

Article

Automated Rediscovery of the Maxwell Equations

Mark Stalzer ^{1,*} and Chao Ju ^{2,†} ¹ California Institute of Technology, 1200 E. Colorado Blvd., Pasadena, CA 91125, USA² Minerva Schools at KGI, 1145 Market St., San Francisco, CA 94103, USA

* Correspondence: stalzer@caltech.edu

† The authors contributed equally to this work.

Received: 12 April 2019; Accepted: 20 May 2019; Published: 19 July 2019



Featured Application: A computer can rediscover physical theories—such as the Maxwell Equations—given the appropriate representation language and experimental data. It may only take a few seconds.

Abstract: There is sufficient information in the far-field of a radiating dipole antenna to rediscover the Maxwell Equations and the wave equations of light, including the speed of light c . We created *TheoSea*, a Julia program that does this in a few seconds, and the key insight is that the *compactness* of theories drives the search. The program is a computational embodiment of the scientific method: observation, consideration of candidate theories, and validation.

Keywords: automated discovery; computational science; meta-programming; electrostatics

1. Introduction

This paper flows from a comment in the concluding remarks of a recent review (2016) of work in data-driven scientific discovery [1]. Specifically,

... it may be within current computing and algorithmic technology to infer the Maxwell Equations directly from data given knowledge of vector calculus.

This paper reports on the progress towards this objective.

The overarching goal is to develop methods that can infer compact theories from data. Most data intensive analysis techniques are based on machine learning or statistics. These are quite useful, but do not lead to deep understanding or insight. The scientific method and creative scientists have been very good at observations (experiments) and building human understandable models (theory). In this program, we turn both of these ideas on their heads: can a computer given an appropriate *virtual experiment* (VE) figure out mathematically compact theories [2,3]? The initial application is electrostatics (the Maxwell Equations) [4]. Eventually, it is hoped the methods developed will be applicable to datasets from real measurements in a wide variety of fields in physics, engineering, and economics.

Past work. Attempts to use computers to rediscover physical laws may have originated in 1979 with BACON.3 [5]. The program successfully found the ideal gas law, $PV = nRT$, from small data tables. One of our team (Stalzer) and William Xu of Caltech have also rediscovered the ideal gas law with Van der Waals forces using the approach of this paper [6]. In 2009, researchers rediscovered the kinematic equation for the double pendulum essentially using optimization methods to fit constants to candidate equations [7].

What differentiates this work is twofold: the concept of search driven by compactness and completeness (“compactness” here means that we assign weights or complexity scores to each theory

being searched; “completeness” means that we can always find theories, if any, whose weights are within the specified limit) and targeting a much more difficult theory: electrodynamics. Indeed, electrodynamics was the first unification (the electric and magnetic fields), and Einstein’s special relativity is baked right into the equations once the brilliant observation is made that c is the same in all inertial reference frames. *TheoSea* also finds the wave equations of light as a consequence of the rediscovered free space Maxwell Equations.

Julia. *TheoSea* is written in Julia [8], a relatively recent language (roughly 2012) that is both easy to use and has high performance. Julia can be programmed at a high expressive level, and yet, given enough type information, it automatically generates efficient machine code. *TheoSea* is a Julia *meta-program* that writes candidate theories in terms of Julia sets that are then validated against data. The set elements are compiled Julia expressions corresponding to terms in the candidate theories. Simultaneously having both high levels of expression and good low-level performance is hard to do in other languages like C and Fortran, and that was a key motivation in Julia’s creation.

Plan. The next section presents some preliminaries that are used by *TheoSea* in general. Central to the approach is the section on rapid enumeration (Section 2.1). Section 3 then presents the virtual experiment, rediscovery, and run-times. The paper ends with some concluding remarks (Section 4).

2. Preliminaries

TheoSea rests on two key algorithms: rapid enumeration of theories and connecting the theories to virtual experiments (Section 2.2).

2.1. Rapid Enumeration of Candidate Theories Over a Language \mathcal{L}

Given an alphabet \mathcal{A} of symbols, such as operators and fields, a language \mathcal{L} is *recursively enumerable* if there exists a Turing machine that will enumerate all valid strings in the language [9].

By the *infinite monkey theorem* [10], the solution can be found—if the constants are limited to rationals—just by enumeration and validation. The goal of this section is to show a way of doing this enumeration in a tractable way that also finds the *most compact theory*.

Abstract enumeration. Think of an alphabet $\mathcal{A} = [A, B, C, \dots]$ where any letter can appear once in a sentence and the length of the alphabet is n . This is a simple combinatorial enumeration problem and the solution to the number of sets of size m (later m will be relabeled q) taken from \mathcal{A} is $C(n, m)$. However, what if the symbols—letters—in the alphabet have different weights? What if the alphabet is more like $\mathcal{A} = [A = 1, B = 1, C = 4, D = 4, E = 4, F = 4, G = 7, H = 7, I = 7, J = 7, K = 4, L = 19]$. This can dramatically decrease the enumeration size and the underlying motivation is shown below in the *decoder ring for electrodynamics*.

The *TheoSea* enumeration algorithm builds sets of increasing complexity q , where q is the sum of the alphabet letter weights in a given candidate theory. It can be thought of as a form of depth-first iterative-deepening (DFID) [11] first formalized by Korf in 1985. Optimality flows from a theorem by Korf:

Theorem 1. *Depth-first iterative-deepening is asymptotically optimal among brute-force tree searches in terms of time, space, and length of solution.*

By length of solution, Korf means the depth of the search where a solution is found. For *TheoSea* compactness is the sum of the symbol weights along a potential solution branch in the search.

It is perhaps easiest to think of the algorithm inductively. There is a data structure `theos` that holds all theorems (sets) of length q and it is built up from $q = 1$. The base cases are the singleton theories of a given complexity, so for the alphabet \mathcal{A} , we have `theos[1] = [A, B]` and `theos[4] = [C, ...]` and so on. Therefore, the base cases, such as $q = 1$ are all set; and then for $q > 1$, we use a $q : l, m$ “squeeze”. At step q , consider all theories that can possibly be of length q , *marching l* upward from 1

and m downward from $q - 1$. The correctness is immediate by Korf 4.2 and the fact that $q = l + m$: too short theories are discarded ($< q$), and set elements are unique. More details can be found in [2].

Performance. To test the performance, we run an experiment comparing the enumeration of \mathcal{A} (Fast) with the same set of letters but with unity weights (Slow), with a cutoff at $q = 14$. The total times are Fast = 0.006 s for the weighted \mathcal{A} above, and Slow = 20.1 s (the machine was a MacBook Pro (Retina, 13-inch, Late 2013) running a 2.8 GHz Intel Core i7 single-threaded using Julia 0.5). A graph is shown in Figure 1: *compactness matters*.

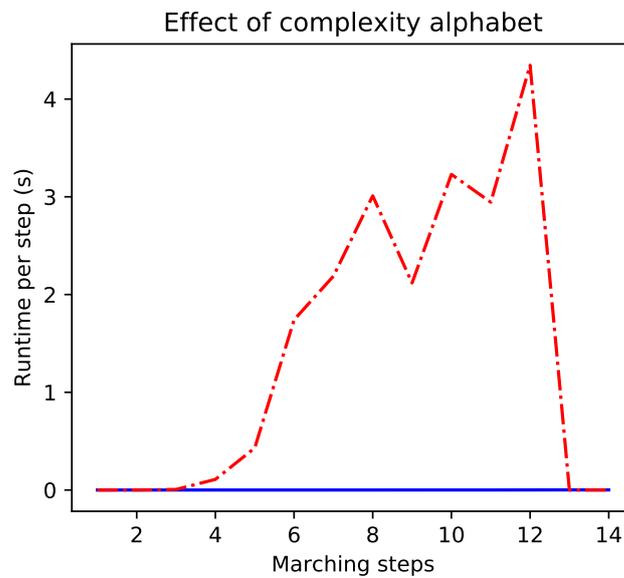


Figure 1. Time to discovery (s): Fast versus Slow brute force. The Slow search cuts off at 12 due to maximum complexity.

Decoder ring for electrodynamics. The underlying motivation was described above and here is the decoder ring; think of the electric and magnetic fields with the \mathcal{A} assignments of $A = \mathbf{E}$ electric field, $B = \mathbf{B}$ magnetic field . . .

The Maxwell Equations are expressed in linear combinations of these terms as is described in Section 3.1. The complexity metric is just 1+ the number of space–time derivatives taken (Table 1).

Table 1. Complexity of each operator term working on a field $\mathcal{F} \in \{\mathbf{E}, \mathbf{B}\}$.

Operator	Term Cost	Alphabet
\mathcal{F}	1	A, B
$\nabla \cdot \mathcal{F}$	4	C, D
$\frac{\partial}{\partial t} \mathcal{F}$	4	E, F
$\nabla \times \mathcal{F}$	7	G, H
$\nabla^2 \mathcal{F}$	19	I, J
$\frac{\partial^2}{\partial t^2} \mathcal{F}$	19	K, L

2.2. Theory Validation: Fitting Constants

The glue that connects the enumeration (Section 2.1) to the virtual experiment (Section 3.2) is finding constants in the candidate theories that fit the data. The problem is to find an equation of the form (recall that the size of $\mathcal{A} = n$)

$$c_1 \mathcal{A}_1 + \dots + c_n \mathcal{A}_n = 0, \tag{1}$$

where many of the c_i are implicitly zero by the enumeration. If such c_i do not exist, which is almost always the case, the theory is invalid. However, if they do exist, then the theory is valid with high probability. This is a linear algebra problem as described below.

Finding the constants is equivalent to finding the null space of a linear system (for example, the data matrix extracted from Table 2). If the dimension of the null space is 0, then the theory is not valid because the only solution is a trivial zero vector. If the dimension of the null space is nonzero, it can only be 1, which corresponds to a unique solution. The reason is that in our enumerative method, we remove all valid sub-theories from the candidate theory before determining its constants. Had the dimension of the null space of the resulting system been larger than 1, it would have implied that some sub-theory had not been removed, contradicting the assumption. Next, to find the rank of the null space and the null space itself, we cannot simply use Julia’s built-in `rank()` or `nullspace()` functions because the dynamic ranges are large ($>10^{30}$) (this is one of those cases when symbolics and numerics do not play well together).

The solution is to use singular value decomposition, in which the number of zero singular values (SVs) is equivalent to the dimension of the null space. The insight is that if we scale \mathbf{B} by a factor of c , it will be on the same scale as \mathbf{E} , and after normalizing each column of the matrix so that each is on the same scale with another, the resulting singular values (if nonzero) should also be on the same scale. After scaling and normalizing, we use Julia’s built-in `svdvals()` function to obtain a list of SVs ranked from the largest to the smallest.

As discussed previously, the dimension of the null space can either be 1 or 0, and we only need to compare the smallest SV with the largest one to see whether the former is orders of magnitude smaller than the latter. If so, we can regard that as a zero, and proceed to retrieve the null space vector from the last column of V^T (as in $A = U\Sigma V^T$) by calling Julia’s `svd()` function. The elements of the null space vector are the constants we look for. If not, it implies that the dimension of the null space is zero, and we conclude that the theory is invalid.

3. Results: Electrodynamics and Light

In what follows, we introduce the equations, describe the virtual experiment, and show the results of the rediscovery.

3.1. The Maxwell Equations

The Maxwell Equations in free space with the transformation $\mathbf{B}' = c \mathbf{B}$ are [4]:

$$\nabla \cdot \mathbf{E} = 0 \tag{2}$$

$$\nabla \cdot \mathbf{B}' = 0 \tag{3}$$

$$\nabla \times \mathbf{E} + \frac{1}{c} \frac{\partial \mathbf{B}'}{\partial t} = 0 \tag{4}$$

$$c \nabla \times \mathbf{B}' - \frac{\partial \mathbf{E}}{\partial t} = 0, \tag{5}$$

where $c = 2.99792458 \times 10^8$ m/s (MKS units). The spatial-temporal coupling of \mathbf{E} and \mathbf{B} is how we get electromagnetic waves. The utility of the \mathbf{B} transformation for numerical stability is discussed in the previous section and that is why the equations appear to be in a slightly strange form in terms of constants.

Connecting back to language, the divergence Equations (2) and (3) are of complexity 4, and the space–time couplings (4) and (5) are of complexity 11. *TheoSea* does not know it, but the code is doing vector calculus *by* validating its candidate theories (strings over \mathcal{L}) against data.

3.2. Observations and the Virtual Experiment

The validation data is from the far-field of a radiating antenna for \mathbf{E} , \mathbf{B} as shown in the geometry Figure 2 and data Table 2.

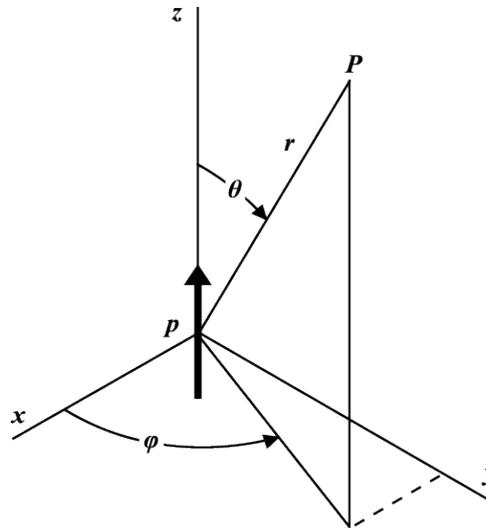


Figure 2. Geometry of the far-field of a dipole with moment \mathbf{p} at the origin oscillating at angular frequency ω at a far point P .

Table 2. Experiments with geometry parameters and the corresponding field observations at point P and time t ; adjusted for scale.

r [10^{17}]	ϕ	θ	$\mathbf{E}_P(0)$ [10^{-16}]	$\mathbf{B}_P(0)$ [10^{-16}]
1	0.905	0.997	(2.713, 3.455, -6.793)	(-6.363, 4.996, 0)
1	2.767	2.908	(2.018, -0.794, -0.516)	(-0.816, -2.074, 0)
1	4.631	0.291	(-0.214, -2.639, -0.793)	(2.754, -0.224, 0)
1	5.597	3.051	(-0.666, 0.546, -0.078)	(0.548, 0.669, 0)
1	0.468	2.369	(-4.299, -2.170, -4.690)	(-3.029, 6.001, 0)

The fields at a far point P are [12]

$$\mathbf{E} = -\frac{\mu_0 p_0 \omega^2}{4\pi} \left(\frac{\sin \theta}{r}\right) \cos[\omega(t - r/c)] \hat{\theta} \tag{6}$$

$$\mathbf{B} = -\frac{\mu_0 p_0 \omega^2}{4\pi c} \left(\frac{\sin \theta}{r}\right) \cos[\omega(t - r/c)] \hat{\phi}, \tag{7}$$

where $\mu_0 = 4\pi \times 10^{-7}$ is the permeability of free space, p_0 is the strength of the dipole, and ω is the frequency of the dipole oscillation.

Five virtual experiments were done with various parameters r, ϕ, θ with a fixed ω . The observables are $\mathbf{E}(\mathbf{x}, t)$ and $\mathbf{B}(\mathbf{x}, t)$, where \mathbf{x} is in the region of the point P . The nice thing about this VE is that various space–time derivatives can be computed analytically. These experiments are shown in Table 2, with the fields given at a steady state $t = 0$.

3.3. Rediscovery

TheoSea has rediscovered the Maxwell Equations as shown in the screenshot Figure 3 and it took about 5 s using the methods described above (the Julia code is attached as supplemental information). For a bit more intuition, the magnitude of the Poynting vector is shown in Figure 4: the code is essentially quickly enumerating space–time theories on the surface of the saucer-shaped object and finding the most compact ones.

```
The singleton theorem is of the form:  $\nabla \cdot \mathbf{E} = 0$ 
The singleton theorem is of the form:  $\nabla \cdot \mathbf{B} = 0$ 
elapsed time: 9.9791e-5 seconds

The theorem is of the form:  $1.0 \nabla \times \mathbf{E} + 3.33564104655293e-9 \partial \mathbf{B} / \partial t = 0$ 
The theorem is of the form:  $1.0 \partial \mathbf{E} / \partial t - 2.997924494987667e8 \nabla \times \mathbf{B} = 0$ 

The theorem is of the form:  $1.0 \nabla^2 \mathbf{E} - 1.1126501163632473e-17 \partial^2 \mathbf{E} / \partial t^2 = 0$ 
The theorem is of the form:  $1.0 \partial^2 \mathbf{B} / \partial t^2 - 8.987551300210622e16 \nabla^2 \mathbf{B} = 0$ 
elapsed time: 4.979688875 seconds
```

Figure 3. Screenshot of *TheoSea*'s output showing the Maxwell Equations and the wave equations of light including the speed of light $c \approx 2.99 \times 10^8$.

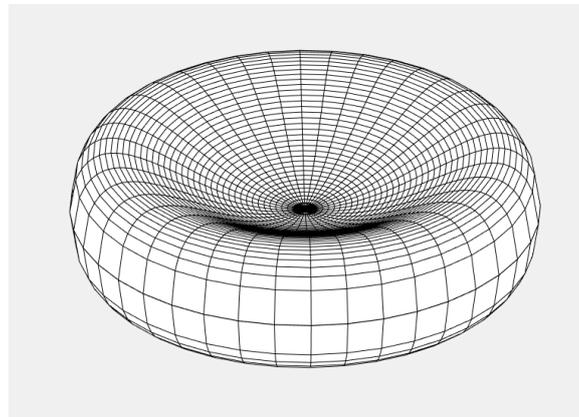


Figure 4. Average intensity of the Poynting vector, the vector is proportional to $(\mathbf{E} \times \mathbf{B})(\omega)$.

In addition, it rediscovered

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} \mathbf{E} - \nabla^2 \mathbf{E} = 0 \tag{8}$$

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} \mathbf{B} - \nabla^2 \mathbf{B} = 0 \tag{9}$$

that is a plane electromagnetic wave traveling in free space: *Light* (the derivation of these wave equations from the Maxwell Equations takes humans some non-trivial vector calculus, and yet the machine did it by “enlightened” search. This machine was a MacBook Pro (Retina, 15-inch) running a 2.5 Ghz Intel Core i7 single-threaded).

4. Concluding Remarks and Future Work

There are many avenues for future development as briefly listed below.

- Expand the enumeration language to allow more expressive theories. Right now *TheoSea* is limited to theories of the form $c_1 A_1 + c_2 A_2 + \dots$ where the A 's are operators over fields like \mathbf{E} , \mathbf{B} . The Xu ideal gas law code works with scalar fields and exponents. Somehow these should be merged;
- Bigger data and parallelism. The dataset used was very small but semantically very rich. Other datasets, such as for macro-economics, will be far larger. Here Julia's on-the-fly compilation (of candidate theories) and support for parallel processing will be very helpful, and this is one of the reasons the language was chosen (the author encourages the Julia developers to continue work on threads as the model is natural for multicore processors. For example, the main thread could enumerate candidate theories and then send them to several worker threads for validation. At any instant, several theories would be under consideration);

- Develop methods to work with more noisy data. The current approach works with numerical noise but not measurement noise (Equation (1));
- The fully general Maxwell Equations can be rediscovered just by adding a current \mathbf{J} and source region ρ . The changes to the virtual experiment and language \mathcal{L} are straightforward;
- Field discovery. The fields \mathbf{E}, \mathbf{B} are treated as observables. It would be nice if *TheoSea* could discover the fields from the forces, e.g., $\mathbf{F} = q\mathbf{E}$. One step is to use a relativistic moving charge q with velocity \mathbf{u} , where the magnetic field can be written in terms of the electric field [13]: $\mathbf{B} = (1/c^2) \mathbf{u} \times \mathbf{E}$. Then, the field discovery problem is reduced to finding the electric field and then the magnetic field will fall out from the search.

However, perhaps the most exciting extension is to apply *TheoSea* to other domains; we already have results in thermodynamics (entropy) [14] and the Navier–Stokes equations (laminar fluid flow). It is also beginning to look like it should work for the Schrödinger Equation and quantum mechanics [15]. There may also be applications in the social sciences, such as macro-economics. Work is progressing in these areas, and focusing on the applicable representation language and executable semantics are the keys for extensions to new domains.

The ultimate goal, however, is to generalize *TheoSea*: if a language and validator are supplied, is there a theory? Hamming stated [16] that

[Einstein] knew in advance what the theory should look like, and he explored the theories with mathematical tools, not actual experiments.

The authors would like to suggest an alternative: *if you know the representation language of a theory and perform a validated search, you may find the truth. It may be quicker than you thought.*

Supplementary Materials: The following are available online at <http://www.mdpi.com/2076-3417/9/14/2899/s1>: the Julia code Maxwell.jl.

Author Contributions: All authors contributed equally to this work.

Funding: This research was funded by the Gordon and Betty Moore Foundation through Grant GBMF4915 to the Caltech Center for Data-Driven Discovery.

Acknowledgments: Discussions with William Xu of Caltech Math/Computer Science were very helpful. The authors are grateful to S.G. Djorgovski of Caltech Astronomy and V. Chandler of KGI Natural Sciences for their support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Stalzer, M.; Mentzel, C. *A Preliminary Review of Influential Works in Data-Driven Discovery*; SpringerPlus: Berlin, Germany, 2016.
2. Stalzer, M. On the enumeration of sentences by compactness. *arXiv* **2017**, arXiv:1706.06975.
3. Stalzer, M.; Ju, C. *TheoSea*: Marching theory to light. *arXiv* **2017**, arXiv:1708.04927.
4. Maxwell, J.C. A dynamical theory of the electromagnetic field. *Philo. Trans. R. Soc. Lond.* **1865**, *155*, 459–512.
5. Langley, P. Rediscovering physics with BACON.3. *Proc. Int. Jt. Conf. Artif. Intell.* **1979**, *152*, 505–507.
6. Xu, W.; Stalzer, M. Deriving compact laws based on algebraic formulation of data set. *arXiv* **2017**, arXiv:1706.05123.
7. Schmidt, M.; Lipson, H. Distilling free-form natural laws from experimental data. *Science* **2009**, *324*, 81–85. [[CrossRef](#)] [[PubMed](#)]
8. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.B. Julia: A fresh approach to numerical computing. *SIAM Rev.* **2017**, *59*, 65–98. [[CrossRef](#)]
9. Denning, P.J.; Dennis, J.B.; Qualitz, J.E. *Machines, Languages, and Computation*; Prentice-Hall: Upper Saddle River, NJ, USA, 1978.
10. Borel, E. Mécanique statistique et irréversibilité. *J. Phys.* **1913**, *53*, 189–196.
11. Korf, R.E. Depth-First Iterative-Deepening: An optimal admissible tree search. *Artif. Intell.* **1985**, *27*, 97–109. [[CrossRef](#)]

12. Griffiths, D.J. *Introduction to Electrodynamics*, 4th ed.; Pearson Education: London, UK, 2015.
13. Fitzpatrick, R. *Maxwell's Equations and the Principles of Electromagnetism*; Firewall Media: New York, NY, USA, 2008.
14. Ju, C.; Stalzer, M. Finding exact forms on thermodynamic manifolds. *arXiv* **2018**, arXiv:1808.03863v2.
15. Dirac, P.A.M. *The Principles of Quantum Mechanics*; Oxford University Press: Oxford, UK, 1967.
16. Hamming, R.W. The unreasonable effectiveness of mathematics. *Am. Math. Mon.* **1980**, *87*, 81–90. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).