Data-driven discovery of free-form governing differential equations

Steven Atkinson[*] steven.atkinson1@ge.com
Waad Subber[*] waad.subber@ge.com
Liping Wang[*] wangli@ge.com
Genghis Khan[*] khan@ge.com
Philippe Hawi[†] hawi@usc.edu
Roger Ghanem[†] ghanem@usc.edu

Abstract

We present a method of discovering governing differential equations from data without the need to specify *a priori* the terms to appear in the equation. The input to our method is a dataset (or ensemble of datasets) corresponding to a particular solution (or ensemble of particular solutions) of a differential equation. The output is a human-readable differential equation with parameters calibrated to the individual particular solutions provided. The key to our method is to learn differentiable models of the data that subsequently serve as inputs to a genetic programming algorithm in which graphs specify computation over arbitrary compositions of functions, parameters, and (potentially differential) operators on functions. Differential operators are composed and evaluated using recursive application of automatic differentiation, allowing our algorithm to explore arbitrary compositions of operators without the need for human intervention. We also demonstrate an active learning process to identify and remedy deficiencies in the proposed governing equations.

1 Introduction

The modern scientist enjoys access to ever-growing amount of laboratory data to help uncover new knowledge. However, this abundance of data can be unwieldy to analyze using traditional methods of deduction from which governing laws usually arise. Motivated by the notion of machine learning as a partner in the scientific process, we introduce a method to automate the process of understanding and manipulating data for the purpose of hypothesizing, criticizing, and ultimately discovering novel physical governing laws. Our work is similar in spirit to a growing literature on machine learning and differential programming to solve differential equations (1; 2; 3; 4), and calibration of physical laws where the involved terms are known *a priori* (5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18).

Here, we focus on discovering *free-form* equations in that, unlike previous work, we do not require an *a priori* specification of the terms that may appear in the equation to be discovered; instead we take as input a library of primitive (algebraic and differential) operators on functions from which expressions are composed, calibrated, and evaluated on the fly. The result is a method that allows substantial flexibility in discovering differential equations while still producing a highly structured result (a compositional expression represented as a graph) conducive to further study and analysis.

[*]GE Research, Niskayuna, NY, USA.
[†]University of Southern California, Los Angeles, CA.

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

2 Methodology

Consider the space of pairs of functions $\mathcal{U}_\mathcal{L} = \{(\boldsymbol{u} : \Omega_x \to \Omega_u \subseteq \mathbb{R}^{d_u}, f : \Omega_x \to \mathbb{R})\}$ such that $\mathcal{L}[\boldsymbol{u}] = f \; \forall (\boldsymbol{u}, f) \in \mathcal{U}_\mathcal{L}$, where $\Omega_x \subset \mathbb{R}^{d_{st}}$ is a d_{st}-dimensional spatiotemporal domain of interest. Some (possibly stochastic) observation operator $\mathcal{O} : \mathcal{U} \to \mathcal{D}$ produces discrete samples of these functions as a data set $\boldsymbol{D}_i = \{\boldsymbol{d}_{ij}\}_{j=1}^{d_u+1}$ (e.g. sensor digital readouts). Given N "experiments" $\{\boldsymbol{D}_i\}_{i=1}^{N}$ from possibly-distinct $(\boldsymbol{u}, f)$ pairs, we seek to determine the differential equation $\mathcal{L}$ obeyed within our class of physical systems of interest $\mathcal{U}_\mathcal{L}$ as well as to calibrate any parameters in $\mathcal{L}$ particular to the data observed. We present a methodology, summarized schematically in Fig. 1, to solve this problem, explained presently.

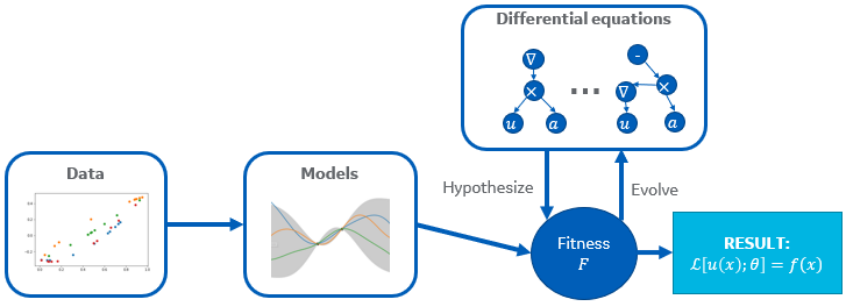


Figure 1: Schematic of our methodology for discovering governing differential equations from data.

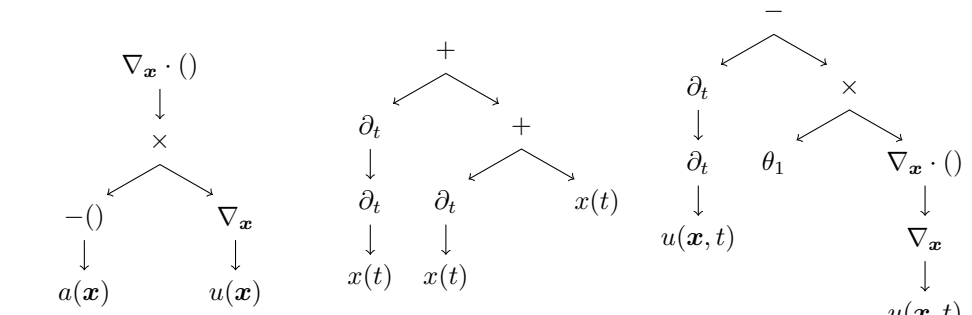
The first step to our approach is to fit explicit, differentiable models $\{M_{ij}\}_{i,j=1}^{N,d_u}$, $M_{ij} \in \mathcal{M}$ to each scalar output data set $\{\boldsymbol{d}_{ij}\}$. We considered fully-connected feedforward neural networks (15) and Gaussian processes[3]. Note that the model architectures (e.g. choice of nonlinearities and kernels) must be selected to reflect the desired differentiability properties matching our inductive biases about the nature of $\mathcal{U}_\mathcal{L}$.

Second, having equipped our data $\{\boldsymbol{d}_{ij}\}$ with differentiable function representations $\{M_{ij}\}$, we utilize a genetic programming algorithm to compose differential equations. Let a differential equation (hereafter referred to as an individual) be represented by a tree graph $G(V, E)$ whose leaves are instances of the fitted models and parent verticies are n-ary operators selected from a user-supplied library of operators. We do not restrict ourselves to algebraic operators $(+, -, \times, \dots)$, but include *differential* operators as well $(\partial/\partial x_i, \nabla(\cdot), \nabla \cdot (\cdot), \dots)$. Each leaf of the graph template is primarily associated with models of a particular output dimension j, but *realizations* of the graph instantiate all of the leaves from some "experiment" i. Differential operators are computed using automatic differentiation using PyTorch (19); Critically and in contrast to prior work, arguments to the operator nodes are functions, not arrays of numbers, allowing us to compose G as a *function* to evaluate at any point in Ω_x to return a residual $r(\boldsymbol{x})$. Figure 2 provides some example expressions represented as graphs that such as those that our method handles.

Finally, parameters $\boldsymbol{\theta}$ to be calibrated enter G as leaves representing constant functions; we calibrate them in a Bayesian manner using black-box variational inference (20). Reflecting our ignorance about the role of the $\{\theta_i\}$'s in G, we use a flat prior for $p(\boldsymbol{\theta})$ and non-factorized multivariate Gaussian variational posterior $q(\boldsymbol{\theta})$. The likelihood is a Gaussian $r(\boldsymbol{x})$ with point estimate variance.

The hypothesis space over graphs is explored via genetic programming (21); we use the evolutionary algorithm implemented by deap (22) to initialize, mutate, and mate graphs. Individuals compete using the evidence lower bound (ELBO) evaluated at the inputs associated with the $\{\boldsymbol{D}_i\}$'s as the fitness function L. By iterating the EA, differential equations are automatically hypothesized and criticized; the outcome of the EA is a list of candidate differential equations $(G_1, G_2, \dots)$ ranked by their fitness so $L(G_1, D) \geq L(G_2, D) \geq \dots$, reflecting the degree to which they (through the fit models $\boldsymbol{M}$) explain the observed data. Minimality may be encouraged through multi-objective optimization with a second fitness function favoring parsimony as in (23; 24) but was not considered in our current work as we did not require it to discover the correct underlying equations in our examples.

[3]We use the posterior mean of the GP model in subsequent manipulations.



(a) The elliptic operator: $\nabla \cdot (-a(\boldsymbol{x})\nabla u(\boldsymbol{x}))$.

(b) Second-order ordinary differential operator: $\ddot{x}(t) + \dot{x}(t) + x(t)$.

(c) The wave equation operator: $\ddot{u}(\boldsymbol{x}, t) - \theta_1 \nabla^2 u(\boldsymbol{x}, t)$

Figure 2: Examples of compositional differential operators represented as graphs combining functions with primitive algebraic and differential operators.

If one has the ability to gather more data, then our approach may be paired with an active learning loop to sequentially obtain additional samples to augment the D_i's using $a(\boldsymbol{x}) = r^2(\boldsymbol{x})$ under G_1 as an acquisition function. This causes one to look more closely where the best explanation of the observed data is most inadequate. One may also define a tolerance δ such that when $a^* = \arg\max_{\boldsymbol{x} \in \Omega_x} a(\boldsymbol{x}) < \delta$, the iterative algorithm is said to have converged and the discovered equation explains the observed data to a user-specified degree.

3 Examples

Code for examples in this section will be made public upon publication. In all of the following examples, we use the posterior means of Gaussian process models with squared exponential kernels (25) to fit functions to the data; Fully-connected neural networks tended not to capture the gradients of the data accurately. We hypothesize this can be attributed to inferior inductive biases conferred by typical architectures, and is a subject of ongoing investigation. The evolutionary algorithm uses a population of 512 individuals, probability of mutation 0.2, and probability of mating 0.8. These hyperparameters were not fine-tuned as we did not find it necessary in order to obtain satisfactory performance.

3.1 Synthetic benchmarks

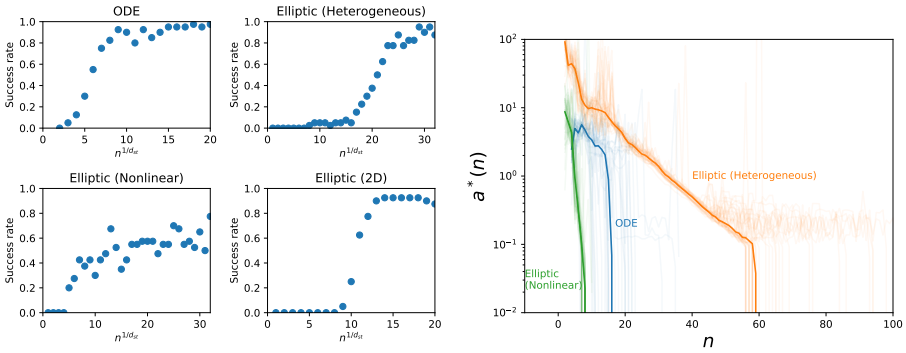
We benchmark the above methodology by applying it to a number of classic differential equations enumerated in Table 1. We select ground truths with unity coefficients to focus here on discovering the structure of the equation; calibration of parameters will be emphasized in the example of Sec. 3.2. Data for the ODEs and one-dimensional elliptic problems were produced using Python's SciPy library and FEniCS (26), respectively. For the two-dimensional elliptic equation, data was obtained from (27). We repeat each experiment 40 times with different random number generator seeds to understand the robustness of our method.

Table 1: Summary of the benchmark differential equations discovered using our methodology.

Differential equation	Differential operator	Dependent variables	$f \stackrel{?}{=} 0$
ODE	$\ddot{u} + \dot{u} + u$	$u(t) : [0, 10] \to \mathbb{R}$	Yes
Heterogeneous elliptic PDE	$\frac{d}{dx}(-a(x)\frac{du}{dx})$	$a : [0, 1] \to \mathbb{R}^+$, $u(x) : [0, 1] \to \mathbb{R}$	No
Nonlinear elliptic PDE	$\frac{d}{dx}(-a(u)\frac{du}{dx})$	$u(x) : [0, 1] \to \mathbb{R}$	No
2D Heterogeneous elliptic PDE	$\nabla \cdot (-a(x)\nabla u)$	$a : [0, 1]^2 \to \mathbb{R}^+$, $u(x) : [0, 1]^2 \to \mathbb{R}$	Yes

Figure 3a shows the success rate for the benchmark systems listed in Table 1 as the number of samples available from the experiment n is varied. We report results in terms of $n^{1/d_{st}}$, where d_{st} is the

number of independent (spatiotemporal) variables in the problem. Figure 3b shows the performance of our method in the adaptive setting in which data are added sequentially following Sec. 2, as quantified by the evolution of a^* with increasing n. We see that a termination tolerance of $\delta = 0.1$ seems sufficient to ensure the discovery of the correct equations, and many runs consistently identify the correct equation far in advance of termination.



(a) Success rates (b) Adaptive sampling

Figure 3: (a) Frequency of success in discovering the underlying differential equations for the benchmark problems as a function of $n^{1/d_{st}}$. (b) Maximum discrepancy as quantified by a^* during adaptive acquisition of data. Different curves show repeats for a given problem. Heavy lines show the median performance over 40 repeats.

3.2 Real-world ultrasound experiment

Finally, we demonstrate our approach on discovering the physics of a laboratory ultrasound experiment. A laser is used to measure the deflection of a cracked Aluminum alloy specimen subjected to an acoustic impulse. The wave travels through the material and exhibits back-scattering at the crack location (Fig. 4).

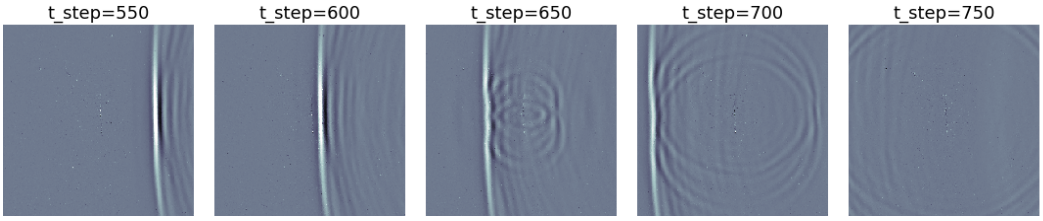


Figure 4: Snapshots of the ultrasound experiment at various time steps.

It is expected based on expert knowledge that the data should be well-described by the wave equation, $\mathcal{L}^w[u] = u_{tt} - \alpha\Delta^2 u_{xx} = 0$, throughout the bulk of the homogeneous material. However, this fails to capture inhomogeneities, dissipative forces, and out-of-plane behavior that might be amended with corrective terms. It is of interest to build a computational model based on the discovered physics from which we may ultimately solve the inverse problem of inferring unseen crack morphologies from a sparse sensor array. We restrict ourselves to searching for a corrector $\mathcal{L}'$ such that $u_{tt} - \mathcal{L}[u;\boldsymbol{\theta}] = 0$, with $\mathcal{L} = \theta_1\nabla^2 u + \mathcal{L}'[u;\boldsymbol{\theta}_{2:}]$. Parameters $\boldsymbol{\theta}$ are calibrated as explained in Sec. 2.

We preprocess the data by cropping to a smaller window in spacetime to exclude data where no significant activity is observed; the resulting data are fitted with a GP. Table 2 enumerates the fittest correctors discovered from the first 32 proposed individuals along with the mean and standard deviation of the parameters' marginal posteriors and fitness (ELBO) associated with the individuals. Figure 5 illustrates a slice of the dataset along with its GP model, the differential terms u_{tt} and $\nabla^2 u$, and the resulting residual associated with the calibrated wave equation using the mode of $q(\boldsymbol{\theta})$.

Table 2: Potential correctors discovered for the ultrasound dataset. The uncorrected equation is provided last as a baseline.

$\mathcal{L}[u]$	$q(\boldsymbol{\theta})$ (mean, std)	ELBO
$\theta_1\nabla^2 u + \theta_2 + \theta_3 u$	$(21.7, 0.14), (120, 2300), (-69, 2900)$	$\mathbf{-6.17445 \times 10^4}$
$\theta_1\nabla^2 u + \theta_2 u + \theta_3 u^2$	$(21.7, 0.14), (-69, 3400), (-3.78, 4.70)$	-6.17450×10^4
$\theta_1\nabla^2 u + \theta_2 + \theta_3 u^2$	$(21.7, 0.14), (110, 2300), (83, 2400)$	-6.17452×10^4
$\theta_1\nabla^2 u + \theta_2 u^2 + \theta_3 u^3$	$(21.7, 0.14), (84, 2200), (-45, 1400)$	-6.17455×10^4
$\theta_1\nabla^2 u + \theta_2 u + \theta_3 u_t$	$(21.7, 0.14), (-69, 3400), (-3.78, 4.70)$	-6.17509×10^4
$\theta_1\nabla^2 u$	$(21.7, 0.14)$	-6.17626×10^4

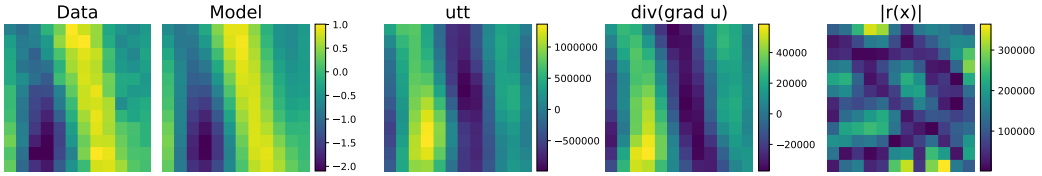


Figure 5: From left to right: A slice in time of the ultrasound data, its GP function representation used in exploring differential equations, the term $u_{tt}(x)$ evaluated on the function, the term $\nabla^2 u$ evaluated on the function, and the residual of the calibrated wave equation using the mode of $q(\boldsymbol{\theta})$.

4 Conclusions

We have described a novel method for discovering governing differential equations with arbitrary structure from raw data and demonstrated its effectiveness on a number of standard differential equations. Our method paves the way for an artificial intelligence "research assistant" as a companion to scientists seeking to understand novel physics. Furthermore, the output of our method, being human-readable differential equations, is compatible with existing workflows available to engineers and scientists such as theoretical analysis and numerical simulation, including so-called physics-informed machine learning methods.

Acknowledgments

The authors thank the United States Air Force Research Laboratory for providing the ultrasound data analyzed in Sec. 3.2 for public release under Distribution A as defined by the United States Department of Defense Instruction 5230.24. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111990032. Approved for public release; distribution is unlimited.

References

[1] Ioannis G Tsoulos and Isaac E Lagaris. Solving differential equations with genetic programming. *Genetic Programming and Evolvable Machines*, 7(1):33–54, 2006.

[2] Dario Izzo, Francesco Biscani, and Alessio Mereta. Differentiable genetic programming. In *European Conference on Genetic Programming*, pages 35–51. Springer, 2017.

[3] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-net: Learning PDEs from data. *arXiv preprint arXiv:1710.09668*, 2017.

[4] Waldir Jesus de Araujo Lobão, Marco Aurélio Cavalcanti Pacheco, Douglas Mota Dias, and Ana Carolina Alves Abreu. Solving stochastic differential equations through genetic programming and automatic differentiation. *Engineering Applications of Artificial Intelligence*, 68:110–120, 2018.

[5] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[6] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[7] Daniel L Ly and Hod Lipson. Learning symbolic representations of hybrid dynamical systems. *Journal of Machine Learning Research*, 13(Dec):3585–3618, 2012.

[8] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[9] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.

[10] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[11] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[12] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[13] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.

[14] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[15] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

[16] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

[17] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019.

[18] Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I. Siettos, and Ioannis G. Kevrekidis. Coarse-scale PDEs from fine-scale observations via machine learning. *arXiv preprint arXiv:1909.05707*, 2019.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[20] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[21] Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. *Genetic programming: an introduction*, volume 1. Morgan Kaufmann San Francisco, 1998.

[22] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[23] Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863, 2013.

[24] Adam Gaier and David Ha. Weight agnostic neural networks. *arXiv preprint arXiv:1906.04358*, 2019.

[25] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[26] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

[27] Steven Atkinson and Nicholas Zabaras. Structured bayesian gaussian process latent variable model: Applications to data-driven dimensionality reduction and high-dimensional inversion. *Journal of Computational Physics*, 383:166–195, 2019.