

# Numerical Evaluations of the Stochastic Power Law Logistic Model

Ingemar Nåsell

Numerical evaluations are made at several points in the manuscript "Approximations of Cumulants of the Stochastic Power Law Logistic Model".

This Maple worksheet contains the procedures that are used for such evaluations. All of them are based on numerical evaluations of the QSD of the model that is studied.

We give therefore first several Maple procedures that are used to determine the QSD numerically.

The worksheet contains 6 sections, as follows:

- 1) Procedures for determining the QSD numerically.
- 2) Numerical evaluations of  $q_1$ , with  $R_0=2$ ,  $\alpha=1$ ,  $N=100, 200, 400$ ,  $s=1, 2, 3, 4$ . Used in Table 1...
- 3) Numerical evaluations of the first 7 cumulants, with integer  $s$ -values from 1 to 4. Used in Table 2 of the manuscript.
- 4) Numerical evaluations of error terms of approximations of the first 3 cumulants for various  $s$ -values. The results derived for the  $s$ -values 0.5 and 3.5 are used in Table 4 of the manuscript.
- 5) Numerical evaluations of error terms of approximations of the first 3 cumulants of the QSD derived with alternate approaches. Used in Table 5 of the manuscript.
- 6) Numerical evaluations of the QSD for  $N=100$ ,  $R_0=5$ ,  $\alpha=1$ , and the 5  $s$ -values 0.2, 0.5, 1, 3, 10. Plots of these QSDs are shown in Figure 1.

## 1) Procedures for determining the QSD numerically.

The numerical determination of the QSD uses the restart map PSI studied by Ferrari et al (1995). It requires knowledge about the stationary distributions  $p_0$  and  $p_1$  of the 2 auxiliary processes  $X_0$  and  $X_1$ .

```
> restart;
```

First we give the transition rates  $\lambda_n$  and  $\mu_n$ :

```
> lambda:=n->mu*R0*(1-(n/N)^s)*n:
```

```
mu:=n->mu*(1+alpha*(n/N)^s)*n:
```

Next we give a procedure that determines the stationary distribution  $p_0$  of the auxiliary process  $X_0$ :

```
> p0PLaw:=proc(N,R0,alpha,s)
  local pi,p0,k0,k,sumpi:
  description "The stationary distribution p0 of the Power Law
  Logistic Process":
  pi:=Array(1..N,1.);
  if R0<1 then
```

```

    k0:=1;
else
    k0:=1+floor(N*((R0-1)/(R0+alpha))^(1/s));
fi;
for k from k0+1 to N do
    pi[k]:=pi[k-1]*lambda(k-1)/mu(k);
od;
for k from k0-1 by -1 to 1 do
    pi[k]:=pi[k+1]*mu(k+1)/lambda(k);
od;
sumpi:=add(pi[k],k=1..N);
p0:=pi/sumpi;
end proc:

```

Next is a procedure that determines the stationary distribution  $p_1$  of the auxiliary process  $X_1$ :

```

> p1PLaw:=proc(N,R0,alpha,s)
    local rho,p1,k0,k,sumrho;
    description "The stationary distribution p1 of the Power Law
Logistic Process":
    rho:=Array(1..N,1.);
    if R0<1 then
        k0:=1;
    else
        k0:=1+floor(N*((R0-1)/(R0+alpha))^(1/s));
    fi;
    for k from k0+1 to N do
        rho[k]:=rho[k-1]*lambda(k-1)/mu(k-1);
    od;
    for k from k0-1 by -1 to 1 do
        rho[k]:=rho[k+1]*mu(k)/lambda(k);
    od;
    sumrho:=add(rho[k],k=1..N);
    p1:=rho/sumrho;
end proc:

```

A procedure that determines the sum of  $p[j]$  for  $j$  from 1 to  $k$ :

```

> cdf:=proc(p)
    local F,N,k;
    description "The sum of p[j] from 1 to k";
    N:=op(2,op(2,(p)));
    F:=Array(1..N,1.);
    F[1]:=p[1];
    for k from 2 to N do
        F[k]:=F[k-1]+p[k];
    od;
    F;
end proc:

```

A procedure that determines the sum  $S_n$  in (3.4) in the manuscript:

```

> SnPLaw:=proc(N,R0,alpha,s,p)

```

```

local p1,p11,pcum,Sn,k;
description "The sum Sn in (3.4)";
p1:=p1PLaw(N,R0,alpha,s);
p11:=p1[1];
pcum:=cdf(p);
Sn:=Array(1..N,1.);
for k from 2 to N do
  Sn[k]:=p11*(1-pcum[k-1])/p1[k];
od;
Sn;
end proc:

```

A procedure that determines the image under the restart mapping PSI of a probability vector p:

```

> PSIPLaw:=proc(N,R0,alpha,s,p)
  local p0,pi,pin,Sn,S,pp,ppsum;
  description "The image under PSI of p";
  p0:=p0PLaw(N,R0,alpha,s);
  pi:=p0/p0[1];
  Sn:=SnPLaw(N,R0,alpha,s,p);
  S:=cdf(Sn);
  pp:=pi*S;
  ppsum:=add(pp[k],k=1..N);
  pp:=pp/ppsum;
end proc:

```

A procedure that determines the quasi-stationary distribution qsd for the stochastic power law logistic model:

```

> qsPLaw:=proc(N,R0,alpha,s)
  local qs1,kvot,qs2,kvotmax,kvotmin,kv;
  description "The quasi-stationary distribution";
  if R0>1 then
    qs1:=p0PLaw(N,R0,alpha,s);
  else
    qs1:=p1PLaw(N,R0,alpha,s);
  fi;
  kv:=2.;
  while kv>1+1e-8 do
    qs2:=PSIPLaw(N,R0,alpha,s,qs1);
    kvot:=qs1/qs2;
    kvotmax:=max(kvot);
    kvotmin:=min(kvot);
    kv:=max(kvotmax,1/kvotmin);
    qs1:=qs2;
  od;
  qs2;
end proc:

```

## 2) Numerical evaluations of $q_1$ , with $R_0=2$ , $\alpha=1$ , $N=100, 200, 400$ , and the $s$ -values $1,2,3,4$ .

The results are found in Table 1 of the manuscript.

```
> Digits:=10: R0:=2: alpha:=1.0:
s:=1: N:=100: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=1: N:=200: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=1: N:=400: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=2: N:=100: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=2: N:=200: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=2: N:=400: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=3: N:=100: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=3: N:=200: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=3: N:=400: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=4: N:=100: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=4: N:=200: qs:=qsPLaw(N,R0,alpha,s): qs[1];
s:=4: N:=400: qs:=qsPLaw(N,R0,alpha,s): qs[1];
```

0.00001301458858

1.490837763  $10^{-10}$

1.271061069  $10^{-20}$

6.592379802  $10^{-12}$

1.815689233  $10^{-23}$

9.567061498  $10^{-47}$

7.737105036  $10^{-16}$

1.849263118  $10^{-31}$

7.371468626  $10^{-63}$

2.183546548  $10^{-18}$

1.216948232  $10^{-36}$

2.639475267  $10^{-73}$

(1)

## 3) Numerical evaluations of the first 7 cumulants, with integer $s$ -values from 1 to 4.

A procedure for determining the first 7 cumulants of the QSD of the power law logistic model follows. Note that  $\mu_1$  is a raw moment (about zero), but all other ( $\mu_2, \mu_3, \dots, \mu_7$ ) are central moments (about  $\mu_1$ )! Expressions for the cumulants  $\kappa_1, \kappa_2, \dots, \kappa_7$  in terms of central moments are found in a separate Maple worksheet, which is included among the worksheets that are used to determine asymptotic approximations of the first 3 cumulants of the QSD. The first procedure below determines numerical values of the first 7 cumulants for given values of  $N, R_0, \alpha, s$ :

```
> PLawcum7:=proc(N,R0,alpha,s)
```

```

local qs,mu1,muc2,muc3,muc4,muc5,muc6,muc7,kappa1,kappa2,
kappa3,kappa4,kappa5,kappa6,kappa7;
qs:=qsPLaw(N,R0,alpha,s);
mu1:=add(k*qs[k],k=1..N);
muc2:=add((k-mu1)^2*qs[k],k=1..N);
muc3:=add((k-mu1)^3*qs[k],k=1..N);
muc4:=add((k-mu1)^4*qs[k],k=1..N);
muc5:=add((k-mu1)^5*qs[k],k=1..N);
muc6:=add((k-mu1)^6*qs[k],k=1..N);
muc7:=add((k-mu1)^7*qs[k],k=1..N);
kappa1:=mu1;
kappa2:=muc2;
kappa3:=muc3;
kappa4:=muc4-3*muc2^2;
kappa5:=muc5-10*muc2*muc3;
kappa6:=muc6-15*muc2*muc4-10*muc3^2+30*muc2^3;
kappa7:=muc7-21*muc5*muc2-35*muc4*muc3+210*muc3*muc2^2;
[kappa1,kappa2,kappa3,kappa4,kappa5,kappa6,kappa7];
end proc:

```

Determine numerically the first 7 cumulants of the QSD of the Power Law Logistic model with  $R_0=10$ ,  $\alpha=1$ ,  $s=1$ , and the 3 N-values 100, 200, 400:

```

> Digits:=10:
N:=100: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=400: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [81.61307411, 16.73947333, -13.75005084, 8.6066403, -0.532688,
-10.0089, 17.7558]
cum7 := [163.4328171, 33.26613786, -27.26965240, 16.930422, -0.625105,
-20.975, 39.389]
cum7 := [327.0699482, 66.32288739, -54.31503815, 33.59169, -0.84383,
-42.802, 84.37]

```

(2)

We need a higher value of Digits for determining the last one of the cumulants with sufficient precision. So we rerun the same evaluation with a higher value of Digits! (We have checked that Digits=15 is sufficiently high). We note that kappa5 does not seem to follow the pattern of being approximately doubled when N is doubled. The reason is that we need a larger value of N to discover this. We show this in the following run:

```

> Digits:=15:
N:=100: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=400: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);

```

```

N:=800: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=1600: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=3200: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [81.6130740764851, 16.7394733167713, -13.7500492175362,
8.606639679193, -0.53282480434, -10.008578529, 17.772078065]
cum7 := [163.432817381411, 33.2661379145239, -27.2696803053587,
16.93044611312, -0.62047565250, -20.98940100, 38.33269573]
cum7 := [327.069948363481, 66.3228873761831, -54.3150707521796,
33.5917285884, -0.8327803226, -42.83322581, 79.0227500]
cum7 := [654.343056056411, 132.438045866994, -108.408783080007,
66.9207167748, -1.274419110, -86.4679106, 160.212098]
cum7 := [1308.88870001039, 264.669180522984, -216.597642081610,
133.581808972, -2.165888825, -173.712087, 322.51974]
cum7 := [2617.97970360663, 529.131855707806, -432.976069696797,
266.905528224, -3.95271644, -348.18852, 646.6937]

```

**(3)**

We turn now to production runs for Table 2. It contains numerical values of the first 7 cumulants of the QSD of the Power Law Logistic Model with  $R_0=10$ ,  $\alpha=1$ , the 3 N-values 100, 200, and 400, and the 2 s-values 1 and 4. We show here also runs with the s-values 2 and 3, but the results of these runs are not included in Table 2. Begin with s=1:

```

> Digits:=15:
N:=100: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
N:=400: R0:=10.: alpha:=1.0: s:=1:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [81.6130740764851, 16.7394733167713, -13.7500492175362,
8.606639679193, -0.53282480434, -10.008578529, 17.772078065]
cum7 := [163.432817381411, 33.2661379145239, -27.2696803053587,
16.93044611312, -0.62047565250, -20.98940100, 38.33269573]
cum7 := [327.069948363481, 66.3228873761831, -54.3150707521796,
33.5917285884, -0.8327803226, -42.83322581, 79.0227500]

```

**(4)**

Determine numerically the first 7 cumulants of the QSD of the Power Law Logistic model with  $R_0=10$ ,  $\alpha=1$ ,  $s=2$ , and the 3 N-values 100, 200, 400:

```

> Digits:=15:
N:=100: R0:=10.: alpha:=1.0: s:=2:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=2:
cum7:=PLawcum7(N,R0,alpha,s);

```

```

N:=400: R0:=10.: alpha:=1.0: s:=2:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [90.2996726280174, 9.29405722110359, -8.56642041931104,
7.351317521046, -5.590154555625, 3.9006785189, -5.471555216]
cum7 := [180.754193556791, 18.4293609698093, -16.9626968295494,
14.49839018662, -10.87103349968, 7.157066676, -9.17956285]
cum7 := [361.661551718958, 36.7020866549278, -33.7584437861473,
28.79861015203, -21.4473185746, 13.71180550, -16.73638465]

```

(5)

Determine numerically the first 7 cumulants of the QSD of the Power Law Logistic model with  $R_0=10$ ,  $\alpha=1$ ,  $s=3$ , and the 3 N-values 100, 200, 400:

```

> Digits:=15:
N:=100: R0:=10.: alpha:=1.0: s:=3:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=3:
cum7:=PLawcum7(N,R0,alpha,s);
N:=400: R0:=10.: alpha:=1.0: s:=3:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [93.3930002413738, 6.43820278192966, -6.15056982128722,
5.720042658651, -5.276416483196, 5.59272325943, -9.7310075645]
cum7 := [186.923847865025, 12.7352416500241, -12.1502140083217,
11.259783532303, -10.28189378516, 10.6258018490, -17.942042115]
cum7 := [373.983944467261, 25.3312243226241, -24.1521745978082,
22.34400538271, -20.30364952629, 20.722235006, -34.46332100]

```

(6)

Determine numerically the first 7 cumulants of the QSD of the Power Law Logistic model with  $R_0=10$ ,  $\alpha=1$ ,  $s=4$ , and the 3 N-values 100, 200, 400:

```

> Digits:=15:
N:=100: R0:=10.: alpha:=1.0: s:=4:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=4:
cum7:=PLawcum7(N,R0,alpha,s);
N:=400: R0:=10.: alpha:=1.0: s:=4:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [94.9785825953827, 4.93467792944513, -4.79772041143912,
4.6312008583882, -4.614460579904, 5.52014629452, -10.1791384472]
cum7 := [190.086659674167, 9.73680851168043, -9.45386110871316,
9.093555145240, -8.975821945220, 10.5188142402, -18.945165040]
cum7 := [380.301183075857, 19.3429585217139, -18.7686886709968,
18.02260446303, -17.70818123883, 20.542797166, -36.56368250]

```

(7)

The results indicate that the first 7 cumulants are all of order N for the s-values 1,2, 3,4, since each cumulant is approximately doubled when N is.

Run this determination of numerical values of the first 7 cumulants, with  $s=10$ :

```

> Digits:=15:

```

```

N:=100: R0:=10.: alpha:=1.0: s:=10:
cum7:=PLawcum7(N,R0,alpha,s);
N:=200: R0:=10.: alpha:=1.0: s:=10:
cum7:=PLawcum7(N,R0,alpha,s);
N:=400: R0:=10.: alpha:=1.0: s:=10:
cum7:=PLawcum7(N,R0,alpha,s);
cum7 := [97.8992059881713, 2.09668703471766, -2.10430059788534,
2.1675088044329, -2.4411924813077, 3.450505750788,
-6.97132562517]
cum7 := [195.913998804234, 4.07510655788475, -4.08231634180123,
4.1843613171735, -4.656115545192, 6.43341382288, -12.6729571102]
cum7 := [391.941326856636, 8.03442713348545, -8.04144207304280,
8.222928312978, -9.096218946887, 12.4270188548, -24.165243267]

```

(8)

Again, each cumulant is approximately doubled when N is. These results indicate that the first 7 cumulants are all of order N also for s=10:

#### 4) Numerical evaluations of error terms of approximations of the first 3 cumulants for various values of s.

We use numerical evaluations to confirm our results concerning the magnitudes of the approximation errors for the first 3 cumulants.

The approximations are known to be asymptotic when s takes any of the integer values from 1 to 10.

We claim that they are asymptotic also for any positive integer value of s, and also for any positive non-integer value of s.

The numerical values of the first 3 cumulants are determined by the following procedure:

```

> PLawcum3:=proc(N,R0,alpha,s)
  local qs,mu1,muc2,muc3,kappa1,kappa2,kappa3;
  qs:=qsPLaw(N,R0,alpha,s):
  mu1:=add(k*qs[k],k=1..N):
  muc2:=add((k-mu1)^2*qs[k],k=1..N):
  muc3:=add((k-mu1)^3*qs[k],k=1..N):
  kappa1:=mu1:
  kappa2:=muc2:
  kappa3:=muc3:
  [kappa1,kappa2,kappa3];
end proc:

```

The next procedure is used to determine approximations of the first 3 cumulants. The approximations are  $\text{kap1app}=x1*N + x2 + x3/N$ ,  $\text{kap2app}=y1*N + y2$ ,  $\text{kap3app}=z1*N$ .

```

> PLawcum3app:=proc(N,R0,alpha,s)
  local h1,h2,A,B,x1,x2,x3,y1,y2,z1,kap1app,kap2app,kap3app:
  h1:=s->(s^2+6*s+5)/s/12;
  h2:=s->(s^2+12*s+11)/s^2/24;

```



```

x1:=((R0-1)/(R0+alpha))^(1/s):
A:=(alpha+1)*R0/(R0+alpha)/(R0-1):
B:=A/(R0+alpha)/(R0-1);
x2:=-A*(s+1)/(2*s);
x3:=-B*(h1(s)*(R0^2+alpha)+h2(s)*(alpha+1)*R0)/x1;
y1:=A*x1/s;
y2:=B*(R0^2+alpha)*(s+1)/(2*s);
z1:=-B*x1*((R0^2+alpha)/s-(alpha+1)*R0/s^2);
kap1app:=x1*N+x2+x3/N;
kap2app:=y1*N+y2;
kap3app:=z1*N;
[kap1app,kap2app,kap3app];
end proc:

```

Determine cumulants, approximations, and approximation errors for  $R_0=10$ ,  $\alpha=1$ ,  $s=1,2,3,4$ , and  $N=100, 200, 400$ . These results are not included in the manuscript. Start with  $s=1$ : Note that Digits remains at 15:

```
> Digits;
```

15

(9)

Determine approximation errors for the 3 N-values 100, 200, 400:

```
> N:=100: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=200: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=400: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
apperror := [-0.0000697070290, 0.0044462787140, -0.2263828013077]
```

```
apperror := [-0.000017136609, 0.0021852566319, -0.2223474729017]
```

```
apperror := [-4.249064 10-6, 0.0010834786217, -0.2204050872656]
```

(10)

The errors are  $O(1/N^2)$ ,  $O(1/N)$ ,  $O(1)$ , respectively, for  $\kappa_1$ ,  $\kappa_2$ ,  $\kappa_3$ . This is supported by the numerical results, since the error is divided by approximately 4, resp 2, resp 1, for each doubling of  $N$ .

Next take  $s=2$  and determine approximation errors for the 3 N-values 100, 200, 400:

```
> N:=100: R0:=10.: alpha:=1.0: s:=2:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=200: R0:=10.: alpha:=1.0: s:=2:
```

```
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=400: R0:=10.: alpha:=1.0: s:=2:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
apperror := [-0.0000385758577, 0.00277374598922, -0.16803279868768]
```

```
apperror := [-9.529383 10-6, 0.0013700832475, -0.1659215883027]
```

```
apperror := [-2.368358 10-6, 0.0006809454712, -0.1648933036539] (11)
```

As above, the errors of the approximations of kappa1, kappa2, kappa3 are divided by approximately 4, 2, 1, respectively, for each doubling of N.

Next take s=3 and determine approximation errors for the 3 N-values 100, 200, 400:

```
> N:=100: R0:=10.: alpha:=1.0: s:=3:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=200: R0:=10.: alpha:=1.0: s:=3:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=400: R0:=10.: alpha:=1.0: s:=3:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
apperror := [-0.0000339117347, 0.00249973782424, -0.14915738387811]
```

```
apperror := [-8.391032 10-6, 0.0012365072950, -0.1473891335035]
```

```
apperror := [-2.087112 10-6, 0.0006149826476, -0.1465248481718] (12)
```

As above, the errors of the approximations of kappa1, kappa2, kappa3 are divided by approximately 4, 2, 1, respectively, for each doubling of N.

Next take s=4 and determine the approximation errors for the 3 N-values 100, 200, 400:

```
> N:=100: R0:=10.: alpha:=1.0: s:=4:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=200: R0:=10.: alpha:=1.0: s:=4:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
```

```
N:=400: R0:=10.: alpha:=1.0: s:=4:
```

```

cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
apperror := [-0.0000335144003, 0.00248099983867, -0.13989394528898]
apperror := [-8.301147 10-6, 0.00122803885665, -0.13820817641288]
apperror := [-2.065778 10-6, 0.0006109624555, -0.1373828063962] (13)

```

As above, the errors of the approximations of the first 3 cumulants kappa1, kappa2, kappa3 are divided by approximately 4, 2, 1, respectively, for each doubling of N. This supports our claim that they are asymptotic in N.

Next, take s=10 and determine the approximation errors for the 3 N-values 100, 200, 400:

```

> N:=100: R0:=10.: alpha:=1.0: s:=10:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;

N:=200: R0:=10.: alpha:=1.0: s:=10:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;

N:=400: R0:=10.: alpha:=1.0: s:=10:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
apperror := [-0.0000448890107, 0.00326472129902, -0.12423406448915]
apperror := [-0.000011183468, 0.00161771106992, -0.12218327500885]
apperror := [-2.790993 10-6, 0.00080521987824, -0.12117593945804] (14)

```

As above, the errors of the approximations of the first 3 cumulants kappa1, kappa2, kappa3 are divided by approximately 4, 2, 1, respectively, for each doubling of N. This supports our claim that they are asymptotic in N.

The results concerning approximation errors derived so far in the present section are not included in the manuscript.

Finally we consider two non-integer positive values of s. First take s=0.5:

```

> N:=100: R0:=10.: alpha:=1.0: s:=0.5:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;

N:=200: R0:=10.: alpha:=1.0: s:=0.5:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;

N:=400: R0:=10.: alpha:=1.0: s:=0.5:
cum3:=PLawcum3(N,R0,alpha,s):

```

```

cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
apperror := [-0.0002273093088, 0.0111221963874, -0.3525718649053]
apperror := [-0.000055221832, 0.0053951087832, -0.3396067934278]
apperror := [-0.000013615783, 0.002658455787, -0.3335562155237]

```

(15)

We find again that the approximation errors of the 3 cumulants kappa1, kappa2, kappa3 are divided by approximately 4, 2, 1, respectively, for each doubling of N. This supports our claim that they are asymptotic in N.

Next take s=3.5:

```

> N:=100: R0:=10.: alpha:=1.0: s:=3.5:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;

N:=200: R0:=10.: alpha:=1.0: s:=3.5:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;

N:=400: R0:=10.: alpha:=1.0: s:=3.5:
cum3:=PLawcum3(N,R0,alpha,s):
cum3app:=PLawcum3app(N,R0,alpha,s):
apperror:=cum3-cum3app;
apperror := [-0.0000334465548, 0.00247379684285, -0.14384339417955]
apperror := [-8.280441 10-6, 0.0012241390707, -0.1421327497957]
apperror := [-2.060151 10-6, 0.0006089424338, -0.1412958058358]

```

(16)

We find again that the approximation errors of the 3 cumulants kappa1, kappa2, kappa3 are divided by approximately 4, 2, 1, respectively, for each doubling of N. This supports our claim that they are asymptotic in N.

The results that we have presented here for the s-values 0.5 and 3.5 are included in Table 4 in the manuscript.

## 5) Numerical evaluations of error terms of approximations of the first 3 cumulants of the QSD for 3 different methods, with s=1.

The 3 methods that are discussed in this section are as follows:

- 1) The method presented in the manuscript, based on asymptotic approximations, denoted by N.
- 2) The first of the two approaches initiated by Renshaw, based on the BGL method, denoted by BR1.
- 3) The method discussed in Section 4 of the paper by Bhowmick et al. (2016), based on the BGL method, denoted BB.

We use the parameter values R0=10, alpha=1, s=1, and the 3 N-values 100, 200, and 400, and evaluate numerical values of the error terms for the 3 cases listed above.

The results for the first case have already been treated in the previous section.

For convenience we repeat them here:

```
> N:=100: R0:=10.: alpha:=1.0: s:=1:
  cum3:=PLawcum3(N,R0,alpha,s):
  cum3appN:=PLawcum3app(N,R0,alpha,s):
  errorN:=cum3-cum3appN;

N:=200: R0:=10.: alpha:=1.0: s:=1:
  cum3:=PLawcum3(N,R0,alpha,s):
  cum3appN:=PLawcum3app(N,R0,alpha,s):
  errorN:=cum3-cum3appN;

N:=400: R0:=10.: alpha:=1.0: s:=1:
  cum3:=PLawcum3(N,R0,alpha,s):
  cum3appN:=PLawcum3app(N,R0,alpha,s):
  errorN:=cum3-cum3appN;
  errorN:= [-0.0000697070290, 0.0044462787140, -0.2263828013077]
  errorN:= [-0.000017136609, 0.0021852566319, -0.2223474729017]
  errorN:= [-4.249064 10-6, 0.0010834786217, -0.2204050872656] (17)
```

Approximations of the first 3 cumulants in the second case (BR1) are given by (8.11)-(8.13) in the manuscript, where the coefficients  $x_1, x_2, x_3, y_1, y_2, z_1$  are given by (8.29)-(8.34).

The resulting approximations of the first 3 cumulants are determined by the following procedure:

```
> PLawcum3appBR1:=proc(N,R0,alpha)
  local x1,x2,x3,y1,y2,z1,kap1app,kap2app,kap3app:
  x1:=(R0-1)/(R0+alpha):
  x2:=- (alpha+1)*R0/(R0+alpha)/(R0-1):
  x3:=(alpha+1)*R0*(R0^2+alpha-5*(alpha+1)*R0)/(R0+alpha)/(R0-1)
  ^3:
  y1:=(alpha+1)*R0/(R0+alpha)^2:
  y2:=- (alpha+1)*R0*(R0^2+alpha-4*(alpha+1)*R0)/(R0+alpha)^2/(R0-1)^2:
  z1:=(alpha+1)*R0*(R0^2+alpha-3*(alpha+1)*R0)/(R0+alpha)^3/(R0-1):
  kap1app:=x1*N+x2+x3/N;
  kap2app:=y1*N+y2;
  kap3app:=z1*N;
  [kap1app,kap2app,kap3app];
end proc;
```

The approximation errors are found to be as follows:

```
> N:=100: R0:=10.: alpha:=1.0: s:=1:
  cum3:=PLawcum3(N,R0,alpha,s):
  cum3appBR1:=PLawcum3appBR1(N,R0,alpha):
  errorBR1:=cum3-cum3appBR1;

N:=200: R0:=10.: alpha:=1.0: s:=1:
  cum3:=PLawcum3(N,R0,alpha,s):
```

```
cum3appBR1:=PLawcum3appBR1(N,R0,alpha):
errorBR1:=cum3-cum3appBR1;
```

```
N:=400: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3appBR1:=PLawcum3appBR1(N,R0,alpha):
errorBR1:=cum3-cum3appBR1;
errorBR1 := [-0.0031124804422, 0.2534004670621, -20.5953618479728]
errorBR1 := [-0.001538523316, 0.2511394449800, -40.9603055662319]
errorBR1 := [-0.000764942417, 0.2500376669698, -81.6963212739260]
```

(18)

Approximations of the first 3 cumulants in the third case (BB) are given by (8.51), (8.53), and (8.64) in the manuscript.

The resulting approximations of the first 3 cumulants are determined by the following procedure:

```
> PLawcum3appBB:=proc(N,R0,alpha)
  local x1,x2,x3,y1,y2,z1,kap1app,kap2app,kap3app:
  x1:=(R0-1)/(R0+alpha):
  x2:=- (alpha+1)*R0/(R0+alpha)/(R0-1):
  x3:=(alpha+1)^2*R0^2/(R0+alpha)/(R0-1)^3:
  y1:=(alpha+1)*R0/(R0+alpha)^2:
  y2:=-2*(alpha+1)^2*R0^2/(R0+alpha)^2/(R0-1)^2:
  z1:=- (alpha+1)*R0*(R0^2+alpha-4*(alpha+1)*R0)/(R0+alpha)^3/(R0
-1):
  kap1app:=x1*N+x2+x3/N;
  kap2app:=y1*N+y2;
  kap3app:=z1*N;
  [kap1app,kap2app,kap3app];
end proc;
```

The approximation errors are found to be as follows:

```
> N:=100: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3appBB:=PLawcum3appBB(N,R0,alpha):
errorBB:=cum3-cum3appBB;

N:=200: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3appBB:=PLawcum3appBB(N,R0,alpha):
errorBB:=cum3-cum3appBB;

N:=400: R0:=10.: alpha:=1.0: s:=1:
cum3:=PLawcum3(N,R0,alpha,s):
cum3appBB:=PLawcum3appBB(N,R0,alpha):
errorBB:=cum3-cum3appBB;
errorBB := [-0.0035863549901, 0.2921720209852, -10.2439134799955]
errorBB := [-0.001775460590, 0.2899109989031, -20.2574088302773]
errorBB := [-0.000883411054, 0.2888092208929, -40.2905278020168]
```

(19)

**6) Numerical evaluations and plots of the QSD for  $N=100$ ,  $R_0=5$ ,  $\alpha=1$ , and the 5 s-values 0.2, 0.5, 1, 3, and 10.**

We use the letters A, B, C, D, E to refer to the 5 s-values. Thus, A means  $s=0.2$ , B means  $s=0.5$ , C means  $s=1$ , D means  $s=3$ , and E means  $s=10$ .

```
> N:=100: R0:=5: alpha:=1:
  s:=0.2: qsA:=qsPLaw(N,R0,alpha,s):
  s:=0.5: qsB:=qsPLaw(N,R0,alpha,s):
  s:=1.0: qsC:=qsPLaw(N,R0,alpha,s):
  s:=3.0: qsD:=qsPLaw(N,R0,alpha,s):
  s:=10.0: qsE:=qsPLaw(N,R0,alpha,s):
> with(plots,display,textplot);
                                     [display, textplot] (20)
> X:=[seq(i,i=1..100)]:
  plA:=plot(X,qsA,labels=[n,q_n],labelfont=[TIMRS,ROMAN,16]):
  plB:=plot(X,qsB):
  plC:=plot(X,qsC):
  plD:=plot(X,qsD):
  plE:=plot(X,qsE):
  pltext:=textplot([[10,0.08,"s=0.2"],[43,0.07,"s=0.5"],[67,0.08,
"s=1"],[87,0.117,"s=3"],[92,0.19,"s=10"]]):
  display(plA,plB,plC,plD,plE,pltext);
```

