In the literature, quantity $Q_2$ (with $\lambda$ known) has been optimized numerically using the function `optim()` (in the R package `base`). However, $Q_2$ is optimized symbolically in this chapter by plugging (4.3) into the MLEs derived in Section 3.5. This means that both $Q_1$ and $Q_2$ are optimized symbolically by `symFitmNH()`, see Appendix C.2.

4.4 Numerical examination

This section compares computational speed between `fit.mNH()` and `symFitmNH()` when both of the functions use the MCECM algorithm. Difference in computational speed between the EM and MCECM algorithm of `symFitmNH()` is also of interest.

4.4.1 Technical details

Two important technical details include:

1. Both `fit.mNH()` and `symFitmNH()` use the same tolerance level $tol = 10^{-10}$ as the stopping rule. This is the default stopping criterion of the former. It is obvious from (4.1) that $tol$ is a crucial factor for convergence speed, stability and accuracy of an EM based fitting routine.

2. Both `fit.mNH()` and `symFitmNH()` fit the MNIG distribution (i.e., $\lambda = -\frac{1}{2}$) to the data. However the former fits the multivariate hyperbolic distribution, which is also a subclass of the MGH with $\lambda = 1$ while the latter fits the
$d$-DHYP (e.g., 2-DHYP, 4-DHYP) multivariate hyperbolic distribution (i.e., subclasses with $\lambda = \frac{d+1}{2}$), (see p.80 of McNeil et al. (2005)). Consequently, the pseudo data to be generated for computational speed comparison between these routines in cases other than the MNIG is multivariate hyperbolic data (i.e., $\lambda = 1$). This is to ensure that \texttt{fit.mNH()} fits the data that it is intended for. In these cases, this chapter function, \texttt{symFitmNH()}, follows the approach used by the function \texttt{fit.hypmv()} in package \texttt{ghyp} to treat any multivariate fitted data as $d$-DHYP data (i.e., $\lambda = \frac{d+1}{2}$). The following R code shows how $\lambda$ is set by the function \texttt{fit.hypmv()} using dimension $d$.

```r
fit.hypmv <- function (...) {
  call <- match.call(expand.dots = TRUE)
  if (!is.null(list(...)$lambda)) {
    stop("Do not submit lambda! Lambda is defined as 
       (dimension+1)/2.\n")
  }
  lambda <- (min(dim(data)) + 1)/2
}
```

As a result, the values of $\lambda$ to be output by \texttt{symFitmNH()} are $\frac{3}{2}$ and $\frac{5}{2}$ for 2-DHYP and 4-DHYP respectively.

3. Both \texttt{fit.mNH()} and \texttt{symFitmNH()} use the same number of iterations $\texttt{nit}$ and parameterization as specified by (1.7).
4.4.2 Comparison of fitting outputs

If fit.mNH() and symFitmNH() are used to fit MNIG data then the only difference between them is the method used for optimizing quantity $Q_2$. This means that the fitting results by these two functions should be the same if symFitmNH() is correctly implemented. The R code in Appendix C.2 confirms that this is the case.

4.4.3 Comparison of computational speed

This section compares the computational speed of fit.mNH() and symFitmNH() by fitting 200 samples of size $n = 30$. The parameter values used to generate the data are presented in Appendix C.3.

As can be seen from Figure 4.1, speed of symFitmNH() represented by Boxplots 1 and 2 is significantly faster than fit.mNH(). This improvement is due to the feature that the former optimizes quantity $Q_2$ symbolically. In addition, it appears that there is no difference in the computational speed of the EM and MCECM algorithm. This shows that the latter does not always result in faster convergence speed.
4.4.4 Fitting real multivariate data

This section uses real daily return data which is available at [McNeil and Ulman (2010)](http://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average) to find mle of the 2-DHYP, 4-DHYP and MNIG distribution. The data is the return on the Dow-Jones Industrial Average (DJ30) stock index\(^1\). If stocks of two or more companies are used to construct a portfolio of stock, then the portfolio returns can be modelled as a multivariate variable. Specifically, when a portfolio consists of stocks of 2 and 4 companies the 2-DHYP and 4-DHYP distributions are used, respectively. The MNIG can be used for a portfolio of stocks of any numbers of companies (more than 1).

\(^1\) [http://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average](http://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average) contains comprehensive information about the DJ30
<table>
<thead>
<tr>
<th>Trading codes</th>
<th>Company names</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXP</td>
<td>American Express</td>
</tr>
<tr>
<td>BA</td>
<td>Boeing</td>
</tr>
<tr>
<td>MSFT</td>
<td>Microsoft</td>
</tr>
<tr>
<td>JPM</td>
<td>JPMorgan Chase</td>
</tr>
<tr>
<td>KO</td>
<td>Coca-cola</td>
</tr>
<tr>
<td>HWP</td>
<td>Hewlett-Packard</td>
</tr>
</tbody>
</table>

Table 4.1: Fitting of 2-DHYP. Starting values $\frac{3}{2}, 10, 5,$ and $n = 30.$

$\hat{\chi} = 6.611 \times 10^{-3}, \hat{\psi} = 2.813 \times 10^4$

<table>
<thead>
<tr>
<th></th>
<th>MSFT</th>
<th>JPM</th>
<th>KO</th>
<th>HWP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\mu}$</td>
<td>0.040</td>
<td>-0.04</td>
<td>-0.08</td>
<td>0.003</td>
</tr>
<tr>
<td>$\hat{\Sigma}$</td>
<td>2.268</td>
<td>0.762</td>
<td>0.350</td>
<td>0.626</td>
</tr>
<tr>
<td></td>
<td>JPM</td>
<td>0.762</td>
<td>1.653</td>
<td>-0.433</td>
</tr>
<tr>
<td></td>
<td>KO</td>
<td>0.350</td>
<td>-0.433</td>
<td>0.494</td>
</tr>
<tr>
<td></td>
<td>HWP</td>
<td>0.626</td>
<td>-0.097</td>
<td>-0.009</td>
</tr>
</tbody>
</table>

Table 4.2: Fitting of 4-DHYP. Starting values $\frac{5}{2}, 15, 5,$ and $n = 50.$

$\hat{\chi} = 3.340 \times 10^{-3}, \hat{\psi} = 5.377 \times 10^4$

<table>
<thead>
<tr>
<th></th>
<th>JPM</th>
<th>AXP</th>
<th>BA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\mu}$</td>
<td>-0.022</td>
<td>0.022</td>
<td>-0.005</td>
</tr>
<tr>
<td>$\hat{\Sigma}$</td>
<td>JPM</td>
<td>1.416</td>
<td>0.664</td>
</tr>
<tr>
<td></td>
<td>AXP</td>
<td>0.664</td>
<td>1.261</td>
</tr>
<tr>
<td></td>
<td>BA</td>
<td>0.164</td>
<td>0.273</td>
</tr>
</tbody>
</table>

Table 4.3: Fitting of 3 dimensions MNIG. Starting values $-\frac{1}{2}, 1, 5,$ and $n = 30.$

4.5 Conclusion

This chapter presents a new EM-based approach for fitting 2-DHYP, 4-DHYP and MNIG (any dimensions) distributions. The obtained function finds mle of the mixing distribution by evaluating symbolically-derived MLE of the GIG for $\lambda = -\frac{1}{2}, \frac{3}{2}$ and $\frac{5}{2}.$ The method can be extended for fitting multivariate hyperbolic distribution of higher dimensions. From extensive numerical tests, it appears that the routine is computationally stable even with small sample sizes. Moreover, its computational speed is significantly faster than $\text{fit.mNH}(\cdot)$ in package QRMlib. Important techni-
cal information of package \texttt{QRMLib} and documentation of \texttt{fit.mNH()} can be found in Appendix C.1 and C.2 respectively.
Chapter 5

Evaluation of the incomplete Bessel function

5.1 Introduction

The cumulative distribution function (CDF) of the GIG involves an integral which is related to the integral of a Bessel function. This chapter is reserved for studying the evaluation of the incomplete Bessel function. It can be considered to be foundational for the next chapter, which deals with calculation of tail probabilities of the GIG.

The modified Bessel function of the second kind, which can be referred to as the Bessel function hereafter without causing confusion, appears with numerous integral representations. However, this chapter specifically derives analytical methods to evaluate the incomplete Bessel functions given by one or other of the forms

\[
K_{\lambda}(z, x) = \frac{\sqrt{\pi}}{\sqrt{2z}} \frac{1}{\Gamma(\lambda + \frac{1}{2})} e^{-z} \int_0^x e^{-\xi} \xi^{\lambda-1/2} \left(1 + \frac{\xi}{2z}\right)^{\lambda-1/2} d\xi, \quad (5.1)
\]

\[
\tilde{K}_{\lambda}(z, x) = \frac{1}{(2z)^\lambda} \int_0^x e^{-\left\{z^2 \xi^2 + 1/(4\xi^2)\right\}} \xi^{-2\lambda-1} d\xi. \quad (5.2)
\]

In addition, the chapter also examines an accurate approach to numerically approx-
imate the incomplete Bessel function given by

\[ \tilde{\mathcal{K}}_{\lambda}(\chi, \psi) = \int_{1}^{\infty} e^{-\left(\chi \xi + \psi / \xi\right)} \xi^{-\lambda - 1} d\xi. \]  

(5.3)

These integrals converge for arbitrary \( z, \psi, \chi \in \mathbb{R}^+ \) and \( \lambda \in \mathbb{R} \). An additional condition for (5.1) is \( \lambda + \frac{1}{2} > 0 \). Some standard references for the Bessel function are Gray and Mathews (1931), Watson (1966) and Temme (1996).

Currently, no research on any analytical approach to evaluate the incomplete Bessel function is known to myself. However, its evaluation by a numerical approach has been studied in the literature and the form presented by (5.3) has attracted most research effort. Hantush and Jacob (1955) proposed the original leaky aquifer function \( \tilde{\mathcal{K}}_{0}(\chi, \psi, 1) \) to analyze the water levels in pumped aquifer systems with finite transmissivity and leakage. However, the name persists even when the order assumes any values and (5.3) is used in other fields of physics. The evaluation of the incomplete Bessel function has been known to be very challenging. Some of the difficulties were discussed in Harris (1997). Hantush and Jacob (1955) applied a method which was based on a combination of numerical integration and infinite series representations. Terras (1981) introduced a Miller algorithm using continued fraction for a quotient of confluent hypergeometric functions. Recently, Harris (2009) presented a method to numerically calculate (5.3) using the relationship between functions \( K_0(z) \) and \( K_1(z) \) and the exponential integral \( E_1(z) \). Slevinsky and Safouhi (2010) derived an algorithm which appears to be the most accurate, robust and flexible program to numerically approximate (5.3).

The chapter is as follows. Section 5.2 derives closed-form expressions of the lower incomplete Bessel function defined by (5.1) and its upper incomplete counterpart when \( \lambda = j + \frac{1}{2} \) with \( j = 0, 1, 2, \ldots \). Section 5.3 derives formulae to evaluate an integral due to Laplace using the Schlömilch transformation. Section 5.4 utilizes the results of Section 5.3 to derive formulae in the form of the error functions for
the incomplete Bessel function (5.2) when \( \lambda = \pm \frac{1}{2} \). Section 5.5 implements an algorithm proposed in \cite{Slevinsky and Safouhi 2010} to numerically approximate the incomplete Bessel function (5.3) for any value of \( \lambda \in \mathbb{R} \). Section 5.6 is reserved for establishing the mathematical relationships between the incomplete Bessel functions defined by (5.2), (5.3) with the incomplete Bessel function in the density function of the GIG. The last section studies the accuracy of the obtained incomplete Bessel functions. In this thesis, Erf() and Erfc() denote the error and complementary error function respectively. The relations between these special functions and the normal cumulative distribution function \( \Phi(x) \) are well-known:

\[
\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt, \\
= \frac{1}{2} \left[ 1 + \text{Erf} \left( \frac{x}{\sqrt{2}} \right) \right], \\
= \frac{1}{2} \text{Erfc} \left( -\frac{x}{\sqrt{2}} \right),
\]

after rearranging

\[
\text{Erf}(x) = 2 \Phi \left( x\sqrt{2} \right) - 1, \\
\text{Erfc}(x) = 2 \Phi \left( -x\sqrt{2} \right).
\]
5.2 Evaluation of the incomplete Bessel function defined by (5.1) using the Whittaker function

5.2.1 Relationship between the Whittaker function and the Bessel function in finite range

The relation between the Bessel function $K_\lambda(z)$ and the Whittaker function $W_{k,m}(z)$ is given by

$$K_\lambda(z) = \sqrt{\frac{\pi}{2z}} W_{k,m}(2z) \quad \text{where} \quad k = 0, m = \lambda,$$

which is well-known, see [Watson (1966)]. However, the relation between the incomplete Bessel function and the Whittaker function has not been documented. This section aims at establishing the relationship between the incomplete Bessel function defined by (5.1) and the Whittaker function. The result will be used to derive closed-form formulae of the incomplete Bessel (5.1) for $\lambda = j + \frac{1}{2}$ with $j \in \mathbb{Z}^*$.

Cauchy’s theorem on the integral of a function round a contour.

**Definition 5.2.1.** A closed curve on the $z$-plane formed by arranging points $A$, $B$, $C$ and $D$ counter-clockwise along the arc of the curve and if $f(z)$ is a continuous function of $z$ at all points on the arc then the integral $\int_{ABCDA} f(z) \, dz$ is called the integral of function $f(z)$ around a contour and can be illustrated as

![Figure 5.1: Integral of function $f(z)$ round a contour](image)

Figure 5.1: Integral of function $f(z)$ round a contour
\textbf{Theorem 1} (Cauchy). If \( f(z) \) is an analytic and continuous function of \( z \) at all points on and inside any contour \( C \) then \( \int_C f(z) \, dz = 0 \), where the integration is taken around the contour.

The theorem holds regardless of the direction and starting point of the integral as long as the contour is closed and regains its value at the final point of \( C \).

\textbf{Whittaker Function}

Whittaker (1903) proposed the Whittaker function \( W_{k,m} \) as one of the solutions of the differential equation

\[ \frac{d^2w}{dz^2} + \left\{ \frac{k}{z} - \frac{m^2 - \frac{1}{4}}{z^2} - \frac{1}{4} \right\} w = 0. \]

Its solution is given by

\[ W_{k,m}(z) = -\frac{\Gamma \left( k + \frac{1}{2} - m \right)}{2\pi i} e^{-z^2/2} z^k \int_C (-t)^{k-1/2+m} \left( 1 + \frac{t}{z} \right)^{k-1/2+m} e^{-t} \, dt \]

(5.4)

provided the contour of integration is chosen to equate the integral to 0. The notation \( (-t)^{-k-1/2+m} \) is to denote

\[ (-t)^{-k-1/2+m} = e^{(-k-1/2+m) \log(-t)} \]

where the real value of \( \log(-t) \) is taken when \( t \) is real and negative. Applying Cauchy’s Theorem, such a contour can be one that starts from positive infinity, circles around zero positively once and returns to the starting point.

More importantly, if the real part of \(-k - \frac{1}{2} + m\) is positive then the path of integration can be deformed to coincide with the real axis in the \( t \)-plane so that the complex integral (5.4) is replaced by the real integral (cf. p.126 of Whittaker 1903)

\[ W_{k,m}(z) = \frac{e^{-z^2/2} z^k}{\Gamma \left( \frac{1}{2} - k + m \right)} \int_0^\infty t^{-k-1/2+m} \left( 1 + \frac{t}{z} \right)^{k-1/2+m} e^{-t} \, dt, \quad z > 0. \]
Important properties of the $W_{k,m}(z)$ function, which were pointed out in Whittaker (1903) include

1. The function $W_{k,m}(z)$ degenerates into an elementary function when $k - \frac{1}{2} \pm m$ is a positive integer or zero. This explains the well-documented feature of the Bessel function, which that is it degenerates into an elementary function when the order is half an odd integer.

2. The function $W_{k,m}(z)$ degenerates into the definite integral of an elementary function when $k - \frac{1}{2} \pm m$ is a negative integer. The incomplete gamma function, logarithm integral, error function, and cosine integral, illustrate this feature.

Function (5.5) can be transformed to various functions by considering special relations between the parameters $k$ and $m$. The functions of the parabolic cylinders and the incomplete gamma function are obtained when $m = \frac{1}{2}$ and $k + m = -\frac{1}{2}$ respectively. The error function can be derived from $W_{k,m}$ function when $k + m = 0$. Specifically, its connection with the lower incomplete gamma function with parameter $\nu$ is

$$
\gamma(\nu, x) = \int_0^x t^{\nu-1} e^{-t} dt = \Gamma(\nu) - x^{(\nu-1)/2} e^{-x/2} W_{(\nu-1)/2, \nu/2}(x)
$$

(5.6)
5.2.2 Closed-form formulae of the incomplete Bessel function when \( \lambda \) equals half of a positive odd integer

When \( \lambda = j + \frac{1}{2} \) with \( j \in \mathbb{Z}^* \) \( (j = 0, 1, 2, 3, \ldots) \) the Bessel function can be written as an expression of elementary function (This is well-known in the literature). Entry 8.468 in Gradshteyn and Ryzhik (2007) gives

\[
K_{j+1/2}(z) = \sqrt{\pi} \frac{e^{-z}}{2z} \sum_{r=0}^{j} \frac{(j + r)!}{r! (j - r)! (2z)^r}.
\]

(5.7)

However, to the best of my knowledge, an expression of the incomplete Bessel function in terms of elementary functions has not been obtained for any value of \( \lambda \). This section aims at deriving closed-form expression of the lower incomplete Bessel function defined by (5.1) and its upper counterpart.

From (2.15) and for a given value \( |\lambda| \) equals half of a positive odd integer, the modified Bessel function of the second kind can be written as

\[
K_\lambda(z) = \sqrt{\frac{\pi}{2z}} e^{-z} \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \int_0^{\infty} e^{-\xi} \xi^{\lambda+r-1/2} d\xi,
\]

(5.8)

where \( s = j + 1 \) with \( j \in \mathbb{Z}^* \) is the number of terms of the summation if the expression in the braces of (5.8) are expanded in the manner which gave (2.15).

Using (5.8), the lower incomplete Bessel functions defined by (5.1) is given by

\[
K_\lambda(z, x) = \sqrt{\frac{\pi}{2z}} e^{-z} \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \int_0^{x} e^{-\xi} \xi^{\lambda+r-1/2} d\xi,
\]

(5.9)

and the upper incomplete counterpart is defined as

\[
K^U_\lambda(z, x) = \sqrt{\frac{\pi}{2z}} e^{-z} \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \int_{x}^{\infty} e^{-\xi} \xi^{\lambda+r-1/2} d\xi.
\]

(5.10)
Theorem 2. Let $\lambda = j + \frac{1}{2}$. Then the incomplete Bessel functions (5.9) and (5.10) are given by

$$K_\lambda(z, x) = A_z \sum_{r=0}^{s-1} \left\{ \frac{1}{\beta r!} \left( \frac{1}{2z} \right)^r \left[ (\alpha - 1)! - x^{\alpha-1} e^{-x} \sum_{\varphi=0}^{\alpha-1} \left( \frac{\alpha}{\varphi} \right) x^{-\varphi} \right] \right\},$$

(5.11)

$$K^U_\lambda(z, x) = A_z \sum_{r=0}^{s-1} \left\{ \frac{1}{\beta r!} \left( \frac{1}{2z} \right)^r \left[ x^{\alpha-1} e^{-x} \sum_{\varphi=0}^{\alpha-1} \left( \frac{\alpha}{\varphi} \right) x^{-\varphi} \right] \right\},$$

(5.12)

respectively where

$$A_z = \sqrt{\frac{\pi}{2z}} e^{-z}, \quad s = \lambda + \frac{1}{2}, \quad \beta = \left( \lambda - r - \frac{1}{2} \right) !$$

and

$$\alpha = \lambda + r + \frac{1}{2}$$

(5.13)

Proof. Using (5.13), equation (5.9) can be rewritten as

$$K_\lambda(z, x) = A_z \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \int_0^x e^{-\xi} \xi^{\alpha-1} d\xi$$

(5.14)

where $\gamma(\alpha, x)$ is the lower incomplete gamma with index $\alpha$. A well-known relation between the incomplete gamma functions is given by

$$\Gamma(\alpha, x) = \Gamma(\alpha) - \gamma(\alpha, x)$$

(5.15)

where $\Gamma(\alpha, x)$ is the upper incomplete gamma function. Applying (5.15), equa-
tion (5.10) becomes

\[ K^U_\lambda(z, x) = \sqrt{\frac{\pi}{2z}} e^{-z} \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \int_x^\infty e^{-\xi} \xi^{\alpha - 1} d\xi \]

\[ = \sqrt{\frac{\pi}{2z}} e^{-z} \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \left[ \Gamma(\alpha) - \gamma(\alpha, x) \right]. \quad (5.16) \]

Plugging the expression of the incomplete gamma function in (5.6) into (5.14) and (5.16). By replacing \( \nu \) by \( \alpha \), the incomplete Bessel functions can now be expressed in terms of the Whittaker function

\[ K_\lambda(z, x) = A_z \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \left[ \Gamma(\alpha) - x^{(\alpha-1)/2} e^{-x/2} \tilde{W} \right] \quad (5.17) \]

\[ K^U_\lambda(z, x) = A_z \sum_{r=0}^{s-1} \frac{1}{\Gamma(\lambda - r + \frac{1}{2}) r!} \left( \frac{1}{2z} \right)^r \left[ x^{(\alpha-1)/2} e^{-x/2}(x) \tilde{W} \right] \quad (5.18) \]

where

\[ \tilde{W} = W_{(\alpha-1)/2, \alpha/2}(x). \]

Equations (5.17) and (5.18) establish the required relationship between the incomplete Bessel function defined by (5.1) and the Whittaker function. The feature that (5.17) and (5.18) are expressible in terms of elementary function or definite integrals of an elementary function, depends on whether \( W_{(\alpha-1)/2, \alpha/2}(x) \) satisfies condition (1) or (2) in Section 5.2.1. These conditions can be written as

\[ \frac{\alpha - 1}{2} - \frac{1}{2} + \frac{\alpha}{2} = r + \lambda - \frac{1}{2} \quad (5.19) \]

or

\[ \frac{\alpha - 1}{2} - \frac{1}{2} - \frac{\alpha}{2} = -1. \]

Since (5.17) and (5.18) depend on both \( r \) and \( \lambda \), only (5.19) is of interest. This means that (5.17) and (5.18) can only be expressions of elementary functions if (5.19) is zero or a positive integer. Given that \( r \in \mathbb{Z}^* \), \( j \in \mathbb{Z}^* \) and \( \lambda = j + \frac{1}{2} \), it is obvious
that the RHS of (5.19) is a non-negative integer. This means that the condition for \( W_{(\alpha-1)/2, \alpha/2}(x) \) degenerating into an elementary function is met. Using the integral representation of the Whittaker function in (5.5), \( W_{(\alpha-1)/2, \alpha/2}(x) \) can be rewritten as

\[
W_{(\alpha-1)/2, \alpha/2}(x) = e^{-x/2} x^{(\alpha-1)/2} \int_0^\infty \left( 1 + \frac{t}{x} \right)^{\alpha-1} e^{-t} dt. \tag{5.20}
\]

Since \( \alpha - 1 = r + \lambda - \frac{1}{2} \), it is legitimate to expand the integrand using the binomial theorem

\[
\left( 1 + \frac{t}{x} \right)^{\alpha-1} e^{-t} = \sum_{\varphi=0}^{\alpha-1} \binom{\alpha-1}{\varphi} \left( \frac{t}{x} \right)^\varphi e^{-t}
\]

so that (5.20) can be rewritten as

\[
W_{(\alpha-1)/2, \alpha/2}(x) = e^{-x/2} x^{(\alpha-1)/2} \sum_{\varphi=0}^{\alpha-1} \binom{\alpha-1}{\varphi} x^{-\varphi} \int_0^\infty t^\varphi e^{-t} dt. \tag{5.21}
\]

As

\[
\Gamma(\varphi + 1) = \varphi!
\]

then (5.21) becomes

\[
W_{(\alpha-1)/2, \alpha/2}(x) = e^{-x/2} x^{(\alpha-1)/2} \sum_{\varphi=0}^{\alpha-1} \binom{\alpha-1}{\varphi} x^{-\varphi} \varphi!. \tag{5.23}
\]

Finally, using (5.22) and plugging (5.23) into (5.17) and (5.18), expressions (5.11) and (5.12) are obtainable as required.

The validity of Theorem 2 and its proof can be theoretically checked by firstly substituting \( \lambda = j + \frac{1}{2} \) and \( s = j + 1 \) into (5.11) and (5.12) and then adding the RHS of these equations. These operations give the RHS of (5.7) as required by the summation of the corresponding left hand sides.

Computationally, the formulae can be implemented to calculate exact value of the incomplete Bessel functions for any \( \lambda = j + \frac{1}{2} \) in a straightforward manner. It was implemented as the R function `closedFormIBF()` in Appendix D.2. The Maple
procedure called closedFormIBFpro(), in Appendix D.3, is used to check the accuracy of these formulae by numerically integrating the integral (5.1). These formulae can also be used to manually derive expressions of the incomplete Bessel function when λ assumes a specific value. For example, when λ = 1/2, by plugging s = 1, β = 1! and α = 1 into (5.12) and (5.11), the following are obtained:

\[ K_{1/2}^U(z, x) = \sqrt{\frac{\pi}{2z}} e^{-(z+x)}, \]
\[ K_{1/2}(z, x) = \sqrt{\frac{\pi}{2z}} e^{-z} \left( 1 - e^{-x} \right). \]

As another example, when \( \lambda = \frac{3}{2}, \) s equals 2 which means r assumes 2 values: 0 and 1. There are two sets of α and β to be calculated. For \( r = 0: \) α = 2, β = 1! = 1. For \( r = 1: \) α = 3 and β = 0! = 1. The following are obtained:

\[ K_{3/2}^U(z, x) = \sqrt{\frac{\pi}{2z}} e^{-(z-x)} \left[ 1 + x + \frac{1}{2z} \left( x^2 + 2x + 2 \right) \right], \]
\[ K_{3/2}(z, x) = \sqrt{\frac{\pi}{2z}} e^{-z} \left[ 1 - e^{-z}x \left( 1 + \frac{1}{x} \right) + \frac{1}{z} - e^{-x} \left( \frac{2 + 2x + x^2}{2z} \right) \right]. \]

In the above examples, the term \(^{(α−1)}_{ϕ}\) is readily obtainable using the so-called Pascal’s triangle.

### 5.3 Evaluation of the Laplace integral in finite ranges

This section proposes a method to evaluate an integral due to Laplace, see Amdeberhan et al. (2010) in finite ranges using the Schlömilch transformation which was also called the Cauchy-Schlömilch transformation. The obtained formulae were then used to derive the lower and upper incomplete Bessel function defined by (5.2) when \( \lambda = \pm \frac{1}{2}. \)

The Schlömilch transformation was introduced in 1848. More recently, its power
has been restated in [Boros and Moll (1998) and Boros and Moll (2001)]. An integral
due to Laplace (cf. p.4 of Amdeberhan et al., 2010)
\[
\int_{0}^{\infty} e^{-(a\xi^2 + b/\xi^2)} d\xi = \frac{1}{2} \sqrt{\frac{\pi}{a}} e^{-2\sqrt{ab}}, \quad a > 0, \ b > 0
\] (5.24)
was the original problem to which the transformation was applied to evaluate. The
Schl"omilch transformation has also been found to have important applications in
statistics for generating new distributions from old ones. Baker (2008) stated that
his paper presented a new method of generating new distributions from old, based
on the little-known Schl"omilch transformation. This exemplifies yet another way of
generating new distributions, that of redistributing probability mass.

**Theorem 3** (Schl"omilch). Let \(a > 0\) and \(b > 0\). Then
\[
\int_{0}^{\infty} f \left( (ax - b/x)^2 \right) \, dx = \frac{1}{a} \int_{0}^{\infty} f(u^2) \, du.
\] (5.25)

**Proof.** Transform the integral \(I\) on the left by \(t = b/(ax)\) and add it to the original
to obtain
\[
2I = \frac{1}{a} \int_{0}^{\infty} f \left( (ax - b/x)^2 \right) \left( a + b/x^2 \right) \, dx.
\]
Let \(u = ax - b/x\), then \(du = (a + b/x^2) \, dx\) so
\[
I = \frac{1}{a} \int_{0}^{\infty} f(u^2) \, du.
\]
\[ \square \]

The proof for Theorem 3 is well-documented, e.g., see Baker (2008). The power
of the Schl"omilch transformation can be observed from the feature that it transforms
integral \(I\) of function \(f(x)\) on \([0, \infty)\) onto the real line \((-\infty, \infty)\) in a symmetrical
manner.

124
The integral (5.24) is directly related to the incomplete Bessel function in (5.2)

\[
\int_{0}^{x} e^{-\left(a^{2}\xi^{2} + b^{2}/\xi^{2}\right)} \xi^{-2\lambda-1} \, d\xi, \quad a > 0, \ b > 0,
\]

by letting \( f(x) = e^{-x} \) in (5.25). Analytical approaches to evaluate (5.26) for any value of \( \lambda \) or the integral (5.24) in finite range have not been documented in the literature. Even popular commercial software package such as Maple 15 and Matlab 2009 were not able to evaluate them symbolically. It was stated on p.1 of Amdeberhan et al. (2010) that the Schlömilch transformation can be used to evaluate “many non-elementary definite integrals, most of which cannot be obtained by symbolic packages.”

### 5.3.1 Evaluation steps

Equation (5.24) is entry 3.325 in Gradshteyn and Ryzhik (2007). This section derives steps to evaluate this integral in two ranges \((0, x)\) and \((x, \infty)\) with \(x > 0\).

**Theorem 4.** Let \(a > 0, \ b > 0\). Then

\[
\int_{0}^{x} e^{-\left(a^{2}\xi^{2} + b^{2}/\xi^{2}\right)} \, d\xi = \frac{e^{-2\sqrt{ab}\sqrt{\pi}}}{4\sqrt{a}} \left[ \hat{Q}_{1} - e^{4\sqrt{ab}}\hat{Q}_{2} \right]
\]

where \(\hat{Q}_{1} = \text{Erfc} \left( \sqrt{b}/x - x\sqrt{a} \right)\) and \(\hat{Q}_{2} = \text{Erfc} \left( \sqrt{b}/x + x\sqrt{a} \right)\).

**Proof.** In order to evaluate the integral

\[
\int_{0}^{x} e^{-\left(a^{2}+b^{2}\xi^{2}\right)} \, d\xi
\]

the first step is use a change of variable \(\zeta = \sqrt{a} \xi\) to obtain

\[
\int_{0}^{x} e^{-\left(a^{2}+b^{2}\zeta^{2}\right)} \, d\zeta = \frac{1}{\sqrt{a}} \int_{0}^{x\sqrt{a}} e^{-\left(c^{2}+ab/\zeta^{2}\right)} \, d\zeta
\]

\[
= \frac{1}{\sqrt{a}} \int_{0}^{x\sqrt{a}} e^{-\left(\psi^{2}+ab/\psi^{2}\right)} \, d\psi.
\]
Let \( c = ab \) and then a change of variable \( \psi = \sqrt{c} \zeta^{-1} \) in (5.30), so it can be rewritten as

\[
\int_0^x e^{-(a\xi^2 + b/\xi^2)} d\xi = \frac{1}{\sqrt{a}} \int_{\sqrt{b}/x}^{\infty} e^{-(\xi^2 + c/\xi^2)} \sqrt{c} \frac{1}{\xi^2} d\xi. \tag{5.31}
\]

From (5.29)–(5.31), it follows that (5.28) equals

\[
\int_0^x e^{-(a\xi^2 + b/\xi^2)} d\xi = \frac{1}{2\sqrt{a}} \left[ \int_0^{x\sqrt{a}} e^{-(\xi^2 + c/\xi^2)} d\xi + \int_{\sqrt{b}/x}^{\infty} e^{-(\psi^2 + c/\psi^2)} \sqrt{c} \frac{1}{\psi^2} d\psi \right]. \tag{5.32}
\]

The following relation holds

\[
\int_{\sqrt{b}/x}^{\infty} e^{-(\psi^2 + c/\psi^2)} d\psi = \int_0^{x\sqrt{a}} e^{-(\xi^2 + c/\xi^2)} \left( 1 + \frac{\sqrt{c}}{\xi^2} \right) d\xi \quad \text{iff} \quad \psi = \sqrt{c} \zeta^{-1}. \tag{5.33}
\]

Plugging both sides of (5.33) into the RHS of (5.32) to have

\[
\frac{1}{2\sqrt{a}} \left[ \int_0^{x\sqrt{a}} e^{-(\xi^2 + c/\xi^2)} \left( 1 + \frac{\sqrt{c}}{\xi^2} \right) d\xi - \int_{\sqrt{b}/x}^{\infty} e^{-(\psi^2 + c/\psi^2)} \left( 1 - \frac{\sqrt{c}}{\psi^2} \right) d\psi \right].
\]

To evaluate

\[
I_1 = e^{-2\sqrt{c}} \int_0^{x\sqrt{a}} e^{-(\xi-\sqrt{c}/\xi)^2} \left( 1 + \frac{\sqrt{c}}{\xi^2} \right) d\xi
\]

let \( \rho = \zeta - \frac{\sqrt{c}}{\xi} \) so \( d\rho = \left( 1 + \frac{\sqrt{c}}{\xi^2} \right) d\zeta \), and

\[
I_1 = e^{-2\sqrt{c}} \int_\sqrt{b/x-x\sqrt{a}}^{\infty} e^{-\rho^2} d\rho = \sqrt{\pi} e^{-2\sqrt{c}} \frac{\text{Erfc} \left( \frac{\sqrt{b}}{x} - x\sqrt{a} \right)}{2}. \tag{5.34}
\]

To evaluate

\[
I_2 = e^{2\sqrt{c}} \int_{\sqrt{b}/x}^{\infty} e^{-(\psi + \sqrt{c}/\psi)^2} \left( 1 - \frac{\sqrt{c}}{\psi^2} \right) d\psi
\]

let \( \rho = \psi + \frac{\sqrt{c}}{\psi} \) so \( d\rho = \left( 1 - \frac{\sqrt{c}}{\psi^2} \right) d\psi \), and

\[
I_2 = e^{2\sqrt{c}} \int_{\sqrt{b}/x+x\sqrt{a}}^{\infty} e^{-\rho^2} d\rho = \sqrt{\pi} e^{2\sqrt{c}} \frac{\text{Erfc} \left( \frac{\sqrt{b}}{x} + x\sqrt{a} \right)}{2}. \tag{5.35}
\]
From (5.34) and (5.35), the evaluation result is given by

\[ \int_{0}^{x} e^{-\left(a\xi^2 + b/\xi^2\right)} d\xi = \frac{e^{-2\sqrt{ab}} \sqrt{\pi}}{4\sqrt{a}} \left[ \hat{Q}_1 - e^{4\sqrt{ab}} \hat{Q}_2 \right] \]  

(5.36)

where \( \hat{Q}_1 = \text{Erfc} \left( \frac{\sqrt{b}}{x} - x \sqrt{a} \right) \) and \( \hat{Q}_2 = \text{Erfc} \left( \frac{\sqrt{b}}{x} + x \sqrt{a} \right) \) as required. \( \square \)

**Collorary 1.**

\[ \int_{x}^{\infty} e^{-\left(a\xi^2 + b/\xi^2\right)} d\xi = \frac{e^{-2\sqrt{ab}} \sqrt{\pi}}{4\sqrt{a}} \left[ 2 - \hat{Q}_1 + e^{4\sqrt{ab}} \hat{Q}_2 \right] . \]  

(5.37)

**Proof.** The sum of the RHS of (5.37) and the RHS of (5.27) equals the RHS of (5.24) as required by the sum of the corresponding LHS of these expressions. \( \square \)

**Collorary 2.**

\[ \int_{0}^{\infty} e^{-\left(a\xi^2 + b/\xi^2\right)} \xi^{-2} d\xi = \int_{0}^{\infty} e^{-\left(b\phi^2 + a/\phi^2\right)} d\phi = \frac{1}{2} \sqrt{\frac{\pi}{b}} e^{-2\sqrt{ab}} . \]  

(5.38)

**Proof.** The integral with respect to \( \phi \) is obtainable by change of variable \( \phi = \xi^{-1} \). It is then evaluated using (5.24). \( \square \)

**Collorary 3.**

\[ \int_{0}^{x} e^{-\left(a\xi^2 + b/\xi^2\right)} \xi^{-2} d\xi = \frac{e^{-2\sqrt{ab}} \sqrt{\pi}}{4\sqrt{b}} \left[ \hat{Q}_1 + e^{4\sqrt{ab}} \hat{Q}_2 \right] . \]  

(5.39)

**Proof.** The following relation holds

\[ \int_{0}^{x} e^{-\left(a\xi^2 + b/\xi^2\right)} \xi^{-2} d\xi = \int_{1/x}^{\infty} e^{-\left(b\phi^2 + a/\phi^2\right)} d\phi \quad \text{iff} \quad \phi = \xi^{-1} . \]  

(5.40)
Applying (5.36) and let $x = y^{-1}$, the RHS of (5.40) can be calculated using

$$
\int_y^\infty e^{-(b\phi^2 + a/\phi^2)} \, d\phi = \int_0^\infty e^{-(b\phi^2 + a/\phi^2)} \, d\phi - \int_0^y e^{-(b\phi^2 + a/\phi^2)} \, d\phi
$$

$$
= \frac{e^{-2\sqrt{ab} \sqrt{\pi}}}{2\sqrt{b}} - \frac{e^{-2\sqrt{ab} \sqrt{\pi}}}{4\sqrt{b}} \left[ Q'_1 - e^{4\sqrt{ab} Q'_2} \right]
$$

(5.41)

where $Q'_1 = \text{Erfc} \left( \sqrt{a}/y - y\sqrt{b} \right)$ and $Q'_2 = \text{Erfc} \left( \sqrt{a}/y + y\sqrt{b} \right)$. The RHS of (5.41) is rearranged to give

$$
\int_y^\infty e^{-(b\phi^2 + a/\phi^2)} \, d\phi = \frac{e^{-2\sqrt{ab} \sqrt{\pi}}}{4\sqrt{b}} \left\{ 1 + 1 - Q'_1 + e^{4\sqrt{ab} Q'_2} \right\}.
$$

Now, $1 - Q'_1 = \text{Erf} \left( \sqrt{a}/y - y\sqrt{b} \right) = -\text{Erf} \left( y\sqrt{b} - \sqrt{a}/y \right) = Q^*$ so

$$
\int_y^\infty e^{-(b\phi^2 + a/\phi^2)} \, d\phi = \frac{e^{-2\sqrt{ab} \sqrt{\pi}}}{2\sqrt{b}} \left\{ 1 - Q^* + e^{4\sqrt{ab} Q'_2} \right\}.
$$

Since $1 - Q^* = \text{Erfc} \left( y\sqrt{b} - \sqrt{a}/y \right)$ and $y = x^{-1}$ and using (5.40) results in

$$
\int_0^x e^{-(a\xi^2 + b/\xi^2)} \xi^{-2} \, d\xi = \frac{e^{-2\sqrt{ab} \sqrt{\pi}}}{4\sqrt{b}} \left[ \hat{Q}_1 + e^{4\sqrt{ab} \hat{Q}_2} \right]
$$

as required. \qed

\textbf{Collorary 4.}

$$
\int_x^\infty e^{-(a\xi^2 + b/\xi^2)} \xi^{-2} \, d\xi = \frac{e^{-2\sqrt{ab} \sqrt{\pi}}}{4\sqrt{b}} \left[ 2 - \hat{Q}_1 - e^{4\sqrt{ab} \hat{Q}_2} \right].
$$

(5.42)

\textbf{Proof.} The sum of the RHS of (5.39) and the RHS of (5.42) equals the RHS of (5.38) as required by the sum of the corresponding LHS of these expressions. \qed
5.4 Analytical evaluation of the incomplete Bessel function defined by (5.2) when \( \lambda = \pm \frac{1}{2} \)

Since the Bessel function (5.26) and the Laplace’s integral are related, this section utilizes the results in Section 5.3.1 to calculate the incomplete Bessel function when \( \lambda = \pm \frac{1}{2} \).

5.4.1 Formulae for the lower incomplete Bessel function

When \( \lambda = \pm \frac{1}{2} \), \( a = z^2 \) and \( b = \frac{1}{4} \) are substituted into (5.2) then the lower incomplete function of the form given by (5.27) and (5.39) are obtained. Thus, the former and the latter can be used to evaluate the lower incomplete Bessel function when \( \lambda = -\frac{1}{2} \) and \( \frac{1}{2} \) respectively. Here, \( \tilde{Q}_1 \) and \( \tilde{Q}_2 \) are numeric values of \( \hat{Q}_1 \) and \( \hat{Q}_2 \) respectively when \( a = z^2 \) and \( b = \frac{1}{4} \).

\[
\hat{K}_{\pm 1/2}(z, x) = \frac{e^{-z}}{2} \sqrt{\frac{\pi}{2z}} \left[ \tilde{Q}_1 \pm e^{2z} \tilde{Q}_2 \right] \tag{5.43}
\]

where \( \tilde{Q}_1 = \text{Erfc}(1/(2x) - zx) \) and \( \tilde{Q}_2 = \text{Erfc}(1/(2x) + zx) \).

5.4.2 Formulae for the upper incomplete Bessel function

The upper incomplete Bessel function corresponding to (5.2) is given by

\[
\hat{K}^U_\lambda(z, x) = \frac{1}{(2z)^\lambda} \int_x^\infty e^{-\left\{z^2\xi^2 + 1/(4\xi^2)\right\}} \xi^{-2\lambda - 1} d\xi, \quad x > 0. \tag{5.44}
\]

When \( \lambda = \pm \frac{1}{2} \), \( a = z^2 \) and \( b = \frac{1}{4} \) are substituted into (5.44), the upper incomplete Bessel functions of the forms given by (5.37) and (5.42) are obtainable. Thus the former and the latter are used to evaluate the upper incomplete Bessel function when \( \lambda = -\frac{1}{2} \) and \( \frac{1}{2} \) respectively. Here, \( \tilde{Q}_1 \) and \( \tilde{Q}_2 \) are numeric values of \( \hat{Q}_1 \) and \( \hat{Q}_2 \).
respectively when \( a = z^2 \) and \( b = \frac{1}{4} \).

\[
\hat{K}^U_{\pm 1/2}(z, x) = \frac{e^{-\frac{z}{2}}}{\sqrt{\pi}} \left( 1 + \overline{Q}_1 \pm e^{2z} \tilde{Q}_2 \right).
\]  
(5.45)

Note that the term \((1 - \tilde{Q}_1)\) is replaced by \( \overline{Q}_1 = \text{Erf}(1/(2x) - zx) \) using

\[
1 - \text{Erfc} \left( \frac{1}{2x} - zx \right) = \text{Erf} \left( \frac{1}{2x} - zx \right)
\]

to enhance computational accuracy and stability. Formulae (5.43) and (5.45) were implemented as the \texttt{R} function \texttt{halfOrderAIBF()} in Appendix D.4.

5.5 Numerical approach to approximate incomplete Bessel function given by (5.3)

Computationally, the advantage of obtaining analytical expressions for the incomplete Bessel functions is the ability to achieve the highest level of accuracy because elementary and error functions have been well-known for being implemented and tested to the most accurate level in major software. Secondly, the speed of calculation is also fastest because there is no integration nor iteration. Moreover, obtaining analytic results is significant mathematically since the results retain their validity for the complex cases, due to the principle of analytic continuation. Statistically, expressing of the incomplete Bessel function in terms of elementary and error function allows for analytic descriptions of the cumulative distribution function and tail probabilities of the GIG. However, analytic expression restricts the applicability of the incomplete Bessel function in practice. This is because such expressions can only be obtained for certain value of \( \lambda \). For these reasons, the next section is devoted to implementing a numerical routine detailed in Slevinsky and Safouhi (2010) to approximate the incomplete Bessel function (5.3) for any \( \lambda \in \mathbb{R} \).
5.5.1 Description of the algorithm

Theoretically, the core of the algorithm is the evaluation of oscillatory integrals by using the $D^{(g)}_n$ transformation [Levin and Sidi (1981)] and its confluent form the $G^{(g)}_n$ transformation in [Gray and Wang (1992)]. In this section, the positive integer $g$ stands for the order of the linear differential equation satisfied by the integrand and $n$ is the order of the transformation. As $n$ becomes large the approximation of the integral approaches its exact value since incomplete Bessel functions satisfy a first-order linear homogeneous differential equation of order $g = 1$. [Slevinsky and Safouhi (2009) and Slevinsky and Safouhi (2010)] successfully derived an analytic expression of the $G^{(1)}_n$, which can be used to compute the incomplete Bessel function given by

$$K_\lambda(\chi, \psi) = \chi^\lambda \int_\chi^\infty e^{-\xi - \chi \psi/\xi} \xi^{-\lambda-1} d\xi. \quad (5.46)$$

It was also established on p.2 of [Slevinsky and Safouhi (2010)] that

$$\overline{K}_\lambda(\chi, \psi) = \int_1^\infty e^{-(\chi \xi + \psi/\xi)} \xi^{-\lambda-1} d\xi = \overline{K}_\lambda(\chi, \psi). \quad (5.47)$$

The incomplete Bessel function (5.46) can be calculated, using the relation

$$\tilde{G}^{(1)}_n(\lambda, \chi, \psi) = \frac{\tilde{N}^{(1)}_n(\lambda, \chi, \psi)}{\tilde{D}^{(1)}_n(\lambda, \chi, \psi)} \quad (5.48)$$

and $\overline{K}_\lambda(\chi, \psi) = \chi^\lambda \tilde{G}^{(1)}_n(\lambda, \chi, \psi)$.

An R function called `numeIBF()` was implemented to recursively calculate the quantities $\tilde{D}^{(1)}_n(\lambda, \chi, \psi)$ and $\tilde{N}^{(1)}_n(\lambda, \chi, \psi)$ by coding the following formulae:

$$\tilde{D}^{(1)}_n(\lambda, \chi, \psi) = (-\chi \psi)^n \chi^{\nu+1} e^{\chi + \psi} \sum_{r=0}^n \binom{n}{r} (-\psi)^{-r} \sum_{i=0}^r A_i^r \chi^i$$

131
\[ \widetilde{N}_n^{(1)}(\lambda, \chi, \psi) = \frac{e^{-\chi - \psi}}{\chi^{\lambda} \psi} \sum_{r=1}^{n} \left( \frac{n}{r} \right) \tilde{D}_{n-r}^{(1)}(\lambda, \chi, \psi) (\chi \psi)^r \sum_{s=0}^{r-1} \left( \frac{r-1}{s} \right) \psi^{-s} \sum_{i=0}^{s} A_i^s(-\chi)^i. \]

Here,

\[ A_i^l = \begin{cases} 1, & \text{for } i = 1, \\ (n - \lambda - (l-1)(\mu + 1)) A_{l-1}^0, & \text{for } i = 0, \ l > 0, \\ (n - \lambda + i(g + 1) - (l-1)(\mu + 1)) A_{l-1}^i + A_{l-1}^{i-1}, & \text{for } 0 < i < l. \end{cases} \]

The function approximates integral (5.46), and the result approaches its true value as \( n \) increases. It stops when

\[ \text{Machine\$double.eps} \geq \left| \widetilde{G}_n^{(1)}(\lambda, \chi, \psi) - \widetilde{G}_{n-1}^{(1)}(\lambda, \chi, \psi) \right|, \quad (5.49) \]

where \( 0 < \text{eps} < 1 \) is set by the user. For the values of \( \chi \geq \psi \), (5.48) is used to calculate (5.46) straightforwardly. But for \( \chi < \psi \), the routine is applied to calculate \( \overline{K}_{-\lambda}(\psi, \chi) \) first, then \( \overline{K}_{\lambda}(\chi, \psi) \) is evaluated using the well-known relation

\[ \overline{K}_{\lambda}(\chi, \psi) = 2 \left( \frac{\chi}{\psi} \right)^\lambda \overline{K}_{\lambda} \left( 2\sqrt{\chi \psi} \right) - \overline{K}_{-\lambda}(\psi, \chi) \]

where \( \overline{K}_{\lambda} \left( 2\sqrt{\chi \psi} \right) \) can be readily obtainable using the \( \text{R} \) function \texttt{besselK()} . Details of the function \texttt{numeIBF()} can be found in Appendix D.6.
5.6 Relationships between the incomplete Bessel function defined by (5.2), (5.3) and the generalized inverse Gaussian distribution

This section establishes the mathematical relationship between the integral representations in (5.2) and (5.3) with the density function of the GIG given by

\[
\text{GIG}(w|\lambda, \psi, \chi) = \frac{(\chi/\psi)^{\lambda/2}}{2K_\lambda(\sqrt{\psi\chi})} w^{\lambda-1} e^{-\{(1/2)(\psi/w + \chi w)\}}, \quad w > 0,
\]

(5.50)

5.6.1 Relationships between (5.2) and the density function of the generalized inverse Gaussian distribution (5.50)

Replacing \(\xi^2\) in (5.2) by \((2z\xi)^{-1}\) yields

\[
\widehat{K}_\lambda(z, x) = \frac{1}{2} \int_0^\infty e^{-z/2(\xi + 1/\xi)} \xi^{\lambda-1} d\xi
\]

(5.51)

where \(u = (2zx^2)^{-1}\). Substituting \(z = \sqrt{\chi/\psi}\) into (5.51) and letting \(\xi = w\sqrt{\chi/\psi}\) gives

\[
\frac{1}{2} \int_u^\infty e^{-z/2(\xi + 1/\xi)} \xi^{\lambda-1} d\xi = \frac{1}{2} \left(\frac{\chi}{\psi}\right)^{\lambda/2} \int_v^\infty e^{-1/2(\psi/w + \chi w)} w^{\lambda-1} dw
\]

(5.52)

where

\[
v = u \sqrt{\psi/\chi} = \frac{\sqrt{\psi/\chi}}{2\sqrt{\chi/\psi}x^2} = \frac{1}{2} \frac{x^2}{\chi}.\]

(5.53)

The RHS of (5.52) is the form of the incomplete Bessel function presences in the density function (5.50). If it is denoted as

\[
^*K_\lambda^U(v, \psi, \chi) = \frac{1}{2} \left(\frac{\chi}{\psi}\right)^{\lambda/2} \int_v^\infty e^{-1/2(\psi/w + \chi w)} w^{\lambda-1} dw
\]

(5.54)
then
\[
\frac{1}{(2z)^\lambda} \int_0^x e^{-\left\{z^2\xi^2 + 1/(4\xi^2) \right\}} \xi^{-2\lambda-1} \, d\xi = \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} \int_0^\infty e^{-1/2(\psi/w + \chi w)} w^{\lambda-1} \, dw
\]
(5.55)
or
\[
\hat{K}_\lambda(z, x) = \star K_{\lambda}^U(v, \psi, \chi).
\]
(5.56)

It is obvious from (5.53) that \( v \to 0 \) when \( x \to \infty \). This gives
\[
\frac{1}{(2z)^\lambda} \int_0^\infty e^{-\left\{z^2\xi^2 + 1/(4\xi^2) \right\}} \xi^{-2\lambda-1} \, d\xi = \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} \int_0^\infty e^{-1/2(\psi/w + \chi w)} w^{\lambda-1} \, dw,
\]
(5.57)
which holds \( \forall \lambda \in \mathbb{R} \) and agrees with the derivation on p.51 of Gray and Mathews (1931). By using (5.55) and (5.57) the following relations can be deduced
\[
\frac{1}{(2z)^\lambda} \int_0^\infty e^{-\left\{z^2\xi^2 + 1/(4\xi^2) \right\}} \xi^{-2\lambda-1} \, d\xi = \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} \int_0^\infty e^{-1/2(\psi/w + \chi w)} w^{\lambda-1} \, dw,
\]
(5.58)
\[
\hat{K}_\lambda^U(z, x) = \star K_\lambda(v, \psi, \chi).
\]
(5.59)

Explicit formulae to evaluate either side of Equations (5.55) and (5.58) cannot be located even after a reference such as Olver et al. (2010) was consulted. Relations (5.56) and (5.59) are important because they imply that the formulae given by (5.43) and (5.45), which are the evaluation of the LHS of these integrals for \( \lambda = \pm \frac{1}{2} \), can be used to obtain the formulae to calculate tail probabilities for the GIG given by (5.50).

For \( \lambda = \pm \frac{1}{2} \), the evaluation for (5.55) and (5.58), which are given by (5.43) and (5.45) were coded as the R function \texttt{halfOrderAIBF()} in Appendix D.4. The evaluation of the integral on the RHS of (5.55) and (5.58) can be numerically obtained using the Maple 15 procedure called \texttt{halfOrderAIBFproc()} in Appendix D.5.
5.6.2 Relationships between (5.3) and the density function of the generalized inverse Gaussian distribution (5.50)

Equation (5.54) was used to derive this relationship. Replacing \( w \) by \( w = \phi v \) where \( v \) is given by (5.53) to obtain

\[
*K^U_\lambda(v, \psi, \chi) = \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} v^\lambda \int_1^\infty e^{-\left(1/2\right)|\psi/(\phi v) + \chi \phi v|} \phi^{\lambda-1} d\phi
\]

\[
= \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} v^\lambda \int_1^\infty e^{-\left(\bar{\psi}/\phi + \bar{\chi}\phi\right)} \phi^{\lambda-1} d\phi, \quad (5.60)
\]

where

\[
\bar{\chi} = \frac{\chi v}{2} \quad \text{and} \quad \bar{\psi} = \frac{\psi}{2v}.
\]

Equation (5.60) can also be rewritten as

\[
*K^U_\lambda(v, \psi, \chi) = \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} v^\lambda \tilde{K}_{-\lambda} \left( \bar{\chi}, \bar{\psi} \right) \quad (5.61)
\]

where

\[
\tilde{K}_{-\lambda} \left( \bar{\chi}, \bar{\psi} \right) = \int_1^\infty e^{-\left(\bar{\chi}\phi + \bar{\psi}/\phi\right)} \phi^{\lambda-1} d\phi. \quad (5.62)
\]

The RHS of (5.62) is evaluated numerically using (5.47) because the evaluation of the LHS of (5.47) is obtainable by the function \texttt{numeIBF( )}. It is noted that the incomplete Bessel function in (5.62) is defined as \( \tilde{K}_{-\lambda}( ) \) in this chapter because the sign of \( \lambda \) in the integrand is opposite with the sign of \( \lambda \) appears in (5.3). However, Terras (1981) explicitly identified (5.62) as \( \tilde{K}_{\lambda}( ) \) by applying an incomplete Bessel function of a type characterized in Agrest and Maximov (1971). Equation (5.61) establishes the mathematical relationship between (5.3) and the distribution function of the GIG. This relation will be used in the next chapter to approximate the tail probabilities of the GIG using the \texttt{R} function \texttt{numeIBF( )}. 

135
5.7 Numerical studies

This section examines the computational accuracy and stability the derived methods to calculate three different types of the incomplete Bessel functions. The following table summarizes major characteristics of these methods.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>Definitions</th>
<th>Methods</th>
<th>Function names</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j + \frac{1}{2}$</td>
<td>(5.11), (5.12)</td>
<td>Analytical</td>
<td>closedFormIBF ()</td>
</tr>
<tr>
<td>$\pm \frac{1}{2}$</td>
<td>(5.43), (5.45)</td>
<td>Analytical</td>
<td>halfOrderAIBF ()</td>
</tr>
<tr>
<td>$\in \mathbb{R}$</td>
<td>(5.3)</td>
<td>Numerical</td>
<td>numeIBF ()</td>
</tr>
</tbody>
</table>

5.7.1 Accuracy of the analytical methods

The accuracy of the functions closedFormIBF () and halfOrderAIBF () is examined by comparing the results returned by these functions against those obtained by numerically integrating the involved integrals using Maple and Mathematica. The evaluation results from these commercial symbolic packages are considered as “exact” values.

Performance of the function closedFormIBF ()

The incomplete Bessel functions defined by (5.11) and (5.12) were coded as the R function closedFormIBF (). The accuracy of this function is now compared against the values obtained by numerically integrating the integral (5.1) using the Maple procedure closedFormIBFpro () and Mathematica.

As can be seen from Table 5.1, the evaluation results by closedFormIBF () agree with the “exact” values. When $\lambda = \frac{151}{2}$, $z = 390$ and $x = 3$, the Maple procedure returned 0 but evaluation by closedFormIBF () agree with the evaluation result by Mathematica.

Given the values of $x$, $z$, lambda in Table 5.1. For the evaluation of the lower incomplete Bessel function (5.11), the values in columns closedFormIBF () and closedFormIBFpro () can be reproduced using:
closedFormIBF(x, z, lambda/2, lowerBK = TRUE, digits = 13,
   expon.scaled = FALSE)

and

closedFormIBFproc(x, z, lambda/2, TRUE, 50, FALSE) # Maple code

For the evaluation of the upper incomplete Bessel function (5.12), the values in these columns can be reproduced using:

closedFormIBF(x, z, lambda/2, lowerBK = FALSE,
   expon.scaled = FALSE)

and

closedFormIBFproc(x, z, lambda/2, FALSE, 50, FALSE) # Maple code
Table 5.1: Analytical evaluation incomplete Bessel function (5.1) versus numerical integration using Maple 15 and Mathematica.

Values of $\alpha$ are to be divided by 2. Underlined numbers indicate the decimal places where the numbers agree.

```
<table>
<thead>
<tr>
<th>$x$</th>
<th>$z$</th>
<th>Formula</th>
<th>closedFormIBF</th>
<th>closedFormIBFpro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>$10^{-1}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(5.12) Upper

<table>
<thead>
<tr>
<th>$x$</th>
<th>$z$</th>
<th>Formula</th>
<th>closedFormIBF</th>
<th>closedFormIBFpro</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>$10^{-1}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

(5.11) Lower
Performance of the function halfOrderAIBF()

Using (5.54) and (5.56), the accuracy of the lower incomplete Bessel function on the LHS of the latter can be checked by numerically integrating the RHS of the former, see Appendix D.5. For $\lambda = -\frac{1}{2}$, Table 5.2 was obtained by using the following R and Maple 15 code.

```r
# The LHS of (5.56) is a lower incomplete BK
halfOrderAIBF(v, chi, psi, lambda = -1/2, lowerBK = TRUE, digits = 20)

# The RHS of (5.54) is an upper incomplete BK
halfOrderAIBFproc(v, chi, psi, -1/2, FALSE, 60) # Maple code
```

respectively. The values of $v$, $\chi$, $\psi$ are given in the table. Note that the latter took longer time to compute than the former. Reducing the number of `Digits:= 40` can improve calculation speed but this may result in inaccurate results because the floating point accuracy number of Maple 15 depends on `Digits:= n` where $n$ is a positive number. The maximum value of $n$ is obtained from `kernelopts(maxdigits)`. For more information, see `?Digits`.

<table>
<thead>
<tr>
<th>$v$</th>
<th>$\chi$</th>
<th>$\psi$</th>
<th>halfOrderAIBF()</th>
<th>halfOrderAIBFproc()</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>5</td>
<td>2.643 072 497 311 $\times 10^{-09}$</td>
<td>2.643 072 497 311 $\times 10^{-09}$</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>5</td>
<td>2.624 754 465 492 $\times 10^{-09}$</td>
<td>2.624 754 465 494 $\times 10^{-09}$</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1</td>
<td>2.135 176 165 224 $\times 10^{-09}$</td>
<td>2.135 176 165 224 $\times 10^{-09}$</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1.790 504 714 861 $\times 10^{-09}$</td>
<td>1.790 504 714 862 $\times 10^{-09}$</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>5</td>
<td>6.523 846 319 630 $\times 10^{-14}$</td>
<td>6.523 846 319 650 $\times 10^{-14}$</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>5</td>
<td>5.176 154 502 457 $\times 10^{-13}$</td>
<td>5.176 154 502 465 $\times 10^{-13}$</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>1</td>
<td>5.291 030 829 948 $\times 10^{-14}$</td>
<td>5.291 030 829 948 $\times 10^{-14}$</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>100</td>
<td>1.767 943 122 443 $\times 10^{-05}$</td>
<td>1.767 943 122 443 $\times 10^{-05}$</td>
</tr>
</tbody>
</table>

Table 5.2: Accuracy comparison between analytic lower incomplete Bessel function evaluated in R and numerical integration using Maple 15. Underlined numbers indicate the decimal places where the numbers agree.
Table 5.2 indicates that the analytic expression was correctly derived, and that elementary and error functions are implemented to the highest level of accuracy in major software such as R.

The following code is used to check equation (5.56) when \( \lambda = \frac{1}{2} \).

\[
\text{halfOrderAIBF}(v, \chi, \psi, \lambda = 1/2, \text{lowerBK} = \text{FALSE})
\]

\[
\text{halfOrderAIBFproc}(v, \chi, \psi, 1/2, \text{TRUE}, 40) \quad \# \text{Maple code}
\]

### 5.7.2 Accuracy of the numerical method

Using the tested parameters specified by Slevensky and Safouhi

In order to assess the accuracy of the function \( \text{numeIBF}() \), its evaluation results were compared against the “correct evaluation” of the incomplete Bessel function (5.3) presented in Harris (2008). This was the approach used by Slevensky and Safouhi to establish the accuracy of their algorithm (cf. p.1416 of Slevinsky and Safouhi, 2010).

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>( \text{numeIBF}() )</th>
<th>Correct values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.225 310 761 266 483</td>
<td>2.225 310 761 266 469</td>
</tr>
<tr>
<td>1</td>
<td>0.213 894 166 822 953</td>
<td>0.213 894 166 822 940</td>
</tr>
<tr>
<td>2</td>
<td>0.054 503 469 799 706</td>
<td>0.054 503 469 799 701</td>
</tr>
<tr>
<td>3</td>
<td>0.023 253 121 507 712</td>
<td>0.023 253 121 507 708</td>
</tr>
<tr>
<td>4</td>
<td>0.013 042 750 996 079</td>
<td>0.013 042 750 996 080</td>
</tr>
<tr>
<td>5</td>
<td>0.008 567 534 990 648</td>
<td>0.008 567 534 990 649</td>
</tr>
<tr>
<td>6</td>
<td>0.006 208 676 806 600</td>
<td>0.006 208 676 806 601</td>
</tr>
<tr>
<td>7</td>
<td>0.004 801 085 238 177</td>
<td>0.004 801 085 238 177</td>
</tr>
</tbody>
</table>

Table 5.3: Underlined numbers indicate the decimal places where the numbers in two columns agree.

The numbers in the column heading \( \text{numeIBF}() \) can be reproduced using

\[
\text{lambdaRanges} \leftarrow 0:7
\]

\[
\text{sapply}(\text{lambdaRanges}, \text{function}(\text{lambda}.)
\]

140
As can be seen from Table 5.3, \texttt{numeIBF()} gave evaluation results which agree with the accurate values up to 13 decimal places. These high level of accuracy was also further checked using the following \texttt{Maple} code

\begin{verbatim}
for lambda from 0 to 7 do
    numeIBFproc(0.01, 4, lambda, 30);
end do;  # Maple code
\end{verbatim}

the procedure \texttt{numeIBFproc()} is presented in Appendix D.7.

From the above results, it can be concluded that the algorithm derived by Slevensky and Safouhi was correctly implemented as the function \texttt{numeIBF()} and that the function works correctly when it was tested using the parameters specified by the authors.

### 5.7.3 Difficulties with the numerical algorithm

This section is to test \texttt{numeIBF()} using a wider range of parameters $\chi$, $\psi$ and $\lambda$ then the set of parameters used in Tables 1–2 of \textcite{Slevinsky and Safouhi 2010}. In these numerical studies, the evaluation results by the \texttt{Maple} function \texttt{numeIBFproc()} with the accuracy level of 60 decimal places were considered as “exact evaluation” of the incomplete Bessel function.

The algorithm was based on 3 cases of parameter values $\chi \geq \psi$ and $\chi < \psi$. The following tests are to illustrate that the algorithm encountered numerical problems in all of these cases.

**When $\psi > \chi$**

As can be seen from Table 5.4, the evaluation results by \texttt{numeIBF()} agree with the correct values for only 4 decimal places.
Table 5.4: Underlined numbers indicate the decimal places where the results given by \( \text{numeIBF}(\cdot) \) and \( \text{numeIBFproc}(\cdot) \) (correct values) agree. Symbols \(-\) indicate repeated values.

<table>
<thead>
<tr>
<th>( \psi )</th>
<th>( \chi )</th>
<th>( \lambda )</th>
<th>( \text{numeIBF}() )</th>
<th>( \text{numeIBFproc}() )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>5</td>
<td>4</td>
<td>1.350 806 070 514 207 ( \times 10^{-10} )</td>
<td>1.350 827 068 786 651 ( \times 10^{-10} )</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>1.272 916 270 428 235 ( \times 10^{-10} )</td>
<td>1.272 929 461 289 356 ( \times 10^{-10} )</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>1.203 094 567 009 222 ( \times 10^{-10} )</td>
<td>1.203 103 422 772 194 ( \times 10^{-10} )</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>1.140 198 881 819 637 ( \times 10^{-10} )</td>
<td>1.140 205 246 317 588 ( \times 10^{-10} )</td>
</tr>
</tbody>
</table>

# R code for testing \( \chi = 15 > \psi = 5 \)

```r
lambdaRanges <- 4:7
sapply(lambdaRanges, function(lambda.
    numeIBF(15, 5, lambda., traceIBF = FALSE, eps = 0.85, digits = 16))
```

# Maple code to obtain correct evaluation when \( \chi = 15 > \psi = 5 \)

```maple
for lambda from 4 to 7 do
    numeIBFproc(15, 5, lambda, 60);
end do;  # End Maple code
```

When \( \chi = \psi \)

Table 5.5 shows that the evaluation results by \( \text{numeIBF}(\cdot) \) agree with the correct values for only 3 decimal places.

<table>
<thead>
<tr>
<th>( \psi )</th>
<th>( \chi )</th>
<th>( \lambda )</th>
<th>( \text{numeIBF}() )</th>
<th>( \text{numeIBFproc}() )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>1/2</td>
<td>1.277 118 907 903 529</td>
<td>1.277 434 686 070 312</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>-</td>
<td>NaN</td>
<td>1.659 006 286 883 282</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>-1/2</td>
<td>3.564 664 391 694 504</td>
<td>3.564 732 987 553 647</td>
</tr>
<tr>
<td>0.03</td>
<td>0.03</td>
<td>-</td>
<td>8.181 662 432 105 290</td>
<td>8.209 993 602 832 297</td>
</tr>
</tbody>
</table>

Table 5.5: Underlined numbers indicate the decimal places where the results given by \( \text{numeIBF}(\cdot) \) and \( \text{numeIBFproc}(\cdot) \) (correct values) agree.

# R code for testing \( \chi = \psi = 0.05 \) and \( \lambda = 1/2 \)

```r
numeIBF(0.05, 0.05, 1/2, traceIBF = TRUE, eps = 0.9, digits = 22)
```

142
When $\chi > \psi$

As can be seen from Table 5.6, none of the results by `numeIBF()` agree with the correct values.

<table>
<thead>
<tr>
<th>$\psi$</th>
<th>$\chi$</th>
<th>$\lambda$</th>
<th><code>numeIBF()</code></th>
<th><code>numeIBFproc()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>12</td>
<td>25</td>
<td>6.715 217 252 763 842 × 10^{-06}</td>
<td>1.167 723 311 994 233 × 10^{-11}</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>30</td>
<td>7.684 776 594 630 232 × 10^{-04}</td>
<td>9.704 045 486 398 849 × 10^{-12}</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>35</td>
<td>1.729 712 363 967 051 × 10^{-01}</td>
<td>8.291 869 369 341 973 × 10^{-12}</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>40</td>
<td>7.160 614 401 757 894 × 10^{+01}</td>
<td>7.233 795 108 572 345 × 10^{-12}</td>
</tr>
</tbody>
</table>

Table 5.6: None of the results given by `numeIBF()` match the “exact evaluation”. Symbols – indicate repeated values.

When $\lambda$ assumes large values

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th><code>numeIBF()</code></th>
<th><code>numeIBFproc()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>143</td>
<td>7.233 795 108 572 345 × 10^{-12}</td>
</tr>
</tbody>
</table>
This \texttt{R} output indicates that the maximum value of the incomplete Bessel function is $1.0786 \times 10^{+39}$. But the following \texttt{Maple 15} code gives $5.74 \times 10^{-8}$ as the maximum correct value.

\begin{verbatim}
for lambda from 50 to 90 by 5 do
  numeIBFproc(1, 12, lambda, 60);
end do; # Maple code
\end{verbatim}

From the above numerical tests, it can be concluded that there are computational problems with the numerical approach. It is important to note that \texttt{numeIBF()} did not give any warning message accepts the case where NaNs were produced. This means that the routine converged within the limit of the numbers of the $G$ transformation allowed.

\section*{Discussion}

Relating to the inaccuracies of \texttt{numeIBF()} when $\chi \geq \psi$, the algorithm encountered the numerical limitation which is due to the stopping rule specified by Equation (30) of \cite{slevinsky2010}. This criterion was coded as (5.49) in the routine \texttt{numeIBF()}. Specifically,

\begin{equation}
\text{Machine}\text{\$double\_eps}^{0.98} \geq \left| \tilde{G}^{(1)}_{n}(\lambda, \chi, \psi) - \tilde{G}^{(1)}_{n-1}(\lambda, \chi, \psi) \right| = \varepsilon \quad (5.63)
\end{equation}

was used to produce the values on Table 5.4. Here the limitation appears to be that the routine needs $k$ iterations to give the most accurate evaluation but it stops after $i$ iterations ($i < k \leq 90$) when the stopping criterion was satisfied. This type of stopping rule is typical in numerical approximation using iterative algorithms. It only works well in cases where the value of the approximated function between
two consecutive iterations is greater than the smallest floating point number (e.g.,
the EM algorithm). However, if a stopping rule of the form (5.63) is used when
\( G_n^{(1)}(\lambda, \chi, \psi) \) assume values less than \( \text{Machine\DoubleEps} \) then numerical inaccuracy
occurs. Note that the approximation errors in Tables 1 and 2 of Slevinsky and
Safouhi (2010) are all smaller than \( \text{Machine\DoubleEps}^{0.98} \). This explains why the
routine converged properly and gave accurate results when it was tested by them.

Relating to the inaccuracy presented in Table 5.6, Equations (18)–(20) of Slevin-
sky and Safouhi (2010) show that the numerator and denominator of the ratio on
the RHS of (5.48) assume positive values for all \( n \) (\( n \in [1, 90] \)). However, for the
value of \( \chi, \psi \) and \( \lambda \) given in Table 5.6 the algorithm gave inaccurate results because
the values of either \( \tilde{N}_n() < 0 \) or \( \tilde{D}_n() < 0 \). The evaluation given by \text{numeIBF()} \) in
the first row can be obtained using

\[
\text{numeIBF}(10, 12, 25, \text{traceIBF} = \text{TRUE}, \text{eps} = 0.98, \text{digits} = 22)
\]

**Suggested diagnostic tool**

While the task of finding solutions to the above issues is not attempted here, this
section proposes a diagnostic tool to detect when these problems occur by providing
the argument \text{traceIBF} with the function \text{numeIBF()} \). When it is \text{TRUE} the value of
the numerator and denominator of (5.48) and \( \epsilon \) for each iteration \( n \) is printed. The
following gives details on how to use this argument

- If none of values of the numerator and denominator of \( G_n \) in (5.48) are nega-
tive and

  1. the routine converges when the value of \( \epsilon \) given at \( n = 2 \) (i.e., the second
     iteration) is not less than the LHS of (5.63) then the evaluation is correct.

     All cases which were presented on Table 1 and 2 of Slevinsky and Safouhi
     (2010) satisfy these conditions.

  2. the routine stops at \( n = 2 \) (i.e., stops after 2 iterations) when \( \epsilon \) is smaller
than the LHS of (5.63) then it is possible that the routine has stopped prematurely. All tested values on Table 5.4 satisfied these conditions.

- If either or both of the values of the numerator or denominator of $G_n$ is negative then the evaluation is incorrect.

### 5.7.4 Numerical versus analytical incomplete Bessel functions

Numerical accuracy of halfOrderAIBF() was established in Section 5.7.1 using the Maple function in Appendix D.3. The purpose of this section is to show cases where halfOrderAIBF() gave accurate results but the numerical method (i.e., function numeIBF()) was inaccurate.

Firstly, the mathematical relationship between the tested incomplete Bessel functions was established using (5.56) and (5.61) to obtain

$$
\hat{K}_\lambda(z,x) = \frac{1}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} e^{\lambda \psi} K_{-\lambda} \left( \tilde{\chi}, \tilde{\psi} \right).
$$

Equation (5.64) is numerically examined using the function methodComp() in Appendix D.8. It is important to note that the routine numeIBF() was derived using the integral given by (5.3) (cf. p.1412 of Slevinsky and Safouhi [2010]). However, the incomplete Bessel function on the RHS of (5.64) has the form given by (D.2). If the evaluation of these integrals is to be compared then the value of $\lambda$ in one of them needs to be multiplied by $-1$. The reason for the co-existence of both integral representations of this incomplete Bessel function in this research is because (D.2) occurs in the density function of the GIG.

When $\lambda = \frac{1}{2}$

The evaluation values in Column halfOrderAIBF() can be reproduced using the following R code

146
v <- 6
chi <- 30
psi <- seq(10, 16, by = 2)
lambda <- 1/2
for(j in 1:length(psi)) {
    methodsComp(v = v, chi = chi, psi = psi[j], lambda = lambda,
               traceIBF = FALSE, eps = 0.9, lowerBK = TRUE, digits = 12)
}

Because of the disagreement between evaluation results given by the analytical and numerical method in Table 5.7, the following Maple 15 code was then used to obtain “correct” evaluation results

v := 6;
chi := 30;
lambda := 1/2;
for psi from 10 to 16 by 2 do
    halfOrderAIBFproc(v, chi, psi, lambda, FALSE, 90)
end do  # Maple code

The results are not reported here but all numbers in Column halfOrderAIBF () agree with Maple evaluation. The Maple function halfOrderAIBFproc() can be found in Appendix D.3 R code for significantly more extensive test can be found in Appendix D.8.
When $\lambda = -\frac{1}{2}$

<table>
<thead>
<tr>
<th>$v$</th>
<th>$\chi$</th>
<th>$\psi$</th>
<th>halfOrderAIBF()</th>
<th>numeIBF()</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>10</td>
<td>6.497 857 359 063 $\times 10^{-11}$</td>
<td>6.497 847 079 286 $\times 10^{-11}$</td>
</tr>
<tr>
<td>-</td>
<td>12</td>
<td></td>
<td>2.687 948 398 728 $\times 10^{-11}$</td>
<td>2.687 932 201 564 $\times 10^{-11}$</td>
</tr>
<tr>
<td>-</td>
<td>14</td>
<td></td>
<td>1.108 472 580 695 $\times 10^{-11}$</td>
<td>1.108 449 953 822 $\times 10^{-11}$</td>
</tr>
<tr>
<td>-</td>
<td>16</td>
<td></td>
<td>4.567 136 175 803 $\times 10^{-12}$</td>
<td>4.566 846 429 731 $\times 10^{-12}$</td>
</tr>
</tbody>
</table>

Table 5.8: Accuracy comparison between analytic and numeric evaluation of the lower incomplete Bessel function when $\lambda = -1/2$.

The evaluation values in Column halfOrderAIBF() of Table 5.8 can be reproduced using the following \texttt{R} code

\begin{verbatim}
v <- 1
chi <- 30
psi <- seq(10, 16, by = 2)
lambda <- -1/2
for(j in 1:length(psi)) {
    methodsComp(v = v, chi = chi, psi = psi[j], lambda = lambda,
                traceIBF = FALSE, eps = 0.9, lowerBK = TRUE, digits = 20)
}
\end{verbatim}

All values in Column halfOrderAIBF() of Table 5.8 agree with the evaluation using Maple 15 which can be reproduced using the code

\begin{verbatim}
v := 1;
chi := 30;
lambda := -1/2;
for psi from 10 to 16 by 2 do
    halfOrderAIBFproc(v, chi, psi, lambda, FALSE, 90)
end do  # Maple code
\end{verbatim}

The Maple function halfOrderAIBFproc() can be found in Appendix D.3. \texttt{R} code for significantly more extensive test can be found in Appendix D.8.
Numerical discussion

The numerical tests which were conducted to compare the accuracy of the analytical and numerical method confirm that the function `numeIBF()` has numerical issues that make it inaccurate in some circumstances. However, the analytical method is more accurate than the numerical approach. The reason for this is that the former does not have the numerical limitations that the latter has. These limitations were discussed in Section 5.7.3. Note that if the argument `traceIBF` of the function `methodComp()` in the above R code is set to `TRUE` then this diagnostic tool clearly shows that the function `numeIBF()` stopped “prematurely”. In other words, the algorithm was halted because the stopping rule was satisfied before the “most accurate” approximated values of the incomplete Bessel function was reached.

5.8 Summary

This chapter deals with the challenge of computing the incomplete Bessel function using analytical and numerical methods. The analytical method was used to derive formulae to evaluate the incomplete Bessel functions (5.1) and (5.2) when $\lambda = j + \frac{1}{2}$ and $\lambda = \pm \frac{1}{2}$ respectively. The derived expressions were then coded as R routines which were numerically tested extensively using different software packages. The chapter also implements a numerical method to evaluate the incomplete Bessel function for any $\lambda \in \mathbb{R}$. However, extensive tests also showed that the numerical approach was inaccurate in some circumstances because it has numerical problems. This chapter proposes a diagnostic tool to identify what those circumstances are. The chapter also establishes the mathematical relationships between the derived formulae of incomplete Bessel functions and the density function of the GIG. These relationships are crucial for using this chapter’s results to derive analytic cumulative distribution function, and numerically approximate the tail probabilities of the GIG in the next chapter.
Chapter 6

Distribution function of the generalized inverse Gaussian distribution

6.1 Introduction

This chapter concerns the calculation of the tail probabilities of the GIG with the density function given by (5.50). Here, $W$ and $w$ denote the random variable $W$ and the specific value $w$ it obtains.

The CDF and CCDF are given by

$$F(v|\lambda, \psi, \chi) = P(W = w \leq v),$$
$$\overline{F}(v|\lambda, \psi, \chi) = P(W = w > v)$$

respectively.

Both lower tail probabilities (i.e., CDF) and upper tail probabilities (i.e., CCDF) of the GIG in cases where $\chi > 0$ and $\psi > 0$ (i.e., normal case) are obtainable by
integrating the density function \((5.50)\) to have

\[
F(v|\lambda, \psi, \chi) = \frac{C}{2} \left(\frac{\chi}{\psi}\right)^{\lambda/2} \int_0^v e^{-\{(1/2)(\psi/w + \chi w)\}} w^{\lambda-1} dw, \tag{6.1}
\]

\[
\bar{F}(v|\lambda, \psi, \chi) = \frac{C}{2} \left(\frac{\chi}{\psi}\right)^{\lambda/2} \int_v^\infty e^{-\{(1/2)(\psi/w + \chi w)\}} w^{\lambda-1} dw \tag{6.2}
\]

where

\[
C = \left[K_\lambda \left(\sqrt{\chi \psi}\right)\right]^{-1}.
\]

The first objective of this chapter is to obtain analytic CDF and analytic CCDF denoting as \(F^A(v|\lambda, \psi, \chi)\), \(\bar{F}^A(v|\lambda, \psi, \chi)\) when \(\lambda = \pm \frac{1}{2}\) by applying (6.1) and (6.2). The second purpose is to approximate the numerical CDF and numerical CCDF for any values of the order \(\lambda \in \mathbb{R}\). This is achieved by using the R function `numeIBF()` implemented in Chapter 5 to evaluate the integrand in (6.2). The notations for numerical CDF and CCDF are \(F^N(v|\lambda, \psi, \chi)\) and \(\bar{F}^N(v|\lambda, \psi, \chi)\) respectively.

Some standard references for the GIG distribution include Barndorff-Nielsen et al. (1978), Barndorff-Nielsen (1978), Halgreen (1979) and Jørgensen (1982). These works, among others, are the basis for strong research interest in the distribution. However, effort on deriving analytic CDF of the distribution seems to have been overlooked. There is only one special case of the GIG where analytical CDF has been obtained and it is the inverse Gaussian (IG) or the Wald distribution \((\lambda = -\frac{1}{2})\). Explicit formula for CDF the IG in terms of error function was obtained in Zigan-girov (1962), Seshadri and Shuster (1968) and Chhikara and Folks (1974) by applying different methods. However, none of these approaches involves the evaluation of the incomplete Bessel function which is the method to be presented here. Note that the tail probabilities obtained by evaluating a derived formula was called exact probabilities on p.1 of Seshadri and Shuster (1968).

This chapter has four sections. Section 6.2 derives analytical CDF and CCDF for two special cases of the GIG. Section 6.3 proposes a numerical method to ap-
proximate tail probabilities of the GIG by numerical approach. Section 6.4 conducts numerical studies to evaluate the accuracy of both numerical and analytical results.

6.2 Analytical distribution functions for the subclasses of the generalized inverse Gaussian

6.2.1 Cumulative distribution function

This section utilizes the results of Sections 5.4.2 and 5.6.1 to derive explicit formulae for the CDF of the special cases of the GIG when \( \lambda = \pm \frac{1}{2} \). Equation (5.59) gives

\[
F^A \left( v \left| -\frac{1}{2}, \psi, \chi \right. \right) = C \hat{K}^-_{-1/2}(z, x),
\]

\[
= C \frac{e^{-z}}{2} \sqrt{\frac{\pi}{2z}} \left( 1 + \bar{Q}_1 + e^{2z} \tilde{Q}_2 \right),
\]

and

\[
F^A \left( v \left| \frac{1}{2}, \psi, \chi \right. \right) = C \hat{K}^+_{1/2}(z, x)
\]

\[
= C \frac{e^{-z}}{2} \sqrt{\frac{\pi}{2z}} \left( 1 + \bar{Q}_1 - e^{2z} \tilde{Q}_2 \right)
\]

where \( z = \sqrt{\chi \psi}, x = \left( \frac{1}{2} v \chi \right)^{-1/2}, \bar{Q}_1 = \text{Erf}(1/(2x) - zx), \text{and} \tilde{Q}_2 = \text{Erfc}(1/(2x) + zx) \).

6.2.2 Complementary cumulative distribution function

This section utilizes the result of Sections 5.4.1 and 5.6.1 to obtain explicit formulae for the CCDF when \( \lambda = \pm \frac{1}{2} \). Equation (5.56) gives

\[
\bar{F} \left( v \left| -\frac{1}{2}, \psi, \chi \right. \right) = C \hat{K}^-_{-1/2}(z, x)
\]

\[
= C \frac{e^{-z}}{2} \sqrt{\frac{\pi}{2z}} \left[ \bar{Q}_1 - e^{2z} \tilde{Q}_2 \right],
\]
and
\[
F \left( v \mid \frac{1}{2}, \psi, \chi \right) = C \hat{K}_{1/2}(z, x)
\]
= \[ C \frac{e^{-z}}{2} \sqrt{\frac{\pi}{2z}} \left[ \tilde{Q}_1 + e^{2z} \tilde{Q}_2 \right] \]

\[
z = \sqrt{\chi \psi}, \quad x = (\frac{1}{2} v \chi)^{-1/2}, \quad \tilde{Q}_1 = \text{Erfc}(1/(2x) - zx), \text{ and } \tilde{Q}_2 = \text{Erfc}(1/(2x) + zx).
\]

### 6.3 Numerical cumulative and complement cumulative distribution function of the generalized inverse Gaussian

Using the relations established in Section 5.6.2 and Equation (5.61), the numerical CCDF of the GIG can be written as

\[
F^N \left( v \mid \lambda, \psi, \chi \right) = \frac{C}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} v^\lambda \tilde{K}_{-\lambda} \left( \tilde{\psi}, \tilde{\chi} \right)
\]

where

\[
\tilde{K}_{-\lambda} \left( \tilde{\psi}, \tilde{\chi} \right) = \int_1^\infty e^{-\left( \tilde{\psi}/\phi + \tilde{\chi}/\phi \right)} \phi^{\lambda - 1} d\phi
\]

and

\[
\tilde{\chi} = \frac{\chi v}{2} \quad \text{and} \quad \tilde{\psi} = \frac{\psi}{2 v},
\]

\[
F^N \left( v \mid \lambda, \psi, \chi \right) = \frac{C}{2} \left( \frac{\chi}{\psi} \right)^{\lambda/2} v^\lambda \tilde{K}_{-\lambda} \left( \tilde{\psi}, \tilde{\chi} \right).
\]

The numerical CDF is given by

\[
F^N(v|\lambda, \psi, \chi) = 1 - F^N(v|\lambda, \psi, \chi).
\]
6.4 Numerical studies

This section is to compare accuracy of the analytical against numerical method to evaluate the CDF and CCDF of the subclasses of the GIG when $\lambda = \pm \frac{1}{2}$. The density function of the GIG used for the purpose of this section is given by (1.1). This is to agree with the density function used by the function pgig() of package GeneralizedHyperbolic (cf. p.416 of Barndorff-Nielsen, 1977). Consequently, the argument densBN of function gigCDF(), see Appendix E.2, is set to TRUE for all numerical tests in this section.

$P_{\text{Analytical}}$ Probabilities are calculated by evaluating the expressions in Sections 6.2. In these cases, the value of the involved incomplete Bessel functions are obtainable by evaluating the analytically derived formulae.

$P_{\text{Numerical}}$ Probabilities are calculated by evaluating the expressions in Section 6.3. The involved incomplete Bessel function, $\tilde{K}_\chi()$, is evaluated numerically using the algorithm by Sleevsky and Safouhi which was coded as the function gigCDF(). The values in this column agree with those returned by the function pgigGH(), which is an alias of pgig() of package GeneralizedHyperbolic, see Appendix E.1.

$P_{\text{Exact}}$ Probabilities are calculated using the Maple function in Appendix E.3 to numerically integrate the integral given by (1.1). These values are consider as “exact”.

6.4.1 When $\lambda = -\frac{1}{2}$

Evaluation of the cumulative distribution function

Given the values of parameters $q, \chi, \psi$ in Table 6.1, the following code can be used to reproduce the probabilities in each of the columns.

$P_{\text{Analytical}}$ gigCDF(q, chi, psi, -1/2, numeMethod = FALSE, densBN = TRUE, lower.tail = TRUE).
\(P_{\text{Numerical}}\) \(\text{gigCDF}(q, \chi, \psi, -1/2, \text{numeMethod} = \text{TRUE}, \text{traceIBF} = \text{FALSE}, \text{eps} = 0.85, \text{densBN} = \text{TRUE}, \text{lower.tail} = \text{TRUE}, nmax = 100)\)

The probabilities in this column agree with the probabilities calculated using \(\text{pgigGH}()\) which are not shown here but can be obtained using

\(\text{pgigGH}(q, \text{param} = c(\chi, \psi, -1/2), \text{lower.tail} = \text{TRUE})\)

\(P_{\text{Exact}}\) \(\text{gigCDFproc}(q, \chi, \psi, -1/2, \text{TRUE}, 90)\)

<table>
<thead>
<tr>
<th>(q)</th>
<th>(\chi)</th>
<th>(\psi)</th>
<th>(P_{\text{Analytical}})</th>
<th>(P_{\text{Numerical}})</th>
<th>(P_{\text{Exact}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>1</td>
<td>2</td>
<td>0.225 380 035 638 875</td>
<td>0.225 380 035 638 674</td>
<td>0.225 380 035 638 875</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0.056 892 627 116 680</td>
<td>0.056 892 627 116 497</td>
<td>0.056 892 627 116 680</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2</td>
<td>0.014 014 670 184 638</td>
<td>0.014 014 670 184 516</td>
<td>0.014 014 670 184 638</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>0.003 381 519 716 374</td>
<td>0.003 381 519 716 075</td>
<td>0.003 381 519 716 374</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>0.000 802 452 151 736</td>
<td>0.000 802 452 151 458</td>
<td>0.000 802 452 151 736</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>0.000 187 859 106 935</td>
<td>0.000 187 859 106 829</td>
<td>0.000 187 859 106 935</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>30</td>
<td>\underline{0.866 532 222 029 158}</td>
<td>\underline{0.867 355 170 725 715}</td>
<td>\underline{0.866 532 222 029 158}</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td></td>
<td>\underline{0.901 274 431 737 669}</td>
<td>\underline{0.899 912 503 082 038}</td>
<td>\underline{0.901 274 431 737 669}</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td></td>
<td>\underline{0.928 138 317 524 894}</td>
<td>\underline{0.930 934 589 656 491}</td>
<td>\underline{0.928 138 317 524 893}</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td></td>
<td>\underline{0.948 478 516 451 689}</td>
<td>\underline{0.950 012 627 734 676}</td>
<td>\underline{0.948 478 516 451 688}</td>
</tr>
<tr>
<td></td>
<td>38</td>
<td></td>
<td>\underline{0.963 584 978 823 662}</td>
<td>\underline{0.964 428 605 340 996}</td>
<td>\underline{0.963 584 978 823 662}</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td></td>
<td>\underline{0.974 606 481 052 737}</td>
<td>\underline{0.975 071 135 947 453}</td>
<td>\underline{0.974 606 481 052 736}</td>
</tr>
</tbody>
</table>

Table 6.1: Accuracy comparison between different methods to calculate \(P(W \leq w)\) for \(\lambda = -\frac{1}{2}\). Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CDF. Symbol \(-\) indicates repeated values.
Evaluation of the complementary cumulative distribution

Given the values of parameters \( q, \chi, \psi \) in Table 6.2, the following code can be used to reproduce the probabilities in each of the columns.

\[
P_{\text{Analytical}} = \text{gigCDF}(q, \chi, \psi, -1/2, \text{numeMethod} = \text{FALSE}, \text{densBN} = \text{TRUE}, \varepsilon = 0.85, \text{lower.tail} = \text{FALSE})
\]

\[
P_{\text{Numerical}} = \text{gigCDF}(q, \chi, \psi, -1/2, \text{numeMethod} = \text{TRUE}, \text{traceIBF} = \text{FALSE}, \text{densBN} = \text{TRUE}, \text{lower.tail} = \text{FALSE})
\]

The probabilities in this column agree with the probabilities calculated using \( \text{pgigGH()} \) which are not shown here but can be obtained using

\[
\text{pgigGH}(q, \text{param} = c(\chi, \psi, -1/2), \text{lower.tail} = \text{FALSE})
\]

\[
P_{\text{Exact}} = \text{gigCDFproc}(q, \chi, \psi, -1/2, \text{FALSE}, 90)
\]

<table>
<thead>
<tr>
<th>( q )</th>
<th>( \chi )</th>
<th>( \psi )</th>
<th>( P_{\text{Analytical}} )</th>
<th>( P_{\text{Numerical}} )</th>
<th>( P_{\text{Exact}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>1</td>
<td>1</td>
<td>0.999 979 426 935 233</td>
<td>0.999 979 426 935 252</td>
<td>0.999 979 426 935 232</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>3</td>
<td>0.999 959 141 468 832</td>
<td>0.999 959 141 468 839</td>
<td>0.999 959 141 468 832</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>5</td>
<td>0.999 935 399 153 930</td>
<td>0.999 935 399 153 945</td>
<td>0.999 935 399 153 930</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>7</td>
<td>0.999 907 054 049 643</td>
<td>0.999 907 054 049 673</td>
<td>0.999 907 054 049 643</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>9</td>
<td>0.999 873 481 719 986</td>
<td>0.999 873 481 720 038</td>
<td>0.999 873 481 719 986</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>11</td>
<td>0.999 834 139 961 950</td>
<td>0.999 834 139 962 035</td>
<td>0.999 834 139 961 950</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>30</td>
<td>0.463 877 916 472 010</td>
<td>0.681 337 288 999 150</td>
<td>0.463 877 916 472 010</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>32</td>
<td>0.393 742 524 408 322</td>
<td>0.243 410 607 941 572</td>
<td>0.393 742 524 408 322</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>34</td>
<td>0.328 885 683 092 246</td>
<td>0.236 423 173 472 920</td>
<td>0.328 885 683 092 246</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>36</td>
<td>0.270 470 801 845 749</td>
<td>0.214 529 597 904 605</td>
<td>0.270 470 801 845 748</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>38</td>
<td>0.219 115 842 595 153</td>
<td>0.185 543 075 627 905</td>
<td>0.219 115 842 595 152</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>40</td>
<td>0.174 963 722 223 139</td>
<td>0.154 881 697 068 010</td>
<td>0.174 963 722 223 139</td>
</tr>
</tbody>
</table>

Table 6.2: Accuracy comparison between different methods to evaluate \( P(W > w) \) for \( \lambda = -\frac{1}{2} \). Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CCDF. Symbol – indicate repeated values.
6.4.2 When $\lambda = \frac{1}{2}$

Evaluation of the cumulative function distribution

Given the values of parameters $q, \chi, \psi$ in Table 6.3, the following code can be used to reproduce the probabilities in each of the columns.

- $P_{\text{Analytical}}$: `gigCDF(q, chi, psi, 1/2, numeMethod = FALSE, densBN = TRUE, lower.tail = TRUE)`
- $P_{\text{Numerical}}$: `gigCDF(q, chi, psi, 1/2, numeMethod = TRUE, traceIBF = FALSE, densBN = TRUE, eps = 0.85, lower.tail = TRUE)`

The probabilities in this column agree with the probabilities calculated using `pgigGH()` which are not shown here but can be obtained using

- $P_{\text{Exact}}$: `gigCDFproc(q, chi, psi, 1/2, TRUE, 90)`

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\chi$</th>
<th>$\psi$</th>
<th>$P_{\text{Analytical}}$</th>
<th>$P_{\text{Numerical}}$</th>
<th>$P_{\text{Exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>0.205 960 519 371 504</td>
<td>0.205 960 519 371 266</td>
<td>0.205 960 519 371 504</td>
</tr>
<tr>
<td>−</td>
<td>2</td>
<td></td>
<td>0.084 953 318 671 071</td>
<td>0.084 953 318 670 864</td>
<td>0.084 953 318 671 071</td>
</tr>
<tr>
<td>−</td>
<td>3</td>
<td></td>
<td>0.035 927 166 620 621</td>
<td>0.035 927 166 620 453</td>
<td>0.035 927 166 620 621</td>
</tr>
<tr>
<td>−</td>
<td>4</td>
<td></td>
<td>0.015 284 857 774 166</td>
<td>0.015 284 857 774 049</td>
<td>0.015 284 857 774 167</td>
</tr>
<tr>
<td>−</td>
<td>5</td>
<td></td>
<td>0.006 505 684 022 866</td>
<td>0.006 505 684 022 791</td>
<td>0.006 505 684 022 866</td>
</tr>
<tr>
<td>−</td>
<td>6</td>
<td></td>
<td>0.002 764 535 633 950</td>
<td>0.002 764 535 633 790</td>
<td>0.002 764 535 633 950</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>20</td>
<td>0.133 467 777 970 842</td>
<td>0.132 644 829 274 284</td>
<td>0.133 467 777 970 841</td>
</tr>
<tr>
<td>−</td>
<td>32</td>
<td></td>
<td>0.098 725 568 262 330</td>
<td>0.100 087 496 917 962</td>
<td>0.098 725 568 262 330</td>
</tr>
<tr>
<td>−</td>
<td>34</td>
<td></td>
<td>0.071 861 682 475 106</td>
<td>0.072 537 725 446 068</td>
<td>0.071 861 682 475 106</td>
</tr>
<tr>
<td>−</td>
<td>36</td>
<td></td>
<td>0.051 521 483 548 310</td>
<td>0.051 859 743 298 150</td>
<td>0.051 521 483 548 311</td>
</tr>
<tr>
<td>−</td>
<td>38</td>
<td></td>
<td>0.036 415 021 176 337</td>
<td>0.035 571 394 659 003</td>
<td>0.036 415 021 176 337</td>
</tr>
<tr>
<td>−</td>
<td>40</td>
<td></td>
<td>0.025 393 518 947 263</td>
<td>0.024 928 864 052 546</td>
<td>0.025 393 518 947 263</td>
</tr>
</tbody>
</table>

Table 6.3: Accuracy comparison between different methods to evaluate $P(W \leq w)$ for $\lambda = \frac{1}{2}$. Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CDF. Symbol − indicate repeated values.
Evaluation of the complementary cumulative distribution

Given the values of parameters $q$, $\chi$, $\psi$ in Table 6.4, the following code can be used to reproduce the probabilities in each of the columns.

\[ P_{\text{Analytical}} \quad \text{gigCDF}(q, \chi, \psi, 1/2, \text{numeMethod} = \text{FALSE}, \text{densBN} = \text{TRUE}, \text{lower.tail} = \text{FALSE}) \]

\[ P_{\text{Numerical}} \quad \text{gigCDF}(q, \chi, \psi, 1/2, \text{numeMethod} = \text{TRUE}, \text{traceIBF} = \text{FALSE}, \text{densBN} = \text{TRUE}, \text{lower.tail} = \text{FALSE}) \]

The probabilities in this column agree with the probabilities calculated using \text{pgigGH()} which are not shown here but can be obtained using

\[ \text{pgigGH}(q, \text{param} = \text{c}(\chi, \psi, 1/2), \text{lower.tail} = \text{FALSE}) \]

\[ P_{\text{Exact}} \quad \text{gigCDFproc}(q, \chi, \psi, 1/2, \text{FALSE}, 90) \]

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\chi$</th>
<th>$\psi$</th>
<th>$P_{\text{Analytical}}$</th>
<th>$P_{\text{Numerical}}$</th>
<th>$P_{\text{Exact}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>0.999 649 585 462 791</td>
<td>0.999 649 585 462 795</td>
<td>0.999 649 585 462 791</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.998 843 542 610 088</td>
<td>0.998 843 542 610 107</td>
<td>0.998 843 542 610 088</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.997 735 075 728 394</td>
<td>0.997 735 075 728 441</td>
<td>0.997 735 075 728 393</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.996 300 050 718 259</td>
<td>0.996 300 050 718 355</td>
<td>0.996 300 050 718 258</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.994 519 518 736 490</td>
<td>0.994 519 518 736 663</td>
<td>0.994 519 518 736 489</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.992 376 529 168 932</td>
<td>0.992 376 529 169 028</td>
<td>0.992 376 529 168 932</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>20</td>
<td>0.866 532 222 029 158</td>
<td>0.867 355 170 725 715</td>
<td>0.866 532 222 029 158</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.812 823 868 503 060</td>
<td>0.815 220 845 792 791</td>
<td>0.812 823 868 503 060</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.750 685 453 385 827</td>
<td>0.757 185 530 303 768</td>
<td>0.750 685 453 385 827</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.682 172 394 351 786</td>
<td>0.641 835 327 066 013</td>
<td>0.682 172 394 351 786</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.609 775 707 454 834</td>
<td>0.752 659 874 843 139</td>
<td>0.609 775 707 454 835</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.536 122 083 527 988</td>
<td>1.089 555 147 220 593</td>
<td>0.536 122 083 527 989</td>
</tr>
</tbody>
</table>

Table 6.4: Accuracy comparison between different methods to evaluate $P(W > w)$ for $\lambda = \frac{1}{2}$. Underlined numbers indicate the decimal places where an evaluation agrees with the accurate values of the CCDF. Symbol ⋆ and − indicate greater than 1 probabilities and repeated values respectively.
6.4.3 Numerical discussion

As can be seen from Tables 6.1–6.4, the probabilities calculated using the numerical approach (i.e., columns $P_{\text{Numerical}}$) closely agree with the “exact” probabilities (i.e., columns $P_{\text{Exact}}$) when the value of parameters $\chi$ and $\psi$ are small. These tested parameters are presented in the first half of these tables. However, if the values of $\chi$ and $\psi$ become larger (i.e., tested parameter values on the second half of these tables) then the numerical method is inaccurate. The reason for these inaccuracies is that the incomplete Bessel function, which is a strictly decreasing function of its arguments (i.e, $\sqrt{\chi \psi}$), assumes values which makes the stopping rule (5.63) computationally a flaw. In these cases, if the diagnostic tool $\text{traceIFB}$ is set to $\text{TRUE}$ then the numerical issue of the algorithm by Slevensky and Safouhi discussed in Section 5.7.3 are clearly shown. Tables 6.1–6.4 also show that the analytical method is highly accurate for all values of $q$, $\chi$ and $\psi$. 
Chapter 7

Goodness-of-fit test for the hyperbolic distribution

7.1 Introduction


Given a sample which consists of $x_1, \ldots, x_n$ observations (i.i.d.) and a specified continuous distribution $F(x; \theta)$. It is of interest to test the hypothesis that the sample comes from $F(x; \theta)$ where $\theta$ is the parameter vector with $p$ components. The Cramér-von Mises test statistic, $W^2$, is based on the comparison of $F(x; \theta)$ with the empirical distribution function (EDF)

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} 1 \left( x_i \leq x \right).$$

Let

$$\beta(x) = n^{1/2} \left[ F_n(x) - F(x; \theta) \right], \quad (7.1)$$
then the Cramér-von Mises is given by

\[ W^2 = \int_{-\infty}^{\infty} \beta^2(x) dF(x; \theta). \]

Cramér-von Mises (CvM) is one of the EDF tests which include the Anderson-Darling \( A^2 \), the Kolmogorov \( D \) and the Watson \( U^2 \), see Stephens (1974). In cases where some or all parameters of \( p \) must be estimated, expression (7.1) is called estimated empirical process and it can be defined as (cf. p.243 of Stute et al. 1993)

\[ \hat{\beta} = n^{1/2} \left[ F_n(x) - F(x; \hat{\theta}) \right]. \] (7.2)

Theorem 1 of Durbin (1973) shows that under regularity condition, as the sample size tends to infinity, the resulting sequence of the empirical processes converges weakly to a Gaussian process. This result provides a rigorous foundation for computing critical values of the EDF statistics. However, the application of asymptotic distribution when some or all components of \( \theta \) must be estimated leads to a well-known problem: the distribution of \( W^2 \) (and others EDF statistics) depends critically on the distribution tested and on the true value of \( \theta \) (cf. p.97 of D’Agostino and Stephens 1986). Consequently, different tables of critical values are required for each particular situation, depending on the number of unknown parameters. Stephens (1976) gave detailed discussions for each situation involving goodness-of-fit for the normal distribution.

Stute et al. (1993) suggested the use of bootstrap methodology to give a parametric bootstrap version of the invariance principle of the empirical process stated in Theorem 1 of Durbin (1973). The validity of bootstrap-based goodness-of-fit was reviewed in Genest and Rémillard (2008). The authors stated on p.1096 that if \( F_n^*(x) \) and \( F^*(x; \hat{\theta}) \) are analogs of \( F_n(x) \) and \( F(x; \hat{\theta}) \) computed from a sample then
the bootstrap version of the empirical processes

\[ \hat{\beta}^* = n^{1/2} \left[ F^*_n(x) - F^*(x, \hat{\theta}) \right] \]

and \( \hat{\beta} \) in (7.2) converge jointly in distribution to independent copies of the same limit. Bootstrap-based goodness-of-fit tests for copulas were studied in Genest et al. (2006), Genest and Rémillard (2008), Kojadinovic et al. (2010), Kojadinovic and Yan (2011).

Besides providing Table 7.1, Puig and Stephens (2001) also gave details of the algorithm they used to obtain this table. This algorithm can be seen as a numeric quantile function of the asymptotic distribution of the Cramér-von Mises test statistics for the HYP. It was implemented in R by Zhu (2007) to obtain a routine called \texttt{criticalValues}(\xi, \chi, \alpha) for calculating \( W^2_\alpha \). Given \( \alpha \) and the estimates of \( \xi \) and \( \chi \), this function allows one to conduct test by comparing the calculated \( W^2_\alpha \) with \( W^2_n \) in (7.4). The null hypothesis \( H_0 \) is rejected if \( W^2_n > W^2_\alpha \). However, it takes about 3 minutes to compute a value of \( W^2_\alpha \).

This chapter has two purposes. Firstly, it is to implement an R routine which can speedily compute the goodness-of-fit \( p \)-value of Puig and Stephens (2001). Secondly, it implements the Moran goodness-of-fit test in Cheng and Stephens (1989). The chapter is structured as follows. Section 7.1 gives a literature review. Section 7.2 examines the current and suggested methods for conducting goodness-of-fit test for the HYP. Section 7.3 aims at building an extended table of critical values and applying the function \texttt{polymars}() to approximate \( p \)-value of the test. Section 7.4 implements the Moran log-spacings statistic for goodness-of-fit for the HYP. Section 7.5 is a numerical study.
7.1.1 Cramér-von Mises goodness-of-fit test for the hyperbolic distribution

Tests for the HYP related distributions, among others, include Cheng and Stephens (2000) and Cheng and Stephens (2007) for the Laplace and skewed-Laplace distribution respectively. However, Puig and Stephens (2001) seems to be the only work on goodness-of-fit for the HYP where it was established that the asymptotic distribution of $W^2$ is the same as that of

$$S = \sum_{i=1}^{\infty} y_i \lambda_i,$$

with $y_i$ being independent $\chi^2_1$ random variables and $\lambda_i$ the weights. Once the weights are found, and for a given level of significance $\alpha$, the critical values of the test statistic $W^2_\alpha$ are obtained by numerically inverting the characteristics function (7.3). Detailed discussion on this approach can be found in Imhof (1961).

Durbin (1973) showed that the distribution of the Cramér-von Mises test statistic is scale and location invariant. Thus, the tables of critical values constructed by Puig and Stephens (2001) depend on the shape parameters $\xi$ and $\chi$ but not on the location $\mu$ and the scale $\delta$. The authors used $\mu = 0$ and $\delta = 1$ to calculate the percentage points of the test. Finally, since the distribution of $W^2$ under the null hypothesis is different when parameter(s) must be estimated, the critical values $W^2_\alpha$ were presented on Tables 3–5 for the following cases:

- **Case 0.** All four components of $\theta$ are known. Thus, the distribution tested is completely specified. Tests in this case have been well-studied in the literature so it is not considered further.

- **Case 1.** Parameters of location $\mu$ and scale $\delta$ are both known.

- **Case 2.** Location parameter $\mu$ is known.
• **Case 3.** All four parameters of $\theta$ must be estimated. This is the most frequent situation and it is the focus of this chapter. For this case, Table 5 of [Puig and Stephens (2001)] gives critical $W_\alpha$ of the test. This table was reproduced as Table 7.1.
Table 7.1: Critical points of $W^2$ when all the parameters are unknown.

<table>
<thead>
<tr>
<th>$\xi$</th>
<th>$\alpha$</th>
<th>$\chi$</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.250</td>
<td>0.051</td>
<td>0.052</td>
<td>0.060</td>
<td>0.073</td>
<td>0.088</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.100</td>
<td>0.068</td>
<td>0.072</td>
<td>0.086</td>
<td>0.108</td>
<td>0.132</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.081</td>
<td>0.086</td>
<td>0.106</td>
<td>0.136</td>
<td>0.168</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.095</td>
<td>0.102</td>
<td>0.128</td>
<td>0.165</td>
<td>0.204</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.112</td>
<td>0.122</td>
<td>0.157</td>
<td>0.205</td>
<td>0.254</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.250</td>
<td>0.041</td>
<td>0.043</td>
<td>0.050</td>
<td>0.062</td>
<td>0.078</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>0.100</td>
<td>0.055</td>
<td>0.059</td>
<td>0.070</td>
<td>0.089</td>
<td>0.115</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.065</td>
<td>0.070</td>
<td>0.086</td>
<td>0.111</td>
<td>0.144</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.075</td>
<td>0.082</td>
<td>0.103</td>
<td>0.133</td>
<td>0.174</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.088</td>
<td>0.099</td>
<td>0.126</td>
<td>0.164</td>
<td>0.215</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.250</td>
<td>0.038</td>
<td>0.039</td>
<td>0.046</td>
<td>0.056</td>
<td>0.073</td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td>0.100</td>
<td>0.049</td>
<td>0.052</td>
<td>0.064</td>
<td>0.079</td>
<td>0.106</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.058</td>
<td>0.062</td>
<td>0.077</td>
<td>0.098</td>
<td>0.131</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.066</td>
<td>0.072</td>
<td>0.092</td>
<td>0.116</td>
<td>0.157</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.078</td>
<td>0.086</td>
<td>0.111</td>
<td>0.142</td>
<td>0.193</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.250</td>
<td>0.036</td>
<td>0.037</td>
<td>0.044</td>
<td>0.054</td>
<td>0.074</td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>0.100</td>
<td>0.046</td>
<td>0.049</td>
<td>0.059</td>
<td>0.076</td>
<td>0.106</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.054</td>
<td>0.058</td>
<td>0.071</td>
<td>0.092</td>
<td>0.131</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.063</td>
<td>0.068</td>
<td>0.083</td>
<td>0.108</td>
<td>0.157</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.073</td>
<td>0.080</td>
<td>0.099</td>
<td>0.131</td>
<td>0.191</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.250</td>
<td>0.036</td>
<td>0.038</td>
<td>0.044</td>
<td>0.055</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td>0.100</td>
<td>0.047</td>
<td>0.050</td>
<td>0.059</td>
<td>0.076</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.057</td>
<td>0.061</td>
<td>0.073</td>
<td>0.093</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.065</td>
<td>0.070</td>
<td>0.084</td>
<td>0.110</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.076</td>
<td>0.082</td>
<td>0.100</td>
<td>0.132</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.250</td>
<td>0.040</td>
<td>0.042</td>
<td>0.048</td>
<td>0.066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>0.100</td>
<td>0.052</td>
<td>0.055</td>
<td>0.065</td>
<td>0.091</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.062</td>
<td>0.066</td>
<td>0.077</td>
<td>0.111</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.071</td>
<td>0.076</td>
<td>0.090</td>
<td>0.131</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.084</td>
<td>0.089</td>
<td>0.107</td>
<td>0.158</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.250</td>
<td>0.043</td>
<td>0.045</td>
<td>0.052</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.100</td>
<td>0.057</td>
<td>0.060</td>
<td>0.071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.068</td>
<td>0.071</td>
<td>0.084</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.078</td>
<td>0.083</td>
<td>0.098</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.092</td>
<td>0.098</td>
<td>0.117</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7.1: continued

<table>
<thead>
<tr>
<th>ξ</th>
<th>α</th>
<th>χ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.250</td>
<td>0.046 0.048 0.062</td>
</tr>
<tr>
<td></td>
<td>0.100</td>
<td>0.062 0.065 0.085</td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.073 0.077 0.103</td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.085 0.090 0.120</td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.100 0.106 0.144</td>
</tr>
<tr>
<td>0.40</td>
<td>0.250</td>
<td>0.049 0.051</td>
</tr>
<tr>
<td></td>
<td>0.100</td>
<td>0.066 0.069</td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.078 0.083</td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.091 0.096</td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.108 0.114</td>
</tr>
<tr>
<td>0.30</td>
<td>0.250</td>
<td>0.050 0.053</td>
</tr>
<tr>
<td></td>
<td>0.100</td>
<td>0.068 0.071</td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.085 0.094</td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.099 0.111</td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.117 0.134</td>
</tr>
<tr>
<td>0.20</td>
<td>0.250</td>
<td>0.053</td>
</tr>
<tr>
<td></td>
<td>0.100</td>
<td>0.071</td>
</tr>
<tr>
<td></td>
<td>0.050</td>
<td>0.086</td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.100</td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.118</td>
</tr>
</tbody>
</table>

Table 7.1: A reproduction of table of Table 5 by Puig and Stephens

7.2 Current and recommended approaches

This section describes and discusses issues with the current approaches for conducting Cramér-von Mises based test.

7.2.1 Look-up table

Table look-up is the approach which was suggested on p.3 of [Puig and Stephens (2001)]. It involves the following steps:
Step 1. Find estimates of the unknown parameters using an efficient method. The function `standHyperbFit()` described in Appendix B.4 can be used for this purpose. Note that `standHyperbFit()` returns vector of estimated parameters as Parameterization 1 of package `GeneralizedHyperbolic` so these estimates must be converted to $\hat{\chi}, \hat{\xi}$ which are in Parameterization 4.

Step 2. Make the transformation $z(i) = G(x(i), \hat{\theta})$, for $i = 1, \ldots, n$, where $x(i)$’s are the order statistics of the sample in ascending order. The Cramér-von Mises statistic is given by

$$W_n^2 = \sum_{i=1}^{n} \left( z(i) - \frac{2i - 1}{2n} \right)^2 + \frac{1}{12n}. \quad (7.4)$$

Step 3. Refer to Table 7.1. The distribution of the Cramér-von Mises statistic does not depend on sign of $\chi$. Therefore enter the table with the value of $\hat{\xi}$ and $|\hat{\chi}|$. The table then gives critical point for 5 values of $W_\alpha^2$. If the value of $W_n^2$ is greater than $W_\alpha^2$, the null hypothesis is rejected at this level of significance.

Step 4. If a $p$-value is required then this can be easily found by interpolation provided the value is less than 0.25. In general if $p > 0.25$ then a very accurate value is not required. An estimate can be made approximating the distribution by the random variable of the form $a \cdot \chi_b^2$ with $a$ and $b$ chosen to match 25% and 10% in the table.

The obvious issue with the look-up table approach is that an accurate decision about $H_0$ is only obtainable if $\hat{\xi}$ and $|\hat{\chi}|$ are identical or very close to the value of $\xi$ and $\chi$ presented on Table 7.1. However, this is not always likely because $\hat{\xi}$ and $|\hat{\chi}|$ are parameter estimates. For example, given a significant level $\alpha$ what decision about $H_0$ should be made if $0.8 < \hat{\chi} < 0.9$ and $0.95 < \hat{\xi} < 0.99$, say.
7.2.2 Interpolation

The function `hypCvMtestSpline()` in package `HyperbolicDist` implements a linear interpolation algorithm to fill-in the “gaps” of Table 7.1 and to compute p-value of the test as suggested in Step 4. However, if the value of $W_n^2$ is beyond the margin of Table 7.1 (e.g., $\hat{\xi} = 0.99$ and $\hat{\chi} > 0.8$) where there are no tabulated values of $W_\alpha^2$ then the interpolation approach cannot be used while extrapolation is not a desired method. Below is an example output:

```r
> Theta <- c(2,2,2,2)
> dataVector <- rhyperb(500, Theta)
> fittedTheta <- hyperbFit(dataVector)
> hyperbCvMTest(dataVector, fittedTheta)
```

Cramer-von Mises test of hyperbolic distribution
data: dataVector
$Wsq = 0.2521$, $xi = 0.995$, $chi = 0.733$, p-value < 0.01
Warning message:
In `print.hyperbCvMTest(RVAL, ...)`: Estimated parameters are outside the table.
p-value may be incorrect

7.2.3 Multivariate adaptive polynomial spline regression

[Zhu (2007)] attempted to solve the issues of the look-up table and interpolation approach. It was stated that “We are primarily interested in expanding the table and finding a better approach to approximate the p-value of the Cramér-von Mises test statistic for the hyperbolic distribution with all parameters unknown.” The function `polymars()` in package `polspline` and the $W_\alpha^2$ values presented on Table 7.1 were used to predict p-value. Unfortunately, the fitted p-value values were in $(−0.27, 1.19)$ for some large values of $\xi$ and $\chi$ (cf. p.24 of [Zhu] 2007).
7.2.4 Recommended approaches

The approaches which were recommended by the examiners are the bootstrap-based goodness-of-fit test and the use of logit transformation.

Bootstrap based goodness-of-fit for the HYP

By following the steps on p.247 of [Stute et al. (1993)], goodness-of-fit for the HYP can be obtained. However, Associate-Professor David Scott and I think that computational speed of this approach may be an issue, because bootstrap involves a large number of repeated sampling coupled with the need to find maximum likelihood estimates multiple times. Referring to the bootstrap based goodness-of-fit in Stute et al. (1993), Kojadinovic and Yan (2011) noted that “the main inconvenience of this approach is its very high computational cost”. Thus, the focus is on the objective of this chapter which is to make an improvement to the computing speed of the Cramér-von Mises based test in Puig and Stephens (2001).

Logit transformation

The logit transformation is to confine the fitted \( p \)-value inside the \([0, 1]\) interval. This recommendation is implemented in Section 7.3

7.3 Numeric distribution function of the test statistic for the hyperbolic distribution

For the goodness-of-fit purpose, this section proposes a method to obtain the so-called numeric distribution function (NDF) of the test statistic. Firstly, Table 7.1 is extended by using \( \text{criticalValues}(\xi, \chi, \alpha) \). A network of computers was used because the algorithm of \( \text{criticalValues}(\xi, \chi, \alpha) \) is extremely computationally intensive.

\[\text{These study is also available at the first author’s website: http://ikojadin.perso.univ-pau.}\]
(cf. p.312 of [Puig and Stephens, 2001]). The function `polymars()` is then applied to fit a nonparametric regression model to the table for approximating \( p \)-value. In other words, the NDF is a fitted object of class polymars which is used to approximate the asymptotic distribution function of the Cramér-von Mises test statistic for the HYP. It appears that the approach pursued here is a typical one in applied research. When analytic form of a function is not obtainable or computational cost is an issue, fitting the function pre-calculated values to a regression model, among others, is a viable approach to speedily approximate the function true values. However, before any effort is committed, a question which may be asked is: why not directly improve the calculation speed of `criticalValues(\( \xi, \chi, \alpha \))` itself? Though this may be a possible approach, the practical gain in computing time may be limited.

### 7.3.1 Extended table of critical values for the test

Table 7.2 presents the extended table of \( W^2_\alpha \) values. The variable \( \xi, \chi \) and \( \alpha \) assumes 11, 6 and 7 distinct values respectively while the critical values, \( W^2_\alpha \), takes 462 values (i.e., \( 11 \times 6 \times 7 \)). Figures 7.1 and 7.2 are graphical representation of Table 7.2. Comparing between the plots, it appears that \( W^2_\alpha \) varies differently for all values of \( \alpha \). Secondly, the variation of \( W^2_\alpha \) decreases as the value of \( \alpha \) does. The plots seem to indicate certain degrees of collinearity between the test statistic and the shape parameters. Finally, Figures 7.1 shows some critical values which look like “outliers”. Closer examination reveals that these are not typographical errors and their occurrence was due to the numerical instability of `criticalValues(\( \xi, \chi, \alpha \))` in cases where \( \xi \approx \chi \). The issue was also noted on p.18 of [Zhu (2007)]. It is likely that this problem was due to the `integrate()` function (in `base`) which was called by `criticalValues(\( \xi, \chi, \alpha \))`.  

170
Table 7.2: Critical points of $W^2$ when all the parameters are unknown.

<table>
<thead>
<tr>
<th>$\xi$</th>
<th>$\alpha$</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>.99</td>
<td>.100</td>
<td>.068</td>
<td>.072</td>
<td>.086</td>
<td>.108</td>
<td>.132</td>
<td>.147</td>
</tr>
<tr>
<td></td>
<td>.075</td>
<td>.071</td>
<td>.076</td>
<td>.091</td>
<td>.114</td>
<td>.145</td>
<td>.163</td>
</tr>
<tr>
<td></td>
<td>.050</td>
<td>.081</td>
<td>.086</td>
<td>.106</td>
<td>.136</td>
<td>.168</td>
<td>.186</td>
</tr>
<tr>
<td></td>
<td>.040</td>
<td>.082</td>
<td>.092</td>
<td>.113</td>
<td>.134</td>
<td>.173</td>
<td>.200</td>
</tr>
<tr>
<td></td>
<td>.025</td>
<td>.095</td>
<td>.102</td>
<td>.128</td>
<td>.165</td>
<td>.204</td>
<td>.227</td>
</tr>
<tr>
<td></td>
<td>.015</td>
<td>.100</td>
<td>.112</td>
<td>.140</td>
<td>.179</td>
<td>.229</td>
<td>.258</td>
</tr>
<tr>
<td></td>
<td>.010</td>
<td>.112</td>
<td>.122</td>
<td>.157</td>
<td>.205</td>
<td>.254</td>
<td>.283</td>
</tr>
<tr>
<td>.95</td>
<td>.100</td>
<td>.055</td>
<td>.059</td>
<td>.070</td>
<td>.089</td>
<td>.115</td>
<td>.132</td>
</tr>
<tr>
<td></td>
<td>.075</td>
<td>.057</td>
<td>.062</td>
<td>.076</td>
<td>.097</td>
<td>.126</td>
<td>.145</td>
</tr>
<tr>
<td></td>
<td>.050</td>
<td>.065</td>
<td>.070</td>
<td>.086</td>
<td>.111</td>
<td>.144</td>
<td>.165</td>
</tr>
<tr>
<td></td>
<td>.040</td>
<td>.075</td>
<td>.082</td>
<td>.103</td>
<td>.133</td>
<td>.174</td>
<td>.201</td>
</tr>
<tr>
<td></td>
<td>.025</td>
<td>.066</td>
<td>.073</td>
<td>.090</td>
<td>.117</td>
<td>.152</td>
<td>.177</td>
</tr>
<tr>
<td></td>
<td>.015</td>
<td>.080</td>
<td>.090</td>
<td>.114</td>
<td>.149</td>
<td>.196</td>
<td>.228</td>
</tr>
<tr>
<td></td>
<td>.010</td>
<td>.088</td>
<td>.099</td>
<td>.126</td>
<td>.164</td>
<td>.215</td>
<td>.249</td>
</tr>
<tr>
<td>.90</td>
<td>.100</td>
<td>.049</td>
<td>.052</td>
<td>.064</td>
<td>.079</td>
<td>.106</td>
<td>.125</td>
</tr>
<tr>
<td></td>
<td>.075</td>
<td>.050</td>
<td>.055</td>
<td>.067</td>
<td>.087</td>
<td>.098</td>
<td>.138</td>
</tr>
<tr>
<td></td>
<td>.050</td>
<td>.058</td>
<td>.062</td>
<td>.077</td>
<td>.098</td>
<td>.131</td>
<td>.156</td>
</tr>
<tr>
<td></td>
<td>.040</td>
<td>.058</td>
<td>.064</td>
<td>.080</td>
<td>.104</td>
<td>.121</td>
<td>.166</td>
</tr>
<tr>
<td></td>
<td>.025</td>
<td>.066</td>
<td>.072</td>
<td>.092</td>
<td>.116</td>
<td>.157</td>
<td>.188</td>
</tr>
<tr>
<td></td>
<td>.015</td>
<td>.070</td>
<td>.079</td>
<td>.100</td>
<td>.132</td>
<td>.160</td>
<td>.212</td>
</tr>
<tr>
<td></td>
<td>.010</td>
<td>.078</td>
<td>.086</td>
<td>.111</td>
<td>.142</td>
<td>.193</td>
<td>.232</td>
</tr>
<tr>
<td>.80</td>
<td>.100</td>
<td>.046</td>
<td>.049</td>
<td>.059</td>
<td>.076</td>
<td>.106</td>
<td>.133</td>
</tr>
<tr>
<td></td>
<td>.075</td>
<td>.048</td>
<td>.052</td>
<td>.063</td>
<td>.081</td>
<td>.109</td>
<td>.113</td>
</tr>
<tr>
<td></td>
<td>.050</td>
<td>.054</td>
<td>.058</td>
<td>.071</td>
<td>.092</td>
<td>.131</td>
<td>.126</td>
</tr>
<tr>
<td></td>
<td>.040</td>
<td>.055</td>
<td>.060</td>
<td>.074</td>
<td>.096</td>
<td>.130</td>
<td>.134</td>
</tr>
<tr>
<td></td>
<td>.025</td>
<td>.063</td>
<td>.068</td>
<td>.083</td>
<td>.108</td>
<td>.157</td>
<td>.151</td>
</tr>
<tr>
<td></td>
<td>.015</td>
<td>.066</td>
<td>.073</td>
<td>.091</td>
<td>.120</td>
<td>.164</td>
<td>.169</td>
</tr>
<tr>
<td></td>
<td>.010</td>
<td>.073</td>
<td>.080</td>
<td>.099</td>
<td>.131</td>
<td>.191</td>
<td>.184</td>
</tr>
<tr>
<td>.70</td>
<td>.100</td>
<td>.047</td>
<td>.050</td>
<td>.059</td>
<td>.076</td>
<td>.076</td>
<td>.090</td>
</tr>
<tr>
<td></td>
<td>.075</td>
<td>.050</td>
<td>.054</td>
<td>.064</td>
<td>.081</td>
<td>.090</td>
<td>.098</td>
</tr>
<tr>
<td></td>
<td>.050</td>
<td>.057</td>
<td>.061</td>
<td>.073</td>
<td>.093</td>
<td>.093</td>
<td>.110</td>
</tr>
<tr>
<td></td>
<td>.040</td>
<td>.058</td>
<td>.062</td>
<td>.075</td>
<td>.095</td>
<td>.106</td>
<td>.116</td>
</tr>
<tr>
<td></td>
<td>.025</td>
<td>.065</td>
<td>.070</td>
<td>.084</td>
<td>.110</td>
<td>.110</td>
<td>.130</td>
</tr>
<tr>
<td></td>
<td>.015</td>
<td>.070</td>
<td>.076</td>
<td>.092</td>
<td>.119</td>
<td>.132</td>
<td>.144</td>
</tr>
<tr>
<td></td>
<td>.010</td>
<td>.076</td>
<td>.082</td>
<td>.100</td>
<td>.132</td>
<td>.132</td>
<td>.156</td>
</tr>
<tr>
<td>.60</td>
<td></td>
<td>.490</td>
<td>.550</td>
<td>.590</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.100</td>
<td>.052</td>
<td>.055</td>
<td>.065</td>
<td>.070</td>
<td>.076</td>
<td>.082</td>
</tr>
<tr>
<td></td>
<td>.075</td>
<td>.055</td>
<td>.058</td>
<td>.069</td>
<td>.076</td>
<td>.083</td>
<td>.089</td>
</tr>
<tr>
<td></td>
<td>.050</td>
<td>.062</td>
<td>.066</td>
<td>.077</td>
<td>.084</td>
<td>.092</td>
<td>.099</td>
</tr>
<tr>
<td></td>
<td>.040</td>
<td>.064</td>
<td>.068</td>
<td>.080</td>
<td>.089</td>
<td>.097</td>
<td>.105</td>
</tr>
<tr>
<td></td>
<td>.025</td>
<td>.071</td>
<td>.076</td>
<td>.090</td>
<td>.099</td>
<td>.108</td>
<td>.116</td>
</tr>
<tr>
<td></td>
<td>.015</td>
<td>.077</td>
<td>.082</td>
<td>.098</td>
<td>.110</td>
<td>.120</td>
<td>.130</td>
</tr>
</tbody>
</table>
Table 7.2: Columns with underlined values of $\chi$ and also the rows with $\alpha = 0.075$, 0.04, 0.15 are not available in Table 7.1 (e.g., Table 5 by Puig and Stephens.)

<table>
<thead>
<tr>
<th>$\xi$</th>
<th>$\alpha$</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>.100</td>
<td>.010</td>
<td>.084</td>
<td>.089</td>
<td>.107</td>
<td>.118</td>
<td>.129</td>
<td>.140</td>
</tr>
<tr>
<td>.075</td>
<td></td>
<td>.060</td>
<td>.064</td>
<td>.071</td>
<td>.074</td>
<td>.079</td>
<td>.084</td>
</tr>
<tr>
<td>.050</td>
<td></td>
<td>.068</td>
<td>.071</td>
<td>.078</td>
<td>.084</td>
<td>.087</td>
<td>.093</td>
</tr>
<tr>
<td>.040</td>
<td></td>
<td>.070</td>
<td>.074</td>
<td>.083</td>
<td>.087</td>
<td>.092</td>
<td>.098</td>
</tr>
<tr>
<td>.025</td>
<td></td>
<td>.078</td>
<td>.083</td>
<td>.091</td>
<td>.098</td>
<td>.102</td>
<td>.109</td>
</tr>
<tr>
<td>.015</td>
<td></td>
<td>.085</td>
<td>.090</td>
<td>.101</td>
<td>.106</td>
<td>.113</td>
<td>.121</td>
</tr>
<tr>
<td>.010</td>
<td></td>
<td>.092</td>
<td>.098</td>
<td>.109</td>
<td>.117</td>
<td>.122</td>
<td>.130</td>
</tr>
<tr>
<td>.250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.330</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.350</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.390</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.075</td>
<td></td>
<td>.062</td>
<td>.065</td>
<td>.065</td>
<td>.069</td>
<td>.071</td>
<td>.074</td>
</tr>
<tr>
<td>.050</td>
<td></td>
<td>.073</td>
<td>.077</td>
<td>.078</td>
<td>.083</td>
<td>.085</td>
<td>.089</td>
</tr>
<tr>
<td>.040</td>
<td></td>
<td>.076</td>
<td>.080</td>
<td>.082</td>
<td>.088</td>
<td>.090</td>
<td>.094</td>
</tr>
<tr>
<td>.025</td>
<td></td>
<td>.085</td>
<td>.090</td>
<td>.091</td>
<td>.097</td>
<td>.099</td>
<td>.104</td>
</tr>
<tr>
<td>.015</td>
<td></td>
<td>.092</td>
<td>.097</td>
<td>.101</td>
<td>.108</td>
<td>.110</td>
<td>.115</td>
</tr>
<tr>
<td>.010</td>
<td></td>
<td>.100</td>
<td>.106</td>
<td>.109</td>
<td>.116</td>
<td>.118</td>
<td>.124</td>
</tr>
<tr>
<td>.150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.290</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.075</td>
<td></td>
<td>.066</td>
<td>.067</td>
<td>.069</td>
<td>.070</td>
<td>.071</td>
<td>.071</td>
</tr>
<tr>
<td>.050</td>
<td></td>
<td>.070</td>
<td>.072</td>
<td>.074</td>
<td>.076</td>
<td>.076</td>
<td>.077</td>
</tr>
<tr>
<td>.040</td>
<td></td>
<td>.078</td>
<td>.080</td>
<td>.083</td>
<td>.084</td>
<td>.085</td>
<td>.086</td>
</tr>
<tr>
<td>.025</td>
<td></td>
<td>.081</td>
<td>.084</td>
<td>.086</td>
<td>.089</td>
<td>.089</td>
<td>.090</td>
</tr>
<tr>
<td>.015</td>
<td></td>
<td>.091</td>
<td>.093</td>
<td>.096</td>
<td>.098</td>
<td>.099</td>
<td>.100</td>
</tr>
<tr>
<td>.010</td>
<td></td>
<td>.100</td>
<td>.106</td>
<td>.109</td>
<td>.116</td>
<td>.118</td>
<td>.119</td>
</tr>
<tr>
<td>.150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.190</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.075</td>
<td></td>
<td>.068</td>
<td>.069</td>
<td>.069</td>
<td>.070</td>
<td>.062</td>
<td>.097</td>
</tr>
<tr>
<td>.050</td>
<td></td>
<td>.073</td>
<td>.074</td>
<td>.075</td>
<td>.076</td>
<td>.067</td>
<td>.119</td>
</tr>
<tr>
<td>.040</td>
<td></td>
<td>.085</td>
<td>.082</td>
<td>.083</td>
<td>.085</td>
<td>.074</td>
<td>.117</td>
</tr>
<tr>
<td>.025</td>
<td></td>
<td>.086</td>
<td>.086</td>
<td>.087</td>
<td>.089</td>
<td>.078</td>
<td>.125</td>
</tr>
<tr>
<td>.015</td>
<td></td>
<td>.096</td>
<td>.097</td>
<td>.099</td>
<td>.087</td>
<td>.138</td>
<td>.138</td>
</tr>
<tr>
<td>.010</td>
<td></td>
<td>.104</td>
<td>.106</td>
<td>.107</td>
<td>.109</td>
<td>.096</td>
<td>.151</td>
</tr>
<tr>
<td>.117</td>
<td>.113</td>
<td>.115</td>
<td>.117</td>
<td>.104</td>
<td>.165</td>
<td>.165</td>
<td></td>
</tr>
<tr>
<td>.020</td>
<td>.030</td>
<td>.040</td>
<td>.050</td>
<td>.090</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.010</td>
<td>.071</td>
<td>.070</td>
<td>.072</td>
<td>.070</td>
<td>.071</td>
<td>.105</td>
<td></td>
</tr>
<tr>
<td>.075</td>
<td>.077</td>
<td>.076</td>
<td>.078</td>
<td>.076</td>
<td>.077</td>
<td>.113</td>
<td></td>
</tr>
<tr>
<td>.050</td>
<td>.086</td>
<td>.084</td>
<td>.086</td>
<td>.084</td>
<td>.085</td>
<td>.132</td>
<td></td>
</tr>
<tr>
<td>.040</td>
<td>.089</td>
<td>.089</td>
<td>.091</td>
<td>.089</td>
<td>.090</td>
<td>.137</td>
<td></td>
</tr>
<tr>
<td>.025</td>
<td>.100</td>
<td>.098</td>
<td>.101</td>
<td>.098</td>
<td>.099</td>
<td>.153</td>
<td></td>
</tr>
<tr>
<td>.015</td>
<td>.109</td>
<td>.108</td>
<td>.112</td>
<td>.109</td>
<td>.110</td>
<td>.178</td>
<td></td>
</tr>
<tr>
<td>.010</td>
<td>.118</td>
<td>.117</td>
<td>.120</td>
<td>.117</td>
<td>.118</td>
<td>.186</td>
<td></td>
</tr>
</tbody>
</table>
Figure 7.1: Trellis plot $\xi$ vs $W^2_\alpha$ conditioned on 7 levels of $\alpha$ in Table 7.2

The variables

A scalar probability $\pi$ is defined as

$$\pi = P(W^2 \leq W^2_\alpha | \xi, \chi) = 1 - \alpha.$$ 

Since $\alpha$ (in Table 7.2) is a vector of 7 elements, the logit transformation of $\pi$ is

$$\varphi = \log \left( \frac{\pi}{1 - \pi} \right).$$

The convention in [Wood (2006)] is followed in this chapter to call a variable the response variable if its behaviour is modelled by changes in the predictors. Response variable and predictors are also called dependent and explanatory variables respectively elsewhere.
Figure 7.2: Trellis plot $\chi$ vs $W_\alpha^2$ conditioned on 7 levels of $\alpha$ in Table 7.2

7.3.2 Multivariate adaptive polynomial spline fitting

This section proposes a method to fit the multivariate adaptive polynomial spline algorithm of `polymars()` to Table 7.2.

Some common approaches

Before attempting to use `polymars()` a question that should be raised is: why not try fitting this data to models such as multiple linear, generalized linear or additive regression?

- Multiple linear model:

  1. A model with $\varphi$ is the response variable and $\chi$, $\xi$, $W_\alpha^2$ are the predictors. This model is based on a rigid assumption that there is a linear relationship between $\varphi$ and each of the predictors. The model can be straightforwardly fitted using `lm()` (in `stats`), see [Wood (2006)](Wood2006). Expect-
edly, this model does not fit the data in Table 7.2. Some of the problems are: the residual plot show there are patterns left by the model, which is confirmed by very low residual sum of square (i.e., 41%). However, these patterns do not justify a quadratic or cubic model.

2. The second linear model in which the logit transformation $W_{\alpha}^2$ is the response and $\xi$, $\chi$ and $\alpha$ are predictors was also examined. Note that logit transformation of $W_{\alpha}^2$ is necessary because the domain of variation of $W_{\alpha}^2$ is the interval [0, 1]. This model, however, also fits the data very poorly.

- As multiple linear models are not suitable in this situation, it is natural to consider regression models which relax the rigid assumption of a linear relationship between the response and predictors. The generalized linear model (GLM) by McCullagh and Nelder [1989] and the generalized additive model (GAM) by Hastie and Tibshirani [1986], among others, are such models. The latter extends the former by replacing a linear predictor by an additive one. Harrell [2001] gives extensive theoretical discussion on regression modelling and Wood [2006] is a practical guide for modelling data using these tools in R. However, both the GAM and GLM are not suitable here because the response, neither $\varphi$ nor $W_{\alpha}^2$, justify the use of any of the specified link functions. This is because $\alpha$ is not a random number, while the distribution of $W_{\alpha}^2$ is not obtainable for a finite sample size but its asymptotic distribution is given by (7.3).

**Adaptive regression algorithm**

The algorithm of polymars() is called multivariate adaptive polynomial spline (MAPS) in Kooperberg et al. [1997]. The MAPS algorithm is an adaptive regression procedure, using piecewise-linear splines to model the response. It is different from the multivariate adaptive regression splines (MARS) algorithm² by Friedman [1991].

²MARS™ is a registered trademark of Jeril, Inc
though it has many similarities. An important feature of these approaches is that they employ the so-called adaptive computation. Friedman (1991) stated on p. 7 that “Adaptive computation strategies attempt to approximate general functions in high dimensionality are based on adaptive computation. An adaptive computation is one that dynamically adjusts its strategy to take into account the behaviour of the function to be approximated.” In statistics, two well-known adaptive algorithms for function approximation are the recursive partitioning (RP) by Morgan and Sonquist (1963) and projection pursuit regression (PPR) by Friedman and Stuetzle (1981) and Friedman et al. (1983). These algorithms were derived to address the problem of multi-dimensional regression called curse of dimensionality (cf. p. 83 of Hastie and Tibshirani, 1990). Recursive partitioning is the core part of both MAPS and MARS.

Let \( \hat{f}(y) \) be the fitted multi-dimensional function with \( y = (y_1, \ldots, y_n)^T \) vector of predictors. Friedman (1991) gave the following description of the recursive partitioning regression

\[
\hat{f}(y) = \sum_{m=1}^{M} a_m B_m(y).
\]

Here, \( M \) is the number of nonconstant basis functions in the MARS model and \( a_m \) is the recursive partitioning model which can be a piecewise-constant approximation. RP tackles the issue of curse of dimensionality in high-dimensional regression fitting by carving the predictor space into disjoint blocks in binary style, and hence the model is commonly represented by a binary tree. In the MAPS model, the basis functions \( B_m \) are the piecewise-linear functions or their tensor products, but MAPS choose these splines in an adaptive way, hence the name multivariate adaptive polynomial spline. Detailed explanation and real data example of MAPS were given on p. 121 of Kooperberg et al. (1997). Technical descriptions of the MARS can be found on pp. 8–20 of Friedman (1991).

Conducting regression fitting using polymars() is as simple as using lm(), glm() or gam() (in mgcv). The functions summary(), predict() and residuals() (in stats) can call the returned object of class polymars by polymars() to display coefficient values.
and model diagnostic. In addition, `plot.polymars()` and `persp.polymars()` produce 2 and 3-dimensional plots of the fitted values from a `polymars`. Features which make `polymars()` more flexible than `lm()`, `glm()` and `gam()` include the ability to fit multiple responses, and the function allows the modeller to specify whether the fitted model be additive in the predictors. The function `hypCvMPvalueSpline()` described in Appendix F.1 fits the data in Table 7.2 to a `polymars()` algorithm and returns predicted `p-value` for the Cramér-von Mises test. This function can be considered as the numeric distribution function (NDF) of the Cramér-von Mises test statistic. This function offers two important features:

1. It obviates the need for dividing up the ranges of parameters `ξ` and `χ`.
2. Its computational speed is extremely fast compared to these of the function `criticalValues(ξ, χ, α)`. This is the objective of this chapter and as expected.

**Residuals examination**

Examination of the distribution of the residual, which is given by

$$\hat{e} = α - \hat{α}$$

is an important step in regression fitting. Here, the residuals of all 7 fitted values of `α` are not of interest but those of `α = 0.1, 0.05` and `0.01` only. By using the function `pvalSplines()` and the corresponding values of `chiVec`, `xiVec` and `wsqVec` the predicted values `\hat{α}` for these levels of significance can be straightforwardly obtained. There are 66 values of `\hat{e}` (i.e., `11 \times 6`) for a given `α`. The residual plots in Figure 7.3 show that most of the residuals are centered around zero, which indicates that the model fits the data fairly well (`$R^2 = 0.8$`). However, it appears the distribution of the residual is not completely random (i.e., patternless). There is a curve in the residuals plots for `α = 0.1` and `0.01`. In addition, the 57th residual appears to be an outlier for all three values of `α`. It turns out that these residuals were calculated using the values
of $\hat{\alpha}$ given by $\text{pvalSplines}(\xi = 0.95, \chi = 0.9, W^2_\alpha)$ where $W^2_\alpha$ had been obtained using $\text{criticalValues}(\xi = 0.95, \chi = 0.9, \alpha)$ ($\alpha = 0.1, 0.05 \text{ and } 0.01$). Moreover, the fitted residuals when $\alpha = 0.01$ are far from centred about zero. This could be because of a problem with the function $\text{integrate()}$ in R when it was used to calculate critical values $W^2_\alpha$. This confirms the feature shown in Plots 7.1 and 7.2 that the values of $W^2_\alpha$ were changing very fast when $\chi \to \xi$. More importantly, the existence of outliers in the residual plot indicates that $\text{polymars()}$ seems sensitive to abrupt changes in the value of $W^2_\alpha$. This issue should be investigated in future research to improve the fit.

![Figure 7.3: Distribution of $\text{polymars()}$ regression fitting residuals.](image)

Figure 7.3: Distribution of $\text{polymars()}$ regression fitting residuals.
7.4 Moran log spacing goodness-of-fit test for the hyperbolic distribution

This section describes the steps required to conduct goodness-of-fit for the HYP applying the Moran test when \( \theta \) are unknown. Suppose \( x_1 < \cdots < x_n \) are the ordered statistics of the sample to be tested. Let \( y_i = F(x_i, \theta) \) be the values given by the probability transformation. The spacings \( D_i(\theta) \) are

\[
D_i(\theta) = y_i - y_{i-1},
\]

where \( i = 1, \ldots, m, m = n + 1, y_0 \equiv 0 \) and \( y_n \equiv 1 \). Then the Moran test statistic is

\[
M(\theta) = \sum_{i=1}^{m} \log D_i(\theta).
\]

\[\text{Wong and Li (2006)}\] showed that under the null hypothesis, \( M(\theta) \), being independent of the unknown parameters, has normal distribution and a chi-square approximation exists for small samples with mean and variance approximated respectively by

\[
\mu_M \approx (n + 1) \log(n + 1) - \frac{1}{2} - \frac{1}{12(n + 1)}
\]

and

\[
\sigma^2_M \approx (n + 1) \left( \frac{\pi^2}{6} - 1 \right) - \frac{1}{2} - \frac{1}{6(n + 1)}.
\]

Note that (7.5) and (7.6) assume the form of an infinite series in \[\text{Cheng and Stephens (1989)}\]. Furthermore,

\[
C_1 = \mu_M - \left( \frac{n}{2} \right)^{1/2} \sigma_M, \quad C_2 = (2n)^{-1/2} \sigma_M.
\]

The following are the steps for conducting goodness-of-fit for the hyperbolic distribution using Moran test statistic.
• **Step 1.** Obtain maximum likelihood estimates of $\hat{\theta}$ for the hyperbolic distribution, using the function `symFitShapeHyp()` or `hyperbFit()`.

• **Step 2.** Calculate the Moran test statistic.

$$T(\hat{\theta}) = \frac{M(\hat{\theta}) + k/2 - C_1}{C_2}.$$

• **Step 3.** Reject $H_0$ at significance level $\alpha$ if $T(\hat{\theta}) > \chi^2_n(\alpha)$

These steps were coded to obtain the function `hypMoranTest()` in Appendix F.2.

### 7.5 Numerical study

#### 7.5.1 Tests using skew normal distribution

This section examines the power of the Cramér-von Mises and Moran test using pseudo data of the skew-normal (SN) distribution. The family of SN distributions is an extension of the normal family, via the introduction of a shape parameter which regulates skewness; when shape equals 0, the SN distribution reduces to the normal one.

The null hypothesis for these tests is that the data is from the hyperbolic distribution, which is false. The power of these tests is the proportion of the times they correctly reject the false null hypothesis. The reason for choosing SN data is because this distribution have significant mass in the tails and greater kurtosis than the normal distribution which are the features of the HYP. The idea here is to obtain a test data set which is as similar to the data modelled by the null hypothesis as possible.

The data was generated using the random generator in package `sn`. The fitting functions `standHyperbFit()` and `hyperbFit()` were used to fit this data. The fitted parameters and the data were then used by the functions `hypCvMtestSpline()`
and `hypMoranTest()` to conduct goodness-of-fit test, see Appendix F.3. The number of tests is 800.

In the table of test results, the column headed by the name of the goodness-of-fit test function indicate the type of test performed. The columns headed by the name of the fitting function indicate by which method the fitted parameters were obtained. The fractional numbers show the proportion of the times that the test correctly reject the null hypothesis, which is the power of the test.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th><code>hypCvMtestSpline()</code></th>
<th><code>hypMoranTest()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>standHyperbFit()</code></td>
<td><code>hyperbFit()</code></td>
</tr>
<tr>
<td></td>
<td><code>standHyperbFit()</code></td>
<td><code>hyperbFit()</code></td>
</tr>
<tr>
<td>1</td>
<td>0.98</td>
<td>0.16</td>
</tr>
<tr>
<td>0.5</td>
<td>0.91</td>
<td>0.32</td>
</tr>
<tr>
<td>10</td>
<td>0.89</td>
<td>0.33</td>
</tr>
<tr>
<td>2</td>
<td>0.98</td>
<td>0.51</td>
</tr>
<tr>
<td>-5</td>
<td>0.92</td>
<td>0.31</td>
</tr>
<tr>
<td>10</td>
<td>0.85</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Table 7.3: $\beta$ is the shape parameters. Sample size $n = 500$. Level of significance $\alpha = 0.05$.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th><code>hypCvMtestSpline()</code></th>
<th><code>hypMoranTest()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>standHyperbFit()</code></td>
<td><code>hyperbFit()</code></td>
</tr>
<tr>
<td></td>
<td><code>standHyperbFit()</code></td>
<td><code>hyperbFit()</code></td>
</tr>
<tr>
<td>2</td>
<td>0.98</td>
<td>0.04</td>
</tr>
<tr>
<td>-5</td>
<td>0.92</td>
<td>0.23</td>
</tr>
<tr>
<td>10</td>
<td>0.85</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Table 7.4: $\beta$ is the shape parameters. Sample size $n = 500$. Level of significance $\alpha = 0.01$.

As can be seen from Tables 7.3 and 7.4, for a given test method and comparing between two fitting approaches the power of the Cramér-von Mises test is significantly higher than the Moran test.

### 7.5.2 Goodness-of-fit test using real financial data

The following code was used to conduct a goodness-of-fit test using real return on Novartis stock. The code can be used to conduct the same test for return on stock of Nestle and the Swiss SMI index.
As can be seen, $p$-values calculated by `hypCvMtestSpline()` using the fitted parameters returned by `standHyperbFit()` and `hyperbFit()` are $7.9 \times 10^{-5}$ and 0.0025 respectively. This means that the null hypothesis is rejected at $\alpha = 0.05$ or even $\alpha = 0.01$. However, for the same fitted parameters the Moran test indicated that the null hypothesis is not rejected.

### 7.6 Conclusion

This chapter examines two goodness-of-fit tests namely the Cramér-von Mises and Moran log-spacing test. The purpose is to obtain a reliable and fast routine to conduct goodness-of-fit for the hyperbolic distribution with all four parameters un-
known. Specifically, the computational speed of the Cramér-von Mises test proposed in Puig and Stephens (2001) needs improving by fitting precalculated value of the Cramér-von Mises test statistic and then it is used to predict the $p$-value of goodness-of-fit. The Moran test was also considered because of its simplicity. By applying the logit transformation and the adaptive polynomial spline regression fitting, a reliable and fast goodness-of-fit based on Cramér-von Mises test statistic was obtained. However, after being tested extensively, it is clear that the power of the Moran test is not as good as the power of the Cramér-von Mises test. This is as expected because the latter has been known to have strong power when it is used with heavy-tailed data such as the HYP.
Chapter 8

Conclusions and future research

Each of Chapters 2–6 of this thesis proposes our solution to the significant problems concerning the GH and the GIG distributions. They are all new at the time this thesis is rewritten. All of these R routines were clearly documented and extensively tested.

Chapter 2

This chapter is motivated by the fact that the presence of the modified Bessel function of the second kind is well-known for being, among others, a major challenge for the tractability and applicability of the family of GH and GIG distributions.

The underlying idea to deal with this problem is that if the modified Bessel function can be accurately approximated with a definite number of terms then it can be replaced by a summation of elementary functions. This means that mathematical and computational issues which are due to the presence of such a special function can be dealt with in a much simpler manner. Among a numbers of approaches which were investigated, asymptotic approximations appeared to be most suited. It is well-known that an asymptotic series of this Bessel function is a divergent series, but its approximation error is bounded if the argument is sufficiently large. This means that the sufficiently large value of $z$ for a given $\lambda$ must be calculated, if this
method is of any use. However, answers to the following are still uncertain

- For a given value of $\lambda$ what value of $z$ should be considered as large?

- For a given value of $\lambda$ and sufficiently large $z$, what are the definite number of terms required for the approximation to give a specific approximation error?

(even after a latest and comprehensive reference such as \cite{Olver et al. (2010)} was consulted.)

The results of this chapter give accurate answers to those questions for $|\lambda| \in [0, 90]$. The limitation of this approach is its accuracy decreases when $\lambda$ increases. This is because while $K_\lambda(z)$ is an increasing function of $\lambda$ and a decreasing function of $z$, the approach proposed here is restricted by computational capabilities (minimum and maximum floating point numbers) of the system which was used to test the function \texttt{besselKseriesAppro}(z, \lambda).

It is noted that the approximation of $K_\lambda(z)$ has also been implemented as the function \texttt{besselK.nuAsym}(z, \lambda) in package \texttt{Bessel} using Debye polynomials \cite{Abramowitz and Stegun (1970)}. This approximation is also to be used with sufficiently large values of $z$, but it appears that the author provides no quantification of what value of $z$ should be considered as large.

The advantage of using asymptotic approximations over the Debye polynomials approximation is that it is much simpler if calculus operations are applied. For example, the use of an asymptotic series with a definite number of terms in the densities function of the GH family of distribution offers significant mathematical advantage because those densities involve a ratio of the Bessel functions. More importantly, Debye polynomials approximation is only applicable for $\lambda > 0$. For completeness, a method to approximate the Bessel function $K_1(z)$ when $z$ is small was also implemented.

This research opens the door to using today’s computing power to explore possible useful features of the asymptotic series of other special functions. Specifically,
Whittaker (1903) pointed out the relationship between the Whittaker function $W_{k,m}$ with different types of Bessel functions, the error and gamma function. He also derived the asymptotic approximation series of $W_{k,m}$. It is possible that this chapter's approach be applied to the approximation series of $W_{k,m}$. This means approximation of the related special functions can also be obtained with accuracy and definite number of terms.

**Chapters 3 and 4**

This chapter is motivated by the following well-known problems in fitting the univariate hyperbolic and GIG to the data using a numerical optimization routine:

- it is hard to find “good starting values” for the routine to converge, and
- the log-likelihood functions are flat.

The co-existence of these problems presents a challenging issue in fitting these distributions to the data with the focus on small sample sizes. Chapter 3 presents the attempt to deal with these problems when sample size $n \leq 50$.

A symbolic method to the problem of obtaining maximum likelihood estimates of these distributions is sensible. Because numeric optimization routines require starting values and it was stated in Barndorff-Nielsen and Blæsild (1981) that several estimated parameter values can equally describe the same log-likelihood function. Thus, this chapter presents a symbolic method to obtain maximum likelihood estimates of the shape parameter of the HYP and three subclasses of the GIG with $\lambda$ known. The major strength of the symbolic method is it does not depend on starting values. Consequently, the symbolic routines are stable even for sample size of $n = 20$ observations.

When the symbolic method is used for estimation the shape parameter of the HYP, its applicability is limited by a set of assumptions. In order to relax these assumptions, numeric optimization methods are applied to find mle of the pseudo
likelihood of the standardized hyperbolic distribution. Numerical studies show that the method is stable and significantly faster than the numeric optimization of the non-standardized hyperbolic distribution. However, the fitting of the SHYP to the data critically depends on starting values.

Finding ways to improve the computational speed of the EM algorithm has been a very active research issue. Chapter 4 applies the results of Chapter 3 to propose an EM based multivariate fitting method, in which both quantities of the objective function are maximized symbolically. The benefit of this method is stable and significant improvement in computational speed of the EM and its variant algorithm. Note that both symbolic and numeric EM based fitting methods require starting values to initialize an optimization process.

The results of Chapter 3 and 4 have important implications for future research because even though symbolic methods for maximum likelihood estimate have been proposed in the literature, the approach has never been applied to the family of the GH and GIG distributions. These Chapters’ method should be applied to other subclasses of the GH and the GIG because

1. the symbolic method eliminates the use of starting values. This is very appealing in fitting these distributions because their log-likelihood functions are flat so the use of different stating values by a numerical method is likely to result in different parameter estimations.

2. it is hard to find “good” starting values.

Chapter 3 proposes a numeric fitting method to fit the standardized hyperbolic distribution. The purpose of this method is to reduce the parameter space of the HYP from 4 to 2 dimensions. This allows for the fitting of the shape parameters only, while the location and scale parameters are considered as nuisance parameter. This approach has never been attempted in the literature. It offers a fast and stable fitting algorithm. The method should be applied to other subclasses of the SGH
such as the standardized normal inverse Gaussian distribution.

Section 1.4 presented arguments for the fitting of the subclasses of the GH (i.e., known $\lambda$). However, it is noted that recent R routines, which can fit these distributions with unknown $\lambda$ (i.e., fitting the GH and GIG distributions) have been developed. The symbolic method proposed in Chapter 3 can also be applied to obtain the symbolic method maximum likelihood estimators for the GH and GIG. In these cases, the symbolic fitting method is even more appealing than the numeric approach because optimization with an added dimension is even more difficult which means that the elimination of the use of starting values is more useful. Here, the method is to use the function $\text{besselKseriesApp}(z, \lambda)$ proposed in Chapter 2 to obtain the asymptotic approximations for a range of $\lambda$, say $\lambda \in [0,5]$. The maximum likelihood estimator (MLE) of the GH and GIG are then derived using the approximate log-likelihood function with the value of $\lambda$ assumed to be unknown and is within the $[0,5]$ interval. The limitation of the symbolic method is the use of the approximate log-likelihood functions.

Chapters 5

The evaluation of the incomplete Bessel function has been considered as a challenging topic both mathematically and computationally. This chapter evaluates 3 integral representations of the incomplete Bessel function using both analytical and numerical methods. The former results in closed form formulae for the upper and lower incomplete Bessel function (5.1) for $\lambda = j + \frac{1}{2}$ and the formulae in terms of the error functions to evaluate the integral (5.24) in the intervals $(0, x)$ and $(x, \infty)$ where $x > 0$. Note that Maple 15 (which is the latest version) is not able to evaluate it symbolically neither. The analytical formulae to evaluate (5.24) has important implications for future research because the incomplete Bessel function given by (5.2) is just a special case of the integral (5.24) when $a = z^2$, $b = \frac{1}{4}$ and $\lambda = \pm \frac{1}{2}$. This means that the evaluation of the type of integral in (5.2) with $\lambda = \pm \frac{1}{2}$ but $a$
and $b$ assume other positive values can be obtained straightforwardly. A specific reference regarding to the analytical results of this chapter cannot be located, even after a very recent and comprehensive reference such as Olver et al. (2010) was consulted. The numerical accuracy of the obtained formulae was tested using Maple and Mathematica. The results of these tests showed that these formulae were correctly derived and they are highly accurate.

This chapter also implements an R routine by Slevensky and Safouhi to approximate the incomplete Bessel function given by (5.3). Here the purpose is to have a routine to evaluate the incomplete Bessel function when $\lambda$ assume a wider value range. However, this approach has computational problems which was not mentioned by the authors. This chapter discusses when these problems can occur and provides a diagnostic tool for detecting them.

This chapter has important implications for future research. Firstly, it shows that the evaluation of the Bessel function of the second kind and the closely related integral (5.24) can be obtained by the mean of deriving analytical formulae. This approach has never been attempted in the literature. Coupled with the fact that the analytical method offer highly accurate evaluation result, future research effort should be spent on obtaining formulae to evaluate (5.2) when $\lambda \neq \pm \frac{1}{2}$. A possible approach is to establish the relationship between the integrals given by (5.1) and (5.2) in the finite interval. The hope is that by using this relationship and the obtained closed-formed formulae to evaluate the former, formulae to evaluate the latter for other values of $\lambda$ can also be obtained.

Chapters 6

By applying the results of Chapter 5, this chapters derives formulae for the CDF and CCDF of the GIG when $\lambda = \pm \frac{1}{2}$. The numerical routine was also implemented to approximate tail probabilities of the GIG for $\lambda \in \mathbb{R}$. A series of numerical examinations by using highly accurate evaluation by Maple show that for $\lambda = \pm \frac{1}{2}$
the analytical method is accurate for all values of the tested parameters but the numerical method become inaccurate when parameter values were large.

Chapter 7

Chapter 7 obtains a reliable and fast routine to conduct goodness-of-fit based on Cramér-von Mises test. However, the problem with the Cramér-von Mises test is that it depends on the estimated value and the number of unknown parameters (i.e., different table of critical values for different numbers of unknown parameters). The parametric bootstrap-based goodness-of-fit test does not have this limitation, but its computational speed is very likely to be an issue. To explore the strength and address the possible problem of the parametric bootstrap based goodness-of-fit test, a possible research direction could be one which precalculates the values of the test statistic for the HYP using parametric bootstrap methodology and uses this chapter’s approach to obtain a spline regression-fitting object.
Appendix A

Main functions of Chapter 2

A.1 Important testing information

The accuracy of the function \texttt{besselKseriesAppro}(z, \lambda) is compared with those of the function \texttt{besselK.nuAsym}(z, \lambda) in the package \texttt{Bessel} whose extracted details are

\begin{verbatim}
Package: Bessel
Version: 0.5-3
Date: 2009-12-22
Title: Bessel -- Bessel Functions Computations and Approximations
Maintainer: Martin Maechler <maechler@stat.math.ethz.ch>
Date/Publication: 2011-01-15 16:36:10
\end{verbatim}

For \( z \geq 705 \) the function \texttt{besselKseriesAppro}(z, \lambda) automatically returns approximate value in exponential scale. This means that the argument \texttt{expon.scale} of the function \texttt{besselK.nuAsym}(z, \lambda) should be \texttt{TRUE} for the results to be comparable. If \( z < 705 \) then this argument should be \texttt{FALSE}.
Example

# For z > 705
> z <- 710
> lambda <- 10
> besselKseriesAppro(z, lambda, details = TRUE, digits = 14)
Approximated value is on expon.scale
Approximated value : 0.05045642397811
Approximation error : 7.632783294298e-17
Number of terms required : 7
> besselK.nuAsym(x = z, nu = lambda, 4, expon.scale = TRUE)
[1] 0.05045642397811
> besselK(z, lambda, expon.scale = TRUE) # "Exact value"
[1] 0.05045642397811
# For z < 705
> z <- 90
> lambda <- 5
> besselKseriesAppro(z, lambda, details = TRUE, digits = 14)
Approximated value : 1.2411034311593e-40
Approximation error : 8.8817841970013e-16
Number of terms required : 8
> besselK.nuAsym(x = z, nu = lambda, 4, expon.scale = FALSE)
[1] 1.2411034312009e-40
> besselK(z, lambda, expon.scale = FALSE) # "Exact value"
[1] 1.2411034311593e-40
A.2 Function to identify sufficient large argument value of $K_\lambda(z)$

Description
This function identifies if argument $z$ of $K_\lambda(z)$, $\lambda \in \mathbb{R}$, is sufficiently large for the given value of $\lambda$. It returns 1 if $z$ is sufficiently large and 0 otherwise. It is called by the function $\text{besselKseriesAppro}(z, \lambda)$ to obtain asymptotic approximations of $K_\lambda(z)$ for $|\lambda| \in [0, 90]$.

Usage

largeVal(z, lambda)

Arguments

z Value of argument $z > 0$.
lambda Value of order $\lambda \in \mathbb{R}$. 
A.3 Function to approximate $K_\lambda(z)$ given $z$ is sufficiently large

Description
This function approximates $K_\lambda(z)$ using asymptotic approximation if $|\lambda| \in [0, 90]$. It calls the function largeVal(z, lambda). If largeVal(z, lambda) returns 0 then besselKseriesAppro(z, lambda) approximates the sufficient large value $z'$ and prompts the user to enter $z \geq z'$.

Usage

besselKseriesAppro(z, lambda, details = TRUE, digits = 10)

Arguments

z Value of the argument $z > 0$. When $z \geq 705$ the calculation is automatically on the exponentiate scale. In other words, the returned approximate value is $K_\lambda(z) e^z$. It returns $K_\lambda(z)$ when $z < 705$.

lambda Value of the order $|\lambda| \in [0, 90]$.

details Logical. Prints the number of terms required, approximation errors and approximated value if TRUE. Otherwise prints approximated value only.

digits Controls the number of digits when printing the evaluation result.

References
Sections 2.3 and references therein. This function will be added to the package GeneralizedHyperbolic in due course.

Example

> z <- 10
> lambda <- 5/2 # Evaluation when lambda is half of an odd integer
> besselKseriesAppro(z, lambda, details = TRUE, digits = 14)

Exact evaluation : 2.3931325864628e-05
Number of terms : 3
A.4 Function to approximate $K_1(z)$ when $z$ is small

Description
This function approximates $K_1(z)$ when $z$ is small.

Usage

\[
\text{smallValAppro}(z, \lambda, \text{digits} = 15)
\]

Arguments

- \(z\) Value of argument \(z\).
- \(\lambda\) Value of order \(\lambda = 1\).
- \(\text{digits}\) Controls the number of digits when printing the evaluation result.

References
This function is the implementation of the algorithm described in Harris (2009). It will be added to the \texttt{GeneralizedHyperbolic} in due course.

Example

```r
# Differences between besselK() and approximation values
zRanges <- seq(0.01, 20, length.out = 50)
sapply(zRanges, function(x.)
    besselK(x = x., nu = lambda) - smallValAppro(z = x., lambda = 1))
```
Appendix B

Main functions of Chapter 3

B.1 Notes on the Q–Q plots of the hyperbolic distribution

Figure B.1 shows histograms and Q–Q plots of two sets of randomly generated hyperbolic data. In each row, the parameter values (param) used for generating data (histogram) and calculating theoretical quantiles (Q–Q plot) for the hyperbolic distributions are identical.

Theoretically, because the sample and theoretical quantiles are both hyperbolic distribution with the same parameters. The points in the Q–Q plots should be on the straight line with slope 1. However, as can be seen this is merely the case for the more symmetrical data. The Q–Q plot of the skew data wrongly suggests the sample and theoretical quantiles are different.

This example indicates that the hyperbolic Q–Q plot may not be reliable for judging the conformity of the sample and theoretical quantiles for skewed data.
B.2 Important testing information

B.2.1 Relating to the fitting functions in Sections 3.3

This appendix details important technical information which plays critical role in the accuracy and stability of the function \texttt{symFitShapeHyp()}, which is documented in Appendix B.3. The function \texttt{symFitShapeHyp()} is going to be added to the package \texttt{HyperbolicDist} whose extracted details are:

<table>
<thead>
<tr>
<th>Package:</th>
<th>HyperbolicDist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version:</td>
<td>0.6-2</td>
</tr>
<tr>
<td>Date:</td>
<td>2009-09-09</td>
</tr>
<tr>
<td>Title:</td>
<td>The hyperbolic distribution</td>
</tr>
</tbody>
</table>
For plotting the parameter estimates, `symFitShapeHyp()` calls, among others, the
density (`dhyperb()`) and quantile (`qhyperb()`) function of the hyperbolic distri-
bution. These functions and all other functions of package `HyperbolicDist` use

**Parameterization 1** in the following parameterization settings

1. $\pi, \zeta, \delta, \mu$
2. $\alpha, \beta, \delta, \mu$
3. $\phi, \gamma, \delta, \mu$
4. $\xi, \chi, \delta, \mu$

where the numbers from 1–4 indicate the parameterization numbers. Detailed inform-
ation about the parameterizations of `HyperbolicDist` package can be accessed
using

```r
require(HyperbolicDist)
?hyperbolicChangePars
```

**B.2.2 Relating to the fitting function in Section 3.4**

The fitting function in this Section is `standHyperbFit()`. It is documented in Ap-
pendix B.3.1. This fitting function follows the parameterization setting of the pack-
age `GeneralizedHyperbolic` whose extracted details are

```r
Package: GeneralizedHyperbolic
Version: 0.4-1
Date: 2010-09-10
Title: The generalized hyperbolic distribution
Maintainer: David Scott <d.scott@auckland.ac.nz>
Date/Publication: 2009-09-23 16:45:00
```
For plotting the parameter estimates, `standHyperbFit()` also calls the density `dhyperb()` and quantile function `qhyperb()` of the hyperbolic distribution. However, `dhyperb()`, `qhyperb()` and all other functions of package `GeneralizedHyperbolic` work under **Parameterization 2** of this package which is

1. mu, delta, pi, zeta
2. mu, delta, alpha, beta
3. mu, delta, phi, gamma
4. mu, delta, xi, chi

The numbers from 1–4 indicate the parameterization numbers. More information can be obtained using

```
require(GeneralizedHyperbolic)
?hyperbolicChangePars
```

### B.2.3 Relating to the fitting functions in Sections 3.5

All function relating to the GIG of package `HyperbolicDist` work under **Parameterization 1** of the following parameterization setting

1. lambda, chi, psi
2. lambda, delta, gamma
3. lambda, alpha, beta
4. lambda, omega, eta

where the numbers from 1–4 indicate the parameterization numbers. Following this convention, the function of Section 3.5, which is described in Appendix B.5 also works under the Parameterization 1.

All functions of the GIG distribution in package `GeneralizedHyperbolic` work under **Parameterization 1** of the following parameterization setting

1. chi, psi, lambda
2. delta, gamma, lambda
3. alpha, beta, lambda
4. omega, eta, lambda

where the numbers from 1–4 indicate the parameterization numbers.

### B.2.4 Critical effects of different parameterization conventions

As can be seen from Appendices B.2.1 and B.2.2, the parameterizations of the HYP that \textit{symFitShapeHyp()} works under (i.e., Parameterization 1 of \textit{HyperbolicDist}) is different with the parameterization of the HYP that \textit{standHyperbFit()} belongs to (i.e., Parameterization 2 of package \textit{GeneralizedHyperbolic}). Likewise, the order of Parameterization 1 of the GIG in \textit{HyperbolicDist} is different from the order of Parameterization 1 of the GIG in package \textit{GeneralizedHyperbolic}.

Once the functions \textit{symFitShapeHyp()}, \textit{standHyperbFit()} and \textit{symFitSubGIG()} are added to the corresponding packages in due course, the differences in parameterizations used by them will not be an issue. At this stage, however, if both packages are loaded into R memory to provide the required functions for the plotting results by \textit{symFitShapeHyp()}, \textit{standHyperbFit()} and \textit{symFitSubGIG()} then differences in parameterization used by the required functions cause \textit{symFitShapeHyp()}, \textit{symFitSubGIG()} and \textit{standHyperbFit()} to crash.

To solve this temporary issue, the functions of the package \textit{HyperbolicDist}, which are required by \textit{symFitShapeHyp()} and \textit{symFitSubGIG()}, were copied and pasted to create a text file called “Functions of HyperbolicDist Package.R”. Letter “H” was then added as a suffix to the name of each of the functions on this file. For example, the function \textit{dhyperb()} of package \textit{HyperbolicDist} appears on this text file with the name \textit{dhyperbH()}. All other technical details of \textit{dhyperb()} remain unaltered. Similarly, “GH” was added as an suffix to the name of each of
the required functions of package GeneralizedHyperbolic. These functions were stored in a text file called “Functions of GeneralizedHyperbolic Package”. For example, the function `rhyperb()` appears as `rhyperGH()` while all other technical details of its remain unchanged. Once the needed functions, which use different system of parameterizations, were marked then the remaining task for `symFitShapeHyp()` and `symFitSubGIG()` is just call the functions with letter “H” only. Similarly, the function `standHyperbFit()` only calls function with suffix “GH”. By marking the functions belong to two system of parameterizations, they can be loaded into the memory of R and called at the same time without interfering with one another.

B.2.5 Intended sample sizes for testing

Symbolic fitting functions

The purpose for deriving the symbolic maximum likelihood estimate routines to fit the HYP and 3 subclasses of the GIG is to obtain stable and accurate routines to fit these distributions when sample size is less than $n < 100$ by avoiding the use of starting values. Appendices B.3.3 and B.3.3 show `symFitShapeHyp()` and `symFitSubGIG()` work stably even when $n = 30$ but the numeric method failed. Consequently, it is recommended that the sample size of $n < 100$ observations should be used to compare the performance between the symbolic and numeric methods.

Numeric fitting of the standard hyperbolic distribution

This method finds maximum likelihood of the shape parameters of the HYP using a numeric optimization process and the so-called “pseudo likelihood” function. It performance depends critically on the starting values so it is recommended that the sample size of $n > 500$. 

201
B.3 Symbolic maximum-likelihood of the shape parameters of the hyperbolic distribution

Description
This function symbolically obtains maximum-likelihood estimates of the shape parameters of the hyperbolic distribution (3.4), which is an approximate log-likelihood function. The approximation error is less than $6.76 \times 10^{-16}$.

Usage

```
symFitShapeHyp(x, delta, mu, printOut = FALSE, plots = FALSE)
```

Arguments

- `x` Hyperbolic data vector with the number of observations less than 100.
- `delta, mu` Known values of $\delta$ and $\mu$.
- `plots` Logical. If TRUE prints histogram, log-histogram, Q-Q plot and P-P plot.
- `printOut` Logical. If TRUE, prints estimates of the parameters, maximum likelihood values, first and second derivative tests.

References
Sections 3.3 and references therein. The hyperbolic distribution fitting functions in package `HyperbolicDist`.

This function returns the vector of estimated parameters as $(\hat{\rho}, \hat{\zeta}, \delta, \mu)'$ to be consistent with Parameterization 1 of this package. Note that the parameters under this parameterization were named as $(\pi, \zeta, \delta, \mu)'$. This means that $\rho$ and $\pi$ plays the role of the same shape parameter in the density function of the HYP. The reason for this discrepancy is that if $\pi$ was used to denote this parameter in Chapter 3 then it would be confused with the mathematical constant $\pi$ in (3.4). The $\rho$ notation will be changed to $\pi$ when `symFitShapeHyp()` is added to this package in due course. See Appendix B.2.1 for more information.
Example

# Example
hypParas <- c(5, 23, 1, 5)
data <- rhyperbH(50, hypParas)
par(mfrow = c(2, 2))
symFitShapeHyp(data, delta = hypParas[3], mu = hypParas[4],
               plots = TRUE, printOut = FALSE)
B.3.1 Numeric maximum-likelihood of the shape parameters of the hyperbolic distribution

Description
This function numerically obtains maximum-likelihood estimates of the shape parameters of the hyperbolic distribution (3.4) assuming scale and location parameters known and $\zeta \geq 22$. It serves the purpose of comparing computational stability between numerical and symbolic method in Appendix B.3.

Usage

nume.shapeHypFit(x, delta = Theta[3], mu = Theta[4],
    startValues = c(1,3), optMethods = c("BFGS", "Nelder-Mead", "nlm"))

Arguments

x Hyperbolic data set.
delta, mu Known value of $\delta$ and $\mu$.
startValues Starting values of $\rho$ and $\zeta$ for the optimization process.
optMethods Either "BFGS" or "Nelder-Mead" of optim () or nlm().

References
Sections 3.3, references therein and the fitting functions of package HyperbolicDist.

Example

hypParas <- c(3, 23, 1, 20)
hypdat <- rhyperbH(50, hypParas)
nume.shapeHypFit(hypdat, delta = hypParas[3], mu = hypParas[4],
    optMethods = "BFGS")
B.3.2 Finding starting values for the numerical fitting process using method of moments

Description
This function finds starting values for input to a maximum likelihood routine for the function `nume.shapeHypFit()`. It returns the estimates $\hat{\rho}$ and $\hat{\zeta}$ by minimizing the squared of the differences between sample and theoretical moments. It appears that this function does not return "good" starting values for the function when sample size $n < 100$.

Usage

```r
shapeHypFitStart(x, delta, mu,
                   startOpt = c("BFGS", "Nelder-Mead", "nlm"))
```

Arguments

- `x` Hyperbolic data set.
- `delta, mu` Known value of $\delta$ and $\mu$.
- `startOptMethods` Either "BFGS" or "Nelder-Mead" of `optim()` or `nlm()`.

References

Function `hyperbFitStart()` in package `HyperbolicDist`.

Example

```r
# Works well for a huge sample size
paraHyp <- c(9, 23, 1, 20)
hypdat <- rhyperbH(10000, paraHyp)
shapeHypFitStart(hypdat, delta = paraHyp[3], mu = paraHyp[4],
                  startOptMethods = "BFGS")
```
B.3.3 Comparing numeric with symbolic maximum-likelihood estimation of the shape parameters of the hyperbolic distribution

Description

Compares the function `symFitShapeHyp()` and `nume.shapeHypFit()` in terms of computational stability when different starting values are used by the latter. This function provides additional numerical tests for Section 3.3.4.

Usage

```r
testShapeFitFuncs(ntests, n, Theta, startVal = c("RAN", "MoM"),
                 startOpt = "BFGS", method = "BFGS", printSum = FALSE, warn = "TRUE")
```

Arguments

- **ntests**: Number of tests to be done. Default is 1000.
- **n**: Sample size used for maximum-likelihood estimation and obtaining starting value by the method of moment.
- **Theta**: Vector of the parameter values for generating of the hyperbolic data. See Section B.2.1 for information on the parameterization to be used.
- **startVal**: Method for obtaining the starting values for `nume.shapeHypFit()`. If "RAN" then the starting values are randomly chosen from a vector of 10 values from 1 to 10. If "MoM", the function `shapeHypFitStart()` is called to obtain the method of moments estimates as starting values for the method of maximum likelihood.
- **startOpt**: Only applies when `startVal = "MoM"` is TRUE. This argument specifies the optimization method used to find starting values by the function `shapeHypFitStart()`. It can be either "BFGS" or "Nelder-Mead" of `optim()` or `nlm()`.

References

206
Section 3.3 and references therein.

**Example**

\[
\begin{align*}
\text{rho} & \leftarrow \text{rnorm}(1, \text{mean} = -5, \text{sd} = 6) \\
\text{zeta} & \leftarrow \text{runif}(1, 22, 30) \\
\text{delta} & \leftarrow \text{runif}(1, 0, 1) \\
\text{mu} & \leftarrow \text{rnorm}(1, \text{mean} = 1, \text{sd} = 3)
\end{align*}
\]
\[
\text{ThetaHyp} \leftarrow c(\text{rho}, \text{zeta}, \text{delta}, \text{mu})
\]
\[
\text{shapeHypFitFuncs.test(ntests = 1000, n = 30, Theta = ThetaHyp,}
\text{startVal = "MoM", startOpt = "BFGS", printSum = TRUE})
\]

The above code specifies that 1000 data sets of size \( n = 30 \) were generated using a parameter vector obtained randomly. Note \text{startVal = "MoM"} indicates the starting values for the numeric maximum-likelihood function \text{nume.shapeHypFit()} are the method of moments. The optimization method used by \text{shapeHypFitStart()} is "BFGS".

\[
\text{\$rhoHatDiff}
\begin{array}{|c|c|c|c|}
\hline
\text{SYM} & \text{BFGS} & \text{N-M} & \text{nlm} \\
\hline
\text{Min.} & -0.42 & -0.98 & -0.37 & -32.60 \\
\text{1st Qu.} & -0.09 & -0.10 & -0.10 & -0.06 \\
\text{Median} & -0.01 & -0.01 & -0.01 & 0.10 \\
\text{Mean} & -0.03 & -0.29 & -0.01 & 2.88 \\
\text{3rd Qu.} & 0.07 & 0.06 & 0.07 & 7.35 \\
\text{Max.} & 0.44 & 0.42 & 0.42 & 14.51 \\
\text{NA's} & 400 & 400 & & \\
\hline
\end{array}
\]

\[
\text{\$zetaHatDiff}
\begin{array}{|c|c|c|c|}
\hline
\text{SYM} & \text{BFGS} & \text{N-M} & \text{nlm} \\
\hline
\text{Min.} & -12.0 & -24.9 & -23.5 & -25.0 \\
\text{1st Qu.} & -2.4 & -2.5 & -2.5 & -1.5 \\
\text{Median} & 2.2 & 2.4 & 2.3 & 10.4 \\
\text{Mean} & 3.5 & 3.8 & 3.8 & 3427.6 \\
\text{3rd Qu.} & 7.9 & 8.2 & 8.3 & 2956.0 \\
\text{Max.} & 59.5 & 59.5 & 59.6 & 22001.5 \\
\text{NA's} & 400.0 & 400.0 & & \\
\hline
\end{array}
\]
In the above output, \( \$\rhoHatDiff \) and \( \$\zetaHatDiff \) represent two components of the list which contain 5 numbers summary of \( \hat{\epsilon} = \hat{\pi} - \pi \) and \( \hat{\kappa} = \hat{\zeta} - \zeta \) respectively. The estimates were obtained by using `symFitShapeHyp()` and `nume.shapeHypFit()`.

\[
\text{SYM} \quad \text{The estimates were obtained by the symbolic function.}
\]
\[
\text{BFGS} \quad \text{The estimates were obtained by `optim(..., method = "BFGS")`.}
\]
\[
\text{N-M} \quad \text{The estimates were obtained by `optim(..., method = "Nelder-Mead")`.}
\]
\[
\text{nlm} \quad \text{The estimates were obtained by `nlm()`.}
\]
\[
\text{NA’s} \quad \text{Failures to converge}
\]

As can be seen, the distribution of the differences between the estimate of a parameter and its true value which are given by \( \text{SYM} \), \( \text{BFGS} \) and \( \text{Nelder-Mead} \) are fairly similar when the numeric method were supplied with “good” starting values. This is as expected. However, for these tested parameters these two optimization methods failed 400 times because of not obtaining “good” starting values from `shapeHypFitStart()` (using the method of moments). Neither the symbolic algorithm nor `nlm()` failed even the latter used the same starting values as the \( \text{BFGS} \) and the \( \text{Nelder-Mead} \) method of `optim()`. However, the distribution of \( \hat{\pi} - \pi \) and \( \hat{\zeta} - \zeta \) obtained by `nlm()` is evidently more spread out. This reaffirms that the numeric fitting method is very susceptible to changes in starting values and that finding “good” starting values is difficult even for maximum-likelihood estimation of two unknown parameters.
B.4 Maximum-likelihood of the standardized hyperbolic distribution

Description
Numerically obtains maximum-likelihood estimates of the shape parameters of the SHYP. This function allows for the relaxation of the assumptions imposed on the function in Appendix B.3. Estimates of the location and scale parameters are given by the function hyperbFitStart(..., method = "BN")

Usage
standHyperbFit(x, optMethods = c("Nelder-Mead", "BFGS", "nlm"),
               printOut = FALSE, plots = FALSE)

Arguments

x Hyperbolic data set.

optMethods Fitting methods. Either "Nelder-Mead" or "BFGS" of optim() or nlm().

printOut Logical. If TRUE prints fitting outputs.

plots Logical. If TRUE prints histogram, log-histogram, Q-Q plot and P-P plot.

References
The hyperbolic distribution fitting functions in package GeneralizedHyperbolic and Scott et al. (2009). This function returns the vector of the estimates in the order \((\mu, \delta, \rho, \zeta)'\) to be consistent with the parameter setting of Parameterization 1 of this package. See Appendix B.2.2. However, the parameters under this parameterization were named as \((\mu, \delta, \pi, \zeta)'\). This means that \(\rho\) and \(\pi\) plays the role of the same shape parameter in the density function of the HYP. The reason for this discrepancy is that if \(\pi\) was used to denote this parameter in Section 3.4.1 then it would be confused with the mathematical constant \(\pi\).
hyperbFit() in package GeneralizedHyperbolic fits the non-standardized hyperbolic distribution and returns the vector of the estimates under Parameterization 2 (i.e., \((\mu, \delta, \alpha, \beta)\)). The reason for the discrepancy between the parameterizations returned by hyperbFit() and standHyperbFit() is that the latter follows the parameterization used by [Scott et al. (2009)] to obtain the SHYP in Section 3.4.1. This temporary difference can be straightforwardly solved using the function hyperbolicChangePars() to from Parameterization 1 to 2 and vice versa. The use of \(\rho\) notation will be changed to \(\pi\) when symFitShapeHyp() and the parameter estimates output by standHyperbFit() will make to be consistent with those used by hyperbFit() when it is added to the package GeneralizedHyperbolic in due course.

**Example**

```r
c paramVals <- c(1, 20, 5, 3.09)  # In Parameterization 2
# Use paramVals in Parameterization 2 to generate HYP data
hypData <- rhyperbGH(500, param = paramVals)
# Fit hypData and produce plots
par(mfrow = c(2, 2))
fittedParam <- standHyperbFit(hypData, optMethods = "nlm",
                           plots = TRUE, printOut = TRUE)
# Change fitted parameters from Parameterization 1 to 2
hyperbChangeParsGH(1, 2, fittedParam$Theta)
```

# Fitting real data
```
require(QRMlib)  # Load QRMlib package
data(DJ)
Ret.DJ <- mk.returns(DJ)
window1.start <- timeDate("01/01/1993", format="%m/%d/%Y")
window1.end  <- timeDate("12/31/2000",format="%m/%d/%Y")
sample1 <- (seriesPositions(Ret.DJ) > window1.start & seriesPositions(Ret.DJ) < window1.end)
```
DJ30daily <- Ret.DJ[sample1,]
DJ30daily <- 100*seriesData(DJ30daily)
selection1 <- c("AXP","BA","C", "HWP","INTC","JPM","DIS")
stockRet <- DJ30daily[,"BA"]
par(mfrow = c(2, 2))
fittedInPara1 <- standHyperbFit(stockRet, optMethod = "BFGS",
     plots = TRUE, printOut = TRUE)
fittedInPara2 <- hyperbChangeParsGH(1, 2, fittedInPara1$Theta)
fittedInPara2
hyperbFitGH(stockRet, plots = TRUE)
B.5 Symbolic fitting of subclasses of the generalized inverse Gaussian distributions

Description
This function obtains parameter estimates of 3 subclasses of the generalized inverse Gaussian distribution using maximum likelihood. The log-likelihood function is exact.

Usage

```r
symFitSubGIG(x, lambda, plots = TRUE, printOut = FALSE)
```

Arguments

- `x` Data from the subclasses of the generalized inverse Gaussian data with sample size less than 100.
- `lambda` The value of $\lambda$. Either $-\frac{1}{2}$, $\frac{3}{2}$ or $\frac{5}{2}$. This is the value of $\lambda$ which is used to generate data.
- `plots` Logical. If `TRUE` prints histogram, log-histogram, Q-Q plot and P-P plot.
- `printOut` Logical. If `TRUE` prints fitting outputs.

References
Sections 3.5 and references therein. The fitting functions in package `HyperbolicDist`. The function will be added to this package in due course.

Example

```r
Theta <- c(5/2, 0.05, 30)
# Generate data using the function rgig in HyperbolicDist package.
set.seed(1)
data <- rgigH(30, Theta)
# Fit a sample size of 30 data points
symFitSubGIG(data, lambda = Theta[1], plots = TRUE, printOut = FALSE)
```
Theta <- c(3/2, 5, 25)
set.seed(1)
data <- rgigH(50, Theta)
par(mfrow = c(2, 2))
symFitSubGIG(data, lambda = Theta[1], plots = TRUE,
             printOut = FALSE)
B.5.1 Numeric fitting of the subclasses of the generalized inverse Gaussian distributions

Description
This function numerically obtains maximum-likelihood estimates of the parameters of three subclasses of the GIG. It serves the purpose of comparing computational stability between numerical and symbolic optimization.

Usage

nume.subgigFit(x, lambda, startValues = c(3, 2),
optMethods = c("BFGS", "Nelder-Mead", "nlm"))

Arguments

dat Data from a subclass of the generalized inverse Gaussian specified by lambda.

startValues Starting values of $\chi$ and $\pi$ for the optimization process.

optMethods Either BFGS or Nelder-Mead of optim () or nlm()

References
Sections 3.5 and references therein.

Example

Theta <- c(3/2, 4, 1)
# Generate data using the function rgig in HyperbolicDist package.
gigDat <- rgigH(100, Theta)
nume.subgigFit(gigDat, lambda = Theta[1], optMethods = "BFGS")
B.5.2 Finding start values for the numerical fitting function using method of moments

Description
Finds starting values for input to a maximum likelihood routine for the function `nume.subgigFit()`. This function returns the estimates $\hat{\chi}$ and $\hat{\psi}$ by minimizing the squared of the differences between sample and theoretical moments. It appears that this function does not return “good” starting values for the function `nume.subgigFit` for sample size $n < 300$.

Usage

```r
subgigFitStart(dat, lambda, startOpt = c("BFGS", "Nelder-Mead", "nlm"))
```

Arguments

- `dat`: Data from a subclass of the GIG defined by `lambda`.
- `lambda`: Either $\lambda = -1/2, 3/2$ or $5/2$, which was use to generate `dat`.
- `startOpt`: Either BFGS or Nelder-Mead of `optim()` or `nlm()`.

References
Function `gigFitStart()` in package `GeneralizedHyperbolic`.

Example

```r
Theta <- c(-1/2, 1, 5)
BigGigDat <- rgigH(5000, Theta)
subgigFitStart(BigGigDat, lambda = Theta[1], startOpt = "BFGS")
smallGigDat <- sample(BigGigDat, 300)
subgigFitStart(smallGigDat, lambda = Theta[1], startOpt = "BFGS")
```
B.5.3 Comparing numeric with symbolic maximum-likelihood estimation of the subclasses of the generalized inverse Gaussian distributions

Description

Compares the function `symFitSubGIG()` and `nume.subgigFit()` in terms of computational stability when different starting values are used by the latter. This function provides significantly more extensive tests for Section 3.5.4.

Usage

```r
subgigFitFuncs.test(ntests, n, Theta, startVal = c("RAN", "MoM"),
                      printSum = FALSE, startOpt = c("BFGS", "Nelder-Mead", "nlm"))
```

Arguments

- `ntests` Number of tests to be done. Default is 1000.
- `n` Sample size used for maximum-likelihood estimation and obtaining starting value by the method of moment.
- `Theta` Vector of the parameter values ($\lambda, \chi, \psi$) for generating GIG data.
- `startVal` Method for obtaining the starting values for `nume.shapeHypFit()`. If "RAN", the starting values are randomly chosen from a vector of 10 values from 1 to 10. If "MoM", the function `subgigFitStart()` is called to obtain the method of moments estimates as starting values for the method of maximum likelihood.
- `startOpt` Only applies when `startVal = "MoM"` is TRUE. This argument specifies the optimization method used to find starting values by the function `subgigFitStart()`. It can be either BFGS or Nelder-Mead of `optim()` or `nlm()`.
- `printSum` Logical. Prints 5 numbers summary of the differences between true and estimated value of the parameter.
chi <- rgamma(1, 2)
psi <- rgamma(1, 3)
 ThetaGig <- c(3/2, chi, psi)
subgigFitFuncs.test(ntests = 1000, n = 200, ThetaGig, startVal = "MoM",
startOpt = "BFGS", printSum = TRUE)

In the above output, $\chiHatDiff$ and $\psiHatDiff$ represent two components of
the list which contain 5 numbers statistic of $\hat{\epsilon} = \hat{\chi} - \chi$ and $\hat{\kappa} = \hat{\psi} - \psi$ respectively.
The estimates were obtained by using symFitSubGIG( ) and nume.subgigFit( ).

$\chiHatDiff$

<table>
<thead>
<tr>
<th></th>
<th>SYM</th>
<th>BFGS</th>
<th>N-M</th>
<th>nlm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>-2.30</td>
<td>-4.72</td>
<td>-2.30</td>
<td>-4.72</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>-0.83</td>
<td>-4.72</td>
<td>-0.89</td>
<td>-4.72</td>
</tr>
<tr>
<td>Median</td>
<td>-0.32</td>
<td>-1.35</td>
<td>-0.41</td>
<td>0.13</td>
</tr>
<tr>
<td>Mean</td>
<td>-0.25</td>
<td>-2.31</td>
<td>-0.31</td>
<td>1.75</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.23</td>
<td>-0.32</td>
<td>0.21</td>
<td>3.93</td>
</tr>
<tr>
<td>Max.</td>
<td>3.24</td>
<td>3.24</td>
<td>3.25</td>
<td>241.45</td>
</tr>
</tbody>
</table>

$\psiHatDiff$

<table>
<thead>
<tr>
<th></th>
<th>SYM</th>
<th>BFGSpsiHat</th>
<th>N-M</th>
<th>nlm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>-0.35</td>
<td>-0.73</td>
<td>-0.35</td>
<td>-0.07</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>-0.08</td>
<td>-0.64</td>
<td>-0.09</td>
<td>-0.06</td>
</tr>
<tr>
<td>Median</td>
<td>-0.01</td>
<td>-0.17</td>
<td>-0.01</td>
<td>0.001</td>
</tr>
<tr>
<td>Mean</td>
<td>-0.00</td>
<td>-0.30</td>
<td>-0.01</td>
<td>8.18e+20</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>0.07</td>
<td>-0.00</td>
<td>0.06</td>
<td>3.58e+06</td>
</tr>
<tr>
<td>Max.</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>8.17e+23</td>
</tr>
</tbody>
</table>

SYM The estimates were obtained by the symbolic function.
BFGS The estimates were obtained by optim(..., method = "BFGS").
N-M The estimates were obtained by optim(..., method = "Nelder-Mead").
nlm The estimates were obtained by nlm().
NA’s Failures to converge

217
As can be seen, the distribution of the differences between the estimate of a parameter and its true value which were given by SYM, BFGS and Nelder-Mead are fairly similar when the numeric method were supplied with with “good” starting values. This is as expected. However, for these tested parameters these two numerical optimization methods failed over 400 times because of not getting “good” starting values from subgigFitStart( ) (using the method of moments). Neither the symbolic nor nlm() failed even the latter used the same starting values as the "BFGS" and the "Nelder-Mead" method of optim(). However, the distribution of $\hat{\epsilon}$ and $\hat{\kappa}$ obtained by nlm is evidently significantly more spread out. This reaffirms that the numeric fitting method is very susceptible to changes in starting values and that finding “good” starting values is difficult even for maximum-likelihood estimation of two unknown parameters.
Appendix C

Main functions of Chapter 4

C.1 Important testing information

Performance of the fitting function of this chapter, symFitmNH(), is compared to the performance of the function fit.mNH() in the package QRMlib which has the extract information

Package: QRMlib
Version: 1.4.5
Date: 2010-02-22
Title: Provides R-language code to examine Quantitative Risk Management concepts
Author: Alexander McNeil for S-Plus original; R port by Scott Ulman
Maintainer: R-language version by Scott Ulman <scottulman@hotmail.com>
Description: This is a free R-language library designed to accompany the book Quantitative Risk Management: Concepts, Techniques and Tools.
URL: http://www.ma.hw.ac.uk/~mcneil/book/index.html
Date/Publication: 2010-02-24 08:15:55
C.2 Function to calibrate special cases of the multivariate hyperbolic distributions

Description
This function is an EM based routine which fits MNIG distribution to data of any dimensions, 2-DHYP and 4-DHYP hyperbolic distribution data. Here, the optimization of the mixing distribution (GIG) is carried out symbolically.

Usage

\[
\text{symFitmNH}(x, \text{symmetric} = \text{FALSE}, \text{case} = \text{c("dhyp", "mNIG")}, \\
\text{stpRule} = \text{c("logLik", "mixPara")}, \text{method} = \text{c("EM", "MCECM")}, \\
\text{kvalue} = 1, \text{chiPsiStart} = \text{c(3, 5)}, \text{nit} = 2000, \text{tol} = 1e-10)
\]

Arguments

\text{x} \quad \text{Multivariate hyperbolic data.}

\text{symmetric} \quad \text{Logical. If TRUE the skewness parameter } \gamma \text{ equal zero.}

\text{case} \quad \text{Whether MNIG ("mNIG") or } d\text{-dimensional hyperbolic ("dhyp") should be fitted. The corresponding cases in fit.mNH() are \'NIG\" and \'hyp\" respectively.}

\text{stpRule} \quad \text{Stopping rule for the fitting algorithm. Either changes in log-likelihood (\text{logLik}) or changes in parameter estimates (\text{mixPara}).}

\text{method} \quad \text{Whether the EM or MCECM algorithm should be used.}

\text{kvalue} \quad \text{Value to which to constrain the determinant of dispersion matrix.}

\text{chiPsiStart} \quad \text{Starting value of parameters } \chi \text{ and } \psi \text{ of the mixing distribution.}

\text{Starting value of } \lambda \text{ is set to equal } -\frac{1}{2} \text{ if } \text{case} = \text{"mNIG"} \text{ and } \frac{3}{2} \text{ (2-DHYP) and } \frac{5}{2} \text{ (4-DHYP) if } \text{case} = \text{"dhyp".}

\text{tol} \quad \text{Relative tolerance level to be used as a stopping rule. Default is } tol = 10^{-10} \text{ which is the same with those used by fit.mNH().}
nit  Number of iterations to be used as a stopping rule. Default is 2000 which is the same with those used by \texttt{fit.mNH()}. 

References 
Chapter 4 and references therein. The multivariate fitting functions in package \texttt{ghyp} and \texttt{QRMlib}. This function will be added to the package \texttt{GeneralizedHyperbolic} in due course. 

Example 

# This code shows that parameter estimates of the MNIG 
# returned by \texttt{sym.fitmNH()} and \texttt{fit.mNH()} closely match. 
require(QRMlib) 
Sigma <- diag(c(3, 4, 1, 1, 5)) %*% equicorr(5, 0.6) %*% 
  diag(c(3,4,1,1,5)); 
mu <- c(2, 3, 9, 5, 3); 
gamma <- rep(2, length(mu)) 
Theta <- c(-1/2, 5, 3) 
ghdata <- rmghyp(n = 100, lambda = Theta[1], chi = Theta[2], 
  psi = Theta[3], Sigma, gamma, mu) 
fitOutOne <- sym.fitmNH(ghdata, symmetric = FALSE, case = "mNIG", 
  method = "MCECM", kvalue = 1, chiPsiStart = c(3, 5), 
  nit = 2000, tol = 1e-10) 
fitOutTwo <- fit.mNH(ghdata, case = "NIG") 
fitOutOne 
fitOutTwo
C.3 Parameter values used for the computational speed comparison in Section 4.4

The computational speed comparisons plotted in Figure 4.1 were conducted using the below parameter values.

**Fitting 4 dimensions hyperbolic data**

The following parameters were used to generate 200 data sets of 4-dimensional hyperbolic (i.e., 4-DHYP) data with $n = 100$.

\[
\gamma = (0.1, 0.1, 0.1, 0.1)', \quad \mu = (2, 3, 3, 5)', \quad \Sigma = \begin{pmatrix} 9.0 & 7.2 & 1.8 & 18 \\ 7.2 & 16.0 & 2.4 & 24 \\ 1.8 & 2.4 & 1.0 & 6 \\ 18 & 24 & 6 & 100 \end{pmatrix},
\]

\[
\lambda = 1, \quad \chi = 5, \quad \psi = 30.
\]

**Fitting 2 dimensions multivariate normal inverse Gaussian data**

\[
\gamma = (0.2, 0.2)', \quad \mu = (1, 5)', \quad \Sigma = \begin{pmatrix} 2.0 & 5.6 \\ 5.6 & 16 \end{pmatrix},
\]

\[
\theta = -\frac{1}{2}, \quad \chi = 2, \quad \psi = 10.
\]

**Fitting 5 dimensions multivariate normal inverse Gaussian data**

The following parameters were used to generate 100 data sets of 5 dimensions MNIG data with $n = 30$.

\[
\gamma = (1, 1, 1, 1)', \quad \mu = (2, 3, 3, 5)', \quad \Sigma = \begin{pmatrix} 9.0 & 7.2 & 1.8 & 1.8 & 9 \\ 7.2 & 16.0 & 2.4 & 2.4 & 12 \\ 1.8 & 2.4 & 1.0 & 0.6 & 3 \\ 1.8 & 2.4 & 0.6 & 1.0 & 3 \\ 9.0 & 12.0 & 3.0 & 3.0 & 3.0 \end{pmatrix},
\]

\[
\lambda = -\frac{1}{2}, \quad \chi = 3, \quad \psi = 5.
\]
Fitting 2 dimensions hyperbolic data

The following parameters were used to generate 200 data sets of two dimensions hyperbolic ($\lambda = 1$) data with size $n = 100$.

$$
\gamma = (2, 2)', \quad \mu = (3, 9)', \quad \Sigma = \begin{pmatrix}
1.0 & 3.6 \\
3.6 & 16
\end{pmatrix},
$$

$$
\theta = 1, \quad \chi = 5, \quad \psi = 3.
$$
Appendix D

Main functions of Chapter 5

D.1 Important testing information

This section is to point out that the incomplete Bessel function (5.3) was defined by 2 different integral presentations in the literature.

D.1.1 Form one

A definition due to Harris [2008]

\[ K_{\lambda}(\chi, \psi) = \int_{1}^{\infty} e^{-\chi t - \psi/t} t^{-\lambda-1} dt. \]  

(D.1)

This definition was used to define the incomplete Bessel function (5.3). It was also used by Slevinsky and Safouhi [2010] to derive the algorithm which was coded as the function numeIBF( ).

D.1.2 Form two

A definition due to Terras [1981]

\[ K_{\lambda}(\chi, \psi) = \int_{1}^{\infty} e^{-\chi t - \psi/t} t^{\lambda-1} dt. \]  

(D.2)
D.2 Function to calculate the incomplete Bessel function given by the integral (5.1)

Description
This function evaluates the incomplete Bessel function given by (5.1).

Usage

closedFormIBF(x, z, lambda, lowerBK = FALSE, expon.scaled = FALSE, digits = 20)

Arguments

x The limit of the integration, $x > 0$.

z Value of $z > 0$.

lambda Value of order $\lambda = j + \frac{1}{2}$ where $j = 0, 1, 2, \ldots$

lowerBK Logical. The lower incomplete Bessel function is calculated if TRUE.

expon.scaled Logical. The evaluation is on exponential scale if TRUE.

digits Number of decimal places to print.

References
Section 5.2 and references therein.

Example

z <- 2
x <- 100
lambda <- 115/2

# This gives an incomplete BK value which
# agrees with numerical integration using Maple 15 and Mathematica 6.
closedFormIBF(z, x, lambda, lowerBK = FALSE, expon.scaled = FALSE)
D.3 Maple procedure to check the accuracy of the function in Appendix D.2

Description

This Maple 15 procedure is used to check the accuracy of the function in Appendix D.2.

Usage

```maple
closedFormIBFproc := proc (x::positive, z::positive, lambda::positive,
                      lowerBK::name, tol::posint, exponScaled::name)
local cons, cp, f, lc, IBF, isWholeNumber, s;
  option trace;
  Digits := tol;
  if tol < 20 then error "tol should be greater than 20" end if;
  isWholeNumber:= proc(n::positive)
    if abs(n - round(n)) < Float(1.5, -8) then
      return true;
    else return false;
    end if;
  end proc;
  if exponScaled = FALSE then
    s:= 1;
  elif exponScaled = TRUE then
    s:= exp(z);
  else error "only two values"
  end if;
  lc:= evalf(lambda + 1/2);
  if isWholeNumber(lc) = true then
    cons := sqrt(Pi/(2 * z)) * (1/(GAMMA(lambda + 1/2))) * exp(-z);
  end if;
end proc;
```

226
\[
\begin{align*}
\text{cp} &= \text{evalf(cons)}; \\
f &= \exp(-\xi) \cdot \xi^{(\lambda - 1/2)} \cdot (1 + \xi/(2 \cdot z))^{(\lambda - 1/2)}; \\
\text{if lowerBK} = \text{FALSE} \\
\quad \text{then IBF} &= \text{cp} \cdot \int f \, d\xi \text{ where } \xi = x \text{ to } \infty \\
\text{elif lowerBK} = \text{TRUE} \\
\quad \text{then IBF} &= \text{cp} \cdot \int f \, d\xi \text{ where } \xi = 0 \text{ to } x \\
\text{else error } "\text{lowerBK should be either TRUE or FALSE}" \\
\end{align*}
\]

\[
\text{end if;}
\]

\[
\text{evalf(s \cdot IBF);} \\
\text{else error } "\text{lambda should be half of an odd integer}"; \\
\text{end if;}
\]

\[
\text{end proc;}
\]

closedFormIBFproc(16, 3, 51/2, \text{FALSE}, 50, \text{TRUE})
D.4 Function to calculate the incomplete Bessel function given by the integral (5.2)

Description
This function evaluates the incomplete function given by (5.2).

Usage

\[
\text{halfOrderAIBF}(v, \chi, \psi, \lambda, \text{lowerBK} = \text{FALSE}, \text{digits} = 14)
\]

Arguments

- \textit{v} \quad \text{The limit of the integration, } v > 0. \text{ The value of } z \text{ and } x \text{ in (5.2) is given by } z = (\chi \psi)^{1/2} \text{ and } x = (2v\chi)^{-1/2} \text{ respectively.}
- \textit{chi, psi} \quad \text{Value of } \chi > 0 \text{ and } \psi > 0.
- \textit{lambda} \quad \text{Value of order } \lambda \text{ which currently assumes only } \lambda = \pm \frac{1}{2}.
- \textit{lowerBK} \quad \text{Logical. The lower incomplete Bessel function is calculated if } \text{TRUE.}

References
Section 5.4 and references therein.

Example

\[
\begin{align*}
v & \leftarrow 10 \\
\chi & \leftarrow 3 \\
\psi & \leftarrow 5 \\
\lambda & \leftarrow -1/2 \\
\text{halfOrderAIBF}(v = v, \chi = \chi, \psi = \psi, \lambda = \lambda, \text{lowerBK} = \text{TRUE})
\end{align*}
\]
D.5 Maple procedure to check the accuracy of the function in Appendix D.4

halfOrderAIBFproc := proc (v::positive, chi::positive, psi::positive, lambda::numeric, lowerBK::name, tol::posint)
local cons, fun, cp, IBF;
if tol < 15 then
   error "tol should be at least 20 Digits"
end if;
Digits := tol;
if abs(lambda * 2) <> 1 then
   error "lambda should be +/- a half"
end if;
cons := 1/2 * (chi/psi)^(lambda/2);
cp := evalf(cons);
fun := exp(-1/2 * (chi * omega + psi/omega)) * omega^(lambda - 1);
if lowerBK = TRUE
   then IBF := cp * integrate(fun, omega = 0..v);
elif lowerBK = FALSE
   then IBF := cp * integrate(fun, omega = v..infinity);
else error "lowerBK should be either TRUE or FALSE"
end if;
evalf(IBF);
end proc;

halfOrderAIBFproc(2, 50, 1/2, 1/2, FALSE, 40)
D.6  Function to calculate the incomplete Bessel function given by the integral (5.3)

Description
This function numerically approximates the upper incomplete Bessel function given by (5.3).

Usage
\[
\text{numeIBF}(\chi, \psi, \lambda, \text{traceIBF} = \text{TRUE}, \varepsilon = 0.85, \text{digits} = 15)
\]

Arguments
- \(\chi, \psi\)  Value of \(\chi > 0, \psi > 0\).
- \(\lambda\)  Value of order \(\lambda \in \mathbb{R}\).
- \text{traceIBF}  Logical. If \text{TRUE} then the value of the denominator and nominator of \(G_n\), which are specified by (5.48), for each iteration \(n\) will be output. This option works as a diagnostic tool to detect when this function may encounter numerical problems.
- \(\varepsilon\)  Numeric value between \((0, 1)\) to be used by the stopping rule in (5.49).
- \text{digits}  Number of decimal places to print.

References
Section 5.5 and references therein. This function is already available in the R package \texttt{DistributionUtils} as the function \texttt{incompleteBesselK}(). However, this function does not have argument \texttt{traceIBF}. 
Example

> numeIBF(5, 15, 90, traceIBF = TRUE)

Iteration n  2
G[n] numerator : -3.68596483245604e-101
G[n] denominator : 1.43575091242175e-90
Error : 2.56727319520817e-11

Iteration n  3
G[n] numerator : 4.28744727557045e-98
G[n] denominator : -1.66919630018135e-87
Error : 1.29664592114317e-14

[1] 1.01211897595612e+30
D.7 Maple procedure to check the accuracy of numeIBF()

halfOrderAIBFproc := proc (v::positive, chi::positive, psi::positive, lambda::numeric, lowerBK::name, tol::posint)
    local cons, fun, cp, IBF;
    option trace;
    description "Check the accuracy of the R routine halfOrderAIBF()";
    if tol < 20 then
        error "tol should be at least 20 Digits";
    end if;
    Digits := tol;
    if abs(lambda * 2) <> 1 then
        error "lambda should be +/- a half"
    end if;
    cons := 1/2 * (chi/psi)^(lambda/2);
    cp:= evalf(cons);
    fun:= exp(-1/2 * (chi * omega + psi/omega)) * omega^(lambda - 1);
    if lowerBK = TRUE then
        IBF:= cp * integrate(fun, omega = 0..v);
        print("Argument lowerBK of the R routine halfOrderAIBF() must be FALSE")
    elif lowerBK = FALSE then
        IBF:= cp * integrate(fun, omega = v..infinity);
        print("Argument lowerBK of the R routine halfOrderAIBF() must be TRUE")
    else error "lowerBK should be either TRUE or FALSE";
    end if;

232
evalf(IBF);
end proc;

halfOrderAIBFproc(20, 50, 5/2, -1/2, TRUE, 40)
D.8 Function to compare analytical and numerical methods for $\lambda = \pm 1/2$

Description

For $\lambda = \pm 1/2$, this function compares numerical accuracy and stability of the analytical method against those of the numerical method. The equality of the evaluation results by these methods is established by (5.64). The LHS and RHS of this equation are evaluated using \texttt{halfOrderAIBF()} and \texttt{numeIBF()} respectively.

\texttt{methodsComp(v, chi, psi, lambda, numeMethod = TRUE, traceIBF = TRUE, 
eps = 0.9, digits = 20, lowerBK = TRUE)}

Arguments

\begin{itemize}
\item \texttt{v} Value, $v > 0$, for the limit of the integration.
\item \texttt{chi, psi} Value of $\chi > 0$, $\psi > 0$.
\item \texttt{lambda} Value of order $\lambda \in \mathbb{R}$.
\item \texttt{numeMethod} Logical, if \texttt{TRUE} then the function \texttt{numeIBF()} is called. Otherwise the function \texttt{halfOrderAIBF()} is called.
\item \texttt{traceIBF} Logical. If \texttt{TRUE} then the value of the denominator and nominator of $G_n$, which was specified by (5.48), for each iteration $n$ will be output. This option works as a diagnostic tool to detect when this function may encounter numerical issues.
\item \texttt{eps} Numeric value between $(0, 1)$ to be used by the stopping rule in (5.49).
\item \texttt{digits} Number of decimal places to print.
\item \texttt{lowerBK} Logical. If \texttt{TRUE} then the value of the lower incomplete Bessel function is returned.
\end{itemize}

Examples

The following \texttt{R} code extends the numerical tests which were used to construct Table 5.7.
The following R code extends the numerical tests which were used to construct Table 5.7.

```r
# R code
v <- 1
chi <- seq(30, 50, by = 10)
psi <- seq(10, 16, by = 2)
lambda <- -1/2
for(i in 1:length(chi)) {
  for(j in 1:length(psi)) {
    methodsComp(v = v, chi = chi[i], psi = psi[j], lambda = lambda,
                traceIBF = FALSE, eps = 0.9, lowerBK = TRUE, digits = 20)
  }
}
```

```maple
v := 6;
lambda := 1/2;
for chi from 30 to 50 by 10 do;
  for psi from 10 to 16 by 2 do;
    halfOrderAIBFproc(v, chi, psi, lambda, FALSE, 90);
  end do;
end do;
```

The following R code extends the numerical tests which were used to construct Table 5.7.

```r
# R code
v <- 6
chi <- seq(30, 50, by = 10)
psi <- seq(10, 16, by = 2)
lambda <- 1/2
for(i in 1:length(chi)) {
  for(j in 1:length(psi)) {
    methodsComp(v = v, chi = chi[i], psi = psi[j], lambda = lambda,
                traceIBF = FALSE, eps = 0.9, lowerBK = TRUE, digits = 20)
  }
}
```
for(j in 1:length(psi)) {
    methodsComp(v = v, chi = chi[i], psi = psi[j], lambda = lambda, 
    traceIBF = FALSE, eps = 0.9, lowerBK = TRUE, digits = 20)
}

# Maple code
v := 1;
lambda := -1/2;
for chi from 30 to 50 by 10 do
    for psi from 10 to 16 by 2 do;
        halfOrderAIBFproc(v, chi, psi, lambda, FALSE, 90);
    end do;
end do;
Appendix E

Main functions of Chapter 6

E.1 Important testing information

Numerical examination of Chapter 6 involves the function \texttt{pgig()} which is available in package \texttt{GeneralizedHyperbolic} (Version 0.4–1) as

\begin{verbatim}
pgig(q, chi = 1, psi = 1, lambda = 1,
    param = c(chi,psi,lambda), log.p = FALSE, lower.tail = TRUE,
    ibfTol = .Machine$double.eps^(0.85), nmax = 100)
\end{verbatim}

Another version of \texttt{pgig()} is also available in package \texttt{HyperbolicDist} (Version 0.6–2) as

\begin{verbatim}
pgig(q, Theta, small = 10^(-6), tiny = 10^(-10), deriv = 0.3,
    subdivisions = 100, accuracy = FALSE, ...)
\end{verbatim}

with \(Theta = c(lambda,chi,psi)\).

Because both of these functions share the same name but the parameters of the distribution are specified differently (i.e., \texttt{param = c(chi, psi, lambda)} versus \(Theta = c(lambda, \chi, psi)\)) so if both of these packages are loaded into \texttt{R} at the same time then these functions clash one another. To avoid this possible issue during a numerical test, the function \texttt{pgig()} of package \texttt{GeneralizedHyperbolic}
was copied and pasted (i.e., completely unaltered) to a text file called “Functions of GeneralizedHyperbolic Package”. This means that when the function pgigGH() is called it does not clash with the function pgig() of HyperbolicDist should this package is also loaded.

Secondly, the density function of the GIG which is used by pgigGH() is given by (1.1) which is the form specified on p.416 of Barndorff-Nielsen (1977). This means that the argument densBN of gigCDF() (see, Appendix E.2) should be TRUE for the returned probabilities by pgigGH() and gigCDF() to be comparable.

Thirdly, pgig() in package GeneralizedHyperbolic calls incompleteBesselK() to numerically evaluate the incomplete Bessel function in (1.1). incompleteBesselK() is a FORTRAN version of the algorithm by Slevensky and Safouhi described in Section 5.5. However, pgigGH() uses incompleteBesselKR(), which is also an implementation of the algorithm by Slevensky and Safouhi, to evaluate the same integral. Both incompleteBesselK() and incompleteBesselKR() return the same evaluation result but the former is significantly faster in terms of computational speed.
E.2 Function to calculate tail probabilities of the
generalized inverse Gaussian distributions

Description

This function calculates tail probabilities of the generalized inverse Gaussian distribution using both analytical and numerical method.

Usage

\[
gigCDF(q, \chi, \psi, \lambda, \text{numeMethod} = \text{TRUE}, \text{traceIBF} = \text{TRUE}, \text{densBN} = \text{TRUE}, nmax = 80, \text{eps} = 0.85, \text{lower.tail} = \text{TRUE})
\]

Arguments

- **q**: Scalar quantile values.
- **chi**, **psi**: Value of $\chi > 0$ and $\psi > 0$.
- **numeMethod**: Logical. If **TRUE** then the function **numeIBF** in Appendix D.6 is called. If **FALSE** then the the function **halfOrderAIBF** in Appendix D.4 is called.
- **lambda**: Value of order $\lambda \in \mathbb{R}$ if **numeMethod** = **TRUE**. Otherwise only $\lambda = \pm \frac{1}{2}$ is applicable.
- **traceIBF**: Logical. If **TRUE** then details of the evaluation process by the function **numeIBF** is printed. This option serves as a diagnostic tool to detect the numerical problems that the algorithm by Slevensky and Safouhi can encounter. This option is only workable if **numeMethod** = **TRUE**.
- **densBN**: Logical. If **TRUE** then probabilities are calculated using the density function of the GIG specified by (1.1) (cf. p.416 of [Barndorff-Nielsen](1977)), which is the density function used by the function **pgig()** of package **GeneralizedHyperbolic**. If **FALSE** then probabilities are calculated using the density function of the GIG specified by (5.50).
nmax Maximum number of iterations allowed for the $G$ transformation. This only applies if numeMethod = "TRUE" and IBFFortran = "FALSE".

eps Numerical value in $(0.5, 0.98)$. It effects the stopping rule of the function numeIBF() as specified in (5.49).

digits Number of decimal places to print.

lower.tail Logical. If TRUE then the CDF is returned. If FALSE then the CCDF is returned.

References
Chapter 6 and references therein. The numerical method used by this function has already included in package GeneralizedHyperbolic as the function pgig(). However, this function does not have the argument traceIBF.

Example

```r
gigCDF(0.5, 1, 2, 1/2, numeMethod = FALSE, traceIBF = FALSE, 
nmax = 80, eps = 0.85, lower.tail = TRUE)
```
E.3 Maple procedure to calculate tail probabilities of the generalized inverse Gaussian distributions

Description

This function calculates tail probabilities of the GIG by numerically integrating the density function given on p.416 of Barndorff-Nielsen (1977). Here, the values of $\lambda = \pm \frac{1}{2}$ to ensure robustness of the test. The evaluation of this function is considered to give “exact” value of the probabilities required. The results are only comparable to the evaluation of the function numeIBF(), see Appendix E.2, if the argument densBN of this function is TRUE.

\[
gigCDFproc := \text{proc}(q::\text{positive}, \chi::\text{positive}, \psi::\text{positive}, \\
\quad \lambda::\text{numeric}, \text{lowerBK}::\text{name}, \text{tol}::\text{posint}) \\
\text{local} \ cons, x, y, \text{fun, CDF}; \\
\text{option} \ trace \\
\text{description} \ "\text{Procedure to check accuracy of gigCDF() by integrating the density function given on p.416 of BN1977};" \\
\text{if} \ \text{tol} < 30 \ \text{then} \\
\quad "\text{tol should not be less than 30 digits};" \\
\text{end if}; \\
\text{Digits} := \text{tol}; \\
\text{if} \ \text{abs}(\lambda * 2) <> 1 \ \text{then} \\
\quad \text{error} \ "\lambda \text{ should be } +/- \ 1/2" \\
\text{end if}; \\
\text{cons} := 1/2 * 1/BesselK(\lambda, \sqrt{\chi * \psi}) * (\psi/\chi)^{(\lambda/2)}; \\
\text{fun} := \exp(-1/2 * (\chi/\omega + \psi * \omega)) * \omega^{(\lambda - 1)}; \\
\text{if} \ \text{lowerBK} = \text{TRUE} \ \text{then} \ \text{# CDF is calculated}
CDF := cons * integrate(fun, omega = 0..q);

elif lowerBK = FALSE then  # CCDF is calculated
    CDF := cons * integrate(fun, omega = q..infinity);
else error "LowerBK assumes either TRUE or FALSE"
end if;

print("Argument densBN of gigCDF() must be TRUE");
evalf(CDF);
end proc;
Appendix F

Main functions of Chapter 7

F.1 Conducting Cramér-von Mises test for the hyperbolic distribution

Description
This function carries out the Cramér-von Mises test for the hyperbolic distribution. It eliminates the need for dividing up the parameter space as pointed out by the examiner who recommended that the logit transformation should be used to keep the predicted $p$-value in the $[0, 1]$ interval.

Usage

\[ \text{hypCvMTestSpline}(x, \Theta, \text{printOut = "FALSE"}) \]

Arguments

- \text{x} \quad \text{Data to be tested.}
- \text{Theta} \quad \text{Estimated parameters using an efficient method such as maximum-likelihood estimates. Theta must be in Parameterization 1 of package GeneralizedHyperbolic or } \theta = (\mu, \delta, \pi, \zeta)'.
- \text{printOut} \quad \text{Print } p\text{-value if TRUE.}
References

This function will be added to the package GeneralizedHyperbolic in due course.

Example

```r
param <- c(2, 12, 0.5, 0.01)
set.seed(1)
dataVector <- rhyperbGH(500, param = param)
fittedparam <- hyperbFitGH(dataVector)$param
hyperbCvMTest(dataVector, param = fittedparam)
fittedparam <- hyperbChangeParsGH(2, 1, fittedparam)
hypCvMTestSpline(dataVector, Theta = fittedparam, printOut = "FALSE")
```
F.2 Moran test for the hyperbolic distribution

Description
This function conducts the goodness-of-fit test for the HYP applying the Moran log spacing statistic.

Usage

\[ \text{hypMoranTest}(x, \Theta, \alpha = 0.05) \]

Arguments

- **x**: Data to be tested.
- **Theta**: Estimated parameters using efficient method such as the maximum-likelihood estimates. \(\Theta\) must be in Parameterization 1 of package \texttt{GeneralizedHyperbolic}.
- **alpha**: Level of significance for the test. This function returns 1 if the null hypothesis is rejected and 0 otherwise.

References
This function will be added to the package \texttt{GeneralizedHyperbolic} in due course. Section 7.4 and references therein.

Example

```r
require(GeneralizedHyperbolic)
param <- c(2, 12, 0.5, 0.01)
dataVector <- rhyperbGH(500, param = param)
fittedparam <- hyperbFitGH(dataVector)$param
sigLevel <- 0.05
# Change parameterization from Parameterization 2 returned
# by \text{hyperbFit( )} to Parameterization 1 to input \text{hypMoranTest( )}
fittedparam <- hyperbChangeParsGH(2, 1, fittedparam)
hypMoranTest(dataVector, Theta = fittedparam, alpha = sigLevel)
```

245
F.3 Function to examine the power of the tests

Description
This function compares the power of the Cramér-von Mises and Moran goodness-of-fit test for the hyperbolic distribution.

Usage

```r
powerTest(x, level.sig = sig)
```

Arguments

- `x` Data to be tested.
- `alpha` Level of significance for the test.

This function calls `hypCvMtestSpline()` and `hypMoranTest()` to obtain the fitted parameters. It then calls `hypCvMtestSpline()` and `hypMoranTest()`.

Example

```r
require(ghyp) # Load ghyp package
data(smi.stocks) # Obtain real data
stocks <- c("SMI", "Novartis", "Nestle")
data <- smi.stocks[,"Novartis"]
powerTest(x = data, level.sig = 0.05)
```
Bibliography


URL http://pubget.com/paper/pgtmp_10042445


URL http://CRAN.R-project.org/package=ghyp


Maechler, M., 2011. Bessel: Bessel – Bessel functions computations and approximations. R package version 0.5-3.
URL http://CRAN.R-project.org/package=Bessel


URL http://CRAN.R-project.org/package=QRMLib


URL http://www.freidok.uni-freiburg.de/volltexte/15/pdf/15_1.pdf


URL http://CRAN.R-project.org/package=GeneralizedHyperbolic


254


