Live Mathematics on Authorea

A Case for Transparency in Science

Deyan Ginev[1], Alberto Pepe[2], Nathan Jenkins[2]

[1] Computer Science, Jacobs University Bremen, Germany
[2] Authorea, New York, USA

Abstract. Authorea is a collaborative platform for writing in research and education, with a focus on web-first, high quality scientific documents.
We offer a tour through our integration of technologies that evolve math-rich papers into transparent, active objects. To enumerate, we currently employ Pandoc and LaTeXML (for authoring), MathJax (for math rendering and clipboard), D3.js (data visualization), iPython (computation), Flotchart and Bokeh (interactive plots).
This paper presents the challenges and rewards of integrating active web components for mathematics, while preserving backwards-compatibility with classic publishing formats. We conclude with an outlook to the next-to-come mathematics enhancements on Authorea, and a technology wishlist for the coming year.
An "active" version of this paper, demonstrating the discussed features can be found at [GJP15].

1 Introduction

The motivation behind creating Authorea has been to help streamline academic collaboration in writing any flavor of scientific documentation, notably research papers aimed at passing peer-review and getting published as scientific proceedings. While the authorship and submission experience comes first, a goal that comes close second is to also increase the openness of the scientific process, using the final publication as a "looking glass" into the practices and data collection which happened "behind the scenes".

We proceed to motivate why transparent research has superior properties and use "live mathematics" as one example of how Authorea enables it.

The core of the transparency problem is that we are still using the original publishing metaphor for documents, dating back to the innovations of 16th century Galileo Galilei, while simultaneously working on 21st century projects which are potentially large-scale, high-dimensional, multi-author and/or internationally distributed [GPB$^+$14]. The usual scientific document submitted to academic venues today is still oriented towards the printed page, remains opaque to the underlying data, of which it presents static snapshots, and is constrained by page count and margin sizes, often preventing it from providing sufficient detail of methodology and experimental setup.

This disconnect between experimental results and publications offers room for unintentional bias and experimental defects to remain unnoticed, making it difficult for reviewers to verify, and for follow-up experiments to continue the work in question. Studies have shown that even journals of the highest impact factors are vulnerable to retractions – see Fig. 1 for an illustration derived in [FC11]. In 2015 we have also observed a stream of high-profile retractions from some of the best scored journals that illustrate this problem, as tracked and discussed on the website[3] of the recent Retraction Watch initiative [MO11].

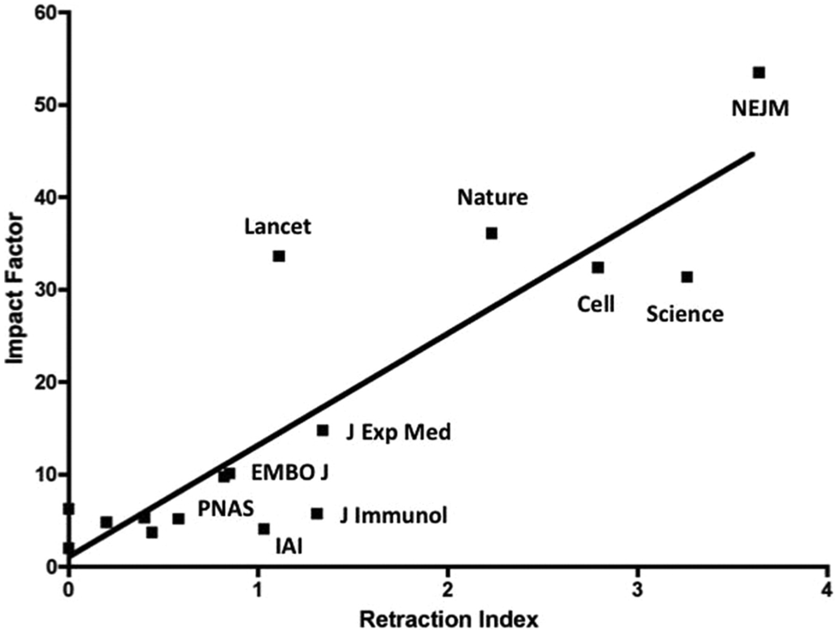

Fig. 1. As shown in [FC11]: "Correlation between impact factor and retraction index. The 2010 journal impact factor (37) is plotted against the retraction index as a measure of the frequency of retracted articles from 2001 to 2010 (see text for details). Journals analyzed were *Cell, EMBO Journal, FEMS Microbiology Letters, Infection and Immunity, Journal of Bacteriology, Journal of Biological Chemistry, Journal of Experimental Medicine, Journal of Immunology, Journal of Infectious Diseases, Journal of Virology, Lancet, Microbial Pathogenesis, Molecular Microbiology, Nature, New England Journal of Medicine, PNAS, and Science*."

[3] `http://retractionwatch.com`, seen June 2015

1.1 Facets of Transparency

To contrast, we offer a brief enumeration of the positive impact of the transparency of methodology and data on the scientific process:

Reproducibility Correctly repeating an experiment, or reproducing a proof, while arriving at the same results is foundational for establishing scientific truths. That is only possible for third-party scientists if the process is described in full detail in the original publication. That includes a range of diverse techniques, from experimental protocols and equipment specifications to exact computational methods and programs, as well as mathematical proof steps and derivations.

Reusability Building on, as well as improving, results achieved in prior work depends on first being able to reproduce them, and then being able to modify each step with enhancements or customization relevant to the follow-up experiment. That is only possible if there are no "black box" components in the methodology, i.e. where any step is open to both scrutiny and modification.

Accessibility While classically referring to people with disabilities, we use the term "scientific accessibility" in a broader sense. The dissemination of published works could be limited not only by impairments of the reader, but also by a language barrier (both geographically and in terms of terminology and mathematical notation used in different fields), by a technological barrier (e.g. use of closed, proprietary standards or badly maintained custom tools) as well as data blackouts (e.g. disconnect from the underlying datasets summarized by a paper's figures and tables).

Availability A substantial prerequisite for using a published result as a building block for follow-up work is the ease of access and quality of curation of all referenced materials and datasets. This could be problematic if resources are located behind institutional "paywalls" or restrictive copyright licenses, are too old to be in digital form, or simply remain unavailable for public review due to being considered too minor to be of importance.

1.2 Live Mathematics

The vision of "Live Mathematics", is a subset of the feature set captured by the "Active Documents Paradigm" for STEM [KCD$^+$11]. We aim to enhance the transparency of mathematical content, by providing the capabilities to attach underlying numerical data, to encode the mathematical properties as targeted programs, embedded in the document, and by feeding that active data into interactive visualization engines, allowing for an exhaustive and immersive understanding of the objects of analysis.

In this paper, we demonstrate an integration of technologies aimed to "illustrate" and "expose" the mathematical content to readers. We are not covering technologies aimed to assist with "verifying" or "co-deriving" mathematical results, such as proof-assistants and automated theorem provers, although we consider them viable future integration projects with Authorea.

The preprint of this paper, and simultaneously the demonstration of all described capabilities, are cross-hosted on Authorea and can be found at [GJP15].

2 Textual Mathematics

Authorea supports writing of web-first scientific texts in MarkDown (via Pandoc [Mac15]) and LaTeX (via LaTeXML [GSMK11]). Both utilize the battle-proven LaTeX syntax for mathematical formulas, which has become the ubiquitous approach to entering mathematics on the web.

In order to display the equations in all modern browsers, Authorea relies on the MathJaX polyfill engine [Cer12], as there are still browsers which do not yet natively render MathML, the mathematics sub-standard in HTML5. Using web-born mathematics has already allowed us to add editing-oriented math services, such as a math-aware word count and MathJaX's math clipboard.

Example 1. As an example, we write down the probability density of the **normal distribution**:

$$\varphi(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

3 Interactive Plots

In printed publications, static plots of mathematical objects are commonly provided in order to better demonstrate key properties. While that technique goes a long way, when dimensionality increases it is not possible to contain all relevant information in a single static snapshot. The ability to observe changes with the variation of parameters, or to filter down to specific dimensions of interest, is a powerful tool in developing a full understanding.

With adding interactivity, we depart the printed page metaphor and focus on web documents. For completeness, we remark that interactive Authorea figures and plots are intended to export to classic PDF documents, with a user-selected static snapshot as a substitute for the live figure.

The ubiquitous method for creating interactive HTML5 content is by using active components written in JavaScript. Prior to active documents, there have been solutions created as desktop applications and used as tools independent from the published work, e.g. GeoGebra [Ven09]. Currently, there are both custom applications leveraging the open HTML5 standards, such as DLMF's use of

WebGL plots for visualizing Complex Function Surfaces [SAWM15], and a variety of community efforts for creating standard libraries for interactive plots. Authorea already provides quick-start templates for two of those libraries: Flotchart [Lau15] and Bokeh [dV15].

A demonstration of using Flotchart to explore the effects of changing the normal distribution's mean and standard deviation can be found at [GJP15].

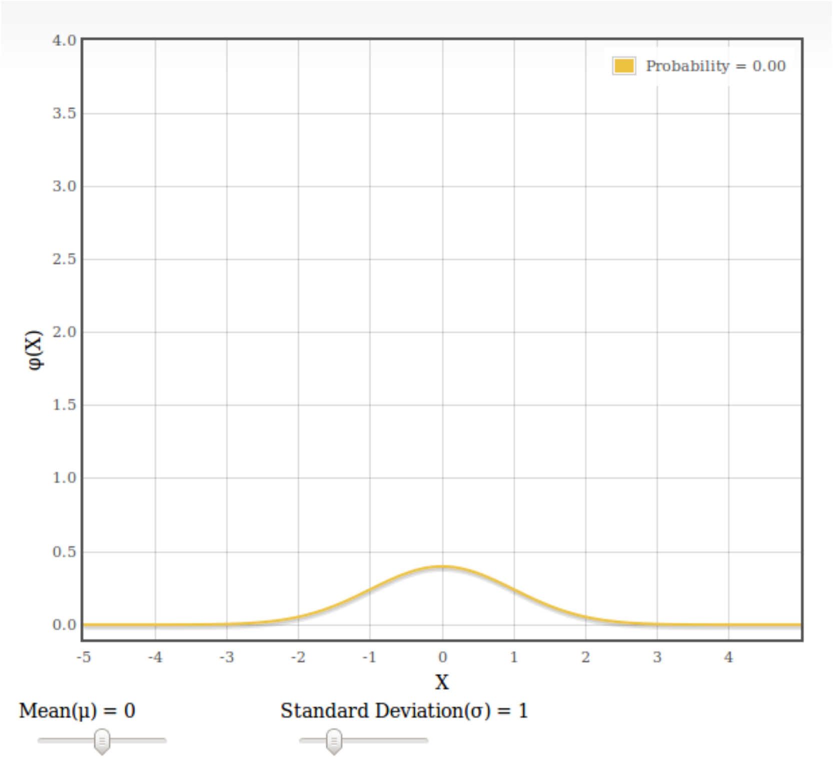

Fig. 2. Exploring the Normal Distribution from Equation 1 in Flot.js

4 Visualizing Data

A different interactive technique is to explore an underlying dataset, in order to empirically verify whether it follows a proposed model. In mathematics, an example would be sampling from probability distributions, or trying to fit a claimed to be normally distributed dataset (such as the height of a population of people) against the normal distribution. By being able to explore datasets directly in the paper describing them, both reviewers and readers have a more powerful handle on the properties inherent to the data. While open to other alternatives, Authorea currently offers support for this workflow via D3.js [Bos15], as well as a quick-start template and help articles.

An example of a D3.js figure, sampling from the normal distribution, can be seen in the web version of this paper [GJP15].

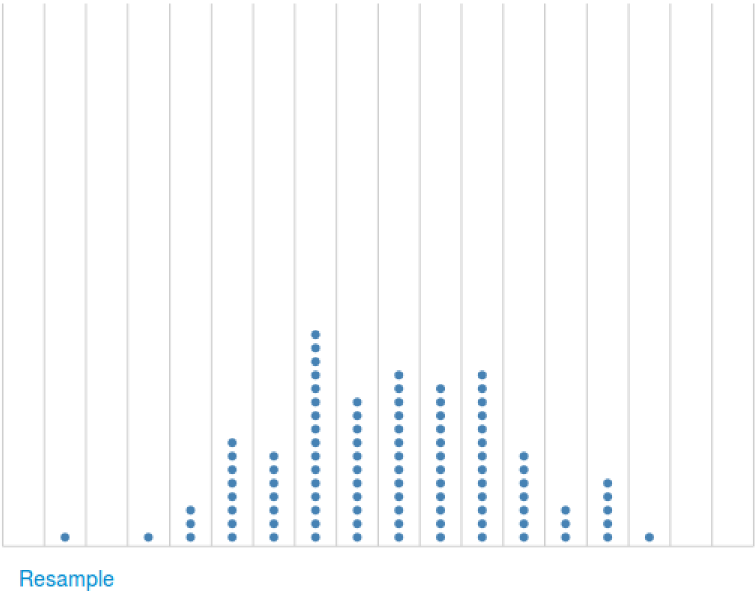

Fig. 3. Sampling from the Normal Distribution in Equation 1, via D3.js [Bos15]

5 Embedded Computational Objects

The last workflow we want to cover is close to the heart of computational and modeling results.

Traditionally, a paper would hint at a dataset, provide pseudo-code for the algorithm behind a model, and then provide a static plot that matches the mathematical expectations with experimental observations. In our "live math" features, we now also provide a capacity for "live algorithms", by enabling iPython Notebooks [PG07] in Authorea. The literal programming approach of iPython allows for a narrative exposition of an algorithm, alongside its actual implementation. Our Authorea integration also provides capabilities to directly embed, or download, a dataset of interest in order to perform live experiments, as a reader is exploring the paper. Unlike JavaScript solutions, the iPython route also provides standardized and well-maintained scientific libraries, such as SciPy and NumPy [LK13], which are easier to review and trust.

At Authroea, we remain open to embedding other scientific computational tools, as long as they have a transparent engine and data model. In fact, supporting the different data needs of the diverse academic communities is one of our long-term goals.

The current setup of adding computation to Authorea articles is one we refer to as “embedded blocks”. The author creates a static figure and attaches an iPython notebook “behind” it, which can be activated on demand. Ideally, the static figure represents the result of the iPython computation. This “computation behind figures“ is a more general technique: we can also imagine “computation behind theorems” where authors would attach formal proofs in a proof-assistant behind their natural language counterparts.

In [GJP15], we also provide a simple iPython example of computing and plotting the normal distribution.

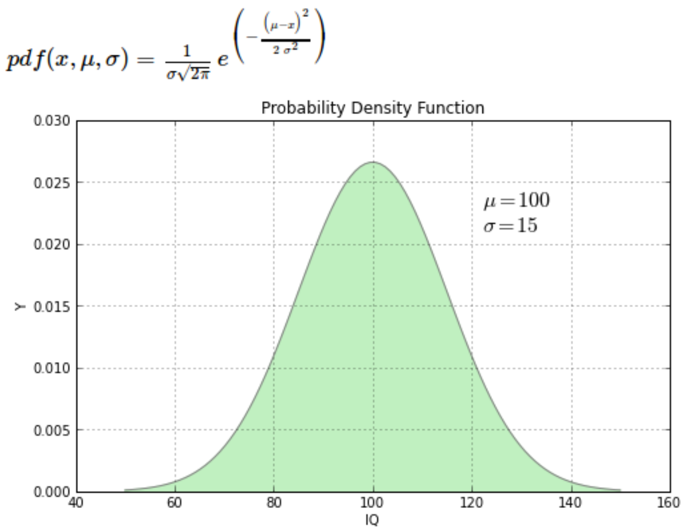

Fig. 4. Computing with the Normal Distribution in iPython [PG07]. Example taken from `http://www.arachnoid.com/IPython/normal_distribution.png`

6 The Authorea Publication Life-cycle

We briefly discuss the life-cycles of mathematical content on Authorea, as supported by the current implementation and necessitated by the publishing workflows in academia.

6.1 Authoring Life-cycle

An Authorea paper mimics the life-cycle of classic publications. An author can designate each stage of their paper as “rough draft”, “working draft”, “preprint”, “peer-reviewed” and “postprint”.

Each stage is intended to be a realized as a versioned tag, allowing for both a transparent understanding of the writing process – by examining the writing history – as well as for a clean handle on each life-cycle stage. This increases the potential compatibility with copyright policies, where the preprint can be offered as a free public open-access copy, while the authors continue to improve the latest peer-reviewed version towards the final postprint copy, both of which may (unfortunately) need to remain behind a publication “paywall”. While we see no technology barrier to “living papers” that continue to be constantly improved on Authorea, without ever being considered “finished”, our current focus is on publications that are considered finished after all peer-review remarks have been addressed.

6.2 Identifiers and Notability

First, we consider all modalities of content (mathematical formulas, tables, figures, etc.) as components of the published scientific work. A successful scientific study concludes with a written publication, which acts as its identifier to the wider scientific community, usually realized via a DOI identifier for the proceedings, and a unique URL for the Authorea preprint.[4] We remain open and compatible with combining supplementary data and information as attachments to the main Authorea paper. In cases where the data surpasses either certain size or notability thresholds, we consider it deserving of a separate DOI identifier and publication, and envision a paradigm of “active citations” of such datasets. By that we understand that a citation to an external dataset, when connected to an Authorea active figure, could dynamically compute views into the external experimental data summarized by the figure. While “active citations” are not yet available on our platform, they are part of the Authorea vision going forward.

To summarize, we expect scientific bodies of work of sufficient notability to receive unique identifiers, with independent recognition for data, descriptions and tools, and expect a rich computational interplay between them in the foreseeable future.

[4] Work is in progress for minting separate DOI identifiers also for Authorea preprints.

6.3 Sharing Life-cycle

The sharing life-cycle is an independent, orthogonal, dimension to authoring. During each authoring stage, the paper can receive feedback and peer-review, in the form of localized comments. Authorea allows customizable privacy for this workflow, and can enable access for a variety of third-parties – from co-authors and colleagues, to anonymous reviews and social media feedback. Crucially, we allow sharing of concrete components of a paper, by allowing e.g. localized links to individual formulas, figures, citations, etc. by utilizing URL fragment identifiers.

6.4 Technology Life-cycle

A common concern in adopting cutting-edge tools is the shelf life of their technology stack. Indeed, as technology keeps evolving, older projects may "bitrot" to an extent where they are impossible to casually use, as programming languages, operating systems and web resources move on. We have two responses to this concern.

Shelf-life affects everything Technology is not an exception to an otherwise stable and reliably constant set of scientific communication tools and practices. Both mathematical notation [Caj30] and natural language itself [PTHS12] continuously reinvent themselves at the margins of cultural memory. We observe a natural evolutionary process, where natural selection reaffirms the useful and discards the erroneous or the unneeded.

Science reinvents itself Of course, recognizing the natural nature of the process does not equate to claiming archival is an unimportant goal. In fact, we would go as far as to take for granted an acceptance of the universal value of archival, in cultural, scientific and historic terms. We would consider it a part of our "Accessibility" discussion in Section 1.1, where we tried to justify that transparent and open methodologies provide the needed foundation for accessibility.

This is also our approach to the archival problem in technology - using open standards for representation (e.g. HTML5) and open tools for computation (e.g. D3.js, iPython) provides a solid foundation for recovering the original content and functionality of a 21st century publication for the archaeologist of the next millennium.

7 Conclusion

We presented a single-system integration of "Live Mathematics" tools in Authorea. The added value to each component is two-fold. First, we provide a comprehensive authoring experience, where we attempt to streamline and simplify writing different types of active mathematical content. Second, and more importantly, by committing to open standards and technologies, we increase the transparency of the scientific process, while at the same time automating the dual nature of publications as printed proceedings and “live” active documents on the web. The final paper’s data, mathematics and algorithms are inter- and intra-connected, becoming a self-contained and reliable nutshell of the successful scientific process.

7.1 Future work

The bulk of our future efforts will be invested in improving the authoring experience and streamlining the different interactivity components on Authorea. We want to have the smallest set of authoring languages independently existing in the same workflow. We plan on achieving that by adding structural editing components that hide high difficulty low-level components, as well as by adding conventions and best-practices that help streamline common usecases. Additionally, we want to have all data and computation cleanly accessible during both authoring and consumption of the final publication.

Going in a different direction, we want to talk to more scientists, from a diverse range of fields, and look into embedding 21st century transparent workflows with their best web components for interactivity and data analysis.

In the long run, we would like to have an intuitive and minimalistic authoring experience, with “live” components auto-generated on demand, for each mathematical equation. Examples are on-demand interactive plots of computable functions, as well as quick-starting an iPython notebook from the math content of a paper’s section, potentially as a new playground for follow-up work. We already offer easy cloning of existing interactive mathematics, via “forking” the underlying git repository of an Authorea article.

7.2 Technology Wishlist

1. For accessibility purposes, it would be great if all major browsers support MathML a year from now, so that we could compartmentalize our use of JavaScript to be for interactivity only.
2. It is great that standard JavaScripts APIs for interacting with data are beginning to emerge, and we are looking forward to them maturing and ideally becoming independent of the concrete libraries that implement them.
3. We are also looking forward to advances in responsive interactivity, which make it easier to provide active mathematical widgets on mobile and tablet devices. As their computational resources are much weaker than the desktop, it may be interesting to investigate sever-side workflows that precompute most of the necessary functions.

References

Bos15. Mike Bostock. D3.js: Data-driven Documents. `http://d3js.org`, 2015.

Caj30. F. Cajori. A History of Mathematical Notations. Vol. I Notations in Elementary Mathematics. *The Mathematical Gazette*, 15(208):170, jul 1930.

Cer12. Davide Cervone. MathJax: A Platform for Mathematics on the Web. *Notices Amer. Math. Soc.*, 59(02):1, feb 2012.

dV15. Bryan Van de Ven. Bokeh: a Python Interactive Visualization Library. `http://bokeh.pydata.org`, 2015.

FC11. F. C. Fang and A. Casadevall. Retracted Science and the Retraction Index. *Infection and Immunity*, 79(10):3855–3859, aug 2011.

GJP15. Deyan Ginev, Nathan Jenkins, and Alberto Pepe. Live Mathematics on Authorea: A Case for Transparency. `https://www.authorea.com/41811-Live-Mathematics-on-Authorea`, 2015.

GPB$^+$14. Alyssa Goodman, Alberto Pepe, Alexander W. Blocker, Christine L. Borgman, Kyle Cranmer, Merce Crosas, Rosanne Di Stefano, Yolanda Gil, Paul Groth, Margaret Hedstrom, David W. Hogg, Vinay Kashyap, Ashish Mahabal, Aneta Siemiginowska, and Aleksandra Slavkovic. Ten Simple Rules for the Care and Feeding of Scientific Data. *PLoS Computational Biology*, 10(4):e1003542, apr 2014.

GSMK11. Deyan Ginev, Heinrich Stamerjohanns, Bruce R. Miller, and Michael Kohlhase. The LaTeXML Daemon: Editable Math on the Collaborative Web. In *Lecture Notes in Computer Science*, pages 292–294. Springer Science + Business Media, 2011.

KCD$^+$11. Michael Kohlhase, Joseph Corneli, Catalin David, Deyan Ginev, Constantin Jucovschi, Andrea Kohlhase, Christoph Lange, Bogdan Matican, Stefan Mirea, and Vyacheslav Zholudev. The Planetary System: Web 3.0 & Active Documents for STEM. *Procedia Computer Science*, 4:598–607, 2011.

Lau15. Ole Laursen. Flot: Attractive JavaScript Plotting for jQuery. `http://www.flotcharts.org`, 2015.

LK13. Thomas Lindblad and Jason M. Kinser. NumPy SciPy and Python Image Library. In *Biological and Medical Physics Biomedical Engineering*, pages 35–56. Springer Science + Business Media, 2013.

Mac15. John MacFarlane. Pandoc: a Universal Document Converter. `http://pandoc.org`, 2015.

MO11. Adam Marcus and Ivan Oransky. Science publishing: The paper is not sacred. *Nature*, 480(7378):449–450, dec 2011.

PG07. Fernando Perez and Brian E. Granger. IPython: A System for Interactive Scientific Computing. *Computing in Science & Engineering*, 9(3):21–29, 2007.

PTHS12. Alexander M. Petersen, Joel Tenenbaum, Shlomo Havlin, and H. Eugene Stanley. Statistical Laws Governing Fluctuations in Word Use from Word Birth to Word Death. *Scientific Reports*, 2, mar 2012.

SAWM15. Bonita Saunders, Brian Antonishek, Qiming Wang, and Bruce R. Miller. Dynamic 3D Visualizations of Complex Function Surfaces Using X3DOM and WebGL. In *20th International Conference on 3D Web Technology (Web3D 2015)*, 2015.

Ven09. Gerard A. Venema. The Elements of GeoGebra. In *Exploring Advanced Euclidean Geometry with GeoGebra*, pages 13–22. The Mathematical Association of America, 2009.