MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A. I. LABORATORY

Artificial Intelligence
Memo No. 290 June 1973

PATERSON'S WORM

Michael Beeler

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

PATERSON'S WORM

abstract: A description of a mathematical idealization of the feeding pattern of a kind of worm is given.

Certain prehistoric worms fed on sediment in the mud at the bottom of ponds. For efficiency, they would not retrace paths which had already been traveled, since little food was left there. Yet food probably occurred in patches, so it was desirable to stay near previous trails. Worms had innate "rules" regarding how close to "eaten paths" to stay, how far to go before turning around, how sharp a turn to make, etc. These rules varied from species to species, and paleontologists can trace the development of species and determine the similarity of different species by comparing fossil records of worm tracks.

(See Science magazine, 21 November 1969, for further details and a discussion of computer simulation of natural worm tracks.)

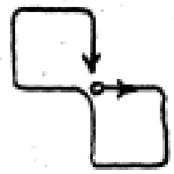
Early in 1971, Michael Paterson mentioned to me a mathematical idealization of the prehistoric worm. He and John Conway had been interested in a worm constrained to eat food only along the grid lines of graph paper. Take, for instance, quadrille paper, and let a "worm egg" hatch at an intersection in an arbitrarily large grid of food. The worm starts eating in some direction, say east (E). When it has traveled one unit of distance, it arrives at a new intersection. Its behavior at this (and every following) intersection is determined by a set of fixed, innate rules. Each rule is of the form, "if the intersection has distribution D of eaten and uneaten segments, then leave the node via (uneaten) grid segment G."

[comment: This can be viewed equivalently as an unmoving, finite-state automaton with an infinite 2-dimensional "tape" which it can mark and read. This is slightly different from automata whose data/program is supplied on the tape; here the tape is entirely blank (or filled, with food) and all information is in the nature of the machine.]

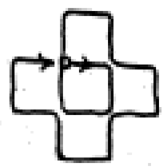
If a worm, arriving at a node with no segments eaten (except of course the one it just ate) should find in its rules, "for this distribution, go straight," then the worm will go straight forever. Since this is neither very interesting to us, nor very useful to a real worm, who would quickly reach the edge of its food patch, we discard it. We require that, upon discovering a virgin node, all sets of rules must say to turn. To avoid mirror-image symmetric duplication, we require that the turn be to the worm's right (clockwise as seen from above).

The intrepid quadrille worm therefore turns right, now eating to the south (S). It will next go W, then N, returning to its birthplace, "the origin." It now encounters a non-trivial situation. To its left and straight ahead are uneaten, but to its right is eaten. It cannot turn right, and which of the two possible directions it takes will depend on what species of worm it is.

Consider one species, where it turns left (W). Then it goes N, E, and S, returning to the origin the second time. This time there are no uneaten segments, so it dies. The fossil it leaves is shown below.



A second species of quadrille worm would go straight (N) when it first returns to the origin. It then goes E, then S, meeting its own path. Here there is only one segment to eat: E. After a few more turns, this worm also finds itself returning to the origin the second time.



These two paths exhaust the variety of species of simple quadrille worms. John Conway introduced more variety by allowing the worm to sense the distribution of eaten and uneaten segments at neighboring nodes, as well as the node where the worm is. This allows distributions which used to be indistinguishable now to be treated independently. The worm can "look ahead" somewhat, and, with a fortuitous set of rules, avoid committing itself to an early demise.

Mike Paterson, on the other hand, introduced more variety by placing the worm on a triangular food grid. Each node is now the meeting point of six segments instead of only four. This leads to a larger set of rules, allowing greater variation in resulting worm tracks.

We already mentioned three general rules:

— A worm must turn if no segments are eaten at the current node.
— When all segments are eaten, the worm dies.
— When only one uneaten segment exists, the worm must take it.

In addition, there are other rules which vary from one species of worm to another:

(1) The turn at a node where no segments are eaten may be either gentle or sharp.

(2) When the worm encounters its path, there are four distributions it may find. The worm must have a separate rule for each of these, specifying which of the three uneaten segments to choose.

(3) When the worm first returns to the origin, it might approach on any of the five uneaten segments. For each of these cases, the worm needs a rule specifying which of the four uneaten segments to choose.

(4) The worm's second return to the origin can happen in ten different ways, but each of the ten rules has to specify a choice between only two uneaten segments.

This includes all the rules the worm needs, for we have accounted for every situation that may arise. As it reaches a node, there can be 0, 1, 2, 3, 4 or 5 segments eaten, and we have discussed each.

The number of different possible sets of rules may seem large, but this is not particularly so. For convenience, each set may be rendered as a number code. Using octal, we can assign rules as follows:

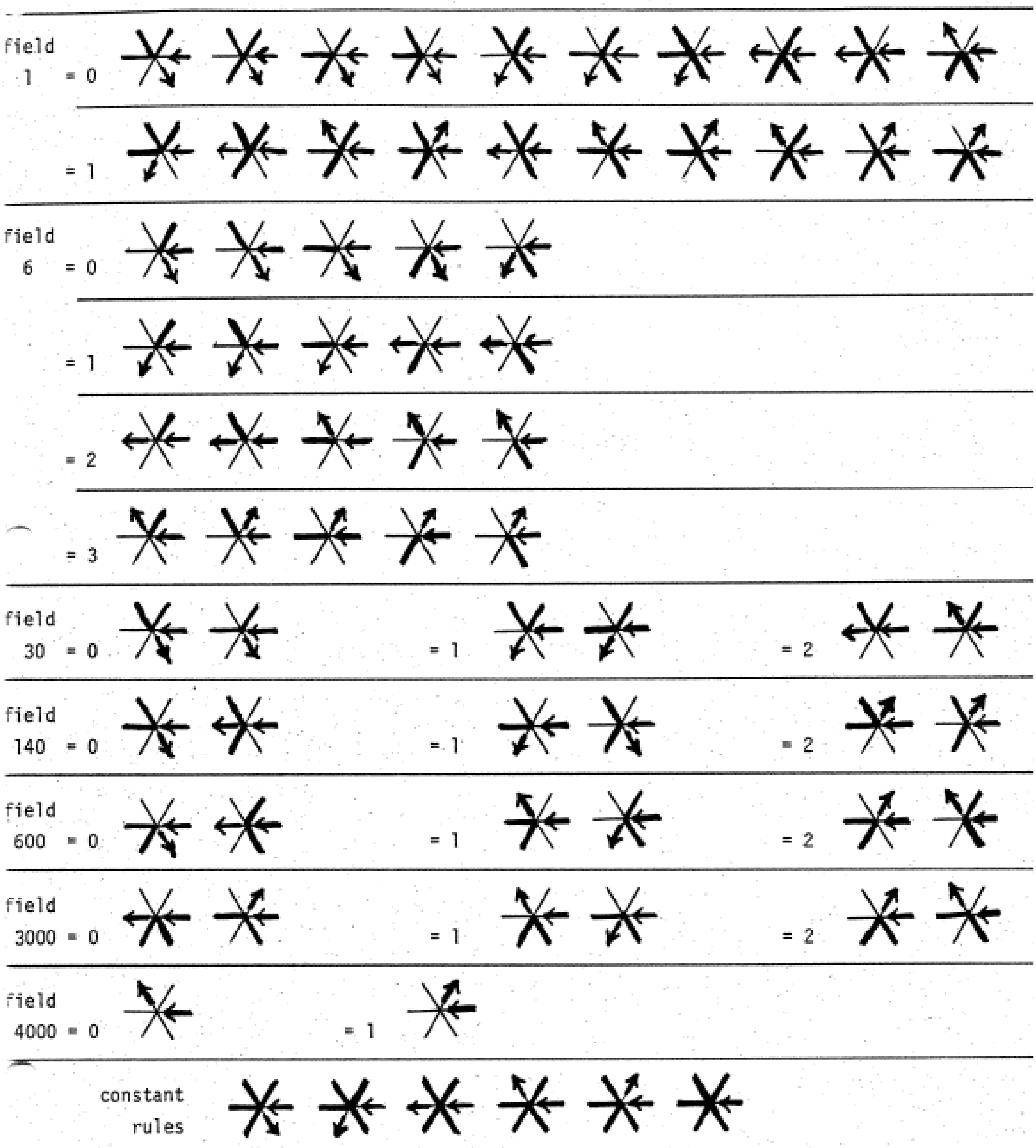
field	rules
4000	gentle or sharp turn where no segments eaten
3000 600 140 30	specify action when worm encounters its path
6	selects action at first return to origin
1	selects action at second return to origin

The data in the 4000 and the 1 fields may be either 0 or 1; that in the 6 field may be 0, 1, 2 or 3; that in the other fields has room (2 bits) for four values, but there are only three uneaten segments in these cases. Thus, the data in fields 3000, 600, 140 and 30 may be only 0, 1 or 2, and never 3!

Thus there are 2 X 3 X 3 X 3 X 3 X 4 X 2 = 1296 possible sets of worm rules.

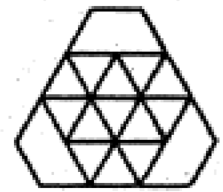
The particular number code that I have used is shown below.

if field...   contains...   then set contains these rules:



For example, rule code 0423 contains the following pertinent rules, which apply in creating its path as shown:

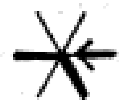
4000 field = 0	600 field = 2	140 field = 0	30 field = 2	6 field = 1	1 field = 1



0423

The distribution of eaten and uneaten segments at a node is rotated until it matches one of the rules. The reader may find it fun to trace out this pattern by applying the rules given. (Newly-hatched worms leave origin to the right.)

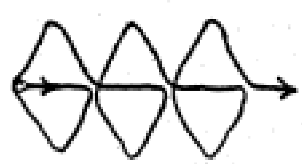
Often several rule codes result in the same path. For example, the above path never encountered the distribution relevant to the 3000 field, namely:



Thus, rule codes 0423, 1423 and 2423 produce the same path. As it turns out, several other rule codes also produce the same path, but not all of these trace it in the same order (2520, for example).

Two statistics which are useful in classifying worm paths are the length (number of segments) and the number of different nodes visited. The 0423 worm's path, for example, is 33 (segments) long and visits 18 nodes. Another parameter is the ratio of length to nodes, which roughly corresponds to density. It can easily be seen that $1 < L/N < 3$. If A, B and C are the number of nodes visited once, twice and three times, respectively, then $L/N = (A+2B+3C)/(A+B+C)$. From this the ratio can be applied to non-finite paths (see below), by computing A, B and C for whatever unit is replicated in their growth.

Some paths never terminate, that is, they never return to the origin a third time. Some of these are trivial paths I call "zippers," such as the rule code 4016:



etc.

Others are spirals, which wrap layer after layer of path on the outer edge of the visited area. The worm circles the origin an infinite number of times in spiral paths, always in a clockwise direction (which is to be expected from the right-turn rule where no segments are eaten).

Yet another class of infinite worm paths is the "shoot growers." These, after some finite interval of fairly standard behavior, fall into a repetitive spiraling action. Each revolution of the spiral is the same size as the previous, but displaced further from the origin. The worm crawls away to infinity in this complex fashion.

About a dozen rule codes result in paths too long for my program to trace, and which are not any of the obviously infinite species mentioned above. Some of these uncertain-length paths seem to be following regular methods of filling in areas. For example, those which take a sharp turn where no segments are eaten have a behavior reminiscent of crystal growth. Other uncertain-length paths appear as unstructured as, and similar to, some finite-length paths.

One area for further investigation is the fate of the worms with uncertain-length paths. I suggest a theoretical approach, examining the rules involved and the path near the origin, rather than more brute force computation. (With a different program, rule code 2327 was traced to 1,150,000 segments without ending, and 5401 to ten million segments.)

Another area for investigation is the interaction between a worm and initial tracks in, or boundaries of, the food grid. Interaction between two worms (not necessarily of the same species) could prove similarly interesting. This sort of work was mentioned by Mike Paterson. He felt it was natural to model a worm which crawled some distance in a straight line (perhaps seeking food, or leaving its hatching ground) before applying its set of rules.

[comment: Note that in such investigations distinctions must be made among groups of worms which I have considered identical. For instance, I have specified a worm's action upon meeting a sharp bend with the same code field as for meeting a gentle bend. For any given worm in my investigation, only one or the other kind of bend will ever be encountered. Also, my codes contain no rule applicable to meeting a straight line.]

Some of these perturbations might provoke drastically different behavior. 2006, for instance, resembles the shoot growers 2000 and 2007 in code and general behavior. The proper stimulus might provoke it to germinate.

Yet another aspect of worm automata open for investigation is the behavior of worms on three-dimensional (or n-dimensional!) lattices. Some problems exist in choosing a good equivalent to the rule, "always turn right at a virgin node." What kinds of paths exist in cubic, or in hexagonal-close-packed lattices? Is there a volume analog to the area-filling spiral paths?

As a final comment, and admittedly an aesthetic one, I point out that some of these paths are quite intricate and beautiful. For example, rule codes 2244, 2202 and 2207 I call the doily, hyper-doily and hyper-hyper-doily; 1247 is a beard, and 1243 an ocarina.

L/N VALUES

As a very general rule, longer worm paths have larger L/N ratios. The following table lists each worm path whose L/N exceeds that of all shorter paths. Entries in parentheses exceed only those paths with both a smaller length and the same bend where no segments are eaten.

rule code (octal)	length	nodes	L/N
4000	9	7	1.2857
4022	12	8	1.5
4021	15	9	1.6667
4027	21	12	1.75
(520	23	16	1.4375)
(420	29	18	1.6111)
423	33	18	1.8333
(5041	37	21	1.7619)
(5101	45	25	1.8)
6007	48	24	2.0
(1223	62	33	1.8788)
525	99	46	2.1522
462	114	52	2.1923
5107	411	181	2.2707
5307	438	181	2.4199
(1204	451	194	2.3247)
5201	609	246	2.4756
(5	970	411	2.3601)
(1007	1515	640	2.3672)
(4	1660	681	2.4376)
512	1742	699	2.4921
2512	2377	940	2.5287
5207	2478	915	2.7082
(2016	3943	1520	2.5941)
(7	5132	1964	2.6130)
(410	5715	2145	2.6643)
2412	10307	3736	2.7588
2416	22847	8066	2.8325
2227	220142	77257	2.8495

The above table includes only paths known to be finite. As far as I have traced them, most of the gentle bend uncertain-length paths have 1.5 < L/N < 2.0. Four, however, have L/N greater than any listed above. And all three sharp bend uncertain-length paths have even greater L/N:

rule code (octal)	length*	nodes*	L/N
2327	157588	55279	2.8508
2322	225302	78892	2.8558
2205	252410	87989	2.8687
2222	345046	119754	2.8813
5401	85614	29241	2.9279
5407	137570	46794	2.9399
5405	183866	62247	2.9538

*partial results only

The L/N of the zippers is 5/3 for sharp bend and 11/5 for gentle bend. The spirals like rule code 56 fill areas with nodes visited twice, so their L/N approaches 2. The L/N values of other infinite-length paths are left as an exercise. The ten paths with lowest L/N are listed below.

rule code (octal)	length	nodes	L/N
447	33	26	1.2692
14	28	22	1.2727
4000	9	7	1.2857
1200	44	34	1.2941
1100	34	26	1.3077
206	48	36	1.3333
1044	46	33	1.3939
1324	93	66	1.4091
2040	50	35	1.4286
520	23	16	1.4375

The "ocarina" (code 1243) has an unusually low L/N (1.6703) for its large length (7565).

NON-UNIQUE PATHS

As mentioned above, some paths are created by more than one rule code. Only 209 of the 1296 rule codes create unique paths. Some paths are created by exactly two rule codes; of these, most (35) are created by the rule code listed in the master table given later, and that rule code plus one. For example, 520 and 521, or 2100 and 2101, which have the shortest and longest paths of all such rule code pairs. The eleven pairs which are not rule code and rule code + 1 are listed below.

rule codes		length
1224	1264	67
1412	1512	112
1413	1416	134
1242	1244	138
1513	1516	152
5307	5507	438
5201	5205	609
5001	5005	615
2303	2305	2754
43	45	infinite
413	416	infinite

There are 44 paths which are created by more than two rule codes. Each of these is listed below, with the numerically smallest rule code which creates it. Also listed is the binary representation of the general form for all rule codes which create it. An "X" means that the bit may be 0 or 1 (subject to certain fields having only three values, as noted previously). An "A" or "B" means the bit is restricted in some way, as noted.

least rule code; (number of codes creating path); length; nodes;
general form of rule codes

gentle bend finites, by length

```
  14  (63)  28  22
            0,XX,XX,XX,01,10,X (54)
         or 0,XX,10,XX,01,11,1 (9)
 420  (22)  29  18
            0,XX,10,00,10,0A,B (9) A,B not= 1,1
         or 0,XX,10,10,10,AB,0 (9) A,B not= 0,0
         or 0,00,10,01,10,00,X (2)
         or 0,01,10,10,10,00,X (2)
  22  (3)   33  22  0,XX,00,00,10,01,0
 423  (14)  33  18
            0,XX,10,00,10,AB,1 (9) A,B not= 0,0
         or 0,10,10,10,10,00,X (2)
         or 0,XX,10,10,10,01,1 (3)
 447  (3)   33  26  0,XX,10,01,00,11,1
1100  (6)   34  26  0,01,00,10,XX,00,X
1200  (18)  44  34  0,01,01,XX,XX,00,X
 206  (45)  48  36
            0,XX,01,XX,XX,11,0 (27)
         or 0,10,01,XX,XX,00,X (18)
 424  (18)  48  27
            0,0X,10,01,10,AB,1 (6) A,B not= 0,0
         or 0,0X,10,01,10,1X,0 (4)
         or 0,XX,10,00,10,1X,0 (6)
         or 0,10,10,01,10,1X,1 (2)
  23  (3)   50  30  0,XX,00,00,10,01,1
  67  (6)   57  36
            0,XX,00,01,10,11,1 (3)
         or 0,XX,10,00,00,10,0 (3)
 125  (3)   86  58  0,XX,00,10,10,10,1
  60  (6)   90  50  0,XX,00,01,10,00,X
 525  (6)   99  46  0,XX,10,10,10,1X,1
  27  (3)   110 57  0,XX,00,00,10,11,1
2264  (3)   534 283 (CODES 2264, 2323, AND 2324)
```

gentle bend infinites, by rule code

```
  12   (6)  0,XX,00,00,01,01,X
  40   (3)
            0,00,00,01,00,00,X (2)
         or 0,00,00,01,00,10,0 (1)
  42   (4)
            0,00,00,01,00,A1,B (3) A,B not= 0,1
         or 0,00,01,01,00,10,0
  52   (18)  0,XX,XX,01,01,01,X
  56   (6)  0,XX,00,01,01,11,X
 102   (18)  0,XX,00,10,XX,01,X
 106   (18)  0,XX,00,10,XX,11,X
 200   (18)  0,00,01,XX,XX,00,X
 207   (9)  0,00,01,XX,XX,11,1
 440   (15)
            0,XX,10,01,00,AB,0 (6) A = B
         or 0,XX,10,01,00,AB,1 (9) A,B not= 1,1
 450   (9)
            0,XX,10,01,01,00,X (6)
         or 0,XX,10,01,01,11,0 (3)
1410   (4)  0,01,10,X0,01,00,X
2042   (5)
            0,10,00,01,00,AB,X (4) A not= B
         or 0,10,00,01,00,11,0 (1)
2460   (4)
            0,10,10,01,10,00,X (2)
         or 0,10,10,01,10,1X,0 (2)
```

sharp bend finites, by length

```
4000  (162)  9   7  1,XX,XX,XX,XX,X0,0
4022  (81)  12   8
            1,XX,XX,XX,10,01,X (54)
         or 1,XX,XX,XX,10,11,0 (27)
4021  (54)  15   9  1,XX,XX,XX,10,X0,1
4002  (81)  18  12
            1,XX,XX,XX,00,01,X (54)
         or 1,XX,XX,XX,00,11,0 (27)
4027  (27)  21  12  1,XX,XX,XX,10,11,1
6001  (18)  27  16  1,10,XX,XX,00,X0,1
4001  (18)  30  18  1,00,XX,XX,00,X0,1
5041  (6)   37  21  1,01,XX,01,00,X0,1
5101  (6)   45  25  1,01,XX,10,00,X0,1
6007  (9)   48  24  1,10,XX,XX,00,11,1
4007  (9)   63  33  1,00,XX,XX,00,11,1
5047  (3)   66  36  1,01,XX,01,00,11,1
```

sharp bend infinites, by rule code

```
4011  (108)
            1,XX,XX,XX,01,01,X (54)
         or 1,XX,XX,XX,01,X0,1 (54)
4016  (54)  1,XX,XX,XX,01,11,X
```

MASTER TABLE

This table lists each of the 299 distinct paths.

gentle bend finite paths

For each of these 227 paths is listed the smallest rule code which creates it, how many rule codes create it, its length, and the number of nodes it visits.

code		L	N
520	(2)	23	16
14	(63)	28	22
420	(22)	29	18
22	(3)	33	22
423	(14)	33	18
447	(3)	33	26
1100	(6)	34	26
1460	(2)	35	20
1400	(2)	39	26
407	(1)	42	28
64	(1)	43	29
1200	(18)	44	34
1403	(1)	44	29
1002	(1)	45	30
1506	(1)	45	31
510	(2)	46	31
1044	(1)	46	33
206	(45)	48	36
424	(18)	48	27
23	(3)	50	30
2040	(2)	50	35
2400	(2)	50	31
1500	(2)	52	34
1064	(1)	53	33
1407	(1)	53	34
2004	(1)	54	34
1213	(1)	55	33
2403	(1)	55	34
67	(6)	57	36
1462	(1)	58	32
1325	(1)	61	40
516	(1)	62	40
1223	(1)	62	33
2064	(1)	66	38
1025	(1)	67	39
1065	(1)	67	42
1224	(2)	67	44
2065	(1)	68	41
2002	(1)	69	41
403	(1)	71	43
1004	(1)	71	43
10	(2)	73	46
1245	(1)	73	45
2410	(2)	73	45
2406	(1)	77	43
1003	(1)	78	46
2510	(2)	79	48
20	(2)	81	44
24	(1)	83	46
2020	(2)	83	46
2024	(1)	85	48
125	(3)	86	58
1317	(1)	86	57
60	(6)	90	50
2010	(2)	90	56
1066	(1)	92	52
1324	(1)	93	66
1020	(2)	94	51
1024	(1)	96	53
405	(1)	97	54
525	(6)	99	46
1265	(1)	99	66
1217	(1)	101	58
1005	(1)	102	59
25	(1)	105	55
2506	(1)	105	59
62	(1)	107	57
27	(3)	110	57
1405	(1)	110	59
1412	(2)	112	66
402	(1)	113	59
502	(1)	113	64
462	(1)	114	52
1406	(1)	114	62
2504	(1)	118	66
1402	(1)	119	69
1305	(1)	122	68
1303	(1)	126	70
304	(1)	130	80
1203	(1)	130	72
1502	(1)	132	71
1413	(2)	134	78
1503	(1)	136	74
1242	(2)	138	84
2402	(1)	140	70
2503	(1)	148	81
2407	(1)	151	78
1513	(2)	152	84
1304	(1)	153	82
2	(1)	159	86
1307	(1)	162	91
412	(1)	165	90
2247	(1)	173	107
513	(1)	176	96
2066	(1)	176	87
1063	(1)	180	95
1026	(1)	181	91
2063	(1)	194	94
65	(1)	196	97
305	(1)	196	121
2003	(1)	196	102
16	(1)	201	111
224	(1)	203	132
1202	(1)	206	112
1444	(1)	212	119
2500	(2)	219	116
225	(1)	235	125
1207	(1)	236	127
1016	(1)	237	126
1212	(1)	242	127
1263	(1)	245	125
2026	(1)	247	119
2502	(1)	247	125
400	(2)	248	124
1227	(1)	248	127
1313	(1)	262	141
2516	(1)	281	141
110	(2)	284	174
1000	(2)	285	143
1312	(1)	286	153
1322	(1)	287	143
2405	(1)	289	139
63	(1)	294	136
2025	(1)	295	141
2005	(1)	296	150
2017	(1)	304	149
212	(1)	306	186
26	(1)	309	142
1262	(1)	321	158
2325	(1)	327	187
2244	(1)	330	169
1302	(1)	338	169

6	(1)	345	174
2245	(1)	347	189
17	(1)	354	178
2062	(1)	354	166
323	(1)	356	168
503	(1)	393	207
1504	(1)	398	184
2047	(1)	415	246
223	(1)	419	197
1204	(2)	451	194
213	(1)	454	281
325	(1)	475	233
1050	(2)	496	262
1042	(1)	512	273
1267	(1)	514	252
1323	(1)	533	249
2264	(3)	534	283
2267	(1)	553	287
2243	(1)	558	287
1507	(1)	561	253
322	(1)	606	273
406	(1)	609	270
2513	(1)	623	284
1017	(1)	631	281
2505	(1)	631	285
1222	(1)	636	307
2262	(1)	663	325
265	(1)	684	346
1010	(2)	697	314
1062	(1)	708	323
2006	(1)	711	338
2317	(1)	731	378

3	(1)	732	335
66	(1)	735	327
1257	(1)	738	354
505	(1)	747	362
1006	(1)	839	377
312	(1)	843	450
0	(2)	897	406
5	(1)	970	411
205	(1)	1020	536
1327	(1)	1063	472
1247	(1)	1116	486
1442	(1)	1457	669
1007	(1)	1515	640
2204	(1)	1545	712
2304	(1)	1550	711
222	(1)	1564	687
2312	(1)	1574	781
2202	(2)	1632	720
4	(1)	1660	681
2413	(1)	1672	686
1040	(2)	1688	765
512	(1)	1742	699
2265	(1)	1831	835
1225	(1)	1932	865
2213	(1)	1978	884
504	(1)	2056	900
2225	(1)	2155	955
2512	(1)	2377	940
2507	(1)	2526	1008
2307	(1)	2565	1144
1046	(2)	2578	1114
2303	(2)	2754	1208

2263	(1)	2811	1214
507	(1)	2857	1179
50	(2)	3566	1556
500	(2)	3793	1566
2016	(1)	3943	1520
324	(1)	4318	1871
203	(1)	4371	1989
2050	(2)	4419	1846
264	(1)	4432	1923
2242	(1)	4802	1967
7	(1)	5132	1964
2212	(1)	5148	2108
2302	(1)	5708	2367
410	(2)	5715	2145
1505	(1)	5865	2330
202	(1)	7143	3052
313	(1)	7275	3005
2224	(1)	7524	2975
1243	(1)	7565	4529
2207	(1)	7584	2976
506	(1)	7882	3126
2217	(1)	9260	3666
2412	(1)	10307	3736
2124	(1)	10460	5185
204	(1)	10795	4418
1045	(1)	17859	7187
2416	(1)	22847	8066
1043	(1)	45477	17411
2223	(1)	52549	19174
2313	(1)	83618	31529
2227	(1)	220142	77257

gentle bend uncertain-length paths

I traced these paths until they got too far from the origin or exceeded 125000 segments in length. The worm leaves the origin going E, which is +x; NE is +y. My program can follow a worm only to +255 or -256. At the length shown, all (including sharp bend uncertains) except rule codes 104 and 105 had returned to the origin twice.

code	length	nodes	X	Y
100	143031	77535	75	-256
101	143025	77531	-76	255
104	109138	58460	-3	255
105	109138	58460	-3	255
120	204894	127041	-8	-199
121	205017	127123	37	172
124	199490	127192	35	188
2104	113361	59357	-127	255
2105	113554	59551	-127	255
2205	252410	87989	-186	255
2222	345046	119754	-256	101
2322	225302	78892	255	-85
2327	157588	55279	-256	43

gentle bend infinite paths

Category, smallest rule code, number of rule codes creating it, and description are given.

DOUBLE-HEXAGON ZIPPERS
2460 (4) UNEVEN START
2462 (2) EVEN START
HEXAGONAL (ZERO POINTED STAR) SPIRALS
200 (18) CHEVRON POINTING AWAY FROM CENTER, HEX CENTER
207 (9) LIKE 200, BUT NO HEX IN CENTER
102 (18) CHEVRON POINTING OBLIQUE TO PROPAGATION, HEX CENTER
106 (18) LIKE 102, BUT TADPOLE CENTER
2110 (2) CHEVRON OBLIQUE, HEX AND CRUD AT CENTER
2120 (2) CHEVRON OBLIQUE, WEIRD CENTER
1124 (1) CHEVRON OBLIQUE, MUCH CRUD AT CENTER
1104 (2) WEIRD CENTER, KINK ON ONE DIAGONAL
2100 (2) WEIRD CENTER, KINKS ON TWO 120-DEGREE RADII
2042 (5) PERIOD 3 EDGE, PERIOD 6 SPIRAL FROM CENTER
DIAMOND (TWO POINTED STAR) SPIRALS
440 (15) ONE MAJOR RADIUS DIAMONDS, OTHER HEXAGONS
2442 (1) ONE MAJOR RADIUS SINGLE HEX, OTHER DOUBLE IN LINE
2444 (1) ONE MAJOR RADIUS SINGLE HEX, OTHER DOUBLE OFFSET
THREE POINTED FROB SPIRAL
444 (1) SINGLE HEX ROW DOWN EACH POINT
ARROW (FOUR POINTED STAR) SPIRALS
42 (4) SINGLE HEX ROW DOWN EACH POINT
43 (2) TWO HEX ROW DOWN ONE POINT, SINGLE DOWN OTHERS
SIX-POINTED STAR SPIRALS
442 (1) ONE HEX ROW DOWN EACH POINT
242 (1) 1, 1, 1, 1, 1, 2 HEX ROWS, WEIRD CENTER
40 (3) TRIPLE HEX ROW ON ONE POINT
245 (1) 1, 1, 1, 2, 2, 2 HEX ROWS, WEIRD CENTER
243 (1) LIKE 245, BUT DIFFERENT WEIRD CENTER
WANKEL SPIRALS
12 (6)
BRICK BUILDING OR SHOWER ROOM CORNER SPIRALS
56 (6) SINGLE HEXAGON AT CENTER
52 (18) TWO HEXAGONS AT CENTER
450 (9) ONE HEXAGON, ONE LOOP AT CENTER
1410 (4) TWO LOOPS, ONE LINE
413 (2) SEVERAL LOOPS FROM CENTER
2257 (1) OH MY GOSH, DIAGONALS TOO
SHOOT GROWERS
262 (1) SHOOT AT ABOUT LENGTH 12000
263 (1) SHOOT AT ABOUT LENGTH 2300
302 (1) SHOOT AT ABOUT LENGTH 1800
303 (1) SHOOT AT ABOUT LENGTH 250
2000 (1) SHOOT AT ABOUT LENGTH 480
2007 (1) SHOOT AT ABOUT LENGTH 440

sharp bend

Each category is listed as in the gentle bend section above.

code		L	N
4000	(162)	9	7
4022	(81)	12	8
4021	(54)	15	9
4002	(81)	18	12
4027	(27)	21	12
6001	(18)	27	16
4001	(18)	30	18
5041	(6)	37	21
5101	(6)	45	25
6007	(9)	48	24
4007	(9)	63	33
5047	(3)	68	36
5107	(1)	411	181
5307	(2)	438	181
5201	(2)	609	246
5001	(2)	615	273
5007	(1)	2373	990
5207	(1)	2478	915

code	length	nodes	X	Y
5401	85614	29241	-83	255
5405	183866	62247	-229	255
5407	137570	46794	-256	87

4011 (108) ZIPPER WITH UNEVEN START
4016 (54) ZIPPER WITH EVEN START

PATH DIAGRAMS

The computer plots of each of the unique paths follow the same sequence as used in the MASTER TABLE above. Uncertain-length and infinite paths are shown at a length of 2048 segments, except for zippers and shoot growers, which are shown at lengths appropriate to their complexity. Both in the MASTER TABLE and in the plots, there are some pairs of uncertain-length paths which differ by a mere rotation. They are nevertheless both given for completeness. They are: 100 = 101; 120 = 121; and 5401 = 5405. 104 may = 105, but it is not proven, since a second return to the origin could cause distinction. 2104 and 2105 are similar, but unique after their second return at a length of 403 segments.

HOW TO LOCATE PATH FOR CODE C

(1) Look in the CROSS-REFERENCE LIST. Chances are about 1 in 4 that it is there. If so, see MASTER TABLE for more details and to get an idea of how far through the plots it appears.

(2) It may be one of a code - and - code+1 pair; try locating C-1 in the CROSS-REFERENCE LIST. If it's there, and the MASTER TABLE says there are two codes giving the same path, and it's not one of the unusual pairs listed under NON-UNIQUE PATHS, then you've got it.

(3) If it is one of those listed under NON-UNIQUE PATHS (like 1264 or 1512), then use the code it's paired with.

(4) Now you're in for some work. Look through the large table of general forms of rule codes in the NON-UNIQUE PATHS section. See which form matches C, and then use the corresponding "least rule code."

NON-PATH-CROSSING WORMS

Mike Paterson wondered what the effect would be of requiring that a worm never cross its path. One way to interpret this question is to ask which of the worms discussed herein do not cross their path. Manual checking seems to quickly produce the following seven rule codes (and their equivalents). Actually, the last five of these do cross their path with the very last segment, as they return to the origin for the third time.

52
4000
4022
4002
4027
6007
4007

CROSS-REFERENCE LIST

CODE	LENGTH	CODE	...								
0	897	263	SHOOT	1043	45477	1462	58	2267	553
2	159	264	4432	1044	46	1500	52	2302	5708
3	732	265	684	1045	17859	1502	132	2303	2754
4	1660	302	SHOOT	1046	2578	1503	136	2304	1550
5	970	303	SHOOT	1050	496	1504	398	2307	2565
6	345	304	130	1062	708	1505	5865	2312	1574
7	5132	305	196	1063	180	1506	45	2313	83618
10	73	312	843	1064	53	1507	561	2317	731
12	WANKEL	313	7275	1065	67	1513	152	2322	UNCERT
14	28	322	606	1066	92	2000	SHOOT	2325	327
16	201	323	356	1100	34	2002	69	2327	UNCERT
17	354	324	4318	1104	HEXSPI	2003	196	2400	50
20	81	325	475	1124	HEXSPI	2004	54	2402	140
22	33	400	248	1200	44	2005	296	2403	55
23	50	402	113	1202	206	2006	711	2405	289
24	83	403	71	1203	130	2007	SHOOT	2406	77
25	105	405	97	1204	451	2010	90	2407	151
26	309	406	609	1207	236	2016	3943	2410	73
27	110	407	42	1212	242	2017	304	2412	10307
40	STAR	410	5715	1213	55	2020	83	2413	1672
42	ARROW	412	165	1217	101	2024	85	2416	22847
43	ARROW	413	BRICKS	1222	636	2025	295	2442	DIAMND
50	3566	420	29	1223	62	2026	247	2444	DIAMND
52	BRICKS	423	33	1224	67	2040	50	2460	ZIPPER
56	BRICKS	424	48	1225	1932	2042	HEXSPI	2462	ZIPPER
60	90	440	DIAMND	1227	248	2047	415	2500	219
62	107	442	STAR	1242	138	2050	4419	2502	247
63	294	444	3POINT	1243	7565	2062	354	2503	148
64	43	447	33	1245	73	2063	194	2504	118
65	196	450	BRICKS	1247	1116	2064	66	2505	631
66	735	462	114	1257	738	2065	68	2506	105
67	57	500	3793	1262	321	2066	176	2507	2526
100	UNCERT	502	113	1263	245	2100	HEXSPI	2510	79
101	UNCERT	503	393	1265	99	2104	UNCERT	2512	2377
102	HEXSPI	504	2056	1267	514	2105	UNCERT	2513	623
104	UNCERT	505	747	1302	338	2110	HEXSPI	2516	281
105	UNCERT	506	7882	1303	126	2120	HEXSPI	4000	9
106	HEXSPI	507	2857	1304	153	2124	10460	4001	30
110	284	510	46	1305	122	2202	1632	4002	18
120	UNCERT	512	1742	1307	162	2204	1545	4007	63
121	UNCERT	513	176	1312	286	2205	UNCERT	4011	ZIPPER
124	UNCERT	516	62	1313	262	2207	7584	4016	ZIPPER
125	86	520	23	1317	86	2212	5148	4021	15
200	HEXSPI	525	99	1322	287	2213	1978	4022	12
202	7143	1000	285	1323	533	2217	9260	4027	21
203	4371	1002	45	1324	93	2222	UNCERT	5001	615
204	10795	1003	78	1325	61	2223	52549	5007	2373
205	1020	1004	71	1327	1063	2224	7524	5041	37
206	48	1005	102	1400	39	2225	2155	5047	68
207	HEXSPI	1006	839	1402	119	2227	220142	5101	45
212	306	1007	1515	1403	44	2242	4802	5107	411
213	454	1010	697	1405	110	2243	558	5201	609
222	1564	1016	237	1406	114	2244	330	5207	2478
223	419	1017	631	1407	53	2245	347	5307	438
224	203	1020	94	1410	BRICKS	2247	173	5401	UNCERT
225	235	1024	96	1412	112	2257	BRICKS	5405	UNCERT
242	STAR	1025	67	1413	134	2262	663	5407	UNCERT
243	STAR	1026	181	1442	1457	2263	2811	6001	27
245	STAR	1040	1688	1444	212	2264	534	6007	48
262	SHOOT	1042	512	1460	35	2265	1831		