# A Map-Folding Problem

By W. F. Lunnon

Abstract. An algorithm is described to compute the number of ways of folding a one-dimensional map, and a table of values given.

Introduction. In how many ways can a map be folded up? What follows is restricted to the one-dimensional problem, that is: Given a plane chain ('map') of p equal segments ('leaves') jointed together, in how many ways ('p-foldings') $G(p)$ can it be rigidly collapsed into one segment? For example $G(3) = 6$, (see the figures (1)–(6) and the table (31)). In the figures, the free edge of the 'front cover' is tagged with a dot; by turning the folding round (in the plane) or over (out of the plane), this tag can always be arranged on the left and pointing upwards. We shall assume it to be so.

$$(1) \qquad (2) \qquad (3) \qquad (4) \qquad (5) \qquad (6)$$
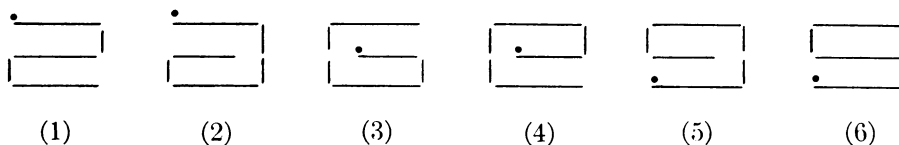
The leaves can be numbered 1, 2, $\cdots$, p in natural order from the front, and in any particular folding they can also be numbered by position from the top; e.g., the figure (3) has leaves 1, 2, 3 in positions 2, 3, 1. So to each folding there corresponds a unique permutation which (as well as its inverse) describes the folding in a natural mathematical fashion. This fact seems of no great utility.

The problem can be restricted by requiring that the two free ends of the map be joined together, creating a closed chain. Here one approach to enumeration would be to first flatten the map into a 'tree' with $\frac{1}{2}p$ branches, then manipulate this tree into a folding (a generalisation of the original problem for $\frac{1}{2}p$).

The Theory. $G(p)$ can be computed by simple enumeration, and we have found no radical improvement on this method. But for given p we can 'reduce' the set of all foldings to a subset, i.e., construct an equivalence relation over the set and then enumerate just one from each equivalence class.

There are several such equivalences of order 2 : e.g., in the figures (1)–(3) the first crease (between leaf 1 and 2) turns down, but in (4)–(6) up; and, in general, just half the foldings turn down at the first crease, so we need only enumerate this subset.

It follows that 2 divides $G(p)$. It is observed empirically (table (34)) that $2p$ divides $G(p)$, which suggests an equivalence of order $2p$. We shall prove the divisibility and demonstrate the subset.

---

(7)   (8)   (9)   (10)   (11)
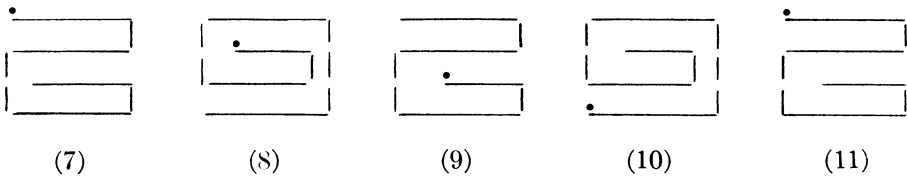
Consider figure (7). Leaf 3 is in position 4. Let it be translated supernaturally through the other leaves to position 1, dragging its creases (connections to leaves 2 and 4) with it, and move all the other leaves down one position to make room. The result is figure (8). We call this operation 'rotation': (8) rotates to give (9), and so on till (11) = (7) again. In defiance of the well-known law that maps never fold up so that the front cover is on top, we shall call such a folding 'normal'—figure (7) is normal. Suppose we take the set of all normal foldings and rotate it $k - 1$ times, where $1 \leqq k \leqq p$; the result will be the set of all foldings whose first leaf is in position k. So all these sets contain the same number of foldings, $G(p)/p$, since there are p possible values of k.
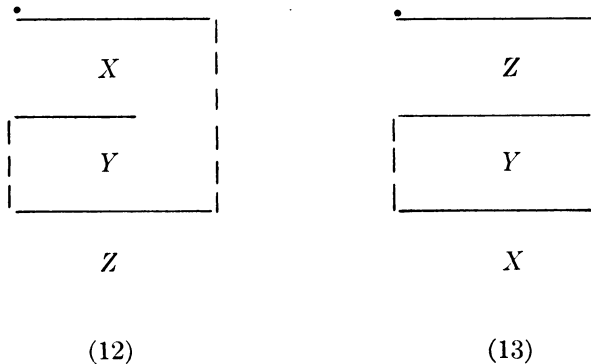
(12)   (13)

We now want to reduce the set of normal foldings by a factor of two. Instead of discriminating on the first crease (which must perforce turn downwards) we use the second : consider figure (12)–(13). Here X, Y, Z denote the rest of the leaves after the third, assuming $p \geqq 3$. If figure (12) is rotated once backwards and turned upside down (out of the plane), the result resembles (13) except that its leaves (and its tag) face down. This is remedied by undoing it, turning it over and doing it up again in the same shape; the entire operation will then transform the set of normal foldings whose second crease turns up (like (12)) into the set whose second crease turns down (like (13)). So both sets contain the same number of foldings, $G(p)/2p$, the latter must be an integer, and we have reduced the set of all foldings to the set of normal foldings whose second crease turns up. Q.E.D.

The prime factors of $G(p)/2p$ turn out depressingly large (table (35)), so that no larger equivalence classes are possible.

The Ratio. The ratio $G(p)/G(p - 1)$ is interesting (table (22)). It can be interpreted as the average number of p-foldings derivable from a given $(p - 1)$-fold-

ing by attaching a pth leaf (see next section). For even and odd p separately, it appears monotone increasing to a pair of limits rather larger than 3; so it seems reasonable to try fitting a pair of polynomials in $1/p$. [This is performed for several polynomial degrees, and repeated as a check on sequences which are known to converge and diverge.] The result is that for both polynomials the constant term is very nearly $3\frac{1}{2}$, suggesting the conjecture that $G(p)/G(p-1) \to 3\frac{1}{2}$.
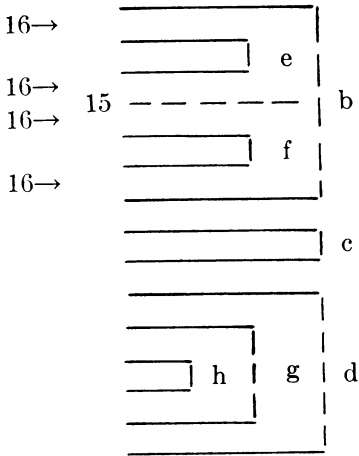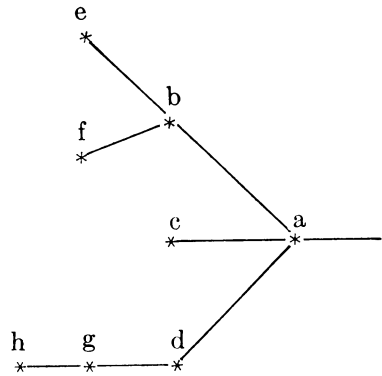
(14)   (15)

In an attempt to prove this, we reason thus. Let p be even and consider just the right-hand half of an arbitrary $(p-2)$-folding (see (14) where $p=16$). This half-folding is abstractly equivalent to an ordered rooted tree with $\frac{1}{2}p$ nodes (figure (15)): each 'm-crease' (immediately enclosing m gaps) corresponds to an 'm-node' (at which there are m branches). [The correspondence is spoiled by the root node a, complete with spurious extra branch. It is convenient to imagine an external crease, enclosing the entire half-folding, to correspond to a; and a's extra branch simplifies the definitions.] Suppose now that leaf $(p-1)$ appears in one of the gaps, which will be enclosed by some m-crease (leaf 15 is enclosed by the 3-crease b in (14)). Then there will be $1+m$ choices for leaf p (4 choices for leaf 16), and $G(p)/G(p-1)$ = the mean of $1+m$ over the set of all $(p-1)$-foldings.

We now make two assumptions, both of which are untrue: that in the set of all $(p-2)$-foldings, each distinct half-folding occurs equally often; and that each gap of a given $(p-2)$-folding is equally likely to be occupied by leaf $(p-1)$. Let $a(p, m)$ be the total of m-creases which occur in the set of all distinct half-foldings = the total of m-nodes in the set of all ordered rooted $\frac{1}{2}p$-trees

(16)
$$= \binom{p-m-2}{\frac{1}{2}p-2}.$$

Then the mean of $1+m$ = 1 + the mean number of gaps immediately within the same crease as an arbitrary gap of an arbitrary half-folding

$$= 1 + \sum_{1}^{\frac{1}{2}p} a(p, m)m^2 \Big/ \sum_{1}^{\frac{1}{2}p} a(p, m)m$$

(17)
$$= 1 + \left[\binom{p-1}{\frac{1}{2}p - 1} + 2\binom{p-1}{\frac{1}{2}p - 2} \right] \Big/ \binom{p-1}{\frac{1}{2}p - 1}$$

$$= 4p/p + 2 \qquad = G(p)/G(p-1) \ ?$$

Steps (16) and (17) require wads of binomial coefficient manipulation for their proof; we shrink from inflicting this on the reader, since for one thing the estimate we have derived is too high (table (33)). Its limit is 4.

As a check on the proceedings, we estimate the mean number of gaps per crease

$$= \sum_{1}^{\frac{1}{2}p} a(p, m)m \Big/ \sum_{1}^{\frac{1}{2}p} a(p, m)$$

$$= \binom{p-1}{\frac{1}{2}p - 1} \Big/ \left[\binom{p-1}{\frac{1}{2}p} - \binom{p-2}{\frac{1}{2}p - 2} \right] = (p-1)/\tfrac{1}{2}p,$$

which is obviously true (independently of our assumptions) since there are $p - 1$ gaps and $\frac{1}{2}p$ creases (including the imaginary one) in any half-folding.

The Algorithm. The foregoing remarks leave the computational problem of $G(p)$ unchanged: to enumerate (a subset of) all p-foldings. The most successful algorithm will be described; it is a natural adaptation of the intuitive hand method. The basic loop constructs and counts all i-foldings which can possibly be made by attaching an ith leaf on the tail end of a given $(i - 1)$-folding, by searching for gaps in the appropriate area of the $(i - 1)$-folding. Starting this loop on the trivial 1-folding and calling it recursively down to level $i = p$, we get all possible p-foldings. A vector FOR is kept : FOR(k) contains the number of the leaf in the next position after leaf k in the current i-folding, so completely describing it, and new leaves are attached by altering the entries in FOR.
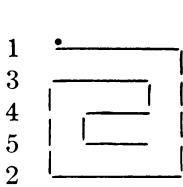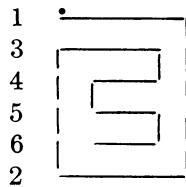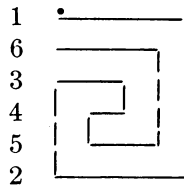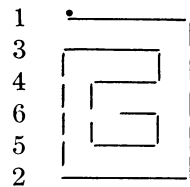

(18)   (19)   (20)   (21)

Consider (18), the FOR-vector for which is (22). Suppose that $p > 6$, and in the course of the enumeration we have constructed the 5-folding shown. To construct all $(i = 6)$-foldings of which it is the front, first set some j equal to 5. Then :

Insert leaf 6 between leaf j and the one in front, by setting FOR(6) = FOR(j) and FOR(j) = 6. Count one more 6-folding, and call the construction recursively to insert leaf 7. On return, reset FOR(j) = FOR(6) and step j on to the leaf joined to FOR(j) on the right (left for odd i).

Now repeat the last paragraph until FOR(j) = 5 again, when return to the level above ($i = 5$).

k	0	1	2	3	4	5
FOR(k)	1	3	0	4	5	2

$$(22)$$

In (18), this process locates three gaps for leaf 6, j taking the values 5, 1, 4. The resulting 6-foldings are (19)–(21). Notice that there is no need to search both down and up from leaf 5; the algorithm automatically continues the search from the lowermost gap (above leaf 2) to the uppermost (below leaf 1). This happy feature can be extended : if leaf 1 is conventionally 'joined' to itself on the left, and a leaf 0 is invented which is joined to itself at both sides, is before the top leaf (FOR(0) = 1 in (22)), and is after the bottom leaf (FOR(2) = 0), then there is no longer any need to test for the free edge of leaf 1 nor for the bottom of the folding : the previous algorithm is sufficient. Figure (23) illustrates these modifications on (18).
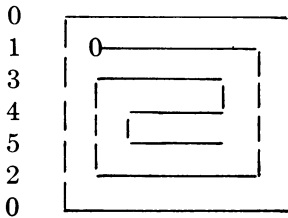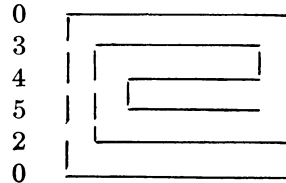
(23)   (24)

In order to compute $G(p)/2p$ rather than $G(p)$, leaf 1 is omitted and the right edge of leaf 2 is joined to leaf 0 (figure (24)). This ensures that leaf 2 is always accessible from the exterior, and so can be joined to leaf 1 in position 1. To force the second crease upwards, j is initialised to 0 instead of to 2 at the start of the search for gaps for leaf 3 (there are only 2 of them anyway).

To speed the inner loop we do not actually insert leaf p; it is enough to just count the gaps at this bottom level. All intermediate values of $G(i)$, $i < p$, come out in the wash instead of requiring separate calculation.

The Program. Three programs realising basically this algorithm were written by the author for the Manchester Atlas I computer : an Atlas Autocode program, a machine code program, and a machine code program in which the recursion was 'unwound' into a nest of $p - 3$ similar sequences. The third was 1.5 times faster than the second, which was 7 times faster than the first. The factor of 7 is due largely to the Atlas I modifier registers ('B-lines') being particularly suited to the algorithm, whereas Atlas Autocode performs all assignments full-length using the accumulator. The time taken is strictly proportional to the size of the answer $G(p)/2p$: 9 basic instructions obeyed per folding constructed in the final program. At fixed intervals the

inner loop punches out the current j's and partial answers, together with a check sum, from which the program can easily be restarted. Very little store is used, which makes the program ideal for time-sharing with magnetic tape- and output-limited jobs.

For $p \leq 20$, table (34) has been computed twice by the same program; but for $21 \leq p \leq 24$, there exists only a computation lasting 55 hours (46 useful), during which undetected machine errors are known to have occurred at least five times. These were probably digit-pairs dropping out of the store. In three cases they led rapidly to a halt through the generation of an illegal address, and in two the computation went into a characteristic loop in which the partial answers below a certain number of leaves ceased to increase. This behaviour seems more probable than apparently correct continuation, but suspicion remains. P. H. Robinson has also programmed the problem for the PDP 8, which has nominal speeds much the same as Atlas but was in practice 8 times as slow, owing to the restricted instruction code, etc. He confirms table (31) for $p \leq 15$.

In a lighter vein, we observe that for $p = 24$ the author's present program is faster by a factor of a million than his first attempt, and by a spectacular 10^{16} than another early program.

p	$G(p)$	$\dfrac{G(p)}{G(p-1)}$	$\dfrac{4p}{p+2}$	$G(p)/2p$	Prime Factors
0	0			0	
1	1			½	
2	2	2.00000	2.00	½	
3	6	3.00000		1	
4	16	2.66667	2.67	2	2
5	50	3.12500		5	5
6	144	2.88000	3.00	12	2.2.3
7	462	3.20833		33	3.11
8	1392	3.01299	3.20	87	3.29
9	4536	3.25862		252	2.2.3.3.7
10	14060	3.09965	3.33	703	19.37
11	46310	3.29374		2105	5.421
12	1 46376	3.16079	3.43	6099	3.19.107
13	4 85914	3.31963		18689	11.1699
14	15 57892	3.20611	3.50	55639	55639
15	52 02690	3.33957		1 73423	61.2843
16	168 61984	3.24101	3.56	5 26937	526937
17	565 79196	3.35543		16 64094	2.3.53.5233
18	1849 40388	3.26870	3.60	51 37233	3.1712411
19	6229 45970	3.36836		163 93315	5.151.21713
20	20502 28360	3.29118	3.64	512 55709	47.73.14939
21	69279 64218	3.37912		1649 51529	3.47.53.22073
22	2 29301 09884	3.30979	3.67	5211 38861	31.16810931
23	7 76921 42980	3.38822		16889 59630	2.5.223.757381
24	25 83605 86368	3.32544	3.69	53825 12216	2.2.2.7529.89363
	(31)	(32)	(33)	(34)	(35)

The author is indebted to C. F. J. Outred for, among other things, the notion of rotation. The referee has pointed out that in the table for P_n ($= G(n)/n$ in the present notation) of [1, p. 397] the last entry should read 12198 instead of 12196. There are further references in [1].

Department of Computer Science
University of Manchester
Manchester 13, England

1. J. Touchard, "Contributions à l'étude du problème des timbres poste," *Canad. J. Math.*, v. 2, 1950, pp. 385–398. MR 12, 312.

The Maxima of $P_r(n_1, n_2)$

By M. S. Cheema* and H. Gupta

1. In this note, we study the maxima of $P_r(n_1, n_2)$, the number of partitions of the vector (n_1, n_2) into exactly r parts (vectors) with positive integral components.

The generating function $\phi_r(x_1, x_2)$ for $P_r(n_1, n_2)$ is given by

$$(1.1) \qquad \prod_{k_1, k_2=1}^{\infty} (1 - zx_1^{k_1} x_2^{k_2})^{-1} = 1 + \sum_{r=1}^{\infty} z^r \phi_r(x_1, x_2)$$

$$(1.2) \qquad \phi_r(x_1, x_2) = 1 + \sum_{n_1, n_2=1}^{\infty} P_r(n_1, n_2) x_1^{n_1} x_2^{n_2} .$$

2. If $q_r(n_1, n_2)$ denotes the number of partitions of (n_1, n_2) into at most r parts (vectors) with nonnegative integral components, then it follows that $q_r(n_1, n_2) = P_r(n_1 + r, n_2 + r)$. It is clear that $q_r(n_1, n_2)$ is an increasing function of r for $1 \leq r < n_1 + n_2$, and becomes constant for $r \geq n_1 + n_2$, on the other hand $P_1(n_1, n_2) = 1$ and $P_r(n_1, n_2) = 0$ for $r > \min (n_1, n_2)$. From the table of values of $P_r(n_1, n_2)$ computed by Cheema, we notice that for $n_1 \geq n_2 > 0$, there is a unique s such that

$$P_1(n_1, n_2) < P_2(n_1, n_2) < \cdots < P_s(n_1, n_2) \geq P_{s+1}(n_1, n_2) \geq \cdots \geq P_{n_2}(n_1, n_2) .$$

We use s in this sense in all that follows. The values of s were computed for all n_1, $n_2 \leq 50$. We might remark that a similar conjecture holds for the number of partitions of n into exactly r summands. An explicit formula for $P_r(n_1, n_2)$ for general r is not known, $P_r(n_1, n_2)$ do satisfy a recurrence relation and behave very much like a polynomial in n_1, n_2, i.e., $P_r(n_1, n_2)$ is a semipolynomial of degree $r - 1$ in n_1 and n_2 relative to modulus $r!$ as shown by Wright [2]. Thus

$$P_r(n_1, n_2) = \sum_{t_1=1}^{r} \sum_{t_2=1}^{r} \beta(t_1, t_2, n_1, n_2) n_1^{t_1-1} n_2^{t_2-1} ,$$

where $\beta(t_1, t_2, n_1, n_2)$ depends on r, t_1, t_2 and on the residues of n_1, n_2 to moduli 1, 2, 3, $\cdots$, $[r/t_i]$, but not otherwise on n_1, n_2. A rough estimate for s is obtained by studying the maxima of a function which behaves very much like $P_r(n_1, n_2)$.