



Computation of the Distribution of the Sum of Independent Negative Binomial Random Variables

Marc Girondot, Jon Barry

► To cite this version:

Marc Girondot, Jon Barry. Computation of the Distribution of the Sum of Independent Negative Binomial Random Variables. Mathematical and computational applications, 2023, 28 (3), pp.63. <10.3390/mca28030063>. <hal-04240863>

HAL Id: hal-04240863

<https://hal.science/hal-04240863v1>

Submitted on 2 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Article

Computation of the Distribution of the Sum of Independent Negative Binomial Random Variables

Marc Girondot ^{1,*}  and Jon Barry ²

¹ Laboratoire Écologie, Systématique et Évolution, Université Paris-Saclay, CNRS, AgroParisTech, 91190 Gif-sur-Yvette, France

² Lowestoft Laboratory, Centre for Environment, Fisheries and Aquaculture Science, Pakefield Road, Lowestoft, Suffolk NR33 0HT, UK

* Correspondence: marc.girondot@universite-paris-saclay.fr

Abstract: The distribution of the sum of negative binomial random variables has a special role in insurance mathematics, actuarial sciences, and ecology. Two methods to estimate this distribution have been published: a finite-sum exact expression and a series expression by convolution. We compare both methods, as well as a new normalized saddlepoint approximation, and normal and single distribution negative binomial approximations. We show that the exact series expression used lots of memory when the number of random variables was high (>7). The normalized saddlepoint approximation gives an output with a high relative error (around 3–5%), which can be a problem in some situations. The convolution method is a good compromise for applied practitioners, considering the amount of memory used, the computing time, and the precision of the estimates. However, a simplistic implementation of the algorithm could produce incorrect results due to the non-monotony of the convergence rate. The tolerance limit must be chosen depending on the expected magnitude order of the estimate, for which we used the answer generated by the saddlepoint approximation. Finally, the normal and negative binomial approximations should not be used, as they produced outputs with a very low accuracy.

Keywords: negative binomial distribution; computation; R package; sum of negative binomial variables



Citation: Girondot, M.; Barry, J. Computation of the Distribution of the Sum of Independent Negative Binomial Random Variables. *Math. Comput. Appl.* **2023**, *28*, 63. <https://doi.org/10.3390/mca28030063>

Academic Editor: Sandra Ferreira

Received: 7 February 2023

Revised: 13 April 2023

Accepted: 26 April 2023

Published: 28 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The negative binomial (NB) distribution is a discrete probability distribution that models counts [1]. It is widely used in statistics, from statistics of accidents [2] to animal counts [3]. The NB distribution can be used to describe the distribution of the number of successes or failures. Suppose that there is a sequence of independent Bernoulli trials, each trial having two potential outcomes called “success” and “failure”. The probability of success is p and of failure $q = 1 - p$. We observe this sequence until a predefined number, r , of successes has occurred. Then, the random number of failures has the NB distribution $X \sim NB(r; p)$ with density $P(X = x)$, x being a particular realization of X :

$$P(X = x) = \frac{(x + r - 1)!}{x!(r - 1)!} p^r (1 - p)^x \quad (1)$$

with $0 < p < 1$, x and r being integer > 0 . The mean is $\mu = r(1 - p)/p$.

The moment-generating function of the NB distribution is:

$$M(t) = \left(\frac{q}{1 - p e^t} \right)^r \quad (2)$$

An alternative parametrization $X \sim NB(\mu, \theta)$ can also be derived from assuming that the mean parameter of a Poisson distribution has a gamma distribution:

$$P(X = x) = \frac{\Gamma(x + \theta)}{x! \Gamma(\theta)} \left(\frac{\theta}{\mu + \theta} \right)^\theta \left(\frac{\mu}{\mu + \theta} \right)^x \quad (3)$$

with $\mu > 0$ and $\theta > 0$. Note that θ is not necessarily an integer, contrary to r in (1); hence, the gamma function is used in (3) instead of a factorial, with $\Gamma(x) = (x - 1)!$. The variance of the NB distribution is $\mu (1 + \mu/\theta)$. As θ approaches infinity, the NB distribution tends to follow the Poisson distribution, with the mean μ .

1.1. Sum of Negative Binomials

The sum of independent NB variables is of special interest in different contexts, such as the study of animal distribution [4,5], fecal egg counts in infected goats [6], the number of emergency medical calls [7], empirical distribution of the duration of wet periods in days [8] or insurance risk [9]. When the sum of several independent NB counts is available, determining the distribution of $\sum X_i$ with $X_i \sim NB(r_i; p_i)$ is a problem. When the p_i s are all the same and equal to p , a classical result is $\sum X_i \sim NB(\sum r_i; p)$ [10], but more general forms without this constraint are often needed. For example, if counts are available for various spatial or temporal units of the form $X \sim NB(\mu_i; r_i)$, p_i being $r_i/(\mu_i + r_i)$, it implies that the p_i s are not all the same, because μ_i varies among the units [4].

With the mean and variance of the $NB(r; p)$ distribution being $r(1 - p)/p$ and $r(1 - p)/p^2$, respectively, it follows that the mean and variance of the sum of n -independent NB variables are respective:

$$\text{mean}(S_n) = \sum_{i=1}^n (r_i(1 - p_i)/p_i) \text{ and } \text{var}(S_n) = \sum_{i=1}^n (r_i(1 - p_i)/p_i^2) \quad (4)$$

Our paper has developed some novel methods in relation to the practical computation and use of the convolution approach [9]. However, the paper also collects five different methods and presents them in one place, a useful resource for the working data scientist or statistician. We describe and reference these methods and outline the computational difficulties in getting them to work. We also point the reader to the freely available R software that implements each of the methods (plus a sixth method based purely on simulation).

Two methods have been published to estimate the distribution of the sum of NB independent variables using a finite-sum exact expression [11] or the convolution method [9]. However, the computer implementation of both methods was not available, and we have detected potential problems when a practitioner implements them. The finite-sum exact expression computer implementation is relatively straightforward, but memory overflow can occur, and the time of computing will increase as a function of the factorial of the number of observations, x . This precision was not given in the original publication [11]. The convolution method is very complex to implement and has been described as being “cumbersome” [12]; indeed, we found that its implementation was not straightforward and was even counterintuitive. The method uses a sum to infinity, and the condition to stop the recursion was not defined in the original publication.

Our solution for computation of the convolution method, presented here, is novel and has proven to be robust for extensive testing. A naïve tolerance condition has been used by one of the authors of this note (MG) (recursion stops when the change is lower than the tolerance limit) as in [4,5], but the other author (JB) found that outputs can be strongly biased in some conditions. It has been the beginning of a collaboration between the two authors to understand and solve the origin of this bias. We detected two problems: (1) the tolerance check must be applied only when the first-order change of the estimate is negative (convergence criteria being adaptive), and (2) the value of the tolerance must be proportional to the expected estimate. Then, it was necessary to have an estimate of the density to set the tolerance, to better estimate the correct density. To solve this, we used the saddlepoint approximation of the density. We show that the absolute error of this

approximation can be on the order of 5%, being too high to be used in many applications, but it is sufficiently low to define a correct tolerance to be used with the convolution method.

1.2. Normal and Negative Binomial Approximations

When working on the sum of variables, the first thought is to use the central limit theorem [13] that establishes that, in many situations, the distribution of the sum-independent random variables tends to go toward a normal distribution. An alternative is to model the distribution of the sum of NB variables, as an NB distribution is based on the observation that the distribution of the sum of NB variables is a mixture NB distribution [9], according to Theorem 2, proposed by Makun, Abdulganiyu, Shaibu, Otaru, Okubanjo, Kudi, and Notter [6].

1.3. Finite-Sum Exact Expression

An exact form for the distribution of the sum of NB is:

$$P(S_n = x) = \sum_{\mu_1 + \dots + \mu_n = x} \prod_{j=1}^n \frac{\Gamma(\mu_j + \theta)}{\mu_j! \Gamma(r\theta_j)} p_j^{\theta_j} q_j^{\mu_j} \quad (5)$$

The expression (5) is compact and the exact value can be computed [11].

1.4. Approximation by Convolution

When $X_i \sim NB(r_i; p_i)$, with i from 1 to n , the distribution of $S_n = \sum X_i$ is a mixture NB [9], with the probability mass function being approximated by:

$$P(S_n = x) = R \sum_{k=0}^{\infty} \delta_k \frac{\Gamma(r + x + k)}{\Gamma(r + k)x!} M_1^{r+k} (1 - M_1)^x, \quad x = 0, 1, 2, \dots \quad (6)$$

$$\begin{aligned} &\text{where } r = \sum_{i=1}^n r_i, \text{ and } M_1 = \max_j(p_j). \\ &R = \prod_{j=1}^n \left(\frac{q_j M_1}{(1 - M_1)p_j} \right)^{-r_j} \\ &\text{and } \delta_{k+1} = \frac{1}{k+1} \sum_{i=1}^{k+1} i \xi_i \delta_{k+1-i}, \quad k = 0, 1, \dots \text{ with } \delta_0 = 1 \text{ and} \\ &\xi_i = \sum_{j=1}^n \frac{r_j (1 - (1 - M_1)p_j / q_j M_1)^i}{i} \end{aligned}$$

Expression (6) is used iteratively, with k being the counter of the rank of iterations, but a condition to stop the iterations when a certain level of approximation is reached was not defined in the original publication [9].

1.5. Saddlepoint Approximation

The saddlepoint approximation method provides a highly accurate approximation formula for any probability density function (continuous distribution) or probability mass function (discrete distribution) of a distribution, based on the moment-generating function [14].

Taking the log of the moment-generating function of the NB distribution (2) and summing over n -independent NB variables, the cumulant of sum of NBs is:

$$K(t) = \sum_{i=1}^n r_i (\log(q_i) - \log(1 - p_i e^t))$$

$$\text{Or } K(t) = \sum_{i=1}^n \theta_i (\log(\theta_i) - \log(\theta_i + \mu_i (1 - e^t))) \quad (7)$$

The first and second order of the derivatives of $K(t)$ are:

$$K'(t) = \sum_{i=1}^n \frac{\theta_i \mu_i e^t}{\theta_i + \mu_i (1 - e^t)} \quad (8)$$

$$K''(t) = \sum_{i=1}^n \frac{\theta_i \mu_i (\theta_i + \mu_i) e^t}{(\theta_i + \mu_i (1 - e^t))^2} \quad (9)$$

The saddlepoint, s_x , is found by solving $K'(s_x) = x$. Once s_x is found, $P(S_n = x)$ can be approximated by:

$$P(S_n = x) \approx \frac{1}{\sqrt{2\pi K''(s_x)}} e^{(K(s_x) - x s_x)} \quad (10)$$

The value $P(S_n = x)$ is normalized to ensure that $\sum P(S_n) = 1$.

In the remainder of this note, we describe the computational problems that applied statisticians or practitioners face in implementing the distribution of the sum of NB-independent variables using finite-sum exact expression [11], the convolution method [9], saddlepoint approximation, or the approximation by normal and NB distributions. We describe how these have been overcome in the publicly available R package (HelpersMG package version 5.9 and higher (<https://CRAN.R-project.org/package=HelpersMG>, accessed on 6 February 2023)). The code can be checked after loading this package with the command `?dSnbinom`.

2. Computations

Figure 1 gives two examples of the sum S of independent NB random variables, and how these distributions are approximated using the four methods (convolution, saddlepoint, single normal, single NB) outlined in this note. In (A), we use $n = 10$, $j = 1 \dots n$, $p_j = 0.4 + \frac{j}{10}$ and $r_j = j \times 10$, and in (B) $n = 2$, $p_j = \frac{j}{10}$, and $r_j = j$.

2.1. Normal and Negative Binomial Approximations

When n is large and standard deviation is small as compared to the mean, the normal approximation with $P(S_n = x) = \int_{x-0.5}^{x+0.5} \mathcal{N}(\mu, \sigma)$, where $\mathcal{N}(\mu, \sigma)$ is the normal probability density function with $\mu = \text{mean}(S_n)$ and $\sigma = \sqrt{\text{var}(S_n)}$ can be used as an approximation for the distribution of the sum of independent NB random variables (Figure 1A). However, for a small n or large standard deviation, as compared to the mean (corresponding to a highly skewed distribution), the approximation can be very poor (Figure 1B). The NB distribution modeled with the probability density function, $NB(\mu, \theta)$, such that $\mu = \text{mean}(S_n)$ and $\theta = \text{mean}(S_n)^2 / (\text{var}(S_n) - \text{mean}(S_n))$, better fits the exact distribution of the sum of NB variables, but still with a bias (Figure 1B). This confirms that the distribution of the sum of independent NB variables is not an NB, as wrongly stated in [6]. It is, rather, a mixture NB (see below) [9]. In summary, the normal and NB approximations generate the highest errors (>30% in some cases) and they should not be used, especially as there are better alternatives.

2.2. Finite-Sum Exact Expression

This method permits the calculation of the exact value for $P(S_n = x)$. It will therefore be used as a reference here.

For the finite form exact expression method [11], a table of n columns with all the combinations of integers, from 0 to x , that produce a sum of x ($m_1 + \dots + m_n = x$), must be first established. The number of different ways to distribute x -indistinguishable objects into n -distinguishable categories is $C(x + n - 1, n - 1)$. This is the memory-consuming part of the Vellaisamy and Upadhye [11] method. The density $P(X = x)$ in Equation (1) is calculated n times for each of these combinations in the final table (the $\prod_{j=1}^n$ part of Equation (5)). This is the computationally time-consuming part of the method.

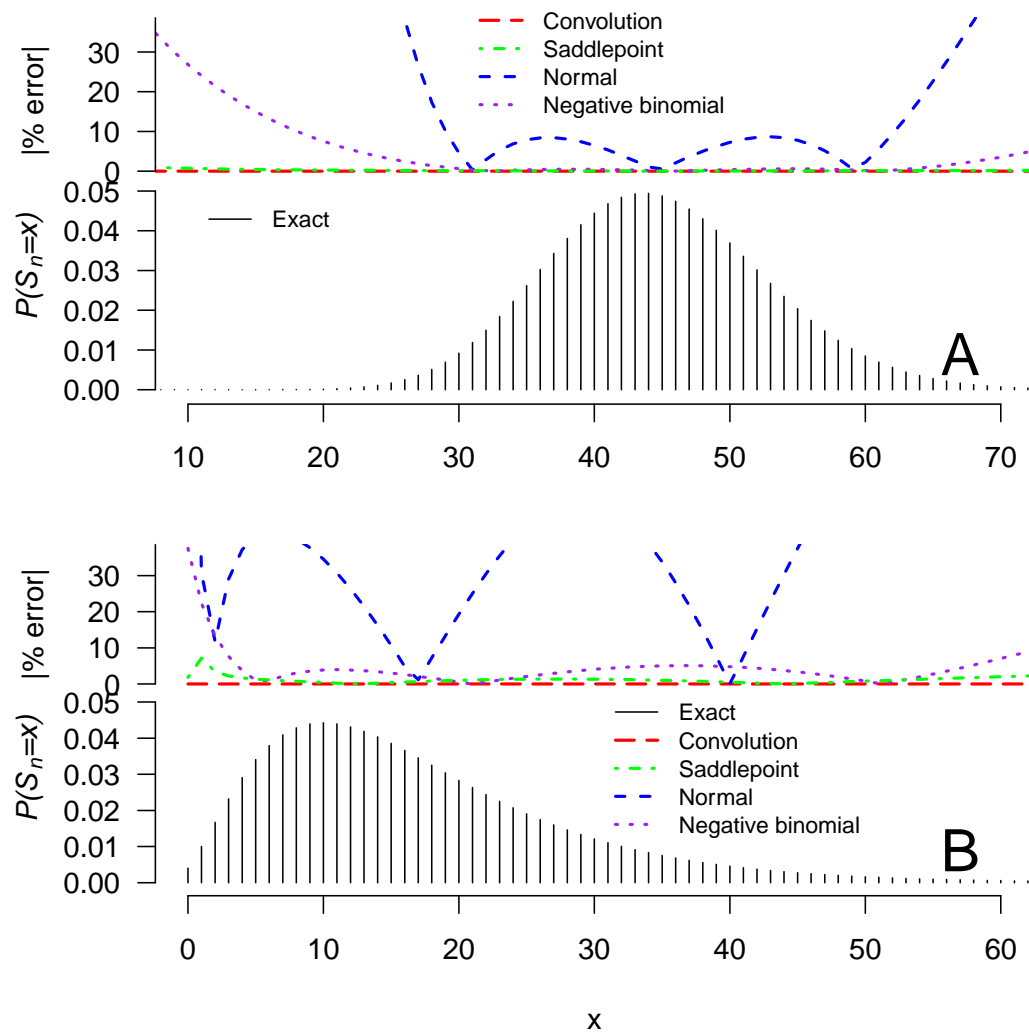


Figure 1. Sum of independent NB distributions approximated with convolution, saddlepoint approximation, normal, and single NB distribution. (A) $n = 10$, $j = 1 \dots n$, $p_j = 0.4 + \frac{j}{10}$ and $r_j = j \times 10$; $\text{mean}(S_n) = 183.92$, $\text{var}(S_n) = 270.75$. (B) $n = 2$, $p_j = \frac{j}{10}$ and $r_j = j$; $\text{mean}(S_n) = 17$, $\text{var}(S_n) = 130$. The bar plots show the exact distribution, and the top graphs show the absolute % of error of the approximation.

When n and/or x are large, this method requires lot of iterations. For example, there are 1,081,575 different combinations of 17 objects in nine categories. Then, Equation (1) must be applied 9,734,175 times to estimate $P(S_n = x)$ when using Equation (5).

2.3. Approximation by Convolution

The coefficients of Equation (6) are iteratively defined, and we rewrite the published formula to make the computation more efficient using the recursion:

$$W(S_n = x)_0 = \frac{\Gamma(r+x)}{\Gamma(r)x!} M_1^r (1 - M_1)^x$$

$$W(S_n = x)_{k+1} = W(S_n = x)_k + \delta_{k+1} \frac{\Gamma(r+x+k+1)}{\Gamma(r+k+1)x!} M_1^{r+k+1} (1 - M_1)^x$$

$$P(S_n = x)_k = R W(S_n = x)_k \quad (11)$$

Intermediate estimates in (11) used log of expressions to prevent a computing overflow. The conditions to stop the iterations were not defined in Furman's original publication.

A typical method in such situations is to stop the recursion when the change in the final output is below a defined tolerance. However, it cannot be used in the context of Equations (6) or (11), because, at the beginning of iterations, the change in $P(S_n = x)$ is sometimes so small that recursions will be stopped immediately and the resulting probability $P(S_n = x)$ will be biased. An example of this is shown in Figure 2A, which shows the value of $P(S_7 = 6)_k$ ($n = 7, j = 1 \dots n, p_j = j/10$, and $r_j = j$) as a function of the recursive iterations k from Equation (6). For the first eight iterations, the change in $P(S_7 = 6)_k$ is very small. To alleviate this problem, many iterations can be used, but without being sure that they are sufficient, and it is done at the expense of running time. This solution was chosen with at least 1000 iterations in [5], but this number of iterations is not always large enough to ensure a correct estimate when n or x are large.

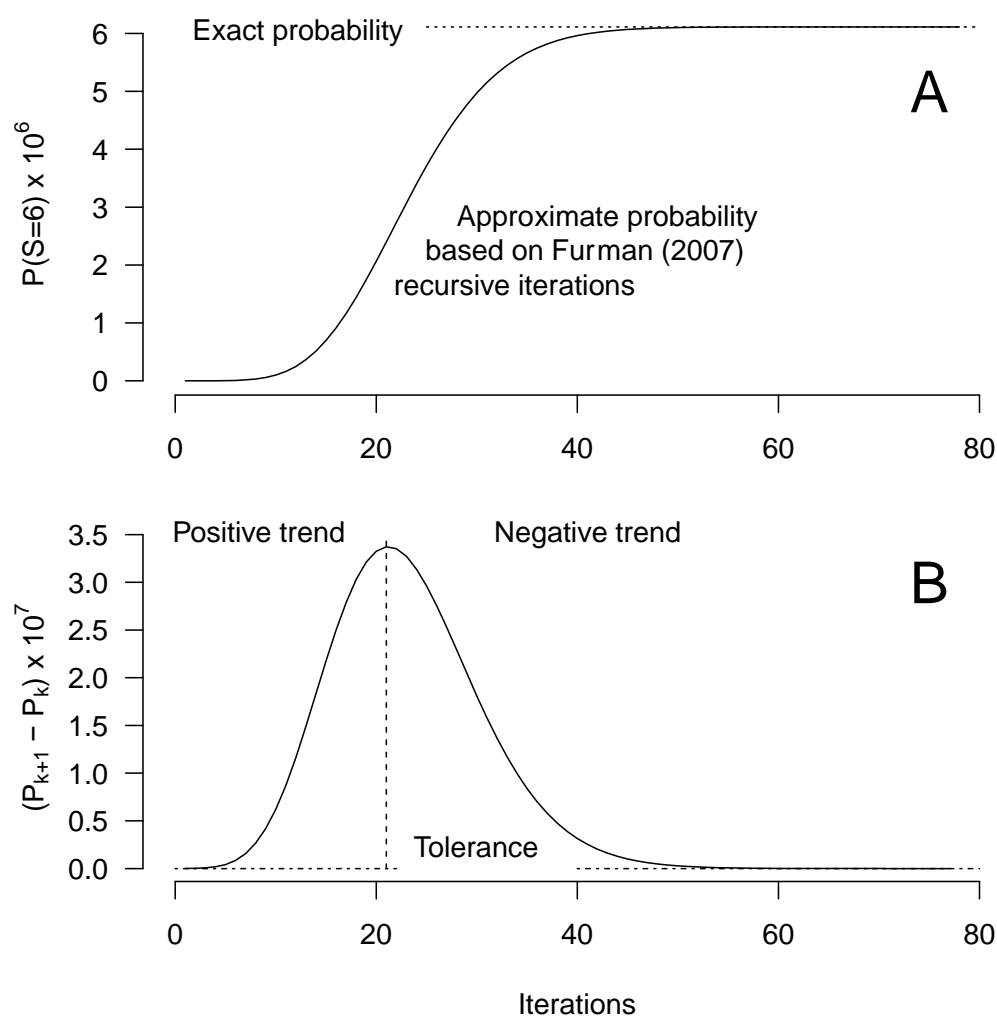


Figure 2. (A) Dynamics of the convergence of $P(S_7 = 6)_k$ with $n = 7, j = 1 \dots n, p_j = j/10$, and $r_j = j$ using Equation (6), with k being the rank of iterations. (B) Trend of the changes in P_k with tolerance $= 10^{-12}$.

A better approach came from the study on the trend of the rate of change of $P(S_n = x)$ according to the rank of iteration k : $P_k - P_{k-1}$ vs. $P_{k+1} - P_k$, where P_k denotes the value of $P(S_n = x)$ at iteration k . In its initial phase, the rate of change of P is positive, with $P_k - P_{k-1} > P_{k+1} - P_k$, then it shows a peak and becomes negative, with $P_k - P_{k-1} < P_{k+1} - P_k$ (Figure 2B). The tolerance threshold must be used only after the occurrence of the peak to ensure that the phase of rapid change of P is reached. The number of iterations before the peak is dependent on the values of n, x, r_i and p_i , and cannot be easily anticipated at the beginning of the iterations. We have therefore developed

an adaptative strategy to stop the recursion when two conditions are met: the rate of change of $P(S_n = x)$ is negative, and the change of $P(S_n = x)$ is less than the user-defined tolerance. The tolerance value must be lower than $P(S_n = x)$ or the output will be biased. As an example, if $\mu = (0.01, 0.02, 0.03)$ and $\theta = (2, 2, 2)$, then $P(S_3 = 20) = 7.73139 \times 10^{-35}$ using the exact method. If the Furman method is used with the tolerance set to 10^{-12} , $P(S_3 = 20) = 3.879379 \times 10^{-35}$, which is two times lower than the exact answer. The solution is to define a tolerance much lower than the anticipated results, for example, here, with the tolerance being 10^{-45} , $P(S_3 = 20) = 7.73139 \times 10^{-35}$, which is the correct probability. This can be done using the saddlepoint estimate (see below).

The comparison of the results obtained by Equation (6), with an adaptative stopping of the recursion and tolerance setup, using saddlepoint approximation (see below) and Equation (5), are shown in Table 1 with the corresponding computing time. This table is similar to those used in Tables 1 and 2 in Vellaisamy and Upadhye [11].

Table 1. Comparison of accuracy and computing time of the sum of n numbers $x = 3, 5, 8, 10$, and 15 obtained from NB distributions with $j = 1 \dots n$, $p_j = j/10$, $r_j = j$, and n from 2 to 7 based on Equations (5), (11), and (10). For each (n, x) combination in (A), the top number is the probability $P(S_n = x)$ and the bottom number is the number of iterations. In (B), the number of recursions required to stabilize $P(S_n = x)$ is shown. The $P(S_n = x)$ values are exactly the same as those in (A) and are not shown. In (C), the $P(S_n = x)$ values for saddlepoint approximation are shown. Computing times for the different methods are shown at the right of each table. The code for Equation (5) was parallelized on an 8-core computer in R 4.2.3 and HelpersMG package version 5.9 (<https://CRAN.R-project.org/package=HelpersMG>, accessed on 6 February 2023).

A: Vellaisamy and Upadhye [11]: Exact Probabilities						No Parallel	Parallel 8-Core
	$x = 3$	$x = 5$	$x = 8$	$x = 10$	$x = 15$	Time (s)	Time (s)
$n = 2$	0.02320400 16	0.03403236 36	0.04283461 81	0.04425234 121	0.03856123 256	0.001	0.011
$n = 3$	0.00273650 40	0.00730772 126	0.01724312 405	0.02421915 726	0.03607386 2176	0.003	0.011
$n = 4$	0.00020980 80	0.00094784 336	0.00408465 1485	0.00785680 3146	0.02099302 13,056	0.014	0.012
$n = 5$	0.00001503 140	0.00010490 756	0.00076597 4455	0.00196540 11,011	0.00920145 62,016	0.062	0.015
$n = 6$	0.00000131 224	0.00001291 1512	0.00014555 11,583	0.00047692 33,033	0.00365038 248,064	0.249	0.023
$n = 7$	0.00000017 336	0.00000218 2772	0.00003427 27,027	0.00013604 88,088	0.00154413 868,224	0.906	0.049
B: Furman [9]: Convolution						Tol = $P_{\text{saddlepoint}}(S_n = x) \times 10^{-10}$	
	$x = 3$	$x = 5$	$x = 8$	$x = 10$	$x = 15$	Time (s)	
$n = 2$	13	14	15	16	18	0.007	
$n = 3$	19	20	23	24	27	0.008	
$n = 4$	27	29	32	34	38	0.009	
$n = 5$	39	42	45	48	54	0.009	
$n = 6$	58	62	67	70	79	0.009	
$n = 7$	92	97	104	109	122	0.011	
C: Normalized Saddlepoint Approximation							
	$x = 3$	$x = 5$	$x = 8$	$x = 10$	$x = 15$	Time (s)	
$n = 2$	0.02372254	0.03448835	0.04314218	0.04442429	0.03841261	0.007	
$n = 3$	0.00283042	0.00748306	0.01754862	0.02458058	0.03637448	0.007	
$n = 4$	0.00021836	0.00097613	0.00418037	0.00802118	0.02132508	0.008	
$n = 5$	0.00001571	0.00010840	0.00078653	0.00201341	0.00938611	0.008	
$n = 6$	0.00000137	0.00001337	0.00014977	0.00048960	0.00373283	0.008	
$n = 7$	0.00000018	0.00000226	0.00003531	0.00013984	0.00158133	0.018	

2.4. Saddlepoint Approximation

The saddlepoint approximation (we used the Brent' algorithm [15] for the minimization needed to find the saddlepoint) is computationally fast. However, the estimate must be normalized so that the density function sums to 1 [16]. The normalization used the sum of $P(S_n = x)$, with x from 0 to $\text{mean}(S_n) + \text{Max}\sqrt{\text{var}(S_n)}$, with $\text{mean}(S_n)$ and $\text{var}(S_n)$ from Equation (4) and $\text{Max} = 20$. A test was performed to ensure that $P(S_n = \text{mean}(S_n) + \text{Max}\sqrt{\text{var}(S_n)})$ was 0 or that the Max was increased until this condition was reached. The relative difference between the exact value of $P(S_n = x)$ and the saddlepoint approximation can be sometimes of the order of 5% (Figure 1). On the other hand, this approximation is good enough to set the tolerance of the approximation by convolution, using a tolerance equal to $P_{\text{saddlepoint}}(S_n = x) \times 10^{-10}$.

The tolerance value to cut the iterations for an approximate Furman [9] method must be of the same order as the value of $P(S_n = x)$, multiplied by the tolerance and set at the value of 10^{-10} , to have an estimate precise up to the 10th digit. The difficulty is that $P(S_n = x)$ needs to be estimated here. The chosen solution was to use a rough estimate of $P(S_n = x)$ from the very fast saddlepoint approximation method first. This approach proved to be very efficient, because the estimates of the approximate Furman [9] method are exactly the same as for the exact method (Table 1A).

Equation (5) has the advantage that it is parallelizable, but for a large n and x (see Table 1A), doing so requires a large number of both iterations and memory. Equations (6) and (11), however, are not disadvantaged by these problems. Vellaisamy and Upadhye [11] indicated that Equation (5) required less computing time than Equation (6), even for $n = 7$ and $n = 15$. This would be true only if the authors used a very large number of iterations to stop the iterations in Equation (6), or if their conditions to stop the iterations were sub-optimal.

As a general conclusion, we consider that the approximate form of distribution for the sum of independent NB proposed by Furman [9] should be used in all the contexts, whatever the parameters n , x , p_i or r_i . The tolerance can be approximated by using the value of $P(S_n = x)$, estimated using the saddlepoint approximation method. This solution is used by default in the R package HelpersMG (version > 5.9), available in CRAN: The Comprehensive R Archive Network [17].

Author Contributions: Conceptualization, M.G. and J.B.; software, M.G. and J.B.; validation, M.G. and J.B.; writing, M.G. and J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All methods used in this note, as well as an approximation using the generation of random numbers, are available in the functions `dSnbinom`, `pSnbinom`, `qSnbinom`, and `rSnbinom` in the R package HelpersMG (version > 5.9), available in CRAN: The Comprehensive R Archive Network [17].

Acknowledgments: We thank all four reviewers for their positive and helpful comments. In particular, one referee pointed us to the saddlepoint approach, of which we were previously unaware.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fisher, R.A.; Corbet, A.S.; Williams, C.B. The relation between the number of species and the number of individuals in a random sample of an animal population. *J. Anim. Ecol.* **1943**, *12*, 42–58. [CrossRef]
2. Carlson, T. Negative binomial rationale. *Proc. Casualty Actuar. Soc.* **1962**, *49*, 177–183.
3. Power, J.H.; Moser, E.B. Linear model analysis of net catch data using the negative binomial distribution. *Can. J. Fish. Aquat. Sci.* **1999**, *56*, 191–200. [CrossRef]
4. Girondot, M. Optimizing sampling design to infer marine turtles seasonal nest number for low-and high-density nesting beach using convolution of negative binomial distribution. *Ecol. Indic.* **2017**, *81*, 83–89. [CrossRef]

5. Omeyer, L.C.M.; McKinley, T.J.; Bréheret, N.; Bal, G.; Balchin, G.P.; Bitsindou, A.; Chauvet, E.; Collins, T.; Curran, B.K.; Formia, A.; et al. Missing data in sea turtle population monitoring: A Bayesian statistical framework accounting for incomplete sampling. *Front Mar. Sci.* **2022**, *9*, 817014. [[CrossRef](#)]
6. Makun, H.J.; Abdulganiyu, K.A.; Shaibu, S.; Otaru, S.M.; Okubanjo, O.O.; Kudi, C.A.; Notter, D.R. Phenotypic resistance of indigenous goat breeds to infection with *Haemonchus contortus* in northwestern Nigeria. *Trop. Anim. Health Prod.* **2020**, *52*, 79–87. [[CrossRef](#)] [[PubMed](#)]
7. Lee, H.; Lee, T. Demand modelling for emergency medical service system with multiple casualties cases: K-inflated mixture regression model. *Flex. Serv. Manuf. J.* **2021**, *33*, 1090–1115. [[CrossRef](#)]
8. Korolev, V.; Gorshenin, A. Probability models and statistical tests for extreme precipitation based on generalized negative binomial distributions. *Mathematics* **2020**, *8*, 604. [[CrossRef](#)]
9. Furman, E. On the convolution of the negative binomial random variables. *Stat. Probab. Lett.* **2007**, *77*, 169–172. [[CrossRef](#)]
10. Johnson, N.; Kotz, S.; Kemp, A. *Univariate Discrete Distributions*, 2nd ed.; Wiley: New York, NY, USA, 1992.
11. Vellaisamy, P.; Upadhye, N.S. On the sums of compound negative binomial and gamma random variables. *J. Appl. Probab.* **2009**, *46*, 272–283. [[CrossRef](#)]
12. Baena-Mirabete, S.; Puig, P. Computing probabilities of integer-valued random variables by recurrence relations. *Stat. Probab. Lett.* **2020**, *161*, 108719. [[CrossRef](#)]
13. Laplace, P.-S. Mémoire sur les approximations des formules qui sont fonctions de très grands nombres, et sur leur application aux probabilités. *Mémoires Cl. Sci. Mathématiques Phys. L'institut Fr.* **1809**, *1809*, 353–415.
14. Daniels, H.E. Saddlepoint approximations in statistics. *Ann. Math. Stat.* **1954**, *25*, 631–650. [[CrossRef](#)]
15. Brent, R. *Algorithms for Minimization without Derivatives*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1973.
16. Lugannani, R.; Rice, S. Saddle point approximation for the distribution of the sum of independent random variables. *Adv. Appl. Probab.* **2016**, *12*, 475–490. [[CrossRef](#)]
17. Girondot, M. *HelpersMG: Tools for Environmental Analyses, Ecotoxicology and Various R Functions*; The Comprehensive R Archive Network: Indianapolis, IN, USA, 2023.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.