A description of software for searchable math and physics equations, identities, and definitions.

Nathaniel Beaver

October 9, 2014

1 License

2 Motivation

Equations are ubiquitous in mathematics and science. However, the way in which we store and reference equations suffers from severe flaws, such as the inability to transparently manipulate equations, the difficulty of connecting disparate systems by shared mathematical structure, and lack of insight into invariance. This hinders scientific inquiry in a variety of ways.

3 Use case

Suppose I want to look up boundary conditions for an electric field to include in a paper I am writing.

3.1 Print resources.

Most physicists will reach for their trusty E&M books in this case.

In the 2nd edition of David J. Griffiths' "Introduction to Electrodynamis", for example, a search of the index for "Boundary conditions" gives these equations (in SI units) on page 313:

$$D_{1_\perp} - D_{2_\perp} = \sigma_f$$
$$E_{1_\parallel} - E_{1_\parallel} = 0$$

The 3rd edition of John David Jackson's "Classical Electrodynamics" also uses SI units, and a quick check of the index gives:

$$(\mathbf{D}_1 - \mathbf{D}_2) \cdot \mathbf{n} = \sigma$$
$$n \times (\mathbf{E}_2 - \mathbf{E}_1) = 0$$

The drawback is that these equations is that they are "read-only." They must be manually typed up for use in a LaTeX document or word processing software, a laborious process even for relatively short equations. They are even less suited for use in computational software.

Ebooks and PDF files do not solve this problem, as they do not use LaTeX or computable formats internally.

If the original document is written in LaTeX however, it is possible to attach the entire LaTeX source code to a PDF, although this is somewhat less than convenient and does not automatically unwrap macros into standard LaTeX commands. This means that the authors must either avoid the use of macros or the user must copy all the necessary macros to use the equations.

3.2   Electronic resources.

Let's see if Wolfram Alpha's database has this. After all, it does a good job with other equations.

```
https://www.wolframalpha.com/input/?i=electric+field+boundary+conditions
```

WolframAlpha computational... knowledge engine

electric field boundary conditions   ☆ ▤

≡ Examples   ⤫ Random

Using closest Wolfram|Alpha interpretation: **field boundary conditions**   ❓

More interpretations: electric field

Input interpretation:

| conditions | Field | | conditions | Boundary |

Results:

| Field | **cloudy** (1 hour 13 minutes ago) |
| Boundary | **cloudy** (1 hour 13 minutes ago) |

📖 Sources   ⊕ Download page   POWERED BY THE **WOLFRAM LANGUAGE**

Standard computation time exceeded...   Try again with additional computation time »

Give us your feedback: [   ]   Send

Hm, that was a little disappointing.

For quick and dirty information finding, Google works well. Let's see what the top results are. `https://www.google.com/search?q=electric+field+boundary+conditions`

The first result, as of October 9, 2014, is `http://farside.ph.utexas.edu/teaching/em/lectures/node59.html` . This website explains their origin from Gauss's Law and gives the equations as embedded png images:

$$E_{\perp A} - E_{\perp B} = \frac{\sigma}{\epsilon_0}$$
$$E_{\parallel A} - E_{\parallel B} = 0 :$$

The second result is here: `http://www.antenna-theory.com/tutorial/electromagnetics/electric-field-boundary-conditions.php`

This page distinguishes between a charged/uncharged surface and uses different notation. They also use images, gif in this case (although I converted them to png for inclusion in this document):

$$D_{1N} = D_{2N} \qquad D_{1N} - D_{2N} = \rho_S \qquad D_{1N} - D_{2N} = \rho_S$$

$$\varepsilon_1 E_{1N} = \varepsilon_2 E_{2N} \qquad \varepsilon_1 E_{1N} - \varepsilon_2 E_{2N} = \rho_S \qquad \varepsilon_1 E_{1N} - \varepsilon_2 E_{2N} = \rho_S$$

Note that already there is considerable variation in notation. Furthermore, neither site includes any equations which can be parsed or modified without performing optical character recognition on the images; they are heavily read-only.

What about Wikipedia?

`https://en.wikipedia.org/wiki/Interface_conditions_for_electromagnetic_fields`

This is the relevant equation.

$$\mathbf{n}_{12} \times (\mathbf{E}_2 - \mathbf{E}_1) = \mathbf{0}$$
$$(\mathbf{D}_2 - \mathbf{D}_1) \cdot \mathbf{n}_{12} = \rho_s$$

Fortunately, there is embedded LaTeX markup available under "Edit Source".

```
:<math>\mathbf{n}_{12} \times (\mathbf{E}_2 - \mathbf{E}_1)   = \mathbf{0} </math>
```

```
:<math>(\mathbf{D}_2 - \mathbf{D}_1) \cdot \mathbf{n}_{12} = \rho_{s} </math>
```

This will only be sufficient if:

1. We want LaTeX markup.

2. the Wikipedia editor uses the LaTeX markup correctly.

3. the Wikipedia markup uses the same symbols and units system that we want to use.

4. We only want to typeset the equation, not use it for actual computation.

4 The problem.

Both the print and electronic resources have a number of deficiencies. In particular,

- Even electronic versions of equations are not generally suitable for using with computational tools.

- There is no reliable standard for representation of variables, even for the simplest and most common equations. Sometimes a surface charge is σ, sometimes it is σ_f, and sometimes it is ρ_s.

- There are a variety of different conventions in use that differ between physics and mathematics. For example, while the right-hand rule is fairly universal, the quantities ϕ and θ are swapped in spherical coordinates. Similar problems exist for Euler angles.

- There are many units systems in use in the sciences, several of which are popular enough to include in, say, the software of a given subfield, and even if everyone were to standardize tomorrow there are centuries of scholarly literature that is expressed in different units. One of the most common examples is the differing forms of equations for SI (MKSA) and Gaussian (cgs) systems. There are valid arguments for using either system, or using the so-called "natural units". Converting between these systems, however, is tedious, and people should not have to spend much time on it.

- In much the same way as there are many unit systems, there are many equally valid choices of gauge in electrodynamics and a proliferation of tensor notations. Dirac notation is another example of elegant notation that is frequently "translated" into more familiar notation for some manipulations.

- Most equations are embedded in a particular coordinate space and vector basis, such as the old standbys of Cartesian, cylindrical, or spherical coordinates. Often, a problem is much easier to attack in a favorable coordinate system. If transformation of coordinate systems was effortless, exploring alternatives solutions would become far less time-consuming and laborious. Wouldn't it be fun to try solving something in a non-holonomic basis?

- Many derivations in physics rely on substitutions that make the equations unitless, yet another related form of the equation that is difficult to find without looking up the original derivation. These kinds of change of variables are very common, but seldom tabulated systematically.

- Curves like conic sections have representations in Cartesian coordinates, polar coordinates, parametric coordinates, and a variety of other exotic options. These can be invaluable for simplifying analysis, but looking them up is time-consuming and re-deriving them is time-consuming and error-prone.

- No print or electronic resource is free from errors, and not everyone checks the errata page.

Together, these problems can make what should be simple tasks into a chore. There is frequently unnecessary friction, for example, in using results from two different papers or collaborating with a colleague who uses a different unit system. And how many scholars have lost hours of work by writing a paper with one choice of symbols and units and having to change them all for a journal submission?

5   What the solution should look like.

Instead of trying to solve these problems by convincing everyone to use the same units, gauges, and symbols, it is much more sane to leverage the ability of machines to do busywork.

An electronic database of equations with the relevant information could easily translate between any system of units, and could produce output suitable for any software, regardless of whether it is intended for use in a Fortran program, a LaTeX document, or a web browser.

If a transformation between these forms is non-trivial, the best thing is simply to have a human provide the input and store them both, and make them easy to find later. Rather than attempting to find the "one true form" of the equation, it should simply associate various useful formats.

Furthermore, it should take free-form input queries that would outpace any web search or paper index. This could include items like common names ("Maxwell equations"), units of variables (all equations with the left side in units of force), and even algebraic forms (all equations of the form `a x^2 + b x + c`). Given sufficient heuristics and metadata (much of which will, admittedly, have to be entered manually), this is a solvable problem.

Finally, it should be personal and collaborative. It should be easy to import the standard equations, but most people have specific, personalized needs and collaborate with a relatively small group of people, so this software should reflect that.

6 Features

A satisfactory electronic equation reference document would:

- Store equation database locally (network connectivity not required).

- Provide a means to write explanatory text for symbols or sub-expressions in the relevant equations, which most texts do anyway as a matter of course. For example, $\vec{L} = I\vec{\omega}$, where $\vec{L}$ is angular momentum, I is the moment of inertia, and $\vec{\omega}$ is the angular velocity.

- Store representations in at least one format of a variety of options (Unicode UTF-8 plaintext, LaTeX, MathML, OpenMath, etc.) as well as a way to add more specific forms. For example, one might be a LaTeX expression with the vector quantities having arrows like this: $\vec{r}$; and another with vector quantities bolded like this: **r**.

- Provide the ability to optionally store and access other formats (e.g. Mathematica, MS Word, Matlab code, Fortran code, C code etc.)

- Call existing software to convert between representations so that not everything has to be converted manually. This falls into two categories:

  1. Conversions that are essentially just formatting conversions. For example, while LaTeX and MathML have some very apparent differences, they both require only enough information to typeset an equation, not actually evaluate it. The automatically generated markup could subsequently be tweaked by hand.

  2. Conversions from a typesetting or markup language like LaTeX or MathML to an expression that can actually be evaluated; one that you can plug in numbers and get an answer. Some equations are simple enough that a conversion to, say, C code is trivial – something just using trig, exponents, and arithmetic functions, for example. Others will be sufficiently abstract as to require manual conversion or avoiding a computable format at all. (Alternately, a link to some remotely hosted code may be in order; see below.) Incidentally, Stephen Wolfram (of Mathematica fame) had this to say about this kind of conversion.

Unlike with ordinary human natural language, it is actually possible to take a very close approximation to familiar mathematical notation, and have a computer systematically understand it. That's one of the big things that we did about five years ago in the third version of Mathematica. And at least a little of what we learned from doing that actually made its way into the specification of MathML.

- Perform queries based on popular names, form of equations, related equations, commonly used symbols.

- Perform queries based on similarity to a given input.

- Provide the option to perform simple substitutions (so a query for $a + b^2$ could return an equation stored as $x + y^2$, and one could specify substitution rules so that it outputs $m + n^2$)

- Provide the option to search for algebraically equivalent forms (e.g. $a(b + c)$ could return $ab + ac$). Note that this would only work for equations that had unambiguous computable forms provided, so that the software could call, e.g. Sage, Maxima, or Mathematica to determine whether or not the forms are algebraically equivalent. MathML or LaTeX compilers will happily typeset algebraically ambiguous or uncomputable gibberish.

- Provide the option to search for algebraically equivalent forms with simple substitutions (e.g. $a(b + c)$ could return $xy + xz$)

- Provide the option to choose units system, so that a single equation would have both cgs and SI forms available, for example.

- Allow extensions to specify new unit systems or gauges.

- Link to internal and external references (refer to another equation in the same database, jump to a specified page of a local or remote PDF or ebook, standard html-style links to urls of relevant websites or source code implementations, digital object identifiers, bibtex references, etc.)

7 Some objections.

7.1 There's more to physics and math than equations, you know.

Yes, and people did complex math for centuries without nice modern algebraic notation, but equations do provide a very lovely and compact way to represent relationships between variables.

Once we've got an easy and reliable way to find the equations we want, we can spend more time reasoning about whether the equation is applicable, what approximations to make, what the physical interpretation is, and how that interpretation squares with experiment and physical intuition.

7.2 Equations without context are dangerous and people will use them when they are not really applicable or assume what they are trying to prove, i.e. use circular reasoning.

A valid concern, which is why it's important to write explanatory text about each symbol and link to more complete discussions.

In any case, people already use equations when they aren't applicable, and this is generally because they don't want to put in the effort to look up the context of the equation. Properly used, this software could help mitigate this problem.

7.3 Why "equations" and not "expressions" or "formulas"?

This is just a nomenclature thing.

I assume that this software could work for mathematical expressions in general, identities, approximations, chemical formulas, etc. Most of the time, though, we need to know about relationships between variables, so "equations" are what most people think of and use regularly.

7.4 Making it easier to change conventions will encourage fragmentation.

Possibly, but it's pretty badly fragmented already. If there is a reliable system to automatically change systems of units and the like, fragmentation will not matter.

Furthermore, such a system would encourage using a standard form when writing papers and textbooks, since everyone could convert to their favorite set of conventions without difficulty.

7.5 Is it really so much work to use a web search or a book index?

Yes.

Try doing a Google search for $L = r \times p$. Symbolab works somewhat better for this, as does searching for likely LaTeX markup.

This is just a simple example of hard it is to find even a basic equation with relatively few ways to express it.

As for books, if you can get everything you need from one book, great. The ones I need are generally scattered across several books and journal articles, none of which have the same notation and only some of which have good indices.

7.6 Surely there are people already working on this, or something similar?

There are some interesting websites out there, but they're more about searching existing websites and scholarly literature and don't accomplish more than a few of the features mentioned above.

Wolfram Alpha is probably the closest right now, but its main goal is much broader than storing equations. The lack of a local, user-controlled database is probably the biggest problem, as is the tendency to only show the "one true form" of an equation.

MathML and OpenMath are projects that employ similar ideas, although MathML is focused mainly on web browsers and OpenMath is still unfinished. (The OpenMath website lists only 58 members, many of whom are professors that work on it in their spare time.)

More importantly, OpenMath is working towards a standard for representing the mathematical objects, not a working piece of software performing the functions mentioned above.

The database of equations could certainly use the OpenMath standard as another representation — a reliable, standardized representation — but it would not require it to work.

MathJax is doing great things for putting math on the web, but not so much for storage and retrieval.

There are also efforts to make derivations automatic. This is a laudable goal, but most people are more interested in the actual results and where they are applicable than rigorously defining the actual mechanism to derive the results.

7.7 Would people actually use this?

I would, and I have reason to believe other people would, too.

7.8 This is way too hard.

It's really not; see below for the actually hard problems. The individual components have existed for decades, they just haven't been tied together yet. I regularly use a desktop search tool to index my various documents and search them, it just isn't geared to equations and so doesn't have the specific features I would like.

7.9 Talk is cheap. Show me the code/data.

There is an example of a minimal prototype/proof of concept on GitHub here:
    https://github.com/nbeaver/equation-manager-prototype

7.10 There are too many equations. Searching for them would be hopeless.

While there are a large (though finite) number of equations of a given finite size, only a tiny subset of them are useful.

For comparison, there are a lot of Unicode characters, but writing a program to search for the one you want is standard. The code for querying the database is pretty simple, too.

8 More ambitious possibilities.

8.1 Semantics of representation and elimination of ambiguity.

Ideally, one would add semantics to equations. This could help avert the namespace problem which is rampant in physics.

For example, introductory kinematics generally uses m for mass and μ for coefficients of friction. However, many upper-level mechanics books use μ for reduced mass. This can get awkward if you want to use reduced mass in a problem with coefficients of friction.

If the semantics could be unambiguously specified, the symbols peculiar to the problem wouldn't matter as much, and symbol collisions could be detected and averted more easily, perhaps even automatically if the software were given a list of candidates for symbolizing a quantity.

8.2 Automatically converting existing documents.

It might possible to identify each equation in, for example, a LaTeX document and automatically convert the units from, say, SI to cgs or vice versa.

In practice, however, because of things like LaTeX macros and ambiguity of representation, automatic conversion tends to be fragile. Some transformations are best done manually or with the aid of ad-hoc search-and-replace text-processing tools.

A more promising possibility is to develop standards like OpenMath with the goal of making it trivial to shift between notations and conventions. This will be largely invisible in terms of form, but such functionality will make collaboration and reuse easier and more robust.

For example, a textbook with basic semantics of unit systems built-in could be used by an engineering class with customary units, an introductory physics class with SI units, or as a supplement to an existing work in cgs units with no confusion or tedious manual conversion.

It could also link each use of an equation back to the derivation as a kind of dependency graph that makes the hierarchical structure of the mathematics explicit.

8.3 "Fingerprints" for equivalent mathematical expressions.

Regardless of the notation used in, for example, the Pythagorean theorem, there are always three independent variables. There is also always two operations (self-multiplication and addition), or three if we count equality as an operation. Also, the commutativity properties means that symmetry groups can be used to describe the equation.

These examples of notation-invariant properties could potentially be tabulated and systematized as a kind of fingerprint for a large number of commonly-used expressions, which could make recognizing mathematical patterns easier, since familiarity with the particular notation would not be necessary to recognize the pattern.

Ideally the fingerprint would be robust enough to help identify a given equation's sub-expression as equivalent or related to another equation.

For example, an expression of the form

$$a = \frac{bc}{b+c}$$

will get many physicists thinking about electronics, since the equivalent expression

$$\frac{1}{a} = \frac{1}{b} + \frac{1}{c}$$

corresponds to adding resistors in parallel or capacitors in series. However, the symmetry is obscured somewhat in the first form.

A desirable mathematical fingerprint would bring this pattern out into the open, and would extend to helping identify sums of any number of reciprocals.

8.4 New ways of seeing and reasoning about equations.

Once an equation can be parsed into a machine-manipulable format, there are many possibilities, including color-coding, dependency graphs to keep track of which parts depend on a given quantity, and whether it is separable into functions of the variables it depends on.

For example, Richard Feynman had grapheme-color synesthesia.

> "When I see equations, I see the letters in colors – I don't know why. As I'm talking, I see vague pictures of Bessel functions from Jahnke and Emde's book, with light-tan j's, slightly violet-bluish n's, and dark brown x's flying around. And I wonder what the hell it must look like to the students."

It might have looked something like this.

$$J_n(ix)J_n'(x) - iJ_n(x)J_n'(ix) = 0$$

With sufficient metadata about an equation, it would be possible to write a filter that would automatically color-code the expression. For example, functions, variables, constants, and integers (e.g. indices) could each have their own color.

Color-coding would provide an extra degree of freedom for defining quantities and dramatically increase the information density of an equation. Furthermore, it would provide a kind of rudimentary visual type system for mathematics.

Equation metadata would provide insight into what is invariant and fundamental in an equation and what is an artifact of conventions such as positive and negative charge, origin, gauge, basis, and orientation (left or right handedness) of a vector space.

8.5 Reasoning about physical systems and connections between them based on mathematical structure.

The practical purpose of associating equivalent forms is to save time and effort. The greater purpose is to make explicit connections between seemingly disparate systems.

There are many famous examples of the same equation describing seemingly unrelated phenomena. For example, the scalar wave equation describes vibrations in both solids and

fluids, and similar equations describe electromagnetic waves and quantum mechanical wavefunctions. The hydraulic analogy provides a useful and intuitive (though potentially misleading) way to reason about electronic circuits.

However, such connections are not isolated to famous results; such connections are discovered or rediscovered regularly. Such connections could be made more quickly and rigorously if governing equations, symmetries, and boundary conditions were explicitly stated in a way that could be compared to other systems to test for equivalence.

Physicists delight in the rich mathematical structure of the systems they study. Sadly, they are all too often divided from the mathematicians and from each other by mere notation. By unifying and linking the language of mathematics, we can avoid reduplication of effort and make explicit connections which were otherwise unknown or neglected.