Series Representations and Approximation of some Quantile Functions appearing in Finance

Asad Ullah Khan Munir

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University of London.

Department of Mathematics
University College London

2013

I, Asad Ullah Khan Munir, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

_____

_____

Abstract

It has long been agreed by academics that the inversion method is the method of choice for generating random variates, given the availability of a cheap but accurate approximation of the quantile function. However for several probability distributions arising in practice a satisfactory method of approximating these functions is not available. The main focus of this thesis will be to develop Taylor and asymptotic series representations for quantile functions of the following probability distributions; Variance Gamma, Generalized Inverse Gaussian, Hyperbolic, α-Stable and Snedecor's F distributions. As a secondary matter we briefly investigate the problem of approximating the entire quantile function. Indeed with the availability of these new analytic expressions a whole host of possibilities become available. We outline several algorithms and in particular provide a C++ implementation for the variance gamma case. To our knowledge this is the fastest available algorithm of its sort.

Acknowledgements

I am especially grateful to my supervisor Prof. William T. Shaw for giving me the opportunity to do a PhD in mathematics. He has been extremely inspirational to me in developing the results presented in this thesis. Indeed without my discussions with Prof. Shaw I fail to see where I would have found the motivation to persevere. I very much appreciate the financial support and resources provided to me by both the computer science and mathematics departments at UCL. I thank the Knowledge Transfer Network (KTN) for Industrial Mathematics and the reinsurance group Willis Re for offering me a research internship. I have only good words to say about both of these organizations. Last but not least I would like to thank my family in particular my loving parents for their endless patience, financial and moral support.

Contents

Chapter 1

Introduction

Analytic expressions for quantile functions have long been sought after. The importance of these functions comes from their widespread use in applications of statistics, probability theory, finance and econometrics. Therefore much effort has been devoted into their study, in particular since closed form expressions for the quantile function of most distributions are not known, several approximations appear in the literature. These approximations generally fall into one of four categories, series expansions, functional approximations, numerical algorithms or closed form expressions written in terms of a quantile function of another distribution. The focus of this report is on the former two categories.

We follow the philosophy of Steinbrecher and Shaw (2008), Shaw et al. (2011) and Shaw and McCabe (2009) closely, that is to "elevate quantile functions to the same level of management as many of the classical special functions of mathematical physics and applied mathematics". In particular this requires efficient computation. According to Lozier and Olver (1994), there are three stages in the development of computational procedures of special functions:

1. Derivation of relevant mathematical properties: This stage is primarily an exercise in applied mathematics, and is concerned with finding representations of various forms such as, asymptotic expansions, difference and differential equations, functional identities, integral representations, and Taylor series expansions.

2. Development of numerical approximations and algorithms: This stage is an exercise in numerical analysis and is concerned with finding Chebyshev series expansions, minimax polynomial and rational approximations, Padé approximations, numerical quadrature and numerical solutions of difference and differential equations.

3. Construction and testing of robust software: This final stage is an exercise in computer science, and though highly dependent on stage 2, great benefits can be

gained at this stage such as the possibility of parallelization and or the utilization of various platform specific features.

The bulk of the work we carry out in this report belongs to stages 1 and 2, in particular we will develop Taylor and asymptotic series expansions for the quantile functions of the following probability distributions; Variance Gamma, Generalized Inverse Gaussian, Hyperbolic, α-Stable and Snedecor's F. With these analytic expressions in place we proceed to stage 2 where we will be concerned with the construction of various types of approximants such as continued fractions, Chebyshev series and minimax approximations. Finally we briefly visit stage 3, where a `C++` implementation of an algorithm used to approximate the variance gamma quantile is provided.

There is no shortage of research articles discussing the approximation of quantile functions, see for example Dagpunar (1989); Derflinger et al. (2009, 2010); Lai (2009); Farnum (1991); Leydold and Hörmann (2011) to name a few. These papers are primarily concerned with applying numerical techniques such as root finding and interpolation to approximate the quantile function. Our approach however is centred around formulating a first order ordinary differential equation problem. We are not the first however to consider this differential equation approach, such a route is also taken by Ulrich and Watson (1987) and Leobacher and Pillichshammer (2002). Here numerical schemes such as Runge-Kutta methods are used to seek solutions. On the other hand we will seek analytic solutions by applying certain well known techniques from applied mathematics.

Aside from Shaw's series of Quantile Mechanics papers which we will discuss in subsequent chapters, few authors have written about series representations of quantile functions. Possibly the earliest known (at least to this author) series representation is the famous Cornish Fisher expansion introduced in (Cornish and Fisher, 1938; Fisher and Cornish, 1960). The Cornish Fisher expansion expresses the u-quantile of a random variable X in terms of its cumulants and the u-quantile of the standard Gaussian distribution. The idea was generalized in Hill and Davis (1968) by allowing for non Gaussian base distributions. The Cornish Fisher expansion is often used for VAR applications in finance, see for example Jaschke (2002).

Another interesting approach was introduced by Takemura (1983). Here a base quantile function Q_B is chosen, based on which an orthonormal basis for the set of square integrable functions on the unit interval is formed. Consequently the Fourier series expansion of the target quantile Q_T with respect to this basis may be developed. Unlike the Cornish Fisher expansion which is asymptotic in nature Takemura's approach yields a convergent series in the L^2 norm. Note however the computation of the Fourier coefficients usually requires numerical quadrature and that for approximation

purposes the L^∞ norm is preferred, see Shaw et al. (2011) for details. We will discuss the Cornish Fisher expansion and Takemura's approach in more detail in chapter 4.

After a brief introduction to the theory of quantile functions in chapter 2 we motivate the discussion with a chapter on applications and another on existing methods. In chapter 5 we discuss some general techniques used to solve non-linear ordinary differential equations. These will be employed in chapter 6, where we provide series representations for some quantile functions not currently available. Key to this approach is the availability of the density function. In chapter 7 some alternative approaches reliant instead on the characteristic and distribution functions respectively are investigated. As a result series representations for the α-stable quantile function will be presented.

The fact that a convergent series for a specific quantile function is available may seem to indicate that the computation of such a function is of no concern. However for modern computing systems this is simply not the case, since:

- they are limited in speed; a big concern for slowly converging power series when a desired accuracy is required,
- can only represent a finite range of numbers; the terms of the series may become extremely small or large leading to underflow and overflow errors respectively,
- and can only perform finite precision arithmetic; which of course is a problem for most numerical computations.

It is for these reasons series acceleration techniques (Brezinski and Zaglia, 1991) prove so useful, and we will use them to good affect in our numerical experiments discussed in chapter 8. Here the reader will find many useful recipes to construct algorithms for approximating Q. The key here is to apply a change of variable. To this end we will employ a technique devised by Shaw et al. (2011), albeit from a slightly different perspective, and consequently tackle the problem of approximating Q. An overview of a `C++` program implementing these ideas is presented in section 8.9.

To summarize the key contributions in this thesis are provided in chapters 6 and 8. Chapter 6 will be concerned with the developing the "ingredients" required to construct numerical recipes for approximating Q. In particular we write down some functional identities and develop some series representations in this chapter. In chapter 8 we put together some numerical recipes based on these ingredients, for example

- we devise a strategy based on continued fractions,
- using the methods of Thacher (1964) and Sidi (1975) we construct Chebyshev-Padé approximants,

- and based on Maehly's indirect method (Maehly, 1963) we construct minimax approximations.

Note that key to all these numerical methods is knowledge of the Taylor coefficients. Some timing and comparison information is provided in chapter 8. In particular we compare Thacher's and Maehly's methods to the standard approaches of computing Chebyshev coefficients and minimax approximants. The point is that on their own Taylor series expansions may not provide good approximations but with a little extra effort, they may be used as stepping stones to construct much superior approximants.

Of course there will be an overhead in generating these approximations. The resulting approximation is only valid for one set of distribution parameters. Given this initial setup time we imagine Monte Carlo users of sample sizes 10^6 and upwards for the same set of parameters may find the techniques in this thesis useful.

Chapter 2

Quantile Functions

In this section we give a brief introduction to quantile functions and fix notation. Consider a random variable, X defined on the probability space $(\Omega, \mathfrak{B}, \mathbb{P})$. The cumulative distribution function (c.d.f.) of X, is the function $F_X : \mathbb{R} \mapsto [0,1]$ defined by $F_X(x) = \mathbb{P}(X \leq x)$. What makes the c.d.f. $F_X(x)$ so useful is that, it completely characterizes the probability distribution of X in the following sense.

Theorem 1. *If two random variables X_1 and X_2 both have the same distribution function F then X_1 and X_2 have the same probability distribution.*

Proof. See Jacod and Protter (2004, p. 39) □

As we will see another way to characterize a probability distribution of a random variable X is through the Quantile function which we define next.

Definition 2. The Quantile function of a random variable X, is the function $Q_X : (0,1) \mapsto \mathbb{R}$ defined by

$$Q_X(u) = \inf \{x : F(x) \geq u\}.$$

The u^{th} quantile of X is defined to be the value $Q_X(u)$, denoted by x_u. If u is a multiple of $1/100$, then the quantile x_u is often referred to as the $(u \times 100)^{\text{th}}$ percentile. Some quantiles have been designated specific names, for example $x_{0.25}$ is referred to as the lower quartile (or the 25^{th} percentile), $x_{0.5}$ is referred to as the median (or the 50^{th} percentile) and $x_{0.75}$ is reffered to as the upper quartile (or the 75^{th} percentile)[1]. The following theorem lists some useful properties of the quantile function. Note in particular when the distribution function F is continuous and strictly increasing the definition of Q coincides with the functional inverse of F.

Theorem 3. *Suppose $X : \Omega \mapsto \mathbb{R}$ is a random variable with c.d.f. F and quantile function Q, then*

[1] According to David (1995) the terms percentile, quartile and median were first introduced by Galton in the late eighteen hundreds.

1. Q is a non-decreasing function
2. $Q(F(x)) \leq x, \forall x \in \mathbb{R}$
3. $F(Q(u)) \geq u, \forall u \in (0,1)$
4. $Q(u) \leq x$ if and only if $F(x) \geq u$
5. If F^{-1} exists, then $Q(u) = F^{-1}(u)$

Proof. Define $A(u) = \{y : F(y) \geq u\}$.

1. Suppose $u_1 \leq u_2$ then since F is non-decreasing we have $A(u_1) \supseteq A(u_2)$ and hence $Q(u_1) \leq Q(u_2)$.
2. $Q(F(x)) = \inf A(F(x))$, but clearly $x \in A(F(x))$ so the result follows.
3. If $A(u)$ is a closed set in $\mathbb{R} \cup \{-\infty, \infty\}$ then it must contain its infimum, that is $Q(u) \in A(u)$ so $F(Q(u)) \geq u$. To show $A(u)$ is closed we prove any convergent sequence $(x_n)_{n\geq 1}$ in $A(u)$ has a limit point which is also in $A(u)$. Suppose $x_n \to x_0$ as $n \to \infty$, then since $x_n \in A(u)$ we have $F(x_n) \geq u$ for all $n \geq 1$ and in particular $\liminf_{n\to\infty} F(x_n) \geq u$. However since F is non-decreasing and right continuous we have also have $\liminf_{n\to\infty} F(x_n) \leq F(x_0)$, hence $F(x_0) \geq u$ which implies $x_0 \in A(u)$.
4. If $Q(u) \leq x$ then $x \in A(u)$ so $F(x) \geq u$. Conversely if $F(x) \geq u$ then again $x \in A(u)$ and $Q(u) \leq x$ since $Q(u)$ is the infimum.
5. In the case the functional inverse exists we may write $A(u) = \{y : y \geq F^{-1}(u)\}$ and the infimum of this set is clearly $F^{-1}(u)$.

□

As mentioned earlier the quantile function characterizes the probability distribution; two random variables with the same quantile function will have the same probability distribution.

Theorem 4. *Let X_1 and X_2 be two real valued random variables with quantile functions Q_1 and Q_2 respectively. If $Q_1(u) = Q_2(u), \forall u \in (0,1)$ then X_1 and X_2 have the same probability distribution.*

Proof. Let the distribution functions of X_1 and X_2 be given by F_1 and F_2 respectively. For a fixed value x_0 by part 2 of theorem 3 we have $Q_1(F_2(x_0)) = Q_2(F_2(x_0)) \leq x_0$ hence $F_1(x_0) \geq F_2(x_0)$ by part 4. On the other hand $Q_2(F_1(x_0)) = Q_1(F_1(x_0)) \leq$

x_0 so $F_2(x_0) \geq F_1(x_0)$ and we may conclude $F_1 = F_2$. The result then follows from theorem 1.

we have suppose $F_1(x_0) < F_2(x_0)$, then by parts 1 and 5 of theorem 3 we have $Q_2(F_1(x_0)) \leq Q_2(F_2(x_0)) \leq x_0$. Applying part 3 of the same theorem yields the contradiction $F_1(x_0) \geq F_2(x_0)$. A similar argument rules out the case $F_1(x_0) > F_2(x_0)$, thus $F_1 = F_2$. □

When working with a location-scale family of distributions, the following theorem is useful. Given that we know the quantile function of a random variable X, with distribution belonging to a location-scale family, it is a simple matter to compute the quantile function of any other member of the family. This follows from the fact if Y is also a member of the same location-scale family then it may be written as a linear transformation of X.

Theorem 5. *Suppose that X is a random variable with a strictly increasing and continuous c.d.f. Let $Y = a + bX$ for some constants $a, b \in \mathbb{R}$ with $b \neq 0$ then,*

$$Q_Y(u) = \begin{cases} a + bQ_X(u) & b > 0 \\ a + bQ_X(1-u) & b < 0 \end{cases}$$

where Q_X and Q_Y are the quantile functions of X and Y respectively.

Proof. The c.d.f. F_Y of Y can be written in terms of the c.d.f. F_X of X as,

$$F_Y(y) = \begin{cases} F_X\left(\frac{y-a}{b}\right) & b > 0 \\ 1 - F_X\left(\frac{y-a}{b}\right) & b < 0 \end{cases}$$

from which the result follows. □

Life also becomes a lot simpler when we are dealing with symmetric distributions, due to the following theorem. Suppose we wanted to approximate the quantile function on its domain $(0, 1)$, then in the symmetric case we need only focus our attention on the shorter interval $[1/2, 1)$.

Theorem 6. *Suppose X is a random variable with a strictly increasing and continuous c.d.f. If the density f_X of X satisfies $f(x) = f(-x)$, $\forall x \in \mathbb{R}$ then*

$$Q_X(u) = -Q_X(1-u), \quad u \in (0,1).$$

Proof. From symmetry of X we have $F_X(x) = 1 - F_X(-x)$ and the result follows. □

There are some probability distributions for which an explicit expression for the quantile function may be written down explicitly.

Example 7 (Pareto)**.** The c.d.f. of the Pareto distribution is given by,

$$F\left(x; x_m, \alpha\right) = 1 - \left(\frac{x_m}{x}\right)^{\alpha}, \quad x \geq x_m,\ x_m, \alpha > 0,$$

which can be directly inverted to obtain,

$$Q\left(u; x_m, \alpha\right) = \frac{x_m}{(1-u)^{\frac{1}{\alpha}}}.$$

Example 8 (Logistic)**.** The c.d.f. of the logistic distribution is given by,

$$F\left(x; \mu, s\right) = \frac{1}{1 + e^{-(x-\mu)/s}}, \quad x, \mu \in \mathbb{R},\ s > 0,$$

which can be directly inverted to obtain,

$$Q\left(u; \mu, s\right) = \mu + s \ln\left(\frac{u}{1-u}\right).$$

There are many other examples for which the c.d.f. is invertible such as in the case of the Student-t distribution with $\nu = 1, 2, 4$ degrees of freedom. For a more comprehensive list the reader is referred to the monograph by Gilchrist (2000). However for the large majority of probability distributions the quantile function may not be expressed in closed form as in the above examples. The Student-t, Beta, Variance Gamma and Generalized Hyperbolic are amongst this class of distributions. There are many reasons why one would want to compute the quantile, some of which are discussed in chapter 3. The aim of this report is to investigate methods used to find useful representations of quantile functions when a closed form expression is not available.

Chapter 3

Applications

Below we will discuss some applications of the quantile function. They provide the motivation behind seeking practical representations of these functions. We have focused on finance related applications but obviously the quantile function has much broader appeal, in particular in applications of statistics and econometrics which we have not included, see Gilchrist (2000) for further examples.

3.1 Value at Risk

Value at risk (VaR) is the most widely used risk measure in finance. It is defined as follows, for a fixed time horizon T, let L denote the potential loss of a portfolio of risky assets over this period. Assume further that the distribution function of the loss L is given by $F_L(x) = \mathbb{P}(L \leq x)$, then for a given confidence level $\alpha \in (0,1)$ the VaR is defined as,

$$\text{VaR}_{\alpha,T} = \inf\{x \in \mathbb{R} : \mathbb{P}(L > x) \leq 1 - \alpha\}. \tag{3.1.1}$$

That is the VaR is the smallest number x such the the probability the loss exceeds x is less than $1-\alpha$ over the time period T. Typical values of α are $\alpha = 0.95$ or $\alpha = 0.99$. Note that equivalently we may write (3.1.1) as,

$$\begin{aligned} \text{VaR}_{\alpha,T} &= \inf\{x \in \mathbb{R} : \mathbb{P}(L \leq x) \geq \alpha\} \\ &= \inf\{x \in \mathbb{R} : F_L(x) \geq \alpha\}. \end{aligned}$$

From this we see the VaR is simply the α quantile of the loss distribution. Although popular there are many shortcomings of VaR, for details we refer the interested reader to McNeil et al. (2005).

3.2 Monte Carlo

A necessary requirement to Monte Carlo simulations is the ability to produce random numbers. Assuming that a good random number generator is available, and capable of producing random deviates from $\mathrm{U}(0,1)$, the standard uniform distribution, a transformation method is required to transform these deviates to the desired distribution, e.g. Normal, Student-t etc. The inversion method is the simplest and most general method for this purpose and makes direct use of the quantile function. Suppose $U \sim \mathrm{U}(0,1)$ then we may generate a random variate X with desired c.d.f. F through the transformation $X = Q(U)$, where Q is the associated quantile function. To verify this claim observe that,

$$\begin{aligned} \mathbb{P}(X \leq x) &= \mathbb{P}(Q(U) \leq x) \\ &= \mathbb{P}(U \leq F(x)) \\ &= F(x). \end{aligned}$$

Note we have not placed any restrictions (such as continuity or strict monotonicity) on F, the second equality follows from part (4) of theorem 3. In some specific cases due to the difficulty in evaluating $Q(U)$ the inverse method may be computationally expensive resulting in a slow algorithm. Dependent on the cost in evaluating $Q(U)$ the inverse method is generally considered to be the method of choice amongst academics (L'Ecuyer, 2012; Korn et al., 2010). One of the objectives of this thesis is to improve the efficiency in evaluating the quantile function so that we can benefit from some of the advantages of the inversion method listed below.

1. It is the most general method, it can be used to generate variates from discrete, continuous and mixed distributions, see Law and Kelton (2000, § 8.2.1).

2. It only requires one uniform random variate U to generate a non-uniform variate X. Also due to the monotonicity of Q there is a one to one correspondence between the U and X, in the case F is continuous and strictly increasing.

3. It allows one to sample from conditional distributions. Suppose the random variable X has c.d.f. F then the random variable X given $X \in [a,b)$ has the c.d.f. $H(x) = \mathbb{P}(a \leq X \leq x \,|\, a \leq X \leq b)$. Using the quantile function Q of X it is easy to generate random variates with c.d.f. H, see for example Korn et al. (2010, § 2.6.3) or Glasserman (2004, e.g. 2.2.5.).

4. Often the only transformation method compatible with important variance reduction techniques such as antithetic variates, common random numbers, stratified

and Latin hypercube sampling is the inversion method, see L'Ecuyer (2012) and references therein. This is due to the fact that variance reduction techniques rely on introducing correlation between the variates and the inversion method is capable of inducing maximum positive or negative correlation between the generated variates, see for example Bratley et al. (1987, p. 47), Law and Kelton (2000, p. 472) or Devroye (1986, p. 29) for details. For other reduction techniques (such as the method of control variates, importance sampling etc.) one may have the freedom to choose another transformation method (rejection sampling, ratio of uniforms etc.). However as pointed out by Korn et al. (2010, § 2.4.1), the best way to transform uniform random numbers is the inversion method because it preserves structures. If the distribution structure of the uniformly distributed random numbers is good, so will the structure of the transformed random numbers (see also the remark below on Neave's effect).

5. It allows one to generate the maximum of a sample in an efficient manner. Suppose we are interested in generating $X = \max\{X_1, \ldots, X_n\}$, where the X_i's are independent identically distributed random variates. One could generate $X_1, \ldots, X_n$ and then take the maximum, however a more efficient method exists. Suppose we find $V = \max\{U_1, \ldots, U_n\}$ where $U_1, \ldots, U_n$ are independent identically distributed uniform $[0, 1]$ random variates, then it happens that $X = Q(V)$. This method avoids having to generate the entire sequence $X_1, \ldots, X_n$, see Devroye (1986, p. 30) for more information.

6. It allows one to generate order statistics in an efficient manner. Suppose $X_1, \ldots, X_n$ are independent and identically distributed random variables and we are interested in generating the order statistic $X_{(i)}$ for some $i \in \{1, \ldots, n\}$. An obvious approach would be to generate the X_i's through any method available and then sort the output $X_1, \ldots, X_n$ in ascending order, and choose the i^{th} smallest value in the list. This technique can be extremely slow when n is large as it requires us to generate n different random variates and sort the result when all we are interested in generating is $X_{(i)}$. An alternative approach is provided by the following algorithm,

   (a) Generate $Y \sim \text{Beta}(i, n - i + 1)$

   (b) Set $X_{(i)} = Q(Y)$

   The validity of this algorithm is established in Law and Kelton (2000, prob. 8.5).

7. When considering low discrepancy sequences (quasi random numbers) some authors suggest the only method that preserves the structure of the sequence is the

inversion method, see for example Galanti and Jung (1997), Moro (1995), Joy et al. (1996, footnote 6), or Korn et al. (2010, § 2.7.5). It should be noted however Ökten and Göncü (2011) have suggested through empirical tests that preference of the inversion method over the Box Muller transform is a result of "common folklore". In the same reference the Marsaglia transform is clearly rejected as a viable choice.

8. Acceptance rejection methods (and their extensions such as adaptive rejection sampling) are generally inapplicable with quasi Monte Carlo methods (Glasserman, 2004, p. 62). The effectiveness of quasi Monte Carlo deteriorates as the dimension increases. The dimension of the problem is typically the maximum number of uniforms required to generate a simulation "path". With a rejection scheme there is no single upper bound on the number of uniforms required to generate a single nonuniform variable; therefore simulations that use rejection methods correspond to infinite dimensional problems. As we have noted since the inversion method requires only one uniform variable per nonuniform variable, this issue does not arise with the inversion method.

9. When uniform pseudo random variate generation methods are paired with transform methods to generate variates from a specified distribution, the two may be incompatible. Four transform methods have been paired and studied with linear congruential generators in Hörmann (1993) and the recommended method of transform is the inversion method. For example when the Box Muller transform is paired with congruential generators strange behaviour known as the Neave effect has been reported, see remark 9 and the references Ripley (1987, p. 55) or Jäckel (2002, § 9.3.2). The root cause of this bad behaviour is the poor performance of the uniform generator; for this particular pairing some workarounds are given in Golder and Settle (1976). However the problem is not just with linear congruential generators, it has also been reported to occur when the Box Muller transform is paired with a Tausworthe generator, see Tezuka (1991). Other transform methods which have been reported to be susceptible to Neave's effect are Marsaglia's and Bailey's polar methods used to generate normal and student-t variates respectively, see Gentle (2003, p. 185).

Remark 9. In his experiment Neave used a linear congruential generator,

$$X_i = (bX_{i-1} + c) \bmod m$$

to generate $\mathrm{U}(0,1)$ random variates $U_i = X_i/m$. He used the following parameter values, multiplier $b = 131$, increment $c = 0$ and modulus $m = 2^{35}$. When paired with the Box Muller transform to generate a pair of independent $N(0,1)$ random variates,

$$\begin{aligned} Y &= \sqrt{-2\ln U_i}\cos\left(2\pi U_{i+1}\right) \\ Z &= \sqrt{-2\ln U_i}\sin\left(2\pi U_{i+1}\right), \end{aligned}$$

Neave observed that for two million random variates generated the pseudo random numbers Z lie in the range $(-3.3, 3.6)$, but obviously there is a positive probability for normal variates to fall outside of this range. We repeated Neave's experiment in Mathematica (Wolfram-Research, 2010) and contrasted it to the inversion method. The code for the experiment along with the test results are supplied in figure C.0.1. Our tests not only confirm Neave's findings but also confirm that this strange behaviour does not occur with the inversion method.

3.3 Copulas

Copula functions are used to model dependency amongst random variables in a very versatile way. They allow us to define a dependency structure independent of how the marginals are defined. In this way arbitrary marginal distributions may be linked together with prespecified dependency structures. In order to be self contained we give a brief overview of this topic. As we will see, quantile functions take a prominent role in copula theory. Standard references for this section are Nelsen (2006) and Joe (1997), we will however closely follow the expositions given in Franke et al. (2011, Ch. 17) and Embrechts et al. (2003). We begin with a definition,

Definition 10. An d-dimensional copula is a continuous function $C : [0,1]^d \mapsto [0,1]$ satisfying the following properties, for all $\mathbf{u} = (u_1, \ldots, u_d)^\top \in [0,1]^d$ and $j \in \{1, \ldots, d\}$:

1. $C(u_1, \ldots, u_d)$ is increasing in each of its arguments.
2. $C(1, \ldots, 1, u_j, 1, \ldots, 1) = u_j$
3. for all $\mathbf{v} = (v_1, \ldots, v_d)^\top \in [0,1]^d$ such that $v_j \leq u_j$,

$$V_C(\mathbf{u}, \mathbf{v}) \geq 0$$

where V_C is the C-volume given by,

$$V_C(\mathbf{u}, \mathbf{v}) = \sum_{i_1=1}^{2} \cdots \sum_{i_d=1}^{2} (-1)^{i_1+\cdots+i_d} C(g_{1i_1}, \ldots, g_{di_d})$$

and $g_{j1} = v_j$ and $g_{j2} = u_j$.

These properties guarantee that copulae are valid distribution functions defined on the d-dimensional unit cube. Property (1) is a required property for any distribution function. Property (2) says that the marginals of the copula are uniform. The final property says that any d-dimensional cube in $[0,1]^d$ has non negative C-volume. In the case where $d = 2$ this property reads $C(v_1, v_2) - C(v_1, u_2) - C(u_1, v_2) + C(u_1, u_2) \geq 0$ and is a generalization of non-decreasing functions to higher dimensions. The following fundamental theorem links copula functions to more general probability distribution functions.

Theorem 11 (Sklar's Theorem)**.** *Let F be a d-dimensional probability distribution function with marginals $F_1, \ldots, F_d$. Then there exists a copula C such that*

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)), \quad \forall x_1, \ldots, x_d \in \mathbb{R}. \tag{3.3.1}$$

If $F_1, \ldots, F_d$ are continuous then C is unique. Conversely if C is a copula function and $F_1, \ldots, F_d$ are distribution functions then the function F defined in (3.3.1) is a joint distribution function with marginals $F_1, \ldots, F_d$.

A corollary to this theorem allows us to extract the copula from the joint distribution, provided we have knowledge of the quantile functions of the associated marginals.

Corollary 12. *Let $\mathbf{X} = (X_1, \ldots, X_d)^\top$ be a d-dimensional random vector with joint distribution function $F_{\mathbf{X}}$. If the marginals $F_1, \ldots, F_d$ of $F_{\mathbf{X}}$ are continuous and C is a copula satisfying (3.3.1) then,*

$$C(u_1, \ldots, u_d) = F_{\mathbf{X}}(Q_1(u_1), \ldots, Q_d(u_d)), \quad \forall u_1, \ldots, u_d \in [0,1], \tag{3.3.2}$$

where Q_i is the quantile function associated with the marginal distribution function F_i.

This allows us to isolate the dependence part of the joint distribution, and the resulting copula in (3.3.2) is called the *implicit* copula of this distribution. The next theorem states that copulae are invariant under monotonically increasing transformations and is particularly useful when trying to extract the implicit copula from a joint distribution function.

Theorem 13. *Let $\mathbf{X} = (X_1, \ldots, X_d)^\top$ be a d-dimensional random vector with copula $C_{\mathbf{X}}$ and continuous marginals. Suppose T_j is a strictly increasing function on the range of X_j for $j = 1, \ldots, d$. Consider the transformed random vector $\mathbf{Y} = (Y_1, \ldots, Y_d)^\top$, where $Y_j = T_j(X_j)$ for $j = 1, \ldots, d$ with copula $C_{\mathbf{Y}}$. Then $C_{\mathbf{X}} = C_{\mathbf{Y}}$ almost everywhere.*

Example 14 (Gaussian Copula)**.** We may extract the dependency structure, that is the implicit copula associated with the Gaussian distribution. This allows us to form distributions in which we are free to choose non-normal marginals, but the dependency structure remains Gaussian. To extract the Gaussian copula suppose $\mathbf{X} = (X_1, \ldots, X_d)^\top \sim N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where$\boldsymbol{\mu} = (\mu_1, ..., \mu_d)$ and $\boldsymbol{\Sigma} = (\sigma_{ij})_{i,j=1,...,d}$. From Sklar's theorem there exists a copula $C_{\mathbf{X}}$ such that distribution function $F_{\mathbf{X}}$, of $\mathbf{X}$ may be written as,

$$F_{\mathbf{X}}(x_1, \ldots, x_d) = C_{\mathbf{X}}(F_1(x_1), \ldots, F_d(x_d)),$$

where F_{X_j} is the distribution function of X_j. Next define the function T_j as,

$$T_j(x) = \frac{x - \mu_j}{\sigma_j},$$

where we have adopted the convention $\sigma_j = \sigma_{jj}$. Let $Y_j = T_j(X_j)$ for $j = 1, \ldots d$ then $Y_j \sim N(\mu_j, \sigma_j)$ and $\mathbf{Y} = (Y_1, \ldots, Y_d)^\top \sim N_d(\mathbf{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = \left(\frac{\sigma_{ij}}{\sigma_i \sigma_j}\right)_{i,j=1,...,d}$ is the correlation matrix associated with $\boldsymbol{\Sigma}$. Again by Sklar's theorem there exists a unique copula $C^{Ga}_{\boldsymbol{\Psi}}$, called the Gaussian copula such that distribution function $F_{\mathbf{Y}}$, of $\mathbf{Y}$ may be written as,

$$F_{\mathbf{Y}}(y_1, \ldots, y_d) = C^{Ga}_{\boldsymbol{\Psi}}(\Phi(y_1), \ldots, \Phi(y_d)),$$

where Φ is the standard normal c.d.f. Since T_j is strictly increasing on $\mathbb{R}$, by theorem (13) we may conclude $C_{\mathbf{X}} = C^{Ga}_{\boldsymbol{\Psi}}$. In addition (3.3.2) allows us to write down an explicit expression for the Gaussian copula.

$$\begin{aligned}
C_{\mathbf{X}} &= C^{Ga}_{\boldsymbol{\Psi}} \\
&= F_{\mathbf{Y}}(y_1, \ldots, y_d) \\
&= F_{\mathbf{Y}}\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\right) \\
&= \frac{1}{(2\pi)^{d/2}\sqrt{|\boldsymbol{\Psi}|}} \int_{-\infty}^{\Phi^{-1}(u_1)} \cdots \int_{-\infty}^{\Phi^{-1}(u_d)} \exp\left(-\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Psi}^{-1}\mathbf{x}\right) dx_1 \ldots dx_d
\end{aligned}$$

Thus any multivariate Gaussian distribution may be constructed from the Gaussian copula and the associated marginals. Similar methods may be used to write down an expression for other implicit copulae such as the Student-t copula.

The primary role of quantile functions in copula theory, is in the simulation of the random vector $\mathbf{X} = (X_1, \ldots, X_d)^\top$ with joint distribution defined by the copula $C_{\mathbf{X}}$ and d marginal distributions $F_1, \ldots, F_d$, which we discuss next. To this end we introduce the concept of conditional copula distributions. Assume $j \in 2, \ldots, d-1$,

Algorithm 1 Conditional Sampling

1. generate d independent pseudo random variables $v_1, \ldots, v_d$ uniformly distributed on $[0, 1]$.
2. for $j = 1, \ldots, d$ set $u_j = C_j^{-1}(u_j | u_1, \ldots, u_{j-1})$; the pseudo random variables $u_1, \ldots, u_d$ will have standard uniform marginal distributions and a dependence structure given by the copula $C_{\mathbf{X}}$.
3. for $j = 1, \ldots, d$ set $x_j = Q_j(u_j)$; the pseudo random variables $x_1, \ldots, x_d$ will have marginal distributions given by $F_1, \ldots, F_d$ respectively and dependence structure given by the copula $C_{\mathbf{X}}$.

then the conditional copula is defined as the conditional distribution of U_j, given the values of $U_1, \ldots, U_{j-1}$,

$$C_j(u_j | u_1, \ldots, u_{j-1}) = \mathbb{P}(U_j \leq u_j | U_1 = u_1, \ldots, U_{j-1} = u_{j-1}),$$

and the j-dimensional marginal copula C_j is defined as,

$$C_j(u_1, \ldots, u_j) = C_{\mathbf{X}}(u_1, \ldots, u_j, 1, \ldots, 1).$$

Denote by c_j^k the derivative of the j-dimensional marginal copula C_j, with respect to the first $k \leq j$ arguments,

$$c_j^k(u_1, \ldots, u_j) = \frac{\partial^k C_j(u_1, \ldots, u_j)}{\partial u_1, \ldots, \partial u_k}.$$

Keeping in mind the copula $C_j(u_1, \ldots, u_j)$ is the joint distribution of the j uniform random variables $U_1, \ldots, U_j$, and using the continuity property of probability, we may prove the following identity, see Franke et al. (2011, p. 347), relating the conditional copula to the partial derivatives of the marginal copulas C_j and C_{j-1},

$$C_j(u_j | u_1, \ldots, u_{j-1}) = \frac{c_j^{j-1}(u_1, \ldots, u_j)}{c_{j-1}^{j-1}(u_1, \ldots, u_{j-1})}.$$

This identity may be used in simulating the random vector $\mathbf{X} = (X_1, \ldots, X_d)^\top$ as described above using algorithm 1.

Although quite general the efficiency of the conditional sampling method depends on the availability of analytic expressions for the inverse of the conditional copula C_j^{-1} and the quantile functions Q_j, for $j = 1, \ldots, d$. If such expressions are not available or do not exist, numerical root finding techniques are usually employed, which result in slow sampling procedures. For this reason more efficient algorithms have

Algorithm 2 Sampling from a Gaussian Copula

Input: the correlation matrix $\mathbf{\Psi}$.
Output: A set of pseudo random numbers $u_1, \ldots, u_d$ with joint distribution function specified by the Gaussian copula $C^{Ga}_{\mathbf{\Psi}}$.

1. Perform the Cholesky decomposition $\mathbf{\Psi} = \mathbf{A}^T\mathbf{A}$.
2. Generate independent and identically distributed standard Gaussian deviates, $\mathbf{Y} = (Y_1, \ldots, Y_d)^T$.
3. Set $\mathbf{X} = (X_1, \ldots, X_d) = \mathbf{AY}$.
4. Return $U_i = \Phi(X_i)$ for $i = 1, \ldots, d$.

been devised for sampling from specific types of copulae, such as Gaussian, Student-t, Archimedean etc. The crucial point is that all these algorithms return a set of uniform distributed pseudo random variables $u_1, \ldots, u_d$ with the desired dependency structure just like step 2 of algorithm 1. To generate pseudo random variables $x_1, \ldots, x_d$ with the required marginals step 3 must be applied, for which knowledge of the associated quantile functions $Q_1, \ldots, Q_d$ is essential. For example algorithm 2 describes an efficient method to generate pseudo random variables $u_1, \ldots, u_d$ from a Gaussian copula, in order to construct the sample $x_1, \ldots, x_d$ a means to compute the marginal quantiles $Q_1, \ldots, Q_d$ is absolutely necessary. For further details on simulation procedures see for example, Embrechts et al. (2003), Franke et al. (2011, p. 347) and Schmidt (2006).

Chapter 4

Existing Methods

We now look at some of the existing methods used to evaluate the u-quantile $x_u = Q_X(u)$ of a continuous random variable X. This is a vast research area and it is difficult to provide a review of all the literature. However most of the techniques utilized are extensions of common techniques used in numerical analysis, such as root finding, rational approximations, interpolation and series expansions, which we discuss in the following sections from a historical perspective. In the spirit of this thesis we will discuss existing series expansions in some detail, in particular the much celebrated series expansion due to Cornish and Fisher is derived in section 4.2.

4.1 Numerical Techniques

Since an analytic expression for the quantile function is not known in the case of most distributions, numerical methods are often deployed, some of which we discuss in this subsection. Good sources of these techniques in the context of Quantile functions include Kennedy and Gentle (1980, Ch. 5), Lange (2010, Ch. 5) and Devroye (1986). We will focus on continuous distributions, since if a distribution is concentrated on a finite set of points and has no parametric dependencies, it is then a matter of tabulating the c.d.f. and employing a search algorithm to find the required u-quantile, see for example Bratley et al. (1987, § 5.2.1).

4.1.1 Root Finding

Suppose the c.d.f. F is continuous and strictly monotone, and that we have the means of computing $F(x)$. By definition the u-quantile x_u of F satisfies the equation $F(x_u) = u$. Define the function $g(x) := F(x) - u$. Then the unique root x_u of g in $\mathbb{R}$ can be found by standard root finding techniques. Probably the most famous technique is Newton's method. It is an iterative technique in which approximations x_k of the solution x_u take the form

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}.$$

A variation of Newton's method is the secant method in which successive approximations are given by

$$x_{k+1} = x_k - \frac{g(x_k)(x_k - x_{k-1})}{(g(x_k) - g(x_{k-1}))}.$$

It is useful when the evaluation of $g'(x)$ is costly. However both these methods suffer from slow convergence when applied to strangely shaped functions, for example when the c.d.f. changes slope too rapidly, corresponding to densities with several modes and especially high spikes, Hörmann et al. (2004, § 7.1). As a final resort the Bisection method may be applied which is known to converge for any c.d.f. even if not continuous at the cost of slower rates of convergence. Of course a hybrid algorithm could be designed, for example to start with the secant method and to resort to the Bisection method when the rate of convergence decreases. In addition root finding techniques are often used to refine the output of another algorithm. For example Acklam (2009) constructs a rational approximant to the standard normal quantile with a relative accuracy no larger than 1.15×10^{-9} and then applies one iteration of Halley's method (a third order root finding scheme) to compute the quantile to full machine precision.

Fixed point iteration or functional iteration may also be used to find the root of the function g. In this case define the function $h(x) := xF(x)/u$, then it is easy to see the fixed point x_u of h is a root of g. Many fixed point algorithms however require for convergence that there exists a constant K such that $|h'(x)| < K < 1$ in a neighbourhood of the fixed point. Clearly h violates this condition. Farnum (1991) however suggests to use Steffensen's acceleration technique which does not require the derivative of h to be bounded above and below by unity, $|h'(x)| < 1$.

4.1.2 Rational Approximation

The idea behind this approach is to approximate the quantile function Q by a function of the form,

$$Q(x) = \frac{a_0 + a_1 x + \cdots + a_n x^n}{b_0 + b_1 x + \cdots + b_n x^n} + e(x), \tag{4.1.1}$$

on the interval domain $(0, 1)$ where $e(x)$ is an error term satisfying some accuracy bound, say $e(x) < \epsilon$.

To keep round off error to a minimum the polynomials appearing in the numerator and denominator are usually evaluated according to Horner's rule. There are several examples in the literature of rational function approximations of the normal quantile see Odeh and Evans (1974), Beasley and Springer (1977), Moro (1995) and Acklam (2009). It is not necessary to approximate Q on its domain by one rational approximation, we may split the domain and use several rational approximations. This was the approach taken in the above references. Recently Shaw et al. (2011) through the

use of a clever change of variable, developed a single rational approximation for the normal quantile. The result is an algorithm that is branch-less and well suited for GPU architectures. Consequently their technique is gaining popularity amongst the finance community. Another algorithm well suited for GPU architectures is provide by Giles (2010) which is employed by the current CUDA library. The coefficients appearing in (4.1.1) can be computed using the second algorithm of Remes or the differential correction algorithm, see Press et al. (2007) or Ralston and Rabinowitz (2001). We will have much more to say about minimax and rational approximations in general in chapter 8.

4.1.3 Interpolation

References for this section are Hörmann et al. (2004, § 7.2) and Bratley et al. (1987, § 5.2.2). The simplest approach is as follows, tabulate a sequence of pairs $(F(x_i), x_i)$, this is normally achieved through some method of numerical integration, and may require significant set-up time. Using a search algorithm find to x_i such that $F(x_i) \leq u \leq F(x_{i+1})$. Next use some interpolation technique to find a more accurate approximation. A more sophisticated approach was provided by Ahrens and Kohrt (1981), which can be described as follows,

1. numerically integrate the p.d.f. to obtain F;
2. decide how to partition $(0, 1)$ the domain of Q, into unequal length intervals based on the behaviour of F;
3. determine the form and the coefficients of the interpolating function for each such interval.

Although the set-up time for this procedure is large Ahrens and Kohrt claim that if many variates need to be generated, then not including the set-up time required the speed and accuracy of their method is as good as the best available methods tailored for specific distributions. There is one notable drawback of their approach however, the set-up requires the parameters of the distribution to be fixed, thus we must repeat the initialization procedure for each set of parameters we are interested in. An interesting extension of this technique however can be found in Derflinger et al. (2010).

4.1.4 Relationship between Distributions

If a relationship between two distributions exists, and the quantile function of one of the distributions is known, then sometimes it is possible to exploit this relationship to derive an expression for the quantile function of the second distribution in terms of the quantile function of the first distribution. In section 6.1 we provide a concrete example, in which the relationship between the beta and F distributions is utilized to derive an expression for the quantile function of the F distribution in terms of the quantile function of the beta distribution.

4.2 Cornish-Fisher Expansions

Probably the best known series representation of the quantile function is the Cornish Fisher expansion introduced in Cornish and Fisher (1938) and Fisher and Cornish (1960). Let F_T be the c.d.f. of an arbitrary distribution which we will call the *target* distribution and F_B be the c.d.f. of some chosen *base* distribution. In their original papers Cornish and Fisher set the base distribution as $F_B = \Phi$, where Φ is the c.d.f. of the standard normal distribution. Later Hill and Davis (1968) generalized the Cornish Fisher expansion by allowing the base distribution to be arbitrary, not necessarily normal. In this section we will derive the generalized Cornish-Fisher expansion and discuss its deficiencies. Let x and z be the u quantiles of F_T and F_B respectively, the idea is to solve the equation,

$$F_T(x) = F_B(z). \tag{4.2.1}$$

for x. Note that (Shaw et al., 2011) also tackle the problem of solving (4.2.1), albeit through a much different approach, which is discussed in chapter 6. Cornish and Fisher provided an asymptotic series expansion of x based on polynomials in z whose coefficients are the cumulants of the target distribution. The derivation of Hill and Davis is based on Lagrange's inversion formula, see for example (Whittaker and Watson, 1927, p. 133). We reproduce the theorem here for convenience.

Theorem 15. *Let f and g be functions of z analytic inside and on a contour C surrounding a point a. Let $t \in \mathbb{C}$ be a point such that the inequality*

$$|tg(z)| < |z - a|$$

is satisfied at all points z on the perimeter C, then the equation

$$\xi = a + tg(\xi), \tag{4.2.2}$$

regarded as an equation in ξ has one root in the interior of C. Further more any function of ξ analytic inside and on C can be expanded as a power series in t by the formula,

$$f(\xi) = f(a) + \sum_{n=1}^{\infty} \frac{t^n}{n!} D_a^{n-1} \left[f'(a) (g(a))^n \right], \tag{4.2.3}$$

where D_x is the differential operator defined by $D_x = \frac{d}{dx}$.

What theorem 15 says is that if a function ξ of t is implicitly defined as in (4.2.2) with respect to a parameter a then any analytic function f of ξ can be expressed as a power series in t. Let $A(x) := F_T(x) - F_B(x)$ then we may rewrite (4.2.1) as,

$$F_B(x) = F_B(z) - A(x). \tag{4.2.4}$$

By defining the variables

$$a = F_B(z), \qquad \xi = F_B(x)$$

and denoting by Q_B the quantile function associated with F_B, (4.2.4) may be written as

$$\xi = a - A(Q_B(\xi)),$$

which is of the form (4.2.2) with $g = A \circ Q_B$. Hence we may solve for $Q_B(\xi) = x$ using Lagrange's inversion formula. Setting $t = -1$ yields,

$$Q_B(\xi) = Q_B(a) + \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} D_a^{n-1} \left[Q_B'(a) (A \circ Q_B(a))^n \right].$$

Substituting $\frac{dz}{da} D_z$ for D_a this may be written in the original variables,

$$x = z + \sum_{n=1}^{\infty} \frac{(-1)^{-n}}{n!} \left(\frac{1}{f_B(z)} D_z \right)^{n-1} \left[\frac{(A(z))^n}{f_B(z)} \right],$$

which can be written as,

$$x = z + \sum_{n=1}^{\infty} \frac{(-1)^{-n}}{n!} D_{(n-1)} \left[\left(\frac{F_T - F_B}{f_B}(z) \right)^n \right], \tag{4.2.5}$$

where $D_{(n)} := \left(D + \frac{f_B'}{f_B}\right)\left(D + 2\frac{f_B'}{f_B}\right) \cdots \left(D + n\frac{f_B'}{f_B}\right)$. To see this note,

$$\begin{aligned}
D_{(n-1)} \left[\left(\frac{F_T - F_B}{f_B} \right)^n \right] &= \left(D + \frac{f_B'}{f_B}\right) \cdots \left(D + (n-1)\frac{f_B'}{f_B}\right) \left[\left(\frac{F_T - F_B}{f_B} \right)^n \right] \\
&= \left(f_B^{-1} D f_B^1\right) \ldots \left(f_B^{-(n-1)} D f_B^{(n-1)}\right) \left[\left(\frac{F_T - F_B}{f_B} \right)^n \right] \\
&= f_B^{-1} D f_B^1 \cdots f_B^{-(n-1)} D f_B^{(n-1)} \left[\left(\frac{F_T - F_B}{f_B} \right)^n \right] \\
&= f_B^{-1} D f_B^{-1} D f_B^{-1} \cdots f_B^{-1} D \left[f_B^{(n-1)} \left(\frac{F_T - F_B}{f_B} \right)^n \right] \\
&= \left(f_B^{-1} D\right)^{n-1} \left[\frac{(F_T - F_B)^n}{f_B} \right],
\end{aligned}$$

where we have used the operator identity $D + n\frac{f_B'}{f_B} = f_B^{-n} D f_B^n$. Equation (4.2.5) is

called the generalized Cornish-Fisher expansion, it provides an approximation to the function $x = Q_T(F_B(z))$, where Q_T is the quantile function associated with F_T. It is shown by Hill and Davis in (Hill and Davis, 1968) that in the case $F_B = \Phi$, (4.2.5) reduces to the standard Cornish Fisher equation. Let $\tilde{Q}_T$ denote an approximation of Q_T obtained by truncating (4.2.5) after a finite number of terms. Although widely used in the finance for VAR calculations, we note some of the drawbacks of the Cornish Fisher expansion:

- $\tilde{Q}_T$ is not necessarily monotone.
- $\tilde{Q}_T$ has the wrong tail behaviour, that is the approximations of the u-quantiles become less reliable as $u \to 0$ or $u \to 1$.
- The series is asymptotic, thus the approximations do not necessarily improve by increasing the order n of the approximation.

These properties may have serious consequences, for example the loss of monotonicity may lead to a situation where the 99% VAR is less than the 95% VAR! See Jaschke (2002) for further discussion of these deficiencies. In the following sections we will derive series expansions of our own, aiming to overcome some of these inadequacies.

Remark 16. Suppose θ_n is a sequence of statistical estimators for a parameter of interest. Let F_n be the distribution function of θ_n and assume θ_n converges in distribution to the standard normal as $n \to \infty$. The motivation behind the Cornish Fisher expansion was to estimate the quantile of F_n so that we may compute confidence intervals for θ_n. When the sequence of distributions F_n converge to the standard normal, the Cornish Fisher approximation provides a better approximation than the normal approximation itself, see Jaschke (2002).

4.3 Orthogonal Expansions

In this section we will develop an orthogonal expansion for the target quantile function Q_T as proposed by Takemura (1983). Consider the set of distribution functions with finite second moment,

$$\Gamma_2 = \left\{ F : \int_{-\infty}^{\infty} x^2 dF < \infty \right\}.$$

By associating each F with its associated quantile function Q, the set Γ_2 is mapped into $L^2[0,1]$ the set of square integrable functions on $(0,1)$ since,

$$\int_0^1 Q(u)^2 du = \int_{-\infty}^{\infty} x^2 dF < \infty.$$

Now let $F_B \in \Gamma_2$ and consider the set of square integrable functions with respect to F_B on a set $A \subseteq \mathbb{R}$,

$$L^2(A; F_B) = \left\{ g : \int_A g(x)^2 \, dF_B(x) < \infty \right\}.$$

The primary theorem of Takemura's paper, states that if we can find an orthonormal basis $\{\psi_i\}_{i=0}^{\infty}$ for the space $L^2(A, F_B)$ then $\{\psi_i \circ Q_B\}_{i=0}^{\infty}$ will serve as an orthonormal basis for the space $L^2[0,1]$, where Q_B is the quantile function associated with F_B.

Example 17. Let $F_B = \Phi$ be the distribution function of the standard normal distribution and $A = \mathbb{R}$, then the space $L^2(\mathbb{R}, \Phi)$ contains those functions for which,

$$\int_{\mathbb{R}} g(x)^2 \, d\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x)^2 \, e^{-x^2/2} dx < \infty.$$

It is well known that these functions form a Hilbert space for which the Hermite polynomials $H_n(x)$ form an orthonormal basis. Hence we have that $\left\{ H_n(Q_B(u)) / \sqrt{n!} \right\}_{n=0}^{\infty}$, where $Q_B(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1)$, is an orthonormal basis for the space $L^2[0,1]$.

Elementary Fourier theory then tells us for any target distribution function $F_T \in \Gamma_2$, we may expand the associated quantile function Q_T as

$$Q_T(u) = \sum_{i=0}^{\infty} a_i \psi_i(Q_B(u)), \tag{4.3.1}$$

where

$$a_i = \langle Q_T, \psi_i \circ Q_B \rangle = \int_0^1 Q_T(u) \, \psi_i(Q_B(u)) \, du. \tag{4.3.2}$$

See Takemura's original paper for an exact statement of the theorem. The crux of the matter then is to find a base distribution for which we can find an orthonormal basis of the set $L^2(A, F_B)$. Takemura gives two other examples of base distributions where this is possible, namely the standard exponential and uniform distributions. In these cases it turns out that the set of Laguerre and Legendre polynomials respectively form an orthonormal basis of $L^2(A, F_B)$.

Takemura's approach is certainly an elegant one, in that he manages to apply classical Fourier theory to express the target quantile function Q_T in terms of a base quantile function Q_B, analogous to the generalized Cornish Fisher expansion. Unlike the Cornish Fisher expansion however (4.3.1) is a convergent series. From a practical perspective the primary setbacks with this approach are that: 1) the convergence of (4.3.1) is in the L^2 norm, and 2) the coefficients a_i usually cannot be computed analytically so one must resort to numerical quadrature. Thus it may happen that approximating Q_T by truncating (4.3.1) the average error is small while the error in the tails

is large. For this reason the L^∞ norm is preferred, so if one must resort to numerical quadrature to compute the coefficients a_i we would imagine that it be more worthwhile to approximate the Chebyshev coefficients. We will take up the topic of approximating the Chebyshev coefficients in chapter 8.

Chapter 5

Solution Techniques

In subsequent sections we will deal with nonlinear ordinary differential equations which in general are known to be notoriously difficult to solve. Thus in modern times many advanced techniques have been developed to deal with such problems. For example Adomian's decomposition method (Adomian, 1990), He's variational iteration method (He and Wu, 2007) and the Lie Group method (Olver, 2000) to name a few, have proved useful in the literature. We have applied some of these techniques to our problem, but find none more effective than two of the oldest methods of solution, namely the method of undetermined coefficients and the method of successive differentiation, see Tenenbaum and Pollard (1985). However in order to apply these methods a set of identities is required to handle the non linear terms appearing in the differential equation. Both of these methods yield recursive definitions of the coefficients appearing in a Taylor series expansion for the solution of the differential equation,

$$y'(x) = f(x, y). \tag{5.0.1}$$

The reader will of course recognize this as Taylor's method of numerical integration. Note that most texts on numerical analysis and applied mathematics, see for example the classics Henrici (1964) and Gear (1971), state that Taylor's method is primarily only of theoretical concern, and can only be applied in practice to very special cases. This is because the analytic determination of the Taylor series coefficients, i.e. the calculation of the higher order derivatives was deemed too complicated in most cases. For this reason Taylor series techniques were not pursued further in favour of other algorithms. However with the advent of digital computers and in particular automatic differentiation (not to be confused with numerical or symbolic differentiation) this argument no longer holds. Before continuing let us be clear what we mean by these three types of differentiation:

- Numerical differentiation is a crude way of approximating derivatives through difference quotients prone to cancellation and truncation errors.

- Symbolic differentiation is the study of algorithms and software for the manipulation of mathematical expressions according to the usual rules of calculus. No value is assigned to variables in the resulting expression (hence the term symbolic), an exact symbolic formula for the derivative is produced. The biggest drawback of symbolic differentiation that it may be both memory and time consuming.

- Automatic differentiation "is a recursive procedure to compute the value of derivatives of certain functions at a given point" see Jorba (2005) and references therein. We will discuss the main tools of automatic differentiation below, as well as what we see as a small extension to the current automatic differentiation tool set found in the literature, namely the use of theorem 24. The key attraction for automatic differentiation is that it does not suffer from truncation error and is fast in comparison to symbolic differentiation.

By the mid fifties J.C.P. Miller had developed a general recurrent power series method to handle a certain class of differential equations (Halin, 1983). However at that time such software was considered "exotic" and did not receive much attention. Indeed according to Rall (1981) a 1962 article by Hanson et al. (1962) titled "Analytic Differentiation by a Computer" was only published in the "Unusual Applications" section of the communications of the ACM. It is due to this neglect that recurrent Taylor series methods were reinvented by several other researchers who were not aware of Miller's contributions (Halin, 1983). Today several authors report that when high precision is required Taylor's method is the method of choice, see for example Corliss and Chang (1982), Barrio et al. (2005) and Jorba (2005). Hence it has been applied to a wide range of problems ranging from astronomical problems to problems in nuclear reactor physics. It can also be coupled with interval arithmetic (Moore et al., 2009) to create a numerical integrator capable of producing guaranteed results. That is interval arithmetic allows us to obtain error bounds that account for both the truncation and round off errors, as opposed to algorithms implemented in floating point arithmetic in which the round off error is difficult to analyse and often ignored. Applied to our problem (at least in theory) this would allow us to produce validated approximations to the quantile function, in which both the round off and truncation errors have been accounted for. However this is not an area of research we have pursued further.

5.1 Taylor's Method

The Taylor series method is one of the oldest known methods for the numerical integration of (5.0.1). It consists of performing a Taylor series expansion of $y(x)$ around some point x_0 at which the value of y is known. The truncated Taylor polynomial,

$$y(x) \approx \sum_{n=0}^{N} \frac{y^{(n)}(x_0)}{n!} (x - x_0)^n , \tag{5.1.1}$$

will then be a useful approximation of y in some neighbourhood of x_0. There are two common techniques for computing the coefficients appearing in (5.1.1),

1. The method of undetermined coefficients: assumes that the solution takes the form,

$$y(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n . \tag{5.1.2}$$

   Then a recursive identity for the coefficients a_n in (5.1.2) is obtained by substituting this infinite sum into (5.0.1) and equating the powers in $(x - x_0)$. Note that it is precisely at this step the recursive identities of section 5.2 will prove invaluable in dealing with the nonlinear terms.

2. The method of successive differentiation: makes use of the fact that the coefficients appearing in (5.1.1) can be obtained by repeatedly differentiating (5.0.1), in particular we have,

$$\begin{aligned} y &= y(x_0) \\ y' &= f \\ y'' &= f_x + f_y f \\ y^{(3)} &= f_{xx} + 2f_{xy} f + f_x f_y + f_{yy} f^2 f \\ \vdots &\ \vdots\ \vdots \end{aligned}$$

   Continuing in this way it soon becomes apparent the complexity of computing the derivatives leads to an unmanageable solution process. It is precisely for this reason Taylor's method has been sidelined in favour of other numerical integration techniques. However with the advent of automated differentiation this is no longer the case. We refer to Rall (1981) for an introduction.

5.2 Recursive Identities

In this section we introduce some useful identities used in the manipulation of power series. These will be crucial in handling the non-linear terms present in the differential equations (6.0.2) and (6.0.3). Note that based on these identities several programs, see for example Lara et al. (1999) or Jorba (2005), have been written which take as input

the right hand side of (5.0.1) and produce as output a stream of Taylor coefficients. The drawback of these programs is that they work for a specific class of differential equations. Some of the differential equations we deal with in this thesis fall outside of this class. Hence we will give explicit expressions for the coefficients appearing in the Taylor series solutions. Consequently this alleviates any dependency on external libraries to construct the required solution. On the other hand if one wishes to construct a program which is capable of producing approximations for a wide variety of quantile functions when only the density function is known as in Ulrich and Watson (1987) or Derflinger et al. (2010), such programs would surely prove useful. For now we are more interested in dealing with specific cases so do not pursue this idea further.

All of the recursive identities we will employ can be derived from a classical result in analysis.

Definition 18. Given two sequences$\{a_n\}_{n=0}^{\infty}$ and $\{b_n\}_{n=0}^{\infty}$ the convolution of the sequences is defined as

$$c_n = \sum_{i=0}^{n} a_i b_{n-i}, \quad n = 0, 1, \ldots$$

The series $\sum_{n=0}^{\infty} c_n$ is called the Cauchy product of the series $\sum_{n=0}^{\infty} a_n$ and $\sum_{n=0}^{\infty} b_n$.

Mertens' theorem states the conditions under which the Cauchy product converges.

Theorem 19 (Mertens)**.** *If $\sum_{n=0}^{\infty} a_n$ converges and $\sum_{n=0}^{\infty} b_n$ converges absolutely then*

$$\left(\sum_{n=0}^{\infty} a_n\right)\left(\sum_{n=0}^{\infty} b_n\right) = \sum_{n=0}^{\infty} c_n$$

where $\{c_n\}_{n=0}^{\infty}$ is the convolution of $\{a_n\}_{n=0}^{\infty}$ and $\{b_n\}_{n=0}^{\infty}$.

Proof. See Apostol (1974, p. 204) □

A direct consequence of this theorem allows us to find a power series representation of the product of two analytic functions.

Theorem 20 (Product)**.** *Suppose f and g are analytic functions on the open balls $B(0; r)$ and $B(0; R)$ respectively, with power series representations about the origin given as $f(z) = \sum_{n=0}^{\infty} a_n z^n$ and $g(z) = \sum_{n=0}^{\infty} b_n z^n$. Then*

$$f(z)\, g(z) = \sum_{n=0}^{\infty} c_n z^n, \quad \forall z \in B(0, r) \cap B(0, R)$$

where

$$c_n = \sum_{i=0}^{n} a_i b_{n-i}, \quad n = 0, 1, \ldots$$

Proof. See Apostol (1974, p. 237). □

With this theorem at hand it is an easy exercise to find a power series representation for the reciprocal of f and the ratio of two functions f and g, provided $g(z) \neq 0$.

Theorem 21 (Reciprocal)**.** *Suppose f is as defined in theorem 20 and in addition $f(0) \neq 0$. Then there exists a neighbourhood $B(0;\delta)$ in which the reciprocal of f has the series representation*

$$\frac{1}{f(z)} = \sum_{n=0}^{\infty} b_n z^n, \quad \forall z \in B(0,\delta)$$

where

$$b_n = \begin{cases} \frac{1}{a_0}, & n = 0 \\ -\frac{1}{a_0} \sum_{i=1}^{n} a_i b_{n-i}, & n \geq 1 \end{cases}.$$

Proof. See Apostol (1974, p. 239). □

Theorem 22 (Quotient)**.** *Suppose f and g are defined as in theorem 20 and in addition $g(0) \neq 0$. Then there exists a neighbourhood $B(0;\delta)$ in which the ratio f/g has the series representation*

$$\frac{f(z)}{g(z)} = \sum_{n=0}^{\infty} c_n z^n, \quad \forall z \in B(0,r) \cap B(0,R)$$

where

$$c_n = \begin{cases} \frac{a_0}{b_0}, & n = 0 \\ \frac{1}{b_0}\left(a_n - \sum_{i=1}^{n} b_i c_{n-i}\right), & n \geq 1 \end{cases}.$$

Proof. Write

$$\sum_{n=0}^{\infty} a_n z^n = \left(\sum_{n=0}^{\infty} c_n z^n\right)\left(\sum_{n=0}^{\infty} b_n z^n\right).$$

The result is then a direct consequence of theorems 20. □

The next theorem can be used to find a power series representation of integer powers of an analytic function g. Let $b_n^{(k)}$ denote the n^{th} coefficient of the power series expansion of $[g(z)]^k$, for some integer power $k \geq 2$. Assume that b_m is the first non zero coefficient in the expansion of $g(z)$, then a recursive relation for the coefficients $b_n^{(k)}$ is given in the next theorem.

Theorem 23. *Let $k \in \{2,3,\ldots\}$ and suppose g is an analytic function on the open ball $B(0;R)$, with power series representation about the origin given by $g(z) = \sum_{n=m}^{\infty} b_n z^n$, $m \geq 0$. Then*

$$[g(z)]^k = \sum_{n=0}^{\infty} b_n^{(k)} z^n, \quad \forall z \in B(0,R)$$

where

$$b_n^{(k)} = \begin{cases} 0 & n < km \\ b_m^k & n = km \\ b_m^{k-1} \sum_{j=1}^{n-k} \left(\left(\frac{k+1}{n-km} \right) j - 1 \right) b_{j+m} b_{n-j}^{(k)} & n > km \end{cases}.$$

Proof. See Knuth (1998, p. 526). □

The coefficients $b_n^{(k)}$ are useful in finding a power series representation of the composition of two analytic functions $f(g(z))$. This is a powerful tool for solving autonomous equations.

Theorem 24 (Substitution)**.** *Suppose f and g are analytic functions on the open balls $B(0;r)$ and $B(0;R)$ respectively, with power series representations about the origin given as $f(z) = \sum_{n=0}^{\infty} a_n z^n$ and $g(z) = \sum_{n=0}^{\infty} b_n z^n$ with $b_0 = 0$. If for a fixed $z \in B(0;R)$, we have $\sum_0^{\infty} |b_n z^n| < r$, then for this z we can write*

$$f \circ g(z) = f(g(z)) = \sum_{n=0}^{\infty} c_n z^n$$

where

$$c_n = \begin{cases} a_0, & n = 0 \\ \sum_{k=1}^{n} a_k b_n^{(k)}, & n \geq 1 \end{cases}$$

and the coefficients $b_n^{(k)}$ are defined as in theorem 23.

Proof. For the convergence part see Apostol (1974, p. 239). The identity for the coefficients c_n is proved in Henrici (1974, p. 36). □

When f in theorem 24 is the exponential function, a relatively simple formula exists for the coefficients appearing in the power series expansion of $\exp\{g(z)\}$.

Theorem 25. *If g is an analytic function on the open ball $B(0;R)$, with power series representation about the origin given as $g(z) = \sum_{n=0}^{\infty} b_n z^n$, then for all $z \in B(0;R)$ we have*

$$e^{g(z)} = \sum_{n=0}^{\infty} a_n z^n$$

where

$$a_n = \begin{cases} e^{b_0}, & n = 0 \\ \frac{1}{n}\sum_{k=1}^{n} k b_k a_{n-k}, & n \geq 1 \end{cases}$$

and $b_n^{(k)}$ *are defined as in theorem 24.*

Proof. The convergence part follows from applying theorem 24 to $e^{g(z)-b_0}$. To derive the given expression for the coefficients a_n, let $h(z) = e^{g(z)}$ and note $h'(z) = g'(z)\, h(z)$. An application of theorem 20 produces the required result. □

Similarly we may derive coefficients for many other elementary functions. For example,

- if $f(z) = \ln(z)$ and g is a positive analytic function with $g(0) \neq 0$ we have

$$\ln(g(z)) = \sum_{n=0}^{\infty} a_n z^n$$

where

$$a_n = \begin{cases} \ln(b_0), & n = 0 \\ \frac{1}{nb_0}\left(nb_n - \sum_{k=1}^{n-1}(n-k)\, b_k a_{n-k}\right), & n \geq 1 \end{cases}.$$

- if $f(z) = z^{\alpha}$, $\alpha \in \mathbb{R}\setminus\{0\}$ and g is an analytic function with $g(0) \neq 0$ we have

$$(g(z))^{\alpha} = \sum_{n=0}^{\infty} a_n z^n$$

where

$$a_n = \begin{cases} b_0^{\alpha}, & n = 0 \\ \frac{1}{nb_0}\left(\sum_{k=0}^{n-1}(\alpha n - (\alpha+1)\, k)\, a_k b_{n-k}\right), & n \geq 1 \end{cases}.$$

Formulas for other elementary functions such as the trigonometric functions may be found in Hairer et al. (1993).

Remark 26. The number of arithmetic operations required to compute the first n coefficients $b_1, \ldots, b_n$ in theorem 21 is $O(n^2)$. To see this note that the number of operations required to compute the k^{th} coefficient once we know $b_0, b_1, \ldots, b_{k-1}$ is $O(k)$. Hence the total number of operations required to compute b_n is $\sum_{k=1}^{n} O(k) = O(n^2)$. In a similar way we may conclude that the computational complexity of computing the first n coefficients in theorems 22, 23 and 25 is $O(n^2)$ and in theorem 24 is $O(n^3)$.

Chapter 6

Quantile Mechanics

For a given c.d.f. F denote by Q the associated quantile function. We restrict our attention to the class of distributions for which F is strictly increasing and absolutely continuous. In this case we have

$$Q(u) = F^{-1}(u) \tag{6.0.1}$$

where F^{-1} is the compositional inverse of F. Suppose the corresponding density function $f(x)$ is known. Differentiating (6.0.1) we obtain the *first order quantile equation*,

$$\frac{dQ(u)}{du} = \frac{1}{F'(F^{-1}(u))} = \frac{1}{f(Q(u))}. \tag{6.0.2}$$

This is an autonomous equation in which the nonlinear terms are introduced through the density function f. Many distributions of interest have complicated densities, for example the densities of the generalized hyperbolic distributions are written in terms of higher transcendental functions. Thus the solutions to (6.0.2) are often difficult to find, and hence this route has been relatively unexplored in the literature (Ulrich and Watson, 1987). However provided the reciprocal of the density $1/f$ is an analytic function at the initial condition $x_0 = Q(u_0)$ we may employ some of the oldest methods of numerical integration of initial value problems, namely the method of undetermined coefficients and the method of successive differentiation, to determine the Taylor series expansion of Q. Since the equations in question are non-linear, finding their solutions requires some special series manipulation techniques, see chapter 5 for details.

Writing (6.0.2) as,

$$f(Q(u))\frac{dQ(u)}{du} = 1,$$

and applying the product rule with a further differentiation leads to the differential equation,

$$\frac{d^2Q}{du^2} = H\left(Q\left(u\right)\right)\left(\frac{dQ}{du}\right)^2, \tag{6.0.3}$$

where

$$H\left(x\right) = -\frac{d}{dx}\log f\left(x\right). \tag{6.0.4}$$

We will refer to (6.0.3) as the *second order quantile equation*. The dependence on the density f in (6.0.3) is now through the H-function (6.0.4). The benefit of this added differentiation is, as we shall see later, in some cases the nonlinear terms appearing in (6.0.3) are more manageable and hence elementary solution techniques may be applied.

This was the precisely the approach taken by Steinbrecher and Shaw (2008). They considered power series solutions to the second order quantile equation focusing on particular distributions belonging to the Pearson family (see appendix A). We will extend their work here by examining some non-Pearson distributions, in particular we will look at the cases where f is the density function of the hyperbolic, variance gamma, generalized inverse Gaussian and Snedecor's F distributions. In addition with the aid of the methods of chapter 5 we will primarily be concerned with solving the first order quantile equation.

In order to approximate the quantile function in the tail regions we will apply a change of variable and build approximants to the transformed function. To fix notation suppose we wish to generate a random variate X from a *target* distribution with continuous c.d.f. F_T and quantile function Q_T. If we generate a random variate Z from a *base* distribution with continuous c.d.f. F_B and quantile function Q_B, then we have from the inversion method

$$X = Q_T\left(F_B\left(Z\right)\right)$$

since $F_B\left(Z\right) \sim \mathrm{U}\left(0,1\right)$. Let us denote this composition by

$$A\left(z\right) := Q_T\left(F_B\left(z\right)\right). \tag{6.0.5}$$

The function $A\left(z\right)$ is obtained by applying the change of variable $u = F_B\left(z\right)$ to the target quantile function $Q_T\left(u\right)$. The target quantile may then be written as

$$Q_T(u) = A\left(Q_B\left(u\right)\right). \tag{6.0.6}$$

For later use we note that the inverse of A is given by

$$A^{-1}\left(z\right) = Q_B\left(F_T\left(x\right)\right).$$

There is no restriction on the choice of base distribution, however as will be seen below in searching for such changes of variable we were motivated by the asymptotic behaviour of Q_T.

We will now derive an ordinary differential equation describing the function A arising from the change of variable. Starting with the first order quantile equation for the target quantile function Q_T,

$$\frac{dQ_T(u)}{du} = \frac{1}{f_T(Q_T(u))}, \tag{6.0.7}$$

we make a change of variable $z = Q_B(u)$. A simple application of the chain rule for differentiation to (6.0.6) and using the fact that Q_B also satisfies the first order quantile equation gives

$$\frac{dQ_T(u)}{du} = \frac{dA}{dz}\frac{dz}{du} = \frac{dA}{dz}\frac{dQ_B}{du} = \frac{dA}{dz}\frac{1}{f_B(Q_B(u))} = \frac{dA}{dz}\frac{1}{f_B(z)}.$$

Substituting this identity into (6.0.7) we obtain a first order nonlinear differential equation

$$\frac{dA}{dz} = \frac{f_B(z)}{f_T(A(z))}, \tag{6.0.8}$$

which we call the *first order recycling equation.* Of course we could have obtained the same result by differentiating (6.0.5) with respect to z. A second order version of this equation was treated in Shaw et al. (2011), namely

$$\frac{d^2A}{dz^2} + H_B(z)\frac{dA}{dz} = H_T(A(z))\left(\frac{dA}{dz}\right)^2,$$

where

$$H_B(x) = -\frac{d}{dx}\ln(f_B(x)), \text{ and } H_T(x) = -\frac{d}{dx}\ln(f_T(x)).$$

Note that the idea of expressing target quantiles in terms of base quantiles is not a new one, indeed this is the idea behind the generalized Cornish Fisher expansion, see section 4.2 for a derivation and some of its drawbacks. Another interesting idea along these lines was introduced by Takemura (1983). Here the the Fourier series expansion of the target quantile Q_T is developed with respect to an orthonormal basis of the form $\{\psi_i \circ Q_B\}_{i=0}^{\infty}$, where $\{\psi_i\}_{i=0}^{\infty}$ is itself an orthonormal basis for a set of square integrable functions, we refer the reader to section 4.3 for details. Unlike the Cornish Fisher expansion which is asymptotic in nature Takemura's approach yields a convergent series in the L^2 norm. Note however the computation of the Fourier

coefficients usually requires numerical quadrature and that for approximation purposes the L^∞ norm is preferred.

Standard numerical techniques such as root finding and interpolation for approximating the associated quantile functions often fail, particularly in the "deep" tail regions of the distribution, see for example Derflinger et al. (2010). Thus in addition to developing convergent power series solutions to (6.0.2) and (6.0.6) we also develop asymptotic expansions of Q at the singular points $u = 0$ and $u = 1$. In general it is not easy to discover these asymptotic behaviours, some trial and error and intelligent fiddling is required, see Bender and Orszag (1978, Ch. 4). Our approach is to consider Q as being implicitly defined by the equation,

$$F(Q) = u.$$

Asymptotic iteration methods, see for example Bruijn (1981, Ch. 2) for an introduction, may then be used to obtain the leading terms in the expansion. The resulting series are divergent but as we will see they are still numerically useful.

In the following sections we will focus on the variance gamma, generalized inverse Gaussian, and hyperbolic distributions. For each of these distributions we will find power series expansions of Q and A as well as the asymptotic behaviour of Q near its singular points. Despite the unsightly appearance of the coefficients appearing in these expansions, they are simple to program if rather tedious.

All results in this chapter are derived with pen and paper with the following exceptions:

- Once the coefficients appearing in the Taylor series were derived (by hand) using the tools of chapter 5, they were programmed in Mathematica (Wolfram-Research, 2010) to generate the first few terms of each of the Taylor series we consider.

- The first few terms of the asymptotic expansion of the hyperbolic quantile were derived with the aid of Mathematica. From this we conjectured the form of the expansion and derived the coefficients by hand.

- Conversely the form of the asymptotic expansions for variance gamma and generalized inverse Gaussian quantile functions were derived by hand and a computer orientated recurrence scheme is then applied to compute the values of the coefficients appearing in these expansions. A computer algebra system such as Mathematica is required to execute this scheme.

It should be pointed out that Mathematica Wolfram-Research (2010) is a useful and powerful tool, but it is of little use to us in deriving the bulk of the results in this

chapter. For example Mathematica has a built in function for the hyperbolic quantile but it has no idea how to differentiate this function, so it can not be used to derive a Taylor series expansion of the function, see the remarks in the introduction of chapter 8 for details. Also in that chapter we provide an example which exposes a bug in Mathematica's implementation of the hyperbolic quantile. Similar comments apply to the generalized inverse Gaussian distribution. The situation is even worse for the variance gamma distribution since Mathematica Wolfram-Research (2010) does not have a built in function for the variance gamma quantile.

6.1 Beta, F and Student-t Distributions

In this subsection we will derive a series representation for the compositional inverse of the regularized incomplete beta function $\mathrm{I}^{-1}(u;\alpha,\beta)$ about the points $u = 0$ and $u = 1$. Then from the relationships between the regularized incomplete beta function $\mathrm{I}(x;\alpha,\beta)$ and the beta, F and student-t distributions we will derive expressions for the corresponding quantiles of each distribution. The results presented in this section are those of Steinbrecher and Shaw (2008), with one minor addition; we provide the expansion of $\mathrm{I}^{-1}(u;\alpha,\beta)$ at $u = 1$ and its link with Snedecor's F quantile. We begin our study with the beta distribution denoted $\mathrm{Beta}(\alpha,\beta)$, where α and β are positive shape parameters. The associated density f_{B} and c.d.f. F_{B} are given by,

$$f_{\mathrm{B}}(x;\alpha,\beta) = \frac{1}{\mathrm{B}(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad x \in [0,1],$$

and

$$F_{\mathrm{B}}(x;\alpha,\beta) = \mathrm{I}(x;\alpha,\beta),$$

respectively, where $\mathrm{B}(\alpha,\beta)$ is the beta function and $\mathrm{I}(x;\alpha,\beta)$ is the regularized incomplete beta function. To avoid confusion later on we will denote the beta quantile by $Q_B(u;\alpha,\beta)$, but for now for the sake of notational convenience we suppress the extra parametrization and simply denote the beta quantile by Q. From the density of the beta distribution we may write down the second order quantile equation (6.0.3) as

$$Q(1-Q)\frac{d^2Q}{du^2} = (1-\alpha+Q(\alpha+\beta-2))\left(\frac{dQ}{du}\right)^2.$$

Boundary conditions can then be imposed from the value of the c.d.f. at the endpoints of the support; $Q(0) = 0$ and $Q(1) = 1$. Of course to be able to apply the Taylor series method we need to formulate an initial value problem. To this end observe the asymptotic behaviour of Q. Note that the first order quantile equation (6.0.2) is given by,

$$\frac{dQ}{du} = \mathrm{B}\left(\alpha,\beta\right)\left(Q\right)^{1-\alpha}\left(1-Q\right)^{1-\beta},$$

from which we obtain the asymptotic relationship,

$$\frac{dQ}{du} \sim \mathrm{B}\left(\alpha,\beta\right)\left(Q\right)^{1-\alpha}, \quad u \to 0,$$

motivating the change of variable $v = \left(\alpha u \mathrm{B}\left(\alpha,\beta\right)\right)^{1/\alpha}$. In the new variable the problem may be stated as,

$$Q\left(1-Q\right)\frac{d^2Q}{dv^2} = \left(1-\alpha+Q\left(\alpha+\beta-2\right)\right)\left(\frac{dQ}{dv}\right)^2 - \frac{\left(1-\alpha\right)}{v}\frac{dQ}{dv},$$
$$Q\left(0\right)=0,\ Q'\left(0\right)=1,$$

where we have used the asymptotic relationship $Q\left(v\right) \sim v$ as $v \to 0$ to derive the second initial condition. We seek a power series solution of the form

$$Q_{\mathrm{B}}\left(v;\alpha,\beta\right) = \sum_{n=0}^{\infty} a_n\left(\alpha,\beta\right)v^n. \tag{6.1.1}$$

After some routine manipulations (see chapter 5) we obtain the following recurrence relation for the coefficients,

$$a_n = \begin{cases} 0, & n=0 \\ 1, & n=1 \\ \frac{b_n}{(n-1)^2+(n-1)\alpha}, & n \geq 2 \end{cases},$$

where

$$\begin{aligned} b_n &= \left(1-\alpha\right)\sum_{i=1}^{n-2}(i+1)(n-i)a_{i+1}a_{n-i} \\ &\quad +(\alpha+\beta-2)\sum_{i=1}^{n-1}\sum_{j=0}^{n-i-1}(j+1)(n-i-j)a_i a_{j+1}a_{n-i-j} \\ &\quad -\sum_{i=2}^{n-1}(n-i-\alpha+1)(n-i+1)a_{n-i+1}\left(a_i-\sum_{j=0}^{i-1}a_j a_{i-j}\right) \\ &\quad +(1-\alpha)\sum_{i=1}^{n-1}a_i a_{n-i}, \end{aligned}$$

and the dependence of the coefficients on the parameters α and β has been suppressed. The number of arithmetic operations required to compute the first n coefficients $a_1, \ldots, a_n$ is $O\left(n^2\right)$. The first few terms of the series are given by,

$$v + \frac{(\beta-1)v^2}{1+\alpha} + \frac{(\beta-1)\left(\alpha^2+\alpha(3\beta-1)+5\beta-4\right)v^3}{2(1+\alpha)^2(2+\alpha)} + \mathrm{O}\left(v^4\right).$$

We have in effect derived a Frobenius series expansion for the inverse of the regularized incomplete beta function $\mathrm{I}^{-1}\left(u;\alpha,\beta\right)$ about the point $u = 0$. For numerical purposes it would also be beneficial to have an expansion about the point $u = 1$. This is easily obtained through the identity,

$$\mathrm{I}\left(x;\alpha,\beta\right) = 1 - \mathrm{I}\left(1-x;\beta,\alpha\right),$$

from which it follows,

$$Q_B\left(u;\alpha,\beta\right) = 1 - Q_B\left(1-u;\beta,\alpha\right). \tag{6.1.2}$$

Thus the series expansion (6.1.1) along with the identity (6.1.2) provide us with good approximations of the beta quantile near the points $u = 0$ and $u = 1$ respectively. Rather than repeating the above procedure for the $\mathrm{F}\left(n_1,n_2\right)$ distribution we can utilize a well known relationship between the c.d.f.'s of the $\mathrm{Beta}\left(\alpha,\beta\right)$ and $\mathrm{F}\left(n_1,n_2\right)$ distributions,

$$F_{\mathrm{F}}\left(x;n_1,n_2\right) = F_{\mathrm{B}}\left(\frac{n_1 x}{n_1 x + n_2}; \frac{n_1}{2}, \frac{n_2}{2}\right) = \mathrm{I}\left(\frac{n_1 x}{n_1 x + n_2}; \frac{n_1}{2}, \frac{n_2}{2}\right).$$

This allows us to write the F quantile $Q_{\mathrm{F}}\left(u;n_1,n_2\right)$ in terms of the beta quantile $Q_{\mathrm{B}}\left(u;\frac{n_1}{2},\frac{n_2}{2}\right)$ as follows,

$$Q_{\mathrm{F}(n_1,n_2)}\left(u\right) = \frac{n_2 y}{n_1\left(1-y\right)}, \tag{6.1.3}$$

where $y = Q_{\mathrm{B}}\left(u;\frac{n_1}{2},\frac{n_2}{2}\right)$. We now have a method of approximating the quantile function of the $\mathrm{F}\left(n_1,n_2\right)$ distribution, in particular in the tail regions.

Denoting the c.d.f. and the quantile function of the student-t distribution with $\nu \in \mathbb{N}$ degrees of freedom by $F_{\mathrm{t}}\left(x;\nu\right)$ and $Q_{\mathrm{t}}\left(u;\nu\right)$ respectively. Utilizing the following relationship between the c.d.f.'s of the student-t and beta distributions

$$F_{\mathrm{t}}\left(x;\nu\right) = \begin{cases} \frac{1}{2}F_{\mathrm{B}}\left(\frac{\nu}{\nu+x^2};\frac{\nu}{2},\frac{1}{2}\right), & x \leq 0 \\ 1 - \frac{1}{2}F_{\mathrm{B}}\left(\frac{\nu}{\nu+x^2};\frac{\nu}{2},\frac{1}{2}\right), & x > 0 \end{cases},$$

we obtain the following relationship between the corresponding quantile functions,

$$Q_{\mathrm{t}}(u;\nu) = \begin{cases} -\left(\frac{\nu}{Q_{\mathrm{B}}\left(2u;\frac{\nu}{2},\frac{1}{2}\right)} - \nu\right)^{1/2}, & 0 \le u \le \frac{1}{2} \\ \left(\frac{\nu}{Q_{\mathrm{B}}\left(2(1-u);\frac{\nu}{2},\frac{1}{2}\right)} - \nu\right)^{1/2}, & \frac{1}{2} < u \le 1 \end{cases} \tag{6.1.4}$$

Note that this formula along with (6.1.1) provide a useful approximation of the student-t quantile in the tail regions. In addition Steinbrecher and Shaw (2008) have developed a series expansion about the point $u = 1/2$ which can be used for an approximation in the central regions. Taking into account the symmetry of the distribution this provides us with all the necessary ingredients to formulate an algorithm for approximating the student-t quantile.

6.2 Hyperbolic Distribution

We now turn to our first major contribution. In this section we will develop Taylor and asymptotic series expansions for the hyperbolic distribution. In addition we will choose a suitable base distribution and solve the recycling equation (6.0.8) when the target distribution is hyperbolic. In later sections we will repeat the analysis for the variance gamma and generalized inverse Gaussian distributions.

Amongst others it was observed by Eberlein and Keller (1995) and Bingham and Kiesel (2001) the hyperbolic distribution is superior to the normal distribution in modelling log returns of share prices. Numerical inversion of the hyperbolic distribution function was considered in Leobacher and Pillichshammer (2002). There numerical methods were considered to solve (6.0.2) and only the leading order behaviour of the left and right tails was given. Here we will provide an analytic solution to (6.0.2) and the full asymptotic behaviour of the tails.

6.2.1 Taylor Series

The density of the of the hyperbolic distribution $\mathrm{Hyp}(\alpha,\beta,\delta,\mu)$ (see example 37 on page 142) is given by,

$$f(x;\alpha,\beta,\delta,\mu) = \frac{\gamma}{2\alpha\delta K_1(\delta\gamma)} e^{-\alpha\sqrt{\delta^2+(x-\mu)^2}+\beta(x-\mu)}, \tag{6.2.1}$$

where $\alpha > 0$, $|\beta| < \alpha$, $\delta > 0$ and $\mu \in \mathbb{R}$ are shape, skewness, scale and location parameters respectively and for notational convenience we have set $\gamma = \sqrt{\alpha^2-\beta^2}$. By defining $\alpha_1 := \delta\alpha$ and $\beta_1 := \delta\beta$, we obtain an alternative parametrization in which α_1 and β_1 are now location and scale invariant parameters. Hence without loss of generality we may set $\mu = 0$ and $\delta = 1$, since $Q(u;\alpha_1,\beta_1,\mu,\delta) = \mu + \delta\, Q(u;\alpha_1,\beta_1,0,1)$. The first order quantile equation (6.0.2) then reads,

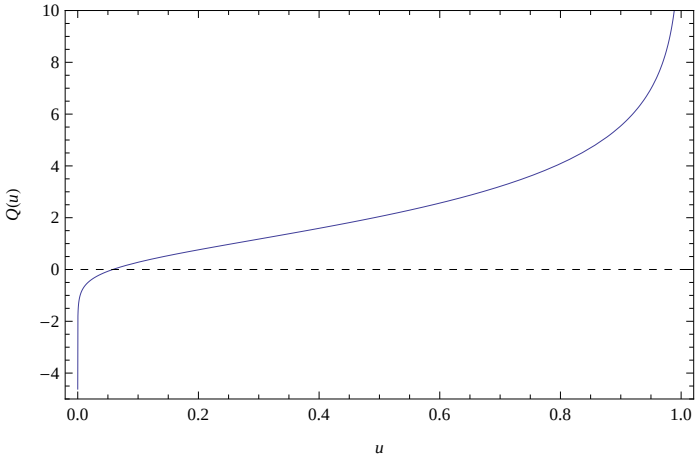


Figure 6.2.1: Graph of $Q_{\text{Hyp}}\left(u; 2, 3/2, 1, 0\right)$

$$\frac{dQ}{du} = N_0 e^{\alpha_1\sqrt{1+Q^2}-\beta_1 Q} \tag{6.2.2}$$

where $N_0 = 2\alpha_1 \mathrm{K}_1\left(\gamma_1\right)/\gamma_1$ and $\gamma_1 = \sqrt{\alpha_1^2 - \beta_1^2}$. To form an initial value problem let $u_0 \in (0,1)$ and impose the initial condition $Q\left(u_0\right) = x_0$. For all practical purposes x_0 is usually determined by solving the equation $F\left(x_0\right) - u_0 = 0$ using a root finding procedure (alternatively we may choose $x_0 \in \mathbb{R}$ and approximate u_0 by numerically integrating the density). By applying the method of undetermined coefficients we find Q admits the Taylor series expansion

$$Q\left(u\right) = \sum_{n=0}^{\infty} q_n (u-u_0)^n, \tag{6.2.3}$$

where the coefficients q_n are defined recursively as follows,

$$q_n = \begin{cases} x_0, & n = 0 \\ \frac{N_0}{n} b_{n-1}, & n \geq 1 \end{cases},$$

$$b_n = \begin{cases} e^{\alpha_1\sqrt{1+x_0^2}-\beta_1 x_0}, & n = 0 \\ \frac{1}{n}\sum_{k=1}^{n} k(\alpha_1 a_k - \beta_1 q_k) b_{n-k}, & n \geq 1 \end{cases},$$

and

$$a_n = \begin{cases} \sqrt{1+x_0^2}, & n = 0 \\ \frac{1}{na_0}\left(nq_n q_0 + \sum_{k=0}^{n-2}(k+1)(q_{k+1}q_{n-k-1} - a_{k+1}a_{n-k-1})\right), & n \geq 1 \end{cases}.$$

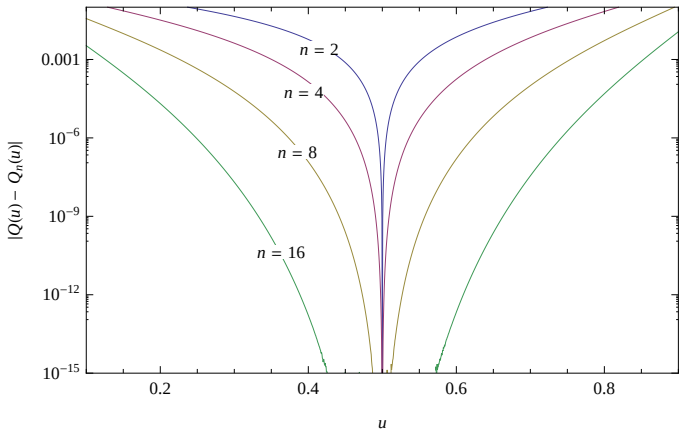


Figure 6.2.2: Error made by truncating (6.2.3) after $n = 2$, 4, 8 and 16 terms.

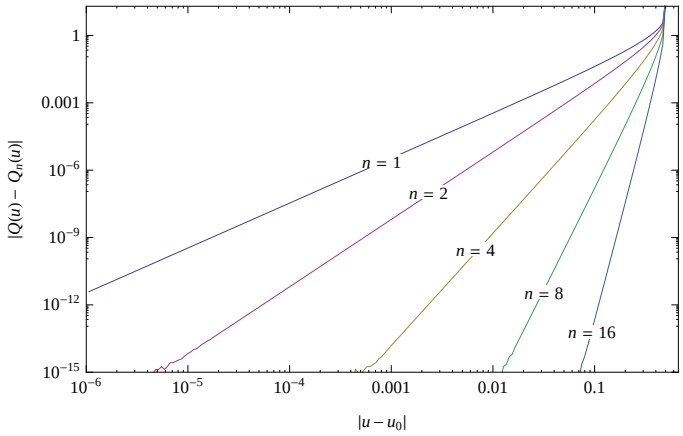


Figure 6.2.3: Error made by truncating (6.2.3) after $n = 1$, 2, 4, 8 and 16 terms.

The number of arithmetic operations required to compute the first n coefficients $q_1, \ldots, q_n$ is $O\left(n^2\right)$. The first few terms of the series are given by

$$Q\left(u\right) = x_0 + \phi N_0\left(u - u_0\right) + \frac{1}{2}\left(\frac{x_0\alpha_1}{\theta} - \beta_1\right)\phi^2 N_0^2\left(u - u_0\right)^2 + O\left(\left(u - u_0\right)^3\right),$$

where $\theta = \sqrt{1 + x_0^2}$ and $\phi = e^{\alpha_1\theta - \beta_1 x_0}$.

To gain some qualitative insight we look at an example, consider the set of distribution parameters $\alpha = 2$, $\beta = 3/2$, $\delta = 1$ and $\mu = 0$. A plot of the corresponding quantile function is given in figure 6.2.1. Denote by $Q_n\left(u\right)$ the n^{th} order Taylor polynomial obtained by truncating (6.2.3) after the n^{th} term. Log-linear and log-log plots of the error in $Q_n\left(u\right)$ for various values of n are given in figures 6.2.2 and 6.2.3 respectively. Of course we do not recommend approximating the quantile function using a Taylor polynomial (more sophisticated approximations will be constructed in chapter 8), our purpose here is to empirically validate the series.

6.2.2 Asymptotic Expansion

To develop the asymptotic behaviour at the singular points $u = 0$ and $u = 1$ note that the hyperbolic distribution function F satisfies the relationship

$$F(x; \alpha_1, \beta_1, 1, 0) = 1 - F(-x; \alpha_1, -\beta_1, 1, 0),$$

which implies

$$Q(u; \alpha_1, \beta_1, 1, 0) = -Q(1 - u; \alpha_1, -\beta_1, 1, 0).$$

Hence without loss of generality we need only look for an asymptotic expansion of Q as $u \to 1$. Now under the assumption $1 \ll Q$ as $u \to 1$, we obtain from (6.2.2),

$$\frac{dQ}{du} \sim N_0 e^{(\alpha_1 - \beta_1)Q}, \quad \text{as } u \to 1.$$

Solving this asymptotic relationship along with the condition $Q(1) = \infty$ yields the leading order behaviour of Q,

$$Q(u) \sim -\frac{\ln(N_0(\alpha_1 - \beta_1)(1 - u))}{(\alpha_1 - \beta_1)}, \quad \text{as } u \to 1.$$

From which we note that Q has a logarithmic singularity at $u = 1$. To develop further terms in the expansion, define $x := Q(u)$, from which it follows $u = F(x)$ and note that Q is implicitly defined by the equation

$$1 - F(x) = \int_x^\infty f(t)\, dt.$$

Rearranging and introducing the variable v we obtain

$$v := (1 - u)N_0 = \int_x^\infty e^{-\alpha_1\sqrt{1+t^2} + \beta_1 t}\, dt.$$

The idea is to expand the integrand appearing in the right hand side and integrate term-wise, this process can be carried out symbolically, giving the first few terms in the expansion,

$$v \sim e^{-x(\alpha_1 - \beta_1)} \left(\frac{1}{\alpha_1 - \beta_1} - \frac{\alpha_1}{2(\alpha_1 - \beta_1)x} + \frac{\alpha_1(4 + \alpha_1(\alpha_1 - \beta_1)\delta^2)}{8(\alpha_1 - \beta_1)^2 x^2} + O\left(\frac{1}{x^3}\right) \right).$$

Taking logs and inverting the resulting series allows us to write down the first few terms in the asymptotic expansion of x,

$$x \sim y + \frac{\alpha_1}{2(\alpha_1 - \beta_1)y} + \frac{\alpha_1}{2(\alpha_1 - \beta_1)^2 y^2} + O\left(\frac{1}{y^3}\right)$$

from which we can conjecture the form the asymptotic expansion of Q as,

$$Q(u) \sim y + \sum_{n=1}^{\infty} \frac{q_n}{y^n}, \quad \text{as } u \to 1, \tag{6.2.4}$$

where,

$$y = -\frac{\ln\left(N_0\left(\alpha_1 - \beta_1\right)(1-u)\right)}{\left(\alpha_1 - \beta_1\right)}. \tag{6.2.5}$$

Substituting (6.2.4) into the first order quantile equation (6.2.2) allows us to derive a recurrence relationship for the coefficients q_n,

$$q_n = \begin{cases} 0, & n = 0 \\ -\frac{\alpha_1}{2(\alpha_1-\beta_1)}, & n = 1 \\ -\frac{1}{(\alpha_1-\beta_1)}\left((n-1)q_{n-1} + \frac{1}{n}\sum_{k=1}^{n-1} k b_k c_{n-k} + \alpha_1 d_n\right), & n \geq 2 \end{cases}, \tag{6.2.6}$$

where

$$a_n = \begin{cases} 0, & n = 0 \\ \frac{1}{2} + q_1, & n = 1 \\ q_n + d_n, & n \geq 2 \end{cases},$$

$$b_n = \alpha_1 a_n - \beta_1 q_n,$$

$$c_n = \begin{cases} 1, & n = 0 \\ \frac{1}{n}\sum_{k=1}^{n} k b_k c_{n-k}, & n \geq 1 \end{cases},$$

and

$$d_n = \frac{1}{n-1}\sum_{k=0}^{n-2}(k+1)(q_{k+1}q_{n-k-2} - a_{k+1}a_{n-k-2}).$$

The number of arithmetic operations required to compute the first n coefficients $q_1, \ldots, q_n$ is $O\left(n^2\right)$. Note that (6.2.4) is a divergent series, however of the summation methods we tested we found Levin's u transform and Padé approximants useful methods for the summation of (6.2.4), see section 8.3 for details. In both cases analytic continuation was observed. Later in chapter 8, we will briefly look at algorithms for constructing rational approximants of Q valid on the domain $[10^{-10}, 1 - 10^{-10}]$, but if necessary one may utilize (6.2.4) to obtain approximations on a wider region.

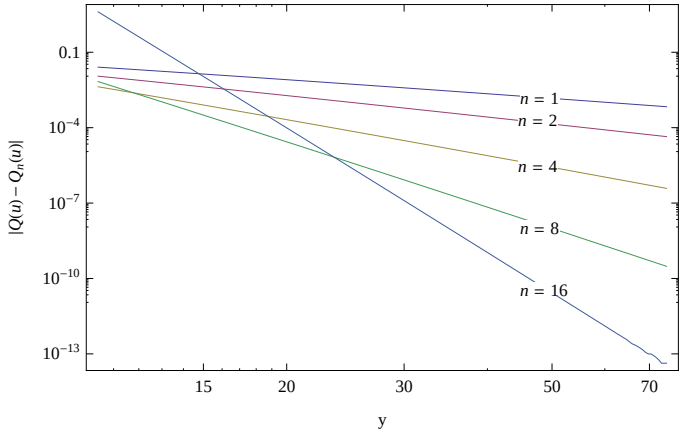


Figure 6.2.4: Error made by truncating (6.2.4) after $n = 1, 2, 4, 8$ and 16 terms.

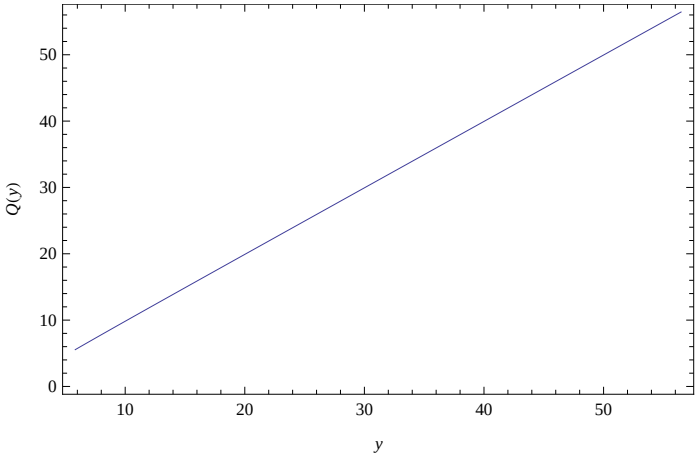


Figure 6.2.5: A plot of $Q_{\text{Hyp}}(\cdot; 2, 3/2, 1, 0)$ as a function of the tail variable y.

Consider again the hyperbolic quantile $Q_{\text{Hyp}}(u; 2, 3/2, 1, 0)$, and denote by $Q_n(u)$ the series (6.2.4) truncated after the n^{th} term. A plot of the error made by $Q_n(u)$ for various values of n is given in figure 6.2.4.

In figure 6.2.5 we plot the quantile function $Q_{\text{Hyp}}(\cdot; 2, 3/2, 1, 0)$ in the right tail region as a function of the tail variable y defined by (6.2.5). Notice in this region the quantile function Q_{Hyp} as a function of y appears to be linear. In the next section we will make a change of variable based on the left and right tail behaviour of Q_{Hyp}, allowing us to build approximations of Q_{Hyp} in the tail regions.

6.2.3 Change of Variable

We now look for a change of variable so that we may better approximate the quantile function in the tail regions. Motivated by the asymptotic behaviour of Q near its singular points, in particular its leading order behaviour we introduce the base distribution defined by the density

$$f_B(x) := \begin{cases} p_- (\alpha_1 + \beta_1) e^{(\alpha_1+\beta_1)x}, & x \le x_m \\ p_+ (\alpha_1 - \beta_1) e^{-(\alpha_1-\beta_1)x}, & x > x_m \end{cases}.$$

Here $x_m := \beta_1/\gamma_1$ is the mode of the hyperbolic distribution, $p_- := e^{-(\alpha_1-\beta_1)x_m} p_m$, $p_+ := e^{(\alpha_1-\beta_1)x_m}(1-p_m)$ and $p_m := F_T(x_m)$. The associated distribution and quantile functions can be written down as

$$F_B(x) = \begin{cases} p_- e^{(\alpha_1+\beta_1)x}, & x \le x_m \\ 1 - p_+ e^{-(\alpha_1-\beta_1)x}, & x > x_m \end{cases}, \tag{6.2.7}$$

and

$$Q_B(u) = \begin{cases} \frac{1}{\alpha_1+\beta_1} \ln\left(\frac{u}{p_-}\right), & u \le p_m \\ -\frac{1}{\alpha_1-\beta_1} \ln\left(\frac{1-u}{p_+}\right), & u > p_m \end{cases}, \tag{6.2.8}$$

respectively. Substituting this choice of f_B into the recycling equation (6.0.8) results in a left and right problem

$$\frac{dA}{dz} = p_-(\alpha_1+\beta_1) N_0 e^{\alpha_1\sqrt{1+A^2} - \beta_1 A + (\alpha_1+\beta_1)z}, \quad z \le x_m$$

and

$$\frac{dA}{dz} = p_+(\alpha_1-\beta_1) N_0 e^{\alpha_1\sqrt{1+A^2} - \beta_1 A - (\alpha_1-\beta_1)z}, \quad z > x_m$$

respectively, along with the suitably imposed initial conditions. For the left problem, we choose $u_0 \in (0, p_m]$ and impose the initial condition $x_0 = A(z_0) = Q_T(u_0)$, where $z_0 := Q_B(u_0)$. Similarly for the right problem choose $u_0 \in [p_-, 1)$. Sometimes we will use the notation $A_L(z)$ and $A_R(z)$ to denote the solutions to the left and right problems respectively, that is

$$A(z) = \begin{cases} A_L(z) & z \le x_m \\ A_R(z) & z > x_m \end{cases}.$$

Again through the method of undetermined coefficients we obtain the Taylor series expansion of A,

$$A(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n, \tag{6.2.9}$$

where the coefficients are defined recursively by

$$a_n = \begin{cases} x_0, & n = 0 \\ \frac{N_0}{n}\theta\left(\alpha_1 + \beta_1\right) d_{n-1}, & n \geq 1 \end{cases},$$

$$b_n = \begin{cases} \sqrt{1+x_0^2}, & n = 0 \\ \frac{1}{nb_0}\left(na_n a_0 + \sum_{k=0}^{n-2}(k+1)\left(a_{k+1}a_{n-k-1} - b_{k+1}b_{n-k-1}\right)\right), & n \geq 1 \end{cases},$$

$$c_n = \begin{cases} \alpha_1\left(b_1 + \phi\right) + \beta_1\left(1 - a_1\right), & n = 1 \\ \alpha_1 b_n - \beta_1 a_n, & n \neq 1 \end{cases},$$

and

$$d_n = \begin{cases} e^{\alpha_1\sqrt{1+x_0^2} - \beta_1 x_0 + \rho z_0}, & n = 0 \\ \frac{1}{n}\sum_{k=1}^{n} k c_k d_{n-k}, & n \geq 1 \end{cases}.$$

For the solution to the left and right problem make the replacements given in table 6.1. The number of arithmetic operations required to compute the first n coefficients $a_1, \ldots, a_n$ is $O\left(n^2\right)$.

Table 6.1: Hyperbolic Coefficients

	Left	Right
θ	p_-	p_+
ϕ	$+1$	-1
ρ	$(\alpha_1 + \beta_1)$	$-(\alpha_1 - \beta_1)$

Consider again the hyperbolic quantile $Q_{\text{Hyp}}\left(u; 2, 3/2, 1, 0\right)$, and denote by $A_n\left(z\right)$ the n^{th} order Taylor polynomial obtained by truncating (6.2.9) after the n^{th} term. A plot of the error made by $A_n\left(z\right)$ for $n = 1, 2, 4, 8, 16$ is given in figure 6.2.6. We end this section by contrasting the error made in the approximations $A_n\left(z\right)$ and $Q_n\left(u\right)$, the n^{th} order Taylor polynomial obtained by truncating (6.2.3). Consider the following experiment, we expand $Q_{\text{Hyp}}\left(u; 2, 3/2, 1, 0\right)$ about $u_0 = 0.95$ and $A\left(z\right)$ about the point $z_0 = Q_B\left(u_0\right)$. We are interested in approximating $Q_{\text{Hyp}}\left(u; 2, 3/2, 1, 0\right)$ at the point $u = 0.99$. The errors made in the approximations $A_n\left(Q_B\left(u\right)\right)$ and $Q_n\left(u\right)$ are listed in table (6.2) for various of the values of the order n.

6.3 Variance Gamma Distribution

Our second major contribution is to apply the above analysis to the variance gamma distribution, which as we will see supplies its own hurdles. The variance gamma dis-

n	$\lvert Q(u) - Q_n(u)\rvert$	$\lvert Q(u) - A_n(Q_B(u))\rvert$
10	0.061609	0.000023
20	0.003810	2.33×10^{-8}
30	0.000289	3.13×10^{-11}
40	0.000024	4.71×10^{-14}
50	2.1×10^{-6}	7.57×10^{-17}

Table 6.2: Comparison of the errors made in the approximations $Q_n(u)$ and $A_n(Q_B(u))$.

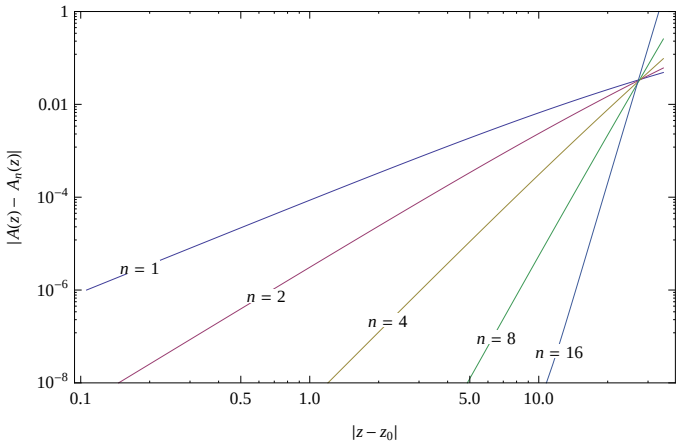


Figure 6.2.6: Error made by truncating (6.2.9) after $n = 1$, 2, 4, 8 and 16 terms.

tribution (see example 39 on page 143) was introduced in the finance literature by Madan and Seneta (1990). To our knowledge there has been very little written on the approximation of the variance gamma quantile.

6.3.1 Taylor Series

The density of the variance gamma distribution is given by,

$$f_{\text{VG}}(x;\lambda,\alpha,\beta,\mu) = \frac{\gamma^{2\lambda}}{(2\alpha)^{\lambda-1/2}\sqrt{\pi}\Gamma(\lambda)}\,|x-\mu|^{\lambda-\frac{1}{2}}\,\mathrm{K}_{\lambda-\frac{1}{2}}(\alpha\,|x-\mu|)\,e^{\beta(x-\mu)}, \tag{6.3.1}$$

where $\lambda > 0$, $\alpha > 0$, $|\beta| < \alpha$, $\gamma = \sqrt{\alpha - \beta}$ and $\mu \in \mathbb{R}$. Setting the location parameter μ to zero and substituting the density (6.3.1) into the quantile equation (6.0.2) gives

$$\frac{dQ}{du} = G(Q(u)), \tag{6.3.2}$$

where the function G is defined as

$$G(y) := N_0 \frac{e^{-\beta y}|y|^{\frac{1}{2}-\lambda}}{\mathrm{K}_{\lambda-\frac{1}{2}}(\alpha|y|)}$$

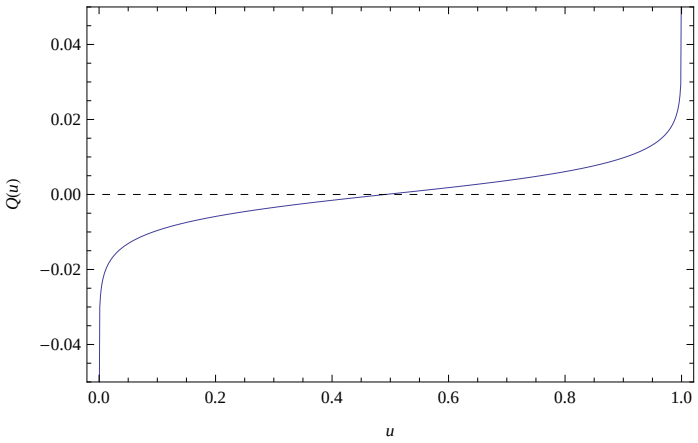


Figure 6.3.1: Graph of $Q_{\mathrm{VG}}(u;\lambda,\alpha,\beta,\mu)$ where $\lambda = 2.2624$, $\alpha = 264.9466$, $\beta = -2.3421$ and $\mu = 0.0002585$.

and $N_0 = (2\alpha)^{\lambda-1/2}\sqrt{\pi}\,\Gamma(\lambda)/\gamma^{2\lambda}$. Our strategy to solve (6.3.2) will be to apply the method of successive differentiation to obtain the Taylor series representation of Q,

$$Q(u) = \sum_{n=0}^{\infty} \frac{q_n}{n!}(u-u_0)^n, \tag{6.3.3}$$

where q_0 is determined by the imposed initial condition[1], and the remaining coefficients are given by

$$q_n = \left.\frac{d^{n-1}}{du^{n-1}}\right|_{u=u_0} G(Q(u)), \quad n \geq 1.$$

Thus the problem reduces to finding the higher order derivatives of the composition $G \circ Q$, which may be obtained recursively through Faà di Bruno's formula. Note first that G can be written as the product of three functions $A(y) := e^{-\beta y}$, $B(y) := |y|^{\frac{1}{2}-\lambda}$, and $C(y) := \left[K_{\lambda-\frac{1}{2}}(\alpha|y|)\right]^{-1}$ (the function A as defined here should not be confused with the solution of the recycling equation (6.0.8), which we shall denote by A_L or A_R in this section). Thus an application of the general Leibniz rule yields

$$G^{(n)}(y) = N_0 \sum_{k=0}^{n}\sum_{j=0}^{k} \binom{n}{k}\binom{k}{j} A^{(n-k)}(y)\, B^{(j)}(y)\, C^{(k-j)}(y). \tag{6.3.4}$$

The higher order derivatives of the functions A and B appearing in (6.3.4) are given by

$$A^{(n)}(y) = (-\beta)^n e^{-\beta y}$$

[1] In the variance gamma case we must be cautious not to impose the initial condition at $u_0 \in (0,1)$ such that $x_0 = Q(u_0) = 0$, i.e. at the zero quantile location since the function $G(y)$ is not analytic at $y = 0$.

and

$$B^{(n)}(y) = y^{\frac{1}{2}-n-\lambda} \prod_{i=0}^{n-1} \left(\frac{1}{2} - \lambda - i\right) \operatorname{sgn}(y)$$

respectively. To find the n^{th} derivative of C note that C can be written as the composition $C(y) = C_1 \circ C_2(y)$, where $C_1(y) := 1/y$ and $C_2(y) := K_{\lambda-\frac{1}{2}}(\alpha|y|)$. Hence we may obtain $C^{(n)}(y)$ through an application of Faà di Bruno's formula,

$$C^{(n)}(y) = \sum \frac{n!}{m_1! m_2! \cdots m_n!} C_1^{(m_1+\cdots+m_n)}(C_2(y)) \prod_{j=0}^{n} \left(\frac{C_2^{(j)}(y)}{j!}\right)^{m_j},$$

where the summation is taken over all solutions $(m_1, m_2, \ldots, m_n) \in \mathbb{Z}_{\geq 0}^n$ to the Diophantine equation[2]

$$m_1 + 2m_2 + 3m_3 \cdots + nm_n = n. \tag{6.3.5}$$

The formulae for the higher order derivatives of C_1 and C_2 are given by

$$C_1^{(n)}(y) = (-1)^n \, n! \, y^{-(n+1)},$$

and

$$C_2^{(n)}(y) = \left(-\frac{\alpha}{2}\right)^n \sum_{k=0}^{n} \binom{n}{k} \times \begin{cases} \mathrm{K}_{\lambda-\frac{1}{2}-(2k-n)}(\alpha y), & y > 0 \\ (-1)^n \, \mathrm{K}_{\lambda-\frac{1}{2}-(2k-n)}(-\alpha y) & y < 0 \end{cases},$$

respectively. Here we have used the identity (Olver et al., 2010, eq. 10.29.5)

$$\mathrm{K}_v^{(n)}(z) = \left(-\frac{1}{2}\right)^n \sum_{k=0}^{n} \binom{n}{k} \mathrm{K}_{v-(2k-n)}(z), \tag{6.3.6}$$

which can easily be proved by induction. Formula (6.3.6) shows that to compute the higher order derivatives of the modified Bessel function of the second kind $\mathrm{K}_v^{(n)}(z)$ with respect to z, we need only a routine to compute $\mathrm{K}_v(z)$. Now given the scheme (6.3.4) to compute $G^{(n)}(y)$ and the first coefficient q_0 defined by the initial condition, we may compute the remaining coefficients q_n for $n \geq 1$ recursively, by another application of Faà di Bruno's formula.

To gain some qualitative insight we look at an example, consider the set of distribution parameters $\lambda = 2.2624$, $\alpha = 264.9466$, $\beta = -2.3421$ and $\mu = 0.0002585$. A

[2]Note that the solutions to (6.3.5) correspond to the integer partitions of n.

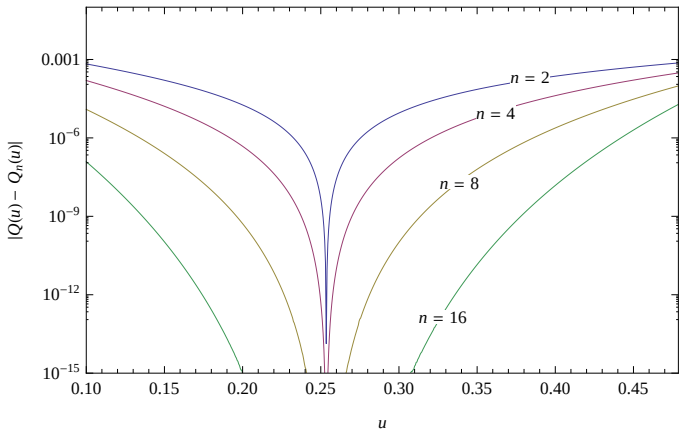


Figure 6.3.2: Error made by truncating (6.3.3) after $n = 2$, 4, 8 and 16 terms.

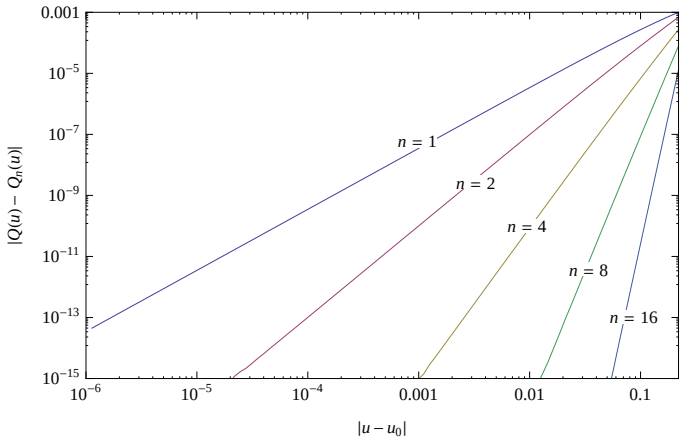


Figure 6.3.3: Error made by truncating (6.3.3) after $n = 1$, 2, 4, 8 and 16 terms.

plot of the corresponding quantile function $Q_{\text{VG}}(u; \lambda, \alpha, \beta, \mu)$ is given in figure (6.3.1). Log-linear and log-log plots of the error made by truncating (6.3.3) are given in figures 6.3.2 and 6.3.3 respectively.

6.3.2 Asymptotic Expansion

Next we will focus on deriving an asymptotic expansion for $Q(u)$. Similar to the hyperbolic quantile the following equality holds,

$$Q(u; \lambda, \alpha, \beta, 0) = -Q(1 - u; \lambda, \alpha, -\beta, 0),$$

so again without loss of generality we need only seek an asymptotic expansion of Q as $u \to 1$. Our strategy here will be to,

1. derive an asymptotic expansion for the density f as $x \to \infty$,

2. integrate term-wise to obtain an asymptotic expansion for the distribution function F as $x \to \infty$,

3. and finally invert this expansion to obtain an asymptotic expansion for the quantile function Q as $u \to 1$.

For the first step we will make use of the asymptotic relationship (Olver et al., 2010, §10.40)

$$\mathrm{K}_v(z) \sim \left(\frac{\pi}{2z}\right)^{\frac{1}{2}} e^{-z} \sum_{k=0}^{\infty} \frac{a_k(v)}{z^k} \quad \text{as } z \to \infty,$$

where

$$a_k(v) := \frac{1}{k!8^k} \prod_{j=1}^{k} \left(4v^2 - (2j-1)^2\right).$$

From which it follows

$$f(x) \sim \frac{\gamma^{2\lambda}}{(2\alpha)^{\lambda}\Gamma(\lambda)} e^{-(\alpha-\beta)x} x^{\lambda-1} \sum_{k=0}^{\infty} \frac{a_k\left(\lambda - \frac{1}{2}\right)}{\alpha^k} x^{-k} \quad \text{as } x \to \infty.$$

Working under the assumption that term-wise integration is a legal operation we obtain an asymptotic expansion for the distribution function,

$$\begin{aligned} 1 - F(x) &\sim \frac{\gamma^{2\lambda}}{2^{\lambda}\Gamma(\lambda)} \sum_{k=0}^{\infty} \alpha^{-\lambda-k} a_k\left(\lambda - 2^{-1}\right) \int_x^{\infty} e^{-(\alpha-\beta)x} t^{\lambda-k-1} dt \\ &= \frac{\gamma^{2\lambda}}{2^{\lambda}\Gamma(\lambda)} \sum_{k=0}^{\infty} \alpha^{-\lambda-k} a_k\left(\lambda - 2^{-1}\right) \Gamma\left(\lambda - k, x(\alpha-\beta)\right), \end{aligned} \tag{6.3.7}$$

where $\Gamma(a, z)$ is the upper incomplete gamma function, which for large z satisfies the asymptotic relationship (Olver et al., 2010, §8.11)

$$\Gamma(a, z) \sim z^{a-1} e^{-z} \sum_{j=0}^{\infty} \frac{\Gamma(a)}{\Gamma(a-j)} z^{-j}. \tag{6.3.8}$$

Substituting (6.3.8) into (6.3.7) and assuming the terms of the series may be rearranged we obtain

$$1 - F(x) \sim \frac{(2\alpha)^{-\lambda}\gamma^{2\lambda}}{\Gamma(\lambda)} x^{\lambda-1} e^{-x(\alpha-\beta)} \sum_{k=0}^{\infty} b_k x^{-k} \quad \text{as } x \to \infty, \tag{6.3.9}$$

where

$$b_k = \sum_{j=0}^{k} (\alpha - \beta)^{-(j+1)} \alpha^{-(k-j)} \left(\prod_{i=0}^{k-1} (\lambda - k + i) \right) a_{k-j} \left(\lambda - \frac{1}{2} \right).$$

The expression (6.3.9) describes the asymptotic behaviour of the variance gamma c.d.f. as $x \to \infty$. Let $u := F(x)$ in (6.3.9), our goal then is to invert this relationship to obtain an asymptotic expansion of the quantile function $Q(u)$ as $u \to 1$. Introducing the variable

$$v := \frac{(2\alpha)^{\lambda} \sqrt{\pi} \Gamma(\lambda)}{\gamma^{2\lambda}} (1 - u),$$

and rearranging (6.3.9) we obtain

$$v \sim x^{\lambda - 1} e^{-x(\alpha - \beta)} D\left(\frac{1}{x} \right) \quad \text{as } x \to \infty, \tag{6.3.10}$$

where D is the formal power series defined by $D(z) := \sum_{k=0}^{\infty} b_k z^k$. Taking logs and introducing the variable

$$y := -\ln v / (\alpha - \beta) \tag{6.3.11}$$

we may write (6.3.10) as

$$x \sim y + \frac{\lambda - 1}{\alpha - \beta} \log x + \frac{1}{\alpha - \beta} \log D\left(\frac{1}{x} \right). \tag{6.3.12}$$

We wish to write x in terms y, this task may at first appear difficult to achieve but as it happens we are in luck. Similar expressions occur frequently in analytic number theory and some useful methods have been developed to invert these kinds of relationships. A drawback of these methods is that they require symbolic computation. The most basic method which one can apply to invert (6.3.12) is the method of asymptotic iteration (Bruijn, 1981). However in our implementation we found that Mathematica Wolfram-Research (2010) struggles when requested to generate even a moderate number of terms (six or more) with this method. Fortunately Salvy (1994) studied these types of asymptotic expansions and observed similar behaviour and provides us with a much more efficient approach. There it was noted that the form of the asymptotic inverse is given by

$$x = Q(u) \sim y + \sum_{n=0}^{\infty} \frac{P_n(\ln y)}{y^n}, \tag{6.3.13}$$

where $P_0(\xi)$ is a polynomial of degree 1 and $P_n(\xi)$ are polynomials of degree n for $n \geq 1$. Following a similar analysis of that in Salvy (1994), it can by shown that the polynomial P_n may be determined up to some unknown constant terms $c_0, \ldots, c_n$ by the recurrence relationship

$$P_n(\xi) = \begin{cases} \frac{(\lambda-1)}{(\alpha-\beta)}\xi + c_0 & n = 0 \\ \frac{(\lambda-1)}{(\alpha-\beta)}(P_{n-1}(\xi) - P_{n-1}(0)) - \frac{(\lambda-1)}{(\alpha-\beta)}(n-1)\int_0^\xi P_{n-1}(t)\ dt + c_n & n \geq 0 \end{cases}. \tag{6.3.14}$$

The unknown constant terms $c_0, \ldots, c_n$ may be computed through the following iteration scheme,

- starting with $u_0(t) = \log(b_0)$, compute

$$u_k(t) = \frac{(\lambda-1)}{(\alpha-\beta)}\log(1 + tu_{k-1}(t)) + \frac{1}{\alpha-\beta}\log D\left(\frac{t}{1+tu_{k-1}(t)}\right), \quad k = 1, \ldots, n+1$$

- extract the constants

$$c_k = \left[t^k\right] u_{n+1}(t), \quad k = 0, \ldots, n$$

where $\left[t^k\right] u_{n+1}(t)$ is used to denote the coefficient of the t^k term in $u_{n+1}(t)$.

An implementation of this scheme in Mathematica (Wolfram-Research, 2010) code has been included in an appendix, see code listing C.0.3. Through this process the first few terms of the asymptotic expansion of Q may be generated as follows,

$$\begin{aligned} Q(u) \quad \sim \quad & y + \frac{(\lambda-1)\log y - \log(\alpha-\beta)}{\alpha-\beta} \\ & + \frac{(\lambda-1)(2\alpha + \alpha\lambda - \beta\lambda + 2\alpha(\lambda-1)\log y - 2\alpha\log(\alpha-\beta))}{2y\alpha(\alpha-\beta)^2} + \cdots \end{aligned}$$

as $u \to 1$. Consider again the variance gamma quantile $Q_{\text{VG}}(u; \lambda, \alpha, \beta, \mu)$ with parameter values $\lambda = 2.2624$, $\alpha = 264.9466$, $\beta = -2.3421$ and $\mu = 0.0002585$. In this case a plot of the error made by truncating (6.3.13) is given in figure 6.3.4.

In figure 6.3.5 we plot the quantile function $Q_{\text{VG}}(\cdot; \lambda, \alpha, \beta, \mu)$ in the right tail region as a function of the tail variable y defined by (6.3.11). Notice in this region the quantile function Q_{VG} as a function of y appears to be linear. In the next section we will make a change of variable based on the left and right tail behaviour of Q_{VG}, allowing us to build approximations of Q_{VG} in the tail regions.

6.3.3 Change of Variable

We now look for a change of variable so that we may better approximate the quantile function in the tail regions. As in the hyperbolic case motivated by the asymptotic

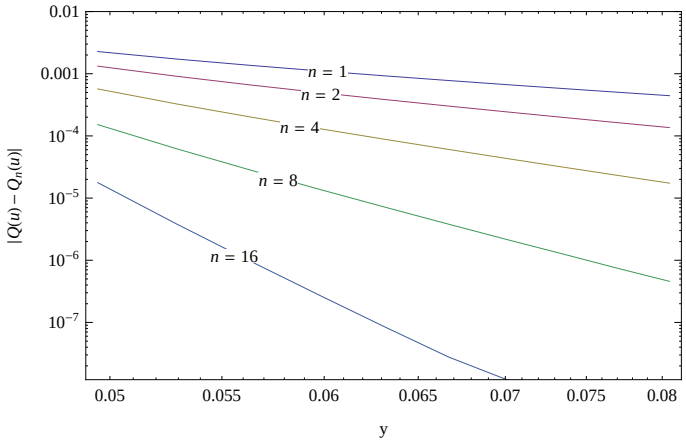


Figure 6.3.4: Error made by truncating (6.3.13) after $n = 1, 2, 4, 8$ and 16 terms.

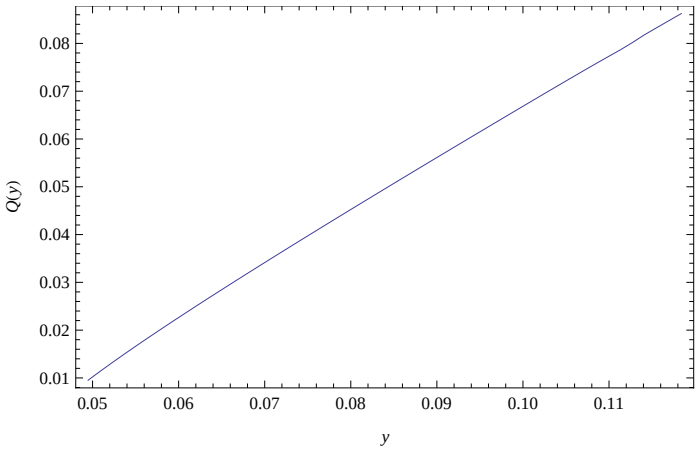


Figure 6.3.5: A plot of $Q_{\mathrm{VG}}(u; \lambda, \alpha, \beta, \mu)$ as a function of tail variable y.

behaviour of the quantile function near its singularities, in particular by its leading order behaviour we introduce the base distribution defined by the density

$$f_B(x) := \begin{cases} p_-(\alpha+\beta)\, e^{(\alpha+\beta)x} & x \le 0 \\ p_+(\alpha-\beta)\, e^{-(\alpha-\beta)x} & x > 0 \end{cases},$$

where $p_- := F(0)$ and $p_+ := 1 - p_-$. The associated distribution and quantile functions can be written down as

$$F_B(x) = \begin{cases} p_- e^{(\alpha+\beta)x}, & x \le 0 \\ 1 - p_+ e^{-(\alpha-\beta)x}, & x > 0 \end{cases}$$

and

$$Q_B(u) = \begin{cases} \frac{1}{\alpha_1+\beta_1} \ln\left(\frac{u}{p_-}\right), & u \le p_- \\ -\frac{1}{\alpha_1-\beta_1} \ln\left(\frac{1-u}{p_+}\right), & u > p_- \end{cases}$$

respectively. Substituting this choice of f_B into the recycling equation (6.0.8) results in a left and right problem given by

$$\frac{dA_L}{dz} = G_L(z), \quad z \le Q_B(p_-) \tag{6.3.15}$$

and

$$\frac{dA_R}{dz} = G_R(z), \quad z > Q_B(p_-) \tag{6.3.16}$$

respectively, along with the suitably imposed initial conditions. For the left problem, we choose $u_0 \in (0, p_-)$ and impose the initial condition $x_0 = A_L(z_0) = Q_T(u_0)$, where $z_0 := Q_B(u_0)$. Similarly for the right problem we choose $u_0 \in (p_-, 1)$. The functions G_L and G_R appearing on right hand side of these differential equations are defined as

$$G_{\mathrm{L}}(z) := p_-(\alpha+\beta) e^{(\alpha+\beta)z} G(A_L(z)),$$

and

$$G_{\mathrm{R}}(z) := p_+(\alpha-\beta) e^{-(\alpha-\beta)z} G(A_R(z)).$$

Suppose that the series solution of either problem is given by

$$\sum_{n=0}^{\infty} \frac{q_n}{n!} (z - z_0)^n. \tag{6.3.17}$$

n	$\left\|Q\left(u\right)-Q_{n}\left(u\right)\right\|$	$\left\|Q\left(u\right)-A_{n}\left(Q_{B}\left(u\right)\right)\right\|$
10	0.00737	1.7×10^{-6}
20	0.00496	1.9×10^{-8}
30	0.00368	3.1×10^{-10}
40	0.00285	5.8×10^{-12}
50	0.00227	1.1×10^{-13}

Table 6.3: Comparison of the errors made in the approximations $Q_{n}\left(u\right)$ and $A_{n}\left(Q_{B}\left(u\right)\right)$.

Here the first coefficient q_{0} is determined by the initial condition imposed at z_{0} and the remaining coefficients are given by

$$q_{n}=G_{\mathrm{L}}^{(n-1)}\left(z_{0}\right),\quad n\geq1,$$

for the left problem and

$$q_{n}=G_{\mathrm{R}}^{(n-1)}\left(z_{0}\right),\quad n\geq1,$$

for the right problem. Both sets of coefficients may easily be computed from an application of Liebniz's rule,

$$G_{\mathrm{L}}^{(n)}\left(z\right)=p_{-}e^{(\alpha+\beta)z}\sum_{k=0}^{n}\binom{n}{k}(\alpha+\beta)^{k+1}\left(G\circ A_{L}\right)^{(n-k)}\left(z\right),$$

$$G_{\mathrm{R}}^{(n)}\left(z\right)=p_{+}e^{-(\alpha-\beta)z}\sum_{k=0}^{n}\binom{n}{k}\left(-1\right)^{k}(\alpha-\beta)^{k+1}\left(G\circ A_{R}\right)^{(n-k)}\left(z\right).$$

Starting with q_{0} the higher order derivatives of the compositions $G\circ A_{L}$ and $G\circ A_{R}$ appearing in these formulae are computed recursively, in precisely the same way we computed the higher order derivatives of $G\circ Q$ above.

Consider again the variance gamma quantile $Q_{\mathrm{VG}}\left(u;\lambda,\alpha,\beta,\mu\right)$ with parameter values $\lambda=2.2624$, $\alpha=264.9466$, $\beta=-2.3421$ and $\mu=0.0002585$ and denote by $A_{n}\left(z\right)$ the n^{th} order polynomial obtained by truncating the series solution of the recycling equation (6.3.17). A plot of the error made by $A_{n}\left(z\right)$, for $n=1,2,4,8,16$ is given in figure 6.3.6. We end this subsection by contrasting the error made in the approximations $A_{n}\left(z\right)$ and $Q_{n}\left(u\right)$, the n^{th} order polynomial obtained by truncating (6.3.3). Consider the following experiment, we expand $Q_{\mathrm{VG}}\left(u;\lambda,\alpha,\beta,\mu\right)$ about $u_{0}=0.001$ and $A\left(z\right)$ about the point $z_{0}=Q_{B}\left(u_{0}\right)$. We are interested in approximating $Q_{\mathrm{VG}}\left(u;\lambda,\alpha,\beta,\mu\right)$ at the point $u=1\times10^{-6}$. The errors made in the approximations $A_{n}\left(Q_{B}\left(u\right)\right)$ and $Q_{n}\left(u\right)$ are listed in table (6.3) for various of the values of the order n.

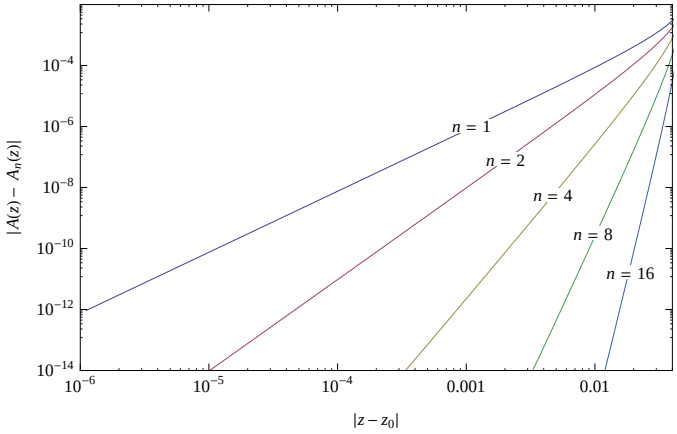


Figure 6.3.6: Error made by truncating (6.3.17) after $n = 1, 2, 4, 8$ and 16 terms.

6.3.4 A more efficient method

In the method described above two applications of Faà di Bruno's formula were required. This formula is useful for finding an explicit expression of the higher order derivatives of composite functions such as $G \circ Q$ and $C_1 \circ C_2$. However the most explicit method is not always the most desirable computationally. More specifically note that Faà di Bruno's formula may be stated as

$$(G \circ Q)^{(n)}(u) = \sum_{r=1}^{n} Y_{n,r} G^{(r)}(Q(u)). \tag{6.3.18}$$

The major draw back in using (6.3.18) is the calculation of the terms

$$Y_{n,r} = \sum \frac{n!}{m_1! m_2! \cdots m_n!} \prod_{j=0}^{n} \left(\frac{Q^{(j)}(u_0)}{j!} \right)^{m_j}, \tag{6.3.19}$$

where the sum is taken over all non-negative integer solutions of (6.3.5) or equivalently the integer partitions of n. Note that the number of partitions $p(n)$ of an integer n grows exponentially with n, thus for large values of n the sum (6.3.19) may be computationally expensive due to the large number of summands. For a method to compute $Y_{n,r}$ see the article by Klimko (1973). We prefer to avoid the use of Faà di Bruno's formula by utilizing the techniques described in section 5.2, in particular theorem 24.

Rather than applying Faà di Bruno's formula, an alternative more efficient scheme to compute the derivatives of $G \circ Q$ at $u = u_0$ exists, provided that we have $q_0 = 0$. If $x_0 = Q(u_0) \neq 0$ then this is easily accomplished by introducing a translation in the dependent variable Q by $-x_0$ in (6.3.2). Then $q_n = (G \circ Q)^{(n)}(u_0)/n!$ for $n \geq 1$ may be computed from a straight forward application of theorem 24. Consequently the coefficients q_n may be expressed as

$$q_n = \begin{cases} 0, & n = 0 \\ \frac{1}{n} h_{n-1}, & n \geq 1 \end{cases},$$

where

$$h_n = \begin{cases} g_0, & n = 0 \\ \sum_{k=1}^{n} g_k q_n^{(k)}, & n \geq 1 \end{cases},$$

$$q_n^{(k)} = \begin{cases} 0, & n < k \\ (q_1)^k, & n = k \\ \sum_{j=1}^{n-k} \left(\left(\frac{k+1}{n-k} \right) j - 1 \right) \frac{q_{j+1}}{q_1} q_{n-j}^{(k)}, & n > k \end{cases},$$

$$g_n = N_0 \sum_{k=0}^{n} \sum_{j=0}^{k} d_{n-k} b_{k-j} a_j,$$

$$d_n = \begin{cases} \frac{1}{c_0}, & n = 0 \\ -\frac{1}{c_0} \sum_{k=1}^{n} c_k d_{n-k}, & n \geq 1 \end{cases},$$

$$c_n = \frac{C^{(n)}(0)}{n!},\ b_n = \frac{B^{(n)}(0)}{n!},\ \text{and}\ a_n = \frac{A^{(n)}(0)}{n!}.$$

The solution to (6.3.2) is then given by

$$Q(u) = x_0 + \sum_{n=0}^{\infty} q_n (u - u_0)^n .$$

The number of arithmetic operations required to compute the first n coefficients $q_1, \ldots, q_n$ is $O(n^3)$. The justification behind this as follows (see also the remarks at the end of chapter 5), it takes

- $O(n^2)$ time to compute $d_1, \ldots, d_n$,
- $O(n^3)$ time to compute $g_1, \ldots, g_n$,
- $O(n^3)$ time to compute $q_j^{(1)}, \ldots, q_j^{(n)}$ for all $1 \leq j \leq n$,
- and $O(n^2)$ time to compute $h_1, \ldots, h_n$.

Thus the coefficients $q_1, \ldots, q_n$ may be computed in $O(n^3)$ time.

In a similar way we can derive solutions to the recycling equations (6.3.15) and (6.3.16). In particular we have

$$A_L(z) = x_0 + \sum_{n=0}^{\infty} q_n (u - u_0)^n ,$$

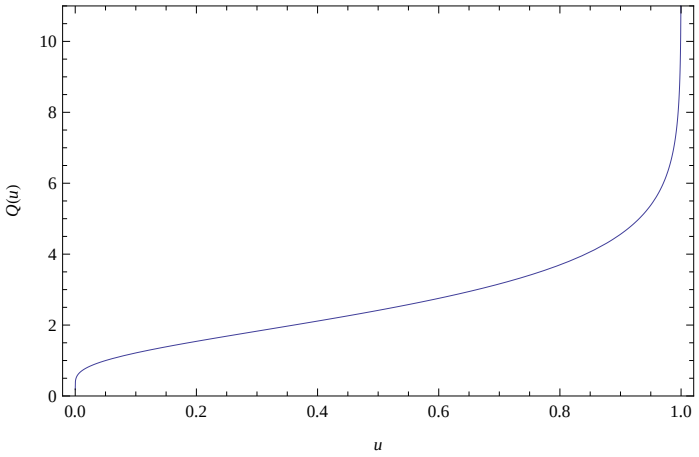


Figure 6.4.1: Graph of $Q_{\text{GIG}}(u;\lambda,\chi,\psi)$ where $\lambda = 3/2$, $\chi = 5$ and $\psi = 2$.

where

$$q_n = \begin{cases} 0 & n = 0 \\ p_-(\alpha+\beta)e^{(\alpha+\beta)z_0}\frac{1}{n}\sum_{k=0}^{n-1}\frac{(\alpha+\beta)^k}{k!}h_{n-k-1} & n \geq 1 \end{cases},$$

and

$$A_R(z) = x_0 + \sum_{n=0}^{\infty} q_n (u-u_0)^n,$$

where

$$q_n = \begin{cases} 0, & n = 0 \\ p_+(\alpha-\beta)e^{-(\alpha-\beta)z_0}\frac{1}{n}\sum_{k=0}^{n-1}(-1)^k\frac{(\alpha-\beta)^k}{k!}h_{n-k-1}, & n \geq 1 \end{cases}$$

and h_n is as before. The number of arithmetic operations required to compute the first n coefficients $a_1,\ldots,a_n$ is $O(n^3)$.

6.4 Generalized Inverse Gaussian Distribution

Our third major contribution is to present similar results for the generalized inverse Gaussian (GIG) distribution. This distribution differs from the hyperbolic and variance gamma in the sense that it does not belong but is related to the generalized hyperbolic family of distributions. Fortunately the work done in the previous two sections will pay dividends, and indeed as we will see there are some remarkable similarities.

6.4.1 Taylor Series

The GIG distribution is used in the construction of generalized hyperbolic distributions; more specifically a normal mean mixture distribution where the mixing distribution is the GIG distribution results in a generalized hyperbolic distribution, see appendix

B for details. Consequently if one can generate GIG random variates then a simple transformation may be applied to generate variates from the generalized hyperbolic distribution (Weron, 2004). The probability density function of a GIG random variable is given by

$$f_{\text{GIG}}\left(x;\lambda,\chi,\psi\right)=\frac{\left(\psi/\chi\right)^{\lambda/2}}{2\mathrm{K}_{\lambda}\left(\sqrt{\psi\chi}\right)}x^{\lambda-1}e^{-\frac{1}{2}\left(\chi x^{-1}+\psi x\right)},\quad x>0, \tag{6.4.1}$$

where $\lambda\in\mathbb{R}$, $\chi>0$, $\psi>0$ and $\mathrm{K}_{v}\left(z\right)$ is the modified Bessel function of the third kind with index v.

We will use an alternative parametrization to the standard one above, let $\eta=\sqrt{\chi/\psi}$ and $\omega=\sqrt{\chi\psi}$, the density then reads

$$f_{\text{GIG}}\left(x;\lambda,\eta,\omega\right)=\frac{1}{2\eta^{\lambda}\mathrm{K}_{\lambda}\left(\omega\right)}x^{\lambda-1}e^{-\frac{\omega}{2}\left(\eta x^{-1}+\frac{1}{\eta}x\right)}. \tag{6.4.2}$$

In this new parametrization ω and λ are scale invariant and η is a scale parameter, so in the following we may set $\eta=1$ without loss of generality. The first order quantile equation (6.0.2) now reads,

$$\frac{dQ}{du}=2\mathrm{K}_{\lambda}\left(\omega\right)e^{\frac{1}{2}\omega\left(\frac{1}{Q}+Q\right)}Q^{1-\lambda}.$$

Let $u_0\in(0,1)$ and impose the initial condition $Q\left(u_0\right)=x_0$. For the case $\lambda\neq1$, the GIG quantile Q admits the Taylor series expansion

$$Q\left(u\right)=\sum_{n=0}^{\infty}q_n(u-u_0)^n, \tag{6.4.3}$$

where the coefficients q_n are defined recursively as follows,

$$q_n=\begin{cases}x_0, & n=0\\ \frac{2}{n}K_{\lambda}(\omega)\sum_{i=0}^{n-1}b_ic_{n-i-1}, & n\geq1\end{cases},$$

where,

$$a_n=\begin{cases}\frac{1}{q_0}, & n=0\\ -\frac{1}{q_0}\sum_{i=1}^{n}q_ia_{n-i}, & n\geq1\end{cases},$$

$$b_n=\begin{cases}e^{\frac{\omega}{2}(a_0+q_0)}, & n=0\\ \frac{\omega}{2n}\sum_{i=1}^{n}i\left(a_i+q_i\right)b_{n-i}, & n\geq1\end{cases},$$

and

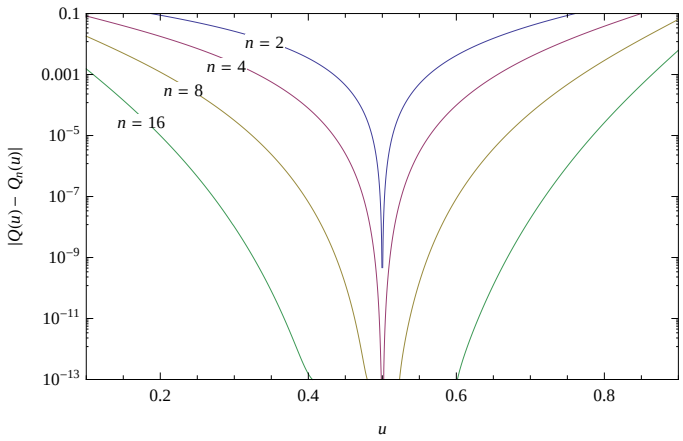


Figure 6.4.2: Error made by truncating (6.4.3) after $n = 2$, 4, 8 and 16 terms.

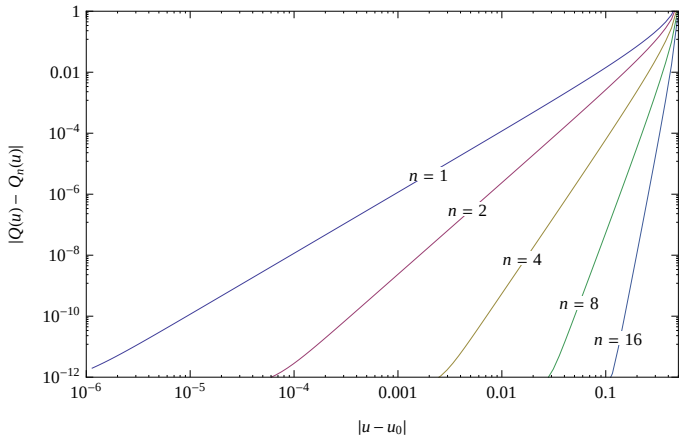


Figure 6.4.3: Error made by truncating (6.4.3) after $n = 1$, 2, 4, 8 and 16 terms.

$$c_n = \begin{cases} q_0^{1-\lambda}, & n = 0 \\ \frac{1}{q_0} \sum_{i=1}^{n} \left(\frac{(2-\lambda)i}{n} - 1 \right) q_i c_{n-i}, & n \geq 1 \end{cases}.$$

The number of arithmetic operations required to compute the first n coefficients $q_1, \ldots, q_n$ is $O(n^2)$. For the special case $\lambda = 1$, the coefficients are somewhat simplified, with a_n and b_n as defined above the coefficients appearing in (6.4.3) become

$$q_n = \begin{cases} x_0, & n = 0 \\ \frac{2}{n} K_1(\omega) b_{n-1}, & n \geq 1 \end{cases}.$$

To gain some qualitative insight we look at an example, consider the set of distribution parameters $\lambda = 3/2$, $\chi = 5$ and $\psi = 2$. A plot of the corresponding quantile function $Q_{\text{GIG}}(u; \lambda, \chi, \psi)$ is given in figure 6.4.1 on page 66. Log-linear and log-log plots of the error in this case made by truncating (6.4.3) are given in figures 6.4.2 and ?? respectively.

6.4.2 Asymptotic Expansion

Next we will focus on developing the asymptotic behaviour of Q as $u \to 1$. We proceed in an analogous fashion to the variance gamma case, and find that remarkably the form of asymptotic expansion of Q_{GIG} is very similar to that of Q_{VG} as $u \to 1$. From the properties of the c.d.f. we have,

$$1 - F(x) = \frac{1}{2K_\lambda(\omega)} \int_x^\infty t^{\lambda-1} e^{-\frac{1}{2}\left(\frac{1}{t}+t\right)\omega} dt.$$

Expanding the $e^{-\omega/2t}$ term and integrating term-wise we obtain,

$$1 - F(x) \sim \frac{1}{2K_\lambda(\omega)} \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \left(\frac{2}{\omega}\right)^{\lambda-2k} \Gamma\left(\lambda - k, \frac{\omega x}{2}\right), \quad \text{as } x \to \infty,$$

where $\Gamma(a, z)$ is the upper incomplete gamma function. As in the variance gamma case substituting (6.3.8) and rearranging the terms provides us with a more convenient form of the asymptotic expansion for the GIG c.d.f.,

$$1 - F(x) \sim \frac{1}{2K_\lambda(\omega)} x^{\lambda-1} e^{-\frac{\omega x}{2}} \sum_{k=0}^{\infty} b_k x^{-k}, \quad \text{as } x \to \infty, \tag{6.4.4}$$

where

$$b_k = \sum_{j=0}^{k} \frac{(-1)^{k-j}}{(k-j)!} \left(\frac{\omega}{2}\right)^{k-2j-1} \left(\prod_{i=0}^{j-1} (\lambda - k + i)\right).$$

Now let $u := F(x)$ and introduce the variable $v := 2K_\lambda(\omega)(1-u)$, then we can rewrite (6.4.4) as,

$$v \sim x^{\lambda-1} e^{-\frac{\omega x}{2}} D(\frac{1}{x}), \tag{6.4.5}$$

where D is the formal power series defined by $D(z) = \sum_{k=0}^{\infty} b_k z^k$. To invert the asymptotic relationship (6.4.5) we start by taking logs and introducing the variable

$$y := -(2/\omega) \ln v. \tag{6.4.6}$$

We may now write (6.4.5) as,

$$x \sim y + \frac{2(\lambda-1)}{\omega} \log x + \frac{2}{\omega} \log D\left(\frac{1}{x}\right). \tag{6.4.7}$$

One may now apply the method of asymptotic iteration (Bruijn, 1981) to invert (6.4.7). However for the same reasons as in the variance gamma case we apply the method of

Salvy (1994). Again the form of the asymptotic inverse is given by,

$$x = Q(u) \sim y + \sum_{n=0}^{\infty} \frac{P_n(\ln y)}{y^n}, \tag{6.4.8}$$

where $P_0(\xi)$ is a polynomial of degree 1 and $P_n(\xi)$ is a polynomial of degree n for $n \geq 1$. Following a similar analysis of that in Salvy (1994) it can by shown that P_n may be determined up to some unknown constant terms $c_0, \ldots, c_n$ by the recurrence relationship,

$$P_n(\xi) = \begin{cases} \frac{2(\lambda-1)}{\omega}\xi + c_0, & n = 0 \\ \frac{2(\lambda-1)}{\omega}(P_{n-1}(\xi) - P_{n-1}(0)) - \frac{2(\lambda-1)}{\omega}(n-1)\int_0^{\xi} P_{n-1}(t)\, dt + c_n, & n \geq 0 \end{cases}. \tag{6.4.9}$$

The unknown constant terms $c_0, \ldots, c_n$ may be computed through the following iteration scheme,

- starting with $u_0(t) = \log(b_0)$, compute

$$u_k(t) = \frac{2(\lambda-1)}{\omega}\log(1 + t u_{k-1}(t)) + \frac{2}{\omega}\log D\left(\frac{t}{1 + t u_{k-1}(t)}\right), \quad k = 1, \ldots, n+1$$

- extract the constants

$$c_k = \left[t^k\right] u_{n+1}(t), \quad k = 0, \ldots, n$$

where $\left[t^k\right] u_{n+1}(t)$ is used to denote the coefficient of the t^k term in $u_{n+1}(t)$.

Through this process the first few terms of the asymptotic expansion of Q may be generated as follows,

$$\begin{aligned} Q(u) \;\sim\; & y + \frac{2\left((-1+\lambda)\ln y + \ln\left(\frac{2}{\omega}\right)\right)}{\omega} \\ & + \frac{4\lambda - \omega^2 - 4 + 4(-1+\lambda)^2 \ln y + 4(-1+\lambda)\ln\left(\frac{2}{\omega}\right)}{\omega^2 y} + \cdots \end{aligned}$$

as $u \to 1$. To observe the asymptotic behaviour of $Q(u)$ as $u \to 0$, we utilize the following identity,

$$Q(u; \lambda, 1, \omega) = \frac{1}{Q(1-u; -\lambda, 1, \omega)}, \tag{6.4.10}$$

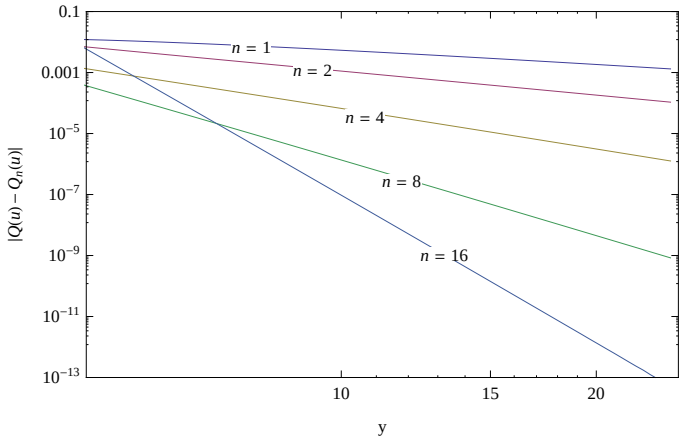


Figure 6.4.4: Error made by truncating (6.4.8) after $n = 2$, 4, 8 and 16 terms.

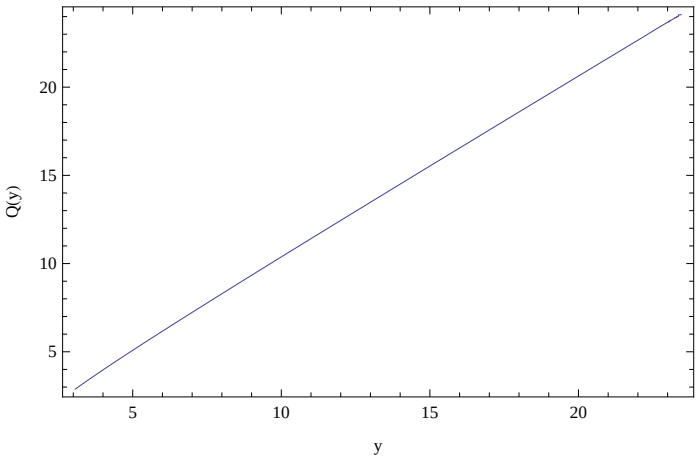


Figure 6.4.5: A plot of $Q_{\text{GIG}}\left(u; 3/2, 5, 2\right)$ as a function of the tail variable y.

which can be easily proven as follows; by definition of the density (6.4.2) we have

$$f\left(x; \lambda, 1, \omega\right) = \frac{1}{x^2} f\left(\frac{1}{x}; -\lambda, 1, \omega\right).$$

Integrating both sides then yields

$$F\left(x; \lambda, 1, \omega\right) = 1 - F\left(\frac{1}{x}; -\lambda, 1, \omega\right),$$

from which (6.4.10) follows. Consider again $Q_{\text{GIG}}\left(u; 5, 2, 3/2\right)$, a plot of the error made in this case by truncating (6.4.8) is given in figure 6.4.4.

In figure 6.4.5 we plot the quantile function $Q_{\text{GIG}}\left(\cdot; 3/2, 5, 2\right)$ in the right tail region as a function of the tail variable y as defined by (6.4.6). Notice in this region the quantile function Q_{GIG} as a function of y appears to be linear. In the next section we will make a change of variable based on the left and right tail behaviour of Q_{GIG}, allowing us to build approximations of Q_{GIG} in the tail regions.

6.4.3 Change of Variable

Next we consider solving the recycling ODE, but first we must choose a base distribution. Again motivated by the asymptotic behaviour of Q as $u \to 0$ and $u \to 1$, in particular the leading order behaviours we suggest the following base distribution characterized by the density function

$$f_B(x) = \begin{cases} p_{\mathrm{L}} \frac{\omega}{2x^2} e^{-\frac{\omega}{2x}}, & x \leq x_m \\ p_{\mathrm{R}} \frac{\omega}{2} e^{-\frac{\omega}{2}x}, & x > x_m \end{cases}. \tag{6.4.11}$$

Here x_m serves as a cut-off point between two suitably weighted density functions, in particular x_m is the mode of the GIG distribution defined by,

$$x_m = \frac{\lambda - 1 + \sqrt{(\lambda-1)^2 + \omega^2}}{\omega}.$$

Note that the density function $f_L(x) := (\omega/2x^2)e^{-\omega/2x}$ belongs to the scaled inverse χ^2 distribution with 2 degrees of freedom and scale parameter $\omega/2$, and that the density function $f_R(x) := (\omega/2)e^{-\omega x/2}$ is the density of an exponential distribution with rate parameter $\omega/2$. The normalizing constants p_{L} and p_{R} are defined by,

$$p_{\mathrm{L}} = e^{\frac{\omega}{2x_m}} p_m,$$

and,

$$p_{\mathrm{R}} = e^{\frac{\omega}{2}x_m} (1 - p_m),$$

where $p_m = F_{\mathrm{GIG}}(x_m)$. The associated base distribution and quantile functions can be written down as,

$$F_B(x) = \begin{cases} p_{\mathrm{L}} e^{-\frac{\omega}{2x}}, & x \leq x_m \\ 1 - p_{\mathrm{R}} e^{-\frac{\omega}{2}x}, & x > x_m \end{cases}$$

and

$$Q_B(u) = \begin{cases} -\frac{\omega}{2\ln(u/p_{\mathrm{L}})}, & u \leq p_m \\ x_m + \frac{2}{\omega} \ln\left(\frac{p_m - 1}{u - 1}\right), & u > p_m \end{cases} \tag{6.4.12}$$

respectively. Substituting this choice of f_B into the recycling equation (6.0.8) then leads to a left and right problem given by

$$\frac{dA}{dz} = p_{\mathrm{L}} \mathrm{K}_\lambda(\omega) \frac{\omega}{z^2} e^{-\frac{\omega}{2z}} e^{\frac{1}{2}\omega\left(\frac{1}{A} + A\right)} A^{1-\lambda}, \quad z \leq x_m \tag{6.4.13}$$

and

$$\frac{dA}{dz} = p_{\mathrm{R}}\omega \mathrm{K}_{\lambda}\left(\omega\right) e^{\frac{1}{2}\omega\left(\frac{1}{A}+A-z\right)} A^{1-\lambda}, \quad z > x_m \tag{6.4.14}$$

respectively, along with the suitably imposed initial conditions. For the left problem, we choose $u_0 \in (0, p_m]$ and impose the initial condition $x_0 = A\left(z_0\right) = Q_T\left(u_0\right)$, where $z_0 := Q_B\left(u_0\right)$. Similarly for the right problem we choose $u_0 \in [p_m, 1)$. Treating the left problem first, we find,

$$A\left(z\right) = \sum_{n=0}^{\infty} a_n (z - z_0)^n, \tag{6.4.15}$$

where the coefficients are computed recursively through the identity,

$$a_n = \begin{cases} x_0 & n = 0 \\ \frac{2p_{\mathrm{L}}}{n}\mathrm{K}_{\lambda}\left(\omega\right)\left(\sum_{k=0}^{n-1}\sum_{j=0}^{n-k-1}(k+1)b_{k+1}d_j e_{n-k-j-1}\right) & n \geq 1 \end{cases},$$

where,

$$b_n = \begin{cases} e^{-\omega/2z_0} & n = 0 \\ \frac{\omega}{2n}\sum_{k=0}^{n-1}(-1)^k \frac{(k+1)}{z_0^{k+2}} b_{n-k-1} & n \geq 1 \end{cases},$$

$$c_n = \begin{cases} \frac{1}{a_0} & n = 0 \\ -\frac{1}{a_0}\sum_{i=1}^{n} a_i c_{n-i} & n \geq 1 \end{cases},$$

$$d_n = \begin{cases} e^{\omega(c_0+a_0)/2} & n = 0 \\ \frac{\omega}{2n}\sum_{i=1}^{n} i(c_i + a_i) d_{n-i} & n \geq 1 \end{cases},$$

and

$$e_n = \begin{cases} a_0^{1-\lambda} & n = 0 \\ \frac{1}{a_0}\sum_{i=1}^{n}\left(\frac{(2-\lambda)i}{n} - 1\right) a_i e_{n-i} & n \geq 1 \end{cases}.$$

The number of arithmetic operations required to compute the first n coefficients $a_1, \ldots, a_n$ is $O\left(n^2\right)$. The coefficients appearing in the series solution (6.4.15) to the right problem 6.4.14 are given by,

$$a_n = \begin{cases} x_0 & n = 0 \\ \frac{\omega p_{\mathrm{R}}}{n}\mathrm{K}_{\lambda}\left(\omega\right)\left(\sum_{k=0}^{n-1} d_k e_{n-k-1}\right) & n \geq 1 \end{cases},$$

where c_n and e_n are defined as in the solution to the left problem above and

n	$\lvert Q(u) - Q_n(u)\rvert$	$\lvert Q(u) - A_n(Q_B(u))\rvert$
10	0.080106	0.00048
20	0.048378	4.2×10^{-6}
30	0.033651	2.8×10^{-9}
40	0.024941	3.6×10^{-11}
50	0.019183	7.5×10^{-12}

Table 6.4: Comparison of the errors made in the approximations $Q_n(u)$ and $A_n(Q_B(u))$.

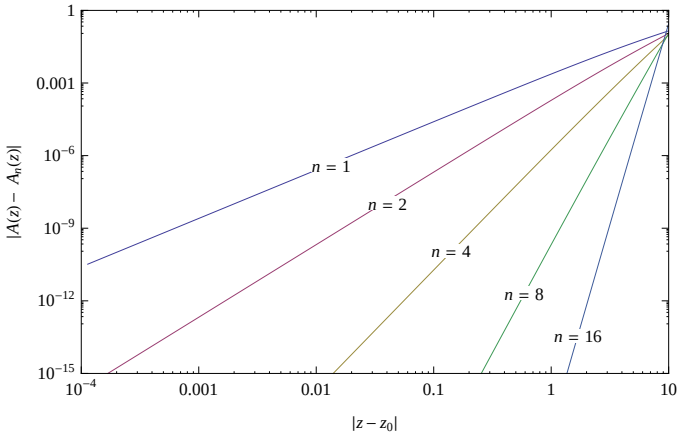


Figure 6.4.6: Error made by truncating (6.4.15) after $n = 1, 2, 4, 8$ and 16 terms.

$$b_n = \begin{cases} \frac{\omega}{2}(c_1 + a_1 - 1) & n = 1 \\ \frac{\omega}{2}(c_n + a_n) & n \neq 1 \end{cases},$$

and

$$d_n = \begin{cases} e^{b_0 - \omega z_0/2} & n = 0 \\ \frac{1}{n}\sum_{i=1}^{n} i b_i d_{n-i} & n \geq 1 \end{cases}.$$

Denote by $A_n(z)$ the n^{th} order polynomial obtained by truncating (6.4.15) and consider again $Q_{\text{GIG}}(u; 3/2, 5, 2)$, a plot of the error made by $A_n(z)$, for $n = 1, 2, 4, 8, 16$ is given in figure 6.4.6. We end this section by contrasting the error made in the approximations $A_n(z)$ and $Q_n(u)$, the n^{th} order polynomial obtained by truncating (6.4.3). Consider the following experiment, we expand $Q_{\text{GIG}}(u; 3/2, 5, 2)$ about $u_0 = 0.1$ and $A(z)$ about the point $z_0 = Q_B(u_0)$. We are interested in approximating $Q_{\text{GIG}}(u; 3/2, 5, 2)$ at the point $u = 1 \times 10^{-3}$. The errors made in the approximations $A_n(Q_B(u))$ and $Q_n(u)$ are listed in table (6.4) for various of the values of the order n.

Chapter 7

Alternative Routes

There is more than one way to gain a handle on the quantile function Q, we refer the reader to figure 7.0.1. Our main contribution thus far has been to develop methods to find representations of Q given the density function f. This route is marked by the diagonal arrow in figure 7.0.1. However the density f need not be our starting point. In the following two subsections we look at a couple of alternative routes that start out by examining the characteristic function ϕ and c.d.f. F respectively. The first of these ideas was introduced by Shaw and McCabe (2009) and the second is an idea which we believe to be new and is applicable when the series representation of F is available.

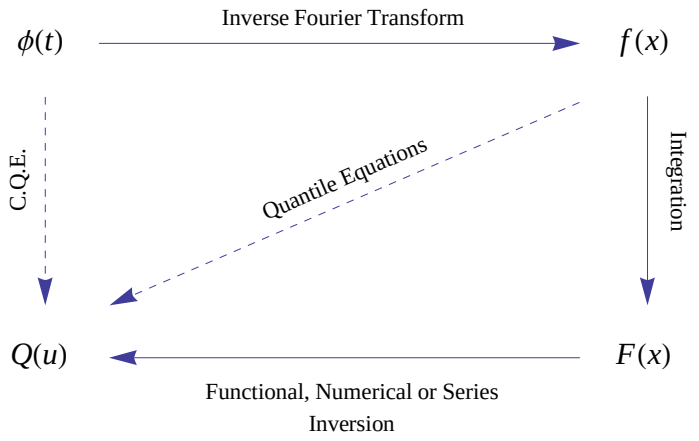


Figure 7.0.1

7.1 Momentum Space

In many situations one does not know the density or the distribution function of a random variable in closed form but the characteristic function is known exactly. Such is the case when dealing with a stable distribution, for which the probability density function does not have a closed form expression. Another case is when modelling with a Lévy process $\{X_t\}_{t\geq 0}$, the density of X_t is rarely known but its characteristic function

is given by the Lévy–Khinchine formula. Motivated by these remarks, the aim of this section is to derive an analytic representation of the quantile function with knowledge only of the characteristic function. The traditional route taken in this scenario (provided that all steps are valid) is as follows, for a particular probability distribution assuming that we have a closed form expression of the characteristic function ϕ_X but not the associated density function f_X,

- apply the inverse Fourier transform to ϕ_X to obtain f_X,
- integrate the density f_X to obtain the distribution function F_X,
- and invert the distribution function F_X to obtain Q_X.

This route has been outlined by the solid arrows in figure 7.0.1. In certain situations the Gil Pilaez inversion formula (see theorem 27) may be applied to the characteristic function to obtain the c.d.f. but an inversion step is still required. In this section we discuss a much more direct approach to the problem introduced by Shaw and McCabe (2009).

7.1.1 Characteristic Quantile Equation (C.Q.E.)

A non-linear second order integro differential equation is derived which links the characteristic function to the associated quantile function. Solving such an equation would allow us to short cut the three step procedure described above.

The characteristic function $\phi_X : \mathbb{R} \mapsto \mathbb{C}$ of a random variable X is defined as the expectation of the random variable e^{itX},

$$\phi_X(t) = E\left[e^{itX}\right] = \int_{-\infty}^{\infty} e^{itx} dF_X(x), \tag{7.1.1}$$

where $i = \sqrt{-1}$ and the integral is of the Riemann-Stieltjes kind. If the density f_X exists then ϕ_X is the Fourier transform of f_X,

$$\phi_X(t) = \mathcal{F}\left\{f_X(x)\right\}(t) = \int_{-\infty}^{\infty} e^{itx} f_X(x)\, dx. \tag{7.1.2}$$

In this case, given ϕ_X the density can be recovered by applying the inverse Fourier transform,

$$f_X(x) = \mathcal{F}^{-1}\left\{\phi_X(t)\right\}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \phi_X(t)\, e^{-itx} dt. \tag{7.1.3}$$

Next we will derive a series of relationships between the characteristic and quantile functions.

- First by making the substitution $u = F_X(x)$ and $x = Q_X(u)$ in (7.1.1) we obtain

$$\phi_X(t) = \int_0^1 e^{itQ_X(u)} du. \tag{7.1.4}$$

- Second by substituting (7.1.3) into the first order quantile equation (6.0.2) we obtain

$$\frac{dQ_X}{du} \int_{-\infty}^{\infty} \phi_X(t) e^{-itQ_X(u)} dt = 2\pi. \tag{7.1.5}$$

- Differentiating (7.1.5) with respect to u yields

$$\frac{d^2Q_X}{du^2} \int_{-\infty}^{\infty} \phi_X(x) e^{-itQ_X(u)} dt - i \left(\frac{dQ_X}{du}\right)^2 \int_{-\infty}^{\infty} t\phi_X(t) e^{-itQ_X(u)} dt = 0. \tag{7.1.6}$$

- Finally combining (7.1.5) and (7.1.6) leads to a fourth identity which we shall refer to as the *characteristic quantile equation* (C.Q.E.)

$$\frac{d^2Q_X}{du^2} = \left(\frac{dQ_X}{du}\right)^3 \frac{1}{2\pi} \int_{-\infty}^{\infty} it\phi_X(t) e^{-itQ_X(u)} dt. \tag{7.1.7}$$

An important property of this equation is that it is location and scale invariant. To see this consider the transformation $Y = a + bX$, for some constants a and $b > 0$. Then the characteristic and quantile functions of Y are given by,

$$\phi_Y(t) = e^{ita}\phi_X(bt)$$

and

$$Q_Y(u) = a + bQ_X(u)$$

respectively. Substituting into (7.1.7) we see that the characteristic quantile equation remains unaltered under location and scale transformations.

7.1.2 Inversion

Given a characteristic function ϕ, many inversion theorems exist which express the associated c.d.f. F_X in terms of ϕ_X. One such expression for F_X is given by the Gil-Pelaez inversion formula (Gil-Pelaez, 1951) and is the subject of this sub section. It has the advantage over other formulations in that constant terms such as $F_X(0)$ have been evaluated. The original proof of the formula given by Gil-Pelaez hinges on an application of Tonelli's theorem followed by the dominated convergence theorem. It is derived in Shaw and McCabe (2009) using tools from residue calculus, we will however base our proof on Shephard (1991), which relies on an application of the Riemann-Lebesgue lemma. In the following $L^1(A)$ denotes the set of Lebesgue integrable functions on the set A.

Theorem 27. *Suppose X is a r.v. with density, distribution, and characteristic functions given by f_X, F_X and ϕ_X respectively. If the mean of X exists, $\mathbb{E}[X] < \infty$, and both the density and characteristic functions are integrable, $f_X, \phi_X \in L^1(\mathbb{R})$, then*

$$F_X(x) = \frac{1}{2} - \frac{1}{2\pi}\int_0^\infty \frac{\phi_X(t)\, e^{-itx} - \phi_X(-t)\, e^{itx}}{it} dt,$$

where $i = \sqrt{-1}$.

Proof. Consider the function,

$$g(y) = \begin{cases} \operatorname{sgn}(y) & , y \in [-h, h] \\ 0 & , y \notin [-h, h] \end{cases}.$$

Its Fourier transform is given by,

$$\phi_h(t) = \int_{-h}^{h} e^{ity} \operatorname{sgn}(y)\, dy = \frac{e^{ith} + e^{-ith} - 2}{it} = \frac{2(\cos ht - 1)}{it}.$$

Define u_h as the convolution of g with f_X,

$$\begin{aligned} u_h(x) &:= f_X * g(x) \\ &= \int_{-h}^{h} f_X(x-y) \operatorname{sgn}(y)\, dy \\ &= -\int_{-h}^{0} f_X(x-y)\, dy + \int_0^h f_X(x-y)\, dy \\ &= 2F_X(x) - F_X(x+h) - F_X(x-h) \end{aligned}$$

Note $\lim_{h\to\infty} u_h(x) = 2F_X(x) - 1$, is clearly a bounded function but not a member of $L^1(\mathbb{R})$. From the convolution theorem u_h has the transform,

$$\widehat{u}_h(t) = \widehat{f_X * g}(t) = 2\phi(t)\frac{(\cos ht - 1)}{it}$$

which is integrable since $(\cos ht - 1)/it$ is continuous and bounded on $\mathbb{R}$ and $\phi_X \in L^1(\mathbb{R})$ by the hypothesis. Thus we may apply the Fourier inversion theorem to obtain the identity,

$$\begin{aligned} u_h(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\widehat{u}_h(t)\, dt \\ &= \frac{2}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\phi_X(t)\frac{(\cos ht - 1)}{it} dt \\ &= \frac{2}{2\pi}\int_0^\infty (\cos ht - 1)\,\eta(t)\, dt \end{aligned}$$

where $\eta(t) = (\phi_X(t)\, e^{-itx} - \phi_X(-t)\, e^{itx})/it$. We show next that $\eta(t)$ is an integrable function on $\mathbb{R}$. Since the mean of X exists, we can conclude ϕ_X is differentiable at 0, since $E[X] = -i\phi'_X(0)$ and hence $\eta(t)$ is bounded on a neighbourhood of 0. To see this define $r(t) := \phi_X(t)\, e^{itx}$, which is differentiable at 0, and note that $\eta(t)$ can be written as

$$\eta(t) = \frac{r(t)}{t} + \frac{r(-t)}{t} = \frac{r(t) - r(0)}{t} + \frac{r(0) - r(-t)}{t}.$$

From which we obtain $\lim_{t\to 0} \eta(t) = 2r'(0)$. Hence for a given $\epsilon > 0$ there exists a $\delta > 0$ such that if $|t| \leq \delta$ then $|\eta(t) - 2r'(0)| < \epsilon$. This implies $\eta(t)$ is bounded and hence integrable on the interval $[-\delta, \delta]$. It is an easy exercise to show $\eta(t)$ is integrable away from the origin, that is $\int_\delta^\infty \eta(t)\, dt < \infty$ and $\int_{-\infty}^{-\delta} \eta(t)\, dt < \infty$. Since $\eta(t) \in L^1(\mathbb{R})$ we may apply the Riemann-Lebesgue lemma to conclude,

$$\begin{aligned}\lim_{h\to\infty} u_h(x) &= \frac{2}{2\pi}\lim_{h\to\infty}\int_0^\infty \cos(ht)\,\eta(t)\, dt - \frac{2}{2\pi}\lim_{h\to\infty}\int_0^\infty \eta(t)\, dt \\ &= -\frac{2}{2\pi}\int_0^\infty \eta(t)\, dt = 2F_X(x) - 1.\end{aligned}$$

and the result follows. □

Later we will find Taylor series expansions of the quantile function centred at the point $u_0 := F_X(0)$. A corollary to this theorem will allow us to locate the point u_0 in the unit interval.

Corollary 28 (The zero quantile location)**.** *Let Q_X be the quantile function of X, if the value u_0 satisfies $Q_X(u_0) = 0$, then*

$$u_0 = \frac{1}{2} - \frac{1}{2\pi}\int_0^\infty \frac{\phi_X(t) - \phi_X(-t)}{it} dt.$$

7.1.3 Taylor Series

The next result due to Shaw and McCabe (2009), will be useful in expressing the coefficients of a Taylor series expansion of the quantile.

Theorem 29. *Suppose X is a random variable with quantile function Q_X and characteristic function ϕ_X. Assume further that Q_X is infinitely differentiable at $u \in (0,1)$, then for $n \geq 2$,*

$$Q_X^{(n)}(u) = [Q'_X(u)]^{n+1} P_n[Q'_X(u); Q_X(u)], \tag{7.1.8}$$

where $P_n[Q'_X(u); Q_X(u)]$ is a polynomial in $Q'_X(u)$ of degree $n-2$ and is given by the recurrence relation,

$$P_n\left[x;Q_X\right] = \begin{cases} \int_{-\infty}^{\infty} it\phi_X\left(t\right)e^{-itQ_X(u)}dt & n=2 \\ nxP_2\left[x;Q_X\right]P_{n-1}\left[x;Q_X\right] & \\ \quad +x^2P_2\left[x;Q_X\right]\frac{\partial}{\partial x}P_{n-1}\left[x;Q_X\right] & \\ \quad +\frac{\partial}{\partial Q_X}P_{n-1}\left[x;Q_X\right] & n\geq 3 \end{cases} \tag{7.1.9}$$

Proof. For $n=2$, the characteristic quantile equation (7.1.7) can be written as

$$Q_X''\left(u\right)=\left[Q_X'\left(u\right)\right]^3P_2\left[Q_X'\left(u\right);Q_X\left(u\right)\right], \tag{7.1.10}$$

which is clearly of the form (7.1.8). Assume that

$$Q_X^{(n)}\left(u\right)=\left[Q_X'\left(u\right)\right]^{n+1}P_n\left[Q_X'\left(u\right);Q_X\left(u\right)\right],$$

then,

$$\begin{aligned} Q_X^{(n+1)}\left(u\right) &= \left(n+1\right)\left[Q_X'\left(u\right)\right]^{n+1}Q_X''\left(u\right)P_n\left[Q_X'\left(u\right);Q_X\left(u\right)\right] \\ &\quad +\left[Q_X'\left(u\right)\right]^{n+1}\frac{\partial}{\partial u}P_n\left[Q_X'\left(u\right);Q_X\left(u\right)\right] \end{aligned} \tag{7.1.11}$$

From the chain rule we have,

$$\frac{\partial}{\partial u}P_n\left[Q_X'\left(u\right);Q_X\left(u\right)\right]=\left[Q_X'\left(u\right)\right]^3P_2\left[Q_X'\left(u\right);Q_X\left(u\right)\right]\frac{\partial P_n}{\partial Q_X'}+\frac{\partial P_n}{\partial Q_X}Q_X'. \tag{7.1.12}$$

Substituting (7.1.10) and (7.1.12) into (7.1.11) then yields,

$$Q_X^{(n+1)}\left(u\right)=\left(n+1\right)\left[Q_X'\left(u\right)\right]^{n+2}P_{n+1}\left[Q_X'\left(u\right);Q_X\left(u\right)\right].$$

□

Let, $M_k := \int_{-\infty}^{\infty} t^k\phi_X\left(t\right)e^{-itQ_X(u)}dt$, then the first few terms of the sequence $\left\{P_n\left(x;Q_x\left(u\right)\right)\right\}_{n=2}^{\infty}$ are given by,

$$\begin{aligned}
P_2\left(x;Q_X\left(u\right)\right) &= iM_1 \\
P_3\left(x;Q_X\left(u\right)\right) &= \frac{3i^2}{4\pi^2}M_1^2x+\frac{M_2}{2\pi} \\
P_4\left(x;Q_X\left(u\right)\right) &= \frac{15i^3}{8\pi^3}M_1^3x^2+\frac{5i}{2\pi^2}M_1M_2x+\frac{i^3}{2\pi}M_3 \\
P_5\left(x;Q_X\left(u\right)\right) &= \frac{105i^4}{16\pi^4}M_1^4x^3+\frac{105i^2}{8\pi^3}M_1^2M_2x^2+\frac{5}{4\pi^2}\left(2M_2^2+3M_1M_3\right)x+\frac{i^2}{2\pi}M_4 \\
P_6\left(x;Q_X\left(u\right)\right) &= \frac{945i^5}{32\pi^5}M_1^5x^4+\frac{315}{4\pi^4}i^3M_1^3M_2x^3+\frac{35i}{4\pi^3}\left(4M_1M_2^2+3M_1^2M_3\right)x^2 \\
&\quad -\frac{7i}{4\pi^2}\left(5M_2M_3+3M_1M_4\right)x+\frac{i}{2\pi}M_5 \qquad (7.1.13)
\end{aligned}$$

As in theorem 29 if we assume the quantile function Q_X is infinitely differentiable, then its Taylor series expansion about the point u_0 takes the form,

$$\begin{aligned}
Q_X\left(u\right) &= \sum_{n=0}^{\infty}\frac{Q_X^{(n)}\left(u_0\right)}{n!}\left(u-u_0\right)^n \\
&= \sum_{n=0}^{\infty}\frac{1}{n!}\left[Q_X'\left(u_0\right)\right]^{n+1}P_n\left[Q_X'\left(u_0\right);Q_X\left(u_0\right)\right]\left(u-u_0\right)^n \qquad (7.1.14) \\
&= Q_X'\left(u_0\right)\sum_{n=0}^{\infty}\frac{P_n\left[Q_X'\left(u_0\right);Q_X\left(u_0\right)\right]}{n!}v^n \qquad (7.1.15)
\end{aligned}$$

where $v=Q_X'\left(u_0\right)\left(u-u_0\right)$. Of course a point of expansion u_0 must be chosen. Due to possible simplifications, the zero quantile location makes a worthy candidate, see corollary 28. Such a u_0 implies $Q_X\left(u_0\right)=0$ and hence $M_k=\int_{-\infty}^{\infty}t^k\phi_X\left(t\right)dt$. Thus we may write

$$Q_X\left(u\right) = Q_X'\left(u_0\right)\sum_{n=0}^{\infty}\frac{P_n\left[Q_X'\left(u_0\right);0\right]}{n!}v^n. \qquad (7.1.16)$$

The final ingredient required to compute the terms of the Taylor series is to evaluate the derivative of Q_X at u_0. From equation (7.1.5) we obtain

$$Q_X'\left(u_0\right)=\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\phi_X\left(t\right)dt\right)^{-1}.$$

Further simplifications are possible if the distribution of X is symmetric (i.e. its density f_X is an even function, assuming $\mu=0$) then Q_X will be an odd function about $u=1/2$. Recall that the Taylor series of an odd function has non-zero coefficients only for the

odd degree terms. If also we denote the zero-quantile location by u_0, again see corollary 28 then (7.1.16) can be written as

$$Q_X(u) = Q'_X(u_0) \sum_{n=0}^{\infty} \frac{P_{2n+1}[Q'_X(1/2);0]}{(2n+1)!} v^{2n+1} \tag{7.1.17}$$

where

$$Q'_X(1/2) = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_X(t)\, dt \right)^{-1}. \tag{7.1.18}$$

Another useful theorem relating the characteristic function to the derivatives of the density function is,

Theorem 30. *If X is a random variable with density f_X and characteristic function ϕ_X, then*

$$f_X^{(k)}(0) = \frac{(-i)^k}{2\pi} M_k,$$

where,

$$M_k := \int_{-\infty}^{\infty} t^k \phi_X(t)\, dt.$$

We now look at a concrete example to demonstrate some of the ideas discussed in this section.

Example 31 (Symmetric Stable Distributions)**.** The stable distribution requires four parameters to describe, an index of stability $\alpha \in (0, 2]$, a skewness parameter $\beta \in [-1, 1]$, a scale parameter $\gamma > 0$ and a location parameter $\mu \in \mathbb{R}$. By taking advantage of the location and scale invariance of the characteristic quantile equation we may set $\mu = 0$ and $\gamma = 1$. Then in the symmetric case $\beta = 0$, the characteristic function of the stable distribution is given by, see for example Weron (2004),

$$\phi_X(t) = \exp\{-|t|^{\alpha}\}.$$

In general the probability density function of the stable distribution does not have a closed form expression. We give a series representation using the above analysis. Since the density in this case is an even function its Maclauren series takes the form,

$$f_X(x) = \sum_{n=0}^{\infty} \frac{f^{(2n)}(0)}{(2n)!} x^{2n} = \frac{1}{2\pi} \sum_{n=0}^{\infty} (-1)^n \frac{M_{2n}}{(2n)!} x^{2n}. \tag{7.1.19}$$

Where we have used theorem 30 to obtain the second equality. The coefficients M_{2n} may be computed as follows,

$$M_{2n} = \int_{-\infty}^{\infty} t^{2n} \exp\left\{-|t|^{\alpha}\right\} dt = 2\int_{0}^{\infty} t^{2n} \exp\left\{-t^{\alpha}\right\} dt.$$

By making the change of variable $y = t^{\alpha}$ we obtain,

$$M_{2n} = \frac{2}{\alpha}\Gamma\left(\frac{2n+1}{\alpha}\right), \quad n = 0, 1, 2, \ldots \tag{7.1.20}$$

We could now of course integrate the resultant series (7.1.19) term by term to obtain a series representation of the associated c.d.f. This series could then be inverted to obtain a series representation of the associated quantile function. Note however the whole point of this section was to short cut this method; that is we derive a series representation of quantile function directly from the characteristic function. The objective is to solve the recurrence relationship (7.1.9) and substitute into (7.1.17) to obtain a series representation for the quantile function. Note $M_{2n+1} = 0$, for $n = 0, 1, 2, \ldots$ and from (7.1.18) we obtain

$$Q'_X\left(1/2\right) = \left(\frac{1}{\pi}\int_{0}^{\infty} \exp\left\{-t^{\alpha}\right\} dt\right)^{-1} = \left(\frac{1/\alpha\Gamma\left(1/\alpha\right)}{\pi}\right)^{-1} = \frac{\pi}{\Gamma\left(1+\alpha\right)}.$$

From an implementation point of view one approach as suggested by Shaw and McCabe (2009) is to use a computer algebra system such as Mathematica (Wolfram-Research, 2010) to generate and store the polynomials of the form (7.1.13) to disk, and then program the coefficients M_n targeted for a specific distribution. However it is not a difficult task to write a polynomial class capable of differentiating itself in lower level languages such as C++. With such a class in place the implementation of the recursively defined polynomials P_n as in theorem 29 is a straightforward task. Along with the numerical methods to be discussed in chapter 8 this technique hugely improves the practicality of the momentum space approach described above.

The bulk of the work involved in computing the terms of the Taylor series (7.1.14) or in the symmetric case (7.1.17) is to compute the polynomials P_n through the recursive definition (7.1.9). Therefore it would be advantages to solve this recurrence relationship. At first it is difficult to see any pattern emerging in the first few terms of the sequence (7.1.13), from which one could form a conjecture. However notice the similarity between the sequence of polynomials $\left\{P_n\left(x; Q_x\left(u\right)\right)\right\}_{n=2}^{\infty}$ and the sequence $\left\{\tilde{P}_n\left(x\right)\right\}_{n=2}^{\infty}$ defined as follows; let $\left\{a_n\right\}_{n=1}^{\infty}$ be some undetermined sequence, $\left\{M_n\right\}_{n=1}^{\infty}$ be defined as above and define

$$\tilde{P}_n\left(x\right) := \sum_{k=1}^{n-1} a_n B_{n,k}\left(M_1, \ldots, M_{n-k+1}\right) x^{k-1}, \ n \geq 2$$

where $B_{n,k}(\cdot)$ are the Bell polynomials, for an introduction and definition see Comtet (1974). The first few terms of this sequence can be computed as,

$$\begin{aligned}
\tilde{P}_2(x) &= a_1 M_1 \\
\tilde{P}_3(x) &= a_2 M_1^2 x + a_1 M_2 \\
\tilde{P}_4(x) &= a_3 M_1^3 x^2 + 3a_2 M_1 M_2 x + a_1 M_3 \\
\tilde{P}_5(x) &= a_4 M_1^4 x^3 + 6a_3 M_1^2 M_2 x^2 + a_2 \left(3M_2^2 + 4M_1 M_3\right) x + a_1 M_4 \\
\tilde{P}_6(x) &= a_5 M_1^5 x^4 + 10a_4 M_1^3 M_2 x^3 + a_3 \left(15M_1 M_2^2 + 10M_1^2 M_3\right) x^2 \\
&\quad + a_2 \left(10M_2 M_3 + 5M_1 M_4\right) x + a_1 M_5.
\end{aligned} \tag{7.1.21}$$

Notice the striking similarity between the first few terms of the sequences (7.1.13) and (7.1.21). Perhaps this provides a clue into solving the recurrence relationship (7.1.9). Solving this recurrence relationship would certainly improve the practicality of the technique and allow us to compute high order terms much more efficiently. To this end one must wonder if we can repeat the above analysis with the characteristic function replaced by the moment generating function when it exists. Possibly the polynomials would be simplified, certainly the even numbered polynomials will not depend on i.

7.2 The Lagrange Approach

Under the appropriate conditions Lagrange's inversion formula is capable of providing us with a series representation of functional inverses. Yet it seems to be ignored in the literature when one wants to find an approximation to the quantile function, which itself is at least in the continuous case, defined as the functional inverse of the c.d.f. Based on this observation we provide a convergent series representation for the quantile function of the asymmetric α-stable distribution. Note however the method is much more general than this; all it requires is a power series representation of the c.d.f.

Suppose the c.d.f. F_X of a random variable X has the Taylor series representation,

$$F_X(x) = \sum_{n=0}^{\infty} \frac{f_n}{n!} x^n,$$

The relationship between F_X and the associated quantile function Q_X is given by $F_X(Q_X(u)) = u$, for all $u \in (0,1)$. We would like to find the Taylor series expansion of the inverse function Q_X. One such expression to achieve this is provided by Lagrange's Inversion formula, see Aldrovandi (2001, § 13.3). However Lagrange's inversion formula applies to nonunits (i.e. power series with no constant term see Henrici

(1974, p. 12)), to this end define the function $H(x) = F_X(x) - f_0$. Applying Lagrange's inversion formula to H then yields a series representation for inverse function $H^{-1}(u) = Q(u + f_0)$,

$$H^{-1}(u) = \sum_{n=1}^{\infty} \frac{q_n}{n!} u^n.$$

Consequently the Taylor series expansion of $Q_X(u) = H^{-1}(u - f_0)$ may be written as

$$Q_X(u) = \sum_{n=1}^{\infty} \frac{q_n}{n!} (u - u_0)^n$$

where $u_0 = f_0$ and the coefficients q_n are provided by Lagrange's Inversion formula written in terms of Bell Polynomials (Aldrovandi, 2001, § 13.3)

$$q_n = \begin{cases} \frac{1}{f_1} & n = 1 \\ -\frac{1}{f_1^n} \sum_{k=1}^{n} q_k \mathbb{B}_{n,k}(f_1, \ldots, f_{n-k+1}) & n \geq 2 \end{cases}, \tag{7.2.1}$$

Here the coefficients $\mathbb{B}_{n,k}(f_1, \ldots, f_{n-k+1})$ appearing in the sum are known as Bell polynomials and are defined by a rather detailed expression

$$\mathbb{B}_{n,k}(f_1, \ldots, f_{n-k+1}) := \sum_{\substack{v_1, v_2, \ldots \geq 0 \\ v_1 + v_2 + \cdots = k \\ v_1 + 2v_2 + 3v_3 \cdots = n}} \frac{n!}{\prod_{j=1}^{n} [v_j! (j!)^{v_j}]} f_1^{v_1} f_2^{v_2} \cdots f_{n-k+1}^{v_{n-k+1}}, \tag{7.2.2}$$

where the summation is taken over all solutions $(v_1, v_2, \ldots, v_n) \in \mathbb{Z}_{\geq 0}^n$ to the Diophantine equation

$$v_1 + 2v_2 + 3v_3 \cdots + nv_n = n \tag{7.2.3}$$

with the added constraint, the sum of the solutions is equal to k, i.e.$\sum_{j=0}^{n} v_j = k$. For example the solutions (v_1, v_2, v_3, v_4) for $n = 4$ are given by

$$\{(4,0,0,0),\ (2,1,0,0),\ (0,2,0,0),\ (0,0,0,1),\ (1,0,1,0)\}$$

and if $k = 2$ this picks out the solutions $(0,2,0,0)$ and $(1,0,1,0)$. Note that solutions to (7.2.3) correspond exactly to the integer partitions of n. An integer partition of a number n is an unordered sequence of positive integers whose sum is equal to n. The added constraint implies we should look for partitions in which the number of non zero summands is equal to k. For example the integer partitions of $n = 4$ are given by $(1,1,1,1)$, $(1,1,2)$, $(2,2)$, (4), and $(1,3)$, and in the case $k = 2$ this singles out

the partitions $(2,2)$ and $(1,3)$. Thus to summarize the sum in (7.2.2) is taken over all integer partitions of n in which the number of summands is given by k.

The above recursion (7.2.1) can be solved (see Aldrovandi (2001, § 13.3)) leading to a computationally more efficient expression for the coefficients,

$$q_n = \begin{cases} \frac{1}{f_1} & n = 1 \\ \frac{1}{f_1^n} \sum_{k=1}^{n-1} (-1)^k \frac{(n+k-1)!}{(n-1)!} \mathbb{B}_{n-1,k}\left(\frac{f_2}{2f_1}, \frac{f_3}{3f_1}, \ldots, \frac{f_{n-k+1}}{(n-k+1)f_1}\right) & n \geq 2 \end{cases}. \tag{7.2.4}$$

7.2.1 α-Stable Distribution.

The α-stable distribution, denoted $\mathrm{S}_\alpha(\beta, \mu, \sigma)$ is commonly characterized through its characteristic function. However there are many different parametrizations in existence which has lead to much confusion. Thus from the outset we state explicitly the two parametrizations we will work with in this report and provide the relationships between them. We will call these parametrizations P_1 and P_2 respectively. In the following a subscript under a parameter denotes the parametrization being used. We will primarily work with Zoltarev's type (B) parametrization, see Zolotarev (1986, p. 12) denoted by P_2. In this parametrization the characteristic function takes the form,

$$\phi(t) = \exp\left\{\sigma_2\left(it\mu_2 - |t|^\alpha e^{-i(\pi/2)\beta_2 K(\alpha)\mathrm{sgn}(t)}\right)\right\}, \tag{7.2.5}$$

where $\alpha \in (0,2]$ is the tail index, $\mu_2 \in \mathbb{R}$ is a location parameter, $\sigma_2 > 0$ is a scale parameter, $\beta \in [-1,1]$ is an asymmetry parameter and $K(\alpha) := \alpha - 1 + \mathrm{sgn}(1-\alpha)$. P_1 is the classic parametrization, and is probably the most common due to the simplicity of the characteristic function given by, see Samorodnitsky and Taqqu (1994, p. 5),

$$\phi(t) = \exp\left\{-\sigma_1^\alpha |t|^\alpha \left(1 - i\beta_1 \tan\left(\frac{\pi\alpha}{2}\right)\mathrm{sgn}(t)\right) + i\mu_1 t\right\}. \tag{7.2.6}$$

A connection between the parametrization P_2 and P_1 can be derived by taking logarithms and equating first the real parts of (7.2.5) and (7.2.6), followed by the coefficients of t and $|t|^\alpha$ in the imaginary parts, leading to the set of relations,

$$\begin{aligned} \mu_1 &= \mu_2\sigma_2 \\ \sigma_1 &= \left(\cos\left(\frac{1}{2}\pi K(\alpha)\beta_2\right)\sigma_2\right)^{\frac{1}{\alpha}} \\ \beta_1 &= \cot\left(\frac{\pi\alpha}{2}\right)\tan\left(\frac{1}{2}\pi K(\alpha)\beta_2\right). \end{aligned}$$

Despite the fact no closed form expression for the c.d.f. $F_\alpha(x;\beta,\mu,\sigma)$ of the stable distribution in the general case is known, it can be expressed in terms of an infinite series expansion. As is usual for a location-scale family of probability distributions, without loss of generality we may set the location μ_2 and scale σ_2 parameters to 0 and 1 respectively. In addition it is sufficient to consider expansions of the c.d.f. $F_\alpha(x;\beta,0,1)$ for values of $x > 0$ only since the following equality holds,

$$F_\alpha(x;\beta,0,1) = 1 - F_\alpha(-x;-\beta,0,1). \tag{7.2.7}$$

Theorem 32. *Let $X \sim S_\alpha(x;\beta_2,0,1)$ be a standard α-stable random variable. Then the cumulative distribution function of X admits the following infinite series representations,*

$$F_\alpha(x;\beta_2,0,1) = \sum_{n=0}^{\infty} \frac{f_n}{n!} x^n, \tag{7.2.8}$$

where

$$f_n = \begin{cases} \frac{1}{2}\left(1 - \frac{\beta_2 K(\alpha)}{\alpha}\right) & n = 0 \\ (-1)^{n-1}\frac{1}{\pi}\frac{\Gamma\left(\frac{n}{\alpha}+1\right)}{n}\sin\left(\frac{\pi n}{2}\left(1+\frac{\beta_2 K(\alpha)}{\alpha}\right)\right) & n \geq 1 \end{cases},$$

and

$$F_\alpha(x;\beta_2,0,1) = \sum_{n=0}^{\infty} \frac{\tilde{f}_n}{n!} x^{-\alpha n}, \tag{7.2.9}$$

where

$$\tilde{f}_n = \begin{cases} 1 & n = 0 \\ (-1)^n \frac{1}{\alpha\pi}\frac{\Gamma(\alpha n+1)}{n} \sin\left(\frac{n\pi}{2}(\alpha+\beta_2 K(\alpha))\right) & n \geq 1 \end{cases},$$

If $1 < \alpha \leq 2$ then (7.2.8) is absolutely convergent for all $x > 0$ and if $0 < \alpha < 1$ then (7.2.8) is an asymptotic expansion of $F_\alpha(x;\beta_2,0,1)$ as $x \to 0$. In contrast if $0 < \alpha < 1$ then the series (7.2.9) is absolutely convergent for $x > 0$ and if $1 < \alpha \leq 2$ then (7.2.9) is an asymptotic expansion of $F_\alpha(x;\beta_2,0,1)$ as $x \to \infty$.

Proof. The proof proceeds by obtaining a series expansion of the density of X. This is achieved by applying the inverse Fourier transform to (7.2.5), expanding the exponential function, and then performing a contour integration, see Lukacs (1970, § 5.8) for details. The expansion in the density can be integrated term by term to obtain an expansion of the c.d.f. □

Note when $1 < \alpha \leq 2$, the series (7.2.8) rapidly converges for values of x near zero, where as (7.2.9) converges rapidly for large values of x. The opposite is true when $0 < \alpha < 1$. Thus we can now apply Lagrange's inversion formula to find a series expansion of the quantile function Q_α in the central and tail regions. Since the expansions (7.2.8) and (7.2.9) are valid only for $x > 0$, the resulting expansions

of Q_α are only valid for $u > u_0$, where u_0 is the zero quantile location defined by $u_0 := F_\alpha(0; \beta_2, 0, 1)$. This does not pose a restriction however since it follows from (7.2.7)

$$Q_\alpha(u; \beta, 0, 1) = -Q_\alpha(1-u; -\beta, 0, 1).$$

To find the functional inverse of (7.2.8) we apply Lagrange's inversion formula to the function $F_\alpha(x) - f_0$, the quantile function then has the following infinite series representation valid for $u > u_0$,

$$Q_\alpha(u; \beta, 0, 1) = \sum_{n=1}^{\infty} \frac{q_n}{n!} (u - u_0)^n \tag{7.2.10}$$

where the coefficients q_n are given by (7.2.4) and

$$u_0 = f_0 = \frac{1}{2}\left(1 - \frac{\beta_2 K(\alpha)}{\alpha}\right).$$

To find the functional inverse of (7.2.9) we make the change of variable $y := x^{\frac{1}{\alpha}}$, and apply Lagrange's inversion formula to the power series,

$$G(y) := \sum_{n=1}^{\infty} \frac{\tilde{f}_n}{n!} y^n.$$

The quantile function is then given by

$$Q_\alpha(u; \beta, 0, 1) = \left[G^{-1}(u-1)\right]^{-\frac{1}{\alpha}},$$

where

$$G^{-1}(u) = \sum_{n=1}^{\infty} \frac{\tilde{q}_n}{n!} u^n, \tag{7.2.11}$$

and the coefficients $\tilde{q}_n$ are given by (7.2.4) with f_n replaced by $\tilde{f}_n$. Note when $1 < \alpha \leq 2$ the series (7.2.10) rapidly converges for values of u near u_0, where as (7.2.11) converges rapidly for values of u close to 1. In this case partial sums of (7.2.10) serve as good approximations of the quantile function in the central regions where as partial sums of (7.2.11) can be used to approximate the tails. The opposite is true when $0 < \alpha < 1$. The first few terms of (7.2.10) are given by,

$$Q_\alpha(u;\beta,0,1) = \frac{\pi \text{csc}\left(\pi\rho\right)}{\Gamma\left(1+\frac{1}{\alpha}\right)}\left(u-u_0\right) + \frac{\pi^2 \text{cot}\left(\pi\rho\right)\text{csc}\left(\pi\rho\right)\Gamma\left(\frac{2+\alpha}{\alpha}\right)}{2\Gamma\left(1+\frac{1}{\alpha}\right)^3}\left(u-u_0\right)^2 + \text{O}\left(\left(u-u_0\right)^3\right)$$

Implementation is a straightforward matter, high level languages such as Mathematica have a built in implementation of the Bell polynomials. We provide a Mathematica implementation used to generate series in figure C.0.2 on page 146. As long as we have a series representation of the c.d.f. using this approach we could derive a series representation for the associated Quantile function. However there is one obvious drawback, even though the coefficients q_n can be computed using elementary algebraic operations, the number of partitions $p\left(n\right)$ of an integer n grows exponentially with n, thus for large values of n the sum in (7.2.2) may be computationally expensive due to the large number of summands.

However the problem of reverting a power series is a classical one in mathematics and many efficient algorithms have been devised as a result. For example Dahlquist and Björck (2008) present a convenient but not so efficient algorithm based on Toeplitz matrices requiring $O\left(N^3 \log N\right)$ floating point operations (flops). Knuth (1998, § 4.7) gives several algorithms for power series reversion including a classical algorithm due to Lagrange (1768) that requires $O\left(N^3\right)$ flops to compute the first N terms. More recently Brent and Kung (1978) provide an algorithm which requires only $O\left(N \log N\right)^{3/2}$ flops. However as we shall see in our numerical methods chapter 8, rarely is there a need to compute more than 200 terms. Due to the constants of proportionaly involved Knuth states that N must become quite large before Lagrange's classical algorithm loses out to Brent and Kung's high speed method. Brent and Kung (1978) also state in their paper that they have not tested the algorithm extensively for stability problems. In addition, Lagrange's classical algorithm is far simpler to program. It is for these reasons we advise the reader interested in implementing the above scheme to compute the α-stable quantile series to base their implementation on Lagrange's classical algorithm.

Concerning the numerical evaluation of the distribution function F_α of the stable distribution it has been remarked by various authors (Stoyanov and Racheva-Iotova, 2004) that the series expansions given in theorem 32 are only useful for approximating F_α for either small or large values of x, due to the slow convergence of the series. It is for this reason standard methods such as the Fast Fourier transform or numerical quadrature techniques are applied to evaluate F_α. However we found in our experimentation that series acceleration techniques such as Padé approximants and Levin type transforms could be applied to (7.2.8) and (7.2.9), see section 8.3. The same comments apply for the series representation of the density and quantile functions. Con-

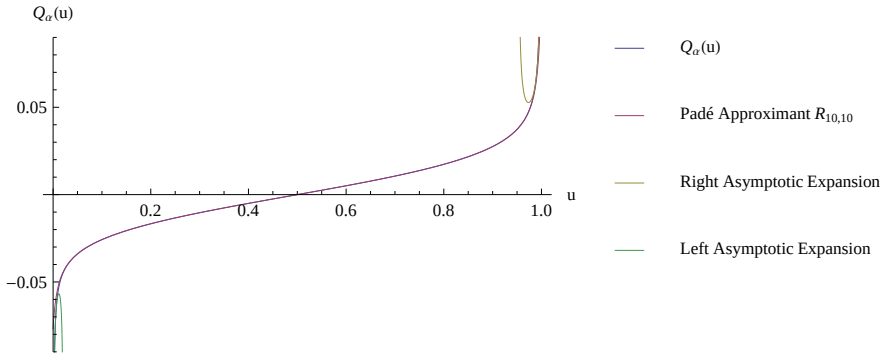


Figure 7.2.1: Plot of Mathematica's implementation of $Q_\alpha(u;\beta_1,\mu_1,\sigma_1)$ where $\alpha = 1.7811$, $\beta_1 = 0.2834$, $\mu_1 = 0.0009$ and $\sigma_1 = 0.0141$, the Padé approximant $R_{10,10}(u)$ and 4^{th} order asymptotic expansions.

sider the following Mathematica Wolfram-Research (2010) code which depends on the functions defined in figure C.0.2 on page 146.

```
nN = 4;
padeQC[u_] = PadeApproximant[QC[u, 20], {u, 0, {10,
   10}}];

(*Cache the cdf coefficients*)
Table[ft[n, \[Beta]2], {n, 0, nN}];
Table[ft[n, -\[Beta]2], {n, 0, nN}];

rightQTail[u_] = \[Mu]1 + \[Sigma]1 QT[u, \[Beta]2, nN
   ]^(-1/\[Alpha]);
leftQTail[    u_] = \[Mu]1 + \[Sigma]1 (-QT[1 - u, -\[
   Beta]2, nN]^(-1/\[Alpha]));

Plot[{InverseCDF[    StableDistribution[1, \[Alpha], \[
   Beta]1, \[Mu]1, \[Sigma]1],     u], \[Mu]1 + \[Sigma]1
    padeQC[u - f0], rightQTail[u],    leftQTail[u]}, {u,
   0, 1}, PlotRange -> {-0.09, 0.09}]
```

The code uses the first 20 terms of the Taylor Series (7.2.10) to generate a Padé approximant $R_{10,10}(u)$ whose graph is visually indistinguishable from Mathematica's implementation of the α-stable quantile, see figure 7.2.1. Although this is promising, we do not have access to a benchmark implementation of the α-stable quantile so unfortunately we can not provide a more precise error analysis (we are not hold much faith in the accuracy of Mathematica's implementation of the α-stable quantile).

Chapter 8

Numerical Techniques and Examples

The goal of this chapter is to discuss some techniques to approximate the VG $(\lambda, \alpha, \beta, \mu)$, Hyp $(\alpha, \beta, \delta, \mu)$ and GIG (λ, χ, ψ) quantile functions. Some of these techniques may be incorporated into algorithms which accept a set of distribution parameters and construct at runtime an approximation Q_A to the quantile function satisfying some prespecified accuracy requirements. The techniques discussed here rely heavily on the results developed in chapter 6, for example the Taylor expansion of Q may be used to construct Padé approximants, Chebyshev expansions and minimax approximants.

We shall call the time it takes to construct an approximation Q_A the setup time, and the time it takes to evaluate Q_A at a point u the execution time of the algorithm. Ideally one would like low setup and execution times, but usually there is a trade-off between the two.

The procedures assume the availability of an integration and root finding routine to compute the distribution and quantile functions respectively to full (or at least near) machine precision. These routines will be used to compute the initial conditions and manage the error in the approximation of Q. To begin with we partition the unit interval and for convenience name each part as follows,

$$\begin{aligned}
(0, \tau_L) &= \text{Left Tail Region} \\
[\tau_L, u_1) &= \text{Left Region} \\
[u_1, u_2] &= \text{Central Region} \\
(u_2, \tau_R] &= \text{Right Region} \\
(1 - \tau_R, 1) &= \text{Right Tail Region}
\end{aligned}$$

This five part partition is motivated by the behaviour of the quantile function, the parts are constructed so that Q has small variation in the central region, relatively large variation in the left and right regions and possibly infinite variation in the tail regions.

Consequently we will build approximations to Q directly in the central region, in the left and right regions we will apply a change of variable and build approximations to the transformed functions, and finally in the tail regions we will employ the asymptotic expansions of Q.

When constructing an approximation a suitable measure of the error must be chosen, some common choices are:

- absolute error
$$e_{\text{abs}}(u) = |Q(u) - Q_A(u)|$$
- relative error
$$e_{\text{rel}}(u) = \frac{e_{\text{abs}}(u)}{|Q(u)|}$$
- u-error
$$e_{\text{u}}(u) = |u - F(Q_A(u))|$$

Note that the first two of these measures require the exact computation of $Q(u)$. In addition as pointed out by Devroye (1986) and Hörmann et al. (2004) the absolute error requires the approximation to be overly accurate in the tail regions, the relative error is not defined at the zero quantile location and the u-error can be problematic when the c.d.f. F is too steep. Therefore no universal recommendation can be made.

Code for all the algorithms we consider is given. We use Mathematica Wolfram-Research (2010) as our programming environment; we use it as a rapid prototyping tool, we do not rely on Mathematica's symbolic capabilities so expect the algorithms to be easily portable to lower level languages. In fact for the purposes of constructing approximations to the quantile function Q the symbolic capabilities of Mathematica are of little use to us, for two key reasons:

1. the built in functions for the hyperbolic and generalized inverse Gaussian quantiles are not reliable, see section 8.1, in addition there is no built in quantile function for the variance gamma distribution.

2. Mathematica has no idea how to differentiate the quantile functions of the hyperbolic, variance gamma or generalized inverse Gaussian distributions. Therefore calls to functions such as `Series[]`, `PadeApproximant[]` or `MiniMaxApproximation[]`[1] to generate the Taylor series, Padé and Chebyshev approximants respectively will fail when applied to these functions.

[1] `MiniMaxApproximation[]` needs the first two derivatives of Q to compute the extremal points of the optimal error curve.

On the other hand Mathematica has no issue differentiating the Normal quantile, so one may easily construct Taylor expansions, Padé and Chebyshev approximants for this function as demonstrated by the code below.

```
Q[u_] := InverseCDF[NormalDistribution[0, 1], u]
(*Taylor Series about u=1/2*)
Series[Q[u], {u, 1/2, 3}]
(*Pade Approximant*)
PadeApproximant[Q[u], {u, 1/2, {3, 3}}] // Simplify
(*Chebyshev Approximant*)
GeneralMiniMaxApproximation[{p, Q[p], 1}, {p, {10^-3, 1 -
    10^-3}, 7,    7}, u]
```

However all is not lost, the "ingredients" that we developed in chapter 6 will prove invaluable in constructing these types of approximants for the hyperbolic, variance gamma or generalized inverse Gaussian quantiles. We discuss the details and implementation of some numerical recipes next. To make the discussion more concrete we consider applying the methods to the hyperbolic quantile $Q_{\text{Hyp}}(u; \alpha, \beta, \delta, \mu)$ with parameters $\alpha = 2$, $\beta = 3/2$, $\delta = 1$ and $\mu = 0$, see figure 6.2.1 on page 47. It should be emphasized however that the methods discussed below apply equally well to the variance gamma and generalized inverse Gaussian quantile functions with some minor modifications.

8.1 Root Finder

We consider root finding a bread and butter method. One because all other methods we consider rely on this method to evaluate the quantile function at some specified points and two some built in library routines to compute the quantile function can not be trusted. Take for example the following Mathematica Wolfram-Research (2010) code.

```
(* evaluate the c.d.f. at the point x = -3*)
u0 = N[CDF[HyperbolicDistribution[2, 3/2, 1, 0], -3]]
(* evaluate the quantile function at the point u = F(-3)
   *)
x0 = N[InverseCDF[HyperbolicDistribution[2, 3/2, 1, 0],
   u0]]
```

The output is `u0 = 5.36058*10^-6` and `x0 = -23.0839`. Clearly there is an inconsistency between the build in `CDF` and `InverseCDF` functions, we expect `x0` to be approximately equal to 3. Since numerically integrating the density function agrees with the `CDF` function we believe the `InverseCDF` function is at fault. Thus we

define the quantile function `QRF[]` in terms of a Newton Raphson based method as follows,

```
precision = 30; (*$MachinePrecision*)
F[x_] := CDF[HyperbolicDistribution[2, 3/2, 1, 0], x]
G[x_, u_] := F[x] - u;
(*FindRoot uses Newton Raphson by default*)
QRF[u_] := x /. FindRoot[G[x, u], {x, 0},
   WorkingPrecision -> precision]
```

Now the output of the quantile function is as expected.

```
In[1]:= u0 = F[-3]
        x0 = Q[u0]
Out[1]= 5.360583842001678639566510041 48*10^-6
Out[2]= -3.000000000000000000000000000
```

The problem is not just with Mathematica's implementation of the hyperbolic quantile, we observed similar problems with Mathematica's generalized inverse Gaussian and generalized hyperbolic quantiles.

Note that in this example we have set the initial estimate for the Newton Raphson scheme as the location parameter $\mu = 0$. This is a naive choice and as we will see in section 8.3.3 by using a Padé approximant as the initial estimate we can improve upon the rate of convergence of the numerical scheme.

8.2 Numerical Integrators

Numerical integrators are used to find numerical solutions of differential equations. We may apply them to the first order quantile equation $Q' = 1/f(Q)$ to approximate the quantile function. The Mathematica code below uses the built in function `NDSolve` to solve the quantile equation in the case of the hyperbolic distribution on the interval $[\epsilon_L, 1 - \epsilon_R]$, where the we have chosen the cutoff points $\epsilon_L = \epsilon_R = 10^{-9}$ (note Leobacher and Pillichshammer (2002) provide analytical formulae to choose ϵ_L and ϵ_R). Selecting a larger region in the unit interval will result in failure for this example, so the algorithm should be supplemented by the asymptotic expansions developed in chapter 6,

```
In[1]:= \[Epsilon] = 10^-9; \[Mu] = 0; \[Delta] = 1; \[
   Alpha] = 2; \[Beta] = 3/2;
f[x_] := PDF[HyperbolicDistribution[2, 3/2, 1, 0], x]
F[x_] := CDF[HyperbolicDistribution[2, 3/2, 1, 0], x];
xm = N[\[Mu] + \[Beta] \[Delta]/Sqrt[\[Alpha]^2 - \[Beta
   ]^2]];
```

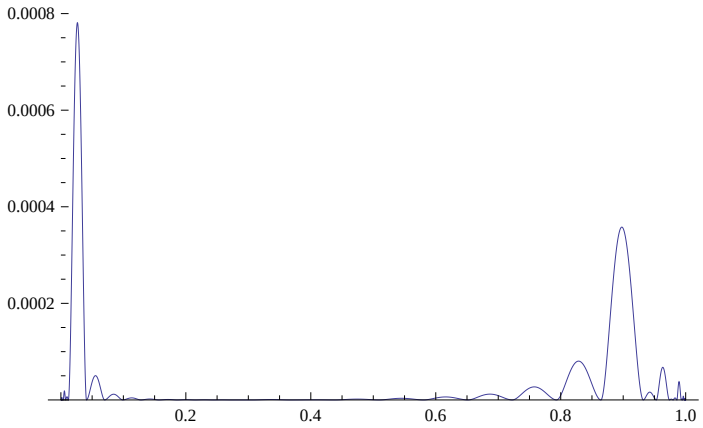


Figure 8.2.1: u-error plot produced by the numerical integrator .

```
um = N[F[xm]];
solution =   NDSolve[{Q'[u] == 1/f[Q[u]], Q[um] == xm},
      Q, {u, \[Epsilon], 1 - \[Epsilon]}, AccuracyGoal ->
    Infinity,    Method -> "StiffnessSwitching"];
Evaluate[Q[N[F[-3]]] /. solution]

Out[1]= {-3.00001}
```

As can be seen from the u-error plot, figure 8.2.1, the accuracy of the method at least when working in double precision is poor.

Given that we can easily generate the Taylor coefficients of Q another plausible approach to approximate the quantile function is to construct a numerical integrator based on Taylor's method. It has been reported by many authors that when high precision is required this is the method of choice, see for example Corliss and Chang (1982), Barrio et al. (2005) and Jorba (2005). The idea here is to discretize the domain $[a, b] \subset (0, 1)$ into a non-uniform grid $a = u_0, \ldots, u_n = b$. To build this grid we must determine the step sizes $h_k = u_k - u_{k-1}$ for $k = 1, \ldots n$. In addition at each grid point u_k we must determine the order m_k of Taylor polynomial so that the required accuracy goals are achieved. That is both the stepsize and order are variable. Based on two or more terms of the Taylor series certain tests have been devised to compute the h_k and m_k, see again the references mentioned above and the review article Halin (1983). The result of Taylor's method is a piecewise polynomial approximation to the quantile function, note however we will not discuss this method further here.

8.3 Continued Fractions and Padé Approximants

Continued fractions and Padé approximants can be used for the summation of various types of series. We give a brief introduction, followed by some examples demonstrating the strengths of these types of approximations. In the context of approximating quantile

functions some simple techniques to manage the error and construct an appropriate partition will also be considered.

8.3.1 Continued Fractions

Much like an infinite sum $\sum_{n=0}^{\infty} a_n$ is a limit process involving addition and an infinite product $\prod_{n=0}^{\infty} a_n$ is a limit process involving multiplication, continued fractions are limit processes involving division. An expression of the form

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \ddots}}}$$

where a_k and b_k are complex numbers and $a_k \neq 0$ for all $k \geq 1$ is called a continued fraction and can be written more compactly as

$$b_0 + \operatorname*{\mathcal{K}}_{n=1}^{\infty} \left(\frac{a_n}{b_n} \right)$$

or

$$b_0 + \frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_n}{b_n} + \cdots$$

The quantities a_n and b_n are called the partial numerators and partial denominators respectively. Analogous to partial sums of an infinite series we can form the n^{th} convergent f_n of a continued fraction defined as

$$\begin{aligned} f_n &= b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \ddots + \cfrac{a_n}{b_n}}}} \\ &= b_0 + \frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_n}{b_n} \end{aligned}$$

A continued fraction $b_0 + \operatorname*{\mathcal{K}}_{n=1}^{\infty} (a_n/b_n)$ is said to converge if and only if the sequence of approximants $\{f_n\}_{n=0}^{\infty}$ converges to a limit $f \in \mathbb{C}$. The n^{th} convergent f_n of a continued fraction may be written as the ratio

$$f_n = \frac{A_n}{B_n}$$

where A_n and B_n are called the n^{th} numerator and denominator respectively and may be computed recursively as follows,

$$A_n = b_n A_{n-1} + a_n A_{n-2}, \qquad B_n = b_n B_{n-1} + a_n B_{n-2} \tag{8.3.1}$$

with $A_{-1} = 1$, $A_0 = b_0$, $B_{-1} = 0$ and $B_0 = 1$.

There are various different types of continued fractions and various methods to construct them, we refer the interested reader to the excellent monograph by Cuyt et al. (2008). In particular we will be interested in continued fractions of the type

$$f(z) = c_0 + \mathop{\mathbb{K}}_{n=1}^{\infty} \left(\frac{a_n z}{1} \right), \quad c_0 \in \mathbb{C}, \quad a_n \in \mathbb{C} \backslash \{0\}, \ n \geq 1,$$

which are known as regular C-fractions. Note that the n^{th} convergent

$$f_n(z) = \frac{A_n(z)}{B_n(z)}$$

is a rational function. The n^{th} numerator $A_n(z)$ is a polynomial of degree $\lceil n/2 \rceil$ and the n^{th} denominator $B_n(z)$ is a polynomial of degree $\lfloor n/2 \rfloor$. Given the formal power series expansion $f(z) = \sum_{n=0}^{\infty} c_n z^n$ the coefficients a_n may be obtained using the progressive form of Rutishauser's quotient difference algorithm (Cuyt et al., 2008, theorem 6.1.2). The algorithm proceeds as follows:

- Compute the coefficients d_n appearing in the expansion

$$\frac{1}{f(z)} = \sum_{n=0}^{\infty} d_n z^n$$

  via theorem 21 on page 36.

- Populate the qd-table consisting of the values $q_l^{(k)}$ and $e_l^{(k)}$ using the recurrence relations

$$\begin{aligned}
q_1^{(0)} &= -\frac{d_1}{d_2}, \quad q_{k+1}^{(-k)} = 0, \quad k \geq 1 \\
e_0^{(1)} &= 0, \quad e_1^{(0)} = \frac{d_2}{d_1}, \quad e_{k+1}^{(-k)} = \frac{d_{k+2}}{d_{k+1}}, \quad k \geq 1 \\
e_l^{(k+1)} &= \frac{q_{l+1}^{(k)}}{q_l^{(k+1)}} e_l^{(k)}, \quad l \geq 1, \ k \geq 1 \\
q_l^{(k+1)} &= q_l^{(k)} + e_l^{(k)} - e_{l-1}^{(k+1)}, \quad l \geq 1, \ k \geq 1
\end{aligned}$$

  where the top two lines are used to initialize the table.

- The coefficients a_n are then given by

$$a_1 = c_1, \quad a_{2l} = -q_l^{(1)}, \quad a_{2l+1} = -e_l^{(1)}, \quad l \geq 1.$$

We provide a `C++` implementation of the algorithm available in an online code repository (Munir, 2012), see the `ProgressiveQD` class and a Mathematica implementation in figure C.0.5 on page 149. Note to construct the n^{th} convergent $f_n(z)$ using this algorithm requires knowledge of the first $n+1$ Taylor coefficients, $c_0, \ldots, c_{n+1}$. We will construct regular C-fractions in section 8.9 where they are used to approximate the VG quantile and in section 8.8 to construct minimax approximations.

8.3.2 Padé Approximants

Again consider the power series representation of a function f given by

$$f(z) = \sum_{n=0}^{\infty} c_n z^n.$$

We introduce the polynomial functions $N_m(z)$ and $D_n(z)$ of maximal degree m and n respectively

$$N_m(z) = a_m z^m + \cdots + a_1 z + a_0, \quad D_n(z) = b_n z^n + \cdots + b_1 z + b_0.$$

A Padé approximant $R_{m,n}(z)$ of f is a rational function of the form

$$R_{m,n}(z) = \frac{N_m(z)}{D_n(z)} \tag{8.3.2}$$

such that the Maclauren expansion of $R_{m,n}(z)$ agrees with that of $f(z)$ as far as possible, more precisely

$$\text{degree } N_m(z) \leq m, \quad \text{degree } D_n(z) \leq n$$

$$\sum_{n=0}^{\infty} c_n z^n = R_{m,n}(z) + O\left(z^{m+n+1}\right).$$

If $n = 0$ then $R_{m,0}(z)$ is simply the m^{th} order Taylor polynomial. Several construction methods are available to compute $R_{m,n}(z)$, see for example Baker and Graves-Morris (1996). Note to construct the Padé approximant $R_{m,n}(z)$ requires knowledge of the first $m+n+1$ Taylor coefficients $c_0, c_1, \ldots, c_{m+n}$. Padé approximants are often superior to Taylor expansions but inferior (in the Chebyshev sense) to some of the other

approximations we consider below. However as the following two examples demonstrate Padé approximants have their advantages.

Example 33 (Series Acceleration)**.** Sometimes Taylor series expansions converge so slow that it renders them useless for numerical purposes. Consider the expansion of the α-stable c.d.f.

$$F_\alpha\left(x;\beta_2,0,1\right)=\sum_{n=0}^{\infty}\frac{f_n}{n!}x^n \tag{8.3.3}$$

where the coefficients f_n are given in theorem 32 on page 87. The following Mathematica code constructs the Taylor polynomial $R_{50,0}\left(x\right)$, the Padé approximant $R_{10,10}$ and plots them against and the α-stable c.d.f. $F_{3/2}\left(x;1/5,0,1\right)$.

```
(*Zoltarev's Type (B) parameterisation*)
\[Alpha] = 3/2; \[Beta]2 = 1/5; \[Mu]2 = 0; \[Sigma]2 =
   1;

(*classical parameterisation*)
k = \[Alpha] - 1 + Sign[1 - \[Alpha]];
\[Mu]1 = N[\[Mu]2*\[Sigma]2];
\[Sigma]1 = N[(Cos[(1/2)*Pi*k*\[Beta]2]*\[Sigma]2)^(1/\[
   Alpha])];
\[Beta]1 = N[Cot[(Pi*\[Alpha])/2]*Tan[(1/2)*Pi*k*\[Beta
   ]2]];
\[Theta] = N[(\[Beta]2/\[Alpha])*k];  \[Rho] = N[(1 + \[
   Theta])/2];

(*cdf coefficients*)
b0 = (1/2)*(1 - (\[Beta]2*k)/\[Alpha]);
b[n_] := Piecewise[{{b[0] = b0, n == 0},           {b[n]
   = (-1)^(n - 1)*(1/Pi)*(Gamma[n/\[Alpha] + 1]/n)*
         Sin[Pi*n*\[Rho]], n >= 1}}];

(*cdf Taylor Series*)
F[x_, n_] := Sum[(b[k]/k!)*x^k, {k, 0, n}];

(*construct Pade approximant*)
pade[x_] = PadeApproximant[F[x, 20], {x, 0, {10, 10}}]

(*plot result*)
```

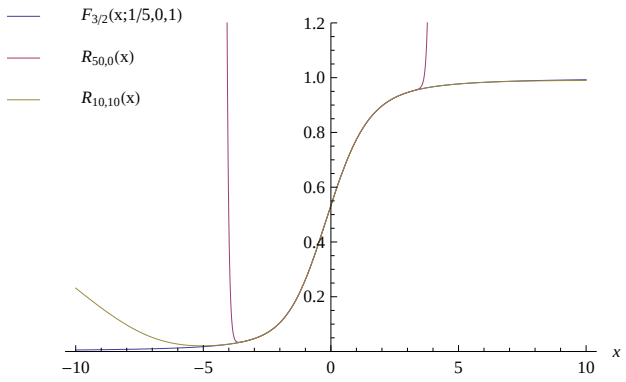


Figure 8.3.1: Series acceleration of the power series of F_α.

```
Plot[{CDF[    StableDistribution[1, \[Alpha], \[Beta]1,
   \[Mu]1, \[Sigma]1], x],    F[x, 50], pade[x]},     {x,
    -10, 10}, PlotRange -> {0, 1.2}]
```

Recall that the series (8.3.3) in this example is valid for $x > 0$ and for values of $x < 0$ we must exploit the relationship (7.2.7). The resulting plot is provided in figure 8.3.1. We see there is no visible difference between $R_{10,10}$ and Mathematica's implementation of the α-stable c.d.f. $F_{3/2}\left(x; 1/5, 0, 1\right)$ for $x > 0$. Unfortunately we may not perform a more detailed error analysis at present since we do not have access to reliable routine capable of computing the α-stable c.d.f. to high accuracy.

Example 34 (Analytic continuation)**.** In this example we will see how the region of convergence is often extended by utilizing Padé approximants. Recall the asymptotic expansion of the Hyperbolic quantile

$$Q(u) \sim y + \sum_{n=1}^{\infty} \frac{q_n}{y^n}, \quad \text{as } u \to 1 \tag{8.3.4}$$

where the coefficients q_n are given by (6.2.6). Let $W\left(z\right) = \sum_{n=0}^{\infty} q_n z^n$, so that $Q\left(u\right) \sim y + W\left(1/y\right)$. Let us now construct the Padé approximant $R_{4,4}\left(z\right)$ of $W\left(z\right)$. We are interested in comparing the truncated asymptotic expansion

$$Q(u) \sim y + \sum_{n=1}^{K} \frac{q_n}{y^n}$$

with the approximation

$$Q(u) \sim y + R_{4,4}\left(\frac{1}{y}\right).$$

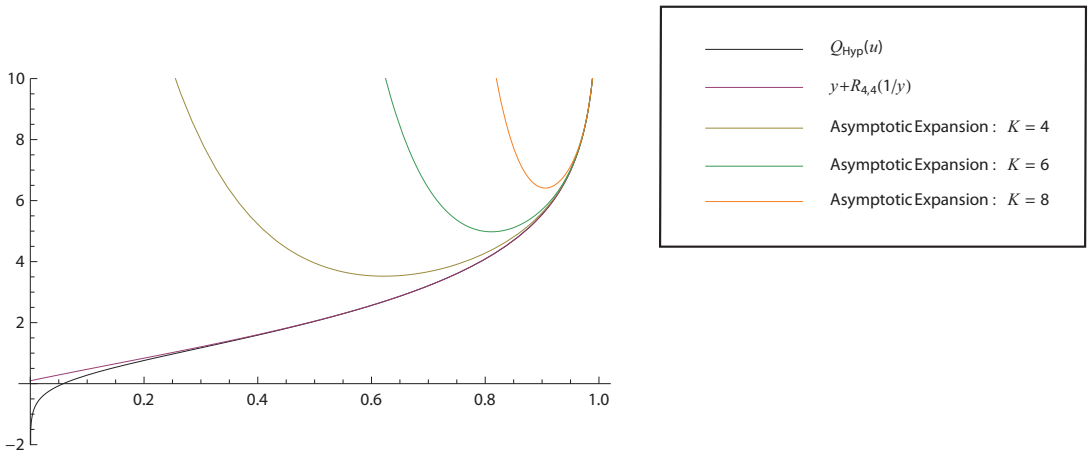


Figure 8.3.2: A Padé approximant used to sum a divergent series.

The Mathematica code to do so is given in figure C.0.4 on page 148. Figure 8.3.2 shows the graph of the asymptotic expansion 8.3.4 truncated after the $K = 4$, $K = 6$ and $K = 8$ terms; clearly the series is divergent. However as the graph of $y + R_{4,4}(1/y)$ suggests we may successfully sum this series by constructing Padé approximants. Such approximations are very useful in the deep tail regions.

There is a deep connection between Padé approximants and various types of continued fractions. In fact under some mild conditions, the convergents of a regular C-fraction f_n correspond the staircase entries $R_{0,0}, R_{1,0}, R_{1,1}, R_{2,1}, R_{2,2}, \ldots$ in the Padé table. The reader is referred to (Cuyt et al., 2008) for a full account.

8.3.3 Error and Partition Management

Padé approximants and regular C-fractions are very easy to construct given a Taylor series expansion. Since the parameters of the probability distribution govern the shape of the quantile function and the radius of convergence of the series, we cannot expect a fixed partition to work for all parameter values. Therefore we devise a simple technique to fix the partition at run time given a set of parameter values and manage the error on each of these parts.

Like Taylor polynomials, Padé approximants provide exceptionally good approximations near the point of expansion but the error deteriorates as we move away from this point. Hence to test whether $R_{m,n}(u)$ is a valid approximation we need only to verify that the accuracy requirements are met at the end points of the region of approximation. More specifically suppose we want to approximate $Q(u)$ (which is increasing and continuous) on $[u_1, u_2]$ to within a prescribed accuracy ϵ. Assume further that by utilizing a bracketing algorithm (like the bisection method for example) we have obtained intervals which enclose the exact value of the quantile function at the points u_1 and u_2, that is $Q(u_1) \in [x_1, y_1]$ and $Q(u_2) \in [x_2, y_2]$ with $|x_1 - y_1| < \epsilon$ and $|x_2 - y_2| < \epsilon$. If we can find an approximation $R_{m,n}(u)$ such

that $R_{n,m}(u_1) \in [x_1, y_1]$, $R_{n,m}(u_2) \in [x_2, y_2]$ and $R_{m.n}(u)$ is free from defects[2] in $[u_1, u_2]$ then $|R_{m.n}(u) - Q(u)| < \epsilon$ for all $u \in [u_1, u_2]$. Based on this observation one method to find a valid approximation of $Q(u)$ on $[u_1, u_2]$ would be to construct a sequence of Padé approximants, for example along the main diagonal of the Padé table, and terminating the procedure when the above criteria is satisfied. The idea behind the algorithm is then as follows,

- The order of the asymptotic expansion is chosen in advance, and a numerical search is conducted to determine τ_L and τ_R, which are typically small values $\approx 10^{-9}$. For example suppose we decide on a 10^{th} degree expansion for the left and right tails, then we must determine for what regions $(0, \tau_L)$ and $(1 - \tau_R, 1)$ these expansions are valid. These expansions provide increasingly better approximations of Q as we approach the singular points 0 or 1. Starting with a large value, say $\tau_L = \tau_R = 10^{-1}$ the algorithm determines whether the truncated expansions are valid in this range using the above technique. The values of τ_L and τ_R are incrementally reduced, say by a factor of 10 until a valid range is found.

- Next a point of expansion u_0 for the central region is determined, one may choose $u_0 = 1/2$, the zero quantile location or $u_0 = F(x_m)$ where x_m is the mode of the distribution or another appropriate choice. On this region the Taylor expansion of Q at u_0 serves as a useful approximation. Hence we construct a sequence of approximants along the main diagonal of the Padé table. The sequence is terminated when an approximant $R_{n,n}$ is found which satisfies the required accuracy goals. We will discuss how to choose the points u_1 and u_2 below.

- On the left and right regions the left and right solutions of the recycling equation (6.0.8) denoted A_L and A_R respectively, serve as good approximations, this is precisely what they were designed to do. Again a series acceleration method is applied to the Taylor polynomials of A_L and A_R. The points at which we impose the initial conditions are critical. For example we may choose $u_0 = F_B(z_0)$ where $z_0 = (Q_B(\tau_L) + Q_B(u_m/2))/2$ and $z_0 = (Q_B(1 - \tau_R) + Q_B(u_m/2))/2$ for the left and right problems respectively. But by varying these initial conditions we alter the range of distribution parameters for which the algorithm is valid. Hence the algorithm proceeds by trial and error to determine at which point the initial conditions are imposed so that the error conditions are satisfied.

[2] A Padé approximant may exhibit spurious poles not present in the original function $f(z)$. A robust algorithm should check that none of the real roots of the denominator polynomial appearing in the Padé approximant lie in the interval of approximation. Such poles are called defects, see Baker and Graves-Morris (1996) for details.

The points u_1 and u_2 enclosing the central region are determined by an estimate $\tilde{r}$ of the radius of convergence r for the series. In particular we set

$$u_1 = u_0 - \bar{r}$$

and

$$u_2 = u_0 + \bar{r}$$

where

$$\bar{r} = \min\left\{\tilde{r}, |u_0 - 0.1|, |u_0 - 0.9|\right\}.$$

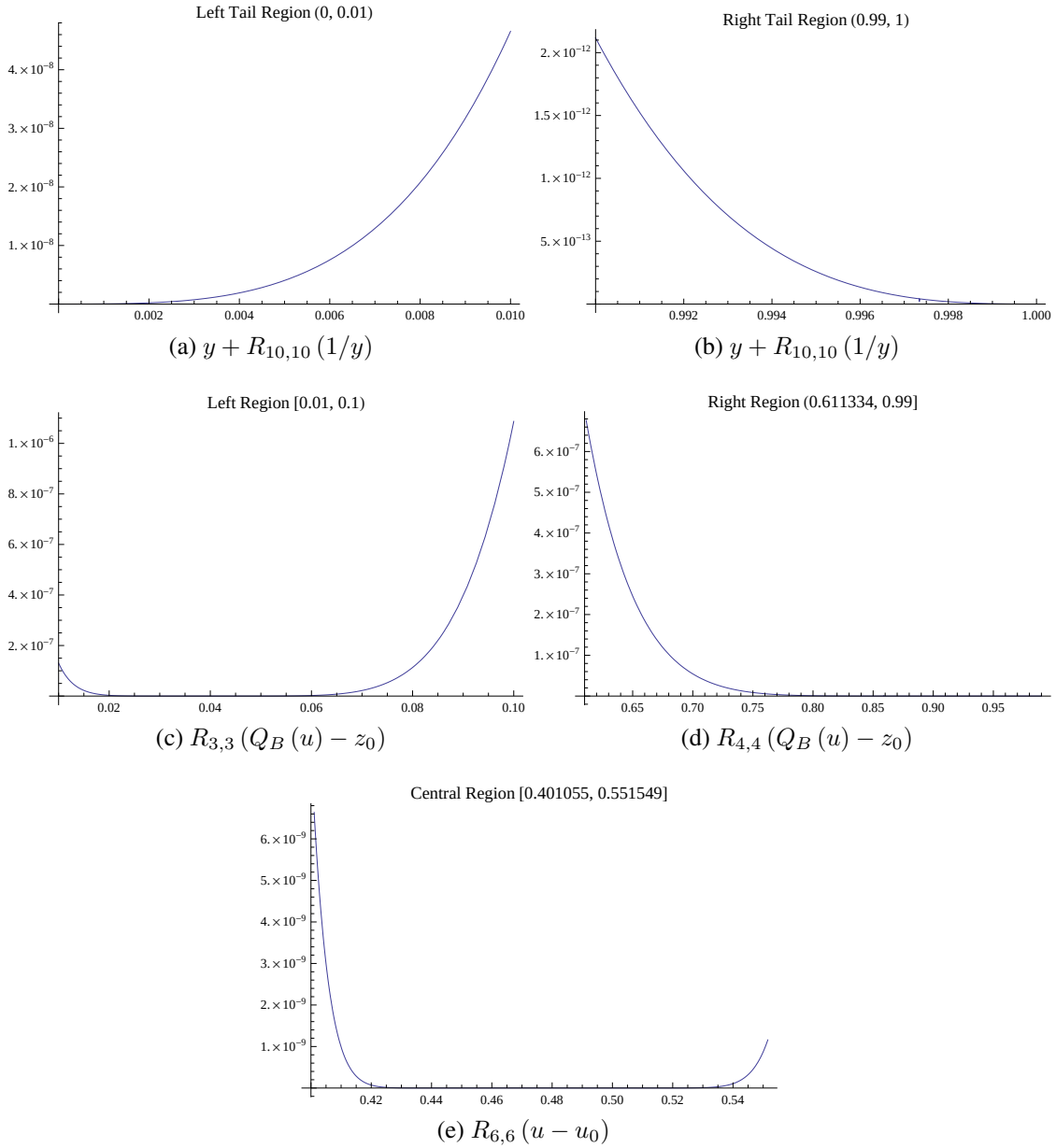
For simplicity our approximation of $\tilde{r}$ will be based on the Cauchy's root test

$$r = \frac{1}{\lim\limits_{n\to\infty} |q_n|^{1/n}}.$$

However note that the problem of estimating the radius of convergence is a rather old one and many more advanced techniques have been developed to determine $\tilde{r}$. For example Chang and Corliss (1980) form a small system of equations based on three, four and five terms of the series (8.5.5) to determine $\tilde{r}$. However we are not overly concerned if $\tilde{r}$ over estimates the radius of convergence of (8.5.5) since this will be compensated by the fact that applying an appropriate summation technique will provide analytic continuation of (8.5.5). Thus for this iteration of the algorithm we will be content with a simple estimate of $\tilde{r}$ based on Cauchy's root test. A complete Mathematica implementation of this algorithm for the hyperbolic quantile and a `C++` implementation for the variance gamma quantile are given in an online code repository (Munir, 2012). We apply the algorithm to our test case $Q_{\text{Hyp}}(u; 2, 3/2, 1, 0)$ with a specified accuracy of $\epsilon = 2 \times 10^{-6}$, the resulting error plots are given in figure 8.3.3.

The resulting error plots are by no means optimal in the Chebyshev sense, however this approximation, lets call it `QApprox[u]` has its benefits. First it is very quick to setup `QApprox[u]` , in this case the setup time is just 0.156 seconds (about the same time as it takes to evaluate $Q(u)$ six times using the root finder of section 8.1 on our test machine). Second `QApprox[u]` is very accurate in the tails, since it exploits the asymptotic expansions of $Q(u)$ developed in chapter 6. Finally by using the output of `QApprox[u]` as the initial estimate for a Newton Raphson based root finder can increase the speed of convergence of the scheme. Consider the following code, two Newton Raphson based root finders are defined `QRFNaive[u]` and `QRFPade[u]`.

```
F[x_] := CDF[HyperbolicDistribution[2, 3/2, 1, 0], x]
G[x_, u_] := F[x] - u;
```



(a) $y + R_{10,10}\left(1/y\right)$

(b) $y + R_{10,10}\left(1/y\right)$

(c) $R_{3,3}\left(Q_B\left(u\right) - z_0\right)$

(d) $R_{4,4}\left(Q_B\left(u\right) - z_0\right)$

(e) $R_{6,6}\left(u - u_0\right)$

Figure 8.3.3: u-error $\left|u - F\left(Q_A\left(u\right)\right)\right|$ plots of Padé Approximants.

```
(*FindRoot uses Newton Raphson by default*)
QRFNaive[u_?NumericQ] := x /. FindRoot[G[x, u], {x, 0}]
QRFPade[u_?NumericQ] := x /. FindRoot[G[x, u], {x,
   QApprox[u]}]
```

`QRFNaive[u]` uses the location parameter $\mu = 0$ as an initial estimate and `QRFPade[u]` uses the output of `QApprox[u]`. The following test code generates 10^3 Hyp $(2, 3/2, 1, 0)$ random variates using the inversion method based on both of these root finders.

```
In[1]:= n = 10^3; uniforms = RandomReal[1, n];
        res1 = Timing[ Map[QRF, uniforms]; ]
        res2 = Timing[Map[QRFPade, uniforms];]
        res1[[1]]/res2[[1]]

Out[1]= {18.689, Null}
Out[2]= {5.99, Null}
Out[3]= 3.12003
```

From the output we see that `QRFPade[u]` is over three times faster than `QRFNaive[u]`.

8.4 Sequence Transforms

Often, it happens that the series $\sum_{n=0}^{\infty} a_n$ converges too slowly to be numerically useful. Fortunately there are several tricks for accelerating the rate of convergence of the series, or equivalently the sequence of partial sums

$$s_n = \sum_{k=0}^{n} a_k.$$

If $\lim_{n\to\infty} s_n = s$ exists, then we may write down the remainder after truncating the series at the n^{th} term as

$$r_n = s - s_n.$$

The idea behind sequence transformation methods is to transform the sequence $\{s_n\}_{n=0}^{\infty}$ into a new sequence $\{s'_n\}_{n=0}^{\infty}$ such that the rate of convergence is improved. The transformed sequence must of course still converge to the same limit $\lim_{n\to\infty} s'_n = s$. Convergence will then be improved if the sequence of transformed remainders $\{r'_n\}_{n=0}^{\infty}$, where $r'_n = s - s'_n$ vanish more rapidly than the original remainders $\{r_n\}_{n=0}^{\infty}$, that is

$$\lim_{n\to\infty}\frac{r'_n}{r_n}=\lim_{n\to\infty}\frac{s-s'_n}{s-s_n}=0.$$

Note that the sequence $\{s_n\}_{n=0}^{\infty}$ need not be convergent, for a general account of the theory see the excellent article by Weniger (2001) or the monographs by Brezinski and Zaglia (1991) and Sidi (2002). In some cases sequence transformation methods may be used to transform a divergent series into a meaningful convergent sequence, for example, to "sum" a power series beyond its radius of convergence (analytic continuation), or to extend the useful region of an asymptotic series.

There is a great variety of sequence transformations discussed in the literature, we briefly discuss Levin's sequence transform (Levin, 1972) next, which is possibly the best single general purpose sequence acceleration method currently known. It is defined as

$$\mathcal{L}_k^{(n)}\left(s_n,\omega_n\right)=\frac{\sum_{j=0}^{k}(-1)^j\binom{k}{j}\frac{(\xi+n+j)^{k-1}}{(\xi+n+k)^{k-1}}\frac{s_{n+j}}{\omega_{n+j}}}{\sum_{j=0}^{k}(-1)^j\binom{k}{j}\frac{(\xi+n+j)^{k-1}}{(\xi+n+k)^{k-1}}\frac{1}{\omega_{n+j}}} \tag{8.4.1}$$

where the quantities ω_n are remainder estimates and ξ is usually chosen as a $1/2$ or 1. For example we may choose $\omega_n=s_{n+1}-s_n=a_{n+1}$ resulting in what is known as Levin's d-transform or we may choose $\omega_n=(1+n)\,a_n$ which yields Levin's u-transform.

From an implementation point of view Levin's transform is not computed as written in (8.4.1). A more stable and efficient approach is to compute the numerator and denominator appearing in (8.4.1) from the following three term recurrence relation

$$D_{k+1}^n\left(\xi\right)=D_k^{n+1}\left(\xi\right)-\frac{(\xi+n)\left(\xi+n+k\right)^{k-1}}{\left(\xi+n+k+1\right)^k}D_k^n\left(\xi\right).$$

The numerator in (8.4.1) may be computed by using the starting values

$$D_0^n\left(\xi\right)=\frac{s_n}{\omega_n}$$

and we obtain the denominator by using the starting values

$$D_0^n\left(\xi\right)=\frac{1}{\omega_n}.$$

For efficient Fortran and C++ implementations of these two transforms see Weniger (2001) and Press et al. (2007) respectively. We provide a Mathematica implementation in appendix C.0.6. Note that Levin's transform acts on the values $s_n,\ldots,s_{n+k}$, thus to compute $\mathcal{L}_k^{(n)}\left(s_n,\omega_n\right)$ requires knowledge of the first $n+k+1$ terms of the series $a_0,\ldots,a_{n+k+1}$.

When Levin's transform is applied to a power series the result is a rational function[3]. These rational functions share many of the properties of Padé approximants, by repeating the experiments in examples (33) and (34) with the Padé approximants replaced by Levin's u-transforms we observe similar benefits. Another example application of Levin's u-transform is given in the next section.

8.5 Chebyshev Series

We are interested in the problem of constructing Chebyshev series expansions of the functions $A_L(z(u))$, $Q(u)$ and $A_R(z(u))$ restricted to the sets $[\tau_L, u_1]$, $[u_1, u_2]$ and $[u_2, 1-\tau_R]$ respectively. In each case we introduce a linear change of variable x which maps the restricted domain onto the set $[-1, 1]$. To ease notation let $g(x) := Q|_{[u_1,u_2]}(x)$; in this case $x|[u_1, u_2] \to [-1, 1]$ is defined by

$$x(u) = \frac{u - \frac{1}{2}(u_2 + u_1)}{\frac{1}{2}(u_2 - u_1)}. \tag{8.5.1}$$

Following from the properties of Q, the function g is continuous and of bounded total variation and thus admits the expansion

$$g(x) = \frac{\tilde{g}_0}{2} + \sum_{k=0}^{\infty} \tilde{g}_k T_k(x), \quad x \in [-1, 1] \tag{8.5.2}$$

where $T_k(x)$ are the Chebyshev polynomials of the first kind and the coefficients c_k are defined by

$$\tilde{g}_k = \frac{2}{\pi} \int_{-1}^{1} g(x) \frac{T_k(x)}{\sqrt{1-x^2}} dx. \tag{8.5.3}$$

Since the quantile functions we have considered in this report are assumed to be infinitely differentiable, elementary Fourier theory tells us that the error made in truncating the series (8.5.2) after K terms goes to zero more rapidly than any finite power of $1/K$ as $K \to \infty$, see Gottlieb and Orszag (1977, § 3). Such rapid decrease of the remainder motivates us to seek efficient methods to evaluate the integral in (8.5.3). We discuss two possible approaches in the following two subsections.

8.5.1 Discrete Fourier Transform

The coefficients $\tilde{g}_k$ can rarely be evaluated analytically so the the usual process is to write (8.5.3) as

$$\tilde{g}_k = \frac{2}{\pi} \int_0^{\pi} g(\cos\theta) \cos k\theta d\theta,$$

[3]The numerator and denominator coefficients may be computed directly, see Roy et al. (1996).

by making the change of variable $x = \cos\theta$. Now by applying the trapezoidal rule with n equal sub-intervals we obtain

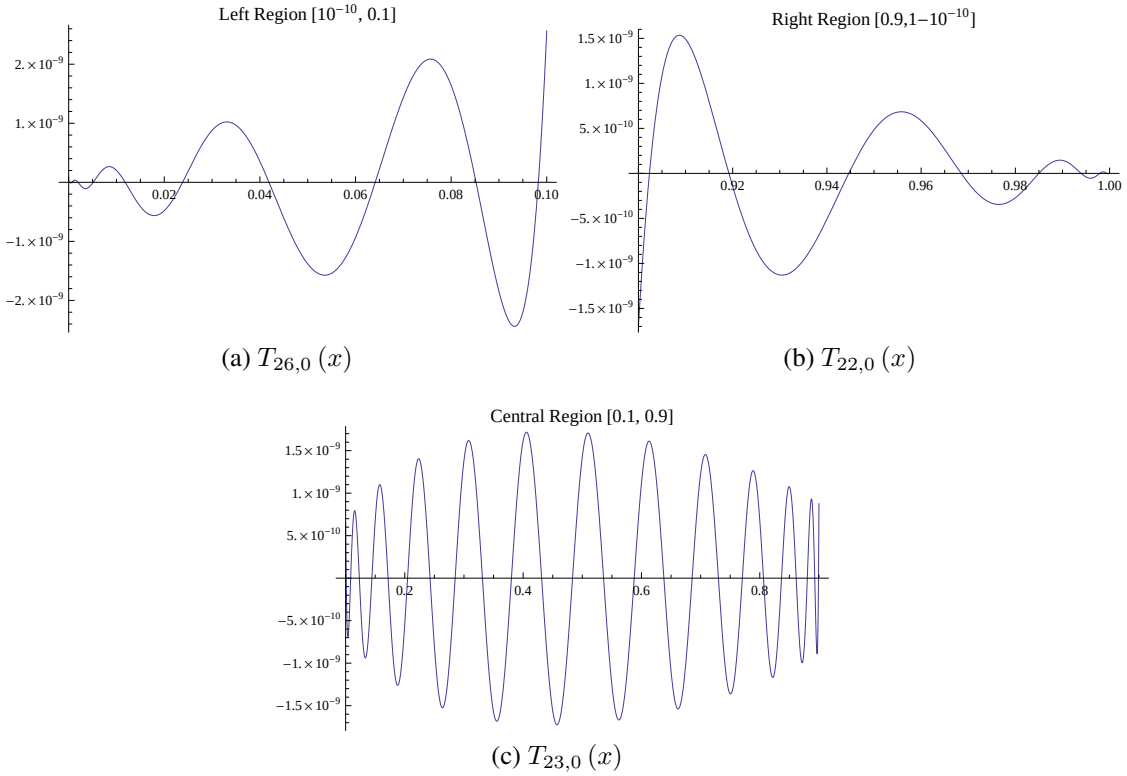
$$\tilde{g}_k \approx \frac{2}{\pi} \sum_{j=1}^{n}{}'' \, g\left(\cos\frac{\pi j}{n}\right) \cos\frac{\pi k j}{n} \tag{8.5.4}$$

where $\sum''$ denotes a summation whose first and last terms are halved. This sum is a discrete cosine transform, so a variant of the fast Fourier transform (Hamming, 1973, § 29.3) may be applied to compute this sum efficiently. We give a Mathematica implementation of the algorithm used to approximate the hyperbolic quantile $Q_{\text{Hyp}}(u; 2, 3/2, 1, 0)$ on the domain $[u_1, u_2] = [0.1, 0.9]$ below.

```
(*Define end points*)
u1 = 0.1; u2 = 0.9;

(*Change of variable maps u range onto [-1,1]*)
xU[u_] := (u - (1/2)*(u2 + u1))/((1/2)*(u2 - u1));
uX[x_] := ((u2 + u1) + (u2 - u1)*x)/2;

g[\[Theta]_] := QRF[uX[Cos[\[Theta]]]]

(*Apply FFT*)
nN = 2^7 - 1;
xData = Table[N[g[(Pi*j)/nN]], {j, 0, nN}];
ccFFT = FourierDCT[xData, 1]*Sqrt[2/nN]; cFFT[n_] :=
   ccFFT[[n + 1]];

(*Truncated Chebyshev Series*)
truncCS[x_, n_] :=   cFFT[0]/2 + Sum[cFFT[k]*ChebyshevT[k
   , x], {k, 1, n}]

Plot[{u - CDF[HyperbolicDistribution[2, 3/2, 1, 0],
   truncCS[xU[u], 23]]}, {u, u1, u2}]
```

The resulting error plot is given in figure 8.5.1c. Note that a rough estimate of the error made by truncating 8.5.2 is given by the magnitude of the first neglected term, so by monitoring this term the algorithm may be used to construct approximations at runtime for a specified accuracy ϵ.

The larger we choose the interval $[u_1, u_2]$ the slower the rate of convergence of the resulting series. Therefore on the intervals $[\tau_L, u_1]$ and $[u_2, 1 - \tau_R]$ we advise use



(a) $T_{26,0}(x)$ (b) $T_{22,0}(x)$

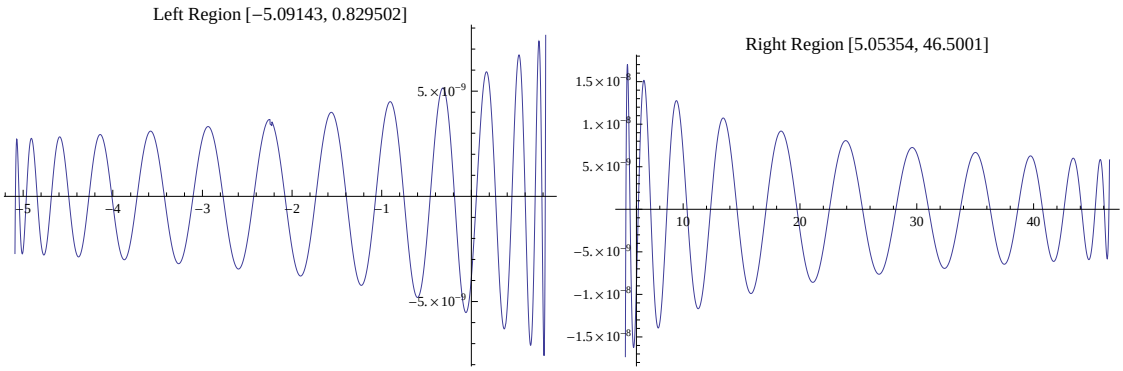
(c) $T_{23,0}(x)$

Figure 8.5.1: u-error $(u - F(Q_A(u)))$ plots obtained from truncated Chebyshev series (FFT)

of the functions $A_L(z)$ and $A_R(z)$ as defined as in section 6.2. Let us choose $\tau_L = \tau_R = 10^{-10}$ as the cutoff points for the left and right regions. We want to compute the Chebyshev expansion of the function $A_R(z)$ on the domain $[z_1, z_2]$ where $z_1 = Q_B(u_2)$ and $z_2 = Q_B(1 - \tau_R)$. The procedure is much the same as before. The complete Mathematica code to compute the Chebyshev expansion of $A_R|_{[z_1,z_2]}(z)$ is given in figure C.0.7 on page 151. Minor modifications are required to compute the Chebyshev expansion of $A_L(z)$. The error plots for the left and right regions are given in figure 8.5.1 (a) and (b) respectively. Clearly the error does not oscillate as one would expect from a truncated Chebyshev series in these regions. Here we have plotted the errors made in approximating the quantile function $Q_T(u) = A(Q_B(u))$ via $T_{n,0}(Q_B(u))$. Figure 8.5.2 instead directly plots the errors made in approximating the function $A(z)$ via $T_{n,0}(z)$ in the left and right regions; the error oscillates as expected, however as observed in figure 8.5.1 making the substitution $z = Q_B(u)$ dampens the oscillations in the far tail regions.

8.5.2 From Taylor Series

There is an alternative approach to approximate the Chebyshev coefficients provided by Thacher (1964) directly from the Taylor series coefficients. Suppose the Taylor series expansion of $Q(u)$ about $u_0 = (u_2 + u_1)/2$ and g as defined above are be given by



(a) $(A_L(z) - T_{26,0}(z))$ (b) $(A_R(z) - T_{22,0}(z))$

Figure 8.5.2: Direct plot of the absolute error made in the truncated Chebyshev series

$$Q(u) = \sum_{n=0}^{\infty} q_n (u - u_0)^n \tag{8.5.5}$$

and

$$g(x) = \sum_{k=0}^{\infty} g_k x^k \tag{8.5.6}$$

respectively. Then by inverting (8.5.1) and substituting into (8.5.5) we see that the coefficients appearing in (8.5.6) may be written as

$$g_k = q_k \left(\frac{u_2 - u_1}{2} \right)^k.$$

Now substituting the following relationship between the monomials x^k and the Chebyshev polynomials (Thacher, 1964, eq. 4),

$$x^k = T_0(x) + \sum_{j=1}^{k} \theta_{k,j} T_j(x)$$

where

$$\theta_{j,k} = \begin{cases} 2^{1-j} \binom{j}{\frac{j-k}{2}} & j-k \text{ even} \\ 0 & j-k \text{ odd} \end{cases},$$

into (8.5.6), one may express the Chebyshev coefficients $\tilde{g}_k$ in terms of the Taylor coefficients g_k as follows,

$$\tilde{g}_k = \sum_{j=k}^{\infty} g_j \theta_{j,k}. \tag{8.5.7}$$

Thacher's approach was simple, he observed that applying Shank's transform to (8.5.7) one may approximate the Chebyshev coefficients even for slowly convergent series. In our experimentation we found Levin's u-transform to also be affective, which of course was not discovered at the time Thacher wrote his paper. Levin's u transform has been reported in many instances to outperform Shank's transform (Smith and Ford, 1982). Thus given the knowledge of the Taylor coefficients g_k we now have a method to approximate the Chebyshev coefficients of $Q(u)$ without having to resort to the computationally expensive task of evaluating (8.5.3) through numerical quadrature. In a similar fashion we can construct Chebyshev series expansions for the restrictions $A_L|_{[\tau_L,u_1]}$ and $A_R|_{[u_2,1-\tau_R]}$. We provide the code for the central region below, the complete Mathematica code to implement this procedure for the all three regions is given in an online code repository Munir (2012).

```
(*Define end points*)
u1 = 0.1; u2 = 0.9;

(*Change of variable maps u range onto [-1,1]*)
xU[u_] := (u - (1/2)*(u2 + u1))/((1/2)*(u2 - u1));
uX[x_] := ((u2 + u1) + (u2 - u1)*x)/2;

(*define Taylor series in x*)
rC = (1/2)*(u2 - u1);
g[n_] := q[n]*rC^n;  (*q[n] Taylor series coefficients
   about u0=1/2*)
G[n_, x_] := Sum[g[k]*x^k, {k, 0, n}]
\[CapitalTheta][k_, n_] :=    Piecewise[{{2^(1 - k)*
   Binomial[k, (k - n)/2], EvenQ[k - n]},           {0,
   OddQ[k - n]}}];
gTpSum1[n_, m_] := Sum[g[k]*\[CapitalTheta][k, n], {k, n,
    n + m}] m = 28;

(*generate Chebyshev Series*)
Table[s[m_] := gTpSum1[n, 2*m];         ccShank[n] =
   SequenceLimit[Table[s[k], {k, 0, m}]], {n, 0, 25}]
cc[n_] := ccShank[n]
PSum[x_, n_] := cc[0]/2 + Sum[cc[k]*ChebyshevT[k, x], {k,
    1, n}]

(*plot the u-error*)
```

```
Plot[{u -     CDF[HyperbolicDistribution[\[Alpha]1, \[
   Beta]1, 1, 0],      PSum[xU[u], 23]]},     {u, u1, u2
   }]
```

8.5.3 Comparison

Table 8.1 lists the setup times for both methods. The total setup time to construct the Chebyshev polynomials via FFT is 14.992 seconds. On the other hand Thacher's method takes 2.636 seconds which is a significant improvement. Note however the sluggishness of the FFT algorithm is due to the slow root finder used to evaluate $Q(u)$ at a 128 points. The root finder employed here is that of section 8.1 based on the Newton Raphson method working in double precision. The root finder uses the location parameter $\mu = 0$ as an initial estimate of the root. The third row of table 8.1 corresponds to the setup times when the FFT algorithm is combined with the improved root finder of section 8.3.3 (which requires a setup time of 0.156 seconds). The total setup time of this method is then 2.652 seconds, similar to Thacher's method.

There is one final approach that is worth mentioning due to its simplicity and effectiveness. To approximate the Chebyshev coefficients the discrete cosine transform is applied to the data points $x_0, \ldots, x_n$, where

$$x_i := Q(p_i),\ p_i := u\left(\cos\frac{\pi i}{n}\right), \quad j = 0, \ldots, n$$

and $u(x)$ is the inverse of (8.5.1)

$$u(x) = \frac{(u_2 + u_1) + (u_2 - u_1)x}{2}.$$

From the properties of the Q, it follows that $x_0, \ldots, x_n$ is a decreasing list of numbers $x_0 > x_1 > \cdots > x_n$ which are in a sense "close". Thus to compute x_1 by a root finding scheme we may use x_0 as an initial estimate, to compute x_2 we may use x_1 as an initial estimate and so on. Updating the initial estimates in this way has a big impact on the computational effort required. The last row in table 8.1 lists the setup times using this approach, the total setup time now becomes 5.585 seconds. The Mathematica code below demonstrates, this technique is very simple to implement.

```
F[x_] := N[   CDF[HyperbolicDistribution[2, 3/2, 1, 0], x
   ]]
G[x_, u_] := F[x] - u;
(*FindRoot uses Newton Raphson by default*)
guess = 0;
QRF[u_] := Module[{},
        (*find root and update the initial estimate*)
```

	Region (Degree of Approximant)			
	Left (26)	Centre (23)	Right (22)	Total Setup Time
FFT (Naive Root Finder)	4.743	2.418	7.831	14.992
Thacher's Method (Taylor coefficients required)	2.075 (116)	0.249 (51)	0.312 (52)	2.636
FFT (Improved Root Finder) Root Finder Setup: 0.156	0.733	0.827	0.936	2.652
FFT (Specialized Root Finder)	1.56	1.467	2.558	5.585

Table 8.1: Setup up times (in seconds) required to compute the truncated Chebyshev series.

```
guess =  x /. FindRoot[G[x, u], {x, guess}];
guess ]
```

Of course updating the estimate in this way should not be used for general purpose applications; for a random point $u_i \in (0,1)$ unless we have other information available perhaps the location parameter, the mean or the mode serve as the reasonable initial estimates.

8.6 Chebyshev-Padé Approximants

Combining FFT with an improved root finder leads to better setup times. Perhaps the greatest drawback of this approach is that it leads to approximants of high degree leading to slow execution times. To reduce the degree of the approximant we consider constructing Chebyshev-Padé approximants of the function $g(x)$ as defined above. Chebyshev-Padé approximants are rational functions of the form

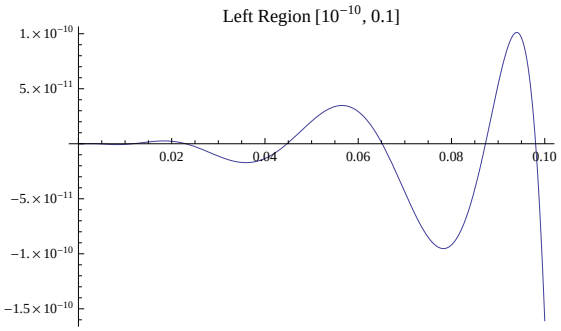
$$T_{m,n}(x) := \frac{a_0 T_0(x) + \cdots + a_m T_m(x)}{b_0 T_0(x) + \cdots + b_n T_n(x)}, \tag{8.6.1}$$

which have a formal Chebyshev series expansion in agreement with (8.5.2) up to and including the term $\tilde{g}_{m+n} T_{m+n}(x)$. Given the knowledge of the first $n+m+1$ Chebyshev coefficients $\tilde{g}_0, \ldots, \tilde{g}_{n+m}$, an efficient way to construct $T_{m,n}(x)$ is to employ the algorithm of Sidi (1975), in which the coefficients appearing in (8.6.1) are computed recursively. The coefficients appearing in the denominator are given by

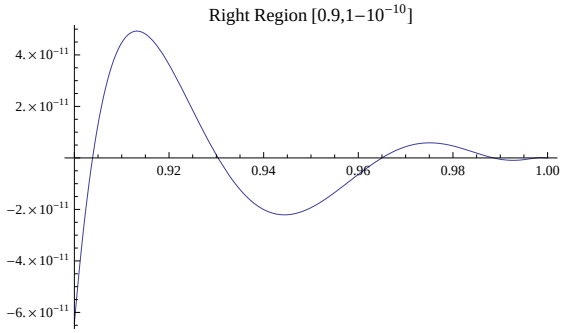
$$b_s = \xi \sum_{i=0}^{n-s} \gamma_i^{(m,n)} \gamma_{s+1}^{(m,n)}, \quad s = 1, 2, \ldots, n$$

where

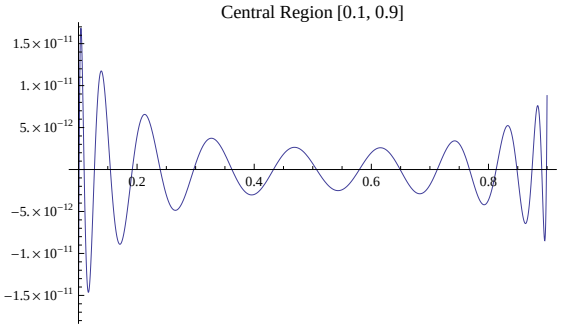
$$\xi = \left(\frac{1}{2} \sum \left(\gamma_i^{(m,n)} \right)^2 \right)^{-1},$$



(a) $T_{9,9}\left(x\right)$



(b) $T_{7,7}\left(x\right)$



(c) $T_{9,9}\left(x\right)$

Figure 8.6.1: u-error $\left(u - F\left(Q_A\left(u\right)\right)\right)$ plots obtained from the Chebyshev-Padé approximants.

$$\gamma_s^{(m,n)} = \begin{cases} 1 & s = 0 \\ 0 & s = -1 \text{ or } s = n+1 \\ \gamma_s^{(m+1,n-1)} + \omega^{(m,n)} \gamma_{s-1}^{(m,n-1)} & 1 \leq s \leq n \end{cases}$$

and

$$\omega^{(m,n)} = -\frac{\sum_{s=0}^{n-1} \gamma_s^{(m+1,n-1)} \tilde{g}_{|m+1-s|}}{\sum_{s=0}^{n-1} \gamma_s^{(m,n-1)} \tilde{g}_{|m-s|}}.$$

The coefficients appearing in the numerator are given by

$$a_r = \frac{1}{2} \sum_{s=0}^{n} b_s \left(\tilde{g}_{r+s} + \tilde{g}_{|r-s|}\right), \quad s = 1, 2, \ldots, n.$$

We provide a Mathematica (Wolfram-Research, 2010) implementation of Sidi's method in figure C.0.8 on page 152. Applying this scheme to the same test case as above we obtain the error plots given in figure (8.6.1).

The benefit of Chebyshev Padé approximants over truncated Chebyshev series is that for lower degree approximants we obtain much better approximations (contrast figure 8.6.1 with figure 8.5.1) hence we can expect faster execution times, but since the

approximants are of lower degree fewer Chebyshev coefficients need to be computed so the set up time may also be improved. However if we are to construct Chebyshev Padé approximants on the fly (i.e. at run time) we need a means to control the error. According to Ralston and Rabinowitz (2001, p. 304) often a good estimate of the error in $T_{m,n}(x)$ is given by $|h_{N+1}T_{m+n+1}(x)|$ where $N = m + n$ and

$$h_{N+1} = \frac{1}{2}\sum_{i=0}^{n} b_i \left(\tilde{g}_{N+1-i} + \tilde{g}_{N+1+i}\right).$$

8.7 Osculatory Rational Interpolation

Consider a partition of the form $\tau_L < u_m < 1 - \tau_R$ where u_m is for example chosen as $u_m = F(x_m)$ and x_m is the mode. Choose the points $p_0, \ldots, p_K \in [\tau_L, u_m]$, and let $s_0, \ldots, s_K \in \mathbb{N}$. Suppose now we want to build a rational approximant $R_{m,n}$ valid on $[\tau_L, u_m]$ which satisfies the following conditions,

$$A_L^{(j)}(z_k) = R_{m,n}^{(j)}(z_k), \quad 0 \le k \le K,\ 0 \le j \le s_k - 1, \tag{8.7.1}$$

where $z_k = Q_B(p_k)$. That is we would like to construct a rational function $R_{m,n}$ such that at each of the K interpolation points z_k the derivatives of the function A_L and $R_{m,n}$ agree up to order $s_k - 1$. Such a problem is called an osculatory rational interpolation problem. The interpolation data on the right hand side of (8.7.1) may be generated by solving the recycling equation (6.0.8) for A_L with initial conditions imposed at the points $p_0, \ldots, p_K$. Provided a solution to the problem exists it may be solved efficiently through the generalized Q.D. algorithm (Graves-Morris, 1980), which yields a continued fraction of the form

$$\begin{aligned} A_L(z) \;=\; & \frac{c_0}{1} - \frac{q_1^0(z - z_0')}{1} - \frac{e_1^0(z - z_1')}{1} - \frac{q_2^0(z - z_2')}{1} \\ & - \frac{e_2^0(z - z_3')}{1} - \cdots \end{aligned} \tag{8.7.2}$$

The convergents $A_L^{(n)}(z)$ of (8.7.2) are rational functions called multipoint Padé approximants satisfying (8.7.1). The input to the algorithm is the interpolation data

$$(z_0, A_L(z_0)), \ldots, \left(z_0, A_L^{(s_0-1)}(z_0)\right), (z_1, A_L(z_1)), \ldots, \left(z_1, A_L^{(s_1-1)}(z_1)\right), \qquad \ldots, (z_1, A_L(z_1)), \ldots$$

Denote by $z_0', z_1', \ldots, z_N'$ the corresponding set of confluent interpolation points, where $N = s_0 + s_1 + \cdots + s_K$. The first s_0 elements of this list are equal to z_0 the next s_1 elements are equal to z_1 and so on. Once this data is generated we may use divided differences to construct Newton's interpolating polynomial (Conte and Boor, 1980)

$$P(z) = c_0 + \sum_{k=1}^{N} c_k \prod_{j=0}^{k-1} (z - z'_j).$$

The Q.D. algorithm requires the coefficients $c_0, \ldots, c_N$ to compute the partial numerators $q_1^0, e_1^0, q_2^0 \ldots$ recursively as follows (Baker and Graves-Morris, 1996, p. 347):

Initialization: For $J = 0, 1, \ldots, N-1$, define

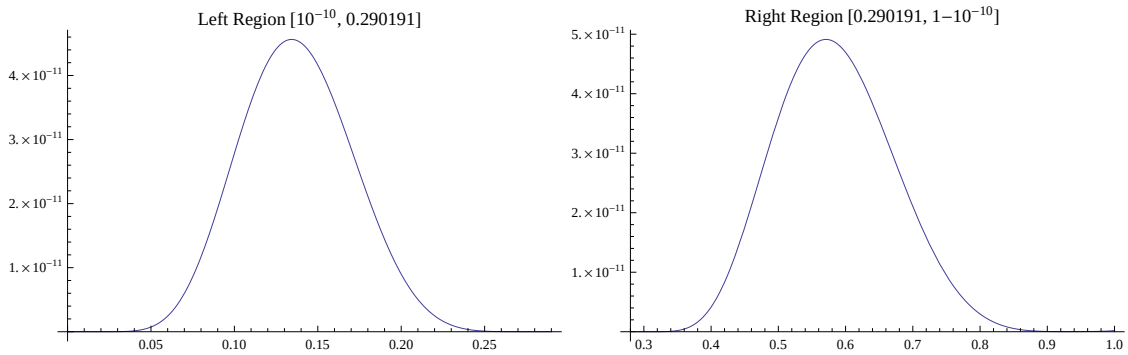
$$\begin{aligned} Z_1^J &= z'_{J+1} - z'_J \\ e_0^{J+1} &= 0 \\ q_1^J &= \left[Z_1^J + \frac{c_J}{c_{J+1}} \right]^{-1} \\ e_1^J &= -q_1^J - q_1^{J+1}(q_1^J Z_1^J - 1) \end{aligned}$$

Recursion: For $J = 0, 1, 2, \ldots$ and $i = 2, 3, \ldots$ we construct all well defined quantities q_i^J, e_i^J recursively from the formulas

$$\begin{aligned} Z_i^J &= z'_{J+2i-1} - z'_{J+2i-2} \\ q_i^J &= \left[Z_i^J - \frac{e_{i-1}^J}{e_{i-1}^J + q_{i-1}^J} \frac{q_{i-1}^{J+1} + e_{i-2}^{J+1}}{q_{i-1}^{J+1}} \frac{Z_i^J e_{i-1}^{J+1} - 1}{e_{i-1}^{J+1}} \right]^{-1} \\ e_i^J &= -q_i^J + \left(Z_i^J q_i^J - 1 \right) \left(e_{i-1}^{J+1} + q_i^{J+1} \right) \left(Z_i^J e_{i-1}^{J+1} - 1 \right)^{-1} \end{aligned}$$

We give a Mathematica (Wolfram-Research, 2010) implementation of this algorithm in figure C.0.9.

Similarly we may construct an approximation for A_R valid on $[u_m, 1 - \tau_R]$. We apply the method to our test case and provide the results in figure (8.7.1). The complete Mathematica code is given in an online code repository (Munir, 2012). Table 8.2 lists the interpolation data used to construct $A_L^{(22)}(z)$. Here we have solved the left recycling equation by imposing initial conditions at two points $z_0 = Q_B(u_m/100)$ and $z_1 = Q_B(u_m)$. The first 16 terms of the Taylor series solution at z_0 and the first 7 terms of the Taylor series solution at z_1 are used to generate the interpolation data. To construct $A_R^{(21)}(z)$ the initial conditions were imposed at the points $z_0 = Q_B(u_m)$ and $z_1 = Q_B(99 + u_m/100)$ and the interpolation data is generated by solving the right recycling equation. The total setup time for this algorithm was 1.56 seconds.



(a) $A_L^{(22)}(z)$. (b) $A_R^{(21)}(z)$.

Figure 8.7.1: u-error $(u - F(Q_A(u)))$ plots produced by the multipoint Padé Approximants.

$A_L^{(n)}(z_i)$	$z_0 = Q_B(u_m/100) = -0.181869$	$z_1 = Q_B(u_m) = 1.13389$
$n = 0$	-1.09983	1.13389
$n = 1$	1.12481	4.14937
$n = 2$	0.16683	14.5228
$n = 3$	0.332126	92.1766
$n = 4$	0.868283	790.862
$n = 5$	2.77638	8926.25
$n = 6$	10.366	$126122.$
$n = 7$	43.7122	
$n = 8$	202.915	
$n = 9$	1015.45	
$n = 10$	5382.01	
$n = 11$	29776.7	
$n = 12$	$170446.$	
$n = 13$	1.01446×10^6	
$n = 14$	6.52123×10^6	
$n = 15$	4.95484×10^7	

Table 8.2: Taylor series are generated about the interpolation points z_0 and z_1 to construct the interpolation data used to construct multi-point Padé approximant $A_L^{(22)}(z)$. A total of 23 $(16 + 7)$ Taylor coefficients are required.

8.8 Minimax Approximations

Let $\xi(t)$ and $\phi(t)$ be continuous and bicontinuous[4] functions respectively over the compact interval $[a, b]$, the Chebyshev problem may be posed in some generality (Cody, 1970) as the determination of a rational function $R_{m,n}(\phi(t))$ of the form

$$R_{m,n}(\phi(t)) = \frac{P_m(\phi(t))}{Q_n(\phi(t))} = \frac{a_0 + a_1\phi(t) + \cdots + a_m\phi^m(t)}{b_0 + b_1\phi(t) + \cdots + b_n\phi^n(t)}.$$

such that the maximum error in the approximation

$$r_{m,n} = \max_{t\in[a,b]} \left| \frac{R_{m,n}(\phi(t)) - \xi(t)}{g(t)} \right| \tag{8.8.1}$$

is minimized. We may consider minimizing the absolute or relative error by setting the weight function g to $g(t) = 1$ or $g(t) = \xi(t)$ respectively. The rational approximation $R^*_{m,n}(\phi(t))$ which minimizes $r_{m,n}$ is called the Chebyshev, minimax or the best fit rational approximant with respect to the weight function g. The characteristic feature of $R^*_{m,n}(\phi(t))$ is that the error curve

$$\delta(t) = \frac{R_{m,n}(\phi(t)) - \xi(t)}{g(t)}$$

oscillates sufficiently often with alternating signs and equal magnitude. Specifically $R^*_{m,n}(\phi(t))$ is characterized by the existence of $L = m + n + 2$ points $t_1 < t_2 < \cdots < t_L$ in $[a, b]$ such that[5]

$$\delta(t_i) = (-1)^{i+1} r_{m,n}. \tag{8.8.2}$$

Based on this characterization several iterative algorithms have been devised to solve Chebyshev's problem. In this section we will focus on two methods, the commonly employed second algorithm of Remez (Cody, 1970) and Maehly's indirect method (Maehly, 1963).

8.8.1 The Second Algorithm of Remez

The second algorithm of Remez is an iterative algorithm which produces a sequence of approximants $R^{(1)}_{m,n}, R^{(2)}_{m,n}, \ldots$ which converges to $R^*_{m,n}$. The k^{th} stage of the algorithm involves two key steps:

1. Given a set of points $t_1^{(k-1)}, \ldots, t_L^{(k-1)}$ the system of equations (8.8.2)

$$\delta\left(t_i^{(k-1)}\right) = (-1)^{i+1} r_{m,n}, \quad i = 1, \ldots, L$$

[4] A bicontinuous function is a continuous function with a continuous inverse.

[5] More technically the error curve $\delta(t)$ attains $L = m + n + 2 - d$ extremal points, but for most practical purposes we may assume $d = 0$ see Ralston and Rabinowitz (2001) for details.

is solved for the unknowns $a_0, \ldots, a_m$, $b_0, \ldots, b_n$ and the quantity $r_{m,n}$ from which we obtain an intermediate approximation $R_{m.n}^{(k)}(\phi(t))$.

2. The corresponding error function

$$\delta(t) = \frac{R_{m,n}^{(k)}(\phi(t)) - \xi(t)}{g(t)}$$

is examined to determine a new set of critical points $t_1^{(k)}, \ldots, t_L^{(k)}$, i.e. points at which the error curve attains a local minimum or maximum.

If all the extrema of the error curve $\left|\delta\left(t_i^{(k)}\right)\right|$, $i = 1, \ldots, L$ agree to within some prescribed accuracy after the k^{th} stage the algorithm terminates, otherwise we proceed to the $(k+1)^{\text{th}}$ iteration step, substituting the set of points $t_1^{(k)}, \ldots, t_L^{(k)}$ for $t_1^{(k-1)}, \ldots, t_L^{(k-1)}$. To start the algorithm we need an initial guess $t_1^{(0)}, \ldots, t_L^{(0)}$ of the set of critical points $t_1, \ldots, t_L$. Such an initial guess may be obtained by examining the error curve produced by a near best approximation such as a Chebyshev-Padé approximant $T_{m,n}$, see Ralston and Rabinowitz (2001) for details. Additional details of how to perform the iteration steps are given in the same reference.

First we highlight some of the difficulties involved in computing $R_{m,n}^*$ for $Q_{\text{Hyp}}(u; \alpha, \beta, \delta, \mu)$ on the central region $[u_1, u_2]$. Let $t = u$, $a = u_1$, $b = u_2$, $\phi(u) = u$ and $\xi(u) = Q_{\text{Hyp}}(u)$ then Chebyshev's problem may be stated as; the determination of the rational function $R_{m,n}^*$ which minimizes the quantity

$$r_{m,n} = \max_{u \in [u_1, u_2]} \left| \frac{R_{m,n}(u) - Q_{\text{Hyp}}(u)}{Q_{\text{Hyp}}(u)} \right|. \tag{8.8.3}$$

To solve this specialized version of Chebyshev's problem we may apply the function `MiniMaxApproximation` which is Mathematica's implementation of Remez's second algorithm. Suppose we want to compute the minimax approximation $R_{4,4}^*(u)$, the most obvious approach is to apply `MiniMaxApproximation` to Mathematica's built in hyperbolic quantile function as follows:

```
Q[u_] = InverseCDF[HyperbolicDistribution[2, 3/2, 1, 0],
   u];
MiniMaxApproximation[Q[u], {u, {0.1, 0.9}, 4, 4}]
```

However the call to `MiniMaxApproximation` fails, Mathematica complains that it does not know how to differentiate the hyperbolic quantile `Q[u]`, the function `MiniMaxApproximation` requires knowledge of the first two derivatives of `Q[u]` in order to locate the critical points of $\delta(u)$. Fortunately `MiniMaxApproximation` has an option called `Derivatives` in which the user may supply this additional information. Since $Q^{(n)}(u_0) = q_n$ is just the coefficient appearing in the Taylor series

expansion of $Q(u)$ about u_0 we may supply `MiniMaxApproximation` with the information it requires, namely q_0, q_1 and q_2. Recall that the recurrent formulas of chapter 6 allow us to compute the Taylor coefficients q_n of $Q(u)$ about an arbitrary point $u \in (0,1)$. The complete implementation details are available in an online code repository. We found this method useful for constructing minimax approximants in the central region, but `MiniMaxApproximation` reports convergence problems when we tried to construct approximants of high degree (> 4) in the left and right regions.

Fortunately there is another approach, let $t = x$, $a = u_1$, $b = u_2$, $\phi(x) = F_{\text{Hyp}}(x)$ and $\xi(x) = x$ then Chebyshev's problem may be stated as; the determination of the rational function $R^*_{m,n}$ which minimizes the quantity

$$r_{m,n} = \max_{x \in \left[Q_{\text{Hyp}}(u_1), Q_{\text{Hyp}}(u_2)\right]} \left| \frac{R_{m,n}\left(F_{\text{Hyp}}(x)\right) - x}{x} \right|. \tag{8.8.4}$$

Note that by making the change of variable $x = Q_{\text{Hyp}}(u)$, (8.8.4) is equivalent to (8.8.3), but by formulating the problem in this way avoids the need to evaluate the quantile function $Q_{\text{Hyp}}(u)$ (except at the endpoints u_1 and u_2) and hence greatly reduces the computational effort required! Put another way the generalized Chebyshev problem (8.8.1) may be interpreted as the determination of the rational function of $R_{m,n}(x)$, that gives a minimax approximation to the parametrically defined curve with x and y coordinates given by

$$(x, y) = (\phi(t), \xi(t)), \quad t \in [a, b].$$

The quantile function $Q_{\text{Hyp}}(u)$ on $[u_1, u_2]$ may be defined parametrically as follows

$$(F(x), x), \quad Q(u_1) \leq x \leq Q(u_2).$$

Mathematica provides us with the function `GeneralMiniMaxApproximation` which may be used to solve the generalized Chebyshev problem (8.8.1), again via Remez's second algorithm. In our notation the calling convention for `GeneralMiniMaxApproximation` is as follows,

$$\texttt{GeneralMiniMaxApproximation}\left[\phi, \xi, \{\texttt{t}, \{\texttt{a}, \texttt{b}\}, \texttt{m}, \texttt{n}\}, \texttt{x}\right]$$

The following Mathematica code computes the best approximation $R^*_{10,10}(u)$ for $Q_{\text{Hyp}}(2, 3/2, 1, 0)$ on $[0.1, 0.9]$. It takes roughly 93 seconds to execute on our test machine and the resulting error curve is plotted in figure 8.8.1 (c).

```
u1 = 1/10; u2 = 9/10; precision = 20;
x1 = QRF[u1];  x2 = QRF[u2];
```

```
F[x_] = CDF[    HyperbolicDistribution[\[Alpha], \[Beta],
    \[Delta], \[Mu]], x];
 gmma = GeneralMiniMaxApproximation[{F[x], x}, {x, {x1,
    x2}, 10, 10},     u, WorkingPrecision -> precision]

mmaQC[u_] = gmma[[2, 1]]
```

Next we look at computing Chebyshev approximants for $A_L(z)$ and $A_R(z)$ on the intervals $[Q_B(\tau_L), Q_B(u_1)]$ and $[Q_B(u_2), Q_B(1-\tau_R)]$ corresponding to the left and right regions respectively. On these intervals $A_L(z)$ and $A_R(z)$ may be defined parametrically as

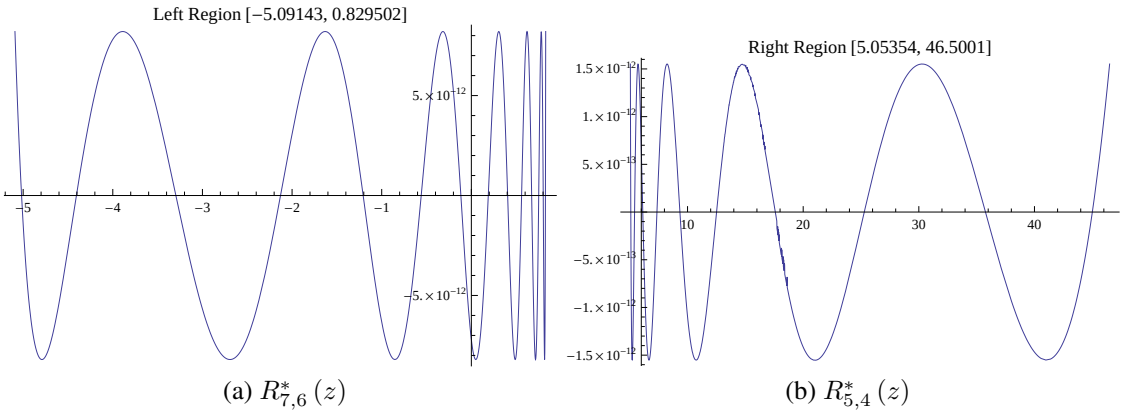
$$\left(A_L^{-1}(x), x\right), \quad Q_{\text{Hyp}}(\tau_L) \leq x \leq Q_{\text{Hyp}}(u_1)$$

and

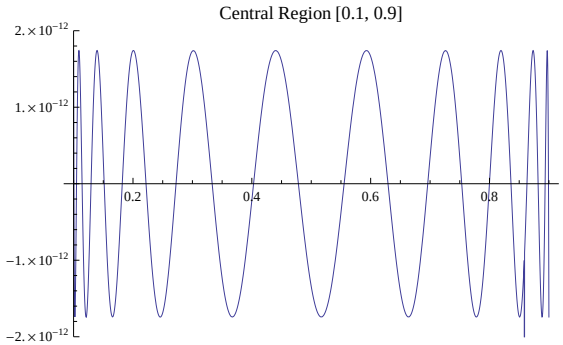
$$\left(A_R^{-1}(x), x\right), \quad Q_{\text{Hyp}}(u_2) \leq x \leq Q_{\text{Hyp}}(1-\tau_R)$$

respectively where recall from chapter 6 that $A_L^{-1}(x) = Q_B(F_{\text{Hyp}}(x))$, $A_R^{-1}(x) = Q_B(F_{\text{Hyp}}(x))$ and the base quantile $Q_B(u)$ is defined as in (6.2.8). The following Mathematica code illustrates how to compute the Chebyshev approximant $R^*_{5,4}(z)$ for $A_R(z)$ on $[Q_B(0.9), Q_B(1-10^{-10})]$.

```
precision = 20;
\[Gamma]1 = N[Sqrt[\[Alpha]1^2 - \[Beta]1^2], precision];
xM =   N[\[Beta]1/\[Gamma]1, precision];
pM = N[F[xM],     precision];
pR = N[E^((\[Alpha]1 - \[Beta]1) xM) (1 - pM), precision
   ];

FBR[x_] := 1 - E^(- (\[Alpha]1 - \[Beta]1) x) pR;(*Right
   Base CDF*)
QBR[u_] :=  -(1/(\[Alpha]1 - \[Beta]1)) Log[(1 - u)/pR];

u2 = 9/10; \[Tau]R = 10^-10;

ARInv[x_] := QBR[F[x]]
x1 = QApprox[u2];  x2 = QApprox[1 - \[Tau]R];
gmmaARInv =    GeneralMiniMaxApproximation[{ARInv[x], x},
    {x, {x1, x2}, 5, 4}, z,     WorkingPrecision ->
```



(a) $R^*_{7,6}(z)$ (b) $R^*_{5,4}(z)$



(c) $R^*_{10,10}(u)$

Figure 8.8.1: Chebyshev approximants

```
precision, Brake -> {20, 20}, Bias -> -0.5]
```

Chebyshev approximants for $A_L(z)$ may be computed similarly. Figure 8.8.1 (a) and (b) plot the resulting error curves. In this case it took 108 seconds to compute the Chebyshev approximant $R^*_{5,4}(z)$ for the right region and 7.3 seconds to compute the Chebyshev approximant $R^*_{7,6}(z)$ for the left region (note the left region is much smaller than the right region), both approximants achieve a similar accuracy.

8.8.2 Maehly's Indirect Method

There is one other method of computing $R^*_{m,n}$ which we discuss next that does not require evaluating $Q(u)$ (or $F(x)$) at a large number of points, it is Maehly's indirect method (Maehly, 1963). This method of determining $R^*_{m,n}(u)$ is applicable when a sufficiently accurate approximation of $Q(u)$ is available. Such an initial approximation may be obtained by constructing a Taylor series expansion or a regular C-fraction (see section 8.3). Maehly's indirect method then seeks to find a more optimal approximation in the Chebyshev sense without the need to evaluate $Q(u)$, which of course in our case is an expensive operation. Thus one would expect reduced setup up times with this approach. In fact according to Acton (1990, p. 310) when a sufficiently accurate power series or continued fraction representation for ξ is available Maehly's indirect method

is superior to all others, since the iterative steps in the other algorithms need to be preformed in high precision.

We first consider the polynomial case, assume that $Q(u)$ is given by a high degree polynomial

$$Q(u) \approx \bar{P}_M(u) = \sum_{k=0}^{M} q_k (u - u_0)^k$$

on the interval $[u_1, u_2]$. Our aim is to construct the best fit polynomial $P_m^*(u)$ ($m \ll M$) which minimizes $r_{m,0}$ with respect to the weight function $g(u) = 1$. Define $\Delta P_m(u) = P_m^*(u) - \bar{P}_m(u)$ and note that the error curve $\delta^*(u) = P_m^*(u) - Q(u)$ which characterizes the best fit polynomial may be written as

$$\delta^*(u) = \Delta P_m(u) - \left(Q(u) - \bar{P}_m(u)\right).$$

This shows that the polynomial $\Delta P_m(u)$ is the Chebyshev approximant to the function $Q(u) - \bar{P}_m(u)$. Key to Maehly's indirect method is a means to compute the discrepancy $D(u) = Q(u) - \bar{P}_m(u)$ without having to evaluate $Q(u)$. Since we have assumed $\bar{P}_M(u)$ is sufficiently accurate this may be achieved from the "tail" of the polynomial

$$D(u) = \sum_{k=m+1}^{M} q_k (u - u_0)^k. \tag{8.8.5}$$

We may now apply any direct method such as the Remez algorithm to (8.8.5) to find the Chebyshev approximant $\Delta P_m(u)$. The m^{th} degree Chebyshev approximant to $Q(u)$ is then given by

$$P_m^*(u) = \bar{P}_m(u) + \Delta P_m(u).$$

Note rather than computing the coefficients of $P_m^*(u)$ directly in Maehly's indirect method we compute the perturbations required to transform the Taylor polynomial $\bar{P}_m(u)$ into the best fit polynomial $P_m^*(u)$. The Mathematica code below is used to compute the Chebyshev approximant $P_{16}^*(u)$ of $Q(u)$ on $[0.1, 0.9]$ from the Taylor polynomial $\bar{P}_{100}(u)$ about $u = 1/2$.

```
m = 20;  M = 100;  u1 = 1/10; u2 = 9/10;
tail[u_] := Sum[q[k]*(u - u0)^k, {k, m + 1, M}];

Timing[
mmlist =    MiniMaxApproximation[1 + tail[u], {u, {u1, u2
   }, m, 0},          Brake -> {20, 20}]
]
```

```
\[CapitalDelta]P[u_] = mmlist[[2, 1]];

(*Chebyshev Approximant*)
P[u_] = Sum[q[k]*(u - u0)^k, {k, 0, m}] + (\[CapitalDelta
   ]P[u] - 1)

Plot[{P[u] - QRF[u]}, {u, u1, u2}]
```

The approximant is computed in a mere 0.531 seconds (including the time required to compute the Taylor coefficients); if on the other hand we were to construct $P^*_{16}(u)$ using `GeneralMiniMaxApproximation` as follows,

```
Timing[
x1 = QRFPade[u1]; x2 = QRFPade[u2];
gmma =    GeneralMiniMaxApproximation[{F[x], x, 1}, {x, {
   x1, x2}, 18, 0}, u,     WorkingPrecision -> 20]
]

mmaQ[u_] = gmma[[2, 1]]
```

the time required to compute $P^*_{16}(u)$ is over one minute! Not only is this method computationally expensive, it also requires high precision arithmetic where as with Maehly's indirect method we were able to construct $P^*_{16}(u)$ using double precision arithmetic, see remarks below. There are minor numerical differences between the coefficients of $P^*_{16}(u)$ generated by `GeneralMiniMaxApproximation` and Maehly's indirect method, but for all practical purposes as figure 8.8.2 shows the graphs of the error curves produced by either method are visually indistinguishable. The code to produce best fit polynomials for $A_L(z)$ and $A_R(z)$ is given in an online code repository (Munir, 2012). We make some remarks on this method below.

- Maehly's indirect method requires lower precision since if $\bar{P}_m(u)$ is a good approximant for $Q(u)$ then the coefficients of the polynomial $\Delta P_m(u) = P^*_m(u) - \bar{P}_m(u)$ will be considerably smaller in magnitude than the coefficients of $P^*_m(u)$, see Maehly's original paper.

- As the example above illustrates, since we need only evaluate $Q(u)$ once (at the point of expansion u_0), Maehly's indirect method is computationally more efficient. The crippling factor for `GeneralMiniMaxApproximation` is that for high order approximants we need to work in high precision. In contrast if we wish to compute a lower order approximant, say the 14^{th} degree Chebyshev approximant $P^*_{14}(u)$ then working in double precision is adequate. In this case the

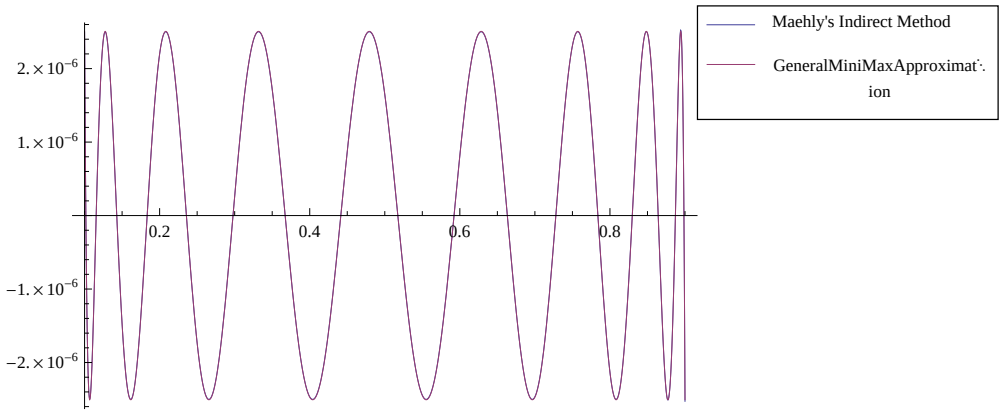


Figure 8.8.2: Comparison of error curve $\delta(u)$ produced by `GeneralMiniMaxApproximation` and Maehly's indirect method.

m	`Gen...Approximation`		Maehly's indirect method ($M = 100$)		Max error
	Setup Time	Precision Required	Setup Time	Precision Required	
$P_{14}^{*}(u)$	1.825	double	0.188	double	1.1×10^{-5}
$P_{16}^{*}(u)$	58.952	extended	0.531	double	2.5×10^{-6}
$P_{20}^{*}(u)$	90.496	extended	0.577	extended	1.27×10^{-7}

Table 8.3: Setup times and precision required by `GeneralMiniMaxApproximation` and Maehly's indirect method.

time required to compute $P_{14}^{*}(u)$ with `GeneralMiniMaxApproximation` is 1.825 seconds in contrast to the 0.188 seconds required by Maehly's indirect method. When higher degree approximants need to be computed say, $P_{20}^{*}(u)$ both methods require the use of high precision arithmetic, i.e. extended double precision. Specifically `GeneralMiniMaxApproximation` requires that the c.d.f. be computed in high precision whereas Maehly's indirect method requires that the Taylor coefficients and the Chebyshev approximant of the discrepancy be computed in high precision. Again we observe that Maehly's indirect method is computationally more efficient; `GeneralMiniMaxApproximation` requires over 90 seconds to compute $P_{20}^{*}(u)$ where as Maehly's indirect method requires only 0.577 seconds, we summarize these results in table 8.3.

- The error reported by Remez's algorithm between the best fit polynomial $\Delta P_m(u)$ and the discrepancy $D(u)$ serves as a good estimate of the maximal error $r_{m,n}$ between $P_m^{*}(u)$ and $Q(u)$.

All these considerations suggest that at least in theory, one could devise a program capable of generating best fit approximants for $Q(u)$ on the fly (i.e. at run time).

Maehly's indirect method is not restricted to polynomial approximations, it may

also be used to produce best fit rational approximants. Suppose that we have a sufficiently accurate continued fraction approximation to $Q(u)$ on $[u_1, u_2]$

$$Q(u) \approx \bar{R}_M(u) = q_0 + \frac{\alpha_1(u-u_0)}{1} + \frac{\alpha_2(u-u_0)}{1} + \cdots + \frac{\alpha_M(u-u_0)}{1}. \quad (8.8.6)$$

Such an approximation may be generated using the first $M+1$ Taylor coefficients of $Q(u)$ as discussed in section 8.3. We wish to compute the Chebyshev approximant $R^*_{m,n}(u)$ (where $n+m \ll M$ and $0 \leq m-n \leq 1$) to $Q(u)$ on $[u_1, u_2]$ which minimizes $r_{m,n}$ with respect to the weight function $g(u) = 1$. Let $v = m+n$, our starting point is the v^{th} convergent of the continued fraction 8.8.6

$$\bar{R}_v(u) = \frac{A_v(u)}{B_v(u)} = q_0 + \frac{\alpha_1(u-u_0)}{1} + \frac{\alpha_2(u-u_0)}{1} + \cdots + \frac{\alpha_v(u-u_0)}{1}$$

where $A_v(u)$ and $B_v(u)$ are the v^{th} numerator and denominator, recall from section 8.3 these are polynomials of degree m and n respectively. Define $\Delta R_{m,n}(u) = R^*_{m,n}(u) - \bar{R}_v(u)$ and note that the error curve $\delta^*(u) = R^*_{m,n}(u) - Q(u)$ which characterizes the best fit rational function may be written as

$$\delta^*(u) = \Delta R_{m,n}(u) - \left(Q(u) - \bar{R}_v(u)\right).$$

This shows that the rational function $\Delta R_{m,n}(u)$ is the Chebyshev approximant to the function $Q(u) - \bar{R}_v(u)$. Again key to Maehly's indirect method is a means compute the discrepancy $D(u) = Q(u) - \bar{R}_v(u)$ without having to evaluate $Q(u)$. In the polynomial case we simply used the "tail" of the Taylor expansion, for the rational case Maehly derived the following analogous formula to evaluate the discrepancy

$$D(u) = Q(u) - \bar{R}_v(u) = \frac{x^v \prod_{i=1}^{v}(-\alpha_i)}{B_v(u)\left[B_{v-1}(u) + \frac{B_v(u)}{Q_{v+1}}\right]} \quad (8.8.7)$$

where $B_v(u)$ is the v^{th} denominator given by (8.3.1) and Q_{v+1} is the "tail" of the continued fraction

$$Q_{v+1} = \frac{\alpha_{v+1}(u-u_0)}{1} + \frac{\alpha_{v+2}(u-u_0)}{1} + \cdots + \frac{\alpha_M(u-u_0)}{1}.$$

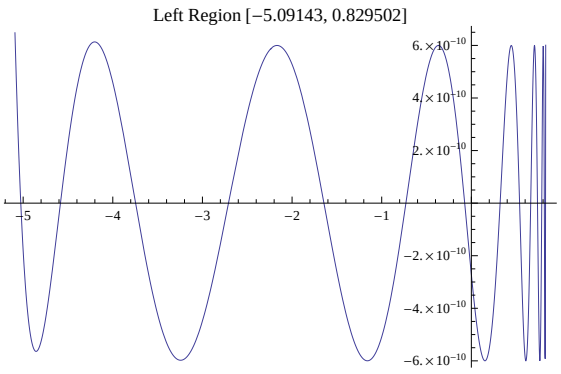
We may now apply any direct method such as the second algorithm of Remez to (8.8.7) to find the Chebyshev approximant $\Delta R_{m,n}(u)$ of the discrepancy $D(u)$ on $[u_1, u_2]$. The Chebyshev approximant $R^*_{m,n}(u)$ to $Q(u)$ is then given by

$$R^*_{m,n}(u) = \bar{R}_v(u) + \Delta R_{m,n}(u).$$

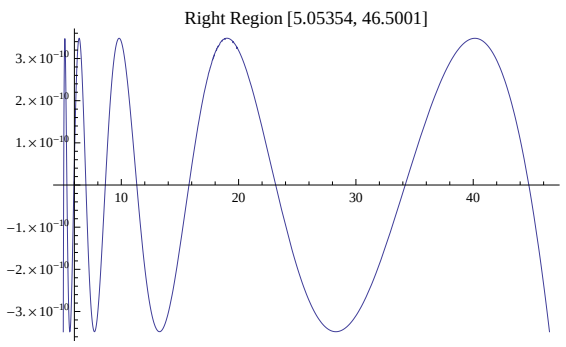
The Mathematica code below is used to compute the Chebyshev approximant $R^*_{9,9}(u)$ of $Q(u)$ on $[0.1, 0.9]$ from $\bar{R}_{50}(u)$. The code relies on some helper routines given in figure C.0.10 used to construct the convergents, v^{th} denominator and the tail of the continued fraction.

```
m = 9;  n = 9;  v = m + n;  M = 50;

Timing[

(*compute R_v from Taylor series expanded about u=1/2*)
taylorCoeffs = Table[q[n], {n, 0, M}];
partialNumerators = ProgressiveQD[taylorCoeffs];
ca[n_] := partialNumerators[[n]];
R[u_] = q[0] + BuildCFraction[partialNumerators[[1 ;; v
   ]], u];

(*compute the discrepancy*)
denominators = nthDenominator[partialNumerators[[1 ;; v
   ]], x];
B[n_] := denominators[[n]];
tail[x_] = conFracTail[v + 1, M, partialNumerators, x];
discrepancy[x_] = Simplify[(x^v*Product[-ca[i], {i, 1, v
   }])/          (B[v]*(B[v - 1] + B[v]/tail[x]))];

mmlist = MiniMaxApproximation[   1 + discrepancy[x - u0],
    {x, {0.1, 0.9}, m, n}, WorkingPrecision -> 20]

]

(*Chebyshev Approximant R_{m,n}*)
\[CapitalDelta]R[x_] = mmlist[[2, 1]]
BR[u_] = R[u - u0] + (\[CapitalDelta]R[u] - 1);
```

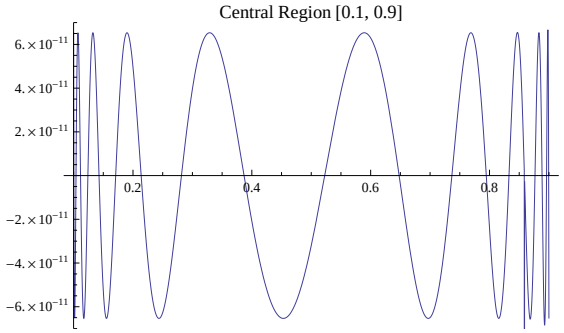
The analysis and code above extend in an obvious way to the functions $A_L(z)$ and $A_R(z)$, related error plots are given in figure 8.8.3 along with the number of Taylor coefficients required to compute the Chebyshev approximants. Similar comments about computational efficiency apply as in the polynomial case. On the practical side the lack of a robust method to generate the rational Chebyshev approximants $R^*_{n,m}(u)$ hinders progress. For example it may happen that `MiniMaxApproximation` and `GeneralMiniMaxApproximation` fail to compute $R^*_{3,3}(u)$ but are able to con-



(a) $A_L(z) - R^*_{7,6}(z)$ $(M = 60)$



(b) $A_R(z) - R^*_{5,4}(z)$ $(M = 90)$



(c) $Q(u) - R^*_{9,9}(u)$ $(M = 50)$

Figure 8.8.3: Chebyshev approximants generated from Maehly's indirect method.

struct a higher degree approximant $R^*_{4,3}(u)$ to the same function, see the Mathematica documentation for details. This is why $R^*_{7,6}(z)$ of figure 8.8.3 is not the true minimax approximation; we could only compute the best fit approximant $\Delta R_{10,3}(z)$ to the discrepancy as opposed to $\Delta R_{7,6}(z)$ using `MiniMaxApproximation`. Still, as figure 8.8.3 demonstrates the continued fraction approximation is significantly economized, i.e. the error is more uniformly spread over the interval of approximation.

8.9 Implementation

In this section we give a brief overview of a `C++` prototype which implements the first algorithm described above for the variance gamma quantile. The code is available from the online code repository Munir (2012). To implement the algorithm we need to model three vital components:

1. the variance gamma distribution,
2. the series expansions of the quantile function and
3. the type of approximant to be employed.

Consequently there are three parts to the software system. Figure 8.9.1 describes a class diagram for the first part. The primary objective of the `VGDistribution` class

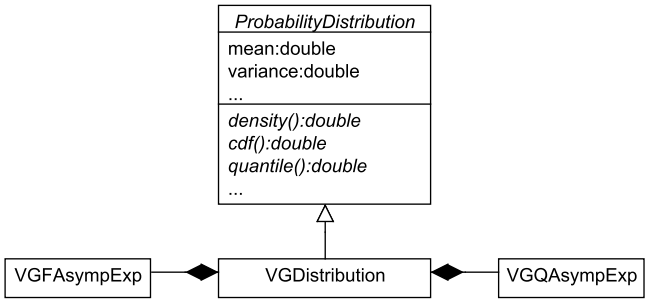


Figure 8.9.1: Probability Distribution Classes

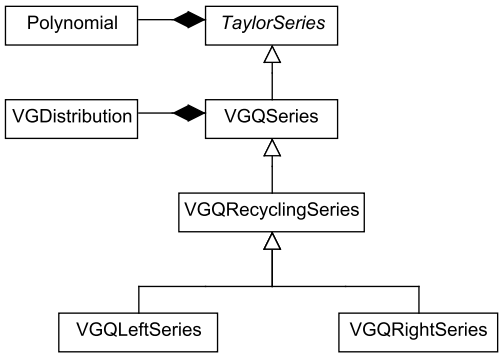


Figure 8.9.2: Series Class Hierarchy

is to provide high precision approximations of the c.d.f. and quantile function. These will be used for generating the initial conditions and controlling the error. Speed is not of primary concern here. To compute the c.d.f. we used numerical quadrature supplemented with the asymptotic expansions of the c.d.f. developed in section 6.3. Note the variance gamma density has an integrable singularity at the point $x = \mu$, quadrature schemes which account for this are particularly useful. Similarly to approximate the quantile function we use a stable root finding scheme supplemented with the asymptotic expansions of the quantile function, see (6.3.13), to generate the initial intervals. Working in double precision for the range of distribution parameters we have tested, a relative accuracy of 10^{-13} is attainable with this setup, albeit the computation of the quantile function is slow.

The class hierarchy for the second part of the system is shown in figure 8.9.2 which is responsible for generating the series developed in section 6.3.4. Most of the work is carried out by the `VGQSeries` class. In computing the coefficients it uses an optimization technique called memoization to avoid any redundant calculations.

An approximant for the variance gamma quantile function is constructed in the final part of the system shown in figure 8.9.3. In particular the `VGQApproximant` class partitions the unit interval as follows, $0 < \tau_L < u_L < u_0 < u_R < \tau_R < 1$, where u_0 is

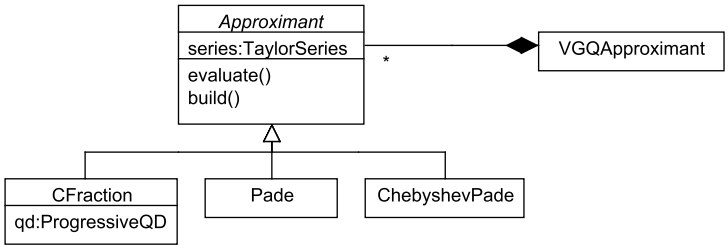


Figure 8.9.3: Approximation Classes

the zero quantile location (recall the Q_{VG} is not analytic at the point u_0), and constructs a continued fraction approximation on each part using the progressive quotient difference algorithm, see section 8.3 for details. In the tail regions the asymptotic expansions of Q are utilized. Of course other approximants such as Chebyshev-Padé approximants could be constructed if required, but we do not consider these in this revision of the software.

To demonstrate the utility of the algorithm we consider the following test case; the parameters of the variance gamma process fitted to four years worth of S&P data as reported in Seneta (2004) are $\hat{\sigma}^2 = 6.447 \times 10^{-5}$, $\hat{\nu} = 0.4220$, $\hat{\theta} = -1.510 \times 10^{-4}$ and $\hat{c} = 2.585 \times 10^{-4}$. These are related to μ, λ, α and β appearing in the variance gamma density (6.3.1) as follows,

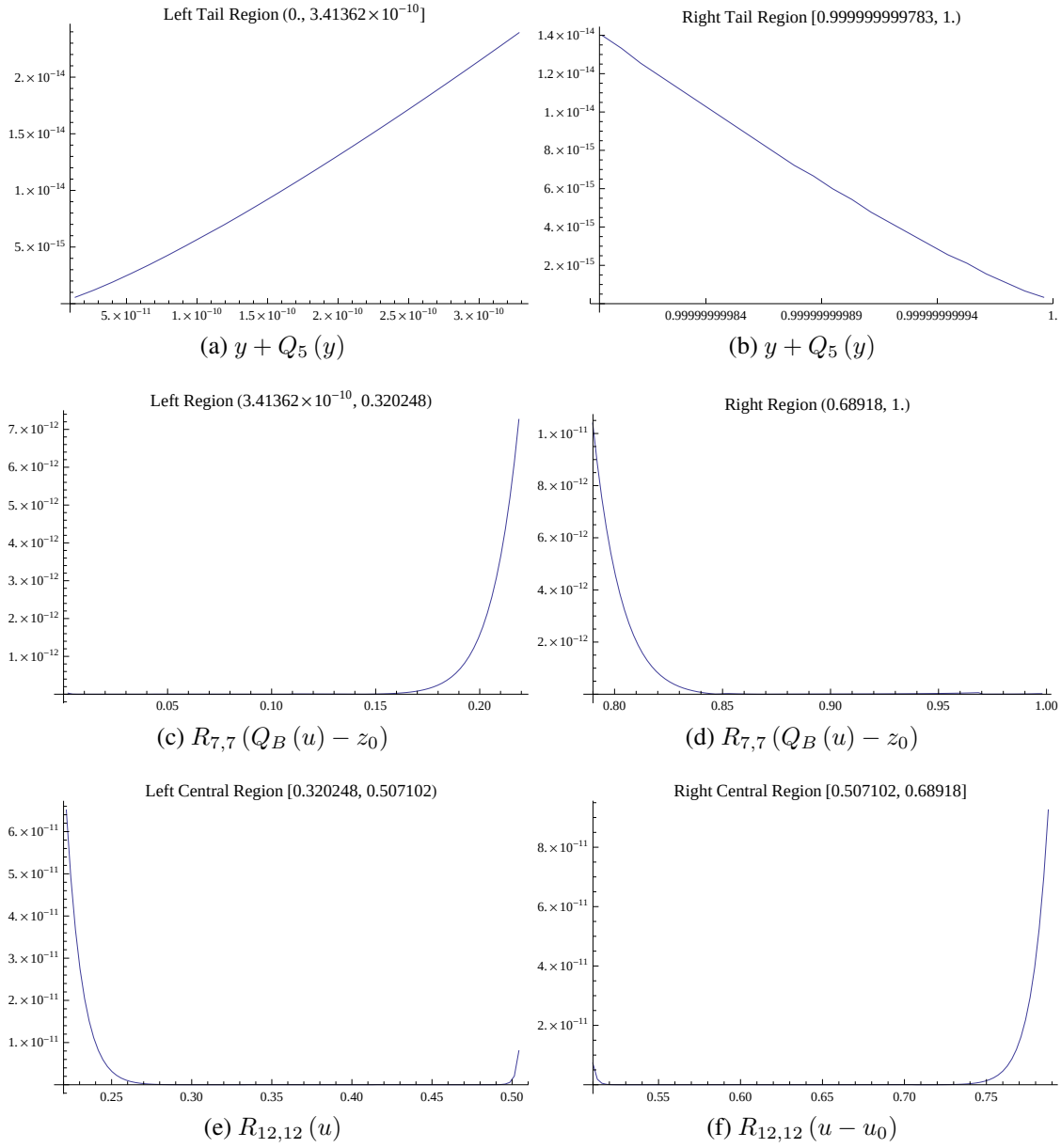
$$\mu = c, \quad \lambda = \frac{1}{\nu}, \quad \alpha = \frac{\sqrt{2\sigma^2/\nu + \theta^2}}{\sigma^2}, \quad \beta = \frac{\theta}{\sigma^2}.$$

Hence in our test case we use the estimated parameters; $\hat{\mu} = 0.0002585$, $\hat{\lambda} = 2.262443$, $\hat{\alpha} = 264.936625$ and $\hat{\beta} = -2.342174$. A plot of $Q_{\text{VG}}(u; \lambda, \alpha, \beta, \mu)$ is given in figure 6.3.1 on page 55. With a specified u-resolution of 10^{-10} on our test machine (an Intel i7 2.4GHz laptop) we observed the total time to sample one million variance gamma random variates to be less than 0.09 seconds. This is the fastest algorithm of its sort we know of. The resulting error plots are given in figure 8.9.4. The total time can be broken down as follows,

- 0.03 seconds setup time, i.e. the time required to build the approximant,
- 0.01 seconds to generate one million $\text{U}(0, 1)$ variates needed to apply the inversion method and
- 0.05 seconds execution time.

Of course the setup and execution times are dependent on the distribution parameters and the desired accuracy. To put things into perspective for the reader we generate one million standard normal uniform random variates using a `C++` implementation

of Acklam's (2009) method, without the refinement step. The execution time of this algorithm is then 0.03 seconds. Notice that the comparison is somewhat unjustified in the sense that our algorithm guarantees a u-resolution (i.e. the maximum u-error) of no greater than 1×10^{-10} where as Acklam's algorithm is designed to produce a maximum relative error of 1.15×10^{-9}. However the point of the comparison was to show that for a small extra cost (0.2 seconds to be precise, not including the setup time) we can sample from a variance gamma distribution. The primary reason Acklam's algorithm has a better execution time is that it uses rational approximants of degree 10 while we are employing rational approximants of degree 24 and 14 in our example. To reduce the degree and improve the execution time (at the expense of slower set up times) there are several possibilities other than increasing the number of subintervals (smaller intervals require lower degree approximants). For example one may construct Chebyshev-Padé approximants as discussed in section 8.6, economize the resulting continued fractions using the methods of Maehly (1963) or Ralston (1963) or construct minimax approximants as discussed in 8.8.



(a) $y + Q_5(y)$ (b) $y + Q_5(y)$

(c) $R_{7,7}(Q_B(u) - z_0)$ (d) $R_{7,7}(Q_B(u) - z_0)$

(e) $R_{12,12}(u)$ (f) $R_{12,12}(u - u_0)$

Figure 8.9.4: u-error $|u - F(Q_A(u))|$ plots generated by the Padé approximants of $Q_{\text{VG}}(u; \lambda, \alpha, \beta, \mu)$ where $\lambda = 2.262443$, $\alpha = 264.936625$, $\beta = -2.342174$ and $\mu = 0.0002585$. Here $Q_n(y)$ is used to denote the asymptotic series 6.3.13 truncated after the n^{th} term.

Chapter 9

Conclusion & Further Work

We summarize what we have achieved in this report and believe to be original contribution,

- For the hyperbolic, generalized inverse Gaussian and variance gamma distributions we have provided,
  - asymptotic expansions for the quantile function Q as $u \to 0$ and $u \to 1$,
  - convergent Taylor series expansions of Q at an arbitrary point $u \in (0, 1)$,
  - and solutions to the left and right recycling equations.

  in sections 6.2, 6.3 and 6.4 respectively.
- A method to compute Snedecor's F quantile was presented in section 6.1.
- We introduced a new approach in section 7.2 to obtain series representations for the quantile function when a Taylor series expansion of the c.d.f. is available. This technique proved particularly useful in determining both convergent and asymptotic series expansions for the quantile function of the α-Stable distribution.
- Various numerical schemes, not commonly found in the literature in the context of approximating the quantile functions for the VG $(\lambda, \alpha, \beta, \mu)$, Hyp $(\alpha, \beta, \delta, \mu)$ and GIG (λ, χ, ψ) distributions were explored in chapter 8, in particular it was seen that algorithms based on
  - Padé approximants,
  - truncated Chebyshev series,
  - Chebyshev-Padé approximants,
  - Multipoint Padé approximants,
  - and minimax approximants

prove useful.

It should be emphasized the techniques we have applied in this report are quite general, and applicable to a much wider range of distributions than we have considered in this report. Indeed examining the density (B.2.1) of the generalized hyperbolic distribution it certainly appears to be amenable to the quantile mechanics approach. From a financial perspective it would be interesting to see whether these techniques are also applicable to the noncentral chi-squared distribution, the CGMY model or the normal inverse Gaussian distribution. See also the remarks at the ends of sections 7.1 and 7.2.

Similarly we have only scratched the surface, in the context of the numerical techniques available for approximating analytic functions. We briefly mention some of these which we have not yet explored in any great detail. Methods to construct Chebyshev series solutions to nonlinear differential equations directly have been developed by Clenshaw and Norton (1963). When the Levin u or Weniger transforms are applied to a power series the result is a rational function, Roy et al. (1996, eq. 24) provide us with closed form expressions for the coefficients appearing in the numerator and denominator polynomials. Hänggi et al. (1980) study the summation of functions defined in terms of various types of series through the use of continued fractions from a numerical point of view. Castelianos and Rosenthal (1993) provide us with a procedure to construct rational Chebyshev approximations of analytic functions. However they fail to mention the coefficients appearing in equation (1.1) of their paper may be obtained through Faà di Bruno's formula. Wynn (1960) discusses very general techniques for constructing rational approximations for functions specifically defined by power series expansions. An interesting survey paper for practical rational approximations is provided by Cody (1970). Several researchers have developed methods to solve an osculatory rational interpolation problem, see for example Graves-Morris (1980), Claessens (1976) and Gustavson and Yun (1979) for some of the methods available. Indeed it would appear the numerical analyst is somewhat spoilt for choice, however the reliability and stability of these algorithms must be investigated.

Appendix A

The Pearson Family of Distributions

The Pearson family or system describes a family of probability distributions. Any distribution with a probability density function f satisfying an ordinary differential equation of the form

$$\frac{1}{f}\frac{df}{dx} = -\frac{a+x}{b_0 + b_1 x + b_2 x^2} \tag{A.0.1}$$

is a member of this family. The parameters a, b_0, b_1 and b_2 control the shape of the probability distribution. There are twelve different types of Pearson distributions, summarised in table A.1. The choice of parameters that recover the corresponding distributions can be found in Johnson et al. (1994). Note that the normal distribution is not assigned a type; it is in fact the limiting distribution of all other types.

Table A.1: Pearson Types

Common Name	**Type**
Normal	NA
Beta (Shifted)	I
Symmetric Beta	II
Gamma	III
Pearson Type IV	IV
Reciprocal of Gamma	V
F	VI
Student-t	VII
	VIII
	IX
Exponential	X
Pareto	XI
	XII

Next we consider the solutions to Pearson's differential equation (A.0.1). The

form of these solutions depends on the quadratic,

$$b_2x^2 + b_1x + b_0 = 0$$

Since the equation is separable its solution is easily obtained as,

$$f(x) = A\exp\left(-\int \frac{a+x}{b_0 + b_1x + b_2x^2}dx\right), \tag{A.0.2}$$

where A is a normalising constant chosen such that $\int_{-\infty}^{\infty} f(x)\,dx = 1$. We now look at some particular cases.

Example 35. The normal curve is recovered by setting $b_1 = b_2 = 0$. Hence,

$$f(x) = Ae^{\frac{-(x+a)^2}{2b_0}},$$

and using the fact $\int_{-\infty}^{\infty} e^{-cx^2}dx = \sqrt{\pi c}$, the constant A can be computed as $A = 1/\sqrt{2\pi b_0}$. The function f is thus the density of the normal distribution with mean $-a$ and variance b_0.

Example 36. The student-t curve is recovered by setting $b_1 = a = 0$, and with $b_1, b_2 > 0$ equation (A.0.2) becomes,

$$f(x) = A\left(b_0 + b_2x^2\right)^{-(2b_2)^{-1}}.$$

Setting $b_2 = 1/(\nu+1)$ and $b_0 = \nu/(\nu+1)$ we find,

$$f(x) = A\left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

where we have absorbed the factor $\nu/(\nu+1)$ into the constant A. In this case the normalising constant is given by,

$$A = \frac{\Gamma\left(1+\frac{\nu}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)}.$$

From which we see that f is the density of the student-t distribution with ν degrees of freedom.

Appendix B

Variance-Mean Mixture Distributions

In this appendix we will introduce normal variance-mean mixture distributions, and in particular the family of generalized hyperbolic (GH) distributions. Amongst others, members of the GH family include the student-t, hyperbolic, normal inverse Gaussian and variance gamma distributions, all of which are of particular interest to finance. For a financial based introduction see Weron (2004) or Bingham and Kiesel (2002). For a more general yet readable treatment see Paolella (2007), our presentation closely follows this reference. The distribution of a random variable X is said to be a normal variance-mean mixture if the conditional distribution of X given Y (the mixing variable) is the normal distribution with mean $\mu + \beta Y$ and variance Y,

$$(X \mid Y) \sim \mathrm{N}\left(\mu + \beta Y, Y\right),$$

where $\mathrm{N}\left(m, s^2\right)$ denotes the normal distribution with mean m and variance s^2, Y is a positive random variable independent of X and μ and β are constants. It follows that X may be expressed as

$$X = \mu + \beta Y + \sqrt{Y} Z,$$

where $Z \sim \mathrm{N}(0, 1)$. This particular form of X makes simulation a simple procedure, see Weron (2004). If Y is a continuous random variable with density function f_Y, the density f_X of X can be written down as

$$\begin{aligned} f_X(x) &= \int_0^\infty f_{X,Y}(x, y)\, dy \\ &= \int_0^\infty f_{X|Y}(x|y)\, f_Y(y)\, dy \\ &= \int_0^\infty f_{\mathrm{N}}(x; \mu + \beta y, y)\, f_Y(y)\, dy. \end{aligned} \tag{B.0.1}$$

Here $f_{X,Y}$ and $f_{X|Y}$ denote the joint and conditional density function of X and Y respectively, and $f_{\mathrm{N}}(\mu; m, s^2)$ denotes the density of $\mathrm{N}(m, s^2)$. From this we see that X has a continuous mixture distribution[1]; we are mixing normal distributions with the distribution of Y. We will denote the distribution of X by $\mathrm{Mix}_{\pi}(\mu, \beta)$, where π is the distribution of Y, μ is a location parameter, and usually if $\beta \neq 0$ the distribution of X will be skewed. Also of interest will be the characteristic function ϕ_X of X which we derive next.

$$\begin{aligned}
\phi_X(t) &= \int_{-\infty}^{\infty} e^{ixt} f_X(x)\, dx \\
&= \int_{-\infty}^{\infty} e^{ixt} \int_0^{\infty} f_{\mathrm{N}}(x; \mu + \beta y, y)\, f_Y(y)\, dy dx \\
&= \int_0^{\infty} \int_{-\infty}^{\infty} e^{ixt} f_{\mathrm{N}}(x; \mu + \beta y, y)\, dx f_Y(y)\, dy \\
&= \int_0^{\infty} e^{i(\mu+\beta y)t - \frac{1}{2}yt^2} f_Y(y)\, dy \\
&= e^{i\mu t} \phi_Y\left(\beta t + it^2/2\right), \qquad \text{(B.0.2)}
\end{aligned}$$

where ϕ_Y is the characteristic function of Y. Now we will look at a special choice of Y which leads to a very interesting family of distributions, namely the generalised hyperbolic distributions.

B.1 Generalized Inverse Gaussian

The generalised inverse Gaussian (GIG) is a three parameter distribution, special cases of which are the inverse gamma, positive hyperbolic and Lévy distributions to name a few. It arises in the context of first passage times of a diffusion process. The probability density function of a GIG random variable is given by,

$$f_{\mathrm{GIG}}(x; \lambda, \chi, \psi) = \frac{(\psi/\chi)^{\lambda/2}}{2\mathrm{K}_{\lambda}\left(\sqrt{\psi\chi}\right)} x^{\lambda-1} e^{-\frac{1}{2}\left(\chi x^{-1} + \psi x\right)}, \quad x > 0, \qquad \text{(B.1.1)}$$

where $\mathrm{K}_v(z)$ is the modified Bessel function of the second kind with index $v \in \mathbb{R}$, and has the integral representation

[1] Suppose C is a continuous random variable with support $\mathcal{C}$. A random variable X is said to follow a continuous mixture distribution if its density function can be expressed as, see Paolella (2007, p. 258),

$$f_X(x) = \int_{\mathcal{C}} f_{X|C}(x\,|c)\, f_C(c)\, dc.$$

$$\mathrm{K}_v(t) = \frac{1}{2}\int_0^\infty x^{v-1}e^{-\frac{1}{2}t\left(x+x^{-1}\right)dx},$$

see Olver et al. (2010, §10.25). Denote by the Θ_{GIG} the parameter space of the distribution, it is defined as the union of three disjoint sets $\Theta_{\text{GIG}} = A \cap B \cap C$ where,

$$\begin{aligned}
A &= \left\{(\lambda,\chi,\psi) \in \mathbb{R}^3 : \lambda \in \mathbb{R}, \chi > 0, \psi > 0\right\} \quad \text{(normal case)}\\
B &= \left\{(\lambda,\chi,\psi) \in \mathbb{R}^3 : \lambda > 0, \chi = 0, \psi > 0\right\} \quad \text{(boundary case I)}\\
C &= \left\{(\lambda,\chi,\psi) \in \mathbb{R}^3 : \lambda < 0, \chi > 0, \psi = 0\right\} \quad \text{(boundary case II)}.
\end{aligned}$$

The GIG distribution generalises quite a number of distributions which are summarised in table B.1. The normal case, that is when $(\lambda,\chi,\psi) \in A$, is referred to as the GIG distribution in the strict sense. The boundary cases are present to include some limiting cases, for example in the first boundary case the GIG distribution reduces to the Gamma distribution.

Table B.1: Special Cases of GIG Distribution

Name	Parameter Range		
Normal Case	$\lambda \in \mathbb{R}$	$\chi > 0$	$\psi > 0$
Gamma	$\lambda > 0$	$\chi = 0$	$\psi > 0$
Inverse Gamma	$\lambda < 0$	$\chi > 0$	$\psi = 0$
Exponential	$\lambda = 1$	$\chi = 0$	$\psi > 0$
Positive Hyperbolic	$\lambda = 1$	$\chi > 0$	$\psi > 0$
Lévy	$\lambda = -1/2$	$\chi > 0$	$\psi = 0$
Inverse Gaussian	$\lambda = -1/2$	$\chi > 0$	$\psi > 0$
Dirac (limiting case, $x \geq 0$)	$\lambda \in \mathbb{R}$	$\sqrt{\frac{\chi}{\psi}} \to x$	$\sqrt{\chi\psi} \to \infty$

Later we will use the GIG distribution as a mixing distribution in a normal variance mean mixture. From equation (B.0.2) we see that the characteristic function of this normal mean mixture will depend on the characteristic function of the GIG distribution. Thus we provide a derivation.

$$\begin{aligned}
\phi_{\text{GIG}}(t;\lambda,\chi,\psi) &= \int_{-\infty}^{\infty} e^{itx} f_{\text{GIG}}(x;\lambda,\chi,\psi)\,dx\\
&= \frac{(\psi/\chi)^{\lambda/2}}{2\mathrm{K}_\lambda\left(\sqrt{\psi\chi}\right)}\int_0^\infty x^{\lambda-1}\exp\left\{-\frac{1}{2}\left(\chi x^{-1} + (\psi - 2it)\,x\right)\right\}dx
\end{aligned}$$

Writing the integrand in this expression as,

$$x^{\lambda-1}\exp\left\{-\frac{1}{2}\sqrt{\chi\left(\psi-2it\right)}\left(\left(\sqrt{\frac{\psi-2it}{\chi}}x\right)^{-1}+\left(\sqrt{\frac{\psi-2it}{\chi}}x\right)\right)\right\},$$

and applying the change of variable $y=\sqrt{\left(\psi-2it\right)/\chi}x$ we obtain,

$$\phi_{\text{GIG}}\left(t;\lambda,\chi,\psi\right)=\left(\frac{\psi}{\psi-2it}\right)^{\lambda/2}\frac{\text{K}_\lambda\left(\sqrt{\chi\left(\psi-2it\right)}\right)}{\text{K}_\lambda\left(\sqrt{\psi\chi}\right)}. \tag{B.1.2}$$

B.2 Generalized Hyperbolic Distribution

The GH distribution is a normal mean mixture distribution where the mixing distribution is a GIG distribution. Although this provides a natural parametrisation of λ, χ and ψ from the GIG distribution along with μ and β from the mixing procedure another parametrisation is more commonly used. We will instead use the parameters λ, α, β, δ and μ. The parameter space of the generalised hyperbolic distribution is defined as $\Theta_{\text{GH}}=\{(\lambda,\alpha,\beta,\delta,\mu)\in\mathbb{R}^5:\ \alpha,\delta\geq 0,\ \lambda,\mu\in\mathbb{R},\ |\beta|\leq\alpha\}$. By setting $\chi:=\delta^2$ and $\psi:=\alpha^2-\beta^2$ we may define the generalised hyperbolic distribution $\text{GHyp}\left(\lambda,\alpha,\beta,\delta,\mu\right)$ as a normal variance mean mixture where the mixing distribution is $\text{GIG}(\lambda,\chi,\psi)$,

$$\text{GHyp}\left(\lambda,\alpha,\beta,\delta,\mu\right):=\text{Mix}_{\text{GIG}(\lambda,\,\delta^2,\,\alpha^2\,-\,\beta^2)}\left(\mu,\beta\right).$$

In order to write down the associated density function we must evaluate the integrals appearing in (B.0.1). This is a rather tedious task, but once we know the density function in the general case we can easily write down the density of the more specialized cases. The same applies to the characteristic function. For notational convenience set $\gamma=\sqrt{(\alpha^2-\beta^2)}$, the density may then be derived as follows,

$$\begin{aligned}
f_{\text{GHyp}}\left(x;\lambda,\alpha,\beta,\delta,\mu\right) &= f_{\text{Mix}_{\text{GIG}\left(\lambda,\,\delta^2,\,\alpha^2\,-\,\beta^2\right)}}\left(x;\mu,\beta\right)\\
&= \int_0^\infty f_{\text{N}}\left(x;\mu+\beta y,y\right)f_{\text{GIG}}\left(x;\lambda,\chi,\psi\right)dy\\
&= \frac{1}{\sqrt{2\pi}}\frac{\left(\gamma/\delta\right)^\lambda}{2\text{K}_\lambda\left(\gamma\delta\right)}\int_0^\infty\exp\left\{-\frac{1}{2}\frac{\left(x-(\mu+\beta y)\right)^2}{y}\right\}\\
&\qquad\times y^{\left(\lambda-\frac{1}{2}\right)-1}\exp\left\{-\frac{1}{2}\left(\delta^2y^{-1}+\gamma^2y\right)\right\}dy\\
&= \frac{1}{\sqrt{2\pi}}\frac{\left(\gamma/\delta\right)^\lambda}{2\text{K}_\lambda\left(\gamma\delta\right)}e^{\beta(x-\mu)}\int_0^\infty y^{\left(\lambda-\frac{1}{2}\right)-1}\\
&\qquad\times\exp\left\{-\frac{1}{2}\left(\left(\delta^2+(x-\mu)^2\right)y^{-1}+\alpha^2y\right)\right\}dy.
\end{aligned}$$

Note the argument of the exponential function appearing in the integrand may be written as

$$-\frac{1}{2}\alpha\sqrt{\delta^2+(x-\mu)^2}\left(\left(\frac{\alpha}{\sqrt{\delta^2+(x-\mu)^2}}y\right)^{-1}+\left(\frac{\alpha}{\sqrt{\delta^2+(x-\mu)^2}}y\right)\right).$$

Hence by applying the change of variable $v=\alpha y/\sqrt{\delta^2+(x-\mu)^2}$ we obtain

$$\begin{aligned} f_{\text{GHyp}}(x;\lambda,\alpha,\beta,\delta,\mu) &= \frac{(\gamma/\delta)^{\lambda}e^{\beta(x-\mu)}}{\sqrt{2\pi}}\left(\frac{\sqrt{\delta^2+(x-\mu)^2}}{\alpha}\right)^{\lambda-\frac{1}{2}} \\ &\quad\times\frac{\mathrm{K}_{\lambda-\frac{1}{2}}\left(\alpha\sqrt{\delta^2+(x-\mu)^2}\right)}{\mathrm{K}_{\lambda}(\gamma\delta)} \end{aligned} \tag{B.2.1}$$

Note that there is another formulation of the GHyp density that will prove useful. To this end we define a new function $\mathrm{k}_{\lambda}(\chi,\psi)$, closely related to the modified Bessel function of the second kind

$$\mathrm{k}_{\lambda}(\chi,\psi):=\int_0^{\infty}x^{\lambda-1}\exp\left\{-\frac{1}{2}\left(\chi x^{-1}+\psi x\right)\right\}dx.$$

It can be shown that (Paolella, 2007, p. 301),

$$\mathrm{k}_{\lambda}(\chi,\psi)=2\left(\sqrt{\frac{\chi}{\psi}}\right)^{\lambda}\mathrm{K}_{\lambda}\left(\sqrt{\chi\psi}\right). \tag{B.2.2}$$

Thus we may rewrite the GHyp density in terms of $\mathrm{k}_{\lambda}(\chi,\psi)$,

$$f_{\text{GHyp}}(x;\lambda,\alpha,\beta,\delta,\mu)=\frac{\mathrm{k}_{\lambda-\frac{1}{2}}\left(\delta^2+(x-\mu)^2,\alpha^2\right)}{\sqrt{2\pi}\mathrm{k}_{\lambda}(\delta^2,\alpha^2-\beta^2)}e^{\beta(x-\mu)}. \tag{B.2.3}$$

Two other identities that can be proven from elementary techniques of integration which we will need later are,

$$\mathrm{k}_{\lambda}(0,\psi)=\left(\frac{\psi}{2}\right)^{-\lambda}\Gamma(\lambda),\quad \lambda>0,\psi>0, \tag{B.2.4}$$

and

$$\mathrm{k}_{\lambda}(\chi,0)=\left(\frac{\chi}{2}\right)^{\lambda}\Gamma(-\lambda),\quad \lambda<0,\chi>0. \tag{B.2.5}$$

From (B.0.2) and (B.1.2) we obtain the characteristic function of the GHyp distribution.

$$\phi_{\text{GHyp}}(t) = e^{i\mu t}\left(\frac{\alpha^2-\beta^2}{\alpha^2-\beta^2+t(t-2i\beta)}\right)^{\frac{\lambda}{2}} \frac{\mathrm{K}_\lambda\left(\delta\sqrt{(\alpha^2-\beta^2-t(t-2i\beta))}\right)}{\mathrm{K}_\lambda(\gamma\delta)}. \tag{B.2.6}$$

Now we will have a look at some specific distributions of the GH family. Focusing in particular on the the density and the characteristic functions, which are just special cases of (B.2.1) and (B.2.6) respectively, with added parameter constraints.

Example 37 (Hyperbolic, if $\lambda = 1$, $\alpha > 0$, $|\beta| < \alpha$, $\delta > 0$, $\mu \in \mathbb{R}$)**.** With these parameter constraints in place we see that, $\lambda = 1$, $\chi = \delta^2 > 0$ and $\psi = \alpha^2 - \beta^2 > 0$, which suggests from table (B.1) that the hyperbolic (Hyp) distribution is a normal variance mean mixture where the mixing distribution is the positive hyperbolic distribution. The Hyp distribution is defined as,

$$\text{Hyp}(\alpha,\beta,\delta,\mu) := \text{GHyp}(1,\alpha,\beta,\delta,\mu).$$

The formulae for the density and characteristic functions are then easily derived as

$$f_{\text{Hyp}}(x;\alpha,\beta,\delta,\mu) = \frac{\gamma}{2\alpha\delta\mathrm{K}_1(\delta\gamma)}\exp\left\{-\alpha\sqrt{\delta^2+(x-\mu)^2}+\beta(x-\mu)\right\} \tag{B.2.7}$$

and

$$\phi_{\text{Hyp}}(t) = \frac{\gamma e^{i\mu t}}{\sqrt{\alpha^2-(it+\beta)^2}}\frac{\mathrm{K}_1\left(\delta\sqrt{\alpha^2-(it+\beta)^2}\right)}{\mathrm{K}_1(\gamma\delta)}$$

respectively.

Example 38 (Normal Inverse Gaussian, if $\lambda = -1/2$, $\alpha > 0$, $|\beta| < \alpha$, $\delta > 0$, $\mu \in \mathbb{R}$)**.** The normal inverse Gaussian (NIG) distribution is defined as,

$$\text{NIG}(\alpha,\beta,\delta,\mu) := \text{GHyp}(-1/2,\alpha,\beta,\delta,\mu).$$

The formulae for the density and characteristic functions are then easily derived as

$$f_{\text{NIG}}(x;\alpha,\beta,\delta,\mu) = \left(\frac{\alpha\delta}{\pi\sqrt{\delta^2+(x-\mu)^2}}\right)e^{\beta(x-\mu)+\delta\gamma}\mathrm{K}_1\left(\alpha\sqrt{\delta^2+(x-\mu)^2}\right),$$

and

$$\phi_{\text{NIG}}(t) = e^{iut+\delta\left(\sqrt{(\alpha^2-\beta^2)}-\sqrt{\alpha^2-(\beta+it)^2}\right)}$$

respectively.

Example 39 (Variance Gamma, if $\lambda > 0$, $\alpha > 0$, $|\beta| < \alpha$, $\delta = 0$, $\mu \in \mathbb{R}$)**.** With these parameter constraints in place we see that, $\lambda = 1$, $\chi = \delta^2 = 0$ and $\psi = \alpha^2 - \beta^2 > 0$, which suggests from table (B.1) that the variance gamma (VG) distribution is a normal variance mean mixture where the mixing distribution is the gamma distribution. The VG distribution is defined as,

$$\text{VG}(\lambda, \alpha, \beta, \mu) := \text{GHyp}(\lambda, \alpha, \beta, 0, \mu).$$

The formula for the density is a little more difficult to compute due to the $\delta = 0$ appearing in the denominator of (B.2.1). It is much more convenient however to work with the form of the GHyp density given by (B.2.3) and use the identities (B.2.4) and (B.2.2) to obtain,

$$f_{\text{VG}}(x; \lambda, \alpha, \beta, \mu) = \frac{\gamma^{2\lambda}}{(2\alpha)^{\lambda-1/2}\sqrt{\pi}\Gamma(\lambda)} |x-\mu|^{\lambda-\frac{1}{2}} \text{K}_{\lambda-\frac{1}{2}}(\alpha|x-\mu|) e^{\beta(x-\mu)}. \tag{B.2.8}$$

It is a simple matter to write down the characteristic function from (B.2.6),

$$\phi_{\text{VG}}(t) = e^{i\mu t}\left(\frac{\alpha^2-\beta^2}{\alpha^2-\beta^2+t(t-2i\beta)}\right)^{\frac{\lambda}{2}}.$$

Appendix C

Code Listings

In this section we provide Mathematica (Wolfram-Research, 2010) implementations of some of the algorithms we have employed in this report.

```
(*
   The Box Muller transform;
   Input: A pair of U(0,1) random deviates;
   Output: A N(0,1) random variate;
*)
BoxMuller[uniformVariates_] := Module[{e, θ, k, n, U1, U2, Y, Z, normalVariates},
  U1 = N[uniformVariates[[1]]];
  U2 = N[uniformVariates[[2]]];
  Y = √(-2 Log[U1]) Cos[2 π U2];
  Z = √(-2 Log[U1]) Sin[2 π U2];
  (* We are only interested in examining Z *)
  Z
 ]

(*Normal Quantile*)
Q[u_] := Quantile[NormalDistribution[0, 1], u];

(*Generate the U(0,1) variates using a congruential generator*)
n = 2 000 000;
uniformVariates =
  BlockRandom[SeedRandom[1, Method → {"Congruential", "Multiplier" → 131,
      "Increment" → 0, "Modulus" → 2^35}]; RandomReal[{0, 1}, n]];
(*Pair them up becuase Box muller works in pairs*)
uniformPairs =
   Table[{uniformVariates[[k]], uniformVariates[[k + 1]]}, {k, 1, n - 1, 2}];

(*Apply the Box Muller Transform*)
boxMullerVariates = Map[BoxMuller, uniformPairs] ;

Print["Range of variates generated using Box Muller method: [",
 Min[boxMullerVariates], ", " , Max[boxMullerVariates], "]"]

(*Do the same with the inverse transform*)
QVariates = Map[Q, uniformVariates[[1 ;; n / 2]]] ;

Print["Range of variates generated using the inversion method: [",
 Min[QVariates], ", " , Max[QVariates], "]"]
Range of variates generated using Box Muller method: [-3.21405, 3.54349]

Range of variates generated using the inversion method: [-5.77654, 4.57778]
```

Figure C.0.1: The Neave Effect

```
(*classical parameterisation P1 of the stable distribution*)
α = N[1.7811]; σ1 = N[0.0141]; β1 = N[0.2834]; μ1 = N[0.0009];

(* Convert to Zoltarev's Type (B) parameterisation P2 *)
k = N[α - 1 + Sign[1 - α]];
β2 = N[2 ArcTan[β1 Tan[π α / 2]] / (π k)];
σ2 = N[σ1^α Sqrt[1 + β1^2 Tan[π α / 2]^2]];
μ2 = N[μ1 σ1^-α / Sqrt[1 + β1^2 Tan[π α / 2]^2]];

θ = N[β2 / α k]; ρ = N[(1 + θ) / 2];

(*CDF central series*)
f0 = 1/2 (1 - β2 k / α);
fc[n_] := {fc[0] = f0                                              n == 0
           fc[n] = (-1)^(n-1) 1/π Gamma[n/α + 1] / n Sin[π n ρ]    n ≥ 1};
FC[x_, n_] := Sum[fc[k] / k! x^k, {k, 0, n}];
(*CDF tail series*)
ft[j_, β_] := {ft[0, β] = 1                                                  j == 0
               ft[j, β] = (-1)^j 1/(α π) Gamma[j α + 1] / j Sin[j π / 2 (α + β k)]    j ≥ 1};
bt[j_, β_] := {bt[0, β] = 1                                                  j == 0
               bt[j, β] = (-1)^j 1/(α π) Gamma[j α + 1] / j Sin[j π / 2 (α + β k)]    j ≥ 1};
v[x_] := x^-α
FT[v_, β_, n_] := Sum[ft[j, β] / (j!) v^j, {j, 0, n}];
(*quantile central series*)
qc[n_] := {qc[n] = 1 / fc[1]                                                  n == 1
           qc[n] = -1/fc[1]^n Sum[qc[k] BellY[n, k, Table[fc[i], {i, 1, n - k + 1}]], {k, 1, n - 1}]    n ≥ 2}

QC[u_, n_] := Sum[N[qc[k]] / k! u^k, {k, 1, n}];
(*quantile asymptotic series*)
qt[n_, β_] :=
 {qt[n, β] = 1 / ft[1, β]                                                  n == 1
  qt[n, β] = -1/ft[1, β]^n Sum[qt[k, β] BellY[n, k, Table[ft[i, β], {i, 1, n - k + 1}]], {k, 1, n - 1}]    n ≥ 2}

QT[u_, β_, n_] := Sum[qt[k, β] / k! (u - 1)^k, {k, 0, n}]
```

Figure C.0.2: Mathematica implementation of α-stable c.d.f. and quantile series

```
(*Generates the first m+
  2 terms of the asyptotic expansion of the VG quantile as u → 1*)
VGAsympQ[m_] := Module[{a, b, AsympF, AsympQ, P, A, u, result, coefficientList},
  (*Asyptotic Expansion of the VG cdf as x → ∞*)
  a[k_, v_] := 1/(k! 8^k) Product_{j=1}^{k} (4 v^2 - (2 j - 1)^2);
  b[k_] := Sum_{j=0}^{k} (Product_{i=0}^{j-1} (λ - k + i)) (α - β)^-(j+1) / α^(k-j) a[k - j, λ - 1/2];
  AsympF[n_, x_] := e^(-x (α-β)) x^-(1-λ) Sum_{k=0}^{n} Simplify[b[k]/x^k];
  (*Asyptotic Expansion of the VG quantile as u → 1*)
  P[n_, ξ_] := Module[{α = (λ - 1)/(α - β)},
    {P[n, ξ] = α ξ + c[0]                                                    n == 0
     P[n, ξ] = c[n] + α (P[n - 1, ξ] - P[n - 1, 0]) - α (n - 1) ∫_0^ξ P[n - 1, t] dt   n ≥ 0}];
  A[n_, x_] := Sum_{k=0}^{n} b[k] x^k;
  u[n_] := Module[{γ = (λ - 1)/(α - β)},
    {u[0] = Log[b[0]]                                                        n == 0
     u[n] = Series[γ Log[1 + t u[n - 1]] + 1/(α-β) Log[A[n, t/(1+t u[n-1])]], {t, 0, n}]   n ≥ 1}];
  AsympQ[n_, y_, β_] := y + Sum_{k=0}^{n} Simplify[P[k, Log[y]]/y^k];
  result = Timing[
    coefficientList = Simplify[CoefficientList[u[m + 1], t]];
    c[n_] := coefficientList[[n + 1]];
    AsympQ[m, y, β]
   ];
  result
 ]

(*Usage*)
res = VGAsympQ[1];
Print[StringForm["Construction Time = ``", res[[1]]] ]
AsympQ[y_, β_] = res[[2]]
Construction Time = 0.06200000000012551`
y + ((-1 + λ) Log[y] + Log[1/(α-β)])/(α - β) + ((-1 + λ) (2 α + α λ - β λ + 2 α (-1 + λ) Log[y] + 2 α Log[1/(α-β)]))/(2 y α (α - β))
```

Figure C.0.3: Mathematica implementation of Salvy's method applied to variance gamma distribution function.

```
(*Some Test Parameters*)
μ = 0; δ = 1; α = 2; β = 3 / 2;

(*location scale Invariant Parameters*)
α1 = α δ; β1 = β δ;

(*coefficients*)
a[n_] := { a[0] = 0            n == 0
         { a[1] = 1/2 + q[1]   n == 1;
         { a[n] = q[n] + d[n]  n ≥ 2

b[n_] := b[n] = α1 a[n] - β1 q[n]

c[n_] := { c[0] = 1                          n == 0
         { c[n] = 1/n Σ_{k=1}^{n} k b[k] c[n - k]  n ≥ 1;

d[n_] := d[n] = 1/(n - 1) Σ_{k=0}^{n-2} (k + 1) (q[k + 1] q[n - k - 2] - a[k + 1] a[n - k - 2])

q[n_] := { q[0] = 0                                                          n == 0
         { q[1] = -α1/(2 (α1 - β1))                                          n == 1;
         { q[n] = -1/(α1 - β1) ((n - 1) q[n - 1] + 1/n Σ_{k=1}^{n-1} k b[k] c[n - k] + α1 d[n])   n ≥ 2

W[n_, z_] := Σ_{k=0}^{n} q[k] z^k;


γ1 = N[√(α1^2 - β1^2)];
A = N[2 α1 BesselK[1, γ1] / γ1];
y[u_] := -Log[A (α1 - β1) (1 - u)] / (α1 - β1);
z[u_] := 1 / y[u];

padeApprox[z_] = PadeApproximant[W[8, z], {z, 0, {4, 4}}];

Plot[{
  InverseCDF[HyperbolicDistribution[α, β, δ, μ], u],
  y[u] + padeApprox[z[u]],
  y[u] + W[4, z[u]],
  y[u] + W[6, z[u]],
  y[u] + W[8, z[u]]}, {u, 0, 1}, PlotRange → {-2, 10}]
```

Figure C.0.4: Generating Hyperbolic Asymptotic Expansion.

```
(*Input: Set of taylor coefficients c0,...,cn
 Output: Set of partial numerators appearing in associated C-Fraction*)
ProgressiveQD[taylorCoeffs_] :=
 Module[{e, q, c, d, n, k, l, i, j, partialNumerators, parNum},
  n = Length[taylorCoeffs] - 1;
  c[m_] := taylorCoeffs[[m + 1]];
  d[m_] := { d[0] = 1/c[0]                          m == 0
           { d[m] = -1/c[0] Sum_{i=1}^{m} c[i] d[m - i]   m ≥ 1 ;
  (*Initialisation*)
  q[0, 1] = -d[1]/d[0];
  e[1, 0] = 0;
  e[0, 1] = d[2]/d[1];
  For[k = 1, k ≤ n - 2, k++,
   q[-k, k + 1] = 0;
   e[-k, k + 1] = d[k + 2]/d[k + 1];
  ];

  partialNumerators = {c[1]};
  For[i = 0, i ≤ n - 2, i++, (*iterate thru the diagonals*)
   For[j = 0, j ≤ i, j++,  (*iterate thru entries in diagonals*)
     (*determine value of k and l,
     see Handbook of Continued fractions see p. 110 diagram*)
     k = -(n - 2) + i;
     l = (n - 1) - i + Floor[j / 2];
     If[EvenQ[j],
      parNum = q[k + 1, l] = (e[k, l] - e[k + 1, l - 1]) + q[k, l]; ,
      parNum = e[k + 1, l] = q[k, l + 1]/q[k + 1, l] e[k, l];
     ];
     If[k == 0, partialNumerators = Append[partialNumerators, -parNum]];
    ];
  ];
  partialNumerators
 ]
```

Figure C.0.5: Progressive qd-algorithm used to construct regular C-fractions

```
levinTransform[s_, ω_, ξ_, n_, k_] := Module[{num, den},
  num[m_, j_] := { num[m, j] = s[m]/ω[m]                                              j == 0
                 { num[m, j] = num[m + 1, j - 1] - ((ξ+m) (ξ+m+j-1)^(j-2))/(ξ+m+j)^(j-1) num[m, j - 1]   j ≥ 1 ;
  den[m_, j_] := { den[m, j] = 1/ω[m]                                                 j == 0
                 { den[m, j] = den[m + 1, j - 1] - ((ξ+m) (ξ+m+j-1)^(j-2))/(ξ+m+j)^(j-1) den[m, j - 1]   j ≥ 1 ;
  (* the Levin transform is computed recursively *)
  num[n, k]/den[n, k]
 ]
```

Figure C.0.6: Levin's Transform

```
α = 2; β = 3 / 2; δ = 1; μ = 0;
α1 = α δ; β1 = β δ;
γ1 = √(α1^2 - β1^2); xM = β1/γ1;
pM = F[xM];
pR = e^((α1-β1) xM) (1 - pM);
FBR[x_] := 1 - e^(- (α1-β1) x) pR; (*Right Base CDF*)
QBR[u_] := -1/(α1 - β1) Log[(1 - u)/pR];

u2 = 9 / 10; τR = 10^-10;
z1 = N[QBR[u2], 20];
z2 = N[QBR[1 - τR], 20];
xZ[z_] := (z - 1/2 (z2 + z1))/(1/2 (z2 - z1));
zX[x_] := ((z2 + z1) + (z2 - z1) x)/2 (*Inverse mapping, i.e. z(x)*)

AR[z_?NumericQ] := QRFSpecialized[FBR[z]]
GR[θ_] := AR[zX[Cos[θ]]]

nN = 2^7 - 1;
xData = Table[N[GR[π j/nN]], {j, 0, nN}];
ccFFT = FourierDCT[xData, 1] * Sqrt[2 / nN];
cFFT[n_] := ccFFT[[n + 1]];
truncCS[x_, n_] := cFFT[0]/2 + Sum[cFFT[k] ChebyshevT[k, x], {k, 1, n}]
(*Plot quantile and approximation on right region*)
Plot[{
  InverseCDF[HyperbolicDistribution[α1, β1, 1, 0], u],
  truncCS[xZ[QBR[u]], 20]},
 {u, u2, 1 - τR}
]
```

Figure C.0.7: Computing the Chebyshev Coefficients of $A_R(z)$ with an FFT algorithm.

```
(*Input: a list chebyshev coefficients {c0,...,cn}
  Output: chebyshev pade approximant*)
BuildChebyPadeApprox[chebyC_] := Module[{ω, γ, μ0, q, p, Q, P, S, c, nC, m, n},
  (*to build an approximant of numerator degree m,
  and denominator degree n requires coefficients c0,...,cm+n*)
  nC = Length[chebyC] - 1;
  m = Ceiling[nC / 2]; (*degree of numerator*)
  n = Floor[nC / 2]; (*degree of denomerator*)

  c[k_] := chebyC[[k + 1]]; (*Chebyshev coefficients*)

  (*This recusive relationship to compute the gamma's is fast but
     will only work if γ[s,m,n-1]chebyC[Abs[m-s]]≠ 0 for s=0,...,n-1*)
  ω[m_, n_] := - (Σ_{s=0}^{n-1} γ[s, m + 1, n - 1] c[Abs[m + 1 - s]]) / (Σ_{s=0}^{n-1} γ[s, m, n - 1] c[Abs[m - s]]);
  γ[s_, m_, n_] :=
    { γ[s, m, n] = 1                                          s == 0
    { γ[s, m, n] = 0                                          s == -1 || s == n + 1;
    { γ[s, m, n] = γ[s, m + 1, n - 1] + ω[m, n] γ[s - 1, m, n - 1]   1 ≤ s ≤ n

  (*if the above condition is not met we can
   compute the gamma's by solving a set of linear equations*)
  (*Timing[eqs=Table[Σ_{s=0}^{n} γs N[chebyC[Abs[r-s]]]==0,{r,m+1,m+n}]/.γ0→ 1;
   sol= Solve[eqs,Table[γs,{s,1,n}]]];
  γ[s_,m_,n_]:=γs/.Prepend[sol[[1]],γ0→ 1]*)

  μ0 = N[(1/2 Σ_{i=0}^{n} γ[i, m, n]^2)^-1];

  q[k_] := q[k] = μ0 Σ_{i=0}^{n-k} γ[i, m, n] γ[k + i, m, n];

  p[k_] := p[k] = q[0] c[k] / 2 + 1/2 Σ_{i=1}^{n} q[i] (c[k + i] + c[Abs[k - i]]);

  P[x_] := p[0] / 2 + Σ_{j=1}^{m} p[j] ChebyshevT[j, x];

  Q[x_] := q[0] / 2 + Σ_{j=1}^{n} q[j] ChebyshevT[j, x];

  S[x_] = P[x] / Q[x];
  S[x]

]
```

Figure C.0.8: Mathematica implementation of Sidi's method for constructing Chebyshev-Padé approximants.

```
(*Input;
    interpolationPoints = {z0,...,zn}, list of confluent interpolation points;
    newtonCoefficients  = {c0,...,cn},  list of coefficients appearing in
         Hermite's interpolating polynomial (in Newton's form);
Output;
    The n^th convergent of the interpolating continued fraction.
*)
GeneralizedQD[interpolationPoints_, newtonCoefficients_] :=
  Module[{Z, e, q, a, b, z, c, n, conFrac},
   n = Length[interpolationPoints] - 1;
   z[k_] := interpolationPoints[[k + 1]];
   c[k_] := { newtonCoefficients[[k + 1]]   0 ≤ k ≤ n
            { 0                             k > n ;

   Z[J_, i_] := { z[J + 1] - z[J]                 i == 1 && 0 ≤ J ≤ n - 1
                { z[J + 2 i - 1] - z[J + 2 i - 2]  i ≥ 2 && J ≥ 0 ;
   e[J_, i_] :=
    { e[J, i] = 0                                                    i == 0 && 1 ≤ J ≤ n
    { e[J, i] = -q[J, 1] - q[J + 1, 1] (q[J, 1] Z[J, 1] - 1)         i == 1 && 0 ≤ J ≤ n - 1
    { e[J, i] = -q[J, i] + (Z[J, i] q[J, i] - 1) (e[J + 1, i - 1]    i ≥ 2 && J ≥ 0 ;
                  + q[J + 1, i]) (Z[J, i] e[J + 1, i - 1] - 1)^-1

   q[J_, i_] :=
    { q[J, i] = (Z[J, 1] + c[J]/c[J+1])^-1                                  i == 1 && 0 ≤ J ≤ n - 1
    { q[J, i] = (Z[J, i] - e[J,i-1]/(e[J,i-1]+q[J,i-1]) (q[J+1,i-1]+e[J+1,i-2])/q[J+1,i-1]   i ≥ 2 && J ≥ 0 ;
                  (Z[J,i] e[J+1,i-1]-1)/e[J+1,i-1])^-1

   (*Coefficients of the Theile type continued fraction, as they appear in
      eq. (1.31) of Ch.7, Pade Approximants by Graves-Morris; *)
           { a[0] = c[0]                    k == 0
   a[k_] := { a[k] = -e[0, k/2] (x - z[k - 1])   EvenQ[k] ; (*partial numerators*)
           { a[k] = -q[0, (k+1)/2] (x - z[k - 1])  OddQ[k]
   b[k_] := 1; (*partial denomitors*)

   (*Construct the n^th convergent of the interpolating continued fraction*)
   conFrac = ComposeCF[a, b, n];
   conFrac
  ]
```

Figure C.0.9: Mathematica implementation of the generalized Q.D. algorithm.

```
(* Constructs the nth convergent of a continued fraction
   given partial numerators a0,...,an and partial denominators
   b0,...,bn *)
ComposeCF[a_, b_, n_] := Module[{s, S},

  s[k_, ω_] := a[k] / (b[k] + ω);

  S[k_, ω_] := { s[0, ω]            k == 0
               { S[k - 1, s[k, ω]]  k ≥ 1;

  S[n, 0]

 ]
(* Constructs the nth convergent of a regular C-fraction
   given partial numerators a0,...,an *)
BuildCFraction[partialNumerators_, x_] := Module[{n, a, b},
  n = Length[partialNumerators] - 1;
  a[k_] := partialNumerators[[k + 1]] x;
  b[k_] := 1;
  ComposeCF[a, b, n]
 ]

nthDenominator[partialNumerators_, x_] := Module[{n, a, b, Q},
  n = Length[partialNumerators] - 1;
  a[k_] := partialNumerators[[k + 1]] x;
  b[k_] := 1;

  Q[n_] := Q[n] = { 0                          n == -2
                  { 1                          n == -1;
                  { b[n] Q[n - 1] + a[n] Q[n - 2]  n ≥ 0

  Table[Q[k], {k, 0, n}]
 ]

conFracTail[v_, M_, partialNumerators_, var_] := Module[{},
  BuildCFraction[partialNumerators[[v ;; M]], var]
 ]
```

Figure C.0.10: Some Mathematica code used to construct continued fractions.

Bibliography

Acklam, P. J. (2009), An algorithm for computing the inverse normal cumulative distribution function, http://home.online.no/~pjacklam/notes/invnorm/.

Acton, F. S. (1990), *Numerical Methods That Work*, The Mathematical Association of America.

Adomian, G. (1990), A review of the decomposition method and some recent results for nonlinear equations, *Mathematical and Computer Modelling* **13**(7), 17–43.

Ahrens, J. H. and Kohrt, K. D. (1981), Computer methods for efficient sampling from largely arbitrary statistical distributions, *Computing* **26**(1), 19–31.

Aldrovandi, R. (2001), *Special matrices of mathematical physics: stochastic, circulant, and Bell matrices*, World Scientific.

Apostol, T. M. (1974), *Mathematical analysis*, Addison-Wesley Pub. Co.

Baker, G. A. and Graves-Morris, P. R. (1996), *Padé Approximants*, Cambridge University Press.

Barrio, R., Blesa, F. and Lara, M. (2005), VSVO formulation of the taylor method for the numerical solution of ODEs, *Computers & Mathematics with Applications* **50**(1–2), 93–111.

Beasley, J. D. and Springer, S. G. (1977), Algorithm AS 111: The Percentage Points of the Normal Distribution, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **26**(1), 118–121.

Bender, C. M. and Orszag, S. A. (1978), *Advanced mathematical methods for scientists and engineers: Asymptotic methods and perturbation theory*, Springer.

Bingham, N. H. and Kiesel, R. (2002), Semi-parametric modelling in finance: theoretical foundations, *Quantitative Finance* **2**(4), 241–250.

Bingham, N. and Kiesel, R. (2001), Chapter 1 - Modelling asset returns with hyperbolic distributions, *in* J. Knight and S. Satchell (eds.), *Return Distributions in Finance*, Butterworth-Heinemann, Oxford, pp.1–20.

Bratley, P., Fox, B. L. and Schrage, L. E. (1987), *A guide to simulation*, Springer.

Brent, R. P. and Kung, H. T. (1978), Fast Algorithms for Manipulating Formal Power Series, *J. ACM* **25**(4), 581–595.

Brezinski, C. and Zaglia, M. R. (1991), *Extrapolation Methods: Theory and Practice*, North Holland.

Bruijn, N. G. d. (1981), *Asymptotic methods in analysis*, Courier Dover Publications.

Castelianos, D. and Rosenthal, W. (1993), Rational Chebyshev Approximations of Analytic functions., *The Fibonacci Quarterly* **31**(3), 205–215.

Chang, Y. F. and Corliss, G. (1980), Ratio-Like and Recurrence Relation Tests for Convergence of Series, *IMA Journal of Applied Mathematics* **25**(4), 349–359.

Claessens, G. (1976), A new algorithm for osculatory rational interpolation, *Numerische Mathematik* **27**(1), 77–83.

Clenshaw, C. W. and Norton, H. J. (1963), The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series, *The Computer Journal* **6**(1), 88–92.

Cody, W. J. (1970), A Survey of Practical Rational and Polynomial Approximation of Functions, *SIAM Review* **12**(3), 400–423.

Comtet, L. (1974), *Advanced combinatorics: the art of finite and infinite expansions*, Springer.

Conte, S. D. and Boor, C. D. (1980), *Elementary numerical analysis: an algorithmic approach*, McGraw-Hill.

Corliss, G. and Chang, Y. F. (1982), Solving Ordinary Differential Equations Using Taylor Series, *ACM Trans. Math. Softw.* **8**(2), 114–144.

Cornish, E. A. and Fisher, R. A. (1938), Moments and Cumulants in the Specification of Distributions, *Revue de l'Institut International de Statistique / Review of the International Statistical Institute* **5**(4), 307–320.

Cuyt, A., Petersen, V. and Verdonk, B. (2008), *Handbook of continued fractions for special functions*, Springer.

Dagpunar, J. (1989), An Easily Implemented Generalised Inverse Gaussian Generator, *Communications in Statistics - Simulation and Computation* **18**, 703–710.

Dahlquist, G. and Björck, Å. (2008), *Numerical Methods in Scientific Computing: Volume 1*, Society for Industrial and Applied Mathematics.

David, H. A. (1995), First (?) Occurrence of Common Terms in Mathematical Statistics, *The American Statistician* **49**(2), 121–133.

Derflinger, G., Hörmann, W. and Leydold, J. (2010), Random variate generation by numerical inversion when only the density is known, *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **20**(4), 18.

Derflinger, G., Hörmann, W., Leydold, J. and Sak, H. (2009), Efficient numerical inversion for financial simulations, *in Monte Carlo and Quasi-Monte Carlo Methods 2008*, Springer, p.297–304.

Devroye, L. (1986), *Non-uniform random variate generation*, Springer-Verlag.

Eberlein, E. and Keller, U. (1995), Hyperbolic Distributions in Finance, *Bernoulli* **1**(3), 281–299.

Embrechts, P., Lindskog, F. and Mcneil, A. (2003), Chapter 8 - Modelling dependence with copulas and applications to risk management, *in* S. T. Rachev (ed.), *Handbook of Heavy Tailed Distributions in Finance*, North-Holland, Amsterdam, pp.329–384.

Farnum, N. R. (1991), A Fixed Point Method for Finding Percentage Points, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **40**(1), 123–126.

Fisher, R. A. and Cornish, E. A. (1960), The Percentile Points of Distributions Having Known Cumulants, *Technometrics* **2**(2), 209–225.

Franke, J., Härdle, W. K. and Hafner, C. M. (2011), *Statistics of Financial Markets*, Springer Berlin Heidelberg.

Galanti, S. and Jung, A. R. (1997), Low-Discrepancy Sequences: Monte Carlo Simulation of Option Prices, *The Journal of Derivatives* **5**(1), 63–83.

Gear, C. W. (1971), *Numerical initial value problems in ordinary differential equations*, Prentice-Hall.

Gentle, J. E. (2003), *Random Number Generation and Monte Carlo Methods*, Springer.

Gil-Pelaez, J. (1951), Note on the inversion theorem, *Biometrika* **38**(3-4), 481 –482.

Gilchrist, W. (2000), *Statistical Modelling With Quantile Functions*, CRC Press.

Giles, M. (2010), Approximating the erfinv function, *GPU Gems* **4**.

Glasserman, P. (2004), *Monte Carlo methods in financial engineering*, Springer.

Golder, E. R. and Settle, J. G. (1976), The Box-Muller Method for Generating Pseudo-Random Normal Deviates, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **25**(1), 12–20.

Gottlieb, D. and Orszag, S. A. (1977), *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM.

Graves-Morris, P. (1980), A generalised Q.D. algorithm, *Journal of Computational and Applied Mathematics* **6**(3), 247–249.

Gustavson, F. and Yun, D. (1979), Fast algorithms for rational Hermite approximation and solution of Toeplitz systems, *IEEE Transactions on Circuits and Systems* **26**(9), 750– 755.

Hairer, E., Nørsett, S. P. and Wanner, G. (1993), *Solving ordinary differential equations: Nonstiff problems*, Springer.

Halin, H. (1983), The applicability of Taylor series methods in simulation, *in 15th Summer Computer Simulation Conference, Vancouver, Canada*, Society for Computer Simulation, p.1032–1076.

Hamming, R. W. (1973), *Numerical methods for scientists and engineers*, Courier Dover Publications.

Hänggi, P., Roesel, F. and Trautmann, D. (1980), Evaluation of infinite series by use of continued fraction expansions: A numerical study, *Journal of Computational Physics* **37**(2), 242–258.

Hanson, J. W., Caviness, J. S. and Joseph, C. (1962), Analytic differentiation by computer, *Commun. ACM* **5**(6), 349–355.

He, J.-H. and Wu, X.-H. (2007), Variational iteration method: New development and applications, *Computers & Mathematics with Applications* **54**(7–8), 881–894.

Henrici, P. (1964), *Elements of numerical analysis*, Wiley.

Henrici, P. (1974), *Applied and Computational Complex Analysis: Power series*, Wiley.

Hill, G. W. and Davis, A. W. (1968), Generalized Asymptotic Expansions of Cornish-Fisher Type, *The Annals of Mathematical Statistics* **39**(4), 1264–1273.

Hörmann, W. (1993), The quality of non-uniform random numbers, *in Operations Research Proceedings*, p.329–335.

Hörmann, W., Leydold, J. and Derflinger, G. (2004), *Automatic nonuniform random variate generation*, Springer.

Jäckel, P. (2002), *Monte Carlo methods in finance*, J. Wiley.

Jacod, J. and Protter, P. E. (2004), *Probability essentials*, Springer.

Jaschke, S. R. (2002), The Cornish-Fisher expansion in the context of delta-gamma-normal approximations, *Journal of Risk* **4**, 33–52.

Joe, H. (1997), *Multivariate Models and Multivariate Dependence Concepts*, 1st edition, Chapman and Hall/CRC.

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994), *Continuous univariate distributions*, Wiley & Sons.

Jorba, À. (2005), A Software Package for the Numerical Integration of ODEs by Means of High-Order Taylor Methods, *Experimental Mathematics* **14**(1), 99–117.

Joy, C., Boyle, P. P. and Tan, K. S. (1996), Quasi-Monte Carlo Methods in Numerical Finance, *Management Science* **42**(6), 926–938.

Kennedy, W. J. and Gentle, J. E. (1980), *Statistical computing*, M. Dekker.

Klimko, E. M. (1973), An algorithm for calculating indices in Fàa di Bruno's formula, *BIT Numerical Mathematics* **13**(1), 38–49.

Knuth (1998), *The Art Of Computer Programming, Volume 2: Seminumerical Algorithms, 3/E*, Pearson Education.

Korn, R., Korn, E. and Kroisandt, G. (2010), *Monte Carlo Simulation of Financial and Actuarial Models*, CRC Press.

Lai, Y. (2009), Generating inverse Gaussian random variates by approximation, *Computational Statistics & Data Analysis* **53**(10), 3553–3559.

Lange, K. (2010), *Numerical Analysis for Statisticians*, Springer.

Lara, M., Elipe, A. and Palacios, M. (1999), Automatic programming of recurrent power series, *Mathematics and Computers in Simulation* **49**(4-5), 351–362.

Law, A. M. and Kelton, W. D. (2000), *Simulation modeling and analysis*, McGraw-Hill.

Leobacher, G. and Pillichshammer, F. (2002), A Method for Approximate Inversion of the Hyperbolic CDF, *Computing* **69**(4), 291–303.

Levin, D. (1972), Development of non-linear transformations for improving convergence of sequences, *International Journal of Computer Mathematics* **3**(1-4), 371–388.

Leydold, J. and Hörmann, W. (2011), Generating generalized inverse Gaussian random variates by fast inversion, *Computational Statistics & Data Analysis* **55**(1), 213–217.

Lozier, D. W. and Olver, F. W. J. (1994), Numerical Evaluation of Special Functions, *In W. Gautshi (ed.), AMS Proceeding of Symposia in Applied Mathematics* **48**, 79—125.

L'Ecuyer, P. (2012), Random Number Generation, *in* J. E. Gentle, W. K. Härdle and Y. Mori (eds.), *Handbook of Computational Statistics*, Springer Handbooks of Computational Statistics, Springer Berlin Heidelberg, pp.35–71.

Lukacs, E. (1970), *Characteristic functions*, Griffin, London.

Madan, D. B. and Seneta, E. (1990), The Variance Gamma (V.G.) Model for Share Market Returns, *The Journal of Business* **63**(4), 511–524.

Maehly, H. J. (1963), Methods for Fitting Rational Approximations, Parts II and III, *J. ACM* **10**(3), 257–277.

McNeil, A. J., Frey, R. and Embrechts, P. (2005), *Quantitative risk management: concepts, techniques and tools*, Princeton University Press.

Moore, R. E., Kearfott, R. B. and Cloud, M. J. (2009), *Introduction to interval analysis*, SIAM.

Moro, B. (1995), The full Monte, *Risk Magazine* **8**(2), 202–232.

Munir, A. (2012), QM3 Code Repository, http://code.google.com/p/quantile-mechanics/.

Nelsen, R. B. (2006), *An introduction to copulas*, Springer Verlag.

Odeh, R. E. and Evans, J. O. (1974), The percentage points of the normal distribution, *Applied Statistics* **23**(1), 96–97.

Ökten, G. and Göncü, A. (2011), Generating low-discrepancy sequences from the normal distribution: Box-Muller or inverse transform?, *Mathematical and Computer Modelling* **53**(5-6), 1268–1281.

Olver, F. W. J., Lozier, D. W. and Boisvert, R. F. (2010), *NIST Handbook of Mathematical Functions*, Cambridge University Press.

Olver, P. J. (2000), *Applications of Lie groups to differential equations*, Springer.

Paolella, M. S. (2007), *Intermediate probability: a computational approach*, Wiley-Interscience.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007), *Numerical recipes: the art of scientific computing*, Cambridge University Press.

Rall, L. B. (1981), *Automatic differentiation: techniques and applications*, Springer-Verlag.

Ralston, A. (1963), On Economization of Rational Functions, *Journal of the ACM* **10**(3), 278–282.

Ralston, A. and Rabinowitz, P. (2001), *A first course in numerical analysis*, Courier Dover Publications.

Ripley, B. D. (1987), *Stochastic simulation*, J. Wiley.

Roy, D., Bhattacharya, R. and Bhowmick, S. (1996), Rational approximants using Levin-Weniger transforms, *Computer Physics Communications* **93**(2-3), 159–178.

Salvy, B. (1994), Fast Computation of Some Asymptotic Functional Inverses, *Journal of Symbolic Computation* **17**(3), 227–236.

Samorodnitsky, G. and Taqqu, M. S. (1994), *Stable non-Gaussian random processes: stochastic models with infinite variance*, Chapman & Hall.

Schmidt, T. (2006), Coping with copulas, *in Copulas from theory to applications in finance*, 1st edition, Risk Books, pp.3–34.

Seneta, E. (2004), Fitting the Variance-Gamma Model to Financial Data, *Journal of Applied Probability* **41**, 177–187.

Shaw, W. T. and McCabe, J. (2009), Monte Carlo sampling given a Characteristic Function: Quantile Mechanics in Momentum Space, *arXiv:0903.1592* .

Shaw, W. T., Luu, T. and Brickman, N. (2011), Quantile Mechanics II: Changes of Variables in Monte Carlo methods and GPU-Optimized Normal Quantiles, *arXiv:0901.0638* .

Shephard, N. (1991), From Characteristic Function to Distribution Function: A Simple Framework for the Theory, *Econometric Theory* **7**(04), 519–529.

Sidi, A. (1975), Computation of the Chebyshev-Padé table, *Journal of Computational and Applied Mathematics* **1**(2), 69–71.

Sidi, A. (2002), *Practical Extrapolation Methods: Theory and Applications*, 1st edition, Cambridge University Press.

Smith, D. A. and Ford, W. F. (1982), Numerical Comparisons of Nonlinear Convergence Accelerators, *Mathematics of Computation* **38**(158), 481–499.

Steinbrecher, G. and Shaw, W. (2008), Quantile mechanics, *European Journal of Applied Mathematics* **19**, 87–112.

Stoyanov, S. and Racheva-Iotova, B. (2004), Numerical methods for stable modeling in financial risk management, *Handbook of Computational and Numerical Methods, S. Rachev (ed). Birkhäuser: Boston* .

Takemura, A. (1983), Orthogonal Expansion of Quantile Function and Components of the Shapiro- Francia Statistic, Technical Report, Defense Technical Information Center.

Tenenbaum, M. and Pollard, H. (1985), *Ordinary differential equations: an elementary textbook for students of mathematics, engineering, and the sciences*, Courier Dover Publications.

Tezuka, S. (1991), Neave effect also occurs with Tausworthe sequences, *in Proceedings of the 23rd conference on Winter simulation*, WSC '91, IEEE Computer Society, Washington, DC, USA, p.1030–1034.

Thacher, H. C. (1964), Conversion of a power to a series of Chebyshev polynomials, *Commun. ACM* **7**(3), 181–182.

Ulrich, G. and Watson, L. T. (1987), A Method for Computer Generation of Variates from Arbitrary Continuous Distributions, *SIAM Journal on Scientific and Statistical Computing* **8**, 185–197.

Weniger, E. J. (2001), Nonlinear Sequence Transformations: Computational Tools for the Acceleration of Convergence and the Summation of Divergent Series, *arXiv:math/0107080* .

Weron, R. (2004), Computationally intensive Value at Risk calculations, *Handbook of Computational Statistics, Springer, Berlin* p.911–950.

Whittaker, E. T. and Watson, G. N. (1927), *A course of modern analysis: an introduction to the general theory of infinite processes and of analytic functions, with an account of the principal transcendental functions*, Cambridge Univ Pr.

Wolfram-Research (2010), Mathematica 8.0.

Wynn, P. (1960), The Rational Approximation of Functions which are Formally Defined by a Power Series Expansion, *Mathematics of Computation* **14**(70), 147–186.

Zolotarev, V. M. (1986), *One-Dimensional Stable Distributions*, American Mathematical Society.