http://mmg.tversu.ru ISSN 2311-1275

Mathematical Modelling and Geometry

Volume 14, No 1, p. 1 – 8 (2026) doi:10.26456/mmg/2026-1411

Application of Gröbner bases for finding accurate states of equilibrium of migration and population models

A. A. Mamonov[1,a]

[1] Peoples' Friendship University of Russia named after Patrice Lumumba, Moscow, 117198
e-mail: [a] mamonov-aa@rudn.ru

Received 1 January 2026, in final form 12 January 2026. Published 13 January 2026.

Abstract. The paper considers the using Gröbner basis to accurately find all equilibrium states of migration and population models with competition, as a solution for nonlinear equation system. Example for four-dimensional and six-dimensional models presented. Using a simple example of such a system, it is shown that standard numerical methods find equilibrium states with low accuracy. The proposed approach using the GINV library (Python) and the SageMath computer algebra system makes it possible to reliably find all the roots and obtain them with arbitrary accuracy. The program code and calculation results are provided.

Keywords: Grobner basis, migration-population model, state of equilibrium, nonlinear equation systems.

MSC numbers: 34A34, 13P10.

1. Introduction

Mathematical modeling of population systems, taking into account migration flows and competition of species, is of considerable scientific and practical interest. For example, they can be used to preserve biodiversity, which is one of the key tasks of modern ecology. To effectively solve this problem, it is necessary to be able to predict population dynamics taking into account a variety of factors, including migration and competition between species.

The Lotka-Volterra model was created in the mid-1920s by two prominent scientists. Alfred Lotka developed his own version of the model in 1925 [1], and Vito Volterra created a similar one independently in 1926[2]. Since that time that model was the main tool for population studying and predictions.

This model, along with subsequent ones, described by systems of nonlinear differential equations. The complexity of solving such systems increases significantly when the dimension of the model is increased. At the same time, there is no universal effective way to solve systems of nonlinear equations.

Although the basic Lotka-Volterra predator-prey model is well described and studied, more complex models are often used to approach real-world tasks. In the dissertation [3], a family of multidimensional models of the "k competitors - k migration areas" type is investigated. For a simple case, $k = 2$ is considered: two competing species inhabit the main range and can migrate to the corresponding migration refuge areas. A four-dimensional simplified model (two competitors - two shelters) is set by the system:

$$\begin{cases} \dot{x}_1 = ax_1 - px_1^2 - rx_1x_3 + \beta x_2 - \beta x_1, \\ \dot{x}_2 = ax_2 - px_2^2 + \beta x_1 - \beta x_2, \\ \dot{x}_3 = ax_3 - px_3^2 - rx_1x_3 + \delta x_4 - \delta x_3, \\ \dot{x}_4 = ax_4 - px_4^2 + \delta x_3 - \delta x_4 \end{cases} \tag{1}$$

where x_1, x_3 are densities of competing populations in the main area; x_2, x_4 are densities of the same populations in shelters; a is the coefficient of natural growth; p is the coefficient of intraspecific competition; r is the coefficient of interspecific competition; β, δ are migration coefficients. In [3] thats the system (2.3)

In [3], the following values for model parameters are obtained by the differential evolution: $a = 4.656449$, $p = 0.582056$, $r = 2.561047$, $\beta = 4.365421$, $\delta = 2.037196$. After that numerical methods of the *sympy* library were used to search for equilibrium states, which found the solutions, but with limited accuracy (error in the second decimal place). In this paper, we propose an algorithm for searching for equilibrium states for the such models with algebraic accuracy by finding the Gröbner basis.

2. Finding Gröbner basis to find equilibrium states

The equilibrium states of system 1 are the roots of a polynomial algebraic system obtained by equating the right-hand sides to zero. Classical numerical iterative methods (Newton-type methods, evolutionary algorithms) solve this system approximately and do not guarantee finding all the roots. In particular, Vasilyeva I. I. points out in [3] that "the dimension of the model and the capabilities of the applied software used make it possible to obtain only a part of the solutions."

An alternative is to use the Gröbner basis method [4, 5]. The Gröbner basis of the ideal $I = [f_1, \dots, f_m]$ in the ring of polynomials $k[x_1, \dots, x_n]$ with lexicographic order provides a triangular (stepwise) decomposition of the system: the last element of the basis depends only on one unknown - x_n. One of the penultimates depends only on x_n and x_{n-1}, and so on. This allows to consistently find all the roots by back-substitution procedure without losing any solutions.

The key advantage of the approach is as follows: unlike numerical methods, which can both skip roots and give superfluous (pseudo-) solutions, the Buchberger algorithm ensures that a complete set of equilibrium states is found - none is lost and none is added.

An additional advantage is accuracy. The roots of the last element of the Gröbner basis in the SageMath system are represented as algebraic numbers: the system represents values as the roots of an algebraic equation, and can calculate their numerical value with an arbitrary specified accuracy (command **n(x, digits=k)**). This is fundamentally different from the results in [3], where the precision is limited to two decimal places.

3. Software implementation and calculation results

3.1 The four-dimensional model

Below is a SageMath .ipynb file reproducing system 1 and finding all equilibrium states using the built-in **variety()** function over the field of algebraic numbers AA:

```
x = var('x1, x2, x3, x4')
a = 4.656449;  p = 0.582056;  r = 2.561047
b = 4.365421;  d = 2.037196
eqs = [
    a*x1 - p*x1^2 - r*x1*x3 + b*x2 - b*x1,
    a*x2 - p*x2^2 + b*x1 - b*x2,
    a*x3 - p*x3^2 - r*x1*x3 + d*x4 - d*x3,
    a*x4 - p*x4^2 + d*x3  - d*x4
]

Dics = (QQ[x]*eqs).variety(AA)
```

Table 1: All equilibrium states for 1

	x_1	x_2	x_3	x_4
S_1	0	0	0	0
S_2	0	0	8.000	8.000
S_3	8.000	8.000	0	0
S_4	0	0	2.436	-1.436
S_5	0	0	-1.436	2.436
S_6	1.419	3.522	4.021	6.625
S_7	5.788	6.843	0.813	5.062
S_8	1.000	3.000	5.000	7.000

The **variety(AA)** command directly finds all variations in the values of the variables x, which turn the ideal generated by the *eqs* system to zero. The argument AA specifies the use of algebraic numbers. The calculation results are presented in 1.

Thus, the method finds 8 equilibrium states, of which 3 are without zero populations. In [3] numerical methods were also detected three positive state with non-zero components: $S_8(0.99, 2.99, 5.06, 7.04)$, $S_9(1.44, 3.55, 4.00, 6.64)$, $S_{10}(5.81, 6.87, 0.81, 5.08)$. The comparison shows that the point S_6, S_7, S_8 the present work correspond to the points S_{10}, S_9, S_8 and easily recognizable. However, the accuracy in [3] is limited to two digits. Provided solution allows you to get the value of each root with arbitrary precision using the command **n(val, digits=?)**, with **?** specifying number of digits.

3.2 The six-dimensional model

Let us consider a more detailed application of the proposed approach using a more complex example - the six-dimensional model ("three competitors - three migration areas"). In [3], let's take the example (3.7), with the proposed set of parameters.

Let us set the system in the Sage system:

```
x=var('x1,x2,x3,x4,x5,x6')
a1=9.87107026, a2=9.97602024, a3=9.94653833
a4=9.96876543, a5=1.42213144, a6=7.54755514
p11=0.10331098, p22=0.10190956, p33=0.14403938
p44=0.10093958, p55=2.44771688, p66=4.59652965
p13=0.12519358, p31=0.10242677, p56=0.97589257
p65=0.27400194, q15=0.10455808, q16=0.10524564
q35=0.11179302, q36=0.1032072, d15=6.31530649
d35=7.24952535, d16=8.66821615, d36=8.96873736
b=3.47237116, g=3.53131469, d=3.15533821, e= 3.04329167
```

```
eqs = [
    a1*x1 - p11*x1^2 - p13*x1*x3 - q15*x1*x5 - q16*x1*x6 + b*x2 - g*x1,
    a2*x2 - p22*x2^2 + g*x1 - b*x2,
    a3*x3 - p33*x3^2 - p31*x1*x3 - q35*x3*x5 - q36*x1*x6 + e*x4 - d*x3,
    a4*x4 - p44*x4^2 + d*x3 - e*x4,
    a5*x5 - p55*x5^2 - p56*x5*x6 + d15*x1*x5 - d35*x3*x5,
    a6*x6 - p66*x6^2 - p65*x5*x6 + d16*x1*x6 - d36*x3*x6,
]
```

Now declare a polynomial ring of our variables and set an ideal for it in accordance with the equations:

```
P = PolynomialRing(QQ, variables, order='lex')
J=P*[P(eq) for eq in eqs]
```

Let's find the Gröbner basis:

```
B=J.groebner_basis()
```

The last element of the basis is always an equation from one unknown:

```
B[-1]
```

$$x_6^{33} - 148\ldots872/801\ldots385 * x_6^{32} - \ldots$$

Monom x_6^{32} here has coefficient with 182 and 179 digits in fraction.

Now let's find roots for these equasion:

```
Roots = QQ[x6](B[-1]).roots(AA, multiplicities=False)
Roots
```

[-148.94?, -142.45?, -77.77?, -77.16?, -74.88?, -30.17?, -13.25?, -13.18?, 0, 0.55?, 0.74?, 1.64?, 1.64?, 10.97?, 12.27?, 21.05?, 37.41?, 38.28?, 52.34?, 70.07?, 100.89?]

Now we can proceed with back-substitution procedure, recreating full roots for our sistem.

It should be noted, however, that finding Gröbner basis is a laborious task, since its computational complexity is equal to a double exponent. For this example, the following computer configuration was used:

1. Two Intel Xeon L5630 processors, a total of 8 cores and 16 logical processors. The base clock frequency of each core is 2134 MHz.

2. RAM: 100 gigabytes of RAM were allocated for computing on the server platform.

And basis for six-dimensional model was found in 205 seconds.

To speed up computational process and widen range of models, accessible to studies, we suggest using GInv system. GInv is a software system, based on a variation of the Buchberger algorithm for finding the Grobner basis for polynomial ideals. Gerdt, Zharkov, and Blinkov proposed a new version of Buchberger's algorithm based on involutive division [6]. In the 2000s, the GInv software was created to implement this algorithm. Recently, this software has been updated, tested, and modified for use in broader areas.

The modern version of the GInv system has been transferred to Python, brought to a stable state, and posted on a separate repository [7], under the name GInvDist.

Now, to use it for out task, we run simple script:

```
from ginv.monom import *
from ginv.poly import *
from ginv.gb import *

variables = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6']
init(variables, Monom.TOPdeglex)
equations = [
    9.87107026*x1 - 0.10331098*x1**2 - ...,
    9.97602024*x2 - 0.10190956*x2**2 + ...,
    ...
]

G = GB()
G.algorithm2(equations)

print('Grobner Basis:')
print(G)
```

On the same computer configuration this script finds Gröbner basis in 0.67 seconds. After that, the same steps could be followed to get solutions for equilibrium states.

This not only saves as a relatively short wait on solving six-dimensional model, but show a promise on more manageable computations for higher-dimensional models.

4. Conclusion

The paper proposes the application of the Buchberger algorithm for constructing Gröbner basis for accurately and completely finding the equilibrium states of migration-population models with competition. That not only provides better accuracy, but, using GInv software, can be more easialy scaled up. For six-dimensional systems, the use of the GInv accelerates calculations by more than 300 times compared to SageMath on the same server.

It is recommended to use the Buchberger algorithm (or the involutive variants implemented in GINV) as the main tool for finding equilibrium states in the study of migration and population models of any dimension. The above program code allows us to directly apply this approach to systems from [3, 8, 9, 10] and their generalizations with a large number of species and ranges.

References

[1] Lotka A. J. Elements of Physical Biology. - Baltimore: Williams and Wilkins, 1925.

[2] Volterra V. Fluctuations in the abundance of a species considered mathematically
Nature. - 1926. - Vol. 118. - P. 558-560.

[3] Vasilyeva I. I. Qualitative and numerical study of multidimensional migration and population models with competition: PhD Candidate of Physical and Mathematical Sciences. Yelets: I. A. Bunin YSU, 2025.

[4] Buchberger, Bruno. (1965). An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal (in German).

[5] LF, H. Ideals, Varieties, and Algorithms (4th ed.) [Cox, Little & O'Shea 2015 06 14]. https://doi.org/10.1007/978-3-319-16721-3

[6] Gerdt V. P., Blinkov Yu. A. Involutive Bases of Polynomial Ideals
Math. and Computers in Simulation. - 1998. - Vol. 45. - P. 519-541

[7] GInvDist, url = https://github.com/TheMumblingMammoth/GInvDist

[8] Vasilyeva I. I., Druzhinina O. V., Masina O. N. Construction and research of a population dynamic model of the "two competitors - two migration areas" type
Nonlinear world. - 2022. - Vol. 20, No. 4. - pp. 60-68.

[9] Vasilyeva I. I., Demidova A. V., Druzhinina O. V., Masina O. N. Computer research of deterministic and stochastic models two competitors two migration areas taking into account the variability of parameters
Discrete and Continuous Models and Applied Computational Science. - 2024. - Vol. 32, No. 1. - P. 61-73.

[10] Vasilyeva I. I., Demidova A. V., Druzhinina O. V., Masina O. N. Construction, stochastization and computer study of dynamic population models two competitors two migration areas
Discrete and Continuous Models and Applied Computational Science. - 2023. - Vol. 31, No. 1. - P. 27-45.