Computation of Stirling numbers and generalizations

S. Ilie
Dept. Mathematics,
Ryerson University
silvana@ryerson.ca

D. J. Jeffrey
Dept. Applied Mathematics
Western University · Canada
djeffrey@uwo.ca

R. M. Corless
Dept. Applied Mathematics
Western University · Canada
rcorless@uwo.ca

X. Zhang
Dept. Applied Mathematics
Western University · Canada
xzhan695@uwo.ca

***Abstract*—We consider the computation of Stirling numbers and generalizations for positive and negative arguments. We describe computational schemes for Stirling Partition and Stirling Cycle numbers, and for their generalizations to associated Stirling numbers. The schemes use recurrence relations and are more efficient than the current method used in MAPLE for cycle numbers, which is based on an algebraic expansion. We also point out that the proposed generalization of Stirling numbers due to Flajolet and Prodinger changes the evaluation of Stirling partition numbers for negative arguments. They are no longer zero, but become rational numbers.**

I. Introduction

Among the remarkable sequences of numbers with important combinatorial significance one can count the sequences of Stirling numbers. These numbers were first introduced by James Stirling in [13] to express the connection between the ordinary powers and the factorial powers. In older literature, they are called Stirling numbers of the first kind and second kind. A modern notation for these numbers follows Knuth's suggestions [10]. He proposed the notations

$$\left\{ {n \atop m} \right\}_{\geq r} \quad \text{and} \quad \left[{n \atop m} \right]_{\geq r}$$

for respectively Stirling Partition numbers (Stirling numbers of the second kind) and Stirling Cycle numbers (Stirling numbers of the first kind). These notations in the special case $r = 1$ were first suggested by Karamata in [8].

The definitions used in this paper give all numbers as non-negative, again following Knuth, in contrast to earlier definitions [1], [2]. The names reflect the combinatorial significance of the numbers, and the notations are inspired by the similar notation for the binomial coefficients.

We describe a new implementation of Stirling cycle numbers in MAPLE, which is faster than the existing implementation in the `combinat` package. The existing implementation does not use recurrence relations, but expands a polynomial. The current MAPLE functions have names `stirling2`, for Stirling partition numbers and `stirling1`, for signed Stirling cycle numbers.

Following a challenge by Knuth, several authors suggested generalizations of Stirling numbers to non-integral arguments [12], [5]. The widely accepted generalization [5] allows Stirling numbers to be extended to complex arguments. The proposal leaves the value of $\left\{ {0 \atop 0} \right\}$ undecided, and we discuss possible values here.

II. Definitions and Properties for $r = 1$

We begin with the case $r = 1$, which has been the traditional meaning given to Stirling numbers.

Definition II.1. The Stirling partition number $\left\{ {n \atop k} \right\}$ is the number of ways to partition a set of n objects into k nonempty subsets.

Definition II.2. The Stirling cycle number $\left[{n \atop k} \right]$ is the number of permutations of n objects having k cycles.

Stirling numbers satisfy the recurrence relations

$$\left\{ {n+1 \atop k} \right\} = k \left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\} , \tag{1}$$

$$\left[{n+1 \atop k} \right] = n \left[{n \atop k} \right] + \left[{n \atop k-1} \right] , \tag{2}$$

similar to the one satisfied by the binomial coefficients

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} .$$

The identities (1) and (2) hold for all integers n and k (positive or negative). The boundary conditions

$$\left\{ {k \atop k} \right\} = \left[{k \atop k} \right] = 1 \quad \text{for} \quad k > 0 ,$$

$$\left\{ {n \atop 0} \right\} = \left[{n \atop 0} \right] = \delta_{0n} \quad \text{for} \quad n > 0 ,$$

where δ_{0n} is the Kronecker delta, lead to unique solutions for all k, n integers. The Stirling cycle numbers and the Stirling partition numbers are connected by the remarkable law of duality

$$\left\{ {n \atop k} \right\} = \left[{-k \atop -n} \right] ,$$

which is valid for all k, n integers. Other special values are

$$\left\{ {n \atop 1} \right\} = 1 , \quad \text{and} \quad \left[{n \atop 1} \right] = (n-1)! \ . \tag{3}$$

For $n, k > 1$, it is possible to write Stirling partition numbers as a sum over binomial coefficients:

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} (-1)^{k-j} j^n . \tag{4}$$

This can be extended to cycle numbers by using the identity

$$(-1)^{n+k}\left[{n \atop k}\right] = \sum_{h=0}^{n-k}(-1)^h \binom{n-1+h}{n-k+h}\binom{2n-k}{n+h}\left\{{n-k+h \atop h}\right\}. \quad (5)$$

The original definitions used by Stirling are important. Using the ‘falling’ and ‘rising’ notation of Knuth, we can write:

$$\begin{aligned} z^{\underline{n}} &:= z(z-1)(z-2)\dots(z-n+1) \\ &= \sum_k \left[{n \atop k}\right](-1)^{n-k} z^k \ . \quad (6) \\ z^{\overline{n}} &:= z(z+1)(z+2)\dots(z+n-1) \\ &= \sum_k \left[{n \atop k}\right] z^k \ . \quad (7) \end{aligned}$$

III. Associated Stirling numbers

For extended discussions of associated Stirling numbers, we refer to [2], [7].

Definition III.1. The number $\left\{{n \atop m}\right\}_{\geq r}$ gives the number of partitions of a set of size n into m subsets, each subset having a cardinality $\geq r$.

Definition III.2. The number $\left[{n \atop m}\right]_{\geq r}$ gives the number of permutations of n objects into m cycles, each cycle having a cardinality $\geq r$.

In [2, pp 221-2], $\left\{{n \atop m}\right\}_{\geq r}$ is called ‘r-associated Stirling number of the second kind’. We abbreviate this nomenclature to ‘Stirling r-partition number’. Likewise, $\left[{n \atop m}\right]_{\geq r}$ is a Stirling r-cycle number (c.f. [2, p 256]). Obviously for $r = 1$ we return to simple Stirling numbers.

We have recurrence relations

$$\left\{{n+1 \atop k}\right\}_{\geq r} = k\left\{{n \atop k}\right\}_{\geq r} + \binom{n}{r-1}\left\{{n-r+1 \atop k-1}\right\}_{\geq r}, \quad (8)$$

$$\left[{n+1 \atop k}\right]_{\geq r} = n\left[{n \atop k}\right]_{\geq r} + n^{\underline{r-1}}\left[{n-r+1 \atop k-1}\right]_{\geq r} \ . \quad (9)$$

Note that $n^{\underline{0}} = 1$. The boundary cases are

$$\left\{{n \atop 1}\right\}_{\geq r} = 1 \ , \quad n \geq r \ , \quad (10)$$

$$\left[{n \atop 1}\right]_{\geq r} = (n-1)! \ , \quad n \geq r \ , \quad (11)$$

$$\left\{{kr \atop k}\right\}_{\geq r} = \frac{(rk)!}{(r!)^k\, k!} \ , \quad k \geq 1 \ , \quad (12)$$

$$\left[{kr \atop k}\right]_{\geq r} = \frac{(rk)!}{r^k k!} \ , \quad k \geq 1 \ . \quad (13)$$

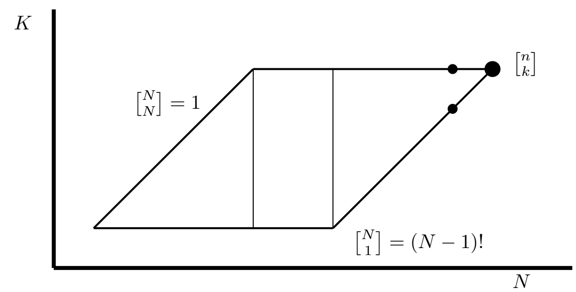

Fig. 1. Calculating Stirling cycle numbers $\left[{n \atop k}\right]$. The 3 solid circles illustrate the calculation of the recurrence relation. The larger circle is calculated from the values of the 2 smaller circles. The case $n > 2k$ is illustrated, showing the 3 regions in which numbers are evaluated.

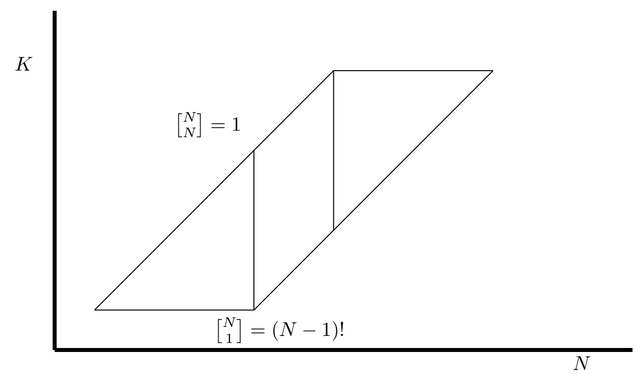

Fig. 2. Calculating Stirling cycle numbers for the case $n < 2k$. The three regions used in the calculation are shown.

There are no known analogues of (4) or (6) for associated Stirling numbers. Instead we shall utilize the generating functions

$$\left(e^z - \sum_{m=0}^{r-1}\frac{z^m}{m!}\right)^k = k!\sum_{n\geq 0}\left\{{n \atop k}\right\}_{\geq r}\frac{z^n}{n!} \ , \quad (14)$$

$$\left(\ln\frac{1}{1-z} - \sum_{m=1}^{r-1}\frac{z}{m}\right)^k = k!\sum_{n\geq 0}\left[{n \atop k}\right]_{\geq r}\frac{z^n}{n!} \ . \quad (15)$$

IV. Computing Stirling numbers ($r = 1$)

At present, the case $r = 1$ is implemented in Maple 2015, as part of the `combinat` package. Stirling partition numbers are computed by the function `stirling2` using (4), for $n, k > 0$. As well as being an explicit sum over $k+1$ terms, the formula contains the factorial function, which is implemented in the Maple kernel and is evaluated quickly. This method does not generalize to associated numbers $r > 1$, but for $r = 1$ is very fast, and is not considered further.

Stirling cycle numbers are computed by `stirling1` using (6); in Maple library code, this is written

```
stirling1:= (n,k)->
    coeff(mul(z-m,m=0..n-1),z,k);
```

If fact the implementation in Maple works a little differently from this. For any given n, the function computes all $n-1$ coefficients of the polynomial, or equivalently all $\left[{n \atop k}\right]$ for fixed n and $0 \le k \le n$, and stores them in cache memory. Thus after one Stirling number is requested, subsequent requests for another number with the same n, but differing k can be returned immediately.

We compare computation of Stirling cycle numbers computed by four methods.

1) equation (5).
2) Maple using (6).
3) equation (15).
4) equation (2).

A. Stirling cycle numbers by recurrence for $0 \le k \le n$

To calculate $\left[{n \atop k}\right]$, the program computes only those elements which are needed for its evaluation, according to the recurrence relation. Thus, in coordinates (N, K), these are the pairs inside the parallelogram delimited by the lines $N = K$, $N = K + k$, $K = 0$, and $K = k$. Notice that, when k is close to 0 or n, the number of pairs computed becomes of order n, because the area of the parallelogram is of order n. (By 'of order n' we mean that the number of pairs will be approximately linear in n, with only weak dependence on k.)

When k approaches $n/2$, more pairs are computed, and so the algorithm is slower. The worst case is when k is closest to $n/2$. Then our algorithm computes a number of pairs of order n^2, because the area of the parallelogram is of order n^2. (The number of pairs will be approximately quadratic in n with a weak dependence on $(k - n/2)$.) Our algorithm is always better than the algorithm existing in MAPLE, even in the case $k = \lfloor n/2 \rfloor$. MAPLE's algorithm is always of order n^2 because the number of arithmetic operations that are required to compute the coefficients of the polynomial (6)) (of order n^2 arithmetic operations are done in order to compute, so when k is farther from $n/2$, the difference between our algorithm and MAPLE's increases. The computation works bottom-up, starting with the known boundary values for $\left[{n \atop n}\right]$ and $\left[{n \atop 1}\right]$ and considers two cases depending on whether $n > 2k$ or not. In either case, three subcases are considered. For example, when $n > 2k$, the algorithm computes as follows:

- the pairs $[i, j]$, $1 \le i \le k$, $1 \le j \le i$,
- the pairs $[i, j]$, $k + 1 < i < n - k + 1$, $1 \le j \le k$, and
- the pairs $[i, j]$, $n - k + 2 \le i \le n$, $i + k - n \le j \le k$.

On the other hand, if $n \le 2k$, we compute as follows:

- the pairs $[i, j]$, $1 \le i \le n - k$, $1 \le j \le i$,
- the pairs $[i, j]$, $n - k + 1 < i < k$, $i - n + k \le j \le i + n - k - 1$, and
- the pairs $[i, j]$, $k + 1 \le i \le n$, $i + k - n \le j \le k$.

The two cases are illustrated in figures 1 and 2. For the transition case $n = 2k$, a third case could be added, but the efficiency gains would be negligible. Our algorithm uses only a vector of length k to store all pairs computed. We can do that because $\left[{n \atop k}\right]$ depends only on the pairs in the level below, that is, with the first component equal to $n - 1$.

B. Testing

Maple procedures based on the above approach were first checked against the library code in MAPLE 2015, taking into account that `stirling1` returns signed values of cycle numbers, whereas the present code returns unsigned values. Then timing tests were made on all 4 methods.

It is well known that the speed of execution in Maple depends on many implementation-specific considerations, in addition to the underlying efficiency of the algorithm. For example, since the Maple kernel is compiled code, but the library is interpreted code, a slower algorithm running in the Maple kernel can out-perform a theoretically faster algorithm running in library code. Also, an algorithm (or a test program) that allows Maple to utilize its `remember` facility may run faster than one that does not. Our tests endeavoured to concentrate on algorithmic differences, rather than effects specific to Maple. For any specific computational engine, a full assessment of the methods might well take into account the possibilities of taking advantage of special features.

Since Maple's implementation computes all $\left[{n \atop k}\right]$ for a given n, the running time will be independent of k. Further, since the values are stored in memory, requests to compute multiple values of k for fixed n will take no additional time. It is possible to force Maple to forget stored values, but it is simpler to fix k and consider changes in n.

We compared methods by computing

$$\left[{n \atop 100}\right] \qquad \text{for} \quad 400 < n \le 500 \ .$$

The large values of the arguments were chosen because the computer used was running an Intel i7 processor running at 2.4 GHz, and smaller arguments made timing difficult. The value of the second argument was chosen $k = 100$ so that methods (2) and (5) were each running near their worst case. Each test was run 4 times and the timings added. The comparative times are given in table I.

It is also interesting to see the profile of the different methods computing $\left[{n \atop k}\right]$ for fixed n and varying k. As mentioned above, Maple computes and remembers all values, so that case is omitted. The method based on recurrence has its worst case at $n/2$ as explained in section IV-A. Method (5) is faster for larger k while (15) is slower for large k.

C. The case $k > n > 0$

For $k > n$ we can rewrite the recurrence as

$$n\left[{n \atop k}\right] = \left[{n+1 \atop k}\right] - \left[{n \atop k-1}\right] \ .$$

Now we can start with the case $k = n + 1$ and compute

$$n\left[{n \atop n+1}\right] = \left[{n+1 \atop n+1}\right] - \left[{n \atop n}\right] \ .$$

TABLE I

TIMINGS IN SECONDS OF COMPUTATIONS OF STIRLING CYCLE NUMBERS. COLUMN HEADINGS GIVE THE NUMBER OF THE EQUATION BEING USED. EACH TIME IS THE SUM OF 4 MEASUREMENTS. FOR REASONS GIVEN IN THE TEXT, THE NUMBERS TESTED WERE $\left[{n \atop 100}\right]$. THE COMPUTER USED WAS A LENOVO YOGA RUNNING INTEL I7 AT 2.4 GHZ.

n \ Method	(6)	(2)	(5)	(15)
410	0.469	0.343	7.531	61.312
420	0.577	0.702	8.266	67.252
430	0.609	0.314	8.406	71.624
440	0.595	0.405	8.968	76.252
450	0.672	0.563	9.250	82.560
460	0.812	0.501	10.469	87.188
470	0.656	0.420	10.515	91.564
480	0.844	0.455	11.361	96.688
490	0.765	0.672	11.750	103.372
500	0.844	0.469	12.234	112.500

TABLE II

TIMINGS IN SECONDS OF COMPUTATIONS OF STIRLING CYCLE NUMBERS. THE TABLE SHOWS TIMINGS FOR CALCULATING $\left[{500 \atop k}\right]$ FOR VARIOUS k. THE DATA ARE THE SUM OF 4 RUNS. FOR REASONS EXPLAINED IN THE TEXT, MAPLE'S TIMINGS ARE NOT GIVEN.

k \ method	(2)	(5)	(15)
0	0.	9.985	0.060
50	0.047	9.360	48.876
100	0.047	7.218	54.876
150	0.063	5.360	65.248
200	0.078	4.078	58.624
250	0.078	2.781	68.188
300	0.078	2.016	65.940
350	0.063	0.952	72.124
400	0.046	0.408	58.748
450	0.032	0.092	57.816
500	0.	0.	69.124

The boundary condition for $k = n$ however makes the right side zero and therefore Stirling cycle numbers are zero for all $k > n > 0$.

The same conclusion applies to Stirling partition numbers, since

$$k\left\{{n \atop k}\right\} = \left\{{n+1 \atop k}\right\} - \left\{{n \atop k-1}\right\} .$$

Again starting from $k = n$, we see that the boundary condition implies zero for $k > n > 0$.

V. COMPUTING ASSOCIATED STIRLING NUMBERS

Associated Stirling numbers have yet to be implemented in either MAPLE or MATHEMATICA. In addition to their obvious combinatorial applicability, they have also been found to occur in other contexts, such as series expansions for the Lambert W function [4]. Implementations will therefore be useful to numerous people. The methods used by MAPLE for simple Stirling numbers do not extend to associated numbers, and so we use recurrence relations (8) and (9).

The two cases identified for $r = 1$ Stirling numbers apply to the new situation. Figure 3 shows the case $n > 2rk$. The two vertical lines defining the 3 regions are at $N = rk$ and

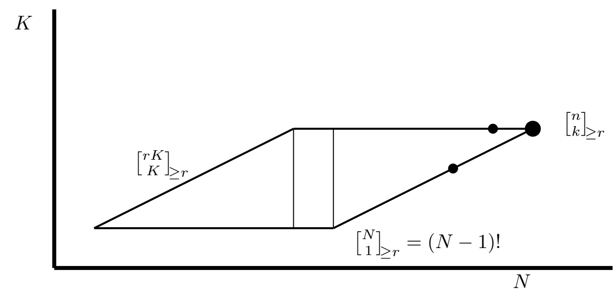

Fig. 3. Calculating Stirling cycle numbers $\left[{n \atop k}\right]_{\ge r}$. The 3 solid circles illustrate the calculation of the recurrence relation. The larger circle is calculated from the values of the 2 smaller circles. The case $n > rk$ is illustrated, showing the 3 regions in which numbers are evaluated.

$N = n - (k-1)r$. The sloping lines in the figure have slope $1/r$. A change from the $r = 1$ case is that now it is not possible to confine the intermediate storage to a vector, and instead a matrix of all computed values must be used.

A. *Testing associated Stirling numbers*

As mentioned above, neither MAPLE nor MATHEMATICA have implementations with which to compare our programs. We have programmed the recurrence relations, as described above, and the generating functions. Thus, Stirling r-partition numbers can be computed as

```
StirRPartGen:= proc(n,k,r) local s,t,z,p;
   s:= add(z^p/p! ,p=0 .. r-1) ;
   t:=series( (exp(z)- s)^k, z=0, n+1 );
   n! * coeff(t,z,n)/k! ;
   end proc;
```

and Stirling r-cycle numbers by

```
StirRCycleGen:= proc(n,k,r) local s,t,z,p;
  s:= add(z^p/p ,p=1 .. r-1) ;
  # Note: for r=1, s=0
  t:=series( (ln(1/(1-z))- s)^k,z=0,n+1);
  n! * coeff(t,z,n)/k! ;
  end proc;
```

Timings were made of evaluations of

$$\left[{500 \atop k}\right]_{\ge 2}, \quad \left[{500 \atop k}\right]_{\ge 3}, \quad \left[{500 \atop k}\right]_{\ge 4},$$

for various k. The lengths of loops used in computing the recurrence relations vary with k and r. This is reflected in the variations in the timings. The generating function method is much slower, for example $\left[{400 \atop 40}\right]_{\ge 5}$ is 829 decimal digits and takes 2.98 seconds by generating function and 0.031 seconds by recurrence relation. The generating-function method is not shown in the timings.

TABLE III

TIMINGS IN SECONDS OF COMPUTATIONS OF STIRLING r-CYCLE NUMBERS. THE TABLE SHOWS TIMINGS FOR CALCULATING $\left[{500 \atop k}\right]_{\geq r}$ FOR VARIOUS k AND r. THE DATA ARE THE SUM OF 10 RUNS.

k \ method	$r = 2$	$r = 3$	$r = 4$
10	0.158	0.109	0.157
20	0.280	0.424	0.234
30	0.607	0.358	0.345
40	0.628	0.797	0.342
50	0.735	0.469	0.610
60	0.606	0.546	0.984
70	0.799	0.844	0.422
80	0.655	0.672	0.407
90	0.736	0.591	0.623
100	1.311	0.564	0.611
110	1.016	0.768	0.247
120	0.968	0.499	0.239

VI. MAPLE-SPECIFIC CONSIDERATIONS

Maple offers the possibility of a function remembering the results of calculations. This has already been noted above in the implementation of Stirling cycle numbers in Maple's function `stirling1`. This can be done by adding the following options to a function declaration:

- `option remember`: This option allows a remember table to be created by the function. Each time the function returns a value, an entry is created in its remember table, and then if the function is called later with the same arguments, the result is retrieved from the table, rather than being recomputed. The table can grow as required and is not limited in size.

  Entries in the table can be created by other means also. If a function is assigned a value, then the value is added to the table. This is how `stirling1(n,k)` remembers all values for a given n, regardless of which value of k is requested by the user; internally all values are assigned.
- `option cache`: This option is similar to `remember`, with the important difference that a maximum size for the table is specified. Once the table reaches its specified maximum size, later values are added to the table and earlier ones are removed to make room. The default value is 512 elements. There are distinctions between temporary and permanent elements which are described in the Maple help pages.
- `option system`: This option allows the remember table to be erased any time there is a garbage collection. The default behaviour, i.e. if this option is not present in a function declaration, is for remember tables to survive garbage collection.

Therefore, in implementing Stirling numbers, a programmer can choose how many numbers to remember. A knowledgable user then has the possibility of computing the largest Stirling number first, for efficiency. Another possibility is seen in the Maple function `bernoulli`. This function can compute *more* values than actually requested. Thus a call to `bernoulli(1000)` will on a quad core computer result in Bernoulli numbers for arguments 1002, 1004, 1006 being calculated and stored, unless the option `singleton` is used. It is interesting that these ideas are seen in the existing Maple implementation of `stirling1` but not `stirling2`.

VII. STIRLING NUMBERS FOR COMPLEX ARGUMENTS

Flajolet and Prodinger [5] have proposed an extension of Stirling numbers to complex arguments. For Stirling partition numbers, they wrote

$$\left\{{x \atop y}\right\} = \frac{x!}{y!}\frac{1}{2\pi i}\int_{\mathcal{H}} (e^z-1)^y \frac{dz}{z^{x+1}} \tag{16}$$

$$= \frac{\Gamma(x)}{\Gamma(y)!}\frac{1}{2\pi i}\int_{\mathcal{H}} e^z(e^z-1)^{y-1} \frac{dz}{z^{x}} \tag{17}$$

where the second form is obtained by integration by parts. The contour $\mathcal{H}$ is a Hankel contour, which starts at $-\infty - 0i$ (i.e. below the negative real axis), circumscribes the origin without crossing the negative real axis and ends at $-\infty + 0i$. The negative real axis is never crossed because for general values of x, y that axis will be a branch cut for the integrand. For positive integers x, y, the contour collapses to a circle going anti-clockwise around the origin, and hence is the standard construction for the coefficients of a Taylor series.

It has been shown [5] that in general (17) implies two well known identities, derived initially for positive integral arguments.

$$\left\{{x \atop y}\right\} = y\left\{{x-1 \atop y}\right\} + \left\{{x-1 \atop y-1}\right\} , \tag{18}$$

$$\left\{{x \atop k}\right\} = \frac{1}{k!}\sum_{j=1}^{k}\binom{k}{j}(-1)^{k-j}j^x , \quad \text{for} \quad k \in \mathbb{N} . \tag{19}$$

Obviously (18) is the generalization of (1). It was proved in [5] for $\Re x > 1$ and then generalized by the uniqueness of analytic continuation. The identity (19) was proposed in [12] as the basis for generalizing Stirling numbers.

A. *Value at the origin*

The traditional definition of Stirling numbers specifies $\left\{{0 \atop 0}\right\} = 1$. Once the various identities are generalized to non-integer arguments, it is not possible to retain all identities everywhere – specifically at the origin. Thus if we start with (19), we have

$$\left\{{x \atop 1}\right\} = \binom{1}{1}(-1)^0(1)^x = 1 .$$

Thus this identity gives $\left\{{0 \atop 1}\right\} = 1$. Now consider (18).

$$\left\{{1 \atop 1}\right\} = \left\{{0 \atop 1}\right\} + \left\{{0 \atop 0}\right\}$$

Since the left side is 1, we cannot have both $\left\{{0 \atop 1}\right\}$ and $\left\{{0 \atop 0}\right\} = 1$.

The same contradiction is obtained from the identities

$$\left\{{n \atop n}\right\} = 1 , \tag{20}$$

$$\left\{{n \atop 0}\right\} = 0 . \tag{21}$$

If these are to hold true when n ceases to be integral, then substituting $n = 0$ in these equations gives a contradiction. One of them must take precedence for the determination of $\left\{ {0 \atop 0} \right\}$.

The integral definition (17) shows the same behaviour. It has been shown that [14]

$$\lim_{x \to 0} \left\{ {x/n \atop x} \right\} = n \ .$$

Thus the origin is a singular point and the value there is a matter of convention.

B. Consequences of different assumptions

We consider here the consequences of retaining the recurrence relation (18), and allowing different conventions for $\left\{ {0 \atop 0} \right\}$. Under the definition $\left\{ {0 \atop 0} \right\} = 1$, we have seen $\left\{ {0 \atop 1} \right\} = 0$. This in turn implies $\left\{ {0 \atop k} \right\} = 0$ for all $k > 0$, and by further extension $\left\{ {n \atop k} \right\} = 0$ for all $k > 0, n < 0$.

In contrast, under the definition $\left\{ {0 \atop 0} \right\} = 0$, we have from the recurrence relation

$$\left\{ {0 \atop 1} \right\} = \left\{ {1 \atop 1} \right\} - \left\{ {0 \atop 0} \right\} = 1 \ .$$

From this we have $2\left\{ {0 \atop 2} \right\} = \left\{ {1 \atop 2} \right\} - \left\{ {0 \atop 1} \right\} = -1$. Thus by induction, it is easy to establish

$$\left\{ {0 \atop k} \right\} = \frac{(-1)^{k+1}}{k!} \ , \qquad \text{for} \quad k > 0 \ .$$

To proceed further, we notice that the recurrence relation for partition numbers can be written

$$k\left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\} = \left\{ {n+1 \atop k} \right\} \ .$$

If n is regarded as fixed and $\left\{ {n+1 \atop k} \right\}$ is regarded as known, then the recurrence relation can be solved. We therefore state the lemma

Lemma. The recurrence relation $mS_m + S_{m-1} = g_m$ has the solution

$$S_m = \frac{(-1)^m}{m!} \left[S_0 + \sum_{j=1}^{m} (-1)^j (j-1)!\, g_j \right] \ .$$

Proof. Direct substitution in the recurrence relation. □

We now apply this to

$$k\left\{ {-1 \atop k} \right\} + \left\{ {-1 \atop k-1} \right\} = \left\{ {0 \atop k} \right\} = \frac{(-1)^{k+1}}{k!}$$

giving

$$\begin{aligned} \left\{ {-1 \atop k} \right\} &= \frac{(-1)^k}{k!}\left\{ {-1 \atop 0} \right\} + \frac{(-1)^{k+1}}{k!} \sum_{j=1}^{k} \frac{1}{j} \\ &= \frac{(-1)^k}{k!}\left\{ {-1 \atop 0} \right\} + \frac{(-1)^{k+1}}{k!} H_k \ , \end{aligned}$$

where H_m is a harmonic number. It seems natural to continue with $\left\{ {-1 \atop 0} \right\} = 0$, giving

$$\left\{ {-1 \atop k} \right\} = \frac{(-1)^{k+1}}{k!} H_k \ .$$

VIII. Conclusions

We have considered two computational problems in this paper. The first problem is a more efficient algorithm for evaluating the known Stirling Cycle numbers with positive arguments. For negative arguments there are no commonly accepted definitions for either cycle or partition numbers. We have therefore explored different possibilities. The advent of proposals for generalizing Stirling numbers to complex arguments has pointed out the possibility that numbers previously considered zero could become nonzero, and new symmetries in the definitions become possible.

Stirling numbers are commonly defined by recurrence relations and boundary conditions. Some of these boundary conditions are determined from the combinatorial interpretation of the numbers, while others are matters of convention. This work has pointed out that different conventions from those used until now can lead to interesting new behaviour.

References

[1] Abramowitz, M., Stegun, LA., editors, Handbook of Mathematical Functions, U.S. National Bureau of Standards, 1964.

[2] Comtet, L., Advanced Combinatorics, D. Reidel Publishing Co., Dordrecht, Holland, 1974.

[3] Corless, R.M., Essential Maple 7. A guide for Scientific Progammers, Springer-Verlag, Berlin, second edition, 2004.

[4] Corless, R.M., Jeffrey, D.J., Knuth, D.E., A sequence of Series for the Lambert W Function, Proceedings of ISSAC 1997, ed. W.W. Kuechlin, ACM Press, 1997.

[5] Phillipe Flajolet & Helmut Prodinger, *On Stirling Numbers For Complex Arguments And Hankel Contours*, SIAM J. Discrete Math., Vol. 12, No. 2, pp. 155–159, 1999.

[6] Graham, R.L., Knuth, D.E., Patashnik, 0., Concrete Mathematics, Addison-Wesley Publishing Co., Reading, Massachusetts, 1994, second edition.

[7] Howard, F.T., Associated Stirling numbers, Fibonacci Quart., vol 18, pp. 303-315, Dec. 1980.

[8] Karamata, J., Theoreme sur la sommabilite exponentielle et d'autres sommabilites rattachant, Mathematica, Cluj, Romania, vol. 9, 1935, 164-178.

[9] Knuth, D.E., The Art of Computer Programming, vol. 1, Reading, Massachusetts, Addison-Wesley, 1997, third edition.

[10] Knuth, D.E., Two Notes on Notation, The American Mathematical Monthly, vol. 99, 1992, 403-422.

[11] Knuth, D.E., Convolution Polynomials, The Mathematica Journal, vol. 2, 4, 1992, 67-78.

[12] Sprugnoli, R. and Del Lungo, A., Semireal Stirling Numbers of the second kind. Technical report, Universitá di Firenze, 1994.

[13] Stirling, J., Methodus Differentialis, London, 1730.

[14] Corless, R.M., Jeffrey, D.J., Wang, Yang, Numerical Evaluation of Contour Integrals for Computation of Stirling Numbers. OPSFA 13, NIST, Washington 2015.