Unit-aware Multi-objective Genetic Programming for the Prediction of the Stokes Flow around a Sphere

Heiner Zille
Sanaz Mostaghim
Otto von Guericke University Magdeburg
{heiner.zille|sanaz.mostaghim}@ovgu.de

Fabien Evrard
Berend van Wachem
Otto von Guericke University Magdeburg
{fabien.evrard|berend.vanwachem}@ovgu.de

ABSTRACT

In this article we apply a unit-aware Genetic Programming (GP) approach to solve a problem from the area of fluid-dynamics: The Stokes flow around a sphere. We formulate 6 test instances with different complexities and explore the capabilities of single- and multi-objective GP variants to solve this problem with physically correct units of measurement. The study is a starting point to investigate the amount of information necessary to solve fluid-dynamics-related problems, and whether the inclusion of physical dimensions is advantageous or not for such optimization tasks. From the simple flow presented in this study we aim to extend this research to more complex flows with multiple spheres and finite Reynolds numbers.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**;

KEYWORDS

Genetic Programming, Fluid Dynamics, Multi-objective

ACM Reference Format:
Heiner Zille, Sanaz Mostaghim, Fabien Evrard, and Berend van Wachem. 2021. Unit-aware Multi-objective Genetic Programming for the Prediction of the Stokes Flow around a Sphere. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459408

1 STOKES FLOW AROUND A SPHERE

The fluid flow around a fixed sphere is governed by the Navier-Stokes (NS) equations. Its nature depends upon Re, the Reynolds number, a dimensionless quantity characterizing the ratio of inertial effects over viscous effects within the fluid. Owing to the non-linearity of the NS equations, there does not exist a general analytical solution to this fluid-dynamics problem. However, when Re $\to 0$ (also referred to as the Stokes limit, or Stokes flow) the NS equations can be linearized, and an analytical solution to the steady-state flow over a sphere of radius a, subject to the far-field velocity $\boldsymbol{u}_\infty$, can be derived. In a spherical coordinate system whose origin is the center of the sphere, and whose zenith is aligned with the far-field velocity vector $\boldsymbol{u}_\infty$, this axi-symmetric flow ($\partial/\partial\phi = 0$) can be expressed in terms of the stream-function

$$\psi(r,\theta) = u_\infty \sin^2\theta \left(\frac{r^2}{2} + \frac{a^3}{4\,r} - \frac{3\,a\,r}{4}\right), \tag{1}$$

as shown in [e.g. 2], resulting in the velocity components

$$u_r(r,\theta) = \frac{1}{r^2\sin\theta}\frac{\partial\psi}{\partial\theta} = u_\infty\cos\theta\left(1 + \frac{a^3}{2\,r^3} - \frac{3\,a}{2\,r}\right), \tag{2}$$

$$u_\theta(r,\theta) = -\frac{1}{r\sin\theta}\frac{\partial\psi}{\partial r} = -u_\infty\sin\theta\left(1 - \frac{a^3}{4\,r^3} - \frac{3\,a}{4\,r}\right). \tag{3}$$

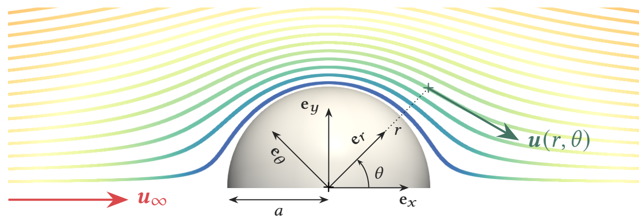

Figure 1: Streamlines of the Stokes flow around a sphere (colored according to the magnitude of velocity).

The streamlines of this flow are shown in Fig. 1. In the remainder of this paper, we will assess the aptitude of unit-aware GP for recovering the governing equations of the flow, from its sampling on a discrete grid. As a next step, we aim to extend this GP approach to consider finite Reynolds number regimes, for which there does not exist analytical velocity field expressions.

2 UNIT-AWARE GENETIC PROGRAMMING

The consideration of physical units inside GP has been introduced in [4], where incompatible units are repaired using a special operator, e.g. adding a length and a time is carried out by artificially transforming the time into a length. Multiple works have since adapted this concept, and used single- as well as multi-objective versions of it [e.g. 1, 3, 6]. Other works on dimensionally correct GP focus on grammar-based GP, where, for instance, the rules of the grammar ensure agreement with physical laws [e.g. 5, 7]. In our approach, in contrast to the method used in [4], whenever incompatible units are used in an operator, the operation is still carried out on the numeric values of the arguments, and the unit of the first input argument is used as the unit for the result of the node (e.g. in case a length and a duration are to be added, the operation is carried out and the result is considered a length). To guide the search towards evolving physically meaningful equations, a penalty value of 1.0 is added every time such a nonphysical operation is carried out. This penalty value is accumulated along the tree together with the results of each function node. Since we know that the result

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459408

	Problem Instance No.					
min	1	2	3	4	5	6
f_1	18 / 18	21 / 21	3 / 3	0 / 0	0 / 2	0 / 6
f_1, f_2	21 / 21	21 / 21	0 / 0	6 / 7	1 / 20	0 / 13
$f_1 + w \cdot f_3$	20 / 20	21 / 21	6 / 6	0 / 0	2 / 2	3 / 9
f_1, f_3	21 / 21	21 / 21	16 / 16	0 / 0	2 / 2	4 / 7
f_1, f_2, f_3	21 / 21	21 / 21	13 / 13	7 / 7	13 / 17	5 / 12

Table 1: Correctly solved runs for each test instance and algorithms. First number indicates numerically and physically correct, second number numerically correct predictions.

should be a model that outputs, for instance, a velocity, any physical unit that is not expressed in `meter/second` is not meaningful for the purpose of the application. Therefore, an additional penalty based on the difference between the obtained units of the tree and the expected units of the output is added. Our approach uses the `deap`-framework (https://github.com/DEAP/deap) and alternates between two different optimization stages: (1) a normal GP variant where crossover and mutation operators are employed and (2) a mutation-only stage where no crossover is used. This two-stage approach avoids extensive growing of the models and allows refinements through a frequent mutation of the population. More details on the parameter settings can be found in the supplement material.

3 EXPERIMENTS AND RESULTS

From the problem described above, we derived 6 problem instances with different numbers of input features and complexities of the correct equations. Details on the problem instances can be found in the supplements. We compare the performance of different variations of GP using classical, dimensionless as well as unit-aware methods. The objective functions used are defined as follows: f_1 = maximum absolute error over the training data. f_2 = Spearman correlation between model output and training data. f_3 = accumulated penalty of the model. The algorithm configurations used are as follows.

(1) single-objective dimensionless GP (min f_1)
(2) multi-objective dimensionless GP (min f_1 and f_2)
(3) single-objective penalized GP (min $f_1 + w \cdot f_3$) with $w = 1/100$
(4) multi-objective unit-aware GP (min f_1 and f_3)
(5) multi-objective unit-aware GP (min f_1 and f_2 and f_3)

As the focus of this work lies on the creation of physically meaningful solutions, a comparison with other numerical methods like neural networks is left for future work. The GP uses the function-set $\mathcal{F} = \{+, -, \times, .^2, .^3, -_1\}$, where $-_1$ is the unary negation, and the terminal-set $\mathcal{T} = \{4, 3, 2, 1, \frac{1}{2}, \frac{1}{4}\}$. In addition to the given input features, we precompute derived features, as preliminary experiments showed that this can be helpful for the optimization. These are $\sin(\theta)$ and $\cos(\theta)$, as well as the square, cubic and multiplicative inverse values for r, a and u_∞. In our proposed multi-objective optimization, we employ the second and third objectives, where applicable, only as a helpful tool for the GP to find solution which predict the goal value correctly. Therefore, as opposed to classical multi-objective experiment analysis, we are not actually interested in the distribution of solutions in the multi-objective space, but rather in the amount of solutions found which are physically meaningful and predict the correct result. Based on this and due to the limited space available in this article, the following analysis concentrates only on the number of successful runs (out of 21 independent runs total) in terms of how often each GP-variant found (1) a numerically correct prediction and (2) a numerically and physically correct prediction of the goal value (i.e. a model with a penalty value of 0 that predicts the correct physical units). Table 1 shows the number of successful runs for both criteria. As for the simple instances 1 and 2, we can observe that all algorithms were mostly able to find the correct equations, and that this relatively simple equation also represents physically correct results. The more complex instance 3 draws a different picture, and we can observe that the algorithms which do not take into account the physical units perform especially poorly, while those which do, particularly the multi-objective versions which optimize f_3, obtain a correct result more often, both physically and numerically. This indicates that using the information of physical units not only leads to more explainable and meaningful solutions, but also enables the GP to solve problems that were not solved in the dimensionless case. Instances 5 and 6 correspond to Eq. (2) and Eq. (3) respectively, and are the most complicated in our experiments. In instances 4, 5 and 6, we can observe that optimizing the second objective is especially helpful to achieve success in these problems. Both algorithms using f_2 show a much higher amount of numerically successful runs. Also, we can observe that only the algorithms which optimize f_3 are able to achieve a high amount of physically correct results.

4 CONCLUSIONS AND FUTURE WORK

In this article, we explore the capabilities of unit-aware GP approaches, both single-and multi-objective, to solve fluid-dynamics-based optimization problems. The results of five different algorithms are analyzed with regard to their numerical solution quality as well as their compliance with the physical units of the measurements. The results indicate that applying multi-objective approaches is beneficial to the success of the GP variants, and that optimizing correlation and violation of physical laws is not only helpful for obtaining meaningful solutions, but also helps to guide the GP toward numerically exact solutions. We aim to extend this research to include more complex problem instances involving, for instance, multiple spheres. In addition, future research aims to examine the influence of the amount and type of input features and of the different objective functions used for the optimization.

REFERENCES

[1] Sunith Bandaru and Kalyanmoy Deb. 2013. A dimensionally-aware genetic programming architecture for automated innovization. In *International conference on evolutionary multi-criterion optimization*. Springer, 513–527.

[2] G. K. Batchelor. 1967. *An Introduction to Fluid Dynamics*. Cambridge University Press, New York.

[3] Marko Durasevic, Domagoj Jakobovic, Marcella Scoczynski Ribeiro Martins, Stjepan Picek, and Markus Wagner. 2020. Fitness Landscape Analysis of Dimensionally-Aware Genetic Programming Featuring Feynman Equations. *arXiv preprint arXiv:2004.12762* (2020).

[4] Maarten Keijzer and Vladan Babovic. 1999. Dimensionally aware genetic programming. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*. 1069–1076.

[5] Robert I Mckay, Nguyen Xuan Hoai, Peter Alexander Whigham, Yin Shan, and Michael O'neill. 2010. Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines* 11, 3-4 (2010), 365–396.

[6] Yi Mei, Su Nguyen, and Mengjie Zhang. 2017. Constrained dimensionally aware genetic programming for evolving interpretable dispatching rules in dynamic job shop scheduling. In *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 435–447.

[7] Alain Ratle and Michèle Sebag. 2001. Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics. *Applied Soft Computing* 1, 1 (2001), 105 – 118.