Guidelines for the IDR1 program

IDR stands for 'Iterative Digital Reversion'

(IDR1*f* ver.1.0)

'f' for 'final'

Compiled by Y. Almirantis. Email address: yalmir@bio.demokritos.gr

::

1. Introduction – Generalities

Books and web-sites introductory to the 'strange and seductive world' of mathematics (see e.g., ref. 1, 2), often refer to the following property of integers as to a mathematical curiosity: "*if you revert any three-figure number (as far as its first and last figures differ by 2 or more), subtract the smaller from the larger in this couple, and then revert again the obtained number and add, the result will always be 1089*". A generalization of this property for input-numbers of any digital length has been presented by R. Webster (ref. 3) in 1995.
Inspired by the above '1089 trick', we have defined an iterative procedure we named 'Iterative Digital Reversion' (IDR). This procedure may be described as follows: ***Let N_0 an integer (input-) number in its decimal form: N_0 = abc ••• fgh. We subtract or add to N_0 its digital reverse $N_0^{(rev)}$ = hgf ••• cba, provided that $N_0^{(rev)} < N_0$ or $N_0^{(rev)} \geq N_0$ respectively. Thus, we obtain N_1, and so on, for as many steps as we desire***.

The 'IDR1f' program we present herein serves the computational investigation of the above iterative procedure. The code is available in FORTRAN, and any interested people can extend it into any further directions. A related research paper is going to appear soon (ref. 4). As we discuss therein, part of the interest in this simple procedure lies on its capability to produce complex numerical outcomes, some of which present similarities with regimes found in non-linear dynamical systems, particularly in discrete dynamical systems (e.g. cellular automata). Such regimes include: periodicities, existence of 'attractors', emergence of explosive outcomes (i.e. unlimited growth), high sensitivity to the initial condition, etc. The program presented herein is developed in order to systematically explore the properties of Iterative Digital Reversion as an open-ended procedure and for the classification and characterization of the emerging regimes.

2. The functionality of 'IDR1f' program in brief – Files that are created and used

Starting, IDR1*f* program opens the data-file **'IDR_dat.txt'** whose content and functions are explained in the next section. An output-file **'IDR_out.txt'** is also opened, where the results of the 'Iterative Digital Reversion' for the processed input-numbers are printed. The compactness of this output varies, and it is specified by parameters in lines 2 & 3 of the data-file. At the beginning of the output file information about the data file parameter choices that have been used is given, and in its last lines some statistics for the results is provided.

Three files with data related to 'attractors' (IDR -invariable sets of numbers where the IDR procedure often ends) are opened: **(i)** the read-only file **'IDR_attrs_old.txt'** where data about attractors found in previous runs have been stored to be used in the present run; **(ii)** the **'IDR_attrs_reused.txt'** file where the old attractors re-found

in the present run are going to be tabulated; and **(iii)** the **'IDR_attrs_new.txt'** where new attractors, found only in this last data-set and absent from the list of 'old attractors', are to be stored.

The appearance of the general motif '10999…8…000(99)' which is the 'signature' of unbounded growth that appears in increasing numbers for higher values of input-numbers is crucial, and part of the present program serves the monitoring of its appearance. The step where this motif is fully established is registered and it is printed in the one-line reports per input-number in the out file, as the 'Step where "109…" appears:' (if such one-line reports are printed, depending on data-file parameter values). Related information appears in the statistics at the end of the out file as: 'Mean Step Nr till establishing the "109…" pattern', and in the next line, as the Global Max. of (number of) steps, and finally, in a list of 'local' maximal values of steps in the tape-file as well.

In the long train of digits (i.e. the IDR output-numbers) produced in the exploding / steadily-growing cases, there is often more structure and variability than the simple appearance of the '10999…8…000(99)' motif. In fact, this pattern appears usually as simple, but also as double, triple, etc., although these multiple appearances are somewhat distorted. Five files where data related to 'explosive behavior' (endless growth followed by the numbers produced after IDR in cases where no attractor is reached) are opened. They are: **IDR_explodingx1/ x2/x3/x4/xx.txt**. There, the last computed IDR product in cases of 'unbounded growth' along with the input-number are printed. This information already appears in the out-file, but in these files, data are separated according to the digital composition of the outcome pattern: digits: **1, 2, 3** in the file-names stand for the number of repeated patterns '10999…8…000' found in the final digital train, and more specifically, they provide the number of 1s and 8s found in the digital sequence. In **'IDR_explodingx4.txt'** are stored all cases with 4 or more such repetitions. Finally, **'IDR_explodingxx.txt'**, is designed to host cases where the rule for the digital composition of the growing integer, i.e. that: "only nines, zeros, and equal numbers of eights and ones are met" is violated. Note that only cases hosted in **x1, x2, x3, and rarely x4** are encountered, i.e. *cases with unequal numbers of 1s and 8s or including digits different than 0, 1, (DD-2) and (DD-1) are transient: after the outcome have reached a sufficient number of digits are never found to be stable outcomes of IDR*.

The digital length of the output number plays a crucial role, as it is needed to be large enough, so that the pattern of digits reaches its 'equilibrium' state. This digital length is determined by the 1st parameter of the 7th line of the data file (DigLen). When this parameter is set to 0 the expansion of the output number of IDR is stopped immediately after the motif '10999…000(99)' is detected. This drives to shorter execution times but it risks to fail to correctly assign to all input-numbers the correct number of repetitions of the pattern '10999…8…000…' (and thus to put them in the correct **'IDR_exploding….txt'** file). In some cases, a zero or low DigLen value leads to a non-empty **'IDR_explodingxx.txt'** which however, when we repeat the run for high enough DigLen values, it always gets empty. E.g. for large sets of 25-digit input numbers, a choice of DigLen = 30 leads to a tiny percent (< 1‰) of false assignment of pattern-type and to a non-empty **'IDR_explodingxx.txt'**, picture which however gets corrected at DigLen ~ 45 and remains invariable for any higher DigLen value. The 2nd integer in line 7, if non-zero, provides a limit so that if an input-nr produced a transient number (within the trail to the attractor) of digits less or equal to this limit, this input-nr is stored in the file '**IDR_stored_LDL_IN.txt**'.

In the file named **'IDR_tape.txt'** some of the information already included in the previously mentioned files is repeated for archiving purposes, while its first lines repeat the parameter lines of the data-file. Additionally, here are included two (interweaved) lists of samples of input-numbers that follow particularly long paths before reaching either the corresponding attractor or the unlimited-growth motif. The global maxima for both these groups are also included at the bottom lines of the 'tape' and 'out' files. With these lists are also interweaved the lists of maximal and minimal values of digits met within the trail of transient to the attractor IDR numbers.

3. Data-file content and function

In its first 7 lines integers are provided using free format. These parameters is explained in what follows:

In this column are provided examples of parameter values.

10 ← **[line 1] determination of the 'numeral system' (DD: decimal, binary, etc). Digits for 10, 11, 12, …, 16 are: a, b, c, …, g.**

2 0 ← **[line 2] 1st par: '2': for each input-number appears only one line of results; '0': additionally, the input-number (in decimal) appears; '1': consecutive steps in the computation for each inp-number also appear. 2nd par: '1 0' (the usual choice) 1-line-report for each input-number appears in the out file; '2 1' (for huge data, ALWAYS WITH 1st No = '2') The one-line report disappears from out and explx1 files (but it still appears for the eleven first input-numbers)[*].**

0 ← **[line 3] Attractor will be omitted: '0'; or it will appear: '1'.**

1 6 0 ← **[line 4] Here is specified how the program is fed with inp. numbers: [A] a list of integers, read from the data-file, below the parameters, serves as the input-number set, when and only when '0 0 0' are given in line 4;**
[B] sequential data numbers. To this end, …000 & …001 must be the two input-nrs (lines), just below the 7 parameter-lines. The 4th line 1st nr must be '1' and the 2nd will be the '$\log_{DD}$' of the higher input-nr to be computed (DD stands for the base of the numeral system);
[C] random input-numbers. In order to activate this, 1st number in the 4th line must be negative. Then, its absolute value provides the number of digits of the generated random input-numbers. The 2nd number of the 4th line provides the total of random numbers to be produced. The 3rd nr of the 4th line is '0' for using always the same seq of random numbers or ≠ 0 for using in every run another set. Important notice: The 8th line data-nr if ≠ 0 is added to all random nrs here, thus allowing sampling in a region of integers. Useful for around changes of order (see example on page 10).

0 ← **[line 5] 'L_IEXEC' if non-zero provides the total population of input-numbers. '0' disactivates this function. Note: 'L_IEXEC' is in the decimal numeral system.**

1000 ← **[line 6]: additive factor for the IDs of the new_attractors. Its use is going to be explained later.**

35 7 ← **[line 7]: [A] number of digits when the computation of the output number stops (DigLen); useful in the study of the explosive scenario. As already mentioned, DigLen = 0 leads to faster execution times, while values ~ 50 lead, for input-numbers of digital length ~ 25, to a safe estimation of the digital pattern of all IDR output numbers. Note that when a low enough DigLen (≠ 0) is selected through the data file, internal readjustment occurs depending on the digital length of the used input-numbers, as mentioned in the 'tape' file. [B] The second integer in the 7th line, if non-zero, activates storage of input-numbers passing from a transient number (in the way to attractor) of length less or equal to that integer in file 'IDR_stored_LDL_IN.txt'.**

[*] When in line 2: '2 1' is chosen, no report lines per input-number in the '…out…' and the '…exp…x1' files are printed. However, '…explodingx2 - xx' files are allowed to be filled in, as inscriptions there are expected to be rare or null.

Typically, the next lines of the data-file read as follows:

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000001
00000 HERE A THIRD LINE IS PROVIDED EQ-TO-0 OR WITH A TERMINAL INPUT-NR 00000
```

If we want the program to be used with a list of specific input-numbers [line 4: '0 0 0'], these have to be given in the next lines (after the 10th) using the same format: 77 digits in each line, with zeros in the first positions, which we call '*IDR format*' and is always in the proper numeral system (see above).

If we want the program to proceed with all integers from 1 to, say, 10000, line 4 parameters must be '1 4 0' and the corresponding data-lines 8th & 9th must be:

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000001
```

Special use of the 8th data-file line: It is important to note here that if we want consecutive integers as input-numbers ranging from a given integer up to a power of 10, we are able to do so by choosing the first input-number line to represent this integer. So, if we want a run for all 4-digit integers, line 4 must be: '1 4 0', and the 8th data-file line must be: … `000001000`.

Note also that the "`TERMINAL INPUT-NR IN IDR FORMAT`" in the 10th data-line is only considered by the program if data-numbers are given in the 'consecutive' way (1st parameter of the 4th data-line = 1), and if the number if the 10th data-line is not zero.

During the program execution, the 'new attractors' are announced, numbered and the input-number of the first appearance of each are shown in the *IDR format* *on screen*.

4. Examples of several runs. The resulting files and some explanations of the tabulated data

SPECIAL NOTE: *In the outputs shown here, minimum numbers of digits of the numbers within the trail towards the attractor or to the emergence of the ever-growing pattern (transient numbers) are not printed (are from a previous version of the program). This information is provided only in the last two examples (pages 11-13).*

(i) Running IDR1 with detailed tabulation of the path to the 'attractor'

Here is the used data-file (the program is fed with a list of input-numbers):

```
10
 1    0
 1
 0    0      0
 0
 0
30    0
   000000000000000000000000000000000000000000000000000000000000000000000000000000000
   000000000000000000000000000000000000000000000000000000000000000000000000000000001
   000000000000000000000000000000000000000000000000000000000000000000000000000000000
   000000000000000000000000000000000000000000000000000000000000000000000000000000104
   000000000000000000000000000000000000000000000000000000000000000000000000000000154
   000000000000000000000000000000000000000000000000000000000000000000000000000000158
   000000000000000000000000000000000000000000000000000000000000000000000000000001824
```

Excerpt of the results (for only three of the input-numbers) follows:

```
  ITERATIVE DIGITAL REVERSION IDR1 - ver.1.0
  Number system:              10
  IDR steps will appear
  Attractor will appear
  Input will be a list of integers
  No max execution counter will be used
  No additive factor for the IDs of the NEW attractors registry: Kadd = 0
  Outcome printed at digital length ~   30
.............................................
Input-Nr = 000000000000000000000000104    (Decimal form:       104)
  Sequence of steps:
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000000505 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000001010 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000000000909 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000001818 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000009999 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000019998 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000109989 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000001099890 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000000109989 :

  Attractor is the loop:
 * 000000000000000000000000000000000000000000000000000000000000000000000000000000109989      1
 * 000000000000000000000000000000000000000000000000000000000000000000000000000001099890      2
 Input-Nr = 000000000000000000000000104 | Attr-STATE #201 | Attr ID:  1 | Steps till =  7
| Attr-length =  2 | Max digitallength =  7 | at step =  9
.....................................................................................
Input-Nr = 000000000000000000000000158    (Decimal form:       158)
  Sequence of steps:
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000001009 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000010010 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000000009009 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000018018 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000099099 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000000198198 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000001090089 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000010890990 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000000981189 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000001962378 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000010695069 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000106754670 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000030297069 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000126376272 :
 + 000000000000000000000000000000000000000000000000000000000000000000000000000399049893 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000000108900 :
 - 000000000000000000000000000000000000000000000000000000000000000000000000000000099099 :

  Attractor is the loop:
 * 000000000000000000000000000000000000000000000000000000000000000000000000000000099099      1
 * 000000000000000000000000000000000000000000000000000000000000000000000000000000198198      2
 * 000000000000000000000000000000000000000000000000000000000000000000000000000001090089      3
 * 000000000000000000000000000000000000000000000000000000000000000000000000000010890990      4
 * 000000000000000000000000000000000000000000000000000000000000000000000000000000981189      5
 * 000000000000000000000000000000000000000000000000000000000000000000000000000001962378      6
 * 000000000000000000000000000000000000000000000000000000000000000000000000000010695069      7
 * 000000000000000000000000000000000000000000000000000000000000000000000000000106754670      8
 * 000000000000000000000000000000000000000000000000000000000000000000000000000030297069      9
```

```
 * 000000000000000000000000000000000000000000000000000000000126376272     10
 * 000000000000000000000000000000000000000000000000000000000399049893     11
 * 000000000000000000000000000000000000000000000000000000000108900     12
 Input-Nr = 000000000000000000000158 | Attr-STATE #201 | Attr ID:  3 | Steps till =  5
| Attr-length = 12 | Max digitallength =  9 | at step = 13
.....
Input-Nr = 000000000000000000001824    (Decimal form:      1824)
  Sequence of steps:
 + 000000000000000000000000000000000000000000000000000000000006105 :
 - 000000000000000000000000000000000000000000000000000000000001089 :
 + 000000000000000000000000000000000000000000000000000000000010890 :
 - 000000000000000000000000000000000000000000000000000000000001089 :

  Attractor is the loop:
 * 000000000000000000000000000000000000000000000000000000000001089      1
 * 000000000000000000000000000000000000000000000000000000000010890      2
 Input-Nr = 000000000000000000001824 | Attr-STATE #202 | Attr ID:  2 | Steps till =  2
| Attr-length =  2 | Max dig. length =  5 | at step =  4
 .....
 Totally processed input-numbers =           4
 Mean Step Nr till Attr. =   4.500000    Nr of att/ted cases =           4
 Global Max steps till Attr.=           7 (for the input-number see the "tape")
 Predicted Exploding (109...& +1dig/2-steps) Tot.nr of cases =           0
 Mean Step Nr till PrExpl=  0.0000000E+00
 Glob.Max steps till PrExpl.=           0 (for the input-number see the "tape")
 Undecided (no attractor after 1000 steps)   Tot.nr of cases =           0
 Predicted Exploding (expl. motifx1) Tot.nr.cases =           0
 Predicted Exploding (expl. motifx2) Tot.nr.cases =           0
 Predicted Exploding (expl. motifx3) Tot.nr.cases =           0
 Predicted Exploding (expl.motive>3) Tot.nr.cases =           0
 Pred.Expl.(motif non-typical:nrs of 8s=/1s)T.n.cs=           0
 All met (reused & new) att/tors =           3

-------------------------------------------------
COMMENTS ON THE TABULATED QUANTITIES:
The number to compute its IDR is given (i) in the IDR-format, in its proper numeral
system;  (ii) as INTEGER*4 in the decimal numeral system
Attr-STATE. This takes four values [in the updated IDRf instead of Attr-STATE only # is written]:
   (i)   101 means 1st appearance of an old/known attractor (included in:
         IDR_attrs_old.txt file)
   (ii)  102 means an old/known attractor already met (included in: IDR_attrs_old.txt)
   (iii) 201 means 1st appearance of a new (that appears for a first time) attractor
   (iv)  202 means a new attractor that reappears
Attr/r's length.  How many numbers compose the loop which, when reached, closes and
repeats itself endlessly.
Attr ID. An ID number is assigned to any attractor. Typically, they start from 1 for
[1089 - 10890] and increase for increasing input-numbers. Then, when a collection of
consecutive input-numbers (usually, up to 10^7 or higher) has produced a first 'old-
attractors' file, then, in any next execution (often for samples of random input-
numbers), the line-6 additive factor in the data-file is suitably adjusted (values like:
100, 1000 etc), in order to avoid confusion.
Steps till (reaching the attractor). Number of steps between the input-number and the
first member of the attractor.
Max Min digital length. Of any number in the way to the attractor (including members of
the attractor itself).
at step The step where this max or min digital length is observed.
```

(ii) Running IDR1 for blocks of consecutive integers – Emphasis here is given to the management of the obtained attractors and attractive behavior

Here is the used data-file:

```
10
 2  0
 0
 1  5   0
 0
 0
15
   000000000000000000000000000000000000000000000000000000000000000000000000000000
   000000000000000000000000000000000000000000000000000000000000000000000000000001
   000000000000000000000000000000000000000000000000000000000000000000000000000000
```

The resulting out-file follows (with the data for almost all intermediate input-numbers eliminated):

```
  ITERATIVE DIGITAL REVERSION IDR1 - ver.1.0
  Number system:             10
  IDR steps and initial number will be omitted
  Attractor will be omitted
  IDR computed from 1 to          10    exp           5
  No max execution counter will be used
  No additive factor for the IDs of the NEW attractors registry: Kadd = 0
  Outcome printed at digital length ~          15
  ..............................................
 Input-Nr = 000000000000000000000000001 | Attr-STATE #201 | Attr ID:  1 | Steps till = 10
| Attr-length =  2 | Max dig. length =  5 | At step = 12
 Input-Nr = 000000000000000000000000002 | Attr-STATE #202 | Attr ID:  1 | Steps till =  9
| Attr-length =  2 | Max dig. length =  5 | At step = 11
 Input-Nr = 000000000000000000000000003 | Attr-STATE #202 | Attr ID:  1 | Steps till = 10
| Attr-length =  2 | Max dig. length =  5 | At step = 12
 Input-Nr = 000000000000000000000000004 | Attr-STATE #202 | Attr ID:  1 | Steps till =  8
| Attr-length =  2 | Max dig. length =  5 | At step = 10
 Input-Nr = 000000000000000000000000005 | Attr-STATE #202 | Attr ID:  1 | Steps till =  6
| Attr-length =  2 | Max dig. length =  5 | At step =  8
.....
Input-Nr = 000000000000000000000012578 | Step w/re"109.."appears:  15 @
000000000000000000000000000000000000000000000000000000000010999909989000099
.....
 Input-Nr = 000000000000000000000099995 | Attr-STATE #202 | Attr ID:  3 | Steps till =  2
| Attr-length =  2 | Max dig. length =  7 | At step =  4
 Input-Nr = 000000000000000000000099996 | Attr-STATE #202 | Attr ID:  3 | Steps till =  2
| Attr-length =  2 | Max dig. length =  7 | At step =  4
 Input-Nr = 000000000000000000000099997 | Attr-STATE #202 | Attr ID:  3 | Steps till =  2
| Attr-length =  2 | Max dig. length =  7 | At step =  4
 Input-Nr = 000000000000000000000099998 | Attr-STATE #202 | Attr ID:  3 | Steps till =  3
| Attr-length =  2 | Max dig. length =  7 | At step =  5
 Input-Nr = 000000000000000000000099999 | Attr-STATE #202 | Attr ID:  5 | Steps till =  2
| Attr-length =  2 | Max dig. length =  8 | At step =  4
 ..............................................
 Totally processed input-numbers =      100000
 Mean Step Nr till Attr. =   3.640635    Nr of at/cted cases =       99509
 Global Max steps till Attr.=          29  (for the input-number see the "tape")
 Predicted Exploding (109...& +1dig/2-steps) Tot.nr of cases =         490
 Mean Step Nr till establish. the  >>  pattern =    21.61225
 Glob.Max steps till establ.  the  >>  pattern =          40 (see the "tape")
```

```
Undecided (no attractor after 1000 steps)   Tot.nr of cases =          0
Predicted Exploding (expl. motifx1) Tot.nr.cases =         490
Predicted Exploding (expl. motifx2) Tot.nr.cases =           0
Predicted Exploding (expl. motifx3) Tot.nr.cases =           0
Predicted Exploding (expl.motive>3) Tot.nr.cases =           0
Pred.Expl.(motif non-typical:nrs of 8s=/1s)T.n.cs=           0
All met (reused & new) att/tors =          13
```

We see that when we run IDR1 up to 10^5,[4th line:'1 5'] we meet 490 'exploding' cases.

SOME COMMENTS ON THE OUT-FILE ANNOTATION:

The 'Mean Step Nr till Attr.', given in the last lines of the file, averages over the input-numbers leading to an attractor (excluding exploding ones). Its invariably low value is in an apparent contradiction with the very few natural numbers which compose the invariant sets (traps) of the IDR procedure. This means that these numerical traps (usually consisting of two members) deserve the name 'attractors', as they exert an influence to the iterative digital reversion of other numbers which converge and quickly reach them. Thus these 'attractors' become ending points of many other numbers' paths.

A similar 'attractivity' characterizes also the path towards explosive behavior. There again, it seems that relatively few integers (the ones stored in the files 'IDR_exploding….txt') also act on their surroundings as points of convergence.

COLLECTIONS OF 'ATTRACTORS'

When we run IDR1 for input-numbers 1-10^5, the following 'IDR_attrs_new.txt' file is produced:

```
   .........0000000000001089     2     1     .........0000000000000001 20225
   .........0000000000010989     2     2     .........0000000000000101 26347
   .........0000000000109989     2     3     .........0000000000000104 26679
   .........0000000000099099    12     4     .........0000000000000158 21500
   .........0000000001099989     2     5     .........0000000000001018  2950
   .........0000000010891089     2     6     .........0000000000010001   723
   .........0000000010999989     2     7     .........0000000000010016   794
   .........0000000109999989     2     8     .........0000000000010106    26
   .........0000000108901089     2     9     .........0000000000010158   150
   .........0001098900010989     2    10     .........0000000000010208     5
   .........0000001098910989     2    11     .........0000000000010298    67
   .........0000001089001089     2    12     .........0000000000010306    31
   .........0000010999999989     2    13     .........0000000000090339    12
```

Note that in the ''IDR_attrs…' files each attractor is represented by one line, which includes only one member of this attractor (full digital representation of one of the numbers which constitute it) and four integers: (i) the number of its members; (ii) its ID; (iii) the first input-number which reached this attractor (in a given run) in the IDR format; (iv) the total of occurrences of this attractor in this data-set. Note that if the attractor has two members, the one is produced by the other via a shift of its digits by one position adding a last zero, i.e. multiplication by the base of the numeral system. In all other cases (which represent a minority of incidences) the attractor is considered as non-standard, and its number of digits always is > 2 (see ref. 4).

Note that the above collection of attractors of the first 100,000 integers includes one attractor with 12 digits. In the decimal and in other numeral systems we meet sporadically such simple non-standard attractor. Other 12-member cases are also met, which however are composite forms of the above (such composite forms occur for any attractor): repetitions of the same digits with intervening zeros. E.g. 9909900099099, or 990990000099099.

In a next run of the program, we first copy the above file, the 'IDR_attrs_new.txt' to 'IDR_attrs_old.txt', and we modify the 4th line of the data file, from "1 5" to "1 6", while the 8th line instead of "……0000000" becomes

"……..0100000" (i.e. we ran the program for all 6-digit data numbers) and we modify the additive parameter of line 6 from zero to "1000". The resulting full set of attractors for 6-digit input-numbers as registered is:

```
 ..................... the above 13 attractors plus 4 more .....................
   .........0000001089001089     2    12
   .........0000010999999989     2    13
   .........0000010890001089     2  1001     .........0000000000100106   288
   .........0000109989109989     2  1002     .........0000000000100208   217
   .........0000108900001089     2  1003     .........0000000000100428   121
   .........0000001099999989     2  1004     .........0000000000100858   277
```

(iii) Running IDR1 for randomly chosen integers

This mode of investigating the IDR phenomenology is particularly useful for increasing values of data numbers (i.e. for higher digital lengths), where, practically, we are able to work only through sampling.

A concrete example of randomly sampled 104 17-digit input-numbers, making the standard choice of random seed (the one repeated the same each time), and again skipping most input-nr lines in the out file, follows:

```
  ITERATIVE DIGITAL REVERSION IDR0 - ver.1.0
  Number system:             10
  IDR steps and initial number will be omitted
  Attractor will be omitted
  Input will be       10000   rndm nrs of digital length:          17
  - standard seed
  No max execution counter will be used
  No additive factor for the IDs of the NEW attractors registry: Kadd = 0
  Outcome printed at digital length ~          55
 ..........................................................................
Input-Nr = 00000000003778083879389630 0 | Step w/re"109.."appears:  15 @00000000000000000
0000109999999999999999999999990089000000000000000000000000099
 Input-Nr = 00000000001079186090950009 0 | Attr-STATE #201 | Attr ID:   1 | Steps till =  15
| Attr-length =   2 | Max dig. length = 18 | At step =  8
 Input-Nr = 00000000007247747590845627 9 | Attr-STATE #201 | Attr ID:   2 | Steps till =   4
| Attr-length =   2 | Max dig. length = 18 | At step =  2
 Input-Nr = 00000000003591514466572327 6 | Attr-STATE #201 | Attr ID:   3 | Steps till =  49
| Attr-length =   2 | Max dig. length = 24 | At step = 51
 Input-Nr = 00000000006277838506284206 5 | Attr-STATE #201 | Attr ID:   4 | Steps till =  17
| Attr-length =   2 | Max dig. length = 17 | At step =  1
 Input-Nr = 00000000006614355591634599 6 | Step w/re"109.."appears:  20 @00000000000000000
0000109999999999999999999999990098900000000000000000000000099
.....
 Input-Nr = 00000000007982150239392640 1 | Step w/re '109..' appears: 50 @  000000000000000
0001099999999999999999999999999989000000000000000000000000000
 Input-Nr = 00000000004111426130733932 9 | Step w/re '109..' appears: 10 @  000000000000000
0010999999999999999999999999999989000000000000000000000000000
 Input-Nr = 00000000003436755827959253 2 | Step w/re '109..' appears: 55 @  000000000000000
0001099999999999999999999999999899900000000000000000000000099
 ..........................................................................
 Totally processed input-numbers =       10000
 Mean Step Nr till Attr. =   13.50232    Total at/cted cases =        4087
 Global Max steps till Attr.=          82 (for the input-number see the "tape")
 Predicted Exploding (109...& +1dig/2-steps) Tot.nr of cases =        5913
 Mean Step Nr till establish. the  >>  pattern =    22.13766
 Glob.Max steps till establ.  the  >>  pattern =         110 (see the "tape")
 Undecided (no attractor after 1000 steps)   Tot.nr of cases =           0
 Predicted Exploding (expl. motifx1) Tot.nr.cases =        5807
```

```
 Predicted Exploding (expl. motifx2) Tot.nr.cases =         106
 Predicted Exploding (expl. motifx3) Tot.nr.cases =           0
 Predicted Exploding (expl.motive>3) Tot.nr.cases =           0
 Pred.Expl.(motif non-typical:nrs of 8s=/1s)T.n.cs=           0
 All met (reused & new) att/tors =         160
 Input-nr in decimal = "999999999" denotes that its value exceeds the 1 billion.
```

(iv) The case of <u>exploring the border between two consecutive digital lengths</u>

```
  ITERATIVE DIGITAL REVERSION IDR1 - ver.1.1                 6/21/2024      ::
  Number system:             10
  IDR steps and initial number will be omitted
  Attractor will be omitted
  Input will be          50  rndm nrs of digital length:           5
  - standard seed
  No max execution counter will be used
  No additive factor for the IDs of the NEW attractors registry: Kadd = 0
  Outcome printed at digital length ~          30
   000000000000000000000000000000000000000000000000000000000000000000000999999950000
   000000000000000000000000000000000000000000000000000000000000000000000000000000001
   000000000000000000000000000000000000000000000000000000000000000000000000000000000
 ..............................................................................
 Input-Nr = 000000000000001000000046300 | Step w/re"109.."appears ..............
 Input-Nr = 000000000000001000000037938 | Attr-STATE #201 | ..................
 Input-Nr = 000000000000001000000028083 | Attr-STATE #201 | ..................
 ------------------------------------------------------------------------------
 Input-Nr = 000000000000001000000028967 | Attr-STATE #201 | ..................
 Input-Nr = 000000000000000999999972369 | Step w/re"109.."appears ..............
 Input-Nr = 000000000000000999999975629 | Step w/re"109.."appears ..............
 Input-Nr = 000000000000001000000014881 | Attr-STATE #201 | .................. ----------
------------------------------------------------------------------------------
 Input-Nr = 000000000000001000000010473 | Attr-STATE #201 | ..................
 Input-Nr = 000000000000000999999992189 | Attr-STATE #201 | ..................
 Input-Nr = 000000000000000999999970978 | Step w/re"109.."appears ..............
 Input-Nr = 000000000000001000000035434 | Step w/re"109.."appears ..............
 ..............................................................................
 Totally processed input-numbers =          50
 Mean Step Nr till Attr. =   7.379310    Total at/cted cases =          29
 Global Max steps till Attr.=          25 (for the input-number see the "tape")
 Predicted Exploding (109...& +1dig/2-steps) Tot.nr of cases =          21
 Mean Step Nr till establish. the  >>  pattern =    7.380952
 Glob.Max steps till establ.  the  >>  pattern =          15 (see the "tape")
 Undecided (no attractor after 1000 steps)   Tot.nr of cases =           0
 Predicted Exploding (expl. motifx1) Tot.nr.cases =          21
 ------ ^ see comment at the end of discussion on data-line-4 on page 3 -------
```

(v) Running IDR1 for number systems other than the decimal

We determine the numeral system to work with by using the first-line parameter of the data file. This can take values 2 – 17, although a simple extension of the program may give us more choices. Note that, input-numbers in decimal format are given for some parameter choices in the first line dedicated to any input-number, provided that its decimal digital length does not exceed the nine digits. If it does, the string '999999999' is printed instead.

We end up with two examples, one from the 7- and one from the 16-numeral system. We have chosen in the first case to use the random choice of 1000 input numbers, and in the second, the same number of input numbers using the +1 choice in the 4th data-file line.

```
  ITERATIVE DIGITAL REVERSION IDR1 - ver.1.0       (f)     6/21/2024      ::
  Number system:              7
  IDR steps and initial number will be omitted
  Attractor will be omitted
  Input will be        1000  rndm nrs of digital length:          17
  - standard seed
  No max execution counter will be used
  No additive factor for the IDs of the NEW attractors registry: Kadd = 0
  Outcome printed at digital length ~          45
   000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
   000000000000000000000000000000000000000000000000000000000000000000000000000000000000001
   000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 ..........................................................................
 Input-Nr = 000000000030010131023126300 | #201 | Attr ID:   1 | Steps till =  21 | Attr-
len.=   2 | Max/Min dig.len.= 21 / 17 | At stepMax/Min = 14 /  1
 Input-Nr   =   000000000010021160202500020   |   Step   where   "109.."   appears:      10
@000000000000000000000000000000010666666666666666666666560000000000000000000
 Input-Nr   =   000000000062400405201456202   |   Step   where   "109.."   appears:      40
@000000000000000000000000000000010666666666666666666666560000000000000000000
 Input-Nr   =   000000000030521514466502320   |   Step   where   "109.."   appears:      10
@000000000000000000000000000000010666666666666666666666560000000000000000000
 Input-Nr = 000000000062001315062142065 | #201 | Attr ID:   2 | Steps till =   5 | Attr-
len.=   2 | Max/Min dig.len.= 17 / 15 | At stepMax/Min =  1 /  6
.....
Input-Nr = 000000000011111004614451444 | #202 | Attr ID:  40 | Steps till =   6 | Attr-
len.=   2 | Max/Min dig.len.= 17 /  9 | At stepMax/Min =  1 /  3
 Input-Nr   =   000000000015022102625226031   |   Step   where   "109.."   appears:      10
@000000000000000000000000000000010666666666666666666666560000000000000000000
 Input-Nr = 000000000035402532153202622 | #202 | Attr ID:   6 | Steps till =  46 | Attr-
len.=   2 | Max/Min dig.len.= 23 / 16 | At stepMax/Min = 43 /  8
 Input-Nr = 000000000045001301656000101 | #202 | Attr ID:  14 | Steps till =  14 | Attr-
len.=   2 | Max/Min dig.len.= 18 /  7 | At stepMax/Min =  3 / 12
 Input-Nr = 000000000010126511052161546 | #201 | Attr ID: 106 | Steps till =  26 | Attr-
len.=   2 | Max/Min dig.len.= 23 / 17 | At stepMax/Min = 28 /  1
 ..........................................................................
 Totally processed input-numbers =        1000
 Mean Step Nr till Attr. =   17.13072    Total at/cted cases =         459
 Global Max steps till Attr.=         119 (for the input-number see the "tape")
 Predicted Exploding (109...& +1dig/2-steps) Tot.nr of cases =         541
 Mean Step Nr till establish. the  >>  pattern =    15.80407
 Glob.Max steps till establ.  the  >>  pattern =         165 (see the "tape")
 Undecided (no attractor after 1000 steps)   Tot.nr of cases =           0
 Predicted Exploding (expl. motifx1) Tot.nr. cases =         534
 Predicted Exploding (expl. motifx2) Tot.nr. cases =           7
 Predicted Exploding (expl. motifx3) Tot.nr. cases =           0
 Predicted Exploding (expl.motive>3) Tot.nr. cases =           0
 Pred.Expl.(motif non-typical:nrs of 8s=/1s)T.n.cs =           0
 All met (reused & new) att/tors =         106
 Input-nr in decimal = "999999999" denotes that its value exceeds the 1 billion
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  ITERATIVE DIGITAL REVERSION IDR1 - ver.1.0       (f)     6/21/2024      ::
  Number system:             16
  IDR steps and initial number will be omitted
  Attractor will be omitted
```

```
  Input will be        1000  rndm nrs of digital length:          7
  - standard seed
  No max execution counter will be used
  No additive factor for the IDs of the NEW attractors registry: Kadd = 0
  Outcome printed at digital length ~          30
   000000000000000000000000000000000000000000000000000000000000000000000000000000
   000000000000000000000000000000000000000000000000000000000000000000000000000001
   000000000000000000000000000000000000000000000000000000000000000000000000000000
 ..........................................................................
 Input-Nr = 0000000000000000000389c966 | #201 | Attr ID:   1 | Steps till =   2 | Attr-
len.=   2 | Max/Min dig.len.=  7 /  5 | At stepMax/Min =  1 /  4
 Input-Nr = 0000000000000000000e0e387f | #201 | Attr ID:   2 | Steps till =   5 | Attr-
len.=   2 | Max/Min dig.len.= 10 /  7 | At stepMax/Min =  7 /  1
 Input-Nr = 0000000000000000000060f6377 | #201 | Attr ID:   3 | Steps till =   8 | Attr-
len.=   2 | Max/Min dig.len.=  7 /  4 | At stepMax/Min =  1 /  7
 Input-Nr   =   0000000000000000000c0f09b6   |   Step   where   "109.."   appears:      10
@00000000000000000000000000000000000000000000010fffffffffffffef00000000000000
 Input-Nr = 0000000000000000000f707f18 | #201 | Attr ID:   4 | Steps till =  10 | Attr-
len.=   2 | Max/Min dig.len.=  9 /  6 | At stepMax/Min =  7 /  4
.....................................................................
Input-Nr = 0000000000000000000006d18d2f | #202 | Attr ID:   3 | Steps till =   3 | Attr-
len.=   2 | Max/Min dig.len.=  8 /  6 | At stepMax/Min =  2 /  5
 Input-Nr = 0000000000000000000f188e60 | #202 | Attr ID:   5 | Steps till =   5 | Attr-
len.=   2 | Max/Min dig.len.=  8 /  7 | At stepMax/Min =  6 /  1
 Input-Nr = 0000000000000000000029b07e2 | #202 | Attr ID:   4 | Steps till =  11 | Attr-
len.=   2 | Max/Min dig.len.=  9 /  6 | At stepMax/Min =  8 /  5
 Input-Nr = 0000000000000000000007786f7a | #202 | Attr ID:   6 | Steps till =   3 | Attr-
len.=   2 | Max/Min dig.len.=  9 /  7 | At stepMax/Min =  5 /  1
 ..........................................................................
 Totally processed input-numbers =        1000
 Mean Step Nr till Attr. =   7.792289    Total at/cted cases =         804
 Global Max steps till Attr.=          44 (for the input-number see the "tape")
 Predicted Exploding (109...& +1dig/2-steps) Tot.nr of cases =         196
 Mean Step Nr till establish. the  >>  pattern =    10.02551
 Glob.Max steps till establ.  the  >>  pattern =          15 (see the "tape")
 Undecided (no attractor after 1000 steps)   Tot.nr of cases =           0
 Predicted Exploding (expl. motifx1) Tot.nr. cases =         196
 Predicted Exploding (expl. motifx2) Tot.nr. cases =           0
 Predicted Exploding (expl. motifx3) Tot.nr. cases =           0
 Predicted Exploding (expl.motive>3) Tot.nr. cases =           0
 Pred.Expl.(motif non-typical:nrs of 8s=/1s)T.n.cs =           0
 All met (reused & new) att/tors =          12
```

In this case we will also present the 'tape' file, as an opportunity to observe the interweaved lists of local maxima of the steps to the attractor and of the steps to the establishment of the exploding pattern and the list of the attractors met.

```
ITERATIVE DIGITAL REVERSION IDR0 - ver.1.0   // data file inscriptions
          16
           2           0
           0
          -7        1000           0
           0
           0
          30      ----->          30        [           0  ]
.....................................................................
 MAX DIGITAL LENGTH T.A.:           7   For INPUT-Nr:  0000000000000000000389c966
 MIN DIGITAL LENGTH T.A.:             5 For INPUT-Nr:  0000000000000000000389c966
 MAX OF STEPS TILL ATTR.:        2      For INPUT-Nr:  0000000000000000000389c966
```

```
MAX DIGITAL LENGTH T.A.:          10   For INPUT-Nr:  0000000000000000000e0e387f
MAX OF STEPS TILL ATTR.:        5      For INPUT-Nr:  0000000000000000000e0e387f
MIN DIGITAL LENGTH T.A.:            4 For INPUT-Nr:  00000000000000000000060f6377
MAX OF STEPS TILL ATTR.:        8      For INPUT-Nr:  00000000000000000000060f6377
MaxSteps till PREDexpl.: 10            For input-nr:  0000000000000000000c0f09b6
MAX OF STEPS TILL ATTR.:       10      For INPUT-Nr:  0000000000000000000f707f18
MAX OF STEPS TILL ATTR.:       15      For INPUT-Nr:  00000000000000000000008836978
MAX DIGITAL LENGTH T.A.:          11   For INPUT-Nr:  000000000000000000006e94e29
MAX OF STEPS TILL ATTR.:       21      For INPUT-Nr:  000000000000000000000a7155d9
MAX OF STEPS TILL ATTR.:       28      For INPUT-Nr:  000000000000000000000e9a0d0e
MAX DIGITAL LENGTH T.A.:          16   For INPUT-Nr:  000000000000000000000f1d458f
MAX OF STEPS TILL ATTR.:       43      For INPUT-Nr:  000000000000000000000f1d458f
MIN DIGITAL LENGTH T.A.:            3 For INPUT-Nr:  000000000000000000008a88979
MAX OF STEPS TILL ATTR.:       44      For INPUT-Nr:  000000000000000000004e17d3b
MaxSteps till PREDexpl.: 15            For input-nr:  000000000000000000003d2ea3d
```

Note here that four kinds of intermediate (local) max/min and the corresponding inp-nrs are printed: ***(i)*** *(local) maximal number of steps till reaching the attractor or the pattern denoting unbound expansion (10999…0000/99) and* ***(ii)*** *(local) max/min values of digital length of members of the sequence of physical numbers (trail) leading to attractor. The last value in all four types of information is the global extremum.*

```
================================================================================
 ////// HERE ARE OMMITTED SOME LINES REPEATING INFORMATION PROVIDED IN THE OUT FILE //////
 ********************** re-used old (already known) attractors ******************
 ***************** new (absent from the old-repository)  attractors *************
  000000000000000000000000000000000000000000000000000000000010fef0    2    1    000000000000000000389c966    32
  000000000000000000000000000000000000000000000000000000000010ef010ef    2    2    0000000000000000000e0e387f    20
  000000000000000000000000000000000000000000000000000000000010ffef    2    3    00000000000000000000060f6377   158
  000000000000000000000000000000000000000000000000000000000010ef10ef    2    4    0000000000000000000f707f18   321
  000000000000000000000000000000000000000000000000000000000010fffef    2    5    00000000000000000000847d47b    56
  000000000000000000000000000000000000000000000000000000000010ffffef    2    6    00000000000000000000d89276d   151
  000000000000000000000000000000000000000000000000000000000010ef0010ef    2    7    000000000000000000006e94e29     5
  000000000000000000000000000000000000000000000000000000000010ef0    2    8    0000000000000000000277e06c    43
  000000000000000000000000000000000000000000000000000000000010ef0000010ef0    2    9    00000000000000000000f1d458f     6
  000000000000000000000000000000000000000000000000000000000010fffffef    2   10    00000000000000000000b0c881f     7
  000000000000000000000000000000000000000000000000000000000010fef10fef    2   11    000000000000000000000076fa196     4
  000000000000000000000000000000000000000000000000000000000010ef000010ef    2   12    00000000000000000000696d0d9     1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

REFERENCES

1. David Acheson. '*1089 and All That: A Journey into Mathematics*'. Oxford University Press, 2002.
2. https://math.hmc.edu/funfacts/magic-1089/
3. Webster, R. (1995). A Combinatorial Problem with a Fibonacci Solution, *The Fibonacci Quarterly* **33**, 26-31.
4. Almirantis Y, Li W. (2024). Extending 1089 attractor to any number of digits and any number of steps. *Preprint*. https://arxiv.org/html/2410.11784v2

::