The Standard Market Risk Model of the Swiss Solvency Test: An Analytic Solution[*]

Andras Niedermayer[†]

November 17, 2018

Abstract

The full standard model of the Swiss Solvency Test requires a Monte Carlo simulation to calculate the regulatory target capital. This paper derives an alternative Fast Fourier transform based computational approach for calculating the target capital of the Swiss Solvency Test that is more than 600 times faster than a Monte Carlo simulation. We also show that the relative computational error of our approach is much smaller than that of the Saddlepoint Approximation method: the error of our approach is less than 10^{-8}, whereas for the Saddlepoint Approach it can be as large as 24% for the numerical examples we consider. Our algorithm is relevant for applications requiring both speed and precision, such as multi-period Swiss Solvency Test analysis, portfolio optimization, and more generally also for non-Swiss insurers' various economic asset and liability management applications where the expected shortfall of assets' and liabilities' fluctuations needs to be calculated in a fast and precise way.

JEL Codes: C63, C88, G22

Keywords: solvency regulation, capital adequacy requirements, computational methods, asset and liability management, portfolio optimization

Nr. words abstract: 155, Nr. words document: 4522, Nr. figures: 3, Nr. tables: 1

[*] I thank the editor and an anonymous referee for helpful comments. This article is dedicated to the memory of Ferenc Niedermayer.

[†] Université Paris-Dauphine, PSL Research University, LEDa, 75016 Paris, France, and SolvencyAnalytics AG (solvencyanalytics.com). Email: andras.niedermayer@dauphine.fr. The author reports no conflicts of interest. The author alone is responsible for the content and writing of the paper.

1 Introduction

The Swiss Solvency Test (SST) has been in force since 2011 and defines capital adequacy requirements for Swiss insurance companies. The standard SST model applied by the vast majority of insurers is defined by the Swiss Financial Market Supervisory Authority (FINMA) with results published yearly (see e.g. FINMA (2010)).

The market model within the standard SST framework defines a model to calculate an insurer's target capital as a function of stochastic market factors and related fluctuations in insurance assets and liabilities. This model can be applied in the simplified (linear) or in its full (quadratic) version. The simplified model is straightforward to apply as the problem is reduced to a normal distribution.[1] The full model's target capital, however, does not rely on a normal distribution of assets and liabilities and therefore a Monte Carlo simulation typically needs to be implemented as closed form solutions are not available. The full model is particularly important for insurance companies who want their SST or ALM model to reflect the non-linear pricing of their assets and liabilities. Non-linear pricing is particularly relevant for insurance companies with long assets and liabilities maturities where fixed income convexity requires more accurate risk modelling or for insurers holding investments with optionalities including options, convertible bonds, and certain structured products.

While the Monte Carlo algorithm in the SST guidelines is easy to implement, it either provides an insufficient level of precision or takes too long to compute for many applications, such as multiple evaluation of the target capital and interactive comparisons. For such applications, a fast and pre-

[1] See also the Excel spread sheet provided on FINMA's website for the calculation based on the simplified model.

cise evaluation of the target capital is essential. Moreover, the computational speed can be relevant for EU-based insurance companies in the context of their *own risk and solvency assessment* (ORSA) who want to apply an economic solvency model in a forward-looking, multi-period framework (see e.g. EIOPA (2010)).[2] In this paper we develop algorithm based on Fast Fourier Transformation that provides a solution to this problem.

To the best of our knowledge, this paper is the first to describe a Fast Fourier Transformation based algorithm for the computation of the Swiss Solvency Test capital requirements. Other aspects of the Swiss Solvency Test, such as its market implications (Eling *et al.*, 2008), its coherence as a risk measure (Filipović and Vogelpoth, 2008), and how it compares to the United States RBC Standard and the European Union's Solvency II (Holzmüller, 2009) have been discussed in the literature.

2 Specification of the Swiss Solvency Test

Here we briefly summarize the requirements specified by the Swiss Financial Market Supervisory Authority (FINMA). See Keller and Luder (2004) for more details.

There is a vector x of n factors that influences the development of the prices of assets in an insurance company's portfolio. x is assumed to be normally distributed with mean μ and covariance matrix Σ. Formally, $x \sim \mathcal{N}(\mu, \Sigma)$, that is the density of x is $p(x) \propto \exp(-\rho(x))$ where

$$\rho(x) = \frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\,. \tag{1}$$

As a first step ignore the possibility of "rare events" such as financial

[2] A multi-period risk framework takes into account not just the probability of insolvency next year, but also in subsequent years.

crises occurring. In this "normal scenario" the value of an asset j is

$$y_j(x) = \frac{1}{2}x'\Gamma_j x + \delta_j' x + c_j \, ,$$

where Γ_j and δ_j are the quadratic and linear coefficients of the impact of the risk factors x on the value of the asset y_j. Γ_j and δ_j can be thought of as the first two terms of a Taylor series. See the FINMA guidelines on how to compute Γ_j, δ_j and c_j for different assets.

A normal distribution does not take into account that "rare events" can occur which lead to exceptionally high losses in financial markets. To take this into account, FINMA has amended the model by multiple scenarios, which occur with small probabilities and are mutually exclusive. The probability of scenario i is p_i, where $i = 0$ stands for the "normal scenario" and $\sum_{i=0}^{K} p_i = 1$. For scenarios $i \geq 1$ the value of an asset has to be modified by the additive term $\bar{x}_i'\Gamma_j\bar{x}_i + \delta_j\bar{x}_i =: S_{ij}$, the factor changes for a scenario $\bar{x}_i$ being defined in the guidelines of FINMA.

The scenario adjusted value of an asset $Y(x)$ is hence given by

$$Y_j(x) = y_j(x) + \sum_i I_i S_{ij} \, ,$$

where the indicator random variables I_i select which scenario occurs. Formally, with probability p_i, $I_i = 1$ and $I_k = 0$ for all $k \neq i$, that is, the random variables are mutually exclusive. The indicator variables I_i are independent of x.

What matters at the end of the day is the sum of the changes of values for all assets, i.e.

$$Y(x) = \sum_j Y_j(x) \, .$$

The target capital (TC) is the Expected Shortfall of an insurance company's change in risk bearing capital, i.e. the average loss that occurs with a probability less than 1%. Formally, this is defined as follows. Define the distribution of the variable $Y(x)$ as $F_T(Y)$, where F_T can be computed from the distributions of the random variables x and I_i. Further, define the threshold for the Expected Shortfall y_0 implicitly by $F_T(y_0) = 0.01$. Then the target capital is defined as

$$\mathrm{TC} = -\frac{1}{F_T(y_0)} \int_{-\infty}^{y_0} y f_T(y) \mathrm{d}y \,.$$

For the following analysis, it is useful to define the sums of Γ_j, δ_j, c_j and S_{ij},

$$\Gamma = \sum_j \Gamma_j u_j \,, \quad \delta = \sum_j \delta_j u_j \,, c = \sum_j c_j u_j \,, \quad s_i = \sum_j S_{ij} u_j \,.$$

Note that for the random variable $Y(x)$ only the sums matter, since

$$y(x) = \frac{1}{2} x' \Gamma x + \delta' x + c \,, \tag{2}$$

and

$$Y(x) = y(x) + \sum_i I_i s_i \,.$$

In the following, we will only work with the sums Γ, δ, c, s_i rather than with Γ_j, δ_j, c_j, S_{ij}.

Observe that if both Σ and Γ were the identity matrix (or the identity matrix times a constant), y would follow a χ^2 distribution for which efficient computational methods are well known. However, typically, this is not the case and one has to derive an algorithm that goes beyond χ^2 distributions.

First, we will consider the random variable $y(x)$ which is a quadratic function of the random variable x and derive an efficient computational method for the distribution of $y(x)$.

3 Efficient Computational of the Distribution of $y(x)$

3.1 Simultaneous Diagonalization of Σ and Γ

In the following, we will derive the linear transformations of the random variable x which ensure the simultaneous diagonalization of Σ and Γ. We will see later on that this diagonalization will allow for a much more efficient calculation of the distribution of $y(x)$.

The quadratic normal random variable y is constructed from x by[3]

$$y(x) = \frac{1}{2}x'\Gamma x + \delta' x + c\,.$$

One can diagonalize both quadratic forms simultaneously through linear transformations of x as follows. In the following we will drop the argument in $y(x)$ and simply write y. As a first step, we make a Cholesky decomposition

$$\Sigma = \Lambda\Lambda'\,,$$

and introduce the random variable ξ through

$$x = \Lambda\xi + \mu\,.$$

With this we obtain

$$\rho(\xi) = \frac{1}{2}\xi'\xi\,,$$

and

$$y = \frac{1}{2}\xi'\hat{\Gamma}\xi + \xi'\hat{\delta} + \hat{c}\,,$$

[3] In the actual problem $c = 0$ but we keep it for generality, since it does not cause any complication.

where

$$
\begin{aligned}
\hat{\Gamma} &= \Lambda' \Gamma \Lambda \,, \\
\hat{\delta} &= \Lambda' \left(\Gamma \mu + \delta \right) \,, \\
\hat{c} &= \frac{1}{2} \mu' \Gamma \mu + \mu' \delta + c \,.
\end{aligned}
$$

Now one can find an orthogonal transformation which diagonalizes $\hat{\Gamma}$:

$$O' \hat{\Gamma} O = A \,.$$

Here O is an orthogonal matrix and A is a diagonal one, with diagonal elements a_k, i.e.

$$A = \begin{bmatrix} a_1 & 0 & ... & 0 \\ 0 & a_2 & ... & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & ... & a_n \end{bmatrix} \,.$$

Introducing

$$\xi = O \eta \,,$$

we have

$$\rho(\eta) = \frac{1}{2} \eta' \eta = \frac{1}{2} \sum_k \eta_k^2 \,. \tag{3}$$

and

$$y = \frac{1}{2} \eta' A \eta + b' \eta + \hat{c} \,,$$

where

$$b = O' \hat{\delta} \,.$$

Equation (3) shows that η_k are independent standard normally distributed random variables. Since A is diagonal, y is a sum independent random variables, which is useful for using a Fourier transform.[4]

[4] In this case, the random variables are independent non-central χ^2 random variables of degrees of freedom 1.

3.2 The Characteristic Function of y

As can be seen from (3), η follows an independent standard normal distribution, formally $\eta \sim \mathcal{N}(0, I)$. In the following, we will derive the characteristic function of the random variable y in terms of η.

The constant c has the effect of shifting Y, hence to simplify the calculations we can perform the calculations without c and add it only at the last step. The density of y is given then by

$$f(y) = \int \left(\prod_{l=1}^{n} \frac{\mathrm{d}\eta_l}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\eta_l^2} \right) \delta \left(\frac{1}{2} \sum_{k=1}^{n} a_k \eta_k^2 + \sum_{k=1}^{n} b_k \eta_k - y \right) ,$$

where b_k are the elements of b.

Introducing the characteristic function (the Fourier transform) of $f(y)$

$$\varphi(t) = \int_{-\infty}^{\infty} \mathrm{d}y \, \mathrm{e}^{ity} f(y) \, , \tag{4}$$

one has

$$f(y) = \int_{-\infty}^{\infty} \frac{\mathrm{d}t}{2\pi} \, \mathrm{e}^{-ity} \varphi(t) \, . \tag{5}$$

Note that

$$\delta(z) = \int_{-\infty}^{\infty} \frac{\mathrm{d}t}{2\pi} \mathrm{e}^{-itz} \, .$$

Since the variables η_k are independent one obtains

$$\varphi(t) = \prod_{k} \phi(t; a_k, b_k) \, , \tag{6}$$

where

$$\begin{aligned} \phi(t; \alpha, \beta) &= \int_{-\infty}^{\infty} \frac{\mathrm{d}\eta}{\sqrt{2\pi}} \exp \left(-\frac{1}{2}\eta^2 + it \left[\frac{1}{2}\alpha\eta^2 + \beta\eta \right] \right) \\ &= \frac{1}{\sqrt{1 - it\alpha}} \exp \left(-\frac{t^2\beta^2}{2(1 - it\alpha)} \right) \, . \end{aligned}$$

Note that the square root of the complex number $(1 - ita_k)$ should be evaluated for each k separately in (6). Taking the square root of the product is dangerous because of the ambiguity of the sign.

When computing the Fourier transform and the inverse Fourier transform, for practical purposes one has to use a discretized version on a finite interval, both for y and for t. We take a grid with N points on the intervals of both y and t. Denoting the corresponding grid points by y_j and t_l, we choose

$$\begin{aligned} y_j &= E[y] + j\Delta y\,, && j = -N/2, \ldots, N/2-1\,, \\ t_l &= l\Delta t\,, && l = -N/2, \ldots, N/2-1\,, \end{aligned}$$

where the expected value is $E[y] = \frac{1}{2}\sum_k a_k$. The step sizes are chosen to be $\Delta y = Q/\sqrt{N}$ and $\Delta t = 2\pi/(Q\sqrt{N})$, where Q is a parameter the choice of which we will discuss in the following. For any constant Q both Δy and Δt go to zero as $\propto 1/\sqrt{N}$ for $N \to \infty$. At the same time, the lower and upper bounds of the intervals go to $-\infty$ and $+\infty$, respectively, since the intervals are $\left[E[y] - Q\sqrt{N}/2, E[y] + Q\sqrt{N}/2\right]$ for y and $\left[-2\pi\sqrt{N}/Q, 2\pi\sqrt{N}/Q\right]$ for t, respectively.

While any Q would do (since in the limit of $N \to \infty$ the error disappears), a natural choice is $Q \approx \sigma_y \approx 1/\sigma_t$, where σ_y and σ_t are the standard deviations of y and t, respectively. This procedure is invariant to a rescaling of y (and correpondingly t).

The computational error can be estimated in a similar way as for adaptive quadrature algorithms: by using our algorithm for a grid size $N_1 = N$ and another grid size $N_2 = 2N$ and taking the difference between the two results.

4 Checking the Results

For the general case, the distribution of y can be easily simulated by Monte Carlo, although that takes considerably more time if a small error is needed.

Some checks are shown in Figures 1 and 2. Figure 1 shows that the density computed using the Fourier transform based algorithm is very close (indeed graphically indistinguishable in this case) from the exact density, which can be computed in this case analytically. Conversely, the Monte Carlo algorithm does have an approximation error. Figure 2 takes an example in which no analytical solution is available and shows that the Fourier transform based algorithm and the Monte Carlo algorithm similar densities.

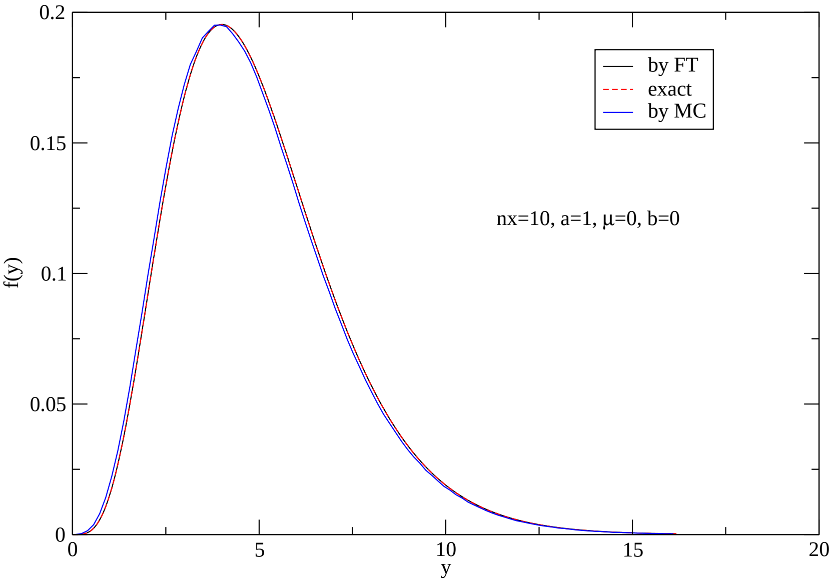

Figure 1: $n = 10$, $a_k = 1$, $\mu = 0$, $b = 0$.

5 Cumulative Distribution Function

We also need the cumulative distribution function $F(y)$. We use the following approximation for $y_i \leq y \leq y_{i+1}$:

$$F(y) = F(y_i) + \int_{y_i}^{y} \overline{f}_i(u)\,\mathrm{d}u\,,$$

where $\overline{f}_i(u)$ is the polynomial approximation in the interval $[y_i, y_{i+1}]$, and $F(y_{-N/2}) = 0$.

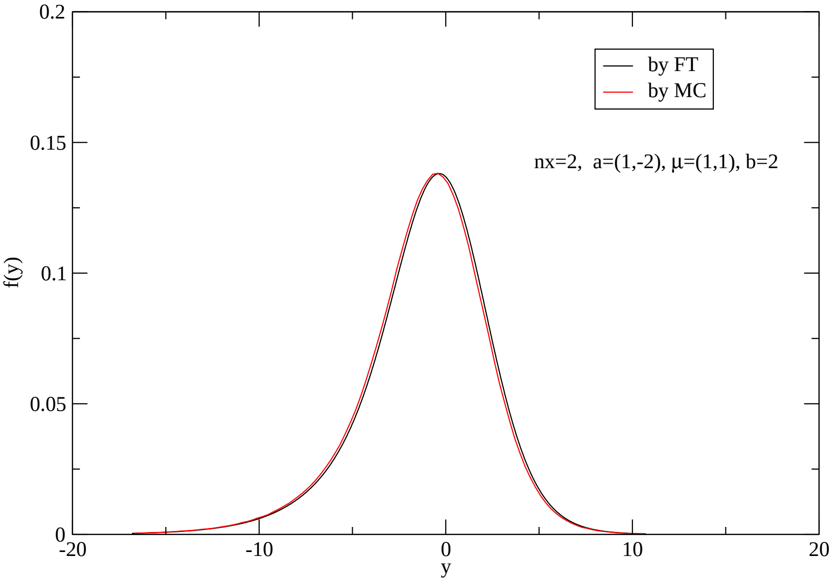

Figure 2: $n = 2$, $a = (1, -2)$, $\mu = (1, 1)$, $b = 2$

Note that alternatively one can also calculate $F(y)$ by the Gil-Pelaes formula (see Davies (1980))

$$F(y) = \frac{1}{2} - \int_{-\infty}^{\infty} \frac{\mathrm{d}t}{2\pi t} \mathrm{Im}\left(\mathrm{e}^{-ity}\varphi(t)\right) .$$

Our analysis indicates, however, that the discretization errors are larger for this choice.

6 Multiple Scenarios

We discuss here a model with the random variable

$$Y = y + v ,$$

where y is distributed according to (1) and (2) and v is independent of y and has a probability distribution

$$f_c(v) = p_0\delta(v) + \sum_{i=1}^{K} p_i\delta(v - s_i) ,$$

where

$$p_0 + \sum_{i=1}^{K} p_i = 1\,,$$

and δ is the Dirac delta function.

Here i labels the different scenarios, $i = 0$ corresponding to a "normal year" (the shift v is $s_0 = 0$).

The total probability distribution of Y is given by the convolution

$$f_T(Y) = \int dy\, dv\, f(y) f_c(v) \delta(y + v - Y) = \sum_{i=0}^{K} p_i f(Y - s_i)\,.$$

Since there is a small number of scenarios it is easy to calculate $f_T(Y)$ once $f(y)$ is known.

7 Putting the Pieces Together

Once one can calculate $f_T(Y)$ efficiently, it is easy to compute the target capital TC. Define the cutoff y_0 as the first percentile, i.e. $F_T(y_0) = 0.01$. Then the target capital can be computed as

$$\mathrm{TC} = -\frac{1}{F_T(y_0)} \int_{-\infty}^{y_0} y f_T(y) \mathrm{d}y\,.$$

The integral can be computed efficiently using standard numerical analysis techniques such as Newton-Cotes formulas or Gaussian quadrature.

We approximate the continuous Fourier integrals in (4) and (5) by discrete Fourier transform. This allows us to use the Fast Fourier Transform algorithm, which has the desirable property that for a grid with N points the computational time increases with $N \ln N$ rather than with N^2 for the naive evaluation of the integral. This leads to considerable speed-up for large values of N, which are necessary for a good approximation.

8 Performance Comparison

We compare the performance of the Fast Fourier transform based computation to the performance of the Monte Carlo method in the following. Computations were done on a laptop with an Intel Core i7-7500U CPU with 2 cores and 2.70 GHz and 16 GB RAM. The Fast Fourier transform based computation used a grid with 2^{16} grid points and needed 0.1209 seconds to compute the target capital. We have estimated the computational error of the Fast Fourier transform based calculation by making the same calculation with 2^{15} grid points and taking the difference between the result of the small and the large grid size. The Monte Carlo based algorithm was used with different numbers of random draws M, with $M = 10^4, 3 \times 10^4, 10^5, 3 \times 10^5, 10^6, 3 \times 10^6, 10^7$. We have used a bootstrap method to compute the 95% confidence interval for the results from the Monte Carlo simulation. For the performance benchmark we have assumed $n = 20$ factors and have chosen $\mu = 0$, $\Sigma = I$, $\Gamma = I$, $\delta = 0$, $c = 0$ for the sake of simplicity.[5]

Results are reported in Figure 3. The Figure reports the SST ratio rather than the target capital since this is the measure relevant in practice. The SST ratio is defined as the ratio of risk bearing capital to target capital. The calculations underlying the Figure assumes a risk bearing capital of 5.4. The computational error for the Fourier transform based algorithm is 1.85×10^{-13} and hence too small to be shown in the Figure. The algorithm achieves this precision already after 0.1209 seconds. In contrast to this the Monte Carlo simulation needs more than 80 seconds to achieve a reasonable

[5] This simple example shortens the computational time of the Fast Fourier algorithm because no Cholesky decompositions of Σ and Γ have to be computed. However, this effect is negligible, since the Cholesky decomposition for a 20 by 20 matrix on the same hardware takes less than 0.000018 seconds to compute, which is less than 0.02% of the computational time of the Fast Fourier based algorithm.

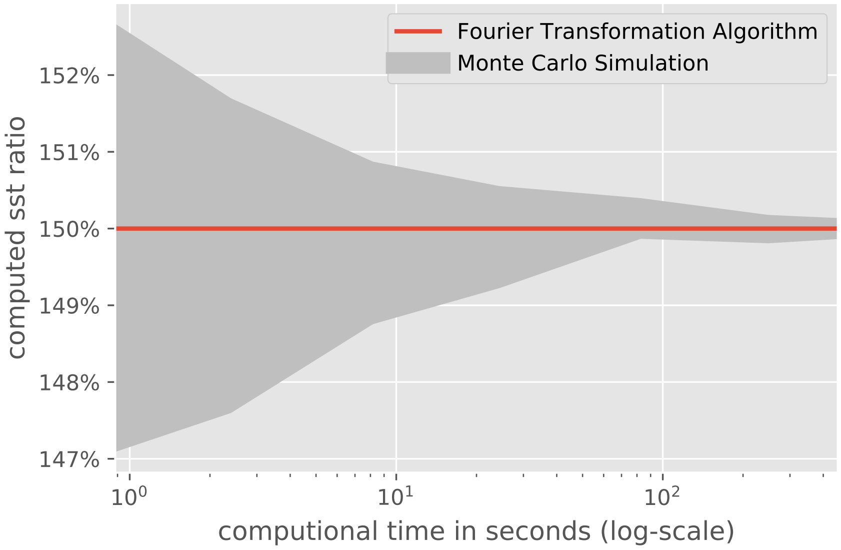

Figure 3: Computational time versus precision for the Fast Fourier Transformation Algorithm and for the Monte Carlo based algorithm

level of precision. This constitutes a speed up of more than a factor 600.

In Appendix A we compare our Fourier transform based algorithm with another commonly used algorithm for computing expected shortfalls: the Saddlepoint Approximation.

9 Conclusions

We have shown a fast and precise algorithm for the computation of the SST target capital. Our benchmarks suggest that our algorithm outperforms the standard Monte Carlo algorithm by more than a factor 600. This increase in speed opens the possibility for new applications of the Swiss Solvency Test, such as the computation of the required target capital for a large number of market scenarios, portfolio optimization under regulatory constraints and multi-period risk frameworks such as the ORSA.

References

Broda, S. A. and Paolella, M. S. (2011). Expected shortfall for distributions in finance. In P. Cizek, W. K. Härdle and R. Weron (eds.), *Statistical tools for finance and insurance*, Springer, pp. 57–99, doi: 10.1007/978-3-642-18062-0_2.

Davies, R. B. (1980). The distribution of a linear combination of χ^2 random variables. *Applied Statistics*, **29** (3), 323–333, doi: 10.2307/2346911.

EIOPA (2010). Guidelines on Own Risk and Solvency Assessment. Available at https://eiopa.europa.eu/GuidelinesSII/EIOPA_Guidelines_on_ORSA_EN.pdf.

Eling, M., Gatzert, N. and Schmeiser, H. (2008). The Swiss Solvency Test and its market implications. *The Geneva Papers on Risk and Insurance-Issues and Practice*, **33** (3), 418–439, doi: https://doi.org/10.1057/gpp.2008.20.

Filipović, D. and Vogelpoth, N. (2008). A note on the Swiss Solvency Test risk measure. *Insurance: Mathematics and Economics*, **42** (3), 897–902, doi: https://doi.org/10.1016/j.insmatheco.2007.10.009.

FINMA (2010). SST 2017 Survey: FINMA Report on the Swiss Insurance Market . Available at https://www.finma.ch/en/~/media/finma/dokumente/dokumentencenter/myfinma/2ueberwachung/sst/results-sst-2017.pdf?la=en.

Holzmüller, I. (2009). The United States RBC standards, Solvency II and the Swiss Solvency Test: A comparative assessment. *The Geneva Papers* *on Risk and Insurance-Issues and Practice*, **34** (1), 56–77, doi: `https://doi.org/10.1057/gpp.2008.43`.

Keller, P. and Luder, T. (2004). White paper of the Swiss Solvency Test. *Swiss Federal Office of Private Insurance, Switzerland*, **12**, available at `https://www.finma.ch/FinmaArchiv/bpv/download/e/WhitePaperSST_en.pdf`.

Lugannani, R. and Rice, S. (1980). Saddle point approximation for the distribution of the sum of independent random variables. *Advances in applied probability*, **12** (2), 475–490, doi: `https://doi.org/10.2307/1426607`.