1 Simulations

1.1 Set-Up

The aim of the simulation exercises is to measure the performance of the designed optimization algorithm in recovering the parameters that explain the stochastic intensity process and the CDS pricing mechanisms. These exercises are conducted with different sample sizes to demonstrate the asymptotic properties and robustness of the estimator. The parameters are estimated through the minimization process of the objective function which is the sum of squared differences between the estimated and observed CDS prices across the different dates and maturities.

$$\sum_{T=1} \sum_{t=1} (\hat{s}_{t,T} - s_{t,T})^2$$

where T is the maturity of the CDS contract and t represents the days where CDS price data observed. Since this simulation exercise concerns only the systemic part of the credit risk, the estimated CDS price formula in Ang and Longstaff (2013) will be simplified by eliminating the terms which represents the idiosyncratic risk. The CDS pricing formula which is explained by both systemic and idiosyncratic risk components is

$$\hat{s}_T^i = \frac{wE\left[\int_0^T D(t)(\gamma_i\lambda_t + \xi_{it}) \exp(-\int_0^t \gamma_i\lambda_t + \xi_{is} ds) dt\right]}{E\left[\int_0^T D(t) \exp(-\int_0^t \gamma_i\lambda_t + \xi_{is} ds) dt\right]} = \frac{w\int_0^T D(t)(A(\lambda,t)C(\xi_i,t)) + \gamma_i B(\xi_i,t)F(\lambda,t))dt}{\int_0^T D(t)A(\lambda,t)B(\xi_i,t)dt}$$

where i represents the sovereign, γ_i is the default probability of a sovereign after a systemic shock occurs, λ_t is the the global systemic default intensity and ξ_{it} represents the idiosyncratic default intensity for sovereign i. Lastly, $D(t)$ is the value of a riskless zero-coupon bond with maturity t ($D(t) = E[e^{-\int_0^T r_t dt}]$). The right hand side of the equation shows the closed form solution where the expressions for $A(\lambda, t)$, $B(\xi, t)$, $C(\xi, t)$ and $F(\lambda, t)$ are given in Appendix. λ and ξ_i represent the current values of the concerning intensity processes.

To eliminate the effect of idiosyncratic risk from the CDS pricing formula, we equate the ξ_{it} to zero for all i and t. This implies that the mean of the standard square-root process for the idiosyncratic intensity will be equal to zero as well. Applying these changes to the both of the equations above we have

$$\hat{s}_T^i = \frac{w\int_0^T D(t)\gamma_i F(\lambda,t)dt}{\int_0^T D(t)A(\lambda,t)dt} = \frac{wE[\int_0^T D(t)\gamma_i\lambda_t e^{-\int_0^t \gamma_i\lambda_s ds} dt]}{E[\int_0^T D(t) e^{-\int_0^t \gamma_i\lambda_s ds} dt]},$$

which is explained in 5.1.

As the goal here is only to examine the source of the systemic risk, but not the effect of it on the sovereigns, we fix $\gamma_i = 1$. Also without a need for the index i in this case, the CDS price formula becomes

$$\hat{s}_T = \frac{w\int_0^T D(t)F(\lambda,t)dt}{\int_0^T D(t)A(\lambda,t)dt} = \frac{wE[\int_0^T D(t)\lambda_t e^{-\int_0^t \lambda_s ds} dt]}{E[\int_0^T D(t) e^{-\int_0^t \lambda_s ds} dt]}. \quad (1)$$

In this simulation exercise, observed CDS prices are generated artificially. This provides the advantage of comparing the estimated parameters with the true ones and to verify the reliability of the model which normally would be impossible to do in real-life examples.

The parameters of interest (α, β, σ^2) also determines the behavior of the daily intensity process which follows a standard square root model.

$$d\lambda_t = (\alpha - \beta\lambda_t)dt + \sigma\sqrt{\lambda_t}dZ_{\lambda t}. \tag{2}$$

β and σ represent the speed of mean reversion and the volatility respectively, the mean of the process can be obtained by the ratio of α and β.

1.2 Data Generation

As the CDS price equation is a function of the parameters and the intensity value, we should determine the true parameters and create the daily intensity values in accordance with the chosen parameters. For this simulation exercise, all the parameters (α, β, σ^2) are chosen as 0.3 and the initial point of the intensity process (λ_0) is fixed to 0.2. By using the standard square-root model given at equation 2, daily intensity values are calculated for one year horizon and it is assumed that there are N number of observation days in a year.

After the creation of the last missing element in equation 1, intensity values for N days, CDS price data are estimated for five different maturities (T $\in$ {1,2,3,4,5}) under the risk-neutral measure. The interest rate is assumed to be fixed at 0.03 for the sake of simplicity in the simulations. For each data point, a normally distributed random error with a mean of zero and a standard deviation corresponding to one percent of the value of the data point added to the true value. To sum up, there are 5xN amount of data points in the system.

1.3 Recovering the Parameters through Optimization

In the optimization process, the econometrician observes the artificially created CDS prices as observed data, however, there is no knowledge of the true parameters and the true intensity values prior to the optimization. As these are the determinants of the observed data, one should somehow disentangle the information of the parameters and the intensity values from these data points. Intensity being a function of the parameters, one can propose replacing the intensity with the parameters and then conducting the optimization to determine the parameters. Nevertheless, the standard square root process does not have a closed form solution, therefore, it is challenging to replace the intensity values in the CDS price equation. Another idea would be to combine two different methods in each iteration where one leg optimizes the error with respect to the intensity values, and the other one conducts the optimization with respect to the parameters as in Ang and Longstaff (2013), and Brownlees *et al.* (2021).

In the first leg of an iteration, with some estimated parameter values at hand, each daily intensity value is bootstrapped by minimizing the sum of squared errors of five different maturities by using non-linear least squares. This process creates intensity values for each N days in the year. In the second leg, a gradient descent algorithm runs to optimize the parameters. That is, the algorithm first calculates the possible marginal improvements in the error resulting from updating each parameters, in other words, the derivatives of the error function with respect to parameters. Then, it chooses the one with the maximum improvement and updates the relevant parameter by subtracting a quantity corresponding to predetermined step size times the value of improvement. After updating the concerning parameter, a new iteration starts and the first leg of the iteration takes place to bootstrap intensity values with the new information, namely with the updated parameter inputs. If the error improvement in any of the legs are below a predetermined threshold, then the optimization algorithm will be terminated.

2 Gradient Descent Algorithm in Detail

(informal explanation)

2.1 Motivation

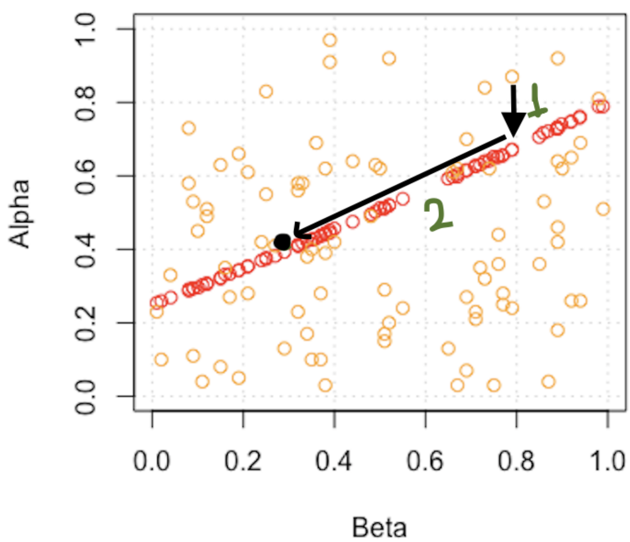


Figure 1: Convergence journey of a point where the black dot represents the true value.

In the old algorithm, where every parameter is updated with the same gradient descent rule, a randomly chosen point in the figure first goes vertically to the red line by only optimizing the alpha (Step 1). This process is quite fast. However the journey on the red line to the true parameter value takes a lot of iterations (Step 2). This is the same for Alpha-Sigma graph.

Therefore, there was a necessity to update the algorithm to make the Step 2 faster.

In the new algorithm:

- Alpha is optimized with traditional gradient descent (Step 1).

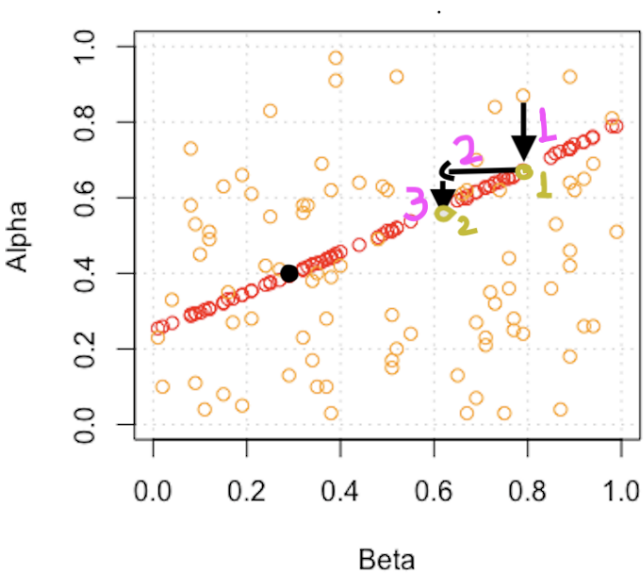


Figure 2: New algorithm description.

- When one of the other parameters started to be optimized, the algorithm checks the direction and update the parameter with a huge step size (e.g. 0.08) in that direction (Step 2).
- Alpha is optimized again to reach the red line in Step 3, which is Step 1 of the new iteration.

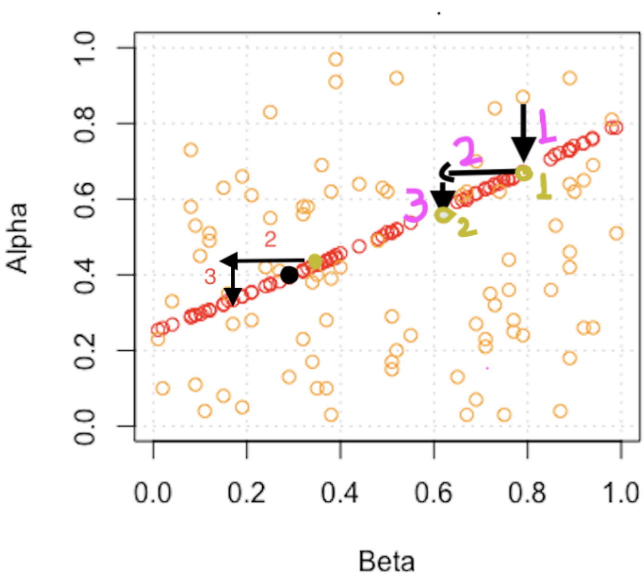


Figure 3: What if the parameter update worsen the error.

However, if the updated value following Step 2 and Step 3 provides a pricing error higher than the initial point (the example on the left of the Figure 3), then the process goes back to the beginning point of the last Step 2 and update the parameter again in the same direction with half of the step size (e.g. 0.04). And tries again.

2.2 Algorithm

- Calculate derivatives (contributions) of each parameters
- Choose the one with the highest absolute value

If α is chosen:

$$\alpha_{t+1} = \alpha_t - m_1 \nabla s_i(\alpha_t)$$

where m_1 is the step size determined for α.

If β or σ^2 is chosen:

- Save current parameters first (Θ_t)
- Save the current error (ϵ_t)
- Save which parameter is optimized now (β_t or σ_t^2)
- Save sign of the derivative (sign_t)

Update the parameter.

$$\theta_{t+1} = \theta_t - \text{sign}_t * m_2$$

where m_2 is the step size vector determined for β and σ^2.

- Continue to the process.

If α:

$$\alpha_{t+1} = \alpha_t - m_1 \nabla s_i(\alpha_t)$$

where m_1 is the step size determined for α.

If β or σ^2 & $\epsilon_{t+n} < \epsilon_t$:

- Save current parameters first (Θ_{t+n})
- Save the current error (ϵ_{t+n})
- Save which parameter is optimized now (β_{t+n} or σ_{t+n}^2)
- Save sign of the derivative (sign_{t+n})

Update the parameter.

$$\theta_{t+n+1} = \theta_{t+n} - \text{sign}_t * m_2$$

where m_2 is the step size vector determined for β and σ^2.

Otherwise

x

If β or σ^2 & $\epsilon_{t+n} > \epsilon_t$:

- Retrieve the saved parameters. (Θ_t)
- Retrieve the saved error (ϵ_t)
- Retrieve the optimized parameter (β_t or σ_t^2)
- Retrieve the saved sign of the derivative (sign_t)

Update the step size.

$$m_2 := m_2/2$$

Update the parameter.

$$\theta_{t+1} = \theta_t - \text{sign}_t * m_2$$

(So the second step is completely reversed. The optimization continues as if we updated the parameter of interest by $m_2/2$ instead of m_2.)

- Continue to the process.
- Stop if error improvement is lower than a threshold (e.g. Current Mean Sq. Error * 10^{-11}).

3 Simulation Results

Simulations are conducted with different sample sizes (N $\in$ {50, 100, 200}) to observe the asymptotic behavior of the estimator. For each sample size, we run the simulations 10.000 times, where the initial parameter points are chosen to be at the true parameter values, in this case all of the parameters are initialized at 0.3. For all cases λ_0 is set to 0.2, and the threshold value to terminate the optimization is set to observed mean squared error times 10^{-11}. The step size of the gradient descent algorithm which determines how fast the parameters are updated is set to $0.3/N$.

	N = 25	N = 50	N = 100	N = 200
Alpha	0.2998993	0.2997685	0.2996485	0.2994380
Beta	0.3006751	0.2994817	0.2994965	0.2990673
Sigma^2	0.2967337	0.2989490	0.2988946	0.2989515
Intensity Error	3.811447e-05	2.388515e-05	1.653401e-05	1.261728e-05
Pricing Error	3.897359e-06	3.733328e-06	3.643603e-06	3.592951e-06
Variance Measurement Error	3.783140e-06	3.673772e-06	3.613941e-06	3.578287e-06

Table 1: Parameter and Error Evolution

In Table 1, the mean values of the estimated parameters and the errors among 10K draws for each sample sizes are shown. Intensity error measures the mean squared differences between the true intensity values and the estimated intensity values. Pricing error (PE) is the mean squared differences between the true CDS prices and the estimated ones, while variance measurement error (VME) is the variance of the differences between the true CDS prices and the estimated ones. These measures are described in equations below.

- Intensity Error = $\frac{1}{N}\sum_t^N (SI_t$ - $BI_t)^2$,
- Pricing Error = $\frac{1}{N}\sum_t^N \epsilon_t^2$,
- Variance Measurement Error = $\frac{1}{N}\sum_t^N (\epsilon_t - \bar{\epsilon})^2$,

where SI_t and BI_t are the simulated intensity and bootstrapped intensity values in day t. ϵ_t represents the difference between the true CDS price and the estimated CDS price in day t, while $\bar{\epsilon}$ is the mean value of these differences.

One can expect that if the designed algorithm is an unbiased estimator of the mean, then the errors of estimated CDS prices has a mean of zero. This makes pricing error and the variance measurement error the same error estimators by definition. Therefore, the discrepancy between these two demonstrates how robust the estimator is.

The variance estimates and the mean squared errors of the estimated parameters are given in Tables 2 and 3 respectively. The discrepancy between the pricing error and the variance measurement error for each sample size is demonstrated in Table 4. As expected the accuracy of the parameters are improved as sample size increases. The descreasing discrepancy between the PE and VME as sample size increases signals that the estimator is asymptotically unbiased.

	N = 25	N = 50	N = 100	N = 200
Alpha	0.0005151625	0.0002641216	0.0001311677	0.0000678012
Beta	0.006846385	0.003431951	0.0017094139	0.0008723027
Sigma^2	0.0042017411	0.0020401068	0.001040559	0.0005174625

Table 2: Variances of Estimated Parameters

	N = 25	N = 50	N = 100	N = 200
Alpha	0.0005151221	0.0002641493	0.0001313783	0.0000681103
Beta	0.0068461691	0.0034318830	0.0017094998	0.0008730871
Sigma^2	0.0042119979	0.0020410114	0.0010416792	0.0005185110

Table 3: Mean Squared Errors of Estimated Parameters

	N = 25	N = 50	N = 100	N = 200
Discrepancy	$5.613*10^{-14}$	$1.451*10^{-14}$	$3.665*10^{-15}$	$8.843*10^{-16}$

Table 4: Mean Squared Differences of Pricing Error and Variance Measurement Error

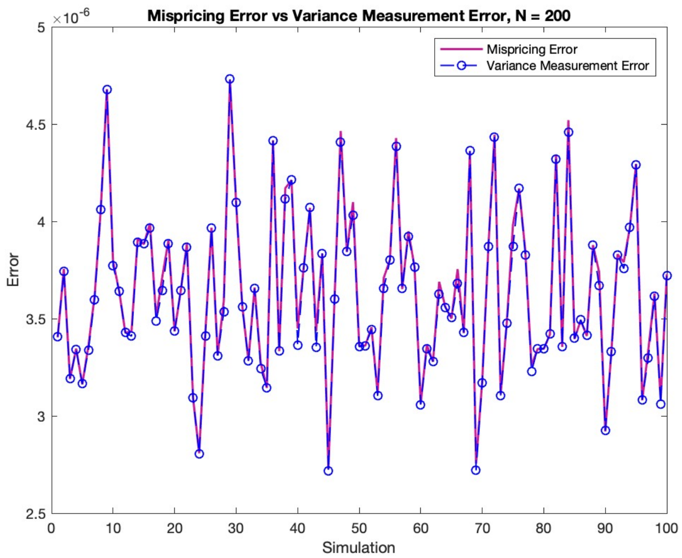
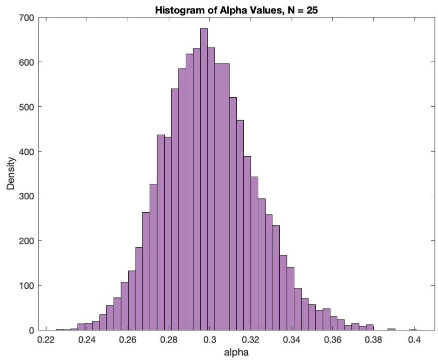
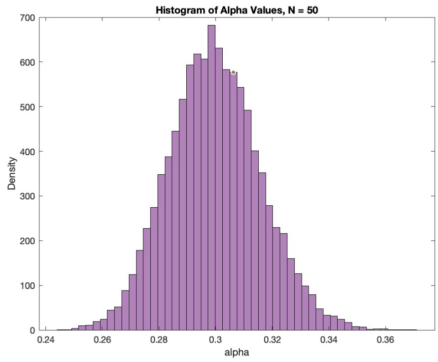


Figure 4: PE vs VME for N = 200

Under the assumption that the CDS prices are prone to have a random error which is normally distributed, we would like to see if our estimator is asymptotically normally distributed. To observe this through these simulation exercises, we plot the values of the simulated parameters as histograms for each sample sizes in Figures 5 to 10.

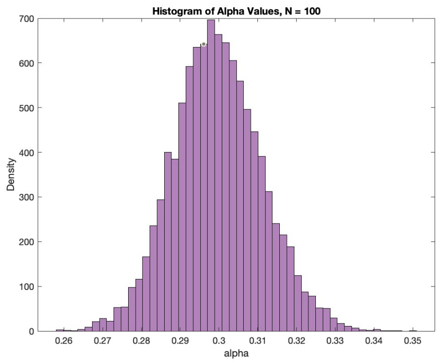


(a) $N = 25$

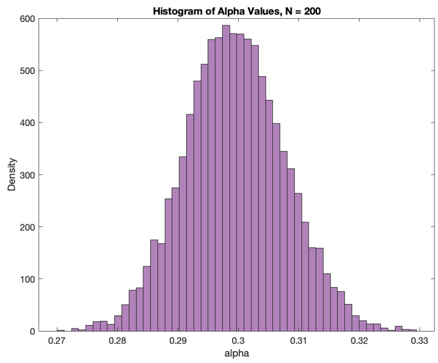


(b) $N = 50$

Figure 5: Histogram of Alpha Values

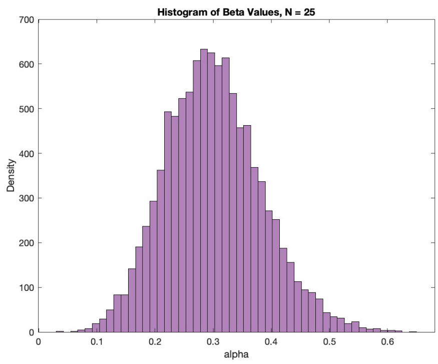


(a) $N = 100$

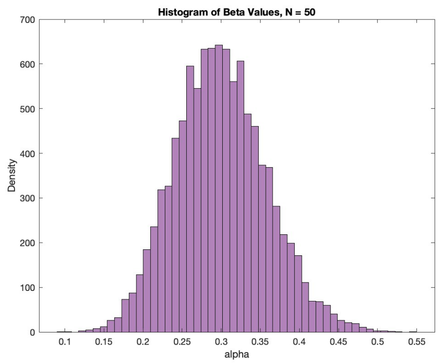


(b) $N = 200$

Figure 6: Histogram of Alpha Values

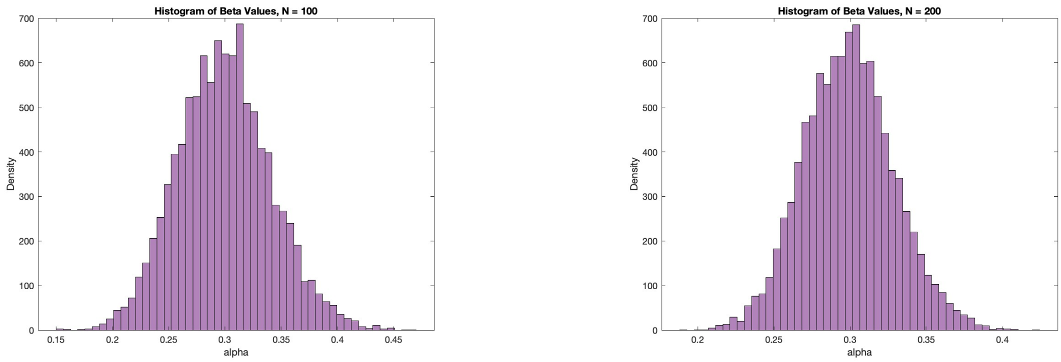


(a) $N = 25$



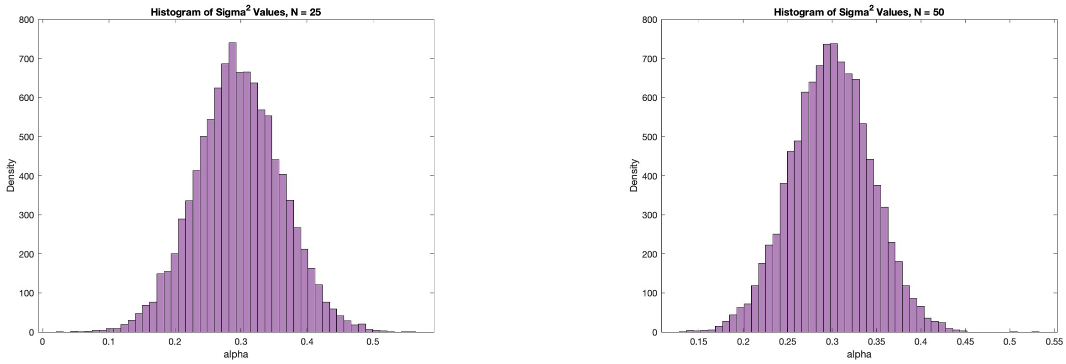
(b) $N = 50$

Figure 7: Histogram of Beta Values



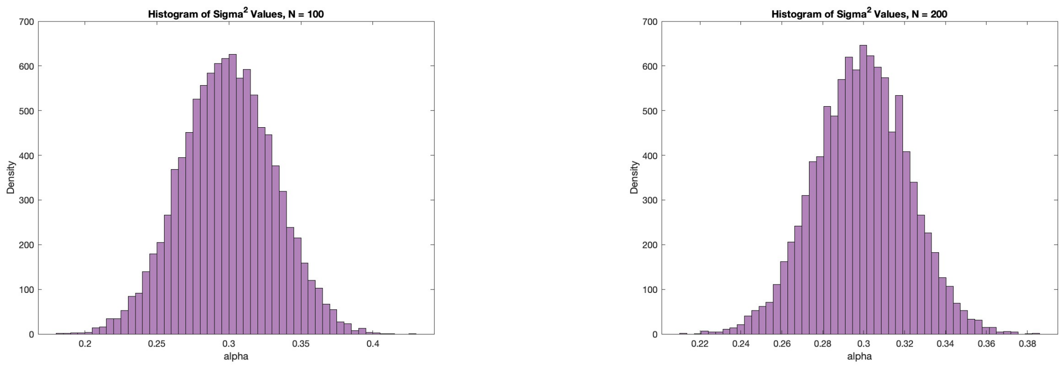
(a) $N = 100$ (b) $N = 200$

Figure 8: Histogram of Beta Values



(a) $N = 25$ (b) $N = 50$

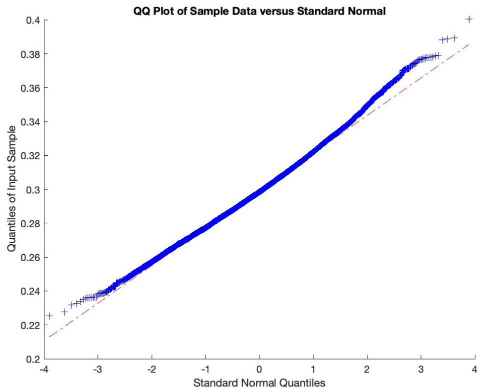
Figure 9: Histogram of Sigma^2 Values



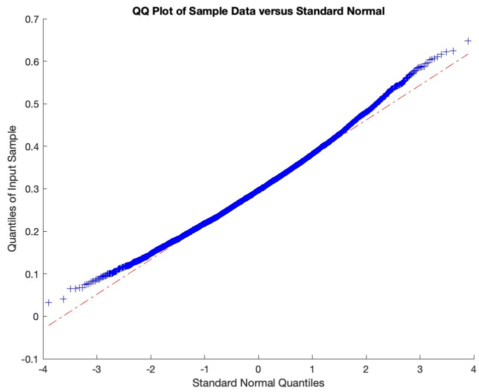
(a) $N = 100$ (b) $N = 200$

Figure 10: Histogram of Sigma^2 Values

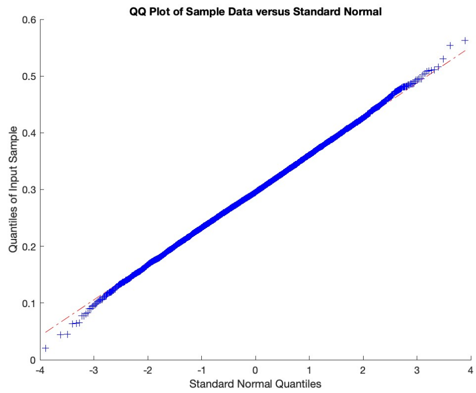
Additionally, we illustrated the data with quantile-quantile (Q-Q) plots. This type of plot is used to compare the two distributions. In this case, one of the distribution is the one belongs to the estimated parameters and the other one is the normal distribution. Figures 11 and 12 illustrates Q-Q plots where sample sizes are equal to 25 and 200 respectively. In these graphs, the quantiles of the standard normal distribution are given in the x-axis and the estimated values of the parameters are given in the y-axis. The red line reveals the correspondence between the parameter values and the standard normal quantiles. The normality of the data is determined by the alignment of the sample points with the red line. Figure 13, illustrates a Q-Q plot where the sample data is normally distributed and randomly generated.



(a) Alpha Values

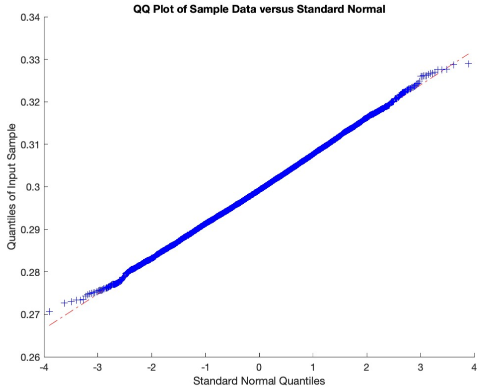


(b) Beta Values

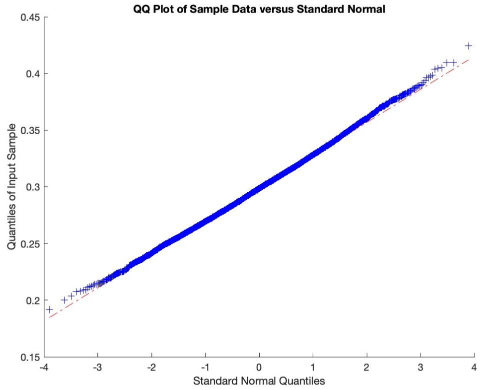


(c) Sigma^2 Values

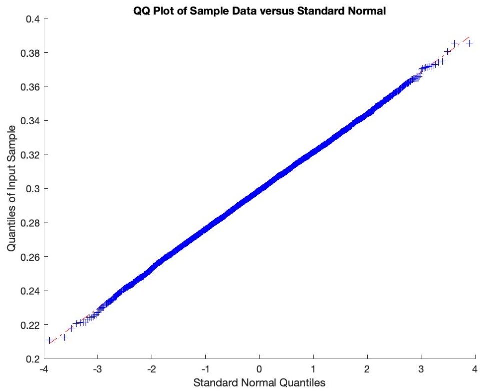
Figure 11: Q-Q Plots of Estimated Parameters, N = 25



(a) Alpha Values



(b) Beta Values



(c) Sigma^2 Values

Figure 12: Q-Q Plots of Estimated Parameters, N = 200

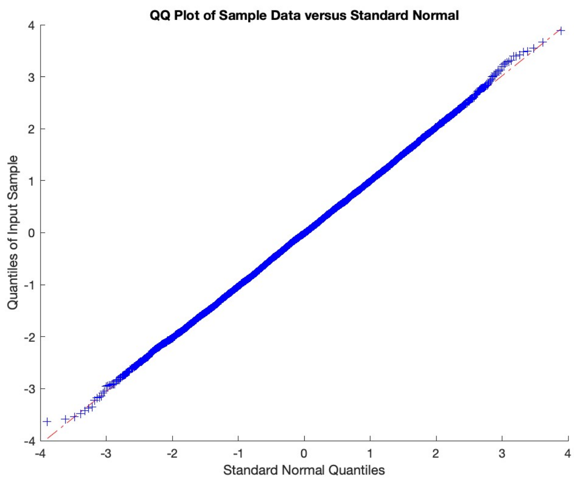


Figure 13: Q-Q Plot of Randomly Generated Normal Data

	N = 25	N = 50	N = 100	N = 200
Alpha	0.295	0.177	0.128	0.075
Beta	0.322	0.232	0.174	0.129
Sigma^2	0.014	-0.002	-0.007	-0.037

Table 5: Skewness Values

	N = 25	N = 50	N = 100	N = 200
Alpha	3.231	3.104	3.147	2.995
Beta	3.183	3.060	3.078	3.053
Sigma^2	3.129	3.068	2.938	3.066

Table 6: Kurtosis Values

3.1 Normality Test

In the report with 1K draws:

Although, each of these figures above shows a tendency for the samples to be normally distributed, we conducted the Shapiro-Wilk Test to assess the normality of the estimated parameters. Table 7 shows the p-values for the each series per the parameter and per the sample size. The values above 0.05 cannot reject the null hypothesis that the evaluated sample is normally distributed. Considering the values at Table 7, the sample is normally distributed as sample size increases. That is, the simulation exercises point out that the estimator is asymptotically normally distributed.

	N = 50	N = 100	N = 200
Alpha	0.000543	0.144119	0.850168
Beta	0.001135	0.045318	0.499443
Sigma^2	0.312960	0.694787	0.705204

Table 7: P-values from the Normality Test

I took this part out since it was infeasible to conduct a Shapiro-Wilk test for a sample larger than 5K. It is said that the test is so sensitive and unreliable for the large samples.

x

4 Appendix

(from Ang and Longstaff)

The expressions for $A(\lambda, t)$, $B(\xi, t)$, $C(\xi, t)$, and $F(\lambda, t)$ in equation (8) are given by

$$\begin{aligned} A(\lambda,t) &= A_1(t)\exp(A_2(t)\lambda), \\ B(\xi,t) &= B_1(t)\exp(B_2(t)\xi) \\ C(\xi,t) &= (C_1(t)+C_2(t)\xi)\exp(B_2(t)\xi) \\ F(\lambda,t) &= (F_1(t)+F_2(t)\lambda)\exp(A_2(t)\lambda), \end{aligned} \tag{3}$$

where

$$\begin{aligned} A_1(t) &= \exp\left(\frac{\alpha(\beta+\psi)t}{\sigma^2}\right)\left(\frac{1-\nu}{1-\nu\ e^{\psi t}}\right)^{2\alpha/\sigma^2}, \\ A_2(t) &= \frac{\beta-\psi}{\sigma^2}+\frac{2\psi}{\sigma^2(1-\nu\ e^{\psi t})} \\ B_1(t) &= \exp\left(\frac{a(b+\phi)t}{c^2}\right)\left(\frac{1-\theta}{1-\theta\ e^{\phi t}}\right)^{2a/c^2}, \\ B_2(t) &= \frac{b-\phi}{c^2}+\frac{2\phi}{c^2(1-\theta\ e^{\phi t})}, \\ C_1(t) &= \frac{a}{\phi}\left(e^{\phi t}-1\right)\exp\left(\frac{a(b+\phi)t}{c^2}\right)\left(\frac{1-\theta}{1-\theta\ e^{\phi t}}\right)^{2a/c^2+1}, \\ C_2(t) &= \exp\left(\frac{a(b+\phi)t}{c^2}+\phi t\right)\left(\frac{1-\theta}{1-\theta\ e^{\phi t}}\right)^{2a/c^2+2}, \\ F_1(t) &= \frac{\alpha}{\psi}\left(e^{\psi t}-1\right)\exp\left(\frac{\alpha(\beta+\psi)t}{\sigma^2}\right)\left(\frac{1-\nu}{1-\nu\ e^{\psi t}}\right)^{2\alpha/\sigma^2+1}, \\ F_2(t) &= \exp\left(\frac{\alpha(\beta+\psi)t}{\sigma^2}+\psi t\right)\left(\frac{1-\nu}{1-\nu\ e^{\psi t}}\right)^{2\alpha/\sigma^2+2}, \end{aligned} \tag{4}$$

and finally

$$\begin{aligned} \psi &= \sqrt{\beta^2+2\gamma\sigma^2}, \\ \nu &= (\beta+\psi)/(\beta-\psi), \\ \phi &= \sqrt{b^2+2c^2}, \\ \theta &= (b+\phi)/(b-\phi). \end{aligned} \tag{5}$$

5 Calculations

5.1 Section 1

$$d\xi_t = (a - b\xi_t)dt + c\sqrt{\xi_t}dZ_{\xi t}. \tag{6}$$

Defining $\xi = 0$ indicates that the mean ($\frac{a}{b}$ of the process is equal to zero. This means $a = 0$. From equation sets 3 and 4 in Appendix, $A(\lambda, t)$ and $F(\lambda, t)$ remain the same. However, $C(\xi, t) = (C_1(t) + C_2(t)\xi)\exp(B_2(t)\xi)$ becomes zero as

$$\begin{aligned}
C_1(t) &= \frac{a}{\phi}\left(e^{\phi t} - 1\right)\exp\left(\frac{a(b+\phi)t}{c^2}\right)\left(\frac{1-\theta}{1-\theta\ e^{\phi t}}\right)^{2a/c^2+1} = 0, && (7)\\
a &= 0, && (8)\\
\xi &= 0,
\end{aligned}$$

where Equation 8 implies Equation 7.

Moreover, $B(\xi, t) = B_1(t)\exp(B_2(t)\xi)$ becomes 1 as

$$\begin{aligned}
B_1(t) &= \exp\left(\frac{a(b+\phi)t}{c^2}\right)\left(\frac{1-\theta}{1-\theta\ e^{\phi t}}\right)^{2a/c^2} = 1, && (9)\\
a &= 0, && (10)\\
\xi &= 0,
\end{aligned}$$

where Equation 10 implies Equation 9. Therefore

$$\hat{s}_T^i = \frac{w\int_0^T D(t)(A(\lambda,t)C(\xi_i,t)) + \gamma_i B(\xi_i,t)F(\lambda,t))dt}{\int_0^T D(t)A(\lambda,t)B(\xi_i,t)dt} \overset{\xi=0}{=} \frac{w\int_0^T D(t)\gamma_i F(\lambda,t)dt}{\int_0^T D(t)A(\lambda,t)dt},$$

and

$$\hat{s}_T^i = \frac{wE\left[\int_0^T D(t)(\gamma_i\lambda_t + \xi_{it})\exp(-\int_0^t \gamma_i\lambda_t + \xi_{is}ds)dt\right]}{E\left[\int_0^T D(t)\exp(-\int_0^t \gamma_i\lambda_t + \xi_{is}ds)dt\right]} \overset{\xi=0}{=} \frac{wE[\int_0^T D(t)\gamma_i\lambda_t e^{-\int_0^t \gamma_i\lambda_s ds}dt]}{E[\int_0^T D(t)e^{-\int_0^t \gamma_i\lambda_s ds}dt]}.$$