Journal of Productivity Analysis (2019) 52:29–35
https://doi.org/10.1007/s11123-019-00554-9

Evaluating the CDF of the distribution of the stochastic frontier composed error

Christine Amsler[1] · Peter Schmidt[1] · Wen-Jen Tsay[2]

Published online: 8 August 2019

Abstract

In the stochastic frontier model, the composed error is the sum (or difference) of a normal and a half normal random variable. Often the composed error is linked to other errors using a copula, and evaluation of the copula requires evaluation of the cdf of the composed error. There is no analytical expression for this cdf, though there are several approximations. We propose a computationally efficient simulation based method of evaluation and use it to evaluate the accuracy of these approximations. We also derive the exact cdf of the composed error for the special case that the stochastic frontier relative variance parameter λ equals one, and we use this expression to investigate the accuracy of our evaluations and the existing approximations.

Keywords Stochastic frontier · Composed error · Skew normal distribution

JEL classification C13

1 Introduction

This paper deals with the evaluation of the cumulative distribution function (cdf) of the stochastic frontier model's composed error. Since the normal/half-normal composed error has a skew-normal distribution, we can also say that the paper deals with the evaluation of the cdf of the skew-normal distribution.

Evaluation of the skew-normal cdf may be important in a number of contexts, including at least the following two. (1) In many multi-equation models, or in a panel data setting, the composed error in a stochastic frontier production or cost function is linked to other errors using a copula. Some examples of this approach include Amsler et al. (2014, 2016, 2017), Carta and Steel (2012), Das (2015), Genius et al. (2012), Huang et al. (2017), Huang et al. (2018), Lai and Huang (2013), Shi and Zhang (2011), Sriboonchitta et al. (2017) and Tran and Tsionas (2015). The evaluation of the likelihood of such a model involves the calculation of the copula density, which in turn requires the calculation of the cdf of each of the marginal distributions of the various errors in the model. Therefore, if one of the errors is a stochastic frontier composed error, evaluating the likelihood requires calculation of the skew-normal cdf. (2) We may want to test the distributional assumptions of the stochastic frontier model by testing whether the composed error has a skew-normal distribution, as suggested by Wang et al. (2011). Their preferred test is a bootstrapped version of the Kolmogorov-Smirnov test, and its calculation requires the calculation of the skew-normal cdf.

There is no known closed form solution for the skew-normal cdf. It can be calculated by simulation, and there are some available approximations, such as Ashour and Abdul-Hameed (2010) and Tsay et al. (2013). In this paper we provide a simulation-based method which is computationally efficient relative to the simple empirical cdf. We use it to evaluate the accuracy of the existing approximations.

The paper has five main contributions. First, it proposes the new simulation-based method of evaluating the skew-normal cdf. Second, it uses this method to evaluate the accuracy of existing approximations, notably that of Tsay et al. (2013). Third, we create a tabulation of the cdf, part of which is given in this paper and most of which is in a supplemental file, which can be used in estimation.

Supplementary information The online version of this article (https://doi.org/10.1007/s11123-019-00554-9) contains supplementary material, which is available to authorized users.

✉ Peter Schmidt
schmidtp@msu.edu

[1] Michigan State University, East Lansing, MI, USA

[2] Academia Sinica, Taipei, Taiwan

Interpolation in such a file is faster than evaluating an approximation, which in turn is faster than simulation-based or quadrature methods.

A fourth objective is to extend the range of values of the composed error for which we can calculate a cdf value that is not zero and is not one. This is important because some commonly used copulas are undefined if the marginal cdf has a value of zero or one. For example, if the composed error ε has cdf F, and if Φ is the standard normal cdf, the Gaussian (normal) copula contains the term $\Phi^{-1}(F(\varepsilon))$, which equals minus infinity when $F = 0$ and equals plus infinity when $F = 1$. This could cause the calculation of the copula density to break down. We are able to calculate non-zero values of $F(\varepsilon)$ and $1 - F(\varepsilon)$ for a much wider range of ε than in previous papers.

Finally, we derive a closed-form solution for the skew-normal cdf in the special case that the relative variance parameter in the stochastic frontier model (λ) equals one. Therefore, in that special case, we have a much-needed exact standard for assessing the accuracy of both simulation-based and approximate methods of evaluating the cdf.

2 Theory

We start with some notation and basics. The composed error is $\varepsilon = v + u$, where $v \sim N\left(0, \sigma_v^2\right)$, $u \sim N^+\left(0, \sigma_u^2\right)$, and v and u are independent. Standard notation is $\sigma^2 = \sigma_u^2 + \sigma_v^2$ and $\lambda = \sigma_u/\sigma_v$. Then ε has the skew-normal density $sn_{\lambda,\sigma}(\varepsilon) = \left(\frac{2}{\sigma}\right)\varphi\left(\frac{\varepsilon}{\sigma}\right)\Phi\left(\frac{\lambda\varepsilon}{\sigma}\right)$, where φ is the standard normal pdf and Φ is the standard normal cdf. We want to calculate and tabulate the skew-normal cdf $P_{\lambda,\sigma}(Q) = P(\varepsilon \leq Q)$, for as large a range of values of Q as we can (i.e. where the calculations are numerically possible).

The above discussion is for the case of $v + u$, which would be natural in a cost frontier, and follows the discussion in Tsay et al. (2013). In the case of a production frontier, as in the original papers of Aigner et al. (1977) and Meeusen and van den Broeck (1977), we would want to consider $\varepsilon_* = v - u$ instead of $\varepsilon = v + u$. But this does not require a separate tabulation, because the distribution of ε_* is the same as the distribution of $(-\varepsilon)$. Explicitly, if $P^*_{\lambda,\sigma}(Q) = P(\varepsilon_* \leq Q)$, then $P^*_{(\lambda,\sigma)}(Q) = 1 - P_{(\lambda,\sigma)}(-Q)$ and we can get values of P^* from a tabulation of P.

It would appear that we would require a three—dimensional tabulation, giving probabilities over values of the two parameters λ and σ, plus values of Q. But in fact we only need a two-dimensional tabulation, over values of λ and Q. Specifically, we can pick $\sigma = 1$ and just tabulate $P_{\lambda,1}(Q)$. For other values of σ, we use the fact that $P_{\lambda,\sigma}(Q) = P_{\lambda,1}(Q/\sigma)$. To see why this equality holds, start with $P_{\lambda,\sigma}(Q) = \int_0^Q \left(\frac{2}{\sigma}\right)\varphi\left(\frac{\varepsilon}{\sigma}\right)\Phi\left(\frac{\lambda\varepsilon}{\sigma}\right)d\varepsilon$ and make the substitutions $z = \frac{\varepsilon}{\sigma}$ and $d\varepsilon = \sigma dz$, and note that the upper limit of integration $\varepsilon = Q$ becomes $z = \frac{Q}{\sigma}$. Thus we have $\int_0^Q \left(\frac{2}{\sigma}\right)\varphi\left(\frac{\varepsilon}{\sigma}\right)\Phi\left(\frac{\lambda\varepsilon}{\sigma}\right)d\varepsilon = \int_0^{Q/\sigma} 2\varphi(z)\Phi(\lambda z)dz = P_{\lambda,1}(Q/\sigma)$.

There is no closed-form expression for the cdf of the skew-normal distribution. The required integral is widely regarded as intractable. (See the Appendix for some explanation of this point.) The cdf can be calculated (or estimated) by numerical integration, or by simulation. Numerical integration (quadrature) is of questionable accuracy, especially in the extreme tails. We calculate cdf values that are sometimes extremely small, like 4.08e−115 for $\lambda = 1$, $Q = -16$, and we cannot expect quadrature to yield an accurate evaluation of a probability that small, whereas as we will see this cdf value is accurately evaluated by our simulation algorithm.

The most obvious path to evaluation by simulation is the empirical cdf, that is, $F(Q)$ is estimated by the fraction of draws from the distribution of ε that are less than or equal to Q. This works reasonably well in the middle of the distribution, but in the tails it requires an unreasonably large number of draws. For example, in Tsay et al. (2013), Table 1, p. 262, for $\lambda = 1.5$, $\sigma^2 = 1.444$, they report $F(-3) =$ 0.0000006, or 6/10,000,000. That is, they used 10,000,000 replications and got six draws that were less than or equal to −3. In the calculations we report below, we have probability values in the tails that are very small, e.g. 3.87e−31 for $\lambda = 1$ and $Q = -8$, and so we would need a number of replications on the order of 10^{31} or larger to hope to estimate this probability. That is obviously not feasible.

Similarly, Wang et al. (2011) calculated and reported the quantiles of $\varepsilon_* = v - u$ based on a sample of 10,000,000 draws, for various values of λ. In their supplemental tables (available on request from the authors), they consider a very large set of values of λ, and they give the empirical quantiles 0.01, 0.02, …, 0.99. (They also give the quantiles zero and one, but these are just the minimum and maximum values in the sample, whereas the population distribution of ε_* does not have a finite minimum or maximum value.) The information in the quantile values is in principle the same as in the cdf values, and they could have considered quantiles smaller than 1% or bigger than 99%, but for exactly the same reasons as given in the previous paragraph they could not have calculated meaningful quantiles very far into the tail without using far more than 10,000,000 draws.

As an alternative, we will propose a method that is very similar to a method often used in the literature on simulated MLE. See, e.g., Greene (2010). A probability is the expectation of an indicator function, and by the law of iterated expectations $P(\varepsilon \leq Q) = P(v + u \leq Q) = P(v \leq Q - u) = E_u P(v \leq Q - u|u) = E_u \Phi[(Q - u)/\sigma_v]$. (The last equality follows from the independence of v and u.) We calculate this by averaging $\Phi[(Q - u)/\sigma_v]$ over a large number of draws from the distribution of u.

Table 1 Values of our evaluation of $F(Q)$, for $Q \leq 0$, in bold, R = 10,000,000

Q	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.50$	$\lambda = 1$	$\lambda = 2$	$\lambda = 4$	$\lambda = 8$
−16	**6.38875e−58** 6.38875e−58	**3.79523e−62** 6.41224e−62	**6.34097e−73** 3.29645e−71	**4.0776e−115** 1.3037e−105	**1.3908e−282** 1.6365e−237	********** **********	********** **********
−12	**1.77648e−33** 1.77648e−33	**4.48929e−36** 5.44250e−36	**2.79129e−42** 1.70533e−41	**3.15338e−66** 1.99473e−61	**9.8920e−161** 8.7245e−137	********** **********	********** **********
−10	**7.61985e−24** 7.61985e−24	**8.84249e−26** 9.70223e−26	**3.47421e−30** 1.01269e−29	**5.80203e−47** 6.76757e−44	**8.4227e−113** 6.74193e−97	********** 2.9412e−301	********** **********
−8	**6.22096e−16** 6.22096e−16	**2.63744e−17** 2.72141e−17	**3.12590e−20** 5.36118e−20	**3.86748e−31** 2.11512e−29	**1.60128e−73** 5.48345e−64	**4.0125e−241** 4.9234e−196	********** **********
−6	**9.86588e−10** 9.86588e−10	**1.22228e−10** 1.22451e−10	**2.10293e−12** 2.57818e−12	**9.72760e−19** 6.20523e−18	**7.10964e−43** 4.78902e−38	**1.6414e−137** 1.5063e−113	********** **********
−5	**2.86652e−7** 2.86652e−7	**5.59415e−8** 5.57399e−8	**2.79935e−9** 3.09890e−9	**8.21208e−14** 2.48729e−13	**8.91478e−31** 1.10261e−27	**9.54130e−97** 7.93329e−81	********** 3.0096e−285
−4	**3.16712e−5** 3.16712e−5	**9.22945e−6** 9.18713e−6	**1.12624e−6** 1.16936e−6	**1.00250e−9** 1.77515e−9	**8.12310e−21** 4.68369e−19	**2.48724e−63** 8.82626e−54	**5.5376e−231** 6.9043e−186
−3	**1.34990e−3** 1.34990e−3	**5.57147e−4** 5.55622e−4	**1.39607e−4** 1.40329e−4	**1.82126e−6** 2.28651e−6	**5.59265e−13** 3.73034e−12	**3.04050e−37** 2.11481e−32	**6.2357e−132** 5.8131e−108
−2	**2.22750e−2** 2.22750e−2	**1.26118e−2** 1.26137e−2	**5.49841e−3** 5.47615e−3	**5.17324e−4** 5.44520e−4	**3.14165e−7** 5.74526e−7	**1.91857e−18** 1.13571e−16	**5.22690e−61** 1.88167e−51
−1	**0.158655** 0.158655	**0.111826** 0.112071	**7.24853e−2** 7.24305e−2	**2.51618e−2** 2.51433e−2	**1.71791e−3** 1.83418e−3	**8.17304e−7** 1.51799e−6	**4.48223e−18** 2.68045e−16
0	**0.500000** 0.500000	**0.421993** 0.422779	**0.352369** 0.352863	**0.249935** 0.250166	**0.147519** 0.147764	**7.79357e−2** 7.81201e−2	**3.95592e−2** 3.96636e−2

********** indicates that the calculation fails (the result is just reported as zero)

Values of Tsay et al. approximation, for $Q \leq 0$, not in bold

To be very explicit, our procedure is as follows. (1) Set $\sigma = 1$ and pick a value of λ. Calculate the implied values of σ_u and σ_v. With $\sigma^2 = 1$, these are $\sigma_v^2 = 1/\left(1+\lambda^2\right)$ and $\sigma_u^2 = \lambda^2/\left(1+\lambda^2\right)$. (2) Pick a value of Q. (3) Now, for replication $r = 1, \ldots, R$, where R is a very large number, take a draw from $N(0,1)$, take its absolute value, and multiply by σ_u to get u_r. This generates a draw from $N^+\left(0, \sigma_u^2\right)$ because the absolute value of a $N(0,1)$ random variable is distributed as $N^+(0,1)$, and multiplying by σ_u converts $N^+(0,1)$ into $N^+\left(0, \sigma_u^2\right)$. (4) Calculate $\Phi[(Q-u_r)/\sigma_v]$. (5) Average this over the R replications.

This is preferable to an evaluation of the empirical cdf because it avoids the randomness from drawing v, and because reliable methods exist for evaluating the normal cdf in the extreme tails, such as 20 standard deviations from zero.

Finally, although the skew-normal cdf is analytically intractable, we were able to derive an exact expression for the cdf for the special case of $\lambda = 1$ ($\sigma_u = \sigma_v$). This is given in the following result, which we prove in the Appendix.

3 Result

Suppose that $\lambda = 1$ ($\sigma_u = \sigma_v$). Then $P_{1,\sigma}(Q) = \Phi^2\left(\frac{Q}{\sqrt{2}\sigma_u}\right)$. When $\sigma^2 = 1$, this simplifies to $P_{1,1}(Q) = \Phi^2(Q)$.

This result is useful because, apart from the trivial case of $\lambda = 0$ (the normal distribution), it provides the only exact standard for assessing the accuracy of both simulation-based and approximate methods of evaluating the skew-normal cdf.

4 Some tabulations and comparisons

Table 1 gives values of the cdf $F(Q) = P(\varepsilon \leq Q)$ for non-positive values of Q, with $-16 \leq Q \leq 0$. Table 2 gives values of $1 - F(Q)$ for positive values of Q, with $1 \leq Q \leq 20$. The reason that we show values of $1 - F(Q)$ for positive values is that otherwise, for the larger values of Q, $F(Q)$ would round to one unless a very large number of decimal places were preserved, and if they were preserved the number of digits "9" would fill a whole line. For example, for $\lambda = 1$ and $Q = 12$, our value of $F(Q)$ is 1−3.64006e−39, and to display that in decimal form would require 38 digits "9" between the decimal place and 635994. Of course, it does not matter whether we report that $1 - F(Q) = 3.64006\text{e}{-39}$ or $F(Q) = 1{-}3.64006\text{e}{-39}$.

For each (Q,λ) "cell," the top number, in bold, is our evaluation of F(Q) or $1 - F(Q)$ and the number underneath it is the approximation of Tsay et al. (2013).

Table 2 Values of our evaluation of $1 - F(Q)$, for $Q > 0$, in bold, $R = 10{,}000{,}000$

Q	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.50$	$\lambda = 1$	$\lambda = 2$	$\lambda = 4$	$\lambda = 8$
1	**0.158655**	**0.205494**	**0.244832**	**0.292229**	**0.315720**	**0.317444**	**0.317435**
	0.158655	0.205239	0.244880	0.292167	0.315476	0.317309	0.317311
2	**2.27501e−2**	**3.23927e−2**	**4.00162e−2**	**4.50189e−2**	**4.55569e−2**	**4.55616e−2**	**4.55581e−2**
	2.27501e−2	3.28866e−2	4.00241e−2	4.49557e−2	4.54997e−2	4.55003e−2	4.55003e−2
3	**1.34990e−3**	**2.14317e−3**	**2.56193e−3**	**2.70246e−3**	**2.70764e−3**	**2.70815e−3**	**2.70581e−3**
	1.34990e−3	2.14417e−3	2.55947e−3	2.69751e−3	2.69980e−3	2.69980e−3	2.69980e−3
4	**3.16712e−5**	**5.41336e−05**	**6.22789e−5**	**6.33965e−5**	**6.32240e−5**	**6.32869e−5**	**6.34124e−5**
	3.16712e−5	5.41554e−5	6.21731e−5	6.33407e−5	6.33425e−5	6.33425e−5	6.33425e−5
5	**2.86652e−7**	**5.17639e−7**	**5.70969e−7**	**5.62731e−7**	**4.90247e−7**	**3.89328e−7**	**3.38328e−7**
	2.86652e−7	5.17563e−7	5.70204e−7	5.73303e−7	5.73303e−7	5.73303e−7	5.73303e−7
6	**9.86588e−10**	**1.85222e−9**	**1.96740e−9**	**1.71000e−9**	**4.58049e−10**	**2.80140e−12**	**6.68121e−23**
	9.86588e−10	1.85072e−9	1.97060e−9	1.97318e−9	1.97318e−9	1.97318e−9	1.97318e−9
8	**6.22096e−16**	**1.21869e−15**	**1.18968e−15**	**2.58074e−16**	**2.92076e−21**	**2.65741e−45**	**1.5975e−130**
	6.22096e−16	1.21698e−15	1.24414e−15	1.24419e−15	1.24419e−15	1.24419e−15	1.24419e−15
10	**7.61985e−24**	**1.51356e−23**	**1.16083e−23**	**5.94021e−26**	**1.25250e−40**	**1.6995e−101**	**********
	7.61985e−24	1.51427e−23	1.52397e−23	1.52397e−23	1.52397e−23	1.52397e−23	1.52397e−23
12	**1.77648e−33**	**3.51550e−33**	**1.39237e−33**	**3.64006e−39**	**1.56865e−68**	**3.8117e−190**	**********
	1.77648e−33	3.54752e−33	3.55296e−33	3.55296e−33	3.55296e−33	3.55296e−33	3.55296e−33
16	**6.38875e−58**	**1.15397e−57**	**1.54967e−59**	**1.78556e−75**	**3.0154e−153**	**********	**********
	6.38875e−58	1.27769e−57	1.27775e−57	1.27775e−57	1.27775e−57	1.27775e−57	1.27775e−57
20	**2.75362e−89**	**3.40590e−89**	**6.15395e−94**	**1.5595e−125**	**1.2335e−270**	**********	**********
	2.75362e−89	5.50724e−89	5.50725e−89	5.50725e−89	5.50725e−89	5.50725e−89	5.50725e−89

Values of Tsay et al. approximation of $1 – F(Q)$, for $Q > 0$, not in bold

The first thing to note is that we are able to calculate a value for $F(Q)$, both for our method and for the approximation of Tsay et al., for a much larger range of Q than has previously been done. The numerical issues involved will be discussed in the next Section. For now, we simply note that Tsay et al., Table 1, reported results for Q from −3.0 to 3.0, and their algorithm would not calculate probabilities smaller than about 1.0e−16. Ashour and Abdul-Hameed (2010) tabulated results for Q in the range from zero to four. Wang et al. (2011) tabulated quantiles, not cdf values, but the smallest quantile they considered was 0.01 and the largest was 0.99.

To ask how close our cdf values are to the Tsay et al. approximation, we have to ask what we mean by close. For example, for $Q = -1$, $\lambda = 1$, the cdf values of 0.02516 and 0.02514 are close in both absolute and relative terms, whereas for $Q = -12$, $\lambda = 1$, the values of 3.153e−66 and 1.994e−61 are close in absolute terms but not in relative terms. Both Tsay et al. and Ashour and Abdul-Hameed comment on closeness in absolute terms, but it is not clear why this is relevant. Indeed, the relevant notion of closeness logically depends on the copula. For example, if we are using the normal copula, what is relevant is the value of $\Phi^{-1}(F(\varepsilon))$. For $Q = -12$, $\lambda = 1$, the value of $\Phi^{-1}(F(\varepsilon))$ is −16.923 for our calculation, −16.259 for the Tsay et al. approximation, and minus infinity for the Ashour and Abdul-Hameed approximation, which equals zero for all $Q < -3$. As another example, for $Q = 16$, $\lambda = 1$, the value of $\Phi^{-1}(F(\varepsilon))$ is 18.134 for our calculation and 15.712 for the Tsay et al. approximation. Of course, these numbers would be different for a different copula, and ultimately the bias in estimation caused by a miscalculated cdf will depend both on the copula and the model that uses the copula.

Having said that, the values of our calculation of $F(Q)$ and the Tsay et al. approximation are quite close in both absolute and relative terms for non-extreme values of Q, say $-4 \le Q \le 3$. For more extreme values of Q, they are close in absolute but not always in relative terms.

The Ashour and Abdul-Hameed approximation, which sets $F(Q) = 0$ for $Q < -3$, is in a sense infinitely bad in relative terms for Q in that range, and we will drop it from further consideration, even though it appears to be accurate in the non-extreme part of the range of Q.

5 Numerical issues and accuracy checks

5.1 Numerical issues

Our calculations were done in MATLAB.

For the non-positive values of Q, we calculated the normal cdf (so that we can calculate $\Phi[(Q-u_r)/\sigma v]$) using the MATLAB command **normcdf**. This gave results that matched those in Marsaglia (2004) for $-16.6 \le z \le -0.1$

where z is generic notation for the normal cdf argument. The MATLAB results also matched the results from the online Casio Keisan normal cdf calculator.

For positive values, some care needed to be taken to keep the cdf from rounding to one. For example, for $z = 12$, normcdf returns "1". However, the MATLAB command **normcdf, 'upper'** returns the upper tail probability $1 - \Phi(12) = 1.77648e{-}33$. The key is to average the values of $1 - \Phi[(Q - u_r)/\sigma_v]$ and then subtract this average from one so that the small deviations from one are preserved. If you subtract the individual deviations from one separately for each replication and then average, you will just get one.

A check of the accuracy of the routine normcdf, 'upper' is that, for positive z, the value of $1 - \Phi(z)$ equaled $\Phi(-z)$, which it did, even for extreme values of z. For example, normcdf evaluated at $z = -20$ gives 2.753624e−89 and normcdf,'upper' evaluated at $z = 20$ gives 1–2.753624e−89.

Similar considerations apply to the calculation of the Tsay et al. approximation. For $Q < 0$, the approximate cdf as given in equation (12) of their text and in the last equation of their Appendix is of the form $2GH$, where G and H are our shorthand for the terms in the last equation in the Appendix. The term G is easily calculated, but H involves the "error function" $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z \exp(-t^2)dt$. Specifically, $H = 1 - \mathrm{erf}(z)$ where z is a linear function of Q, with a negative coefficient on Q. For $Q < 0$ but large in magnitude, z will be a large positive number and erf(z) will round to one and 1−erf(z) will round to zero. The solution is to use not the MATLAB command **erf**, but rather to use the command **erfc** to calculate $\mathrm{erfc}(z) = 1 - \mathrm{erf}(z)$. For example, when $\lambda = 1$ and $Q = -10$, $z = 9.7847$, and we could calculate erf $(9.7847) = 1$, $1 - \mathrm{erf}(9.7847) = 0$, $H = 0$ and approximate cdf = 0, which is not an accurate or useful result. We need to calculate $H = \mathrm{erfc}(9.7847) = 1.5117e - 43$, which yields an approximate cdf of 6.7566e−43.

Conversely, for large positive Q, we have an extra term, which we will call J, as given in the last term of the second to last equation of the Appendix. The approximate cdf is equal to $2(GH^* + J)$, where H^* is like H except for a sign change in one term. G is the same as before and H^* is almost zero and it won't matter numerically if it rounds to zero or not. The value of the approximate cdf is determined by the term $2J$ where J is 0.5 minus a very small number, and it is essential not to let J round to 0.5. So for example for $Q = 20$, $2J = \mathrm{erf}(14.1421)$ and if you calculate erf (14.1421) you will get one "exactly." Instead you need to calculate erf(14.421) as 1 − erfc(14.1421) = 1–5.51281e −89 and this will lead to the approximate cdf equal to 1− 5.51282e−89, a meaningful result.

Finally, we consider the evaluation of the exact cdf $P_{1,1}(Q) = \Phi^2(Q)$ for the case of $\lambda = 1$. For negative Q, there is no numerical issue. We just calculate calculate $\Phi(Q)$ and take the square. For large positive Q, however, we need to take care to keep $\Phi(Q)$ from rounding to one. For example, for $Q = 8$, we use the command normcdf, 'upper' to obtain $1 - \Phi(8) = 6.22096e - 16$, which we translate into $\Phi(8) = 1 - 6.22096e - 16$ and $\Phi^2(8) = 1 - 2 \times 6.2209e{-}16 + 6.22096^2e - 32 = 1 - 1.24419e - 15$.

5.2 Accuracy of the calculations

The Tsay et al. approximation and the exact result for $\lambda = 1$ are closed-form expressions, apart from the need to evaluate the normal cdf and error function. So there is little question of numerical accuracy for these results. However, our simulated $F(Q)$ is not a closed form expression and it could be inaccurate for any number of reasons, most notably the inherent randomness of the simulation and the quality of the random number generator.

The first thing we investigate is how sensitive the results are to the choice of R, the number of replications in the simulation. Table 3 gives results for some values of Q, for R ranging from 1,000,000 to 100,000,000, for the case of $\lambda = 1$, so that we have the exact result to compare to. For $Q \leq 4$, the results do not depend very much on the number of replications, and $R = 1{,}000{,}000$ is sufficient to give reasonably accurate results. Things begin to be less clear for $Q = 6$, and for $Q \geq 8$ the results depend more strongly on R, and they do not converge unambiguously to the exact result even for $R = 100{,}000{,}000$.

Table 4 gives the comparison between the simulated $F(Q)$ and the exact $F(Q)$, for $\lambda = 1$ and for more different

Table 3 Simulated $F(Q)$, $\lambda = 1$, for various values of R

Q	$R = 1$	$R = 2$	$R = 5$	$R = 10$	$R = 20$	$R = 50$	$R = 100$	Exact
–8	3.85678e−31	3.85887e−31	3.86875e−31	3.86748e−31	3.86999e−31	3.87056e−31	3.86972e−31	3.86748e−31
0	0.249868	0.249856	0.249919	0.249935	0.249973	0.250007	0.249992	0.25
2	1–4.51024e−2	1–4.50771e−2	1–4.50544e−2	1–4.50189e−2	1–4.50095e−2	1–4.49759e−2	1–4.49810e−2	1–4.49826e−2
4	1–6.37772e−5	1–6.37196e−5	1–6.34907e−5	1–6.33965e−5	1–6.35013e−5	1–6.32251e−5	1–6.32461e−5	1–6.33418e−5
6	1–1.78380e−9	1–1.69967e−9	1–1.72217e−9	1–1.71000e−9	1–1.82219e−9	1–1.80531e−9	1–1.83839e−9	1–1.97318e−9
8	1–2.6169e−16	1–2.19874e−16	1–2.72587e−16	1–2.58074e−16	1–4.02962e−16	1–4.00577e−16	1–4.49283e−16	1–1.24419e−15

R in millions

Table 4 Exact versus simulated $F(Q)$, $\lambda = 1$

Q	Φ(Q)	Exact $F(Q)$ [$\Phi^2(Q)$]	Simulated $F(Q)$
−16	6.38875e−58	4.08161e−115	4.07757e−115
−12	1.77648e−33	3.15588e−66	3.15338e−66
−8	6.22096e−16	3.87003e−31	3.86748e−31
−6	9.86587e−10	9.73298e−19	9.72760e−19
−4	3.16712e−5	1.00306e−9	1.00250e−9
−2	2.27501e−2	5.17567e−4	5.17324e−4
−1	0.158655	2.51714e−2	2.51618e−2
0	0.5	0.25	0.249935
1	0.841345	0.707861	0.707771
2	1–2.27501e−2	1–4.49826e−2	1–4.50189e−2
4	1–3.16712e−5	1–6.33418e−5	1–6.33965e−5
6	1–9.86588e−10	1–1.97318e−9	1–1.71000e−9
8	1–6.22096e−16	1–1.24419e−15	1–2.58074e−16
12	1–1.77648e−33	1–3.55298e−33	1–3.64006e−39
16	1–6.38875e−58	1–1.27775e−57	1–1.78556e−75

For simulated $F(Q)$ $R = 10{,}000{,}000$

values of Q. The results are the same as described in the previous paragraph. The simulated cdf is accurate for $Q \le 4$ and becomes less accurate thereafter.

Of course, this may just reflect too strict a meaning of the word accurate. The simulated cdf is quite close to the exact cdf in the absolute sense, for all of the values of Q that we consider. The inaccuracy that we have identified is in the relative sense.

There are multiple possible explanations for numerical inaccuracy in a simulation, but in this case it is easy to suspect the random number generator. We used the MATLAB command **rng (s,'twister')** where "twister" denotes the Marsenne twister algorithm and s is the seed. We picked $s = 1$. A sample of 10,000,000 pseudo-random normal deviates from this generator passed standard tests on the first four moments. However, a more focused test of the random number generator is to check whether other random number generators give different results, and, if so, whether they more closely match the exact results for $\lambda = 1$.

We considered two addition pseudo-random number generators. One is the MATLAB command **rng ('default')**, which is the same as rng (0,'twister'). The other creates pseudo-random deviates z such that the values of $\Phi(z)$ uniformly fill the space [0,1]. Explicitly, for $j = 1, \ldots, R$, we choose $z_j = \Phi^{-1}((j - 1/2)/R)$. We could call this **uniform spacing**. This is somewhat similar to a Van der Corput sequence, which is a one−dimensional Halton sequence, but it is simpler because we have no need to have uncorrelated draws.

For values of Q where our results are numerically stable ($Q \le 4$) the choice of random number generator makes very little difference. For example, for $Q = 2$ and $\lambda = 1$, the three random number generators listed above yield $1-F(Q)$ as 0.0450189, 0.0449673 and 0.0449827, respectively. These are all quite close to each other and to the exact value of 0.0449826. However, for larger values of Q the random number generator matters more. For example, for $Q = 8$ and $\lambda = 1$, we obtain 2.58074e−16, 3.69353e−16 and 4.86878e −16. So the choice of random number generators matters for the larger values of Q. However, none of these numbers is particularly close (in relative terms) to the exact value of 1.24419e−15. So the problem could be random number generation, but it is not due to the specific random number generator we used, and the Marsenne twister is considered to be the state−of-the−art random number generation algorithm. There is essentially an infinity of possible random number generators, and it is just not clear how likely it is that we could find one that would solve our inaccuracy problem, if in fact the problem does lie in random number generation.

6 Tabulations

We have created a set of supplemental tables, available on request, that give our calculation of $F(Q)$ as a function of Q and λ. They cover the range $-16 \le Q \le 10$ and $0 \le \lambda \le 8$ and were calculated for $R = 10{,}000{,}000$. We trust these calculations to be accurate for $Q \le 4$. For Q between 4 and 10, they are less accurate, but we have included these numbers because it is not clear what a better alternative would be. An evaluation that is exactly equal to one is not a good alternative. For the case of $\lambda = 1$, for which we have an exact result, the Tsay et al. approximation is quite accurate for large values of Q. We conjecture that this may be so for other values of λ as well, but we have no evidence to support this conjecture.

7 Concluding remarks

In our view, the main contribution of the paper is to have extended the range of the argument over which we can get numerically stable and believable cdf values. This range is not as wide as we would like, but it is considerably wider than in previous papers.

The other substantial contribution of the paper is the derivation of a closed-form expression for the exact cdf, for the special case of $\lambda = 1$. This allows us to check the accuracy of the cdf values that we have calculated and tabulated, at least for one special case.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

8 Appendix

We wish to evaluate

$$P \equiv P_{1,1}(Q) = 2\int_0^\infty \Phi\left(\frac{Q-u}{\sigma_v}\right)\frac{1}{\sigma_u}\varphi\left(\frac{u}{\sigma_u}\right)du.$$

Suppose that $\sigma^2 = \sigma_u^2 + \sigma_v^2 = 1$. Now make the substitutions $z = \frac{u}{\sigma_u}$ and $= \sigma_u dz$, and define $a = \frac{Q}{\sigma_v}$ and $\lambda = \frac{\sigma_u}{\sigma_v}$. This yields

$$P = 2\int_0^\infty \Phi(a - \lambda z)\varphi(z)dz,$$

According to Owen (1980), equation 10,010.6, p. 403,

$$\int_0^\infty \Phi(a + bz)\varphi(z)dz = \frac{1}{2}\Phi\left(\frac{a}{\sqrt{1+b^2}}\right) + T\left(\frac{a}{\sqrt{1+b^2}}, b\right),$$

where (Owen, p. 391)

$$T(h,b) = \int_0^b \frac{\varphi(h)\varphi(hx)}{1+x^2}dx.$$

In our case $b = -\lambda$ and $\sqrt{1+\lambda^2} = 1/\sigma_v$ so $\frac{a}{\sqrt{1+b^2}} = Q$. Therefore

$$P = \Phi(Q) + 2T(Q, -\lambda).$$

According to equation 2.6, p. 414 of Owen, $T(Q,-\lambda) = -T(Q,\lambda)$ and therefore

$$P = \Phi(Q) - 2T(Q, \lambda)$$

There is no closed form expression for the integral that defines $T(Q,\lambda)$, so all that we have done so far is to exchange one intractable integral for another. However, there is an exception, which is the case that $\lambda = 1$. Equation 2.3, p. 414, of Owen says that

$$T(Q, 1) = \frac{1}{2}\Phi(Q)[1 - \Phi(Q)].$$

Therefore when $\lambda = 1$ we have

$$P = \Phi(Q) - 2\left(\frac{1}{2}\right)\Phi(Q)[1 - \Phi(Q)] = \Phi^2(Q).$$

References

Aigner DJ, Lovell CAK, Schmidt P (1977) Formulation and estimation of stochastic frontier production function models. J Econ 6:21–37

Amsler C, Prokhorov A, Schmidt P (2014) Using copulas to model time dependence in stochastic frontier models. Econom Rev 33:497–522

Amsler C, Prokhorov A, Schmidt P (2016) Endogeneity in stochastic frontier models. J Econ 190:280–288

Amsler C, Prokhorov A, Schmidt P (2017) Endogenous environmental variables in stochastic frontier models. J Econ 199:131–140

Ashour SK, Abdul-Hameed MA (2010) Approximate skew normal distribution. J Adv Res 1:1–11

Carta A, Steel MFJ (2012) Modelling multi-output stochastic frontiers using copulas. Comput Stat Data Anal 56:3757–3773

Das A (2015) Copula-based stochastic frontier model with autocorrelated inefficiency. Cent Eur J Econ Model Econ 7:111–126

Genius M, Stefanou S, Tzouvelekas V (2012) Measuring productivity growth under factor non-substitution: an application to us steam-electric power generation utilities. Eur J Oper Res 220:844–852

Greene WH (2010) A stochastic frontier model with correction for sample selection. J Product Anal 34:15–24

Huang T-H, Chiang D-L, Chao S-W (2017) A new approach to jointly estimating the lerner index and cost efficiency for multi-output banks under a stochastic meta-frontier framework. Q Rev Econ Financ 65:212–226

Huang T-H, Liu N-H, Kumbhakar SC (2018) Joint estimation of the lerner index and cost efficiency using copula methods. Empir Econ 54:799–822

Lai H-P, Huang CJ (2013) Maximum likelihood estimation of seemingly unrelated stochastic frontier regressions. J Product Anal 40:1–14

Marsaglia G (2004) Evaluating the normal distribution. J Stat Softw 11:1–11

Meeusen W, van den Broeck J (1977) Efficiency estimation from cobb-douglas production functions with composed error. Int Econ Rev 18:435–444

Owen DB (1980) A table of normal integrals. Commun Stat: Simul Comput 9:389–419

Shi P, Zhang W (2011) A copula regression model for estimating firm efficiency in the insurance industry. J Appl Stat 38:2271–2287

Sriboonchitta S, Liu J, Wiboonpongse A, Denoeux T (2017) A double −copula stochastic frontier model with dependent error components and correction for sample selection. Int J Approx Reason 80:174–184

Tran KC, Tsionas EG (2015) Endogeneity in stochastic frontier models: copula approach without external instruments. Econ Lett 133:85–88

Tsay W-J, Huang CJ, Fu T-T, Ho L-L (2013) A simple closed form approximation for the cumulative distribution function of the composite error of stochastic frontier models. J Product Anal 39:259–269

Wang WS, Amsler C, Schmidt P (2011) Goodness of fit tests in stochastic frontier models. J Product Anal 35:95–118