Hybrid symbolic-numeric methods in geosciences

Joseph L. Awange[1], Béla Paláncz[2], Lajos Völgyesi[3], and Szabolcs Rózsa[3]

1 Spatial Sciences Discipline, School of Earth and Planetary Sciences, Curtin University, Australia
E-Mail: j.awange@curtin.edu.au
2 Department of Photogrammetry and Geoinformatics, Budapest University of Technology and Economics, Hungary
E-Mail: palancz@epito.bme.hu
3 Department of Geodesy and Surveying, Budapest University of Technology and Economics, Hungary
E-Mail: volgyesi@eik.bme.hu, rozsa.szabolcs@epito.bme.hu

Abstract
Modern computing systems, like *Mathematica*, are blending numeric and symbolic evaluation of many algorithms improving their efficiencies (time, accuracy). In this contribution three toy-examples of the application of hybrid symbolic-numeric computations in geosciences are presented in order to illustrate the features of this advanced technique, namely: ranging GNSS satellites, computing GNSS cycle ambiguities and employing symbolic regression for verifying Kepler's third law.

1 Introduction

Hybrid symbolic-numeric computation (HSNC) is a large and growing area at the boundary of mathematics and computer science, devoted to the study and implementation of methods that mix symbolic with numeric computation.

The ideal computational systems for hybrid methods are able to carry out numeric algorithms with any arithmetic precision as well as algorithms with any symbolic objects (rational numbers, symbolic variables without assigned numeric values or other mathematical objects i. e. digital images). *Maple* and *Mathematica* mention only the flagships of such systems.

Mixed-integer programming (wherein some variables are discrete-valued and others continuous) is also a natural area for HSNC since it combines aspects of exact and numeric methods in the handling of both discrete and continuous variables.

Symbolic Regression, a methodology that blends numerics with evolutionary programming, introduced for the purpose of modeling data can be similarly considered as a natural part of HSNC.

This paper provides three geodetic examples illustrating the strengths of these techniques. However in our new book Awange et al. (2018) many other areas of HSNC are discussed with geodetic applications.

2 Numeric versus symbolic solution

Plainly speaking, a *numeric (or numerical) method* is one that could be done with a simple hand-held calculator, using basic arithmetic, square roots, some trigonometric functions, and a few other functions that most people learn about in high school. Depending on the task, one may have to press the calculator's buttons thousands (or even millions) of times, but theoretically a person with a calculator and some paper could implement a numerical method. When finished, the paper would be full of arithmetic.

A *symbolic method* involves algebra. It is a method that, if implemented by a person, it would involve al-

Erschienen bei KIT Scientific Publishing
Schriftenreihe des Studiengangs
Geodäsie und Geoinformatik 2018,1

DOI Einzelbeitrag:
10.5445/KSP/1000080204

Festschrift zur Verabschiedung von
Prof. Dr.-Ing. Dr. h.c. Bernhard Heck
(Schw)Ehre, wem (Schw)Ehre gebührt

DOI Festschrift:
10.5445/KSP/1000080324

gebraic or higher rational thought. A person implementing a symbolic method will rarely need to reach for a calculator. When finished, there may be some numbers, but the paper would be full of variables such as x, y, z.

Numeric computations:

- usually require initial values and iterations. They are sensitive to round off errors, provide only one local solution,
- can be employed for complex problems, and the computation times are short in general because the steps usually translate directly into computer machine language.

Symbolic computations:

- do not require initial values and iterations. They are not sensitive to round-off errors, and provide all solutions,
- often cannot be employed for complex problems, and the computation time is long in general because the steps usually require computer algebra system software.

3 Hybrid solution

Symbolic methods may be hard to develop, and they may be difficult for a computer to implement, but they lead to insight. Fortunately, we are not forced into a strict either/or dichotomy. There are symbolic-numeric methods, hybrids using the strengths of both ideas.

Ideally the best strategy is to divide the algorithm into symbolic and numeric parts in order to utilize the advantages of both techniques. Inevitably, numeric computations will always be used to a certain extent. So using symbolic forms, the *computation time can be reduced* considerably. This so-called *hybrid computation* has an additional advantage too, namely the symbolic part of the algorithm does *not generate round-off errors*.

Another approach of applying the hybrid computation is to *merge* symbolic evaluation with numeric algorithm. For example numeric Runge-Kutta algorithms can be carried out with symbolic variable (s) to solve boundary value problems without iteration.

4 Geodetic applications

4.1 Ranging to more than four GNSS satellites

Throughout history, position determination has been one of the most important tasks of mountaineers, pilots, sailor, civil engineers etc. In modern times, Global Navigation Satellite Systems (GNSS; a collection of the US based GPS, Russian GLONASS, Chinese Beidou and the European Galileo; e.g., Awange (2012, 2018)) provide an ultimate method to accomplish this task. If one has a hand held GNSS receiver, which measures the travel time of the signal transmitted from the satellites, the distance travelled by the signal from the satellites to the receiver can be computed by multiplying the measured time by the speed of light in vacuum. The distance of the receiver from the i-th GNSS satellite, the pseudo-range observations, d_i is related to the unknown position of the receiver, $\{x_1, x_2, x_3\}$ by

$$d_i = \sqrt{(x_1 - a_i)^2 + (x_2 - b_i)^2 + (x_3 - c_i)^2} + x_4\,, \quad (4.1)$$

where $\{a_i, b_i, c_i\}; i = 0, 1, 2, \ldots, n > 3$ are the coordinates of the i-th satellite.

The distance is influenced also by the satellite and receiver' clock biases. The satellite clock bias can be modeled while the receiver' clock bias has to be considered as an unknown variable, x_4. This means, we have four unknowns, consequently we need four satellites to provide a minimum observation. The general form of the equation for the i-th satellite is

$$\begin{aligned} f_i = {} & (x_1 - a_i)^2 + (x_2 - b_i)^2 + (x_3 - c_i)^2 \\ & - (x_4 - d_i)^2\,. \end{aligned} \quad (4.2)$$

The system can be solved in many ways (see Awange et al., 2010; Awange and Paláncz, 2016). However in general there are two steps of the solution:

1) compute approximate solution using Gauss-Jacobi method employing Gröbner basis for every subset of size four,
2) then improving the result with local minimization of the sum of square of errors of the equation.

Employing hybrid numeric symbolic computation one can improve the efficiency of the algorithm in both phases. We shall illustrate the hybrid solution with a small numeric example based on the data in Awange and Paláncz (2016, see Table 15.2 on page 293.).

In the first step, to compute an approximate solution, the Gröbner basis with inexact coefficients can be employed. Computing Gröbner bases with inexact coefficients is often desired in industrial applications, but the computation with floating-point numbers is quite unstable if performed naively (Sasaki, 2014). The solution methods of the Gröbner basis are very sensitive to round-off error, therefore sometimes in case of systems that are over-constrained or have roots with multiplicities, and are given with inexact coefficients, hybrid symbolic-numeric methods are required. Recently Szanto (2011) and Lichtblau (2013) discussed computation of Gröbner bases using approximate arithmetic for coefficients and showed how certain considerations of tolerance, corresponding roughly to accuracy and precision from numeric computation, allow us to obtain good approximate solutions to problems that are overdetermined.

Let us consider the list of equations **eqs** = f_i with the numerical data for $f_i, i = \{0, 1, \ldots, 5\}$, and employ Gröbner basis with inexact coefficients, we get the approximate solution employing *Mathematica*

```
AbsoluteTiming[grb = GroebnerBasis[eqs,
{x1app,x2app,x3app,x4app },Tolerance→10^-3]]
{0.00596268,{1.x4app,-4.08821×10^6+1.x3app,
4.84782×10^6+1.x2app,-596925.+1.x1app}}  .
```

This result can be used in the second step, to improve the result. The error to be minimized is

$$R = \sum_{i=0}^{5} f_i^2 . \tag{4.3}$$

For local minimization Newton method can be employed, since it has quadratic convergency. We can reduce the computation time to half if the gradient (**GradS**) and Hessian matrix (**HessianS**) are precomputed symbolically instead of computing them numerically in every iteration step

```
AbsoluteTiming[
FindMinimum[f,{{x1,x1app},{x2,x2app},
{x3,x3app},{x4,x4app }},Gradient→GradS,
Method→{"Newton", Hessian→HessianS]]
{0.00330805, {2.21338×10^18,
{x1 → 596929.,x2 → -4847845.,
 x3 → 408822×10^6, x4 →13.4524}}}.
```

4.2 GNSS cycle ambiguities

Highly accurate static GNSS positioning in surveying is achieved by the processing of relative phase ranges observed to the visible GNSS satellites at both the reference and the rover stations. To eliminate the time-dependent error sources such as the satellite and receiver clock error, the double differenced phase observations are formed and they are adjusted using a least squares adjustment. An alternative technique is to use extended Kalman-filtering for this purpose.

The observation equation of the double differenced phase observations has the following form:

$$\Delta\Delta\Phi_{AB}^{jk} = a_1\delta x_B + a_2\delta y_B + a_3\delta z_B + \lambda N_{AB}^{jk} \tag{4.4}$$

where $\Delta\Delta\Phi_{AB}^{jk}$ is the double differences phase observations taken to the j-th and k-th satellite, $\delta x_B, \delta y_B, \delta z_B$ are the relative coordinate differences between the reference (A) and rover (B) stations, λ is the wavelength of the signal, N_{AB}^{jk} is the double differenced phase ambiguity and j is the so-called pivot satellite, that is used as a reference for forming the double differences.

Let us assume that five satellites are measured concurrently on both the reference and the rover stations in two consecutive epochs. Since one satellite is used as a pivot satellite, four double differences are formed in each epoch. This means that altogether 8 observation equations are formed, which can be used to evaluate 7 unknowns (3 coordinate differences and 4 double-differenced phase ambiguities). A usual solution of the problem is to estimate the unknowns using a least-squares adjustment, which provides a ‘float’ solution of the integer phase ambiguities. Consequently the computation of the integer least-squares estimates of the GNSS cycle ambiguities reduces to a integer least-squares problem (see Teunissen, 1995, 2012; Grafarend, 2003).

The mixed integer programming problem can be formulated as follows,

$$(y - Ax - Bz)^T Q^{-1} (y - Ax - Bz) \to \min_{x,z} , \tag{4.5}$$

where y, A, B and Q are known real vector and matrices, while x and z are integer unknown vectors, respectively,

$$x \in \mathbb{R} \text{ and } z \in \mathbb{Z}.$$

Let us consider the actual values of the input arrays as

$$y = \begin{pmatrix} 2.883 \\ 3.020 \\ 2.586 \\ 4.405 \\ 2.922 \\ 3.121 \\ 2.663 \\ 4.267 \end{pmatrix} \quad A = \begin{pmatrix} 0.25 & 0.20 & 0.45 \\ -0.30 & 0.42 & 0.56 \\ -0.34 & -0.20 & 0.78 \\ 0.22 & 0.54 & 0.33 \\ 0.28 & 0.23 & 0.40 \\ -0.31 & 0.47 & 0.60 \\ -0.31 & -0.22 & 0.87 \\ 0.20 & 0.48 & 0.30 \end{pmatrix};$$

$$B = \begin{pmatrix} 0.19 & & & \\ & 0.19 & & \\ & & 0.19 & \\ & & & 0.19 \\ 0.19 & & & \\ & 0.19 & & \\ & & 0.19 & \\ & & & 0.19 \end{pmatrix}$$

```
Q = IdentityMatrix[8];
```

Now, in order to convert the problem to pure integer programming, three new integer variables for elimination of the real unknown will be introduced as,

$$\xi_1 = \mathbf{100}x_1, \xi_2 = \mathbf{100}x_2, \xi_3 = \mathbf{100}x_3 .$$

So the real solutions for x_i will be considered to two decimal digits accuracy (cm). Let us rationalize the input arrays

```
{yR,QR,AR,BR} = Rationalize[{y,Q,A,B}]
```

$$\{\{\left\{\frac{2883}{1000}\right\},\left\{\frac{151}{50}\right\},\left\{\frac{1293}{500}\right\},\left\{\frac{881}{200}\right\},$$
$$\left\{\frac{1461}{500}\right\},\left\{\frac{3121}{1000}\right\},\left\{\frac{2663}{1000}\right\},\left\{\frac{4267}{1000}\right\}\},$$

```
{ {1,0,0,0,0,0,0,0},{0,1,0,0,0,0,0,0}
  {0,0,1,0,0,0,0,0},{0,0,0,1,0,0,0,0}
  {0,0,0,0,1,0,0,0},{0,0,0,0,0,1,0,0}
  {0,0,0,0,0,0,1,0},{0,0,0,0,0,0,0,1}},
```

$$\{\ \left\{\frac{1}{4},\frac{1}{5},\frac{9}{20}\right\},\left\{-\frac{3}{10},\frac{21}{50},\frac{14}{25}\right\}\ ,$$
$$\left\{-\frac{17}{50},-\frac{1}{5},\frac{39}{50}\right\},\left\{\frac{11}{50},\frac{27}{50},\frac{33}{100}\right\},$$
$$\left\{\frac{7}{25},\frac{23}{100},\frac{2}{5}\right\},\left\{-\frac{31}{100},\frac{47}{100},\frac{3}{5}\right\},$$
$$\left\{-\frac{31}{100},-\frac{11}{50},\frac{87}{100}\right\},\left\{\frac{1}{5},\frac{12}{25},\frac{3}{10}\right\}\ \}$$
$$\{\ \left\{\frac{19}{100},0,0,0\right\},\left\{0,\frac{19}{100},0,0\right\}\ ,$$
$$\left\{0,0,\frac{19}{100},0\right\},\left\{0,0,0,\frac{19}{100}\right\},$$
$$\left\{\frac{19}{100},0,0,0\right\},\left\{0,\frac{19}{100},0,0\right\}\ ,$$
$$\left\{0,0,\frac{19}{100},0\right\},\left\{0,0,0,\frac{19}{100}\right\},\ \}\ \}$$

Then the objective function to be minimized on the integer field is

```
objective = ((yR - AR.{ξ1 10^-2,ξ2 10^-2,ξ3 10^-2}-
  BR.{z1,z2,z3,z4}) / /Flatten).Inverse[QR];
objective =
  objective.((yR - AR.{ξ1 10^-2,ξ2 10^-2,ξ3 10^-2}-
  BR.{z1,z2,z3,z4}) / /Simplify / /First;
(1/100000000)(8710109300+7220000z1^2+7220000z2^2-
  199462000z3+7220000z3^2-329536000z4+7220000z4^2+
  433740ξ1-247000z3ξ1+159600z4ξ1+6271ξ1^2-
  14615460ξ2-159600z3ξ2+387600z4ξ2+3874ξ1ξ2+
  11006ξ1^2-3800z2(61410+61ξ1-89ξ2-116ξ3)-
  26195180ξ3+627000z3ξ3+239400z4ξ3-10636ξ1ξ3+
  13480ξ2ξ3+26003ξ3^2+3800z1(-58050+53ξ1+43ξ2+85ξ3))
```

First we solve the problem on the real field. The global minimum is

```
NMinimize[objective,{z1,z2,z3,z4,ξ1,ξ2,ξ3,ξ4}]
{3.11759×10^-6,{z1 → 10.1427,z2 → 11.7696,
  z3 → 13.8032,z4 → 16.0825,ξ1 → 109.707,
  ξ2 → 154.713,ξ3 → 82.8285
```

Now we are looking for the integer solution via extending the region of the constraints step by step until no further decreasing in the objective value occurs and while the solutions are inside the constrain regions. The first iteration is,

```
constraints =
  Apply[And,{9 < z1< 11,11 < z2< 12,
  13 < z3< 14,16 < z4< 17,109 <ξ1< 110,
  154 <ξ2< 155,82 <ξ3< 83,
  Element[{z1,z2,z3,z4,ξ1,ξ2,ξ3},Integers]}]
```

```
9 < z1 <11&& 11 <z2 <12&& 13 < z3 <14 &&
 16 < z4 <17 && 109 <ξ1 <110 &&
 154 <ξ2 <155 && 82 <ξ3 <83 &&
 (z1|z2|z3|z4|ξ1|ξ2|ξ3) ∈ Integers]

AbsoluteTiming[Minimize[objective,
 constraints,{z1,z2,z3,z4,ξ1,ξ2,ξ3}]//N]
{0.471928, {0.070728,
 {z1 →10., z2 →12.,z3 →14., z4 →17.,
  ξ1 →110.,ξ2 →155.,ξ3 →83.}}}
```

After six iterations,

```
constraints =
 Apply[And,{6 < z1 < 14,8 < z2 < 15,
 10 < z3 < 17,13 < z4 < 20,118 <ξ1 < 125,
 151 <ξ2 < 158,79 <ξ3 < 86,
 Element[{z1,z2,z3,z4,ξ1,ξ2,ξ3},Integers]}]
6 < z1 <14&& 8 <z2 <15&& 10 < z3 <17 &&
 13 < z4 <20 && 118 <ξ1 <125 &&
 151 <ξ2 <158 && 79 <ξ3 <86 &&
 (z1|z2|z3|z4|ξ1|ξ2|ξ3) ∈ Integers]

AbsoluteTiming[Minimize[objective,
 constraints,{z1,z2,z3,z4,ξ1,ξ2,ξ3}]//N]
{17.7336, {0.0005213,
 {z1 →10., z2 →12.,z3 →14., z4 →16.,
  ξ1 →122.,ξ2 →153.,ξ3 →83.}}}
```

In the last two iteration steps we have got the same objective value therefore $\mathbf{x}_1$=**1.22**, $\mathbf{x}_2$=**1.53** and $\mathbf{x}_3$=**0.83**. Blindly rounding of the real solution, we got incorrect solution for the first two coordinates $\mathbf{x}_1$=**1.10**, $\mathbf{x}_2$=**1.55**.

```
objective/.{z1 →10., z2 →12.,z3 →14.,
 z4 →16., ξ1 →122.,ξ2 →153.,ξ3 →83.}
0.00844597
```

Remarks
Employing 3 decimal digits approximation, namely

$$\xi_1 = 1000\mathbf{x}_1,\ \xi_2 = 1000\mathbf{x}_2,\ \xi_3 = 1000\mathbf{x}_3$$

we can see better results, see Paláncz (2018).

```
constraints =
 Apply[And,{6 < z1 < 14,8 < z2 < 15,
 10 < z3 < 17,13 < z4 < 20,118 <ξ1 < 125,
 151 <ξ2 < 158,79 <ξ3 < 86,
 Element[{z1,z2,z3,z4,ξ1,ξ2,ξ3},Integers]}]
6 < z1 <14&& 8 <z2 <15&& 10 < z3 <17 &&
 13 < z4 <20 && 118 <ξ1 <125 &&
 151 <ξ2 <158 && 79 <ξ3 <86 &&
 (z1|z2|z3|z4|ξ1|ξ2|ξ3) ∈ Integers]

AbsoluteTiming[Minimize[objective,
 constraints,{z1,z2,z3,z4,ξ1,ξ2,ξ3}]//N]
{544.533, {0.0000191054,
 {z1 →10., z2 →12.,z3 →14., z4 →16.,
  ξ1 →1213.,ξ2 →1533.,ξ3 →825.}}}
```

Since the objective function is a second order polynomial and the constraints are linear, the method will always find the global minimum.

4.3 Kepler's third law

Symbolic regression is a type of regression analysis that searches the space of mathematical expressions to find the model that best fits a given dataset, both in terms of accuracy and simplicity. No particular model is provided as a starting point to the algorithm. Instead, initial expressions are formed by randomly combining mathematical building blocks such as mathematical operators, analytic functions, constants, and state variables. (Usually, a subset of these primitives will be specified by the person operating it, but that is not a requirement of the technique.) New equations are then formed by recombining previous equations. To select the optimal set of basic functions, Koza (1992) suggested employment of genetic programming (GP). GP is a biologically inspired machine learning method that evolves computer programs to perform a task. In order to carry out genetic programming, the individuals (competing functions) should be represented by a binary tree. In standard GP, the leaves of the binary tree are called terminal nodes represented by variables and constants, while the other nodes, the so called non-terminal nodes are represented by functions. Since the candidate models can be computed independently, parallel computation is utilized. Complexity and fitness are conflicting features leading to a multi-objective problem. A useful expression is both predictive and parsimonious. Some expressions may be more accurate but over-fit the data, whereas others may be more parsimonious but oversimplify. The prediction error versus complexity or $1-f$ (fitness) versus complexity of the Pareto front represent the optimal solutions as they vary over expression complexity and maximum prediction error. As Fig. 4.1 shows, functions representing the Pareto front have the following features:

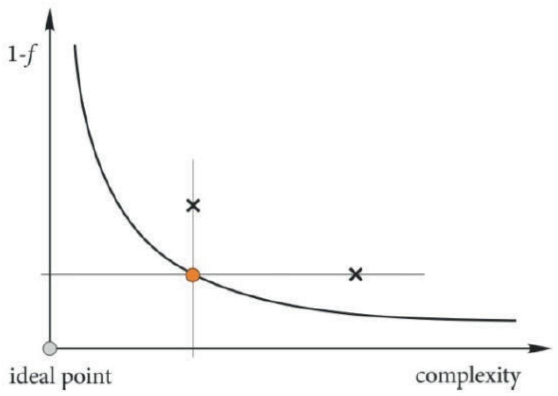

Figure 4.1: The Pareto front

As an illustration let us consider the Kepler problem. The third law of Kepler states: „The square of the orbital period of a planet is directly proportional to the cube of the semi-major axis of its orbit (average distance from the Sun).“

$$P^2 \propto a^3 \tag{4.6}$$

where P is the orbital period of the planet and a is the semi-major axis of the orbit. For example, suppose planet A is 4 times as far from the Sun as planet B. Then planet A must traverse 4 times the distance of planet B for each orbit, and moreover it turns out that planet A travels at half the speed of planet B, in order to maintain equilibrium with the reduced gravitational centripetal force due to being 4 times further from the Sun. In total it takes $4 \times 2 = 8$ times as long for planet A to travel an orbit, in agreement with the law ($8^2 = 4^3$).
The third law currently receives additional attention as it can be used to estimate the distance from an exoplanet to its central star, and help to decide if this distance is inside the habitable zone of that star.
The exact relation, which is the same for both elliptical and circular orbits, is given by the formula above. This third law used to be known as the harmonic law, because Kepler enunciated it in a laborious attempt to determine what he viewed as the „music of the sphere“according to precise laws, and express it in terms of musical notation. His result is based on the Rudolphine table containing the observations of Tycho Brache 1605, see Table 4.1 where a is given in units of Earth's semi-major axis. Let us assume that Kepler could have employed one of the function approximation techniques like polynomial regression, artificial neural networks, support vector machine, thin plate spline. Could he find this simple relation with these sophisticated methods? Surprisingly the answer is no. But symbolic regression will work. Fig. 4.2 shows the Pareto-front of the generated models via DataModeler (2018). The points represent the generated models. The red points stand for the models belonging to the Pareto-front. In Table 4.2 some of the models of the Pareto front can be seen.

Table 4.1: Normalized observation planetary data

Planet	Period P (yr)	Semimajor axis a
Mercury	0.24	0.39
Venus	0.61	0.72
Earth	1.00	1.00
Mars	1.88	1.52
Jupiter	11.86	5.20
Saturn	29.46	9.54
Uranus	84.01	19.19
Neptune	164.79	30.06
Pluto	284.54	39.53

It goes without saying that our candidate is the 4-th model, since it has a small error and at the same time its complexity is low. In Table 4.3 we can see the statistics of relative errors of the different techniques in percentage units.
It is inevitable, that statistically the best model is provided by the symbolic regression. Even though its mean error is higher than that of the Kepler solution, it is simple in practice.

5 Conclusion

The term Hybrid Symbolic-Numeric Computation (HSNC) has been with us for over two decades now. We anticipate the day when it falls into disuse, not because the technology goes out of style, but rather, since it is just an integral part of the plumbing of mathematical computation. Further geodetic solutions using HSNC are presented in the book of Awange et al. (2018).

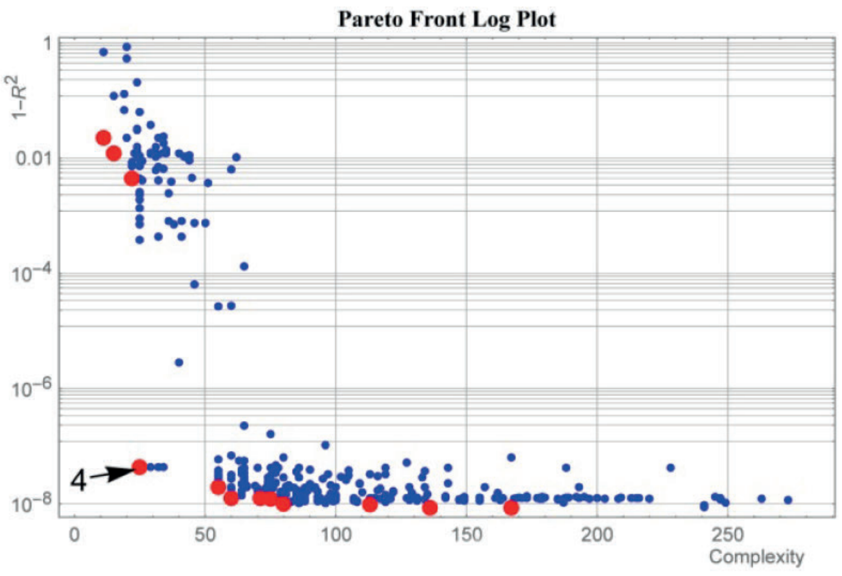

Figure 4.2: The Pareto front (red points) and the evaluated models in case of the Kepler's problem

Table 4.2: Model selection report

Model	Complexity	$1-R^2$	Function
1	11	0.022	$-12.550 + 6.116x$
2	15	0.012	$7.168 + 0.162x^2$
3	22	0.004	$3.174 + 0.135\cdot(-8.024 - x)x$
4	25	$4.284\cdot10^{-8}$	$-0.006 + 1.000\sqrt{x}$
5	55	$1.927\cdot10^{-8}$	$0.009 + 0.0006\cdot(-x + 157.399x^{3/2})$
6	60	$1.243\cdot10^{-8}$	$0.005 + 2.64910\cdot10^{-4}(3768.96x^{3/2} + 2x^2)$
7	71	$1.242\cdot10^{-8}$	$0.004 + 1.25410\cdot10^{-4}(7960.56x^{3/2} + 2x^2)$
8	75	$1.236\cdot10^{-8}$	$0.004 + 1.25410\cdot10^{-4}(x + 7960.56x^{3/2} + 2x^2)$
9	80	$1.001\cdot10^{-8}$	$-0.002 + 0.007\ (\sqrt{x} + 140.884x^{3/2} + 0.049x^2)$
10	113	$9.624\cdot10^{-9}$	$0.028 - 0.002\cdot(5.674/x + 5x - 478.651x^{3/2} + 1/(-9.892 + 1/x + x))$
11	136	$8.632\cdot10^{-9}$	$-0.020 + 0.009\cdot(-x + 115.915x\ \sqrt{x+\sqrt{x}/(12+2x+x^2)})$

Table 4.3: Statistics of the relative error (%) of the different approximation methods

Method	Mean error (%)	Max Error (%)	Standard deviation (%)
Polynomial Regression	25.14	177.49	59.96
Neural Network	26.80	191.34	64.22
Support Vector Machine	8.89	47.47	16.14
Thin Plate spline	29.10	149.87	49.59
Kepler solution	0.23	1.48	0.51
Symbolic Regression	0.32	0.84	0.37

References

Awange, J. L. (2012): Environmental monitoring using GNSS. Springer, Berlin. DOI: 10.1007/978-3-540-88256-5.

Awange, J. L. (2018): GNSS environmental sensing. 2nd Edition. Springer International Publishers. DOI: 10.1007/978-3-319-58418-8.

Awange, J. L., Grafarend, E. W., Paláncz, B., and Zaletnyik, P. (2010): Algebraic Geodesy and Geoinformatics. 2nd ed. Springer, Heidelberg, New York.

Awange, J. L. and Paláncz, B. (2016): Geospatial Algebraic Computations-Theory and Applications. 3rd ed. Springer, Heidelberg, New York.

Awange, J. L., Paláncz, B., Lewis, R., and Völgyesi, L. (2018): Mathematical Geosciences - Hybrid symbolic - numeric methods. Springer, Heidelberg, New York.

DataModeler (2018): symbolic regression package available for Mathematica. URL: http://www.evolved-analytics.com (visited on 01/01/2018).

Grafarend, E. W. (2003): Mixed Integer-Real Valued Adjustment (IRA) Problems: GPS Initial Cycle Am- biguity Resolution by Means of the LLL Algorithm. In: *Geodesy-The Challenge of the 3rd Millennium*. Ed. by E. W. Grafarend, F. W. Krumm, and V. S. Schwarze. URL: https://link.springer.com/chapter/10.1007%2F978-3-%20662-05296-9_32.

Koza, J. R. (1992): Genetic Programming, On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (Massachusetts) USA.

Lichtblau, D. (2013): Approximate Gröbner Bases, Overdetermined Polynomial Systems, and Approximate GCDs. *ISRN Computational Mathematics* 352806:1–12. DOI: 10.1155/2013/352806. URL: https://www.risc.jku.at/Groebner-Bases-Bibliography/gbbib_files/publication_2822.pdf.

Paláncz, B. (2018): Numeric-symbolic solution of GPS phase ambiguity problem with Mathematica. *Wolfram Archive* 2018. URL: http://library.wolfram.com/infocenter/MathSource/9705/.

Sasaki, T. (2014): A Practical Method for Floating- Point Gröbner Basis Computation. In: *Computer Mathematics*. Ed. by R. Feng, W. Lee, and Y. Sato. Berlin, Heidelberg: Springer, pp. 109–124.

Szanto, A. (2011): Hybrid Symbolic-numeric Methods for the Solution of Polynomial Systems: Tutorial Overview. In: Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation, *ISSAC '11, San Jose, California, USA, June 08 - 11, 2011*. New York, NY, USA: ACM, pp. 9–10. ISBN: 978-1-4503-0675-1. DOI: 10.1145/1993886.1993893.

Teunissen, P. J. G. (1995): The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. *Journal of Geodesy* 70:65–82.

Teunissen, P. J. G. (2012): Towards a Unified Frame- work for GNSS Ambiguity Resolution: Problems & Solutions. Tech. rep. URL: https://www.nav.ei.tum.de/fileadmin/w00bkq/www/Colloquium/colloquium_teunissen_slides.pdf.