

NOTE TO FILE:
Garvin H Boyle
Dated: R4: 180214

About Error in FactLn() and GammaLn() Implementations

Frontispiece

Figure 01

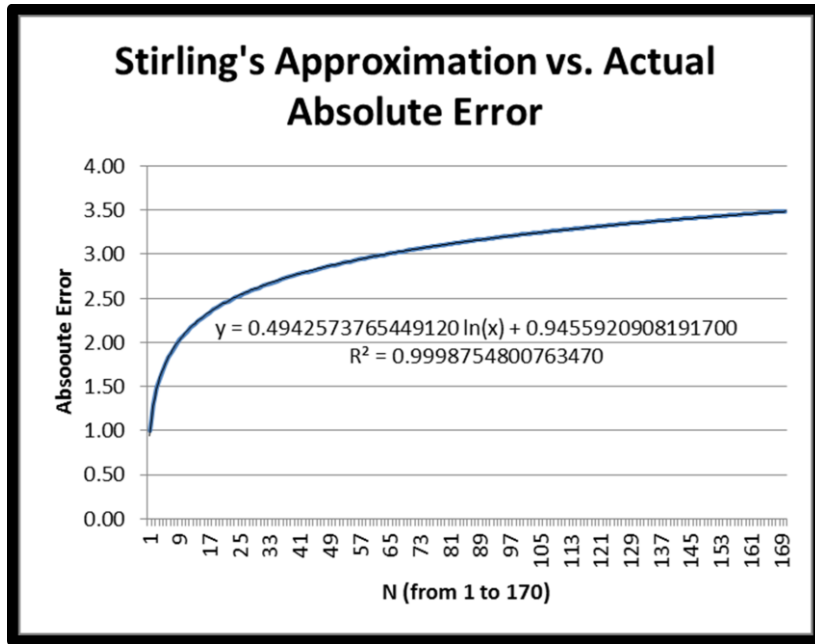


Table of Contents

1 - References	1
2 - Background.....	1
3 - Purpose	1
4 - Discussion.....	2
4.1 - A Two-Part Problem.....	2
4.2 - Examination of $\ln(x!)$ When $x > 170$	2
4.2.1 - Stirling's Approximation – Basic Form	2
4.2.2 - Definition of Entropy for ABM-Related Histograms	3
4.2.3 - Error Evaluations	3
4.2.4 - Summary for $\ln(x!)$	6
4.3 - Examination of $\text{GammaLn}(x)$	7
4.3.1 - Lanczos Approximation of $\Gamma(X+1)$	7
4.3.2 - Spouge's Approximation of $\Gamma(x+1)$	8
4.3.3 - Numerical Recipe for $\text{GammaLn}(X+1)$	8
4.3.4 - Test of $\text{GammaLn}(x)$ for S_{\max}	9
5 - Conclusion	11

1 - References

- A. A. Drăgulescu and V.M. Yakovenko (2000) “Statistical mechanics of money”, Eur. Phys. J. B 17, 723-729.
- B. 140218 Email from Yakovenko R1.pdf
- C. 140409 NTF Discussion With Dr Yakovenko R1.pdf
- D. 150527 PPR - Definition of EI R17.pdf
- E. 180118 NTF Custom Functions in Excel R3.pdf
- F. 180210 NTF Shannon Vs Boltzmann R3.pdf
- G. 180115 XLS Stirling's Approximation R4.xlsm
- H. https://en.wikipedia.org/wiki/Stirling%27s_approximation
- I. https://en.wikipedia.org/wiki/Spouge%27s_approximation
- J. https://en.wikipedia.org/wiki/Lanczos_approximation
- K. Press, Teukolsky, Vetterling and Flannery (1986) “Numerical Recipes in Fortran”, Cambridge, pp 206-209
- L. 180219 Test of GammaLn for Entropic Index R2.xlsx

2 - Background

This note is written in an attempt to understand exactly how Stirling’s approximation to $\ln(A!)$ works, and also to understand the GammaLn() workaround. The context of my intended usage is the calculation of the entropy associated with a histogram derived from an agent-based model. For example, if you create a histogram of the distribution of wealth of a set of agents, there may be many bins with a small number of agents, and some bins with a lot of agents. I need a formula that works well for numbers on the interval $[0, \infty]$ with special emphasis on the very low segment of that domain $[0, 10]$.

Ref A is a paper in which Yakovenko introduces some seminal ideas about calculating entropy in agent-based models. Refs B through F is a series of diary notes in which I examine the nature of entropy as it applies to histograms of conserved quantities in agent-based models.

Ref G is the MS Excel spreadsheet in which I tested some entropy equations and constructed the associated graphs for this diary note.

Refs H, I and J are very informative online articles about the origins and nature of Stirling’s approximation for $\ln(x!)$ and the approximations of Spouge and Lanczos for $\ln(\Gamma(x))$.

Ref K is a book in which one can find short-cut algorithms for implementing $\ln(x!)$ and $\ln(\Gamma(x))$ on a computer.

3 - Purpose

To understand the operational characteristics of the mathematical formulae used to approximate $\ln(x!)$ and GammaLn(x).

4 - Discussion

4.1 - A Two-Part Problem

I need to be able to evaluate $\ln(x!)$ under two different types of circumstances that are normally difficult:

- Many computer programs (most) cannot calculate $\ln(x!)$ when x is greater than 170 due to overflow conditions. The final answer is not too large to express, but, typically, the intermediate value is. Standard computer implementations cannot handle $x!$ for x greater than 170. $\ln(x!)$ rises faster than $\ln(x)$ but more slowly than $x!$. The standard way to avoid this overflow problem is to approximate the answer using an alternate function which does not have to calculate $x!$ first. In particular, some variation of Stirling's Approximation of $\ln(x!)$ is used.
- But also, to calculate the maximum value of the entropy for a histogram, I will need to calculate the lawn of the factorial of a non-integer argument. For example, I may need to calculate $\ln(3.2!)$, and the factorial function is defined only for integral arguments. There is another function called the Gamma function (usually represented as $\Gamma(x)$) which produces the same curve as $x!$ when x is integral, but is also able to produce a smooth curve for non-integral numbers in between. They are related as $x! = \Gamma(x+1)$ for all integral $x \geq 0$. What I will need is the lawn of the Gamma function, and in many computer mathematical libraries there is a function called `GammaLn()` which can be used to evaluate $\ln(x!) \approx \ln(\Gamma(x+1))$.

In short, I need a special replacement for $\ln(x!)$ for when $x > 170$ or when $x \notin \text{Integers}$.

Before I proceed with this, just for clarification purposes, Stirling's Approximation comes into my deliberations in another similar but subtly different way. Boltzmann's version of the equation for entropy can be converted to Shannon's version of the equation for entropy through the substitution of the basic form of Stirling's approximation into Boltzmann's equation, after the Ω is replaced with the multinomial coefficient $\Omega = A! / \prod_{i=1}^K [a_i!]$. For a detailed mathematical derivation, and an exploration of the error this creates between the two famous formulae, see Ref F. So, 'Stirling's Approximation' plays some role in distinguishing between what I am calling the 'Boltzmann regime' and the 'Shannon regime'. I decided to go forward with the Boltzmann regime to avoid the systemic bias introduced into the analytic development of the Shannon regime. Within the so-called Boltzmann regime, nevertheless, to evaluate the equations of the Boltzmann regime, I still need to depend on Stirling's approximation to $\ln(x!)$ and/or the `GammaLn()` function. I need to understand how this bias works, and how to minimize its effects.

4.2 - Examination of $\ln(x!)$ When $x > 170$

4.2.1 - Stirling's Approximation – Basic Form

This is the standard basic version of "Stirling's approximation" for $\ln(A!)$:

$$\ln(A!) \approx (A \ln(A)) - A \quad [1]$$

It is used in the formulae for entropy when $A!$ is too large to be calculated. I say "basic" because Stirling's formula is actually an infinite series of ever-more-accurate terms, but equation [1] is the version most often quoted.

4.2.2 - Definition of Entropy for ABM-Related Histograms

This is the standard equation for the definition of entropy of a histogram within what I am calling the ‘Boltzmann regime’:

$${}_B S = {}_B C \times \ln(\Omega) \quad [2]$$

where ${}_B C$ is a dimensionless scaling factor and Ω is the total number of possible microstates associated with one configuration of the histogram. Ω can be calculated using the multinomial coefficient from combinatorial mathematical theory:

$$\Omega = \frac{A!}{\prod_{i=1}^K (a_i!)} \quad [3]$$

in a K-bin histogram, where A is the number of agents, and a_i is the number of agents in bin i of the histogram.

$${}_B S = {}_B C \times \ln\left(\frac{A!}{\prod_{i=1}^K (a_i!)}\right) = {}_B C [\ln(A!) - \sum_{i=1}^K \ln(a_i!)] \quad [4]$$

But neither MS Excel nor the C++ mathematical libraries can calculate $\ln(A!)$ if $A > 170$ due to overflow. That means it requires too many bits to accurately store the number in the computer’s memory. 170 is a pretty small number. Many runs of my ABMs have more than 170 agents. But what is much more concerning, almost all circumstances for which I calculate entropy, most of the bins contain less than 170 agents. To usefully compare my analytical results (developed in MS Excel) with my experimental results (developed using C++ in the EiLab application), I need to consistently use Stirling’s approximation, but only whenever one of the following happens on any and all implementation platforms:

- $A > 170$; or
- $a_i > 170$.

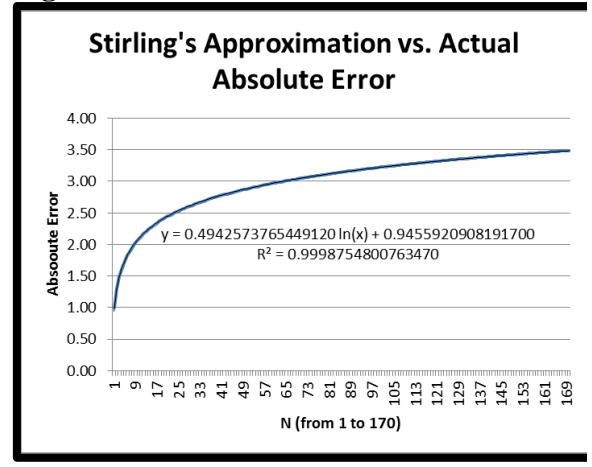
4.2.3 - Error Evaluations

Stirling’s approximation for the function $\ln(x!)$ is actually defined for the complex plane Z (the Arcand plane) and only the real portion is of interest to me. Furthermore, it can be written as a rather complex infinite series that approximates the $\ln(x!)$ more and more closely as terms are added. The most basic form of Stirling’s approximation is $\ln(x!) = (x \ln(x) - x)$. I start there.

If you calculate the exact value of $\ln(N!)$ for N in [1, 170], and Stirling’s approximation of $\ln(N!)$ for the same range of N, and take the difference as a % of $\ln(N!)$, you get a pair of interesting graphs. In preparing the first graph (see Figure 01) I noted three things:

- $\ln(0!)$ is a troublesome value to compute, and I must provide a work-around for it;
- The absolute error is rising; and
- The logarithmic trend line produced by least squares techniques has parameters that can be useful as a correction to the basic form of Stirling’s Approximation.

Figure 01



The blue line is the absolute error. The black line is a logarithmic trend line produced by MS Excel using a “least squares” technique, with the equation for the trend line. I altered the formatting of the equation of the trend line to give me 16 digits of accuracy. Note that the R-squared value is very high.

In the second graph (see Figure 02) the relative error is immense for small N, but when you get to N = 170 the relative error is 0.005%. The absolute error is rising, but the value of ln(N!) is rising faster, so the relative error continues to decline as N gets larger. But for small N, in the area where my ABMs will be active, the relative error in Stirling’s approximation can be quite large. My decision to avoid the use of Stirling’s approximation for small A and small a_i is based on these two graphs.

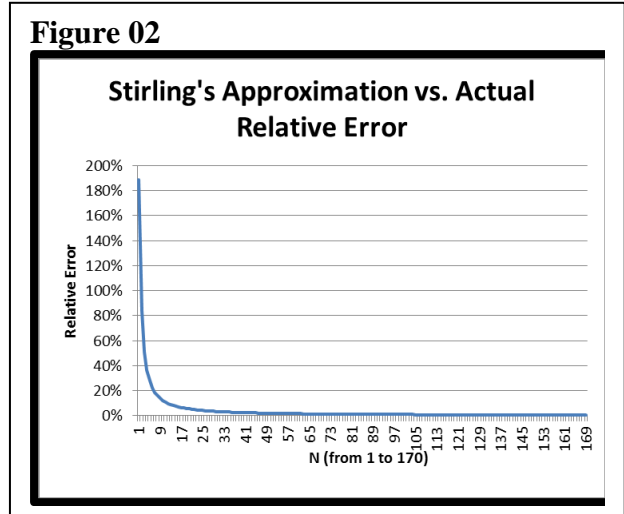
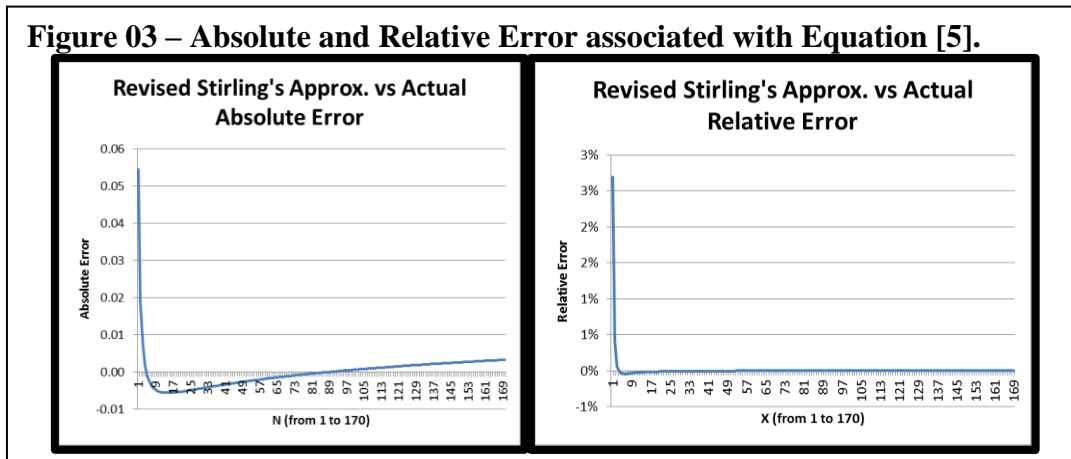


Figure 01 suggests that an improvement on equation [1] might be:

$$\ln(A!) \approx (A + 0.494257376544912) \ln(A) + (0.945592090819170 - A) \quad [5]$$

Figure 03 presents the graphs for absolute and relative error for equation [5].

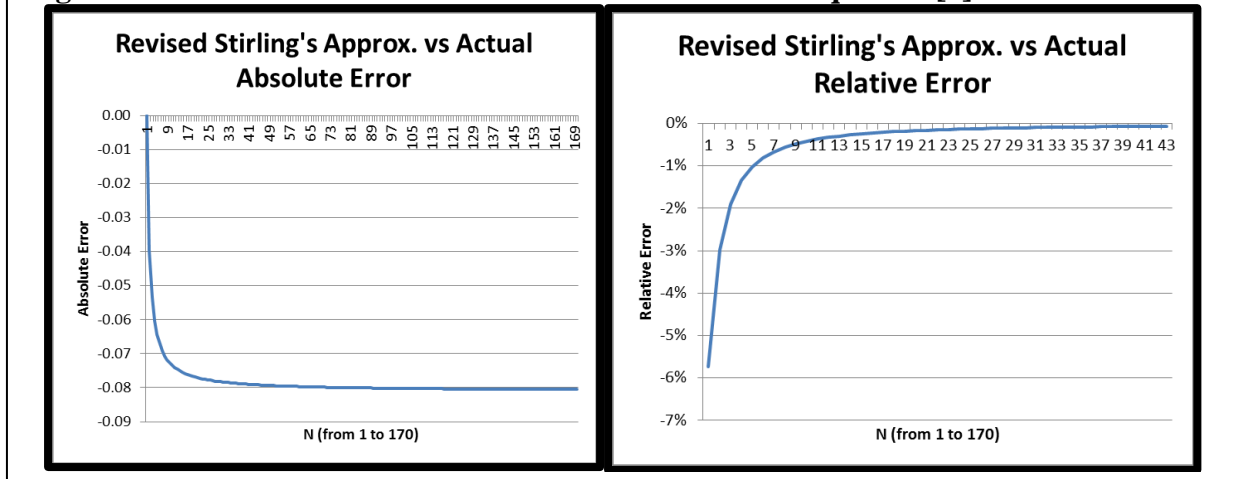


Or more simply, equation [5] could be approximated by:

$$\ln(A!) \approx (A + 0.5) \ln(A) + (1 - A) \quad [6]$$

Figure 04 presents the graphs for absolute and relative error for equation [6].

Figure 04 – Absolute and Relative Error associated with Equation [6]



I note that, however you present Stirling’s Approximation of $\ln(x!)$, the absolute error is the largest when x is near zero. But in any histogram of a distribution of wealth there will almost always be some empty bins. So, using Stirling’s approximation for values of $\ln(x!)$ when x is close to zero will cause serious distortions.

I also note that the relative error, in all three cases, approaches zero as x approaches 170. I should examine that more closely. A couple of tables will clarify. Looking at the results when $N = 169$ and 170 , for both Stirling and revised Stirling, we get these two tables:

N	LN(FACT(N))	Stirling’s Approximation Equ [4]	Absolute Error	Relative Error
169	701.4372638	697.9528828	3.484380987	0.004967488
170	706.5730622	703.0857343	3.487327947	0.004935552

N	LN(FACT(N))	Revised Stirling’s Approximation Equ [5]	Absolute Error	Relative Error
169	701.4372638	701.4339652	0.003298615	4.70265E-06
170	706.5730622	706.5697327	0.003329594	4.71231E-06

With the standard version of Stirling’s approximation of $\ln(N!)$ the absolute error is rising but the relative error is falling. For the revised version of Sterling’s approximation, both the absolute and relative errors are rising. This is not good.

But, curiously, if I use the two simple parameters, things are a little better.
Using 0.5 and 1.0 as the two revised parameters:

N	LN(FACT(N))	Simply Revised Stirling's Approximation Equ [6]	Absolute Error	Relative Error
169	701.4372638	701.5178	-0.080568371	-0.000114862
170	706.5730622	706.6536	-0.080571271	-0.000114031

The absolute error is larger and climbing, not good, but the relative error is larger but falling, good. In either case, when $N \approx 170$, both sorts of revised formulae for Stirling's approximation seem to perform better than the standard form. This is true, of course, because they include one more term in the asymptotic series of improvements on the standard form. (See Wikipedia article)

Here's another way to summarize it:

Option Number	Option	Absolute Error at N=170	Relative Error at N=170	RMS Error on [1,170]
1	Stirling unrevised Equ [4]	Rising	falling	3.044133
2	Revised Stirling with precise parameters Equ [5]	Rising	rising	0.003146
3	Revised Stirling with {0.5, 1} as parameters Equ [6]	Rising	falling	0.078870

The RMS error is computed as

$$E_{rms} = \sqrt{\sum_{N=0}^{170} error^2 / 171}$$

As expected, the standard version of Stirling's formula performs less well on the interval [1, 170] compared to the revised versions.

4.2.4 - Summary for ln(x!)

In summary:

- **Boltzmann over Shannon** – For reasons of exactitude, I do not want to use Shannon's formula for the definition of entropy because it distorts values calculated on sparsely-populated histograms. Therefore I MUST use Boltzmann's version of the definition of entropy with the multinomial coefficient substituted in in place of Ω .
- **Barrier at 170!** – However, Boltzmann's equation requires extensive use of the function $\ln(x!)$, which cannot be calculated exactly for any value of x greater than 170. So a work-around using Stirling's Approximation needs to be implemented for histograms with heavily-populated bins. This work-around should only be used on those bins having more than 170 agents.

- A revised version of Stirling's approximation, as shown in equation [6] seems to perform best for numbers of the order of magnitude of $[x > 170]$, where many ABMs will be functioning, and for which MS Excel and C++ cannot provide exact answers.
- For very large numbers (not shown here) Stirling's approximation is good because the relative error declines as $\ln(N)/N$ and goes to zero as N rises.

Then, I can implement this in both MS Excel when I do my analysis and also in C++ or NetLogo when I build my models.

4.3 - Examination of GammaLn(x)

When given the choice between (a) an accurate calculation using an approximate formula; or (b) an occasionally approximate calculation based on an accurate formula; I have chosen to work with the formula that gives precisely correct answers for the low range of inputs, and this requires that I use GammaLn() for the S_{\max} calculation.

GammaLn(x) is the combined formula for $\ln(\Gamma(x))$ where $\ln(x)$ is the natural logarithm function and $\Gamma(x)$ is the Gamma function.

The Gamma function is a peculiar function which in some ways is just "made-up", not being the result of normal arithmetic. The factorial function is defined for integers, but not for the rationals that fill the spaces between them. The Gamma function was, I believe, constructed to fill that gap. But, it has a VERY peculiar definition.

4.3.1 - Lanczos Approximation of $\Gamma(X+1)$

Here's an excerpt from the Ref I article:

"In mathematics, the Lanczos approximation is a method for computing the gamma function numerically, published by Cornelius Lanczos in 1964. It is a practical alternative to the more popular Stirling's approximation for calculating the gamma function with fixed precision. The Lanczos approximation consists of the formula

$$\Gamma(z + 1) = \sqrt{2\pi} \left(z + g + \frac{1}{2}\right)^{z + \frac{1}{2}} e^{-\left(z + g + \frac{1}{2}\right)} A_g(z)$$

z is a complex number from the Argand plane. This function merely approximates the Gamma function. What's more, it is an incomplete expression. The term at the end needs explanation:

$$A_g(z) = \frac{1}{2}p_0(g) + p_1(g)\frac{z}{z+1} + p_2(g)\frac{z(z-1)}{(z+1)(z+2)} + \dots$$

And, so it goes on, explaining the role of g , the role of the p_i coefficients, etc. And, since there are a lot of exponents in this approximate definition of the Gamma function, taking the logarithm

of $\Gamma(x+1)$ is a relatively easy next step. I can actually follow the mathematical arguments made in these articles, but it is pretty dense stuff.

But, the implementation on a computer is doable, because there are certain values of g and π that make it converge quickly, and those parameters can simply be stored at great precision in an array, and called upon when needed. The routines proposed in the Ref K book tend to be of that sort.

4.3.2 - Spouge's Approximation of $\Gamma(x+1)$

Similarly, at Ref J a new approach published in 1994 is similar to Lanczos' approximation and is derived from the same source (Stirling's approximation of $x!$). It does not converge as quickly, but the coefficients are easier to calculate. Here's an excerpt from the Ref J article:

"In mathematics, the Spouge's approximation is a formula for computing an approximation of the gamma function. It was named after John L. Spouge who defined the formula in a 1994 paper.[1] The formula is a modification of Stirling's approximation, and has the form

$$\Gamma(z + 1) = (z + a)^{z + \frac{1}{2}} e^{-z-a} \left(c_0 + \sum_{k=1}^{a-1} \frac{c_k}{z + k} + \varepsilon_a(z) \right)$$

where a is an arbitrary positive integer and the coefficients are given by

$$c_0 = \sqrt{2\pi}$$

$$c_k = \frac{(-1)^{k-1}}{(k-1)!} (-k+a)^{k-\frac{1}{2}} e^{-k+a} \quad k \in \{1, 2, \dots, a-1\}.$$

Again, the mathematics is dense.

4.3.3 - Numerical Recipe for GammaLn(X+1)

The Ref K book makes a good argument. It is better to implement the logarithm of the Gamma function because it is less likely to overflow. Then, if you want $\Gamma(x)$ you simply calculate it as $e^{\text{GammaLn}(x)}$. On page 207 of the Ref J book is this version of $\text{GammaLn}()$:

```
FUNCTION gammln(xx)
REAL gammln, xx
    Returns the value ln[Γ(xx)] for xx > 0.
INTEGER j
DOUBLE PRECISION ser, stp, tmp, x, y, cof(6)
    Internal arithmetic will be done in double precision, a nicety that you
    can omit if five-figure accuracy is good enough.
SAVE cof, stp
DATA cof, stp/76.18009172947146d0, -86.50532032941677d0,
24.01409824083091d0, -1.231739572450155d0, .1208650973866179d-2,
-.5395239384953d-5, 2.5066282746310005d0/
x=xx
y=x
tmp=x+5.5d0
tmp=(x+0.5d0)*log(tmp)-tmp
ser=1.000000000190015d0
do j=1,6
```

```

        y=y+1.d0
        ser=ser+cof(j)/y
    enddo
    gammln=tmp+log(stp*ser/x)
    return
END

```

This works in FORTRAN and can easily be translated to C++ or NetLogo or whatever computer platform is being used. It is an implementation of Lanczos' approximation.

4.3.4 - Test of GammaLn(x) for S_{\max}

Ref L is an MS Excel spreadsheet in which I did various tests of calculations of entropy s , of maximum entropy S_{\max} , and of the entropic index I_S . I used custom functions for this test, as follows:

```

' Module containing custom functions for use with entropic index calculations.
'
'////////////////////////////////////
'// Orrery Software; Garvin H Boyle; orrery@rogers.com
'////////////////////////////////////
'
'////////////////////////////////////
'// This function computes ln(x!) (using the Stirling approximation when needed).
'// Stirling's approximation is: ln(x!)~((x+0.5)*ln(x))+(1-x)
'// MS Excel is not able to calculate x! if x > 170.
'// So, when it can, this function does the exact calculation.
'// And when it cannot do that, it uses the Stirling approximation
'////////////////////////////////////
Function LawnOfFactorial(x)
    If x < 170 Then
        LawnOfFactorial = Application.Ln(Application.Fact(x))
    Else
        LawnOfFactorial = ((x + 0.5) * Application.Ln(x)) + (1 - x)
    End If
End Function

'////////////////////////////////////
'// This function computes an entropic index for a five-bin histogram. i.e. K=5
'////////////////////////////////////
' It uses 'LawnOfFactorial' for all factorials.
' It should be wrong when alpha is non-integral.

Function FiveBinEntropicIndex01(a1, a2, a3, a4, a5)
    K = 5
    A = a1 + a2 + a3 + a4 + a5
    S = LawnOfFactorial(A)
    S = S - LawnOfFactorial(a1)
    S = S - LawnOfFactorial(a2)
    S = S - LawnOfFactorial(a3)
    S = S - LawnOfFactorial(a4)
    S = S - LawnOfFactorial(a5)
    Alpha = A / K
    Smax = LawnOfFactorial(A) - K * LawnOfFactorial(Alpha)
    FiveBinEntropicIndex01 = S / Smax
End Function

'////////////////////////////////////
'// This function computes an entropic index for a five-bin histogram. i.e. K=5
'////////////////////////////////////
' It uses 'LawnOfFactorial' for all factorials except alpha!.
' It should be correct.

Function FiveBinEntropicIndex02(a1, a2, a3, a4, a5)
    K = 5
    A = a1 + a2 + a3 + a4 + a5

```

```

    S = LawnOfFactorial(A)
    S = S - LawnOfFactorial(a1)
    S = S - LawnOfFactorial(a2)
    S = S - LawnOfFactorial(a3)
    S = S - LawnOfFactorial(a4)
    S = S - LawnOfFactorial(a5)
    Alpha = A / K
    Smax = LawnOfFactorial(A) - K * Application.GammaLn(Alpha + 1)
    FiveBinEntropicIndex02 = S / Smax
End Function

'////////////////////
'// This function computes the entropy for a five-bin histogram. i.e. K=5
'////////////////////
' It uses 'GammaLn()' for all factorials.
' It should be correct.

Function FiveBinEntropy03(a1, a2, a3, a4, a5)
    K = 5
    A = a1 + a2 + a3 + a4 + a5
    S = Application.GammaLn(A + 1)
    S = S - Application.GammaLn(a1 + 1)
    S = S - Application.GammaLn(a2 + 1)
    S = S - Application.GammaLn(a3 + 1)
    S = S - Application.GammaLn(a4 + 1)
    S = S - Application.GammaLn(a5 + 1)
    FiveBinEntropy03 = S
End Function

'////////////////////
'// This function computes the maximum entropy a five-bin histogram. i.e. K=5
'////////////////////
' It uses 'GammaLn()' for all factorials.
' It should be correct.

Function FiveBinEntropyMax03(a1, a2, a3, a4, a5)
    K = 5
    A = a1 + a2 + a3 + a4 + a5
    Alpha = A / K
    Smax = Application.GammaLn(A + 1) - K * Application.GammaLn(Alpha + 1)
    FiveBinEntropyMax03 = Smax
End Function

'////////////////////
'// This function computes an entropic index for a five-bin histogram. i.e. K=5
'////////////////////
' It uses 'GammaLn()' for all factorials.
' It should be correct.

Function FiveBinEntropicIndex03(a1, a2, a3, a4, a5)
    K = 5
    A = a1 + a2 + a3 + a4 + a5
    S = Application.GammaLn(A + 1)
    S = S - Application.GammaLn(a1 + 1)
    S = S - Application.GammaLn(a2 + 1)
    S = S - Application.GammaLn(a3 + 1)
    S = S - Application.GammaLn(a4 + 1)
    S = S - Application.GammaLn(a5 + 1)
    Alpha = A / K
    Smax = Application.GammaLn(A + 1) - K * Application.GammaLn(Alpha + 1)
    FiveBinEntropicIndex03 = S / Smax
End Function

```

These custom functions depended upon the built-in MS Excel function GammaLn().

In the spread sheet I set up a five-bin histogram, and calculated the entropy, maximum entropy, and entropic index using different functions, as shown above.

The results, in short:

- $\text{GammaLn}(x+1)$ is equal to $\text{Ln}(x!)$ for all integer values of $x \in [0, 170]$.
- $\text{GammaLn}(x+1)$ can handle all values, integer or otherwise, for $x \in [0, 10^{307}]$.

5 - Conclusion

$\text{GammaLn}()$ is the preferred function to use in all calculations of entropy associated with agent-based models.
