MASTER'S THESIS

Interuniversity Master in Statistics and Operations Research UPC-UB

Title: Quantile computation via ordinary differential equations. The non-central χ^2 case and its financial application.

Author: Samuel Medina Leva

Advisor: Luis Ortiz-Gracia

Department: Econometrics, Statistics and Applied Economics

University: Universitat de Barcelona

Academic year: 2019 - 2020

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA
Facultat d'Economia i Empresa

Quantile computation via ordinary differential equations

The non-central χ^2 case and its financial application

Master's Thesis
Master in Statistics and Operations Research
2020

Samuel Medina Leva

Directed by Dr. Luis Ortiz-Gracia

Department of Econometrics, Statistics and Applied Economics

Abstract

Mathematics and statistics are widely used in financial problems. In particular, quantile computation methods are an interesting field of study in order to obtain new and better ways to measure risk with Value-at-Risk (VaR) and Expected Shortfall (ES). We will be interested in considering the quantile computation with ordinary differential equations (ODEs), and specifically with numerical methods as Euler and Runge-Kutta which will let us find numerical solutions to these ODEs, and consequently numerical quantiles. The studied problem will be centered in the delta-gamma framework, where the quantile computation of non-central χ^2 distribution will be needed. Thus, implementations of numerical methods and several numerical experiments will be done, comparing the results with other quantile computation methods and understanding the possibilities which ODEs offer us to solve the quantile computation problem.

Keywords: Quantile computation, Ordinary Differential Equations, Non-central χ^2 distribution, Euler Method, Runge-Kutta Method, Value-at-Risk, Expected Shortfall.

Als meus pares,
lluitadors incansables de la dignitat.
Per l'estima incondicional.

Als ningú,
que ho són tot.

Acknowledgements

I would like to take this opportunity to acknowledge everyone who had a relevant paper throughout the preparation and writing of this Master's Thesis.

I am really grateful to Dr. Luis Ortiz-Gracia, my Director, for helping me in everything I needed. He taught me, shared with me a little part of his knowledge and his advice was always clarifying. Moreover, I am writing these lines while we are living an extremely difficult situation around the world due to COVID-19. Despite everything, we were able to find all possible solutions and ways to continue working in a research in which we had trust. I hope we could work together in a near future.

I am thankful to all the teachers and professors I had during my entire life, since I am what I am thanks to them. They taught me the power of knowledge and thought. Specially, I would like to remember Joan del Castillo, who was my Director in the Mathematics Degree Final Project. He was the first person who showed me the wide applications of Mathematics in finance and made me take an interest in this issue.

Last, but by not means least, I would like to thank my parents for helping and supporting me, Carlos for simply being there, Sara because of her grammatical advice, and all my friends, who are my family.

Contents

1 Introduction

Risk management has been described as "one of the most important innovations of the 20th century" by Steinherr [14]. Actually, during last decades risk management and measurement have become in one of the most important issues in finance, in a context where the market has been transformed for instance because of derivative products. Thus it is imperative that mathematicians take a serious interest in derivatives and the risks they generate. Moreover, quantile computation has become a central topic in this financial risk measurement framework. Some regulations have been established (Basel II) and VaR and ES have become common and known risk measures [10].

Throughout the present work, understanding the relevance of quantile computation within a financial context will be a main goal. Concerning that quantile computation, we would like to present a method which uses ordinary differential equations (ODEs), and in particular it solves numerically these ODEs. Therefore, connection between quantile computation and ODEs will be explained, just as some of most common numerical methods for ODEs.

It is thought that the relevance of our work is given by the novelty of the presented method. It is widely known the relation between ODEs and quantiles, and for instance the reader could find some applications in quantile computation using power series solutions [13]. However, to the best of our knowledge, this is the first time that the quantile function is computed by numerically solving the associated ODE. That is why one of the main aims of this work will be presenting innovative results, which could help in future research about this issue.

First, in Section 2 financial concepts will be developed in order to do a first approach to the financial motivation of our problem. In this section, concepts like risk measurement will be introduced and, more specifically, we will focus on VaR and ES, so as to understand the relevance of quantile computation in finance.

Then, some probabilistic and statistical definitions will be done, in order to understand the quantile computation problem, and we will describe the state of the art in Section 3. Thus, some analytical approximations and root-finding algorithms for quantile computation will be presented. Afterwards, in Section 4 quantile computation with ODEs will be introduced so as to understand the main purpose of the present work to deal with quantile computation, and experiments with well-known distributions will be presented. Next, numerical methods for solving ODEs should be introduced, since our intention will be solving numerically the presented ODEs. In particular, Euler and Runge-Kutta methods will be explained in Section 5.

In Section 6, we will focus on the case of non-central χ^2 distribution (specifically with one degree of freedom), and the relation between this probability distribution and delta-gamma framework will be presented. The application of quantile computation in a specific financial problem will be showed, hence the importance of working with this distribution.

Last, the results of our numerical experiments will be presented in Section 7. In a first part, experiments with well-known distributions will be done (Normal and Student), so as to study the ODE results performance. Then, we will study the non-central χ^2 case, considering the results for different non-centrality parameters. Euler and Runge-Kutta methods will be compared and the error terms will be evaluated. Besides, the results will be compared with Python ones, and computation time will also be taken into account. From these numerical results, we will discuss our main conclusions and the most important lines for future research will be mentioned. Furthermore, it will be interesting to show in Appendices the Python code used to obtain the numerical results and to present some tables with quantile results.

2 Financial motivation for quantile computation

We should discuss essential concepts in quantitative risk management, in order to understand the problem we should deal with. We will begin by introducing a probabilistic framework for modelling financial risk and we will give formal definitions for notions such as risk, loss and risk factors.

A central issue in modern risk management is the measurement of risk, because we need to quantify the risk that arises in many different contexts. For instance, a regulator measures the risk exposure of a financial institution in order to determine the amount of capital that institution has to hold as a buffer against unexpected losses. Similarly, the clearing house of an exchange needs to set margin requirements for investors trading on that exchange. We will give an overview of the existing approaches to measuring risk and discuss their strengths and weaknesses. Particularly, we will explain VaR and the related notion of ES.

2.1 General definitions

In banking, the best known type of risk is probably market risk, the risk of a change in the value of a financial position due to changes in the value of the underlying components on which that position depends, such as stock and bond prices, exchange rates, commodity prices, etc. The next important category is credit risk, the risk of not receiving promised repayments on outstanding investments such as loans and bonds, because of the "default" of the borrower. A further risk category that has received a lot of recent attention is operational risk, the risk of losses resulting from inadequate or failed internal processes, people and systems, or from external events.

First step will be to introduce general definitions about loss distributions and risk factors. We represent the uncertainty about future states of the world by a probability space $(\Omega, \mathcal{F}, P)$, which is the domain of all random variables (rvs) we introduce below.

Definition 1. ***Loss.*** *Consider a given portfolio such as a collection of stocks or bonds, a book of derivatives, a collection of risky loans or even a financial institution's overall position in risky assets. We denote the value of this portfolio at time s by $V(s)$ and assume that the rv $V(s)$ is observable at time s. For a given time horizon Δ, such as 1 or 10 days, the loss of the portfolio over the period $[s, s+\Delta]$ is given by*

$$L_{[s,s+\Delta]} = -(V(s+\Delta) - V(s)). \tag{1}$$

While $L_{[s,s+\Delta]}$ is assumed to be observable at time $s+\Delta$, it is tipically random from the viewpoint of time s, and its distribution is termed the loss distribution.

It will be interesting to remark that practitioners in risk management are often concerned with the so-called profit-and-loss (PL) distribution. This is the distribution of the change in value $V(s+\Delta) - V(s)$, i.e. of the rv $-L_{[s,s+\Delta]}$. However, in risk management we are mainly concerned with the probability of large losses and hence with the upper tail of the loss distribution.

We will usually consider a fixed horizon Δ. In that case it will be convenient to measure time in units of Δ and to introduce a time series notation, where we move from a generic process $Y(s)$ to the time series $(Y_t)_{t\in\mathbb{N}}$ with $Y_t = Y(t\Delta)$. Using this notation the loss will be written as

$$L_{t+1} = L_{[t\Delta,(t+1)\Delta]} = -(V_{t+1} - V_t). \tag{2}$$

For instance, in market risk management we oftenwork with financial models where the calendar time s is measured in years and interest rates and volatilities are quoted on an annualized basis. If we are interested in daily losses we set $\Delta = 1/365$ or $\Delta = 1/250$; the latter convention is mainly used in markets for equity derivatives since there are approximately 250 trading days per year. The rvs V_t and V_{t+1} then represent the portfolio value on days t and $t+1$, respectively, and L_{t+1} is the loss from day t to day $t+1$.

Definition 2. ***Risk factors.*** *Following standard risk-management practice the value V_t is modelled as a function of time and a d-dimensional random vector $\mathbf{Z}_t = (Z_{t,1}, \ldots, Z_{t,d})'$ of risk factors, i.e. we will have the representation*

$$V_t = f(t, \mathbf{Z}_t) \tag{3}$$

for some mesurable function $f : \mathbb{R}_+ \to \mathbb{R}^d$. Risk factors are usually assumed to be observable so that $\mathbf{Z}_t$ is known at time t. The choice of the risk factors and of f is of course a modelling issue and depends on the portfolio at hand and on the desired level of precision.

2.2 Risk measurement

Now we will give an overview of existing approaches to measuring risk in financial institutions, and we will focus on VaR and ES cases in order to show the importance of quantile computation in risk measurement. We will also discuss strengths and weaknesses of these approaches and we will we will show practical aspects and the theoretical properties of the risk measures (issues such as subadditivity and coherence).

Existing approaches to measuring the risk of a financial position can be grouped into four different categories: the notional-amount approach; factor-sensitivity measures; risk measures based on scenarios; and risk measures based on the loss distribution. The last one will be the one we are interested in, because it includes VaR and ES. It is of course problematic to rely on any one particular statistic to summarize the risk contained in a distribution. However, the view that the loss distribution as a whole gives an accurate picture of the risk in a portfolio has much to commend it:

- Losses are the central object of interest in risk management and so it is natural to base a measure of risk on their distribution.
- The concept of a loss distribution makes sense on all levels of aggregation from a portfolio consisting of a single instrument to the overall position of a financial institution.
- If estimated properly, the loss distribution reflects netting and diversification effects.
- Loss distributions can be compared across portfolios.

There are two major problems when working with loss distributions. First, any estimate of the loss distribution is based on past data. If the laws governing financial markets change, these past data are of limited use in predicting future risk. The second, related problem is practical. Even in a stationary environment it is difficult to estimate the loss distribution accurately, particularly for large portfolios, and many seemingly sophisticated risk-management systems are based on relatively crude statistical models for the loss distribution (incorporating, for example, untenable assumptions of normality).

However, this is not an argument against using loss distributions. Rather, it calls for improvements in the way loss distributions are estimated and, of course, for prudence in the practical application of risk-management models based on estimated loss distributions. In particular, risk measures based on the loss distribution should be complemented by information from hypothetical scenarios. Moreover, forward-looking information reflecting the expectations of market participants, such as implied volatilities, should be used in conjunction with statistical estimates (which are necessarily based on past information) in calibrating models of the loss distribution.

It will also be interesting to refer to the concept of coherent risk measure, thus we will need to show some axioms that any so-called coherent risk measure should satisfy. In order to introduce the axioms of coherence we have to give a formal definition of risk measures [10].

Definition 3. ***Risk measure.*** *Fix some probability space $(\Omega, \mathcal{F}, P)$ and a time horizon Δ. Denote by $L^0(\Omega, \mathcal{F}, P)$ the set of all rvs on $(\Omega, \mathcal{F})$, which are almost surely finite. Financial risks are represented by a set $\mathcal{M} \subset L^0(\Omega, \mathcal{F}, P)$ of rvs, which we interpret as portfolio losses over some time horizon Δ. The time horizon is left unspecified and will only enter when specific problems are considered. We often assume that $\mathcal{M}$ is a convex cone, i.e. that $L_1 \in \mathcal{M}$ and $L_2 \in \mathcal{M}$ implies that $L_1 + L_2 \in \mathcal{M}$ and $\lambda L_1 \in \mathcal{M}$ for every $\lambda > 0$. Risk measures are real-valued functions $\varrho : \mathcal{M} \to \mathbb{R}$ defined on such cones of rvs, satisfying certain properties.*

We interpret $\varrho(L)$ as the amount of capital that should be added to a position with loss given by L, so that the position becomes acceptable to an external or internal risk controller. Positions with $\varrho(L) \leq 0$ are acceptable without injection of capital; if $\varrho(L) < 0$, capital may even be withdrawn. Note that in order to

simplify the presentation we set interest rates equal to zero so that there is no discounting. Now we can introduce the axioms that a risk measure $\varrho : \mathcal{M} \to \mathbb{R}$ on a convex cone $\mathcal{M}$ should satisfy in order to be called coherent.

Axiom 1. ***Translation invariance.*** *For all $L \in \mathcal{M}$ and every $l \in \mathbb{R}$ we have $\varrho(L + l) = \varrho(L) + l$.*

Axiom 1 states that by adding or subtracting a deterministic quantity l to a position leading to the loss L we alter our capital requirements by exactly that amount. The axiom is in fact necessary for the risk-capital interpretation of to make sense. Consider a position with loss L and $\varrho(L) > 0$. Adding the amount of capital $\varrho(L)$ to the position leads to the adjusted loss $\tilde{L} = L - \varrho(L)$, with $\varrho(\tilde{L}) = \varrho(L) - \varrho(L) = 0$, so that the position $\tilde{L}$ is acceptable without further injection of capital.

Axiom 2. ***Subadditivity.*** *For all $L_1, L_2 \in \mathcal{M}$ we have $\varrho(L_1 + L_2) \leq \varrho(L_1) + \varrho(L_2)$.*

Subadditivity reflects the idea that risk can be reduced by diversification, a time-honoured principle in finance and economics. Besides, if a regulator uses a non-subadditive risk measure in determining the regulatory capital for a financial institution, that institution has an incentive to legally break up into various subsidiaries in order to reduce its regulatory capital requirements. Similarly, if the risk measure used by an organized exchange in determining the margin requirements of investors is non-subadditive, an investor could reduce the margin he has to pay by opening a different account for every position in his portfolio.

Furthermore, subadditivity makes decentralization of risk-management systems possible. Consider as an example two trading desks with positions leading to losses L_1 and L_2. Imagine that a risk manager wants to ensure that $\varrho(L)$, the risk of the overall loss $L = L_1 + L_2$, is smaller than some number M. If he uses a risk measure ϱ, which is subadditive, he may simply choose bounds M_1 and M_2 such that $M_1 + M_2 \leq M$ and impose on each of the desks the constraint that $\varrho(L_i) \leq M_i$; subadditivity of ϱ then ensures automatically that $\varrho(L) \leq M_1 + M_2 \leq M$.

Axiom 3. ***Positive homogeneity.*** *For all $L \in \mathcal{M}$ and every $\lambda > 0$ we have $\varrho(\lambda L) = \lambda \varrho(L)$.*

This axiom is easily justified if we assume that Axiom 2 holds. Subadditivity implies that

$$\varrho(nL) = \varrho(L + \cdots + L) \leq n\varrho(L),$$

for $n \in \mathbb{N}$. Since there is no netting or diversification between the losses in this portfolio, it is natural to require that equality should hold in the expression above, which leads to positive homogeneity. Note that subadditivity and positive homogeneity imply that the risk measure ϱ is convex on $\mathcal{M}$.

Axiom 4. ***Monotonicity.*** *For $L_1, L_2 \in \mathcal{M}$ such that $L_1 \leq L_2$ almost surely we have $\varrho(L_1) \leq \varrho(L_2)$.*

From an economic viewpoint this axiom is obvious: positions that lead to higher losses in every state of the world require more risk capital.

For a risk measure satisfying Axioms 2 and 3, the monotonicity axiom is equivalent to the requirement that $\varrho(L) \leq 0$ for all $L \leq 0$. To see this, observe that Axiom 4 implies that if $L \leq 0$, then $\varrho(L) \leq \varrho(0) = 0$; the latter equality follows from Axiom 3 since $\varrho(0) = \varrho(\lambda 0) = \lambda \varrho(0)$ for all $\lambda > 0$. Conversely, if $L_1 \leq L_2$ and we assume that $\varrho(L_1 - L_2) \leq 0$, then $\varrho(L_1) = \varrho(L_1 - L_2 + L_2) \leq \varrho(L_1 - L_2) + \varrho(L_2)$ by Axiom 2, which implies that $\varrho(L_1) \leq \varrho(L_2)$.

Definition 4. ***Coherent risk measure.*** *A risk measure ϱ whose domain includes the convex cone $\mathcal{M}$ is called coherent (on $\mathcal{M}$) if it satisfies Axioms 1-4.*

Note that the domain is an integral part of the definition of a coherent risk measure. We will often encounter functionals on $L^0(\Omega, \mathcal{F}, P)$, which are coherent only if restricted to a sufficiently small convex cone $\mathcal{M}$ [10].

2.2.1 Value-at-Risk and Expected Shortfall

VaR is probably the most widely used risk measure in financial institutions hence it merits an extensive discussion. We will introduce VaR and discuss practical issues surrounding its use. Also, we will examine

VaR from the viewpoint of coherent risk measures and highlight certain theoretical deficiencies.

Consider some portfolio of risky assets and a fixed time horizon Δ, and denote by $F_L(l) = P(L \leq l)$ the distribution function of the corresponding loss distribution. We do not distinguish between L and L^{Δ}; rather we assume that the choice has been made at the outset of the analysis and that F_L represents the distribution of interest. We want to define a statistic based on F_L which measures the severity of the risk of holding our portfolio over the time period Δ. An obvious candidate is the maximum possible loss, given by $\inf\{l \in \mathbb{R} : F_L(l) = 1\}$, a risk measure important in reinsurance. However, in most models of interest the support of F_L is unbounded so that the maximum loss is simply infinity. Moreover, by using the maximum loss we neglect any probability information in F_L. VaR is a straightforward extension of maximum loss, which takes these criticisms into account. The idea is simply to replace "maximum loss" by "maximum loss which is not exceeded with a given high probability", the so-called confidence level.

Definition 5. ***Value-at-Risk (VaR).*** *Given some confidence level $\alpha \in (0,1)$. The VaR of our portfolio at the confidence level α is given by the smallest number l such that the probability that the loss L exceeds l is no larger than $(1-\alpha)$. Formally,*

$$\text{VaR}_\alpha = \inf\{l \in \mathbb{R} : P(L > l) \leq 1 - \alpha\} = \inf\{l \in \mathbb{R} : F_L(l) \geq \alpha\}. \tag{4}$$

In probabilistic terms, VaR is thus simply a quantile of the loss distribution, so we can understand the importance of quantile computation. Typical values for α are $\alpha = 0.95$ or $\alpha = 0.99$; in market risk management the time horizon Δ is usually 1 or 10 days, in credit risk management and operational risk management Δ is usually one year. Note that by its very definition the VaR at confidence level α does not give any information about the severity of losses which occur with a probability less than $1 - \alpha$. This is clearly a drawback of VaR as a riskmeasure.

It is immediately seen from the representation of VaR as a quantile of the loss distribution that it is translation invariant, positive homogeneous and monotone on $L^0(\Omega, \mathcal{F}, P)$. However, the subadditivity property (Axiom 2) fails to hold for VaR in general, so VaR is not a coherent risk measure.

We should also consider ES, which is closely related to VaR and is now preferred to VaR by many risk managers in practice.

Definition 6. ***Expected Shortfall (ES).*** *For a loss L with $E(|L|) < \infty$ and distribution function F_L the ES at confidence level $\alpha \in (0,1)$ is defined as*

$$ES_\alpha = \frac{1}{1-\alpha} \int_\alpha^1 q_u(F_L) du, \tag{5}$$

where $q_u(F_L) = F_L^{-1}$ is the quantile function of F_L.

ES is thus related to VaR by

$$Es_\alpha = \frac{1}{1-\alpha} \int_\alpha^1 \text{VaR}_u(L) du. \tag{6}$$

Instead of fixing a particular confidence level α we average VaR over all levels $u \geq \alpha$ and thus "look further into the tail" of the loss distribution. Obviously ES_α depends only on the distribution of L and obviously $\text{ES}_\alpha \geq \text{VaR}_\alpha$. We see in the figure 1 a simple illustration of an ES value and its relationship to VaR.

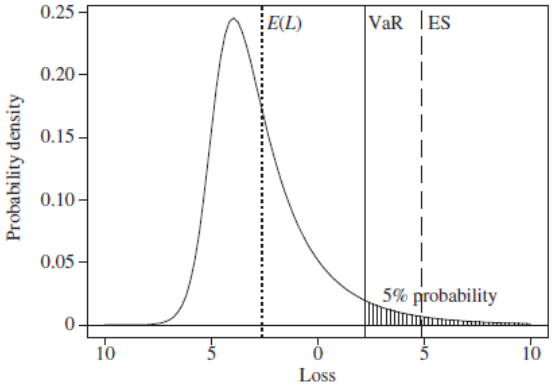


Figure 1: Comparison between VaR and ES for a loss distribution [10].

For continuous loss distributions an even more intuitive expression can be derived which shows that ES can be interpreted as the expected loss that is incurred in the event that VaR is exceeded.

Lemma 1. *For an integrable loss L with continuous distribution function F_L and any $\alpha \in (0,1)$ we have*

$$\mathrm{ES}_\alpha = \frac{\mathbb{E}(L; L \geq q(\alpha)(L))}{1-\alpha} = \mathbb{E}(L | L \geq \mathrm{VaR}_\alpha), \tag{7}$$

where we have used the notation $\mathbb{E}(X;A) = \mathbb{E}(XI_A)$ for a generic integrable rv X and a generic set $A \in \mathcal{F}$.

Also we can give a kind of law of large numbers for ES in terms of order statistics.

Lemma 2. *For a sequence $(L_i)_{i\in\mathbb{N}}$ of iid rvs with distribution function F_L we have*

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{[n(1-\alpha)]} L_{i,n}}{[n(1-\alpha)]} = \mathrm{ES}_\alpha \qquad a.s.,$$

where $L_{1,n} \dots L_{n,n}$ are the order statistics of $L_1, \dots, L_n$ and where $[n(1-\alpha)]$ denotes the largest integer not exceeding $n(1-\alpha)$.

In other words, ES at confidence level α can be thought of as the limiting average of the $[n(1-\alpha)]$ upper order statistics from a sample of size n from the loss distribution. This representation suggests an obviousway of estimating ES in the situation when we have large samples and $[n(1-\alpha)]$ is a relatively large number. Since ES_α can be thought of as an average over all losses that are greater than or equal to VaR_α, it is sensitive to the severity of losses exceeding VaR_α.

Besides, we should comment the coherence of ES, as we have done with VaR. A proof of its coherence can be based on the Lemma above, which gives a representation of ES as the limit of the averages of upper order statistics.

Proposition 1. *ES is a coherent risk mesure.*

The proof of this proposition is seen in [10].

In conclusion, we have defined two risk measures which are widely used in financial institutions, VaR (which is not coherent) and ES (which is coherent). From these definitions, it is seen that we will need to compute the quantiles, hence the relevance of our work.

3 Classical methods for quantile computation

In this section we will present some formal definitions about statistics and particularly about quantile functions and some of the most known classical methods to compute quantiles. When we work with continuous random variables we know two really common ways of definition of the distribution: distribution function and density function. Thus, first step will be to give formal definitions of these concepts [4] [3].

Definition 7. ***Distribution function.*** *Let X be a continuous random variable, its distribution function, denoted by F(x), is defined as the probability of a variable X being less than or equal to some given value, x. Thus, $F(x)$ is defined by*

$$F(x) = P(X \leq x).$$

Definition 8. ***Density function.*** *Let X be a continuous random variable with $F_X(x)$ its distribution function, and let $F_X(x)$ be differentiable. The density function of X is defined by*

$$f(x) = \frac{dF}{dx}(x).$$

Now, we should introduce the quantile function, which provides a third way of defining a distribution.

Definition 9. ***Quantile function.*** *The quantile function, denoted by $q(p)$, provides a way of defining a distribution. Formally, we have*

$$x_p = \textit{the value of } x \textit{ for which } P(X \leq x_p) = p.$$

The value x_p is called the p-quantile of the population. The function $x_p = q(p)$ expresses the p-quantile as a function of p and is called the quantile function.

The definitions of the quantile function and the distribution function can be written for any pair of values (x, p) as $x = q(p)$ and $p = F(x)$. These functions are thus simple inverses of each other, provided that they are both continuous increasing functions. Thus we can also write

$$q(p) = F^{-1}(p) \text{ and } F(x) = q^{-1}(x). \tag{8}$$

Then, we will explore the classical methods for quantile computation. In the literature, we find two main ways of deal with this issue: analytical approximations and root-finding algorithms.

3.1 Analytical approximations

It is well known that for an specific given probability distribution there are several analytical approximations of its distribution function and consequently its quantile function. For instance, the literature contains a vast collection of approximate functions for the normal distribution, which are very complicated, not very accurate, or valid for only a limited range. In [15] we can find an enhanced one using the logistic approximation, with a simpler functional form and higher accuracy.

In our particular case, we will explain in following sections the relevance of non-central χ^2. There are a handful of established approximations for this distribution and it will be interesting to present two of them: Sankaran's approximation and Pearson's one, with emphasis on when they can break down.

Sankaran gives an analytic approximation for the non-central χ^2 distribution function:

$$F_{\nu,\xi}(x) \approx \Phi\left(\frac{\frac{x}{\nu+\xi} - \mu}{\sigma}\right), \tag{9}$$

where Φ is the standard normal cumulative distribution function and

$$\begin{aligned} h &= 1 - \frac{2}{3}\frac{(\nu+\xi)(\nu+3\xi)}{(\nu+2\xi)^2} \\ p &= \frac{\nu+2\xi}{(\nu+\xi)^2} \\ m &= (h-1)(1-3h) \\ \mu &= 1 + hp(h-1-(1-h/2)mp) \\ \sigma &= h\sqrt{2p}(1+mp/2). \end{aligned} \tag{10}$$

Sankaran's distribution function is readily invertible, so we will obtain an analytical approximation to the quantile function

$$q_{\nu,\xi}(\alpha) \approx (\nu+\xi)(\Phi^{-1}(\alpha)\sigma+\mu)^{1/h}. \tag{11}$$

When $\Phi^{-1}(\alpha)\sigma + \mu < 0$ (small α) this approximation will give a complex-value result, so this will be a problem when applying implementations based on this approach or using slight modifications of this.

Pearson gives an approximation to $q_{\nu,\xi}$ of the form $b + c\chi^2_f$, where

$$\begin{aligned} b &= -\frac{\xi^2}{\nu+3\xi} \\ c &= \frac{\nu+3\xi}{\nu+2\xi} \\ f &= \frac{(\nu+2\xi)^3}{(\nu+3\xi)^2} = \frac{\nu+2\xi}{c^2}, \end{aligned} \tag{12}$$

and χ^2_f is the quantile of the central χ^2 with f degrees of freedom. In that case, for small enough α the approximation will give a result less than zero. Therefore, both approximations typically fail for small enough inputs, giving negative or complex values [9].

3.2 Root-finding algorithms

We consider a probability distribution with $F(x)$ its distribution function and we know, as explained before in (8), that quantile function and distribution function are inverses of each other. Thus for an specific $p = \alpha$ we will have

$$F(q) = \alpha. \tag{13}$$

Considering that, the quantile function problem has been reduced to a root-finding problem, where we will define a function $G(q) = F(q) - \alpha$ and we will find the root of $G(q)$, which is any point $x = x_r$ for which the function $G(x_r) = 0$. We know a lot of algorithms in order to tackle this problem, as the bisection method, which has been frequently used in quantile computation. In order to understand that numerical method it will be interesting to introduce the following theorem [6].

Theorem 1. *An equation $f(x) = 0$, where $f(x)$ is a real continuous function, has at least one root between x_L and x_U if $f(x_L)f(x_U) < 0$.*

The bisection method begins by setting an upper and lower bound before and after the root, then at each iteration, it reduces the interval between these bounds to zoom in on the root. The algorithm for the bisection method is a simple three-step iterative process: setup for the iterations, iterate, and terminate the iterations. The steps are explained below:

1. Set the initial lower bound x_L and upper bound x_U around the root of the function $G(q)$. Since the root is a zero of the equation, the two bounds must be on either side of a zero-crossing; in other words, they must be points where the equation evaluates to values of opposite signs, as a result from the Theorem 1.

2. Once points on either side of the zero-crossing have been selected, the bisection method iteratively tightens them by picking the point exactly in-between the two bounds:

$$x_i = \frac{x_L + x_U}{2}$$

The function is then evaluated at that point. In some rare cases, the function will evaluate to zero exactly, in which case the point x_i is the root and the algorithm can terminate. More generally however, the function will evaluate to a positive or negative value, and the middle point will be on one or the other side of the root. The middle point replaces the bound on the same side of the root as itself and becomes the new bound on that side of the root. The interval is thus reduced by half at each iteration, and the root remains bracketed between a lower and upper bound, $[x_i, x_U]$ if $G(x_i)G(x_U) > 0$ and $[x_L, x_i]$ if $G(x_i)G(x_U) > 0$.

3. The iterative process continues until a halting condition is reached. One halting condition mentioned in the previous step, albeit an unlikely one, is that a middle point x_i is found to be exactly the root. More usually, the algorithm will reach a preset maximum number of iterations (failure condition) or some error metric, such as the interval between the two brackets or the relative error on the point x_i, will become lower than some preset error value (success condition) [7].

Following this method it is clearly seen that we will have a numerical approach to the quantile value

$$q(\alpha) = x_i.$$

There are many other root-finding algorithms which could be used in quantile computation, for instance the Newton-Raphson method is possibly the most popular one available. The problem will be that in Newton-Raphson the root is not bracketed, thus we could have the possibility of obtain negative values in iterations and convergence is not guaranteed [6]. Other options would be the secant method or the Brent's method (which combines the bisection method, the secant method and inverse quadratic interpolation) [2].

4 Quantile functions with ordinary differential equations

Once we have shown most of classical methods for quantile computation, we will devote this section to introduce the quantile calculation with ordinary differential equations (ODEs), since we have seen that knowing the quantiles of some distributions will be interesting in finance when we talk about VaR and ES. We know that we will be able to obtain the quantile function q from the density function by solving an ODE. Basically, we can express the derivative of the quantile function q as a the reciprocal of the usual density function expressed in terms of q, then keep differentiating until we obtain a closed differential relation, which is generically non-linear.

In each case, if $F(x)$ is the distribution function and $f(x)$ is the probability density function, from (13) we know that for a $p = \alpha$ we will have $F(q) = \alpha$. Differentiating we will obtain

$$\frac{dF(q)}{dq} = \frac{d\alpha}{dq} \tag{14}$$

Thus, consiering the definition of the density function (Definition 8), the first order quantile ODE is just

$$\frac{dq}{d\alpha} = \frac{1}{f(q)}, \tag{15}$$

where $q(\alpha)$ is the quantile function considered as a function of α, with $0 \leq \alpha \leq 1$.

In [13], the authors differentiate again to find, at least for some key common cases, a simple-second order non-linear ODE that they solve with power series. However, in our case we will solve numerically the first order quantile ODE and we will see if it is enough for our main objectives.

4.1 Examples

It will be interesting to present two most common examples of distributions and its corresponding quantiles ODEs.

4.1.1 Normal distribution

We know that the density function of the Standard Normal distribution is given by

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-x^2/2\right) \tag{16}$$

So, from expressions (15) and (16) we can obtain the quantile ODE

$$\frac{dq}{d\alpha} = \sqrt{2\pi} \exp\left(q^2/2\right), \tag{17}$$

with initial condition $q(1/2) = 0$.

4.1.2 Student distribution

We know that the density function of the Student distribution is

$$f(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})}(1 + x^2/n)^{-\frac{n+1}{2}}, \tag{18}$$

where $n > 0$ are the degrees of freedom.

Thus, as in the Normal case, from expressions (15) and (18) we can obtain the quantile ODE

$$\frac{dq}{d\alpha} = \sqrt{n\pi}\frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n+1}{2})}(1 + q^2/n)^{\frac{n+1}{2}}, \tag{19}$$

with initial condition $q(1/2) = 0$ [13].

5 Numerical methods for ordinary differential equations

As seen, our purpose is to compute quantiles using the solutions of ODEs. We know that the overwhelming majority of ODEs do not have solutions that can be expressed in terms of simple functions, thus we should introduce some numerical methods that produce approximations to the desired solutions. Since the advent of widespread digital computing in the 1960s, a great many theoretical and practical developments have been made in this area, but we will focus on two of most known and used methods: Euler and Runge-Kutta.

The aim is to solve all IVPs of the form

$$\left.\begin{array}{ll} x'(t) = f(t, x(t)), & t > 0 \\ x(t_0) = \eta & \end{array}\right\} \tag{20}$$

that possess a unique solution on some specified interval, $t \in [t_0, t_f]$.

5.1 Euler's method

It is appropriate that we start with detailed descriptions of Euler's method, its derivation and how it behaves numerically. To develop Euler's method for solving the general IVP (20) the approximation process begins by considering the Taylor series of $x(t+h)$ with remainder

$$x(t+h) = x(t) + hx'(t) + R_1(t). \tag{21}$$

The remainder term $R_1(t)$ is called the local truncation error (LTE). If $x(t)$ is twice continuously differentiable on the interval (t_0, t_f) the remainder term may be written as

$$R_1(t) = \frac{1}{2!}h^2 x''(\xi), \qquad \xi \in (t, t+h). \tag{22}$$

Then, if a positive number M exists so that $|x''(t)| \leq M$ for all $t \in (t_0, t_f)$, it follows that $|R_1(t)| \leq \frac{1}{2}Mh^2$, i.e., $R_1(t) = \mathcal{O}(h^2)$.

To derive Euler's method we begin by substituting $x'(t) = f(t, x)$ into the Taylor series (21) to obtain

$$x(t+h) = x(t) + hf(t, x(t)) + R_1(t). \tag{23}$$

We now introduce a grid of points $t = t_n$, where

$$t_n = t_0 + nh, \qquad n = 1 : N \tag{24}$$

and $N = \lfloor b(t_f t_0)/h \rfloor$ is the number of steps of length h needed to reach, but not exceed, $t = t_f$ (being $\lfloor x \rfloor$ the floor function, which takes the integer part of x and ignore the fractional part.). With $t = t_n$ (for $n < N$) in (23) we have

$$x(t_{n+1}) = x(t_n) + hf(t_n, x(t_n)) + R_1(t_n), \qquad n = 0 : N-1, \tag{25}$$

and, with the initial condition $x(t_0) = \eta$, we would be able to compute the exact solution of the IVP on the grid $\{t_n\}_{n=0}^{N}$ using this recurrence relation were the LTE term $R_1(t)$ not present.

However, since $R_1(t) = \mathcal{O}(h^2)$, it can be made arbitrarily small (by taking h to be sufficiently small) and, when neglected, we obtain Euler's method,

$$x_{n+1} = x_n + hf(t_n, x_n), \qquad n = 0, 1, 2, \ldots \tag{26}$$

with which we can compute the sequence $\{xn\}$ given that $x_0 = \eta$. We shall use the notation

$$f_n = f(t_n, x_n) \tag{27}$$

for the value of the derivative at $t = t_n$, so that Euler's method may be written as

$$x_{n+1} = x_n + hf_n. \tag{28}$$

On occasions (such as when dealing with components of systems of ODEs) it is more convenient to write x'_n for the approximation of the derivative of x at $t = t_n$. Then Euler's method would be written as

$$x_{n+1} = x_n + hx'_n. \tag{29}$$

We should analyse the behaviour of the method in the limit $h \to 0$. When h is decreased, it is required that the numerical solution should approach the exact solution, i.e. the size of the global error (GE)

$$|e_n| = |x(t_n) - x_n| \tag{30}$$

at $t = t_0 + nh$ should also decrease. This is intuitively reasonable; as we put in more computational effort, we should obtain a more accurate solution. The situation is, however, a little more subtle than is immediately apparent. If we were to consistently compare, say, the fourth terms in the sequences $\{x(t_n)\}$ and $\{x_n\}$ computed with $h = 0.5, 0.25$, and 0.125, then we would compute the error at $t_4 = t_0 + 2.0$, $t_0 + 1.0$ and $t_0 + 0.5$, respectively. That is, we would be comparing errors at different times when different values of h were employed. Even more worryingly, as $h \to 0$, $t_4 = t_0 + 4h \to t_0$, and we would eventually be comparing x_4 with $x(t_0) = \eta$ (the initial condition). What must be done is to compare the exact solution of the IVP and the numerical solution at a fixed time $t = t^*$, say, within the interval of integration. For this, the relevant value of the index n is calculated from $t_n = t_0 + nh = t^*$, or $n = (t^* - t_0)/h$, so that $n \to \infty$ as $h \to 0$.

By [5] we know that we could, were we prepared to take a sufficient number of small steps, obtain an approximation that was as accurate as we pleased. This suggests that the Euler's method is convergent.

Definition 10. ***Convergent numerical method.*** *A numerical method is said to converge to the solution $x(t)$ of a given IVP at $t = t^*$ if the GE $e_n = x(t_n) - x_n$ at $t_n = t^*$ satisfies*

$$|e_n| \to 0 \tag{31}$$

as $h \to 0$. It converges at a pth-order rate if $e_n = \mathcal{O}(h^p)$ for some $p > 0$ (and by convention we refer to the largest such value of p as the "order of the method").

We will take the view that numerical methods are of no value unless they are convergent (so any desired accuracy can be guaranteed by taking h to be sufficiently small).

It may be proved that Euler's method converges for IVPs of the form given by Equation (20) whenever it has a unique solution for $t_0 \le t \le t_f$. However, the proof is rather technical owing to the fact that it has to cope with a general nonlinear ODE. We shall therefore be less ambitious and provide a result only for a linear constant-coefficient case; the conclusions we draw will be relevant to more general situations. In particular, the result will indicate how the local errors committed at each step by truncating a Taylor series accumulate to produce the GE.

Theorem 2. *Euler's method applied to the IVP*

$$\left.\begin{array}{ll} x'(t) = \lambda x(t) + g(t), & 0 < t \le t_f, \\ x(0) = 1 & \end{array}\right\}$$

where $\lambda \in \mathbb{C}$ and g is a continuously differentiable function, converges and the GE at any $t \in [0, t_f]$ is $\mathcal{O}(h)$.

We can see the proof of the theorem in [5].

5.2 Runge-Kutta

Runge–Kutta (RK) methods are one-step methods composed of a number of stages. A weighted average of the slopes (f) of the solution computed at nearby points is used to determine the solution at $t = t_{n+1}$ from that at $t = t_n$. Euler's method is the simplest such method and involves just one stage.

The general s-stage RK method may be written in the form

$$x_{n+1} = x_n + h \sum_{i=1}^{s} b_i k_i, \tag{32}$$

where the $\{k_i\}$ are computed from the function f:

$$k_i = f\left(t_n + c_i h, x_n + h\sum_{j=1}^{s} a_{i,j}k_j\right), \qquad i = 1:s \tag{33}$$

so we will do this throughout. Thus, given a value of s, the method depends on $s^2 + s$ parameters $\{a_{i,j}, b_j\}$. These can be conveniently displayed in a tableau known as the Butcher array (seen in table 1)

c_1	$a_{1,1}$	$a_{1,2}$	$\dots$	$a_{1,s}$
c_2	$a_{2,1}$	$a_{2,2}$	$\dots$	$a_{2,s}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s}$
	b_1	b_2	$\dots$	b_s

Table 1: The Butcher array for a full (implicit) RK method.

In general, equations (33) constitutes s nonlinear equations to determine $\{ki\}$; once found, these values are substituted into (32) to determine x_{n+1}. Thus, a general RK method is implicit. However, if $a_{i,j} = 0$ for all $j \geq i$ (the matrix $\mathcal{A} = (a_{i,j})$ is strictly lower triangular) the tableau is shown in table 2 and $k_1, k_2, \dots, k_s$ may be computed in turn from (33) without the need to solve any nonlinear equations; we say that the method is explicit. These are the classical RK methods. We shall omit the zeros above the diagonal when writing a Butcher array for an explicit method.

0	0	0	$\dots$	0
c_2	$a_{2,1}$	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	0
	b_1	b_2	$\dots$	b_s

Table 2: The Butcher array for a explicit RK method.

Now, we will look at the issue of choosing parameter values to obtain the highest possible order and then we will look at absolute stability.

Definition 11. ***Local Truncation Error.*** *The LTE, T_{n+1}, of an RK method is defined to be the difference between the exact and the numerical solution of the IVP at time $t = t_{n+1}$:*

$$T_{n+1} = x(t_{n+1}) - x_{n+1},$$

under the localizing assumption that $x_n = x(t_n)$, i.e. that the current numerical solution x_n is exact. If $T_{n+1} = \mathcal{O}(h^{p+1})$ $(p > 0)$, the method is said to be of order p.

In our work, it will be interesting to introduce the four-stage, fourth-order method . A general four-stage RK method has 10 free parameters ($\frac{1}{2}s(s+1)$) and eight conditions are needed to obtain methods of maximum possible order (which is $p = 4$). The four-stage, fourth-order method (the most popular RK method) is shown in table

0	0			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	0	$\frac{1}{2}$	0	
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 3: The classic four-stage, fourth-order RK method.

It may appear from the development in [5] that it is always possible to find RK methods with s stages that have order s. This is not so for $s > 4$. The number of stages necessary for a given order is known up to order 8, but there are no precise results for higher orders. To appreciate the level of complexity involved, it is known that between 12 and 17 stages will be required for order 9 and the coefficients must satisfy 486 nonlinear algebraic equations. Also, for methods of order higher than 4, the order when applied to systems may be lower than when applied to scalar ODEs.

Definition 12. ***Absolute stability.*** *Applying an RK method to the linear ODE $x'(t) = \lambda x(t)$ with $\mathcal{R}(\lambda) < 0$, absolute stability requires that $x_n \to 0$ as $n \to \infty$.*

In general, to s-stage methods of order s (as the four-stage, fourth-order RK) we will have as follows:

(a) When applied to $x'(t) = \lambda x(t)$, the RK method leads to $x_{n+1} = R(\hat{h})x_n$, where the stability function $R(\hat{h})$ is a polynomial in $\hat{h}$ of degree s.

(b) Taylor expansion of the exact solution $x(t_{n+1})$ about $t = t_n$ and using $x'(t) = \lambda x(t)$, $x''(t) = \lambda^2 x(t)$, etc., leads to

$$x(t_{n+1}) = \left(1 + \hat{h} + \frac{1}{2!}\hat{h}^2 + \cdots + \frac{1}{s1}\hat{h}^s\right) x(t_n) + \mathcal{O}(\hat{h}^{s+1}). \tag{34}$$

(c) Under the localizing assumption $x_n = x(t_n)$ and supposing the method to have order s, then $x_{n+1} = x(t_{n+1}) + O(h^{s+1})$. It follows from (a) and (b) that

$$R(\hat{h}) = 1 + \hat{h} + \frac{1}{2!}\hat{h}^2 + \cdots + \frac{1}{s1}\hat{h}^s. \tag{35}$$

Thus, all s-stage, s-order RK methods have the same stability function, $R(\hat{h})$. Clearly

$$R(\hat{h}) = e^{\hat{h}} + \mathcal{O}(\hat{h}^{s+1}). \tag{36}$$

Also $|R(\hat{h})| \to \infty$ as $\hat{h}\infty$ so that no explicit RK method can be either A-stable or A_0-stable. The intervals/regions of absolute stability are defined as the set of real/complex values of $\hat{h}$ for which $|R(\hat{h})| < 1$ (since $R(\hat{h}) = 1$ at $\hat{h} = 0$, RK methods are always zero-stable). For the four-stage, fourth-order RK case the interval of absolute stability (IAS) will be $(-2.785, 0)$ [5].

6 Non-central χ^2 distribution and delta-gamma framework

We will devote this section to focus on an interesting case with some implications in finance: the non-central χ^2 distribution [9].

Definition 13. ***Non-central*** χ^2 ***distribution.*** *Let $(X_1, X_2, ..., X_k)$ be independent, normally distributed random variables with means μ_i and unit variances. Then, the random variable $\sum_{i=1}^{k} X_i^2$ is distributed according to the non-central χ^2 distribution. It will have two parameters: k will determine the degrees of freedom (in usual notation we will use ν), and $\xi = \sum_{i=1}^{k} \mu_i^2$ will be the non-centrality parameter.*

In general, the density function will be given by

$$f_{\nu,\xi}(x) = 2^{-\nu/2} \mathrm{e}^{-\frac{1}{2}(\xi+x)} x^{\nu/2-1} {}_0\tilde{F}_1(\frac{\nu}{2}; \frac{x\xi}{4}), \tag{37}$$

where ν are the degrees of freedom, ξ the non-centrality parameter and ${}_0\tilde{F}_1(a;z) = {}_0F_1(a;z)/\Gamma(a)$ the normalised confluent hypergeometric limit function.

However, we are considering only the case with $\nu = 1$ because, as we will explain, it is the one that will be interesting in finance. We know that the density function of the distribution in that case will be given by

$$f(x) = \frac{e^{-1/2(x+\xi)}}{\sqrt{2\pi x}} \cosh(\sqrt{\xi x}), \tag{38}$$

with $0 < x < \infty$. We know the distribution function too, which will be

$$F(x) = \Phi(\sqrt{x} - \sqrt{\xi}) + \Phi(-\sqrt{x} - \sqrt{\xi}), \tag{39}$$

in terms of Φ, the cumulative distribution function of the Standard Normal distribution.

Furthermore, we can consider the ES of this distribution and it will be interesting to derive a closed-form formula for it.

Proposition 2. *The ES at confidence level α of a non-central χ^2 random variable X with one degree of freedom and non-centrality parameter ξ is,*

$$ES_\alpha(X) = \frac{1}{1-\alpha}\left[(\xi+1)(1-\alpha) + \phi\left(\sqrt{q(\alpha)^\xi} + \sqrt{\xi}\right)\left(\left(\sqrt{q(\alpha)^\xi} + \sqrt{\xi}\right)e^{2\sqrt{q(\alpha)\xi}\,\xi} + \sqrt{q(\alpha)^\xi} - \sqrt{\xi}\right)\right],$$

where ϕ denotes the probability density function of the standard normal distribution.

The proof of this proposition is seen in [11].

Once presented the Non-central χ^2 distribution, we should explain why it is needed in finance. Thus, we will explore the delta-gamma framework, and in particular the one risk factor case. Suppose the current value of a portfolio is $V(t)$, the holding period is Δt, and the value of the portfolio at time $t+\Delta t$ is $V(t+\Delta t)$. The change in the portfolio value during the holding period is ΔV, where $\Delta V = V(t+\Delta t) - V(t)$. The VaR value of ΔV at a confidence level α, is given by $q(\alpha)$, where,

$$P(\Delta V < q(\alpha)) = \alpha.$$

In practice, Δt ranges from one day to two weeks and $\alpha \in (0,1)$ is close to 1. By Lemma 1 in Section 2, the ES risk measure at confidence level α is given by,

$$\mathrm{ES}_\alpha(\Delta V) = \mathbb{E}\left(\Delta V | \Delta V > q(\alpha)\right),$$

or, alternatively,

$$\mathrm{ES}_\alpha(\Delta V) = \frac{1}{1-\alpha}\int_{q(\alpha)}^{+\infty} x f_{\Delta V}(x) dx, \tag{40}$$

where $f_{\Delta V}$ is the probability density function of ΔV.

We assume that there are p risk factors and that $S(t) = (S_1(t), \ldots, S_p(t))^T$ denotes the value of these factors at time t. Define $\Delta S = S(t + \Delta t) - S(t)$ to be the change in the risk factors during the interval $[t, t + \Delta t]$. Then, the delta-gamma approximation is given by,

$$\Delta V \simeq \Delta V_\gamma = \Theta \Delta t + \delta^T \Delta S + \frac{1}{2} \Delta S^T \Gamma \Delta S, \tag{41}$$

where $\Theta = \frac{\partial V}{\partial t}, \delta_i = \frac{\partial V}{\partial S_i}, \Gamma_{i,j} = \frac{\partial^2 V}{\partial S_i \partial S_j}$, are called the Greeks, and all partial derivatives are being evaluated at $S(t)$. Further, ΔS is governed by a normal distribution with mean zero and covariance matrix Σ. Observe that in the univariate case ($p = 1$, i.e. one single risk factor), we have,

$$\Delta V_\gamma = \sum_{i=1}^{n} x_i \frac{\partial v_i}{\partial t} \Delta t + \sum_{i=1}^{n} x_i \frac{\partial v_i}{\partial S} \Delta S + \frac{1}{2} \sum_{i=1}^{n} x_i \frac{\partial^2 v_i}{\partial S^2} (\Delta S)^2, \tag{42}$$

where n represents the number of assets in the portfolio, x_i is the amount of asset i and v_i the value of asset i. In this particular case,

$$\Theta = \sum_{i=1}^{n} x_i \frac{\partial v_i}{\partial t}, \ \delta = \sum_{i=1}^{n} x_i \frac{\partial v_i}{\partial S}, \ \Gamma = \sum_{i=1}^{n} x_i \frac{\partial^2 v_i}{\partial S^2}.$$

Now, we will derive a closed formula to calculate the ES of ΔV_γ for a single risk factor. We follow the results provided in [12], where it is established the link between the density function of ΔV_γ and the density function of a non-central chi-squared distribution with one degree of freedom, called Q, with non-centrality parameter ξ and density function f_Q. More precisely,

$$f_{\Delta V_\gamma}(x) = \frac{2}{|\lambda|} f_Q \left(\frac{2}{\lambda} \left(x - \Theta \Delta t - \bar{\mu}_Q \right) \right), \tag{43}$$

where $\lambda = \Gamma C^2, d = \delta C, C = \sigma \sqrt{\Delta t} S(t), \bar{\mu}_Q = -\frac{1}{2} \frac{d^2}{\lambda}$ and $\xi = \left(\frac{d}{\lambda} \right)^2$. Further, the VaR value of ΔV_γ is given by,

$$q(\alpha) = \begin{cases} \frac{\lambda}{2} q(\alpha)^\xi + \bar{\mu}_Q + \Theta \Delta t, & \text{if } \lambda > 0, \\ \frac{\lambda}{2} q(1-\alpha)^\xi + \bar{\mu}_Q + \Theta \Delta t, & \text{if } \lambda < 0, \end{cases} \tag{44}$$

where the quantiles $q(\alpha)^\xi$ and $q(1-\alpha)^\xi$ represent the VaR value of Q at confidence level α and $1 - \alpha$ respectively.

In order to derive the expression for the ES, we differentiate between positive and negative λ. We start by assuming that $\lambda > 0$. In this case, it is shown in [12] that $f_{\Delta V_\gamma}$ is either unimodal or bimodal in its domain of definition $(\bar{\mu}_Q + \Theta \Delta t, +\infty)$. If we take into account (43) the ES is given by,

$$\mathrm{ES}_\alpha(\Delta V_\gamma) = \frac{1}{1-\alpha} \int_{q(\alpha)}^{+\infty} x f_{\Delta V_\gamma}(x) dx = \frac{1}{1-\alpha} \cdot \frac{2}{\lambda} \int_{q(\alpha)}^{+\infty} x f_Q \left(\frac{2}{\lambda} \left(x - \Theta \Delta t - \bar{\mu}_Q \right) \right) dx.$$

If we make the change of variables $y = \frac{2}{\lambda} \left(x - \Theta \Delta t - \bar{\mu}_Q \right)$, then by (44) we get,

$$\mathrm{ES}_\alpha(\Delta V_\gamma) = \frac{1}{1-\alpha} \int_{q(\alpha)^\xi}^{+\infty} \left(\frac{\lambda}{2} y + \Theta \Delta t + \bar{\mu}_Q \right) f_Q(y) dy = \frac{\lambda}{2} \mathrm{ES}_\alpha(Q) + \frac{1}{1-\alpha} \cdot (\Theta \Delta t + \bar{\mu}_Q) \int_{q(\alpha)^\xi}^{+\infty} f_Q(y) dy,$$

where $\mathrm{ES}_\alpha(Q)$ denotes the ES of the random variable Q. Finally, taking into account that,

$$\int_{q(\alpha)^\xi}^{+\infty} f_Q(y) dy = 1 - \alpha,$$

we conclude that,

$$\mathrm{ES}_\alpha(\Delta V_\gamma) = \frac{\lambda}{2}\mathrm{ES}_\alpha(Q) + \Theta\Delta t + \bar{\mu}_Q.$$

We now consider $\lambda < 0$. In this case, it is shown in [12] that $f_{\Delta V_\gamma}$ is either unimodal or bimodal in its domain of definition $(-\infty, \bar{\mu}_Q + \Theta\Delta t)$, where $x = \bar{\mu}_Q + \Theta\Delta t$ is a vertical asymptote for $f_{\Delta V_\gamma}$. If we take that observation into account then by (43) the ES is given by,

$$\mathrm{ES}_\alpha(\Delta V_\gamma) = \frac{1}{1-\alpha}\int_{q(\alpha)}^{\bar{\mu}_Q+\Theta\Delta t} x f_{\Delta V_\gamma}(x)dx = \frac{-1}{1-\alpha}\cdot\frac{2}{\lambda}\int_{q(\alpha)}^{\bar{\mu}_Q+\Theta\Delta t} x f_Q\left(\frac{2}{\lambda}\left(x-\Theta\Delta t-\bar{\mu}_Q\right)\right)dx.$$

If we make the change of variables $y = \frac{2}{\lambda}\left(x-\Theta\Delta t-\bar{\mu}_Q\right)$, then by (44) we get,

$$\begin{aligned}\mathrm{ES}_\alpha(\Delta V_\gamma) &= \frac{-1}{1-\alpha}\int_{q(1-\alpha)^\xi}^{0}\left(\frac{\lambda}{2}y+\Theta\Delta t+\bar{\mu}_Q\right)f_Q(y)dy \\ &= \frac{1}{1-\alpha}\left[\frac{\lambda}{2}\int_0^{q(1-\alpha)^\xi} y f_Q(y)dy + (\Theta\Delta t+\bar{\mu}_Q)\int_0^{q(1-\alpha)^\xi} f_Q(y)dy\right].\end{aligned}$$

Taking into account that,

$$\int_0^{q(1-\alpha)^\xi} f_Q(y)dy = 1-\alpha,$$

then,

$$\mathrm{ES}_\alpha(\Delta V_\gamma) = \frac{1}{1-\alpha}\cdot\frac{\lambda}{2}\int_0^{q(1-\alpha)^\xi} y f_Q(y)dy + \left(\Theta\Delta t+\bar{\mu}_Q\right). \tag{45}$$

Since Q is a non-central χ^2 distribution with one degree of freedom and non-centrality parameter ξ, we will obtain,

$$\begin{aligned}&\int_0^{q(1-\alpha)^\xi} y f_Q(y)dy = g\left(\sqrt{q(1-\alpha)^\xi}\right) - g(0) \\ &= (\xi+1)\left(\Phi\left(\sqrt{q(1-\alpha)^\xi}-\sqrt{\xi}\right)+\Phi\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)-1\right) - \phi\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)\cdot \\ &\cdot\left(\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)e^{2\sqrt{q(1-\alpha)^\xi\xi}}+\sqrt{q(1-\alpha)^\xi}-\sqrt{\xi}\right) \\ &= (\xi+1)(1-\alpha) - \phi\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)\cdot\left(\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)e^{2\sqrt{q(1-\alpha)^\xi\xi}}+\sqrt{q(1-\alpha)^\xi}-\sqrt{\xi}\right).\end{aligned} \tag{46}$$

Then, by (45) and (46),

$$\begin{aligned}\mathrm{ES}_\alpha(\Delta V_\gamma) = \frac{\lambda}{2}\cdot\frac{1}{1-\alpha}\Bigg[&(\xi+1)(1-\alpha) - \phi\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)\cdot \\ &\cdot\left(\left(\sqrt{q(1-\alpha)^\xi}+\sqrt{\xi}\right)e^{2\sqrt{q(1-\alpha)^\xi\xi}}+\sqrt{q(1-\alpha)^\xi}-\sqrt{\xi}\right)+\Theta\Delta t+\bar{\mu}_Q.\end{aligned}$$

We can summarize in the following proposition the value of the ES for all values of λ.

Proposition 3. *The ES at confidence level α of the delta-gamma approximation ΔV_γ reads,*

$$ES_\alpha(\Delta V_\gamma) = \frac{\lambda}{2} \cdot \frac{1}{1-\alpha} \left[(\xi+1)(1-\alpha) + sign(\lambda)\phi \left(\sqrt{q^\xi} + \sqrt{\xi} \right) \cdot \left(\left(\sqrt{q^\xi} + \sqrt{\xi} \right) e^{2\sqrt{q^\xi \xi}} + \sqrt{q^\xi} - \sqrt{\xi} \right) \right] + \\ + \Theta \Delta t + \bar{\mu}_Q, \tag{47}$$

where $sign(\lambda)$ is the sign function (takes the value 1 for positive λ and -1 for negative λ) and $q^\xi = q(\alpha)^\xi$ for positive λ and $q^\xi = q(1-\alpha)^\xi$ for negative λ.

Proof. The result follows from the expressions above. □

Looking at formula (47), the only required computation for obtaining the ES value in delta-gamma approach is the quantile q^ξ. The quantile q^ξ satisfies $F_\xi\left(q^\xi\right) = \eta$, where $\eta = \alpha$ for positive λ, $\eta = 1-\alpha$ for negative λ, and F_ξ is the distribution function of a non-central chi-squared random variable with one degree of freedom and non-centrality parameter ξ [11]. Thus, we should introduce the quantile computation via the corresponding non-central χ^2 distribution ODE. Considering the probability density function (38) and the method explained in Section 4 we will have

$$\frac{dq}{d\alpha} = \frac{\sqrt{2\pi q}}{e^{-1/2(q+\xi)} \cosh(\sqrt{\xi q})}. \tag{48}$$

In conclusion, we would solve this ODE numerically with methods explained in Section 5 in order to obtain the ES value in delta-gamma framework (and VaR too).

7 Results

In this section we will present our results when applying all the theoretical concepts developed throughout the previous sections. We will apply the numerical methods explained in Section 5 in order to obtain solutions to the quantile ODEs that we have introduced in Section 4, and we will focus on the non-central χ^2 distribution with one degree of freedom which we have presented in Section 6.

First, we will solve the Normal and Student cases, in order to show two common examples of probability distributions which will be used to explain the quantile ODE solution behaviour. Then, we will study our particular case, the non-central χ^2 with one degree of freedom, emphasizing in the differences between the theoretical results and the ODE ones. Also, we will study the non-centrality parameter influence on ODE solution and we will compare our results with the Python ones, in order to observe the main improvements of our methodology and the main problems too.

At this point it is important to state two key points. First, we should specify that Euler and Runge-Kutta solutions are obtained by an implementation of these methods done in Python language and shown in Appendix A. Second, we should mention that in order to evaluate the ODE results accuracy we will apply the quantile definition as the distribution function inverse (13), thus we will have a formal definition for the error, which will be given by

$$\epsilon = |F(q) - \alpha|, \tag{49}$$

where q is our quantile result for a given probability α.

7.1 Previous examples

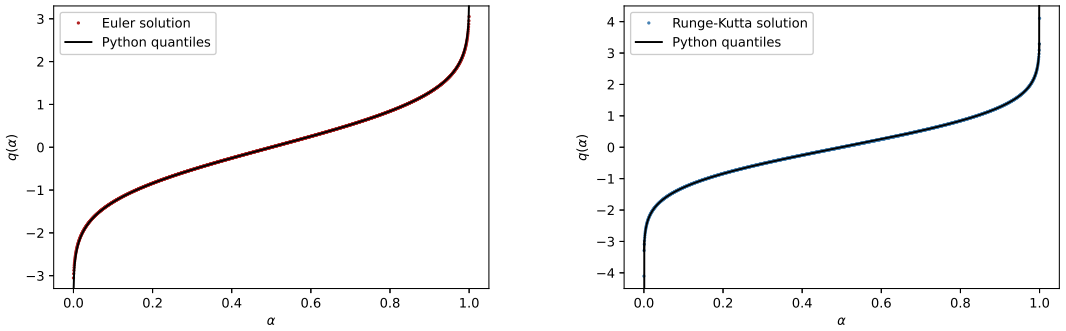
7.1.1 Normal distribution

In the normal case, as said in (17), we will have the quantile ODE

$$\frac{dq}{d\alpha} = \sqrt{2\pi}\exp\left(q^2/2\right), \tag{50}$$

with initial condition $q(1/2) = 0$.

In order to solve this ODE, we will use the two numerical methods that we have explained before: Euler and Runge-Kutta, and in both methods we will use 1000 iterations. Then, we will compare the results between them and with the Python result, so we show in each plot of the figure 2 the Python result and the numerical ones.[1]



(a) Euler

(b) Runge-Kutta

Figure 2: Python quantiles and ODE results for the Normal distribution.

[1] We obtain the results for $\alpha \in [0, 0.5)$ by simmetry.

We observe that the ODE numerical solutions are really close to the Python quantiles with both methods. It will be interesting commenting the performance for extreme values, i.e tails of the distribution. It is seen that for these values with Runge-Kutta we obtain higher results than with Euler. As we know that the theoretical quantiles for $\alpha = 0$ and $\alpha = 1$ are $q(0) = -\infty$ and $q(1) = +\infty$, we could think the results with Runge-Kutta are more adjusted than the Euler ones. In order to see that with more detail, we also show the quantile values for some values of α and their errors in the table 4.

α	Euler quantile	RK quantile	Python quantile	Euler error	RK Error
0	-3.051943	-4.105126	$-\infty$	∞	∞
0.05	-1.641601	-1.644854	-1.644854	0.003252	$4.782175 \cdot 10^{-12}$
0.10	-1.280386	-1.281552	-1.281552	0.001166	$3.148592 \cdot 10^{-13}$
0.20	-0.841306	-0.841621	-0.841621	0.000316	$1.454392 \cdot 10^{-14}$
0.30	-0.524302	-0.524401	-0.524401	0.000099	$3.330669 \cdot 10^{-16}$
0.40	-0.253326	-0.253347	-0.253347	0.000021	$4.996004 \cdot 10^{-16}$
0.50	0	0	0	0	0
0.60	0.253326	0.253347	0.253347	0.000021	$4.996004 \cdot 10^{-16}$
0.70	0.524302	0.524401	0.524401	0.000099	$3.330669 \cdot 10^{-16}$
0.80	0.841306	0.841621	0.841621	0.000316	$1.454392 \cdot 10^{-14}$
0.90	1.280386	1.281552	1.281552	0.001166	$3.148592 \cdot 10^{-13}$
0.95	1.641601	1.644854	1.644854	0.003252	$4.782175 \cdot 10^{-12}$
1	3.051943	4.105126	∞	∞	∞

Table 4: Normal quantiles and numerical results for the ODE.

With these results, we observe a really better adjustment to theoretical quantiles with the Runge-Kutta method for all the α values. For Euler, we observe that the nearer to the tail we are the more different to Python and Runge-Kutta the results become and the higher the error is. We could also represent the Euler and Runge-Kutta errors in order to understand their behaviours.

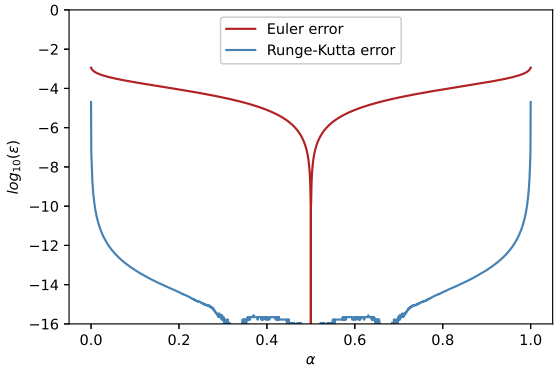


Figure 3: Error logarithm for Euler and Runge-Kutta.

From the figure 3, where we have represented the logarithm of the error in order to see the error performance in more detail, we confirm that the error with Euler is greater than with Runge-Kutta. We can also observe that in tails the error is higher than in the central points.

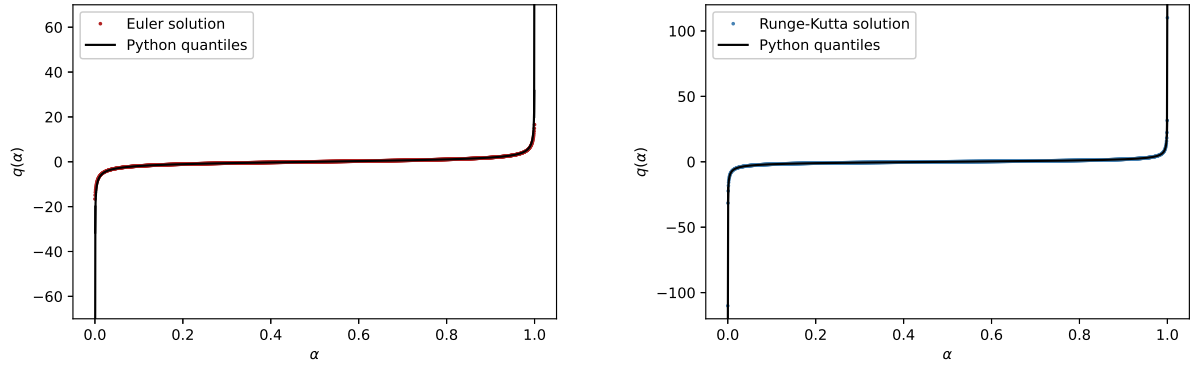
7.1.2 Student distribution

In the student case, as said in (19), we will have the quantile ODE

$$\frac{dq}{d\alpha} = \sqrt{n\pi}\frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n+1}{2})}(1+q^2/n)^{\frac{n+1}{2}}, \tag{51}$$

with initial condition $q(1/2) = 0$.

We will proceed as in the previous example and we will solve the ODE with Euler and Runge-Kutta using 1000 iterations. We show the results, for the case $n = 2$, in figure 4.



(a) Euler

(b) Runge-Kutta

Figure 4: Python quantiles and ODE results for the Student distribution ($n = 2$).

Graphically we observe that the ODE results for both methods are close to Python results. Maybe, the more interesting points will be the tails of the distribution, as we have mentioned in Normal case. It is seen that for these values with Runge-Kutta we obtain higher results than with Euler, and we could think again that the results with Runge-Kutta are more adjusted than the Euler ones. In order to see that with more detail, we also show the quantile values for some values of α and their errors in the table 5.

α	Euler quantile	RK quantile	Python quantile	Euler error	RK error
0	-16.570276	-110.080509	$-\infty$	∞	∞
0.05	-2.899089	-2.919986	-2.919986	0.000617	$2.473577{\cdot}10^{-13}$
0.10	-1.880637	-1.885618	-1.885618	0.000381	$1.043332{\cdot}10^{-13}$
0.20	-1.059738	-1.060660	-1.060660	0.000167	$1.454392{\cdot}10^{-14}$
0.30	-0.616974	-0.617213	-0.6172134	0.000065	$4.996004{\cdot}10^{-16}$
0.40	-0.288629	-0.288675	-0.2886751	0.000015	$1.165734{\cdot}10^{-15}$
0.50	0	0	0	0	0
0.60	0.288629	0.288675	0.2886751	0.000015	$1.110223{\cdot}10^{-15}$
0.70	0.616974	0.617213	0.6172134	0.000065	$5.551115{\cdot}10^{-16}$
0.80	1.059738	1.060660	1.060660	0.000167	$1.443290{\cdot}10^{-14}$
0.90	1.880637	1.885618	1.885618	0.000381	$1.042499{\cdot}10^{-13}$
0.95	2.899089	2.919986	2.919986	0.000617	$2.473577{\cdot}10^{-13}$
1	16.570276	110.080509	∞	∞	∞

Table 5: Student quantiles and numerical results for the ODE.

We observe the same performance than in the Normal case: better results with Runge-Kutta than with Euler. We will show the error $\epsilon = |F(q) - \alpha|$ too.

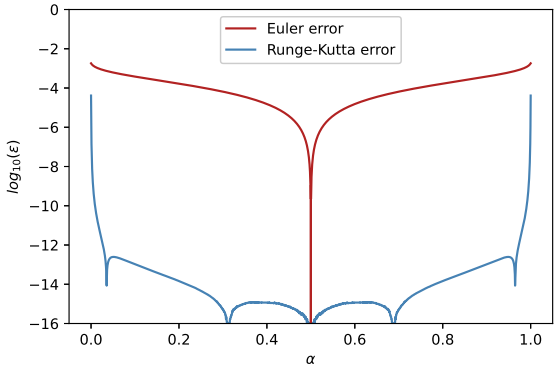


Figure 5: Error logarithm for Euler and Runge-Kutta.

As in the Normal case, we have greater errors in tails, and the results with Runge-Kutta are better.

7.2 Non-central χ^2 distribution

We have explained in Section 6 why it will be interesting for us the quantile computation in the specific case of non-central χ^2 with 1 degree of freedom. Thus, first of all we will show the density performance for different non-centrality parameter values (ξ) in order to have a first approach to this probability distribution.

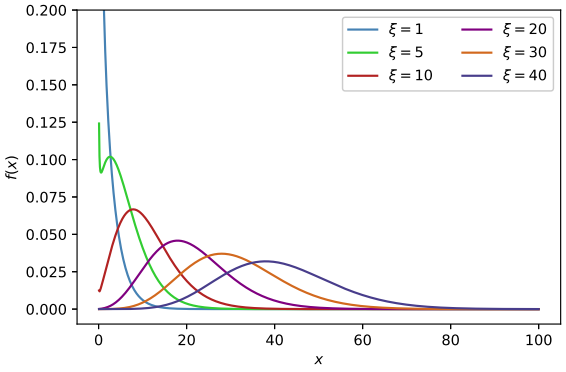


Figure 6: Probability density for non-central χ^2 distribution with one degree of freedom and different ξ values.

We should apply the quantile calculation with ODEs that we have outlined before. First, we have a keen interest in the quantile computation method with Python, so as to understand which results we will be comparing these with. We have explained in Section 3 there are plenty of methods to obtain quantiles. In particular, in Scipy module of Python [16] the first step is the distribution computation, and then the quantiles are obtained by inversion. So as to achieve the distribution function, the Abramowitz and Stegun formula is applied [17], which is an analytical approximation given in [1],

$$P(\nu, \xi) = \sum_{j=0}^{\infty} e^{-\xi/2} \frac{(\xi/2)^j}{j!} P(\nu + 2j). \tag{52}$$

With this method we obtain the results in figure 7 for different values of the non-centrality parameter ξ.

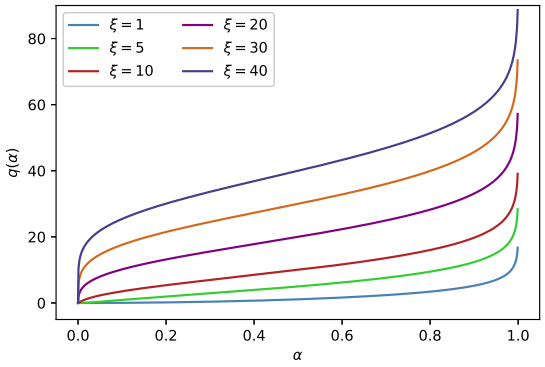


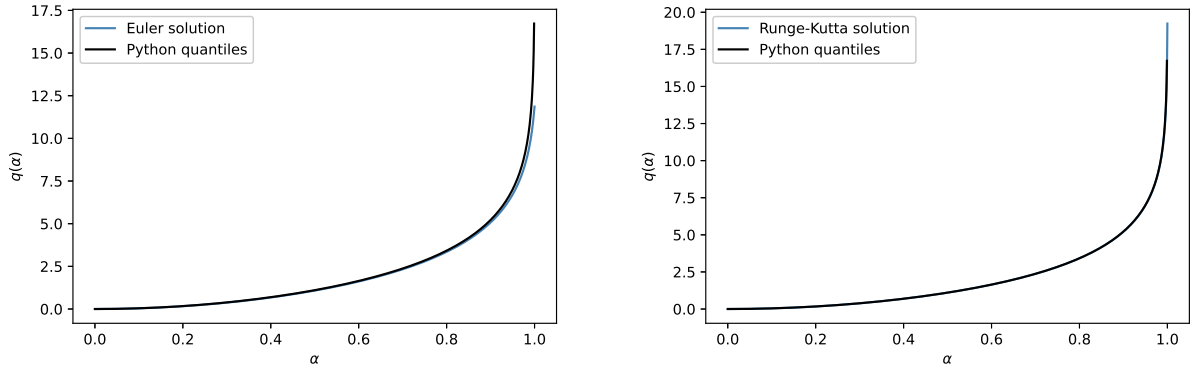
Figure 7: Python quantiles for non-central χ^2 distribution with one degree of freedom and different ξ values.

Now, from expression (48), we know that the quantile ODE will be

$$\frac{dq}{d\alpha} = \frac{\sqrt{2\pi q}}{e^{-1/2(q+\xi)}\cosh(\sqrt{\xi q})}. \tag{53}$$

In order to choose an initial condition for this ODE, we must keep in mind that for the density function of non-central χ^2 distribution we have $x \in (0, \infty)$, thus we cannot consider $q(0)$. We know an approach to the distribution function in terms of the cumulative distribution function of the Standard Normal distribution, which is given in (39), so we will consider the initial condition $q(10^{-20}) = \Phi(\sqrt{10^{-20}} - \sqrt{\xi}) + \Phi(\sqrt{10^{-20}} - \sqrt{\xi})$.

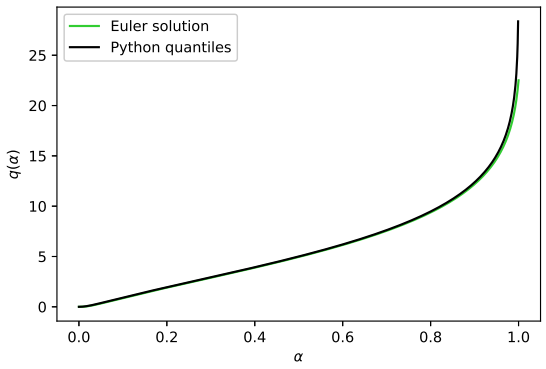
We will deal with this ODE using Euler and Runge-Kutta methods, as in the Normal and Student examples, and we will consider different values of the non-centrality parameter ξ in order to study the influence of this parameter in our results. First, we show in following figures the results for $\xi = 1$, $\xi = 5$ and $\xi = 10$, and the performance of errors for these cases too. It will be important to mention that in these cases we will use our Euler and Runge-Kutta implementations with 700 iterations (numerical methods implementations are seen in Appendix A and some numerical results are in Appendix B).



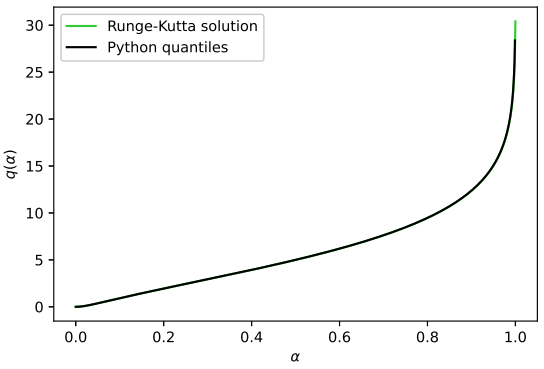
(a) Euler

(b) Runge-Kutta

Figure 8: Python and ODE quantiles for the non-central χ^2 distribution with 1 degree of freedom, $\xi = 1$.

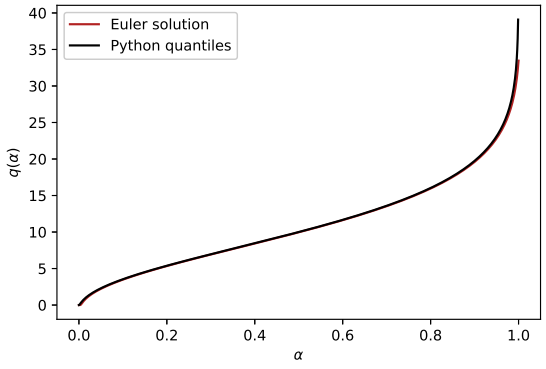


(a) Euler

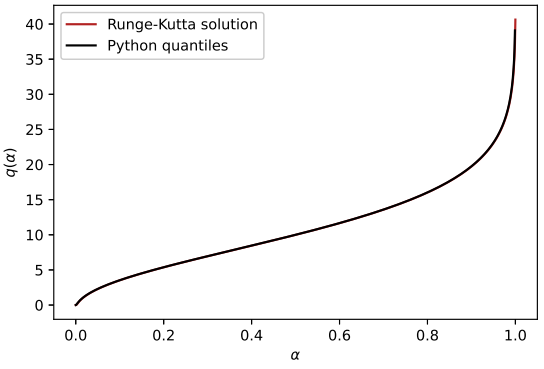


(b) Runge-Kutta

Figure 9: Python and ODE quantiles for the non-central χ^2 distribution with 1 degree of freedom, $\xi = 5$.

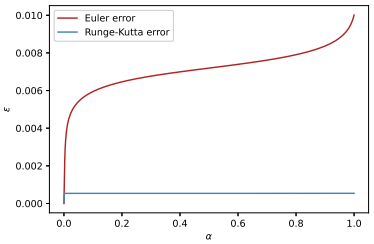


(a) Euler

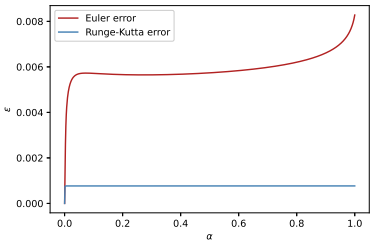


(b) Runge-Kutta

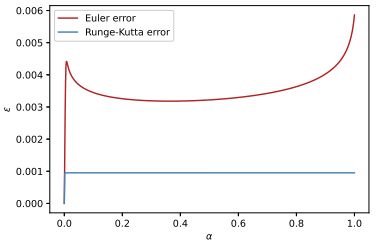
Figure 10: Python and ODE quantiles for the non-central χ^2 distribution with 1 degree of freedom, $\xi = 10$.



(a) $\xi = 1$



(b) $\xi = 5$



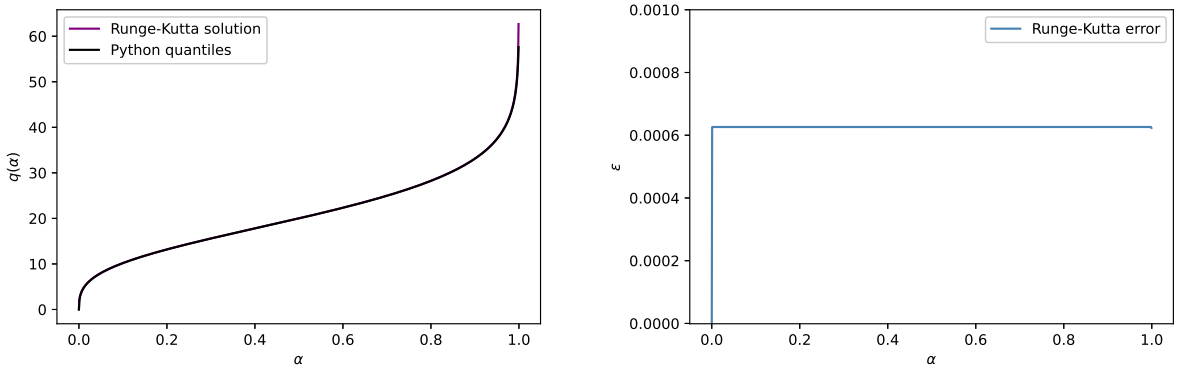
(c) $\xi = 10$

Figure 11: Euler and Runge-Kutta errors.

The main observation is the good fitting of ODE results to the Python quantiles, for both numerical methods and for all the non-centrality parameters which have been considered. In particular, if we only consider the tail results, in quantile figures it is seen that Runge-Kutta is more accurate than Euler because

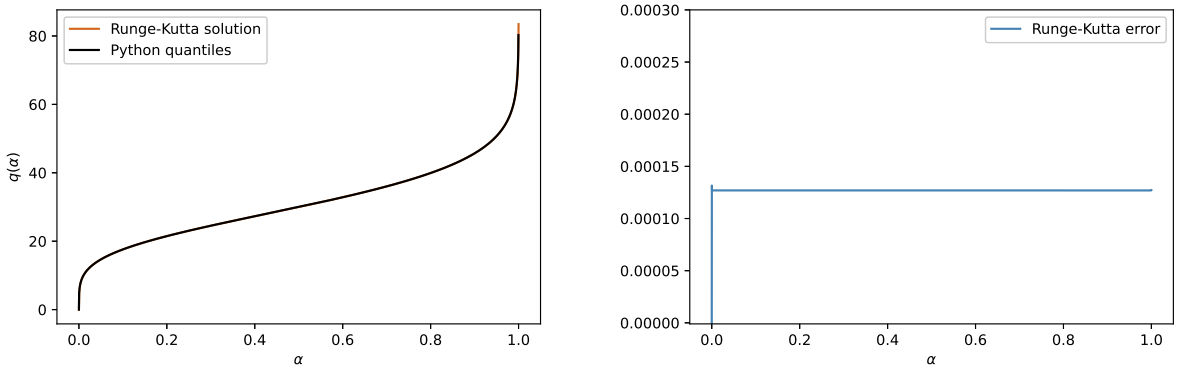
it provides higher results while Euler has limited results in that case. In error figures[2] it is clearly seen the behaviour explained for tails, and the Euler error increases when $\alpha \to 1$. In general, for any α the Runge-Kutta $q(\alpha)$ is better and has a lower error when compared with theoretical quantile. Besides, the Runge-Kutta error seems to be constant while Euler one does not. Thus, for following experiments with greater non-centrality parameters, we will only consider the Runge-Kutta method.

Then, we will consider the Runge-Kutta quantiles for non-centrality parameters $\xi = 20$, $\xi = 30$ and $\xi = 40$. In these cases it was seen during our numerical experiments that an increasing of the iterations number is needed when the non-centrality parameter is higher in order to obtain accurate results. We will consider $n = 1100$, $n = 4000$ and $n = 200000$, respectively (tables with numerical results are in Appendix B).



(a) Runge-Kutta (b) Runge-Kutta error

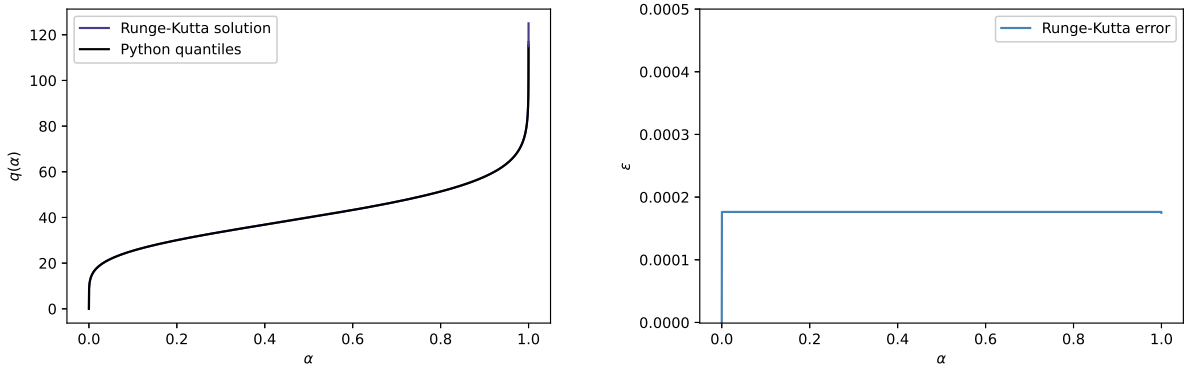
Figure 12: Python and Runge-Kutta quantiles for the non-central χ^2 distribution with 1 degree of freedom, $\xi = 20$, and Runge-Kutta error.



(a) Runge-Kutta (b) Runge-Kutta error

Figure 13: Python and Runge-Kutta quantiles for the non-central χ^2 distribution with 1 degree of freedom, $\xi = 30$, and Runge-Kutta error.

[2]It is convenient to remember that error has been calculated from expression (49) when applying the analytical approximation to non-central χ^2 distribution with one degree of freedom (39).



(a) Runge-Kutta

(b) Runge-Kutta error

Figure 14: Python and Runge-Kutta quantiles for the non-central χ^2 distribution with 1 degree of freedom, $\xi = 40$, and Runge-Kutta error.

In figures 12, 13 and 14 we observe, as in the cases with lower ξ, the good fitting of Runge-Kutta quantiles to Python ones. Besides, we notice that errors are constant in all cases and they have the same order of magnitude (10^{-4}). Thus, we can conclude that ODE quantiles provide us good results for a widely range of non-centrality parameters ξ, specifically Runge-Kutta computation, which has constant error and lower than Euler one.

We should pay attention to the increasing of iterations number which we have mentioned before. We can detect the big increasing of quantiles result when α is near 0, and it is seen that the increasing is more considerable when ξ is higher. This is why we need a greater number of iterations so as to have a good fitting to the theoretical quantiles with the purpose of avoiding that initially the Runge-Kutta result grows excessively. In particular, we have chosen the iterations number considering the error results with the aim of having a similar value of these errors for all cases (as we have said they have the same order of magnitude).

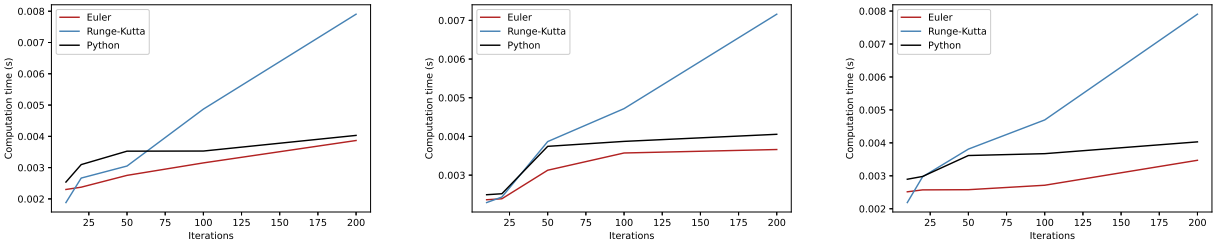
Regarding the iterations increasing, it would be remarkable that it is detected an exponential rise of iterations needed, as seen. Therefore, we have only considered non-centrality parameters until $\xi = 40$, because for higher ξ, millions of iterations will be needed and the computation would not be viable.

At this point, it appears interesting to calculate the computation time in order to have knowledge of the influence of iterations in computation viability. Moreover, this time computation will let us compare the ODE quantile computation with other methods like Python computation. In order to study that, we will calculate computation times for different values of ξ and different numbers of iterations. It is seen that the greater non-centrality parameter values are, the higher numbers of iterations are needed, therefore we will only study cases with low non-centrality parameters. In that cases, we will be able to obtain good results with low numbers of iterations, and consequently with low computation times.

In particular, we will study the cases $\xi = 1/2$, $\xi = 1$ and $\xi = 2$ with numbers of iterations $n = 10$, $n = 20$, $n = 50$, $n = 100$, $n = 200$ and $n = 700$. We will show the results in table 6 and in figure 15.

ξ	Iterations	Euler time (s)	Euler Mean Error	RK time (s)	RK Mean Error	Python time (s)	Python Mean Error
1/2	10	0.0023016	0.2890819	0.0018827	0.0572593	0.0025386	$6.5265537 \cdot 10^{-11}$
1/2	20	0.0023748	0.1702915	0.0026661	0.0272322	0.0030969	$5.5018950 \cdot 10^{-11}$
1/2	50	0.0027525	0.0791010	0.0030510	0.0100671	0.0035262	$4.0164389 \cdot 10^{-11}$
1/2	100	0.0031511	0.0429240	0.0048700	0.0046858	0.0035303	$4.0260998 \cdot 10^{-11}$
1/2	200	0.0038669	0.0229015	0.0079048	0.0021517	0.0040301	$4.5890274 \cdot 10^{-11}$
1/2	700	0.0068237	0.0071562	0.0248006	0.0005005	0.0065031	$4.2206075 \cdot 10^{-11}$
1	10	0.0023643	0.2898188	0.0022909	0.0586846	0.0024940	$1.8477991 \cdot 10^{-11}$
1	20	0.0023895	0.1701621	0.0024335	0.0280177	0.0025191	$3.3487598 \cdot 10^{-11}$
1	50	0.0031270	0.0788841	0.0038673	0.0104196	0.0037443	$1.7038001 \cdot 10^{-11}$
1	100	0.0035725	0.0428011	0.0047187	0.0048785	0.0038723	$2.4962785 \cdot 10^{-11}$
1	200	0.0036621	0.0428010	0.0071593	0.0022576	0.0040555	$3.0495265 \cdot 10^{-11}$
1	700	0.0074509	0.0071559	0.0204198	0.0005372	0.0072466	$3.0562187 \cdot 10^{-11}$
2	10	0.0025149	0.2851696	0.0021878	0.0612461	0.0028962	$2.8571227 \cdot 10^{-11}$
2	20	0.0025731	0.1659009	0.0029699	0.0294734	0.0029802	$3.3198405 \cdot 10^{-11}$
2	50	0.0025790	0.0766029	0.0038108	0.0110773	0.0036160	$4.0257476 \cdot 10^{-11}$
2	100	0.0027139	0.0415818	0.0046959	0.0052372	0.0036724	$3.3623271 \cdot 10^{-11}$
2	200	0.0034713	0.0222423	0.0079048	0.0022517	0.0040301	$4.5890274 \cdot 10^{-11}$
2	700	0.0082042	0.0070040	0.0214160	0.00072371	0.0065299	$3.5840419 \cdot 10^{-11}$

Table 6: Computation times.



(a) $\xi = 1/2$ (b) $\xi = 1$ (c) $\xi = 2$

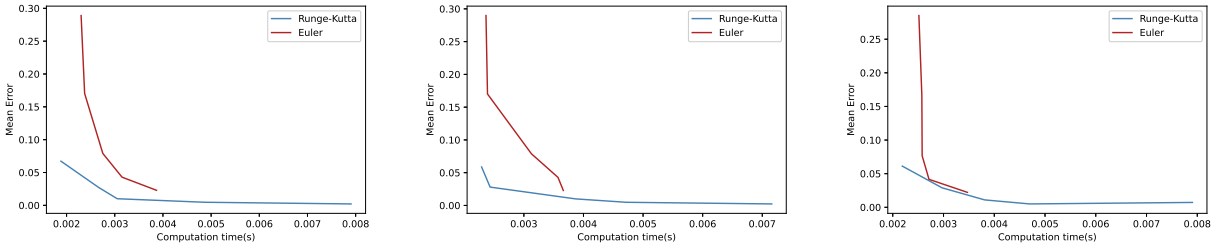
Figure 15: Computation times in function of iterations number.[3]

It is seen that Euler computation times are the lowest of three methods considered. Moreover, in general the Runge-Kutta computation times are higher than Python ones. Specifically, when $n = 10$ and $n = 20$ are used, the Runge-Kutta computation times are lower than the Python ones for all studied cases, and with $n = 50$ the computations times are really similar with both methods. For higher iterations number, the computation time in Runge-Kutta method increases while in Python one it remains approximately constant.

In spite of obtaining improved computation time results when using Euler or Runge-Kutta with low iterations number, the errors should also be considered. In table 6 it is clearly seen that with Python method the mean errors are extremely better for all considered cases (order 10^{-11}). Meanwhile, in Runge-Kutta method, the mean errors have orders between 10^{-2} and 10^{-3}, and in Euler the mean errors have orders between 10^{-1} and 10^{-3}. Besides, these errors go down when the iterations number are increased, but we would not obtain good computation time results in that case.

In conclusion, the ODE quantile computation with Euler or Runge-Kutta could be interesting when the non-centrality parameter is not too high and when not a really big accuracy is needed, because in that cases ODE computation would be faster than Python one. Specifically, if we pay attention to the table 6 and we represent (figure 16) the mean errors in function of computation times, we could notice that Runge-Kutta with low number of iterations would be more interesting than Euler with high number of iterations, because not only we would obtain similar computation times but also lower errors.

[3]Data corresponding to $n = 700$ are not being represented in order to observe better the initial points.



(a) $\xi = 1/2$ (b) $\xi = 1$ (c) $\xi = 2$

Figure 16: Mean errors in function of computation times.[4]

It would also be interesting to mention that in our work basic ODE numerical methods are being used. Nevertheless, it is known that it exists more advanced numerical methods like advanced Runge-Kutta with higher orders or numerical methods including interpolation. If this type of methods had been used maybe the results found would have been better.

[4]Like in previous figure, data corresponding to $n = 700$ are not being represented in order to observe better the initial points.

8 Conclusions

At the beginning of the present work we have considered as a main goal introducing the quantile computation by solving ODEs numerically. That is why some theoretical concepts have been introduced, so as to understand the relevance of quantile computation within a financial context and to have knowledge about ordinary differential equations link with quantiles and about the numerical methods used for solving them. Moreover, we have presented the particular case of non-central χ^2 distribution and explained its relationship with finance.

Once the theoretical concepts have been presented, it has been the moment to present numerical experiments and results. First, Normal and Student distributions have been studied as common examples. For both ODEs numerical solutions with Euler and Runge-Kutta, it has been seen that it exists a really good fitting to theoretical solutions, and to Python results too. Errors have been analysed and it has been noticed that Runge-Kutta solutions are better than Euler ones, since they have lower error terms. Moreover, it has been observed that results for tails are worse than for central points. Summarizing, for both cases the error terms with Runge-Kutta have orders between 10^{-12} and 10^{-16}, and with Euler between 10^{-4} and 10^{-5}. Therefore, the accuracy of the method has been checked, specially for Runge-Kutta.

Afterwards, as introduced throughout the previous sections, the experiments have been focused on non-central χ^2 distribution, and more specifically in the case with one degree of freedom. On the one hand, fitting of ODE results to theoretical and Python ones has been verified. The errors have been examined and we have concluded that for Euler they are around order 10^{-3} and for Runge-Kutta they have order 10^{-4}. Moreover, for Runge-Kutta errors appear to be constant, while for Euler do not.

On the other hand, the influence of non-centrality parameter has been studied. It has been proved that the higher the non-centrality parameter ξ is, the higher the iterations number of numerical methods should be so as to obtain accurate results. That is why for $\xi = 1$, $\xi = 5$ or $\xi = 10$ considering 700 iterations is enough, but 200000 iterations have been needed for $\xi = 40$.

Regarding computation times, an improvement has been noticed with respect to Python when both non-centrality parameters and iterations number are low. In that cases, the advantage of using Runge-Kutta has been verified as long as a big accuracy was not needed.

In conclusion, an interesting methodology for quantile computation by solving numerically ODEs have been introduced. The advantages and limitations of this methodology have been verified, considering accuracy and computation times in our study. Moreover, the possible application of this new method in financial context have been showed.

8.1 Future research

The extension of the present work should be limited, and consequently it has not been possible to present and discuss all the issues which have appeared throughout our research. Therefore, we would also like to present the main future investigation lines which are thought to be interesting.

One of the most relevant issues which have been noticed is the importance of number of iterations n, specially with high non-centrality parameters, and it has been seen an extreme increase of n when ξ exceeds 20 (and specially when it exceeds 30). It could be interesting study with more detail the iterations increasing performance, specifically with Runge-Kutta since it has been verified as the best of the two considered numerical methods.

Moreover, we have considered this quantile computation within a financial context. Therefore, we would like to implement our quantile computation method in order to obtain VaR and ES in some specific contexts of finance. We should study the specific probability distribution needed and apply the proposed methodology. Besides, it would be interesting analyse the problem with more than one risk factor.

Some limitations of the proposed methodology have been discovered. That is why it would be desirable considering some possible improvements. For instance, more advanced numerical methods could be tested, like Runge-Kutta of higher orders or with adaptive step size [5]. Furthermore, some kinds of interpolation could also be implemented, like in [8] where cubic interpolation is used.

Appendices

A Euler and Runge-Kutta implementations in Python

In this Appendix we will show the implementations in Python of both numerical methods used in our experiments in order to solve quantile ODEs: Euler and Runge-Kutta.

```
##  PYTHON CODE ##

## IMPORTS ##
import numpy as np
from numpy import sqrt, cosh, pi, exp
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as scp
from scipy.stats import norm, ncx2

## EULER FUNCTION ##
def Euler(f, xa, xb, ya, n):

    # Definition of initial values for iterations
    h = (xb - xa) / float(n)
    x = xa
    y = ya

    # Lists where we save new values of iterations
    a = []
    sol = []
    a.append(x)
    sol.append(y)

    # Iterations
    for i in range(n):
        y = y + h * f(y)
        x = x + h

        # Save values of iterations
        a.append(x)
        sol.append(y)

    # DataFrame with solutions
    solution = pd.DataFrame()
    solution['alpha'] = a
    solution['quantile'] = sol

    return solution

## RK FUNCTION ##
def runge_kutta(f, xa, xb, ya, n):

    # Definition of initial values for iterations
    y = ya
    x = xa
    h = (xb - xa) / float(n)

    # Lists where we save new values of iterations
    a = []
    sol = []
    a.append(x)
```

```
    sol.append(y)

    # Iterations
    for i in range(n):

        # Values for weighted average (compare with theory)
        k1 = f(y)
        k2 = f(y + (h/2)*k1)
        k3 = f(y + (h/2)*k2)
        k4 = f(y + h*k3)

        # Calculate the new value of y and x as per the Runge-Kutta method
        y = y + (h/6)*(k1 + 2*k2 + 2*k3 + k4)
        x = x + h

        # Save values of iterations
        a.append(x)
        sol.append(y)

    # DataFrame with solution
    solution = pd.DataFrame()
    solution['alpha'] = a
    solution['quantile'] = sol

    return solution
```

It would also be interesting to present an example of application of these functions in order to obtain de numerical solutions presented for specific parameters (quantile ODE f, number of iterations n, and conditions x_a, x_b and y_a)

```
# Chi definition
def chi(x):
    return sqrt(2*pi*x)/(exp(-1/2*(x+l))*cosh(sqrt(l*x)))

# Non-centrality parameter
l=1

# Parameters for numerical methods
x_i = 10**(-20)
x_f = 1
y_i = norm.cdf(sqrt(x_i)-sqrt(l))-norm.cdf(-sqrt(x_i)-sqrt(l))
n = 700

# Euler solution
solution_euler_1_700 = Euler(chi,x_i,x_f,y_i,n)

# RK solution
solution_rk_1_700 = runge_kutta(chi,x_i,x_f,y_i,n)
```

B Non-central χ^2 quantile numerical results

Throughout the Section 7.2, where the non-central χ^2 results have been presented, several numerical experiments have been done. In this Appedix the intention is to present the numerical results obtained, which have been plotted in referenced section, for some α values.

α	Euler	Euler error	RK	RK error	Python	Python error
0	$4.839418 \cdot 10^{-11}$	0.000003	$4.839418 \cdot 10^{-11}$	0.000003	0	0
0.1	$3.777390 \cdot 10^{-2}$	0.005946	$4.224367 \cdot 10^{-2}$	0.000537	0.042824	$1.604272 \cdot 10^{-14}$
0.2	$1.600675 \cdot 10^{-1}$	0.006463	$1.700371 \cdot 10^{-1}$	0.000537	0.171446	$1.475065 \cdot 10^{-10}$
0.3	$3.687044 \cdot 10^{-1}$	0.006767	$3.846754 \cdot 10^{-1}$	0.000537	0.387185	$3.349543 \cdot 10^{-13}$
0.4	$6.682395 \cdot 10^{-1}$	0.006994	$6.909688 \cdot 10^{-1}$	0.000537	0.694917	$3.234224 \cdot 10^{-11}$
0.5	1.070175	0.007193	1.101121	0.000537	1.107008	$1.099121 \cdot 10^{-14}$
0.6	1.599433	0.007391	1.641533	0.000537	1.650209	$7.074230 \cdot 10^{-12}$
0.7	2.308609	0.007615	2.368220	0.000537	2.381377	$5.662137 \cdot 10^{-15}$
0.8	3.319943	0.007904	3.413570	0.000537	3.435430	$1.409595 \cdot 10^{-11}$
0.9	5.010877	0.008366	5.204938	0.000537	5.252308	$1.054712 \cdot 10^{-14}$

Table 7: Results for $\xi = 1$, $n = 700$.

α	Euler	Euler error	RK	RK error	Python	Python error
0	$6.549429 \cdot 10^{-12}$	$1.676122 \cdot 10^{-7}$	$6.549429 \cdot 10^{-12}$	$1.676122 \cdot 10^{-7}$	0	0
0.1	0.8572790	$5.715120 \cdot 10^{-3}$	$9.104798 \cdot 10^{-1}$	$7.678877 \cdot 10^{-4}$	0.920248	$1.146455 \cdot 10^{-11}$
0.2	1.889560	$5.659021 \cdot 10^{-3}$	1.938253	$7.678877 \cdot 10^{-4}$	1.948732	$2.201211 \cdot 10^{-10}$
0.3	2.874624	$5.649265 \cdot 10^{-3}$	2.922637	$7.678877 \cdot 10^{-4}$	2.934418	$4.520701 \cdot 10^{-10}$
0.4	3.873151	$5.677204 \cdot 10^{-3}$	3.923429	$7.678877 \cdot 10^{-4}$	3.937182	$4.385381 \cdot 10^{-15}$
0.5	4.935926	$5.738197 \cdot 10^{-3}$	4.991439	$7.678877 \cdot 10^{-4}$	5.008065	$6.661338 \cdot 10^{-16}$
0.6	6.122372	$5.835228 \cdot 10^{-3}$	6.187313	$7.678877 \cdot 10^{-4}$	6.208272	$2.131628 \cdot 10^{-14}$
0.7	7.525945	$5.980367 \cdot 10^{-3}$	7.608010	$7.678877 \cdot 10^{-4}$	7.636113	$2.431388 \cdot 10^{-14}$
0.8	9.337474	$6.205006 \cdot 10^{-3}$	9.455316	$7.678877 \cdot 10^{-4}$	9.497396	$2.247214 \cdot 10^{-11}$
0.9	12.11616	$6.610164 \cdot 10^{-3}$	12.34297	$7.678877 \cdot 10^{-4}$	12.425561	$4.440892 \cdot 10^{-16}$

Table 8: Results for $\xi = 5$, $n = 700$.

α	Euler	Euler error	RK	RK error	Python	Python error
0	$5.376117 \cdot 10^{-13}$	$3.941867 \cdot 10^{-9}$	$5.376117 \cdot 10^{-13}$	$3.941867 \cdot 10^{-9}$	0	0
0.1	3.462682	$3.447892 \cdot 10^{-3}$	3.516705	$9.512708 \cdot 10^{-4}$	3.540201	$3.716111 \cdot 10^{-12}$
0.2	5.331328	$3.256520 \cdot 10^{-3}$	5.369665	$9.512708 \cdot 10^{-4}$	5.390189	$7.877770 \cdot 10^{-10}$
0.3	6.909950	$3.190590 \cdot 10^{-3}$	6.943959	$9.512708 \cdot 10^{-4}$	6.964908	$6.691902 \cdot 10^{-10}$
0.4	8.413954	$3.183560 \cdot 10^{-3}$	8.447554	$9.512708 \cdot 10^{-4}$	8.470495	$6.263373 \cdot 10^{-11}$
0.5	9.949034	$3.218895 \cdot 10^{-3}$	9.984925	$9.512708 \cdot 10^{-4}$	10.011343	$8.104628 \cdot 10^{-15}$
0.68	$1.160836 \cdot 10$	$3.295244 \cdot 10^{-3}$	$1.164968 \cdot 10$	$9.512708 \cdot 10^{-4}$	11.681679	$5.053735 \cdot 10^{-13}$
0.7	$1.351930 \cdot 10$	$3.422631 \cdot 10^{-3}$	$1.357144 \cdot 10$	$9.512708 \cdot 10^{-4}$	13.612857	$3.994471 \cdot 10^{-12}$
0.8	$1.592806 \cdot 10$	$3.631483 \cdot 10^{-3}$	$1.600405 \cdot 10$	$9.512708 \cdot 10^{-4}$	16.064016	$3.060885 \cdot 10^{-13}$
0.9	$1.954726 \cdot 10$	$4.024102 \cdot 10^{-3}$	$1.969964 \cdot 10$	$9.512708 \cdot 10^{-4}$	19.813186	$8.035803 \cdot 10^{-10}$

Table 9: Results for $\xi = 10$, $n = 700$.

α	RK	RK error	Python	Python error
0	$3.622398 \cdot 10^{-15}$	$2.180185 \cdot 10^{-12}$	0	0
0.1	$1.020256 \cdot 10$	$6.262275 \cdot 10^{-4}$	10.183136	$5.061507 \cdot 10^{-13}$
0.2	$1.319687 \cdot 10$	$6.262275 \cdot 10^{-4}$	13.185356	$3.146344 \cdot 10^{-12}$
0.3	$1.559883 \cdot 10$	$6.262275 \cdot 10^{-4}$	15.590813	$1.165734 \cdot 10^{-14}$
0.4	$1.781186 \cdot 10$	$6.262275 \cdot 10^{-4}$	17.806128	$1.420982 \cdot 10^{-10}$
0.5	$2.001404 \cdot 10$	$6.262275 \cdot 10^{-4}$	20.010201	$1.182054 \cdot 10^{-12}$
0.6	$2.234552 \cdot 10$	$6.262275 \cdot 10^{-4}$	22.343550	$1.462971 \cdot 10^{-9}$
0.7	$2.498339 \cdot 10$	$6.262275 \cdot 10^{-4}$	24.983695	$2.135776 \cdot 10^{-10}$
0.8	$2.825981 \cdot 10$	$6.262275 \cdot 10^{-4}$	28.263685	$1.324163 \cdot 10^{-12}$
0.9	$3.314609 \cdot 10$	$6.262275 \cdot 10^{-4}$	33.158800	$2.261101 \cdot 10^{-10}$

Table 10: Results for $\xi = 20$, $n = 1100$.

α	RK	RK error	Python	Python error
0	$2.440750 \cdot 10^{-17}$	$1.205815 \cdot 10^{-15}$	0	0
0.1	$1.759761 \cdot 10$	$1.269456 \cdot 10^{-4}$	17.604876	$1.053324 \cdot 10^{-14}$
0.2	$2.148462 \cdot 10$	$1.269456 \cdot 10^{-4}$	21.490484	$3.136380 \cdot 10^{-15}$
0.3	$2.452686 \cdot 10$	$1.269456 \cdot 10^{-4}$	24.532613	$1.665335 \cdot 10^{-16}$
0.4	$2.728547 \cdot 10$	$1.269456 \cdot 10^{-4}$	27.291611	$1.424549 \cdot 10^{-11}$
0.5	$2.999651 \cdot 10$	$1.269456 \cdot 10^{-4}$	30.003433	$2.997602 \cdot 10^{-15}$
0.6	$3.283570 \cdot 10$	$1.269456 \cdot 10^{-4}$	32.843915	$5.309253 \cdot 10^{-11}$
0.7	$3.601513 \cdot 10$	$1.269456 \cdot 10^{-4}$	36.025560	$7.509434 \cdot 10^{-10}$
0.8	$3.992210 \cdot 10$	$1.269456 \cdot 10^{-4}$	39.936859	$1.173403 \cdot 10^{-9}$
0.9	$4.567130 \cdot 10$	$1.269456 \cdot 10^{-4}$	45.698419	$1.293632 \cdot 10^{-12}$

Table 11: Results for $\xi = 30$, $n = 4000$.

α	RK	RK error	Python	Python error
0	$1.644573 \cdot 10^{-19}$	$6.569244 \cdot 10^{-19}$	0	0
0.1	$2.544202 \cdot 10$	$1.763321 \cdot 10^{-4}$	25.431916	$1.249237 \cdot 10^{-12}$
0.2	$3.006947 \cdot 10$	$1.763321 \cdot 10^{-4}$	30.062605	$8.651135 \cdot 10^{-13}$
0.3	$3.364768 \cdot 10$	$1.763321 \cdot 10^{-4}$	33.641846	$1.448841 \cdot 10^{-14}$
0.4	$3.686511 \cdot 10$	$1.763321 \cdot 10^{-4}$	36.859632	$5.693512 \cdot 10^{-11}$
0.5	$4.000559 \cdot 10$	$1.763321 \cdot 10^{-4}$	40.000079	$6.563439 \cdot 10^{-10}$
0.6	$4.327481 \cdot 10$	$1.763321 \cdot 10^{-4}$	43.268902	$2.553513 \cdot 10^{-15}$
0.7	$4.691514 \cdot 10$	$1.763321 \cdot 10^{-4}$	46.908334	$2.575717 \cdot 10^{-14}$
0.8	$5.136312 \cdot 10$	$1.763321 \cdot 10^{-4}$	51.354291	$1.862066 \cdot 10^{-12}$
0.9	$5.786816 \cdot 10$	$1.763321 \cdot 10^{-4}$	57.853252	$9.709936 \cdot 10^{-10}$

Table 12: Results for $\xi = 40$, $n = 200000$.

References

[1] M. Abramowitz, I. A. Stegun. *Handbook of Mathematical Functions.* National Bureau of Standards Applied Mathematics (1964), pp. 942-943.

[2] R. P. Brent. *An Algorithm with Guaranteed Convergence for Finding a Zero of a Function. Algorithms Minimization without Derivatives.* Englewood Cliffs, NJ: Prentice-Hall (1973).

[3] J. L. Devore. *Probability and Statistics for Engineering and the Sciences.* Cengage Learning (2010), pp. 137-152.

[4] W. Gilchrist. *Statistical Modelling with Quantiles Functions.* CRC Press (2000), pp. 9-16. London.

[5] D. Griffiths, D. Higham. *Numerical Methods for Ordinary Differential Equations. Initial Value Problems.* Springer (2010), pp. 19-29, pp.123-140. London.

[6] A. K. Kaw, E. E. Kalu, D. Nguyen. *Numerical Methods with Applications.* http://nm.mathforcollege.com/topics/textbook_index.html

[7] R. Khoury, D. Harder. *Numerical Methods and Modelling for Engineering.* Springer (2016), pp. 120-124.

[8] L. P. Lara, M. Gadella. *An approximation to solutions of linear ODE by cubic interpolation.* Computers and Mathematics with Applications (2008), pp. 1488-1495.

[9] T. Luu. *Fast and accurate parallel computation of quantile functions for random number generation.* UCL Department of Mathematics (2016), pp. 71-84.

[10] A. J. McNeil, R. Frey, P. Embrechts. *Quantitative Risk Management. Concepts, Techniques and Tools.* Princeton University Press (2005), pp. 1-15, pp. 25-48, pp. 238-248. New Jersey.

[11] L. Ortiz-Gracia. *ES computation with multiple control variates.* Applied Mathematics and Computation (2020).

[12] L. Ortiz-Gracia, C. W. Oosterlee. *Efficient VaR and ES computations for nonlinear portfolios within the delta-gamma approach.* Applied Mathematics and Computation (2014), pp. 16-31.

[13] G. Steinbrecherand, W. T. Shaw. *Quantile mechanics.* European Journal of Applied Mathematics (2008), vol. 19, pp. 87–112.

[14] A. Steinherr. *Derivatives. The Wild Beast of Finance.* Wiley (1998).

[15] https://core.ac.uk/download/pdf/41787448.pdf

[16] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ncx2.html

[17] https://github.com/scipy/scipy/blob/master/scipy/special/cdflib/cdfchn.f