Celestial Machine Learning
From Data to Mars and Beyond with AI Feynman

Zi-Yu Khoo[1], Abel Yang[1], Jonathan Sze Choong Low[2], and Stéphane Bressan[1]

[1] National University of Singapore. 21 Lower Kent Ridge Rd, Singapore 119077
khoozy@comp.nus.edu.sg, phyyja@nus.edu.sg, steph@nus.edu.sg
[2] Agency for Science, Technology and Research (A*STAR), Singapore Institute of
Manufacturing Technology (SIMTech), Singapore 138634, Singapore
sclow@simtech.a-star.edu.sg

Abstract. Can machine learning discover Kepler's first law from data? We emulate Johannes Kepler's discovery of the equations of the orbit of Mars with the Rudolphine tables using AI Feynman, a physics-inspired tool for symbolic regression.

1 Introduction

In 2020, Silviu-Marian Udrescu and Max Tegmark introduced *AI Feynman* [16], a symbolic regression algorithm that could rediscover from data one hundred equations from the *Feynman Lectures on Physics* [3]. Although the authors motivated their work with the example of Johannes Kepler's successful discovery of the orbital equations of Mars, to our knowledge, they did not report an attempt to rediscover it with their algorithm. We show that AI Feynman can emulate Kepler's discovery of the orbital equation of Mars from the Rudolphine tables.

The discovery of Kepler's laws of planetary motion illustrates of the process of science, encapsulating the principles of parsimony and physical considerations. Prior to Kepler, early astrologers such as Nicolaus Copernicus and Tycho Brahe hypothesized various models to explain the movement of celestial bodies. Armed with the Prutenic tables, Copernicus modelled the orbit of Mars as heliocentric, with a deferent having two epicycles [12]. However, Kepler, assistant to Tycho Brahe, had access to the best available data collected in Europe. In 1627, Kepler compiled Brahe's sightings of Mars into a set of 180 heliocentric data of the position of Mars in the Rudolphine tables. The translation of sightings to the Rudolphine tables only embeds assumptions of heliocentrism and planarity of the orbit. Kepler could have described the motions of Mars as an oval or added additional epicycles to the Copernician model, but instead described it as elliptical in *Astronomia nova* in 1609 [8]. We use AI Feynman to emulate Kepler's discovery of the elliptical orbital equation of Mars, from the Rudolphine tables.

The Rudolphine tables already embed assumptions regarding planarity and heliocentricity of Mars' orbit. To make further inferences using AI Feynman, we can add biases to the data regarding its physical units. We have four experiments. In the first experiment, AI Feynman is oblivious to biases. In the second

and third experiments, AI Feynman is biased through transforming data which are angles and limiting the search space for orbital equations respectively. The fourth experiment combines the biases of the second and third. AI Feynman benefits from these biases, and produces the best result in the fourth experiment. Information regarding the physical units of the data likely also guided Kepler in his discovery of the elliptical orbit of Mars, and in this way, AI Feynman emulates Kepler's discovery from data in the Rudolphine tables. In this paper, we present the design, results and discussion of our experiments with AI Feynman.

2   Related Work

Finding an equation describing the orbit of Mars is a combinatorial challenge. The task is NP-hard in principle due to its exponentially large search space of equations [16]. To circumvent this, one may use universal function approximators such as multilayer perceptron neural networks [5]. Alternatively, symbolic regressions search for a parsimonious and elegant form of the unknown equation.

There are three main classes of symbolic regression methods [10]: regression-based, expression tree-based and physics- or mathematics-inspired. We use AI Feynman, a machine learning and physics-inspired algorithm [16].

Regression-based symbolic regression methods [10], given solutions to the unknown equation, find the coefficients of a fixed basis that minimise the prediction error. As the basis grows, the fit improves, but the functional form of the unknown equation becomes less sparse or parsimonious. Sparse regressions promote sparsity through regularisation, as proposed by Robert Tibshirani [15] who used the l_1 norm, thus inventing the Lasso regression. A state-of-the-art sparse symbolic regression approach is the Sparse Identification of Nonlinear Dynamics by Steven Brunton et al. in [1]. It leverages regularisation and identifies equations of motion of a system using a sparse regression over a chosen basis.

Committing to a basis limits the applicability of regression-based methods. Expression tree-based symbolic regression methods based on genetic programming [10] can instead discover the form and coefficients of the unknown equation.

Seminal work by John Koza et al. [9] represented each approximation of an unknown equation as a genetic programme with a tree-like data structure, with traits (or nodes in the tree) representing functions or operations, and variables representing real numbers. The fitness of each genetic programme is its prediction error. Fitter genetic programmes undergo a set of transition rules comprising selection, crossover and mutation to iteratively find the optimal equation form.

Genetic programmes may greedily mimic nuances of the unknown equation [14], limiting generalisability. David Goldberg [4] used Pareto optimisation to balance the objectives of fit and parsimony in symbolic regression. In each iteration, the fittest genetic programmes lie on the non-dominated Pareto-frontier. State-of-the-art symbolic regression using genetic programming include `Eureqa` by Michael Schmidt and Hod Lipson [13] and `PySR` by Miles Cranmer [2].

Expression tree-based methods do not guarantee that more accurate approximations of an equation are symbolically closer to the truth. If an expression tree-

based method finds a reasonably accurate equation with wrong functional form, it risks getting stuck near a local optimum [16]. Functions of practical interest in physics exhibit simplifying properties such as symmetry or separability [16]. Physics-inspired symbolic regression methods leverage these simplifying properties to guarantee taking a step in the right direction. Udrescu and Tegmark [16] use a neural network to test data describing the unknown equation for simplifying properties and recursively break the unknown equation into simpler unknown equations with fewer variables to solve [16]. Each simpler unknown equation can be solved by regression with a basis set of non-linear functions. AI Feynman then outputs a sequence of increasingly complex equations that provide progressively better accuracy, along a Pareto front, based on work by Goldberg [4] and Guido Smits [14]. We use AI Feynman to rediscover the orbital equation of Mars.

3   Methodology

In publishing the Rudolphine tables, Kepler had already assumed planarity and heliocentricity of the orbit of Mars. We experiment if AI Feynman performs better with biases, based on our knowledge of the physical units of the data.

Informing a learning algorithm of physics amounts to introducing appropriate biases that can steer the learning process towards identifying physically consistent solutions according to George Karnadiakis [6]. Karniadakis identifies three types of bias: observational, inductive and learning biases. Observational biases are introduced directly through data that embody the underlying physics or carefully crafted data augmentation procedures. Training a machine learning model on such data allows it to learn an output that reflects the physical structure of the data. Inductive biases correspond to prior assumptions incorporated by tailored interventions to a machine learning model architecture, so predictions are guaranteed to satisfy a set of given physical laws. Learning biases can be introduced by appropriate choice of loss functions, constraints and inference algorithms that can modulate the training phase of an machine learning model to explicitly favour convergence towards solutions that adhere to the underlying physics. We consider the introduction of observational and inductive biases.

With knowledge that data is known to be an angle, only trigonometric functions can transform it. We introduce an observational bias by applying the sine and cosine functions to inputs of the unknown equation that are known to be angles. The resulting numerical values, hence embodying the underlying periodicity of the data, are input to AI Feynman. The observational bias guides AI Feynman in finding an equation that reflects the periodic nature of Mars' orbit.

With knowledge that data are physical quantities, they cannot be transformed by exponential and logarithmic functions, as these only transform dimensionless quantities. We introduce an inductive bias by eliminating candidate functions. For each simpler recursive unknown equation AI Feynman has to solve, it transforms equations in the current Pareto front by one of several non-linear functions. These non-linear functions include exponential, logarithmic, trigono-

metric, polynomial and radical functions. The inductive bias limits the search space to trigonometric, polynomial and radical functions.

We conduct four experiments corresponding to four possible combinations of observational and inductive biases with the AI-Feynman algorithm. Experiment 1 does not use any bias. Experiment 2 and 3 only use observational and inductive bias respectively. Experiment 4 uses both observational and inductive bias.

While AI Feynman explores the Pareto front, Kepler may have instead made use of thought experiments to hypothesize an elliptical orbit. Fitting data from the Rudolphine tables to the equation of an ellipse using non-linear least squares returns the coefficients representing the eccentricity and semi-major axis. These are 0.0926 and 1.5235 respectively. For reference, the National Aeronautics and Space Administration suggest an 0.0934 and 1.5237 respectively [11].

4   Performance Evaluation

In the Rudolphine tables [7], the table titled *Tabula Aequationum MARTIS*, or Table of Corrections for Mars, contains four columns of data: *Anomalia eccentri*, *Intercolumnium*, *Anomalia coaequata*, and *Intervallu*. These columns represent the eccentric anomaly, an interpolating factor, the coequated or true anomaly, and the distance between the Sun and Mars respectively. The full Rudolphine tables (snippet found in Figure 1) was digitised for this experiment.

Fig. 1: The four columns of data provided in the Rudolphine Tables.

We apply AI Feynman to the data of *Intervallu* and *Anomalia coaequata* to recover the equation of orbit for Mars. *Anomalia coaequata* is an angle in degrees minutes seconds, which we convert to decimal degrees. *Intervallu* is the distance between the Sun and Mars, which we scale from a magnitude of $E+05$ to $E+00$. The code for AI Feynman, with minor modifications to embed observational and inductive biases, is at https://github.com/zykhoo/AI-Feynman.

We compare equations along the AI Feynman Pareto frontier with the orbital equation of Mars in Equation 0. r is the *Intervallu* and θ is the *Anomalia coaequata*. ϵ is the eccentricity of the ellipse, and a is the semi-major axis.

$$r = \frac{a}{1 + \epsilon \times cos\theta} \tag{0}$$

We also present the mean description length loss[16,17] (DL) computed between each predicted and true *Intervallu*. It minimises the geometric mean instead of the arithmetic mean, which encourages improving already well-fit points [17].

For Experiments 1 and 3, the inputs to AI Feynman are θ and r. For Experiments 2 and 4, the inputs to AI Feynman are $\cos(\theta)$, $\sin(\theta)$ and r. We traverse the equations along the Pareto frontier returned by AI Feynman in increasing goodness of fit and increasing complexity (or equivalently decreasing parsimony) and present them. The results of Experiments 1, 2, 3 and 4 are presented in Tables 1, 2, 3 and 4 respectively. We omit results independent of θ.

Eqn No.	Equation	DL
(1a)	$r = \dfrac{4}{3} - 0.09 \times \theta^2$	24.976
(1b)	$r = (2.78 - 0.26 \times \theta^2)^0.5$	24.926
(1c)	$r = \arccos\left(-0.02 \times \theta^3 + 0.09 \times \theta^2 - 0.1\right)$	23.577
(1d)	$r = \dfrac{1}{-0.01 \times \theta^3 + 0.04 \times \theta^2 + 0.6}$	22.515
(1e)	$r = (0.01 \times \theta^3 - 0.04 \times \theta^2 + 1.29)^2$	22.273
(1f)	$r = \arccos\left(-0.02 \times \theta^3 + 0.09 \times \theta^2 + 0.01 \times \theta - 0.1\right)$	21.356
(1g)	$r = \log\left(0.09 \times \theta^3 - 0.38 \times \theta^2 - \dfrac{1}{9} \times \theta + 5.3\right)$	20.841
(1h)	$r = 0.02 \times \theta^3 - 0.09 \times \theta^2 - 0.01 \times \theta + 1.67$	20.238

Table 1: Results of Experiment 1, which took 748 seconds.

Eqn No.	Equation	MSE
(2a)	$r = \log\cos\theta + 5$	26.006
(2b)	$r = \dfrac{1}{7} \times \cos\theta + 1.5$	24.053
(2c)	$r = 1.5 \times \exp\{0.1 \times \cos\theta\}$	23.512
(2d)	$r = \dfrac{1}{\frac{2}{3} - 0.0556244812357114 \times \cos\theta}$	22.857
(2e)	$r = 1.5119670200057298 \times \exp\{(0.1 \times \cos\theta)\}$	22.457
(2f)	$r = 1.510965630582 + (\cos\theta/(\sin\theta + 6))$	21.070
(2g)	$r = 1.51366746425629 \times exp(0.0931480601429939 \times \cos\theta)$	20.762
(2h)	$r = \dfrac{1}{0.662428796291351 - 0.0612906403839588 \times \cos\theta}$	19.781
(2i)	$r = (0.662428796291351 - 0.0612906403839588 \times \cos\theta)^{-1.00133872032166}$	12.211

Table 2: Results of Experiment 2, which took 1451 seconds.

Experiment 1 does not use any bias. The search space is big and AI Feynman does not find an equation form that matches the orbital equation of Mars along its Pareto frontier. In Experiment 2, AI Feynman makes use of an observational bias. As the input data to AI Feynman embodies the underlying periodicity of the data, AI Feynman can use this information to guide its search for an equation that reflects the periodic structure of Mars' orbit. Therefore, three out of nine equations along the Pareto front have an equation form that matches the true orbit of Mars. These are Equations 2d, 2h and 2i. In Experiment 3, AI Feynman makes use of an inductive bias. While the search space for AI Feynman is smaller, an inductive bias does not guide its search for an equation

Eqn No.	Equation	MSE
(3a)	$r = \dfrac{4}{3} - 0.09 \times \theta^2$	24.976
(3b)	$r = (2.78 - 0.25 \times \theta^2)^0.5$	24.842
(3c)	$r = \arccos\left(-0.02 \times \theta^3 + 0.09 \times \theta^2 - 0.1\right)$	23.577
(3d)	$r = \dfrac{1}{-0.01 \times \theta^3 + 0.04 \times \theta^2 + 0.6}$	22.515
(3e)	$r = (0.01 \times \theta^3 - 0.04 \times \theta^2 + 1.29)^2$	22.273
(3f)	$r = \arccos\left(-0.02 \times \theta^3 + 0.09 \times \theta^2 + 0.01 \times \theta - 0.1\right)$	21.356
(3g)	$r = 0.02 \times \theta^3 - 0.09 \times \theta^2 - 0.01 \times \theta + 1.67$	20.238

Table 3: Results of Experiment 3, which took 621 seconds.

Eqn No.	Equation	MSE
(4a)	$r = \dfrac{1}{7} \times \cos\theta + 1.5$	24.053
(4b)	$r = \cos\theta/(\sin\theta + 6) + 1.5$	23.617
(4c)	$r = \arccos(0.0420224035468255 - 0.142857142857143 \times \cos\theta)$	23.392
(4d)	$\mathbf{r = \dfrac{1}{\frac{2}{3} - 0.0566732120453772 \times \cos\theta}}$	22.575
(4e)	$r = 1.511006320056 + (\cos\theta/(\sin\theta + 6))$	21.089
(4f)	$r = \tan(0.0425049090340329 \times \cos\theta + 0.986141372332807)$	20.057
(4g)	$r = \tan(0.0427569970488548 \times \cos\theta + 0.98658412694931)$	20.021
(4h)	$\mathbf{r = \dfrac{1}{0.662420213222504 - 0.0612917765974998 \times \cos\theta}}$	19.747
(4i)	$\mathbf{r = (0.662420213222504 - 0.0612917765974998 \times \cos\theta)^{-1.00130701065063}}$	12.208

Table 4: Results of Experiment 4, which took 1184 seconds.

that reflects the periodic structure of Mars' orbit. Therefore AI Feynman does not find an equation form that matches the orbital equation of Mars along its Pareto frontier. In Experiment 4, AI Feynman makes use of both an observational and an inductive bias. AI Feynman makes use of the observational bias to guide its search for an equation that reflects the periodic structure of Mars' orbit. It also makes use of the inductive bias to limit the search space, resulting in fewer equations along the Pareto front. Therefore, three out of ten equations along the Pareto front have an equation form that matches the true orbit of Mars. These are Equations 4d, 4h and 4i.

Experiments 1 and 2 highlight the importance of an observational bias in guiding AI Feynman. In Experiment 1, none of the equations along the Pareto front match the orbital equation for Mars, compared to three out of eleven equations along the Pareto front for Experiment 2. However, as the observational bias doubles the number of inputs to AI Feynman, it takes approximately twice as long to run. This is because AI Feynman recurses through each input. The depth of the recursion is doubled when the number of inputs is doubled.

Experiments 2 and 4 highlight the importance of an inductive bias in limiting the search space for AI Feynman. In Experiment 2, we can observe many equations have one of two common forms. Three of them utilise an exponential function applied to $\cos\theta$ (Equations 2c, 2e and 2g). Another three utilise an inverse function applied to $\cos\theta$ (Equations 2d, 2h and 2i), which matches the true orbit of Mars. However, in Experiment 4, the equation form with an inverse function applied to $\cos\theta$ in (Equations 4d, 4h and 4i) matches the true obit of Mars, and is also the most prevalent. Therefore, AI Feynman, augmented with both an observational and inductive bias, is best able to rediscover Kepler's first

law for the orbit of Mars. The inductive bias also reduces the time taken for AI Feynman to run. This is because the search space for AI Feynman is limited, and time is saved from having to search a smaller search space.

We also observe that Equations 2h, 2i, 4h, and 4i, with forms that match the true orbit of Mars, have the lowest mean description length loss of less than 20.

Lastly, we observe that Equations 2h and 4h suggest $a = 1.52$ and $\epsilon = 0.0925$ similar to the values suggested in the Rudolphine tables.

5   Conclusion

We have successfully shown that AI Feynman can rediscover from the Rudolphine Tables the equation of Kepler's first law for the planet Mars, given information regarding physical quantities of the data in the form of observational and inductive biases. The discovery of physical laws is a bi-optimisation problem of parsimony and accuracy, that can be guided by physical units of the data available. AI Feynman is able to emulate Kepler's discovery of the orbital equation of Mars because it implements an optimisation of both parsimony and accuracy, and can be guided by information regarding the physical units of the data.

As future work, we are looking into how AI Feynman can repeat this discovery process directly from sightings of Mars and the Sun from Earth. We use a modern reproduction of these sightings from the National Aeronautics and Space Administration's Horizons system. This challenges AI Feynman to perform a change from the geocentric to heliocentric reference frame, and we are investigating how this change can be incorporated within its algorithm. Lastly, we are experimenting with the planet Mercury, which has a precessing orbit.

References

1. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences **113**(15), 3932–3937 (2016)
2. Cranmer, M.: Pysr: Fast & parallelized symbolic regression in python/julia (Sep 2020), `http://doi.org/10.5281/zenodo.4041459`
3. Feynman, R.P.: The Feynman lectures on physics. Reading, Mass. : Addison-Wesley Pub. Co., c1963-1965. (1965 c1963)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edn. (1989)
5. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5), 359–366 (Jan 1989)
6. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nature Reviews Physics **3**(6), 422–440 (Jun 2021)
7. Kepler, J., Brahe, T., Eckebrecht, P.: Tabulæ Rudolphinæ, quibus astronomicæ scientiæ, temporum longinquitate collapsæ restauratio continetur (1627)
8. Kepler, J., Donahue, W.H.: New Astronomy. Cambridge University Press (1992)
9. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA (1992)

10. Makke, N., Chawla, S.: Interpretable Scientific Discovery with Symbolic Regression: A Review (11 2022)
11. National Aeronautics and Space Administration: Mars fact sheet. `https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html` (2022)
12. Rosen, E.: The commentariolus of copernicus. Osiris **3**, 123–141 (1937). https://doi.org/10.1086/368473, `https://doi.org/10.1086/368473`
13. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. Science **324**(5923), 81–85 (2009)
14. Smits, G.F., Kotanchek, M.: Pareto-Front Exploitation in Symbolic Regression, pp. 283–299. Springer US, Boston, MA (2005)
15. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B **58**, 267–288 (1994)
16. Udrescu, S.M., Tegmark, M.: AI Feynman: A physics-inspired method for symbolic regression. Science Advances **6**(16) (2020)
17. Wu, T.: Intelligence, physics and information – the tradeoff between accuracy and simplicity in machine learning (2020)

Appendix

Eqn No.	Eqn	MSE
1	$asin(-666.000000000000 \times ((sin(pi) + 1) - 1))$	2.33
2	1.50000000000000	0.0106
3	$pi/2$	0.0123
4	1.65306122448980	0.0268
5	$1.66666666666667 - 0.09 \times x_0^2$	0.048
6	$(2.78 - 0.26 \times x_0^2)^0.5$	0.0767
7	$acos(-0.02 \times x_0^3 + 0.09 \times x_0^2 - 0.1)$	0.000169
8	$1/(-0.01 \times x_0^3 + 0.04 \times x_0^2 + 0.6)$	0.000706
9	$(0.01 \times x_0^3 - 0.04 \times x_0^2 + 1.29)^2$	0.000458
10	$acos(-0.02 \times x_0^3 + 0.09 \times x_0^2 + 0.01 \times x_0 - 0.1)$	4.7e-05
11	$log(0.09 \times x_0^3 - 0.38 \times x_0^2 - 0.111111111111111 \times x_0 + 5.3)$	1.07e-05
12	$0.02 \times x_0^3 - 0.09 \times x_0^2 - 0.01 \times x_0 + 1.67$	4.41e-05

Table 5: Experiment 1 Results

Eqn No.	Eqn	MSE
1	1.50000000000000	0.0106
2	$log(x_0 + 5)$	0.0092
3	$0.142857142857143 \times x_0 + 1.5$	0.000309
4	$1.5 \times exp(0.1 \times x_0)$	0.00021
5	$acos(0.0418258837357514 - 0.142857142857143 \times x_0)$	0.000166
6	$1/(0.666666666666667 - 0.0571196591636008 \times x_0)$	0.000212
7	$1.5082674607662332 \times exp(0.1 \times x_0)$	7.44e-05
8	$1.510965630582 + (x_0/((((((x_1 + 1) + 1) + 1) + 1) + 1) + 1))$	0.000179
9	$1.51366746425629 \times exp(0.0931480601429939 \times x_0)$	5.39e-06
10	$tan(0.0427570976316929 \times x_0 + 0.986583888530731)$	1.98e-06
11	$tan(0.0428397443727006 \times x_0 + 0.986126406475687)$	4.21e-06
12	$1/(0.662416338920593 - 0.0612923018634319 \times x_0)$	7.26e-07
13	$(0.662416338920593 - 0.0612923018634319 \times x_0)^{(} - 1.0012925863266)$	6.01e-10

Table 6: Experiment 2 Results

Eqn No.	Eqn	MSE
1	0	2.33
2	1.50000000000000	0.0106
3	$pi/2$	0.0123
4	1.65306122448980	0.0268
5	$1.66666666666667 - 0.09 \times x_0^2$	0.048
6	$(2.78 - 0.26 \times x_0^2)^0.5$	0.0767
7	$acos(-0.02 \times x_0^3 + 0.09 \times x_0^2 - 0.1)$	0.000169
8	$1/(-0.01 \times x_0^3 + 0.04 \times x_0^2 + 0.6)$	0.000706
9	$(0.01 \times x_0^3 - 0.04 \times x_0^2 + 1.29)^2$	0.000458
10	$acos(-0.02 \times x_0^3 + 0.09 \times x_0^2 + 0.01 \times x_0 - 0.1)$	4.7e-05
11	$(0.06 \times x_0^3 - 0.26 \times x_0^2 - 0.05 \times x_0 + 2.78)^0.5$	5.61e-06
12	$0.02 \times x_0^3 - 0.09 \times x_0^2 - 0.01 \times x_0 + 1.67$	4.41e-05

Table 7: Experiment 3 Results

Eqn No.	Eqn	MSE
1	0	2.33
2	1.50000000000000	0.0106
3	1.56250000000000	0.0115
4	$0.142857142857143 \times x_0 + 1.5$	0.000309
5	$x_0/(x_1 + 6) + 1.5$	0.000416
6	$1/(0.666666666666667 - 0.0557172402393568 \times x_0)$	0.000265
7	$1.510957104465 + (x_0/((((((x_1 + 1) + 1) + 1) + 1) + 1) + 1))$	0.000179
8	$1/(0.662428081035614 - 0.0612907484173775 \times x_0)$	7.75e-07
9	$0.140863761305809 \times x_0 - 0.0146051803603768 \times x_1 + 1.52623379230499$	1.1e-06
10	$(0.662428081035614 - 0.0612907484173775 \times x_0)^{(} - 1.001335978508)$	6.01e-10

Table 8: Experiment 4 Results

Fig. 2: A plot of the results of Experiment 1. The equations correspond to those presented in Table 1. The y-axis represents the *Intervallu* and x-axis represents the *Anomalia coaequata*. The true values for the *Intervallu* from the Rudolphine tables are also plotted and labeled "original".

Fig. 3: A plot of the results of Experiment 2. The equations correspond to those presented in Table 2. The y-axis represents the *Intervallu* and x-axis represents the *Anomalia coaequata*. The true values for the *Intervallu* from the Rudolphine tables are also plotted and labeled "original".

Fig. 4: A plot of the results of Experiment 3. The equations correspond to those presented in Table 3. The y-axis represents the *Intervallu* and x-axis represents the *Anomalia coaequata*. The true values for the *Intervallu* from the Rudolphine tables are also plotted and labeled "original".

Fig. 5: A plot of the results of Experiment 4. The equations correspond to those presented in Table 4. The y-axis represents the *Intervallu* and x-axis represents the *Anomalia coaequata*. The true values for the *Intervallu* from the Rudolphine tables are also plotted and labeled "original".