Acta Cybernetica **25** (2021) 271–284.

Toolset for Supporting the Research of Lattice Based Number Expansions

Péter Hudoba[ab] and Attila Kovács[ac]

Abstract

The world of generalized number systems contains many challenging areas. Computer experiments often support the theoretical research. In this paper we introduce a toolset that helps to analyze some properties of lattice based number expansions. The toolset is able to (1) analyze the expansions, (2) decide the number system property, (3) classify and visualize the periodic points.

The toolset is implemented in Python, published alongside with a database that stores plenty of special expansions, and is able to store the custom properties like signature, operator eigenvalues, etc. Researchers can connect to the server and request/upload data, or perform experiments on them.

We present an introductory usage of the toolset and detail some results that has been observed by the toolset. The toolset can be downloaded from `http://numsys.info` domain.

1 Introduction

The generalization of positional number representations to a wide range of digit sets or to higher dimensions is a fascinating story. Grünwald (1885) investigated negative-based, Kempner (1936), Knuth (1960), Khmelnik (1964), Penney (1965) complex-based systems. From the 70's Kátai, B. Kovács, Környei, Pethő (the "Hungarian school") and Gilbert examined systematically the radix extensions in *algebraic number fields*. In the 90's the *topological aspects* of radix representations were studied by Bandt, Indlekofer, Járai, Kátai, Lagarias, Wang, Vince, and later by Akiyama, Thuswaldner and others. The canonical number representation was generalized to *arbitrary polynomial systems* by Pethő (1989), and investigated later extensively by many authors (incl. Akiyama, Brunotte, Kovács, Pethő, Rao, Scheicher, Thuswaldner). The number system concept in *general lattices* was investigated first by Vince (1993). The *algorithmic aspects* of canonical (polynomial) systems was initiated by Brunotte (2001) and for general lattices by the second

[a]Eötvös Loránd University, Budapest, Hungary
[b]E-mail: `peter.hudoba@inf.elte.hu`, ORCID: `0000-0001-5810-4193`
[c]E-mail: `attila@inf.elte.hu`, ORCID: `0000-0002-1858-7618`

DOI: `10.14232/actacyb.289524`

author (2000). Recently, a special type of radix systems (SRS) studied in length by Thuswaldner and his co-workers (the "Austrian school").

2 Preliminaries

Let Λ be a lattice in $\mathbb{R}^n$ and let $M : \Lambda \to \Lambda$ be a linear operator such that $\det(M) \neq 0$. Let furthermore $0 \in D \subseteq \Lambda$ be a finite subset. Lattices can be seen as finitely generated free Abelian groups and have many significant applications in pure mathematics (Lie algebras, number theory and group theory), in applied mathematics (coding theory, cryptography) because of conjectured computational hardness of several lattice problems, and are used in various ways in the physical sciences. We note that the number system research in general lattices comprises also the orders.

Definition 1. *The triple* (Λ, M, D) *is called a* number system *(GNS) if every element* x *of* Λ *has a unique, finite representation of the form*

$$x = \sum_{i=0}^{L} M^i d_i \ ,$$

where $d_i \in D$ *and* $L \in \mathbb{Z}$ *(*$L+1$ *is the length of the expansion).*

Here M is called the *base* and D is the *digit set*. It is easy to see that similarity preserves the number system property, i.e., if M_1 and M_2 are similar via the matrix Q then (Λ, M_1, D) is a number system if and only if $(Q\Lambda, M_2, QD)$ is a number system at the same time. If we change the basis in Λ a similar integer matrix can be obtained, hence, there is no loss of generality in assuming that M is integral acting on the lattice $\mathbb{Z}^n$. If two elements of Λ are in the same coset of the factor group $\Lambda/M\Lambda$ then they are said to be *congruent* modulo M. The following theorem gives a necessary condition for the number system property.

Theorem 1. *If* (Λ, M, D) *is a number system, then (1)* D *must be a full residue system modulo* M*, (2)* M *must be expansive, and (3)* $\det(I_n - M) \neq \pm 1$*. (unit condition). If a system fulfils the first two conditions then it is called a radix system.*

We note that the theorem in this form was stated first in [9] but it was well-known and used much earlier by Kátai and Vince. The full residue system property can be decided easily using Smith normal form [8]. Algorithms, that calculate the eigenvalues of M exactly in a finite number of steps exist only for a few special classes of matrices. For general matrices, iterative algorithms produce approximate solutions. In polynomial systems, where M is the companion of a monic integer polynomial f, deciding the Schur or Hurwitz stability of f is computationally equivalent with the expansivity check. Verification of the unit condition is trivial.

Write $\varphi : \Lambda \to \Lambda$, $x \overset{\varphi}{\mapsto} M^{-1}(x - d)$ for the unique $d \in D$ satisfying $x \equiv d \pmod{M}$. Since M^{-1} is contractive and D is finite, there exists a norm $\|.\|$ on Λ and a constant C such that the orbit of every $x \in \Lambda$ eventually enters the finite

set $S = \{x \in \Lambda \mid \|x\| < C\}$ after repeated application of φ. This means that the sequence $x, \varphi(x), \varphi^2(x), \ldots$ is eventually periodic for all $x \in \Lambda$. Clearly, (Λ, M, D) is a number system iff for every $x \in \Lambda$ the orbit of x eventually reaches 0. A point p is called *periodic* if $\varphi^k(p) = p$ for some $k > 0$. The orbit of a periodic point p is a *cycle*. The set of all periodic points is denoted by $\mathcal{P}$. The *signature* $[l_1, l_2, \ldots, l_\omega]$ of a radix system is a finite sequence of non-negative integers in which the periodic structure $\mathcal{P}$ consists of $\#l_i$ cycles with period length i $(1 \leq i \leq \omega)$. Clearly, the signature of a number system is $Sig = [1]$.

The following problem classes are in the mainstream of the research.

- For a given (Λ, M, D) the *decision problem* asks if the triple forms a number system or not.
- For a given (Λ, M, D) the *classification problem* means finding all cycles (*witnesses*).
- The *parametrization problem* means finding parametrized families of number systems.
- The *construction problem* aims at constructing a digit set D to M for which (Λ, M, D) is a number system. In general, construct a digit set D to M such that (Λ, M, D) satisfies a given signature.

We note that the algorithmic complexity of the decision and classification problems are still unknown.

The *fundamental domain* or set of "fractions" in (Λ, M, D) can be defined as

$$H = \left\{ \sum_{i=1}^{\infty} M^{-i} d_i : d_i \in D \right\} \subseteq \mathbb{R}^n .$$

Theorem 2. *(a) H is compact. (b) $x \in \mathcal{P} \Leftrightarrow x \in -H$.*

The compactness was proved by many authors (see e.g. Vince [15]). The $\Rightarrow$ part of (b) was proved in [9]. The other direction is obvious as well, otherwise 0 would have at least two different representations.

The theorem means that in order to determine the periodic points it is enough to localize the lattice points in $-H$. There are two different approaches to overcome this problem: the IFS-method (see [8, 10]), and the covering method (see [8, 4]), which was optimized by the authors [6]. The idea of the latter is that we can put the compact set $-H$ into a box B in which the integer elements are easily enumerable. Then, we can compute the pairs $(x, \varphi(x))$ for all $x \in B$, and finally, we determine the cycles applying one of the cycle finding methods.

There are other algorithms for solving the decision/classification problems. Based on the method of Vince [15], Brunotte [2] described a digit-propagation algorithm for polynomial systems with canonical digits. Later, his method was generalized for arbitrary operators and digit sets [4]. The shortcoming of this method

is the sequential nature of the digit propagation, however, there is an algorithmic attempt to overcome this disadvantage [14].

Let $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + x^n$ be an integer (monic) polynomial. Let us denote the factor ring $\mathbb{Z}[x]/(f)$ by Λ_f. Then Λ_f is a lattice and all the problems regarding number expansions in Λ_f can be formulated in $\mathbb{Z}^n$, where M is the companion of f. If f is irreducible then Λ_f is isomorphic to $\mathbb{Z}[\theta]$ where $f(\theta) = 0$ in an appropriate extension of $\mathbb{Q}$. Hence, if the digit set D is restricted to be a set of non-negative numbers $D = \{0, 1, \ldots \mid a_0 \mid -1\}$, we get a straightforward generalization of the traditional number systems in $\mathbb{Z}$. In this case the digit set is called *canonical*. If the radix system (Λ_f, θ, D) satisfies the unique representation property of Definition 1 with some canonical digit set D then it is called a *canonical number system* (CNS). The notion of canonical digit sets can be extended to form a *j-canonical* set $D_j = \{0, e_j, \ldots, (\mid a_0 \mid -1)e_j\} \subset \mathbb{Z}^n$ (e_j is the jth unit vector) [8]. There exists a canonical number system in $\mathcal{O}_K$ – the ring of integers of the algebraic number field K – if and only if there is a power integral basis in $\mathcal{O}_K$ [12]. We note that canonical digit sets may or may not exist in different lattices and canonicity depends on the chosen basis. The *symmetric digit set* is formed by those integer multiples of e_j which are closest to the origin, and was introduced by Kátai [7]. The *adjoint digit set* consists of those lattice points which are in $\det(M)\big[-\frac{1}{2}, \frac{1}{2}\big)$. The *dense digit set* — in which each digit has the smallest norm in its residue class — were introduced and used extensively by the second author. We note that the adjoint set is a dense one in a special basis.

3 The toolset

In order to be able to support the theoretical research we built a Python based toolset that aid at the investigations and experiments. The toolset implements some basic functionalities for number expansion research. It offers multiple ways to solve the decision or classification problems from a simple brute force to probabilistic solutions. In this section we give a short outline of the functionality of the toolset.

3.1 Construction

The toolset contains multiple classes with different purposes. The main class, named RadixSystem, implements a Radix System that we can create with a base and a digit set. The base matrix can be set directly, but there is a function to convert a polynomial to a matrix (creating its companion) as well. The digit set can be passed directly with a list, exactly determining the digits, or with a generator, that generates specific types of digit sets (*RadixSystemSymmetricDigits*, *RadixSystemCanonicalDigits*, *RadixSystemShiftedCanonicalDigits*, *RadixSystemAdjointDigits*, *RadixSystemDenseDigits*).

An example for creating the following radix systems can be seen in the Listing 1:

$$\left(\mathbb{Z}^2, \begin{bmatrix} 0 & -7 \\ 1 & -7 \end{bmatrix}, \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 6 \\ 0 \end{bmatrix} \right\}\right) \text{ and } \left(\mathbb{Z}^2, \begin{bmatrix} 0 & -7 \\ 1 & -7 \end{bmatrix}, \left\{ \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 3 \\ 0 \end{bmatrix} \right\}\right)$$

Listing 1: Example of construction with the toolset

```
rs = RadixSystem([[0,-7],[1,-7]],
    [[0, 0], [1, 0], [2, 0], [3, 0], [4, 0], [5, 0], [6, 0]])

#Using digit set generator
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemCanonicalDigits())

#Creates the same system with symmetric digit set
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemSymmetricDigits())
```

3.1.1 Necessary conditions

A radix system object can be created by the toolset, but if the necessary conditions for the computations do not hold then the toolset will throw the appropriate exception. The necessary conditions (see Section 2) mean that the base operator has to be regular ($\det(M) \neq 0$) (otherwise the system throws a *RadixSystemRegularityException* exception), the digit set must be a full residue system modulo the base (otherwise the system throws the *RadixSystemFullResidueSystemException*) and the base operator has to be expansive.

The RadixSytem class does not check the unit condition (third necessary condition of Theorem 1) — because it is not a condition for a Radix System, only for a Number Nystem — the class has a function, named *checkUnitCondition*, that returns true if the radix system fulfils that condition.

3.2 Expansions

If the user wants to find the expansion of a lattice point in a specific system she can apply the φ function which is implemented in *phiFunction*. Applying the Smith Normal Form, the system performs the computation efficiently. Applying the *phiFunction* iteratively the system computes the orbit of a point containing the periodic parts (*getOrbitFrom*). Recall that if the orbit of a lattice point ends at zero then it has a finite expansion (*hasFiniteExpansion*).

Listing 2: Example of using φ based functions

```
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemSymmetricDigits())
print(rs.phiFunction([2,3]))
print(rs.getOrbitFrom([6,3]))

# Result:
# [3, 0]
# [[6, 3], [-4, -1], [6, 1], [-6, -1], [6, 1]]
```

3.3 Covers

Based on Theorem 2 ($x \in \mathcal{P} \Leftrightarrow x \in -H$) the system determines a Box that contains all the periodic points (*getCoverBox*). The volume of this Box can also be calculated (*getCoverBoxVolume*) and a Python generator can be obtained to iterate through all of the points within the Box (*getPointsInBox*). It is a simple brute force algorithm for GNS decision (detailed further in Subsection 3.5).

There is a heavyweight algorithm *getCycles* that calculates all of the orbits from all of the points of the Box and returns all of the cycles. Clearly, if the *getCycles* returns only the zero point then the radix system is a number system.

3.4 Drawing tools

The toolset has a class for drawing different aspects of number expansions. The user can analyze the expansions by the expansion graph. By default, it shows the trajectory from all of the points inside the Box. In Figure 1 we can see a radix system that does not fulfil the number system property (since it has a non-trivial cycle $[-1, 1]$). In Figure 2 we can see a plot of some fraction sets.

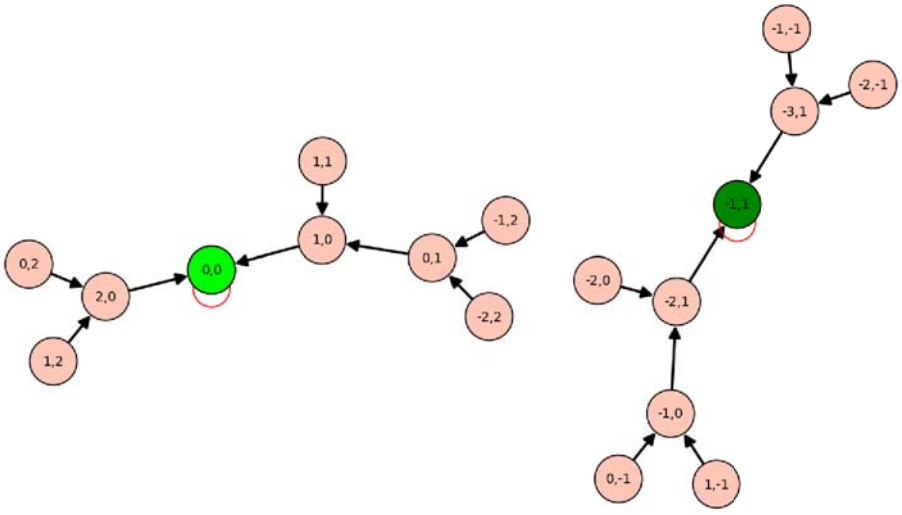


Figure 1: An expansion graph.

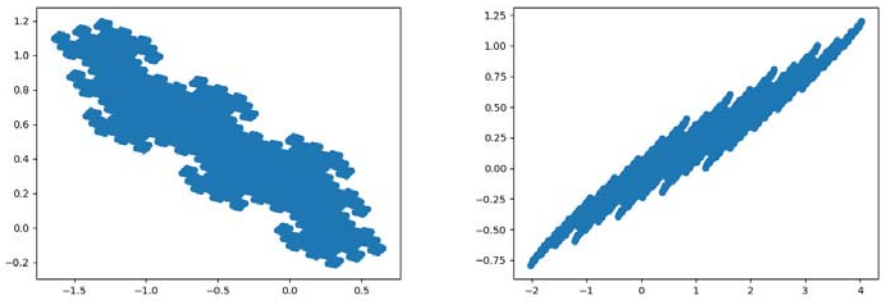


Figure 2: A fraction set of $[[0, -3], [1, 2]]$ and $[[0, -5], [1, -4]]$ with canonical digit sets.

3.5 Decision techniques

In this subsection we discuss some decision methods that can be used by the toolset.

3.5.1 Naive decision

The naive decision method checks the orbits from all points of the cover Box and if there is only one periodic point (should be the zero) then returns true.

Listing 3: Example of how to call a naive decide method of the toolset

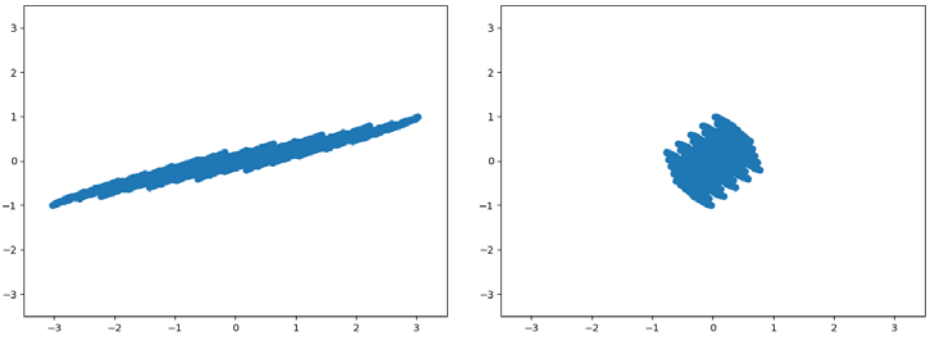
```
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemCanonicalDigits())
print(rs.decideGNS())
```

3.5.2 Volume optimization

The naive method iterates through all of the points within the bounding Box. However, since integer similarity transformations preserve the number system property, we can try to change the basis where the bounding box is smaller. In [4] the authors suggested a simulated annealing genetic algorithm that finds a similarity transformation minimizing the size of the cover box. In Figure 3, we can see an example how the algorithm decreases the volume of the possible space of periodic points. For higher dimensions, the speedup is much higher.

Listing 4: Example of how to call start a cover box volume optimization

```
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemCanonicalDigits())
volumeOptimized = rs.optimize()
print(volumeOptimized.decideGNS())
```



(a) Before optimization (b) After optimization

Figure 3: A fraction set of the operator [[0, -5], [1, -4]] applying the symmetric digit set.

3.5.3 Two-step optimization

In [6] a method was suggested as an extension of the volume optimization. Besides optimizing the volume the authors showed a method of minimizing the number of multiplications in the function φ as well. The amount of multiplications is affected by the Smith Normal Form computation and the inverse computation of the base. If we have the two transformations we can iterate through all of the points of the volume optimized space, transform into the φ optimized space and find there the orbit.

Listing 5: Example of how to use the two-step optimization

```
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemCanonicalDigits())

optimizedVol, optimizeVolT = rs.optimize(returnTransformationAlso=True)

optimizedPhi, optimizePhiT = rs.optimize(
    targetFunction = lambda actVal, T:
          phiOptimizeTargetFunction(actVal, T,optimizeVolT.inverse()),
    returnTransformationAlso=True)

transformMatrix = optimizePhiT * optimizeVolT.inverse()
print(optimizedPhi.decideGNS(startPointSource=optimizedVol,
                             pointTransform=transformMatrix))
```

3.5.4 Length-n cycle

Based on our experiments, we observed that there are significantly larger number of cases when the cycles are short. Therefore we try to find cycles directly based on some digit combinations.

Considering the periodic points with length one, there is a $d \in D$ for $x \in \mathcal{P}$ where $M^{-1}(x-d) = x$ holds. We can reformulate this statement as $x - d = Mx \Rightarrow x - Mx = d \Rightarrow x = (I - M)^{-1}d$. In the algorithm we just simply iterate through all of the digits and check whether the result is a lattice point. If so, we have found a loop. If the digit set is small, this algorithm is really fast.

We can find length two cycles with the same technique. If there is an $x \in \mathcal{P}$ length-two periodic point, then there are $d_1, d_2 \in D$, where $M^{-1}(M^{-1}(x - d_1) - d_2) = x \Rightarrow M^{-1}(x - d_1) = Mx + d_2 \Rightarrow x = M^2x + Md_2 + d_1 \Rightarrow x = (I - M^2)^{-1}(Md_2 + d_1)$. If is there are d_1 and d_2 digits where x is a lattice point then we have found a cycle.

In general, for an length-n periodic point $\varphi^n(x) = x$, hence there are $d_1, ..., d_n \in D$ digits, where $x = (I - M^n)^{-1}(\sum_{i=1}^{n} M^{i-1}d_i)$ is a lattice point.

The weakness of this algorithm is its exponential complexity, i.e., if the digit set is big or there is not any short cycle in the system, then the algorithm finds no cycles (even if it exists).

Listing 6: Example of how to find the directly with fixed lengths cycles

```
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemCanonicalDigits())
print(rs.findNLengthCycle(1))
print(rs.findNLengthCycle(2))

# Result:
# [[0, 0], [0, 0]]
# [[[12, 2], [-12, -2], [12, 2]], [[6, 1], [-6, -1], [6, 1]], [[-18, -3],
    [18, 3], [-18, -3]]]
```

3.5.5 Randomized method

For a given cycle we call all of the lattice points that lead to that cycle by iteration of φ as the **basin of the cycle**. Our experiments showed that at general radix systems most of the orbits lead to a non-zero periodic point. Hence, we can choose random lattice points uniformly and check the orbit to find witnesses (disproving the GNS property).

Listing 7: Example of how to run random test to find non-zero cycle

```
rs = RadixSystem([[0,-7],[1,-7]], RadixSystemSymmetricDigits())
print(rs.probGNSTest(numberOfTrials=100))

# Result:
# [-6, -1]
```

3.5.6 Smart decide

The smart decide algorithm estimates the runtime of the different methods and suggests the best algorithm to solve the decision problem. As we can see in Figure 4 the naive decision time increases faster than the smart decision function.

The algorithm has several steps:

1. If the cover Box is "small", simply brute force the space with the naive method; END.

2. Otherwise search length-n periodic points for small n directly; If it finds a non-trivial witness, return, otherwise continue with the next step.

3. Calculate the orbit of the "close-to-zero" lattice points (maximum the absolute value of $\det(M)$); if it finds a non-trivial witness, return, otherwise continue with the next step (the closeness is in the sense of the infinity norm)

4. Calculate orbits from uniformly chosen random lattice points; if it finds a non-trivial witness, return, otherwise continue with the next step.

5. Volume and φ optimization.

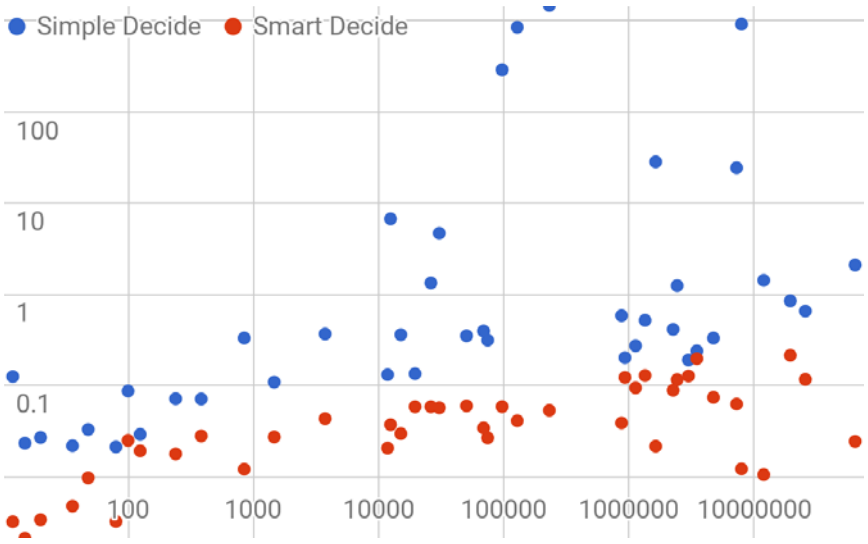


Figure 4: Runtime comparison of the simple decide method and the smart one by the size of the cover Box

3.6 Validation

In order to validate the correct functionality of the toolset we initiated multiple levels of testing.

Let $p(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1} + x^n$ be the characteristic polynomial of the operator M. The following theorems were applied for validation:

- If the strictly dominant condition

$$\sum_{i=1}^{k} |c_i| < |c_0|$$

  holds then then M is expansive. This is an immediate consequence of Rouché's theorem.

- If there is a norm for which $\|M^{-1}\| < 1/2$ then M can serve as a basis for a number system with dense digits [5].

- If $1 \leq c_{n-1} \leq \cdots \leq c_1 \leq c_0$ then the companion of $p(x)$ serve a basis for a number system with canonical digits [11].

- If the strictly strong dominant condition

$$2\sum_{i=1}^{k} |c_i| < |c_0|$$

  holds then the companion of $p(x)$ can serve as a basis for a number system with symmetric digits [5].

We tested the toolset with known special cases of number systems, e.g. in [1] the 2nd theorem states that if

$$c_2, .., c_{n-1}, \sum_{i=1}^{n} c_i \geq 0$$

and the strictly strong dominant condition holds then the companion of $p(x)$ is a number system with canonical digits. We used listing of GNS examples in multiple articles [13, 3], and we sampled polynomials and operators randomly for validation as well.

4 Database

The research area has plenty of unsolved problems. Most of the problems have solutions for a specific forms of radix systems. To state and validate various conjectures it is necessary to collect and filter the partial results and sample candidates. Therefore we implemented a server-side application which is able to store various data on number expansions. At present the database contains more than 10 000

items. The uploaded data are about companions of expansive polynomials with constant terms ±2,±3,±5, ±7 together with their number system status and witnesses, etc. We used canonical and symmetric digit sets as well, and we calculated many combinations of product systems.

The data server allows to read data from the server publicly via a JSON API and the registered users with own API token can send new items/properties to the database. The items can be filtered by any custom property.

The server already stores plenty of properties, like eigenvalues, eigenvectors, periodic points and orbits, classification details, etc.

Listing 8: Example of how to request candidates from the public database

```
result = callServer('http://numsys.info/radix-system/list',{
    '.volume':'<1000',
    '.dimension':'3',
    'size':5
})

for r in result:
    rs = RadixSystem(r['base'],r['digits'])
    print(r['base'],r['digits'],rs.smartDecide())

# Result:
# [[0, 0, -2], [1, 0, -2], [0, 1, -2]] [[0, 0, 0], [1, 0, 0]] True
# [[0, 0, -2], [1, 0, 0], [0, 1, -1]] [[0, 0, 0], [1, 0, 0]] False
# [[0, 0, -2], [1, 0, -1], [0, 1, -1]] [[0, 0, 0], [1, 0, 0]] True
# [[0, 0, -2], [1, 0, 1], [0, 1, 0]] [[0, 0, 0], [1, 0, 0]] True
# [[0, 0, -2], [1, 0, 0], [0, 1, 0]] [[0, 0, 0], [1, 0, 0]] True
```

5 Experimental observations

The database helps the researchers filtering out some special data. Analysing the uploaded data we have some observations.

- In general, the non-zero basins are significantly large on average. Based on this observation the randomized method is a viable alternative for checking the GNS property (more detail in Section 3.5.5).
- In general, the cycle lengths are relatively short. Therefore we suggest the length-n cycle method for the decision, if possible (detailed in Section 3.5.4.)
- There is always at least one lattice point that leads to a non-zero periodic point in the $\mid \det(M) \mid$ neighbourhood of the origin (in infinity norm).
- More than 700 samples in the database suggested the following theorem:

Theorem 3. *Suppose that the system* (Λ, M, D) *is GNS. Then* (Λ, M^n, D_n) *is GNS for all* $n \in \mathbb{N}$*, where*

$$D_n = \{d_0 + Md_1 + M^2 d_2 + \ldots + M^{n-1} d_{n-1} \ : \ d_i \in D\}$$

taking all possible combinations for the digits d_i *above.*

Proof. Let $n > 1$ be fixed. Since (Λ, M, D) is GNS therefore all $x \in \Lambda$ can be written uniquely in the form

$$x = d_0 + Md_1 + \cdots + M^k d_k \,, \tag{1}$$

where $d_i \in D$. Equation (1) can be rewritten as

$$x = (d_0 + \cdots + M^{n-1} d_{n-1}) + M^n (d_n + \cdots + M^{n-1} d_{2n-1}) + \cdots + M^k d_k \,.$$

Since the coefficients of each M^{nj} are digits from D_n for all $j \geq 0$ therefore the system (Λ, M^n, D_n) is GNS as well. □

6 Conclusion and further work

The paper introduced a toolset for supporting lattice-based number expansion computations. The toolset was implemented in Python. Besides, the authors built a database storing different radix system parameters and offers the researchers to upload and search in this database. In the future we plan to improve, extend and distribute the toolset and try to find a mathematical proof for some of our observations.

References

[1] Akiyama, S. and Rao, H. New criteria for canonical number systems. *Acta Arithmetica*, 111(1):5–25, 2004. DOI: 10.4064/aa111-1-2.

[2] Brunotte, H. On trinomial bases of radix representations of algebraic integers. *Acta Scientiarum Mathematicarum*, 67(3–4):521–527, 2001.

[3] Burcsi, P. and Kovács, A. Exhaustive search methods for CNS polynomials. *Monatshefte für Mathematik*, 155(3-4):421, 2008. DOI: 10.1007/s00605-008-0005-y.

[4] Burcsi, P., Kovács, A., and Papp-Varga, Zs. Decision and classification algorithms for generalized number systems. *Ann. Univ. Sci. Budapest. Sect. Comput*, 28:141–156, 2008.

[5] Germán, L. and Kovács, A. On number system constructions. *Acta Mathematica Hungarica*, 115(1-2):155–167, 2007. DOI: 10.1007/s10474-007-5224-5.

[6] Hudoba, P. and Kovács, A. Some improvements on number expansion computations. *Numeration 2016*, page 65, 2017.

[7] Kátai, I. *Generalized number systems and fractal geometry.* Janus Pannonius Tudományegyetem, Pécs, 1995.

[8] Kovács, A. On the computation of attractors for invertible expanding linear operators in z (kappa). *Publicationes Mathematicae Debrecen*, 56(1-2):97–120, 2000.

[9] Kovács, A. *Number Systems in Lattices.* PhD thesis, Eötvös Loránd University, Budapest, Hungary, 2001.

[10] Kovács, A. Number expansions in lattices. *Mathematical and Computer Modelling*, 38(7-9):909–915, 2003. DOI: 10.1016/S0895-7177(03)90076-8.

[11] Kovács, B. Canonical number systems in algebraic number fields. *Acta Mathematica Academiae Scientiarum Hungarica*, 37(4):405–407, 1981. DOI: 10.1007/BF01895142.

[12] Kovács, B. Integral domains with canonical number systems. *Publ. Math. Debrecen*, 36:153–156, 1989.

[13] Pethő, A. On a polynomial transformation and its application to the construction of a public key cryptosystem. *Computational Number Theory*, pages 31–43, 1991. DOI: 10.1515/9783110865950.31.

[14] Tátrai, A. Parallel implementations of brunotte's algorithm. *Journal of Parallel and Distributed Computing*, 71(4):565–572, 2011. DOI: 10.1016/j.jpdc.2010.12.010.

[15] Vince, A. Replicating tessellations. *SIAM Journal on Discrete Mathematics*, 6(3):501–521, 1993. DOI: 10.1137/0406040.