



Transformative Approaches in Scientific Publishing

Using MyST Markdown and Curvenote tools to encourage continuous science practices

Rowan Cockett , Steve Purves , Franklin Koch , Mike Morrison

July 10, 2024

Abstract

should be -- (big dash)

Difficult problems that are globally relevant and urgent like climate change, pandemics, and water security - require us to radically rethink how we publish and work on scientific breakthroughs. Advances in technology for data workflows have increased the speed and scope of scientific discovery. However, scientific dialogue still uses outdated technology for communicating and sharing knowledge, relying on static PDFs that are a poor representation of the complexities of science. This gap impedes the speed of research dissemination, reuse, and uptake: we require new

mediums to compose ideas and ways to share research findings as early and as often as possible; this gap is even more evident in the computational sciences. In this paper we discuss two tools for scientific dialogue, MyST Markdown and Curvenote, and illustrate examples of improving metadata, the reading experience, and transforming publishing practices for individuals and societies.

grammar off w/ this list. The first 2 are attached to "improving", but the third isn't

Keywords: scientific communication, publishing, open science

1 Introduction and Motivation

In the face of mounting global challenges such as climate change, pandemics, and water security, the imperative for rapid, effective scientific discovery and dissemination has never been more acute. The pace at which these problems evolve and impact societies worldwide demands an equally dynamic and innovative approach to how scientific research is published. Despite significant advancements in technology that enhance data collection, analysis, and workflow efficiency, the mechanisms through which scientific knowledge is shared have remained largely unchanged for decades (Bourne *et al.*, 2012). The prevalent reliance on static PDF formats for research papers starkly contrasts with the complex, data-driven and increasingly computational nature of modern science, creating bottlenecks in knowledge dissemination and uptake.

unclear what this is referring to bc it is new paragraph. It would be helpful to clarify you are referring to scientific communication/publishing

This paper documents some of the design decisions made to address these challenges within two integrated tools: (1) MyST Markdown (Markedly Structured Text, <https://mystmd.org>), a community-run open-source project^[1], which is a text-based authoring framework that integrates computational content (e.g. Jupyter Notebooks); and (2) Curvenote (<https://curvenote.com>), which is a set of open-source utilities, command-line tools, actions and services^[2] aimed to improve scientific publishing by journals, societies, lab-groups, and individuals. In this article we provide background, motivation and perspective for our efforts in developing new open-source tools for science communication as well as provide examples from individuals, lab-groups, proceedings, and society journals. We give an overview of MyST Markdown but would like to reiterate that MyST Markdown is a community-run project, we do not speak for all of the participants, and the community

has varied goals for the project (including API documentation, community guidelines, educational tutorials). Our focus in this article is to give our perspectives on scientific writing and publishing and how it intersects with these open-community projects in addition to the open-source efforts that Curvenote is undertaking around scientific publishing.

Nice! I appreciate the clarification in scope.

Our goals in developing these integrated tools and workflows address the related challenges of *authoring* and *publishing* in the context of computational, open-science documents and lowering the barrier to continuously releasing and **iterating on scientific ideas in the open**. Introducing authoring tools that allow for structured, interactive and computational content as researchers are writing can additionally change the way it is checked, shared and published — enabling faster iterations and direct ties to reproducible, interactive content.

This is fine for this audience, but for the future, it would be good to articulate what you have in mind in a bit more depth. eg I don't think many get stuck rapidly publishing every step of their work is "castles in the air" - having a "called paper" to make some safety points gives ideas to add before they are ready to be shared.

1.1 Authoring Structured Content

There are currently many challenges for individuals or groups to author research information that can be shared in a structured and rigorous way. By this we mean the things that *structurally* set a scientific article apart from, for example, a blog post: structured content, cross-references, valid citations with persistent identifiers (PIDs), and standardized metadata for licensing, funding information, authors and affiliations. This structured content and metadata as well as the standards behind them are what defines the "scientific record" and enables archiving, discoverability, accessibility, interoperability and the ability to reuse or cite content ([Wilkinson et al., 2016](#)). One metric for measuring the difficulty of satisfying these scientific standards is to look at the direct costs that are spent on transforming author submissions (e.g. a PDF or a Word Document) into something that conforms to these standards and is ultimately archived. In scientific publishing, about 15% of Article Processing Charges (APCs) go to direct publication costs^[3] ([Grossmann & Brembs, 2021](#)), which provides a very rough global estimate of over USD\$2 billion dollars^[4] that is spent on transforming author submissions (e.g. a word-document, LaTeX, or a PDF) into a copyedited, well-formatted, typeset document that can be archived with appropriate metadata ([Hagve, 2020](#)); this estimate does not include the approximately USD\$230 million spent on reformatting articles by scientists *before* publication ([Clotworthy et al., 2023](#)). Many of these processes are hidden from authors^[5] as well as actionable access to the many of the benefits of structured data beyond citation graphs (e.g. "How many citations did this paper get?").

↑ It isn't clear if this is an example or a counterexample. I can answer this with google scholar, what else are resources missing out on?

One goal of the MyST Markdown project is to *dramatically* reduce these direct-publication costs^[6] and directly provide structured data as an output of authoring as well as the ability to export content in a variety of formats including HTML, PDF and JATS-XML (a NISO standard for archiving scientific articles). In this article, we will demonstrate that having structured data throughout authoring leads to a number of novel reading and authoring experiences (e.g. [Example 1](#)), can provide new opportunities for reuse and quality checks when publishing (e.g. [Example 3](#)). Furthermore, these transformation processes can be run *continuously* opening the possibilities for faster feedback ([See Section 1.3](#)), iterative drafts, small tweaks and versioned improvements that otherwise would not be worth the time and cost.

1.2 Computational Articles

A compounding challenge to scientific publishing that we are exploring through MyST Markdown and Curvenote is how to deeply integrate computational workflows and content into science communication to promote interactive explorations and reproducible practices. There are a host of challenges from user-interface design, to maintenance, to archiving computational content. Many other tools have worked on aspects of integrating computation into scientific articles, notably R-Markdown ([Xie et al., 2018](#)) and its successor Quarto (<https://quarto.org>); both of these projects have similar aims to MyST Markdown. From a user-experience goals perspective, we are interested in questions such as:

- how to make a change in a notebook figure and have that immediately show up in a document?
- how to ensure that a p-value, for example, is computed and inserted directly from a computation, rather than through copy-and-paste?
- how to expose interactivity and exploration that a researcher often has when analyzing a data-set?
- how to provide and launch archived interactive computing environments?

These questions require authoring tools to be able to execute content, to integrate and display computational/interactive outputs directly in reading experiences, as well as scientific publishing systems that can understand and archive computational content (e.g. Docker containers). This deep integration can open up possibilities of embedding interactive visualizations and computational notebooks directly into scientific documents (e.g. [Example 2](#)), transforming articles from

← this isn't written as a question.
How do I/we ... ? or just leave as a
statement

↑ this is quite specific while all other examples are general

static texts into rich, interactive, reproducible narratives. In 2023, the authors helped to lead several working groups related to these challenges *Notebooks Now!*, a Sloan Foundation funded project led by the American Geophysical Union. Those working groups found that integrating computational documents, via Jupyter Notebooks, into scholarly publishing system requires a re-imagination of the publishing processes (from submission to peer-review to production to reading) and that many existing processes, platforms are ill-equipped to handle computational articles (c.f. [Caprarelli et al., 2023](#)). The “executable research articles” project out of eLife ([Tsang & Maciocci, 2020](#)) has similar aims to *Notebooks Now!*, with some differences in how notebooks and articles are separated which we will discuss in [Section 2.3](#).

The ability to deeply link computational content into how we communicate science can improve reproducible practices, and surface more ^{one word?} **inter-linked** content about methods and algorithms in use. If used to their full extent, these can also fully integrate live computational environments into scientific articles, which provides many exciting possibilities for interrogating and extending scientific data and methods.

1.3 Continuous Science Practices

The manual effort involved in article production ([Section 1.1](#)) coupled with the inability to integrate computational work ([Section 1.2](#)) negatively impacts the **number of iterations/versions** and the **immediacy of feedback to authors**^[7]. In other disciplines, such as software development, **these metrics** of iteration and rapid feedback are often highly encouraged, measured and constantly improved. For example, software organizations often measure and improve: the release cadence of a software product (e.g. continuous delivery); how confident you are in that release (e.g. based on continuous integration tests); how you get early feedback and confidence from linters and tests (e.g. the speed of your unit tests and integrated linters into development environments); and how fast you can obtain feedback from real usage and users on in-progress work (e.g. observability, analytics, customer interviews, design prototypes). Continuous delivery practices of software development are also extremely well studied, with large-scale surveys of organizational performance, design, robustness, and speed (e.g. [DORA Reports](#)). One study compared elite teams with low-performing teams for software delivery, and found elite teams were **46x faster** to release to production (i.e. on-demand and multiple times per day vs monthly or bi-

← reference to binder here or earlier?

not entirely sure what this is getting at? it implies more versions of articles is better, but I am not sure in what context? (sounds like more work for a reviewer...)

again, the context isn't totally clear there are huge opportunities to automate some aspects of testing or - submission, but other aspects (eg the scientific merit of an idea) take more time.

not sure what the metrics is!

* for me, this section is missing the discussion of which pieces can be automated and which cannot. Some things like archiving code, checking references, etc can benefit from automation, and there is currently a major gap here that AI can fill! But there is also an aspect that currently relies on humans reading the content being created, and this isn't just a technical challenge, it relies on the willingness of reviewers to provide input and that is a social challenge much more than a technical one.

annually) and had **7x fewer errors** (due in part to better continuous deployment and testing infrastructure as well as smaller changesets) ([2018 DORA Report](#))^[8].

The analogy between scientific publishing and software releases is imperfect and non-prescriptive (i.e. scientific research is very different than developing a product). However, the analogy is illustrative in areas where there is a focus on iterations, smaller changesets and releasing in-progress work as soon as possible to get feedback from peers (i.e. scientific peer-review) or users (in the case of software products). The speed of scientific progress depends *on part* on the speed of iteration and feedback. The time it takes for the peer-review process is over three months, in high profile journals like Nature that time has almost doubled over the past decade ([Powell, 2016](#)). Rejections are anywhere from 50-90% ([Grossmann & Brembs, 2021](#)) with valuable reviews and expertise coming months or even years after the work is completed^[9]. There are wide spread efforts in scientific publishing that focus both on sharing smaller components of research (e.g. FigShare ([Jones & Hahnel, 2016](#)), Octopus ([Martinez Garcia et al., 2023](#)), MicroPublications ([Clark et al., 2014](#)), NanoPublications ([Groth et al., 2010](#)), Protocols ([Teytelman et al., 2016](#)), PreRegistrations ([Nosek et al., 2018](#))) as well as sharing research sooner in the life cycle especially through preprints, for example, ASAPbio's mission to support "rapid open sharing, discussion, and collaboration around research" and promotion of "Preprints in Progress" ([Penfold & Polka, 2020](#)).

We refer to these related concepts as "*continuous science*", adopting language and concepts from "continuous integration and deployment". The mechanisms to support continuous processes are through automation, rapid feedback on errors, and focusing on small, rapid changesets to accelerate feedback from peers. This gives us a technical lens to assess, for example:

- How can you assess and test that the structure of a document applies to editorial rules?
- How long does it take to get feedback if your metadata or DOI is incorrect?
- When a computational figures or data output changes, how long does it take to integrate that into your document?

In science there is a highly manual, absurdly expensive and disconnected process between authoring and publishing. By moving to continuous practices and investing in the appropriate infrastructure to support *continuous science* we believe

there is an opportunity to accelerate scientific discovery by multiple orders of magnitude while simultaneously increasing reproducibility and robustness of the underlying science.

1.4 Article Outline

For research-communication to be transformative on a similar scale as open-source, researchers require modern tools for authoring and publishing. There are two inter-related capabilities that are necessary for this transition:

Software?

1. authoring mediums that support data, computation and structured content without the need for expensive typesetting; and
2. publishing that is open and accessible to researchers at a variety of scales – individual publishing, lab-groups, societies and institutions.

Through the lens of MyST Markdown and Curvenote, this paper will explore how these tools can help to address critical gaps in current scientific publishing practices. Our motivation is to enhance the *speed* and *impact* of research dissemination, fostering a scientific ecosystem that is more collaborative, reproducible, and equipped to tackle the urgent global challenges of our time.

2 Authoring Tools

MyST Markdown (Markedly Structured Text, <https://mystmd.org>) is a community-driven markup language that is a superset of **CommonMark** (a standard form of Markdown) with special syntax for citations, cross-references, and block and inline extension points called “directives” and “roles”. The block-level content provides multi-line containers surrounded by either backticks or colons; examples include callout panels, figures, equations and tables (see [documentation](#)). MyST is influenced by **reStructuredText (RST)** and **Sphinx** – pulling on the nomenclature and introducing additional standards where appropriate. There are also many similarities of MyST Markdown to R-Markdown ([Xie et al., 2018](#)), Pandoc, and Quarto (<https://quarto.org>), especially in citation syntax and frontmatter structure. The initial use case driving the development and

design of MyST Markdown has been JupyterBook, which can create educational online textbooks and tutorials with Jupyter Notebooks and narrative content written in MyST.

In 2022, the Executable Books project (<https://executablebooks.org>, which hosts Jupyter Book and MyST) started work on `mystmd` command line interface (CLI), which was initially developed as the Curvenote CLI, and later transferred to the ExecutableBooks project. At its core, MyST adheres to the CommonMark standard, ensuring a familiar foundation for those versed in traditional Markdown. However, MyST distinguishes itself through the introduction of specialized syntax aimed at facilitating scientific discourse including citations, cross-references, figures, proofs, equations, and tables. There is also specialized support for the types of metadata that are important to collect for scientific articles (funding, ORCIDs, CRediT Roles, etc.).

This article will not attempt to describe the markup syntax directly, for that we suggest browsing the documentation at <https://mystmd.org>, instead we will focus our attention on the use cases for scientific publishing that we are trying to make as easy as possible. Specifically, the ability to add persistent identifiers (PIDs) and links to other structured content; ^{comma} hover previews to show details on demand; ^{comma} integrating live computational content and interactive figures; and exporting to many different formats including those used by scholarly publishing.

2.1 Utility of Links and Identifiers

In MyST Markdown, citations can be added inline using `@cite-key`, following Pandoc-style syntax and referencing a BibTeX file in a project. It is also possible to directly link to DOIs using `@10.5281/zenodo.6476040`, which will create a hover reference ([Cockett, 2022](#)) as well as a references section at the bottom of the page.

This enhanced-links concept can be extended to Wikipedia, RRIDs, RORs, GitHub issues or code, and other scientific databases to augment writing. For example, the link `<rrid:SCR_008394>` becomes `RRID: SCR_008394`, with rich-metadata and citations created. Wikipedia links come with previews, for example, `<wiki:gravitational_waves>` becomes `gravitational waves`. GitHub links to pull-requests also give hover information (for example, this example admonition was implemented in [#87](#)).

A figure here could be impactful (as illustrated by me reading on my iPad 😊)

The use of DOIs and other structured scientific metadata can be reused in multiple different formats such as JATS XML and CrossRef deposits to create a DOI. Our goal with these integrations is to make the use of persistent identifiers (PIDs) both easy and rewarding to the author as they are writing; this metadata is generally only added at publication time.

2.2 Hover Previews for Content and Metadata

In [Head et al. \(2021\)](#) the authors show a speed up in comprehension of an article by 26% when showing information in context (i.e. “details on demand” ([Shneiderman, n.d.](#))), rather than requiring researchers to scroll back and forth to find figures and equations. MyST supports these concepts natively for cross-referencing equations, figures, and tables using hover-previews ([Example 1](#)). This enhances the reading experience of scientific documents and retrieval of information.

Example 1 (Hover and Dive Deeper)

Any figure, table, or equation can be referenced in MyST and in addition to automated numbering the cross-references have hover-references. This design feature is important for two reasons: (1) it improves reading comprehension; and (2) it focuses on structured data which can be accessible between papers, creating an open-ecosystem of machine-actionable, reusable content. The referenced content can also be interactive or computational.

This is nice! Another thing you could comment on is the necessity of supporting interactive content and the ability to fall back to simpler formats

Links & Cross-References

As you have seen in the links in MyST (e.g. [Frontmatter](#)), there is information that is pulled forward into your reading context on hover or click. We believe it is important to provide as much possible context when you are reading on elements like links to other pages, cross-references to figures, tables and equations as well as traditional academic citations^[2] (👉 **click the footnote!**). Additionally, all of these have fallbacks in static PDF or Word documents.

To link to a document, for example [Frontmatter](#), is done through a simple Markdown link `[]` (`./frontmatter.md`), you can put your own content in between the square brackets, but if you leave it out the link contents will be filled in with the title of the page. If you define the frontmatter on that page (i.e. the description and tooltip), you will also see that information when you hover over the link. This also works for links to Wikipedia (e.g. [Ponyies](#) 🐾) as well as Github code (e.g. [README.md](#)).

To create a cross-reference, you need to label a "target", like a figure, section, equation or table (or anything!!). To be referenceable, these elements can add the `label` option in many directives. To then reference the figure, use the link syntax again pointing to the label as the target `[](#my-fig)`. If you leave the title blank the default will fill in with an enumerated "Figure 1".

```
```{figure} https://source.unsplash.com/random/500x200/?mountain
:name: my-fig
:align: center

My mountain 🏔️.
```

Check out [](#my-fig)!!
```

IN THIS ARTICLE

- Overview
- Typography
- Directives and Roles
- Frontmatter
- Links & Cross-References**
- Citations
- What's Next?

Figure 1: Instantly accessible information can deep-dive link all the way to interactive figures. These practices help with reading comprehension by around 26% by providing information when the reader needs it (Head et al., 2021).

In MyST Markdown we have also extended this “details on demand” concept to abbreviations to make it trivial to disambiguate the meaning of acronyms. In an analysis of over 18 million articles in [Barnett & Doubleday \(2020\)](#), the authors found that the vast majority of abbreviations (79%) appeared fewer than 10 times and many abbreviations had conflicting meanings even in the same discipline. In MyST Markdown, there is a trivial way to document abbreviations in YAML frontmatter or the project configuration ([See Program 1](#)), these are then applied to all instances of that abbreviation in the article or notebooks giving a hover-preview and accessible HTML (e.g. try hovering over these in the online version: JATS, XML, VoR).

```
abbreviations:  
  UA: ulnar artery
```

Code 1: YAML metadata used in MyST frontmatter to give accessible hovers to all abbreviations with minimal effort for the author. In this case, the acronym **UA** has over 18 distinct meanings in medicine ([European Association of Science Editors, 2019](#)) not to mention other disciplines.

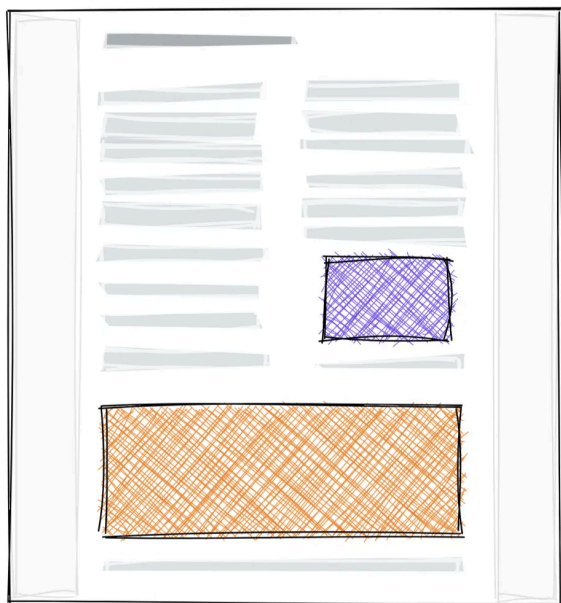
2.3 Integrating Computational Content

Beyond structured typography, integrated metadata and hover-previews, MyST Markdown understands computational content and has been integrated with Jupyter ([Kluyver et al., 2016](#)). The goal of this is two fold: (1) allowing for updates to computational content, figures, tables and calculations to directly update documents; and (2) to bring interactive figures and integrated computation directly into articles. In the composition of a scientific narrative, scientists often use individual notebooks to create various components of their research (e.g. the preparation of a single figure, table or calculation). The outputs of these notebooks (a figure, table, or calculation) can then be used in a main, narrative-driven document — a scientific article or presentation.

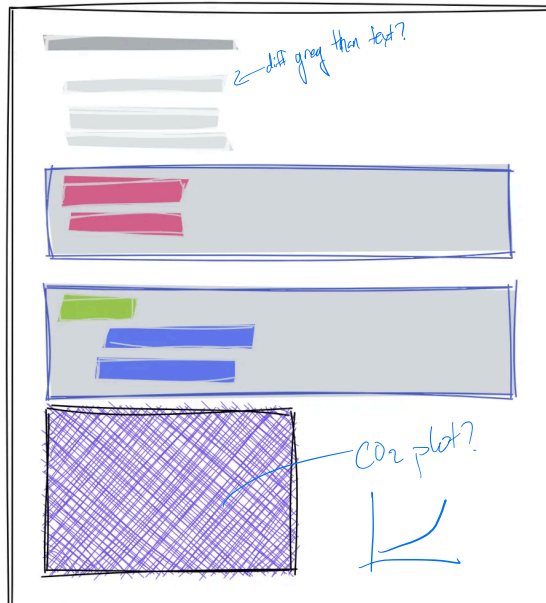
In some cases, it is possible to collapse all computational information into a single “Computational Article”, and visually hide the code to focus on the narrative or presentational flow ([c.f. Tsang & Maciocci, 2020](#)). This approach is appropriate for tutorials or the reproduction of visualizations rather than reproduction of a distinct detailed methodology that requires its

own explanation and/or lengthy computation. However, given the possibility of publishing a Computational Article, even in these cases authors can rethink how to communicate their work and prepare specific visualizations and compact results datasets to take advantage of the format. Another approach is to include supplemental notebooks that can capture those individual steps (c.f. [Caprarelli et al., 2023](#)), and *transclude* content ([Figure 2](#)). Both approaches are appropriate in different circumstances, and depends on the goal of the communication, nature of the research, speed of execution, and if individual steps require dedicated narrative explanation.

Scientific Article



Computational Notebook - Figure 1



Computational Notebook - Figure 2

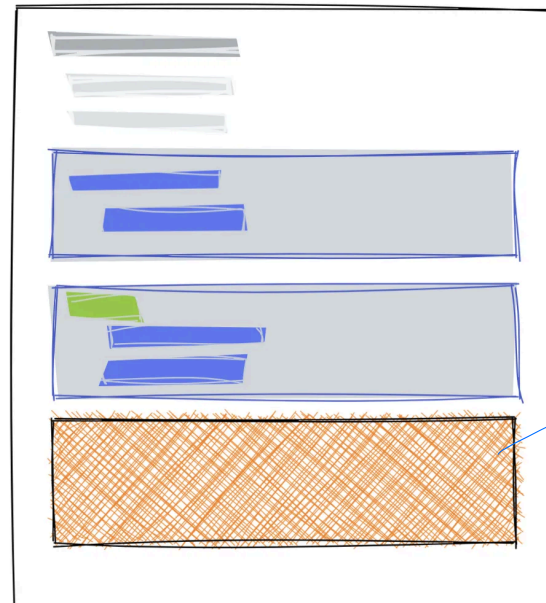


Figure 2: A schematic of embedding content from Jupyter Notebooks into an article. The purple and orange components, interactive figure or other computational outputs, are created in a computational notebook and subsequently used in other narrative or presentation-focused scientific article.

In [Figure 2](#), we show an example of reusing computational outputs, such as figures or tables directly in a single computational research article. By *embedding* these rather than using a screenshot or copy-paste, any changes to the

computational content can be immediately applied on a re-render. The embedding is completed through a simple MyST Markdown syntax that references a labeled cell in a Jupyter Notebook in, for example, a figure or table ([Program 2](#)).

```
:::{figure} #embedded-cell  
Additional caption.  
:::
```

Code 2: Embedding a cell from a supplementary notebook directly into a computational document by referencing the label/ID of the cell and adding a caption. The cell in the notebook must be labeled with a `#| label: embedded-cell` as the first line of the content.

Similar to the hover-references in [Example 1](#), this approach improves the metadata around the notebooks and exposes individual outputs or code-snippets to be imported into other documents or projects. Additionally, we can attach either static-interactivity (e.g. Plotly, Altair) or dynamic computation (e.g. BinderHub, JupyterHub or JupyterLite) to these figures to run live computations directly in the article ([Example 2](#)). Here we are aiming at a much richer, structured information commons that moves beyond just tracking scientific metadata towards easy-to-use tools that reuse scientific content.

Example 2 (Computational Reproducibility and Interactivity)

MyST allows for the full reproducible environment to be specified (via REES) and reproduced through tools like MyBinder. Figures can be integrated directly into articles, pressing a button to launch live and interactive figures that build upon the Jupyter ecosystem. These tools build on the Jupyter protocols and reuse components from the JupyterLab ecosystem, extending that into various pages using a package called thebe.

is there a reference for thebe?

The screenshot shows a web interface for an article titled "All Together Now". At the top, there is a navigation bar with a hamburger menu, the title "Pixels and their neighbours", a settings icon, and a "Download PDF" button. Below this, the article is categorized as a "TUTORIAL" under "THE LEADING EDGE". Social media sharing icons are visible. The authors listed are Rowan Cockett, Lindsey Heagy, and Douglas Oldenburg, all affiliated with the University of British Columbia. The article is dated August 1, 2016, with a DOI link. The main text discusses discretizing equations over a mesh, mentioning the use of Kronecker products and matrix operations. Two equations are presented:

$$\text{diag}(\mathbf{v})\mathbf{D}\mathbf{j} = \mathbf{q} \tag{1}$$

$$\mathbf{M}_f(\sigma^{-1})\mathbf{j} = \mathbf{D}^T \text{diag}(\mathbf{v})\phi \tag{2}$$

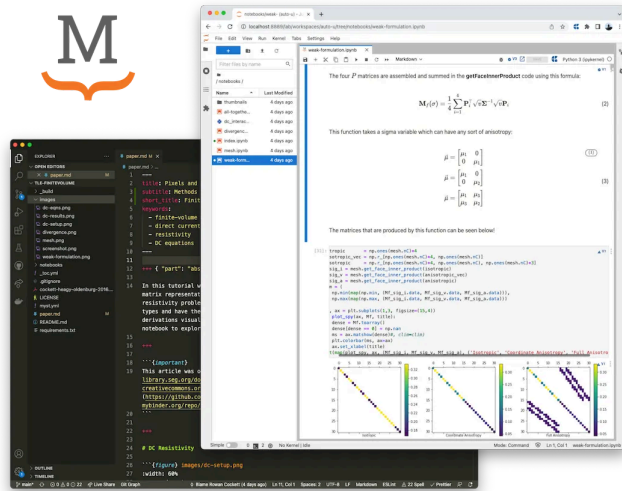
A note at the bottom states: "Note that now all variables are defined over the entire mesh. We could solve this coupled system or we could eliminate \mathbf{j} and solve for ϕ directly (a smaller, second-order system)."

Figure 3: Embedded notebook cells with live computation directly in an articles with computation backed by Jupyter. These can be running on BinderHub or directly in your browser through JupyterLite.

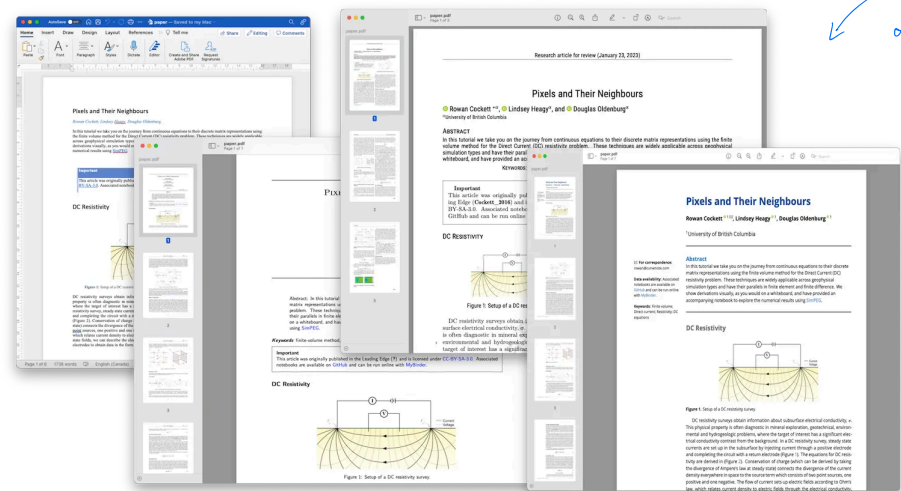
2.4 Single Source to Many Outputs

These capabilities of cross-references, typography and embedding visualizations and data-frames are complemented by an single-source export system that supports professional article templates and JATS XML (Figure 4). This is referred to as single-source publishing (c.f. Dunn, 2003), however, many implementations in scientific publishing focus first on manual translation to XML (e.g. from Word or LaTeX), rather than on an author-facing implementation.

Write Once



Export to any Template



Format?

show 'published version' or ArXiv? in a web browser

MyST Markdown & Notebooks

PDF + Word + LaTeX

Figure 4: Export to PDF using LaTeX or Typst is supported for hundreds of different journal templates in addition to Microsoft Word or JATS XML, which is used throughout scientific publishing.

Cite Pixels + Neighbors?

With single-source publishing, we can rely on rich transformations of the source content that can create professional PDFs, interactive HTML, and structured metadata such as JATS XML, which is the current standard for scientific archiving and text-mining.

3 Publishing

There are many ways to publish a MyST Markdown site. The native way that MyST is represented is through content-as-data, which is a collection of JSON files that represent the full documents and all associated metadata (if you are reading this online, you can add a `.json` to the URL to access the content), and enables dynamic serving of the content in a number of ways and when run locally, the MyST content server is dynamically creating HTML based on these JSON files (very similar to XML-based single source publishing (Dunn, 2003), but using modern web-tooling). This server-side approach has a number of advantages, in that it allows you to maintain a journal/site theme, without having to upgrade or rebuild all content as would be required by static tooling. It also puts content addressability as a first class concern (via the `.json`), enabling global cross referencing and opening up opportunities for varied publishing models in future. This is the approach that we take with Curvenote journals, and provide managed services to maintain journal sites, manage and curate content, as well as provide the editorial management tools and workflows. *Very cool! And clear-thinks*

You can also easily self-host a MyST Markdown site using GitHub Pages. To create a GitHub pages output you can run `myst init --gh-pages`, which will walk you through the steps of creating a GitHub action to publish your content as a static website. In this scenario, a static HTML site is built from your content which can be hosted as any other static website, while some of the advantages of dynamic hosting are lost, it is an easy and accessible way for individuals to self-publish.

In 2024, Curvenote was asked to improve our integrations to GitHub to support the SciPy Proceedings and re-imagine a MyST based publishing approach that uses GitHub for open-peer-review, implementing a submission, editorial and peer review process with GitHub issues, PRs and actions as a fabric. The process previously used technology shared with the Journal of Open Source Software (JOSS), which popularized this approach (Smith et al., 2018). The workflow was updated to use MyST Markdown for the authoring process^[10] (previously RST and LaTeX were supported, and the build process was in Sphinx), and the submission process now uses the Curvenote CLI in combination with dedicated Curvenote GitHub Actions to build, check and preview the content of each commit, using GitHub workflows to automate the process, providing **immediate feedback** for authors and the conference editorial team.

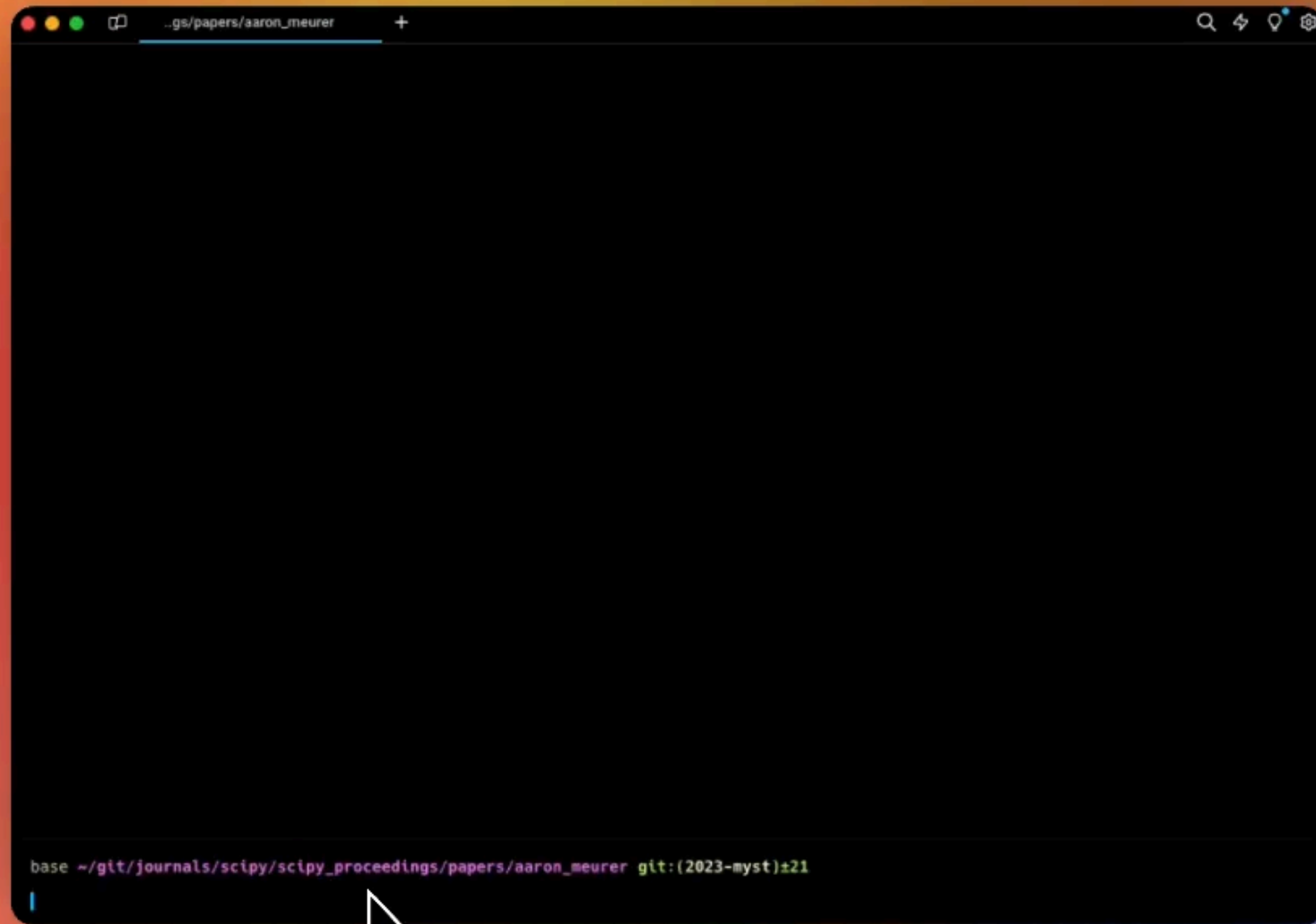
3.1 Structural Checks

The open-source Curvenote CLI (<https://github.com/curvenote/curvenote>) provides checks for the structure of a document to ensure it meets automated quality controls for things like references, valid links, author identifiers (e.g. ORCID), or funding information. Executing `curvenote check` in a MyST project will build the project and use the structured data to assess metadata (e.g. do authors provide ORCIDs), structural checks (e.g. does the article have an abstract; is the article below a word count?), and check that references have valid DOIs.

We have designed these checks in a similar pattern to linting and/or unit-tests in scientific software, which continually give feedback on the structural health of a code-base with near immediate feedback to authors (Example 3). Authors can take action to improve their metadata directly, including DOIs, CRediT Roles, ORCIDs and structural checks such as word-count or missing sections. For example, in the SciPy Proceedings in 2024, which used Curvenote checks for their submission system, required and optional metadata was improved by authors reacting to these automated checks without any intervention from the proceedings editorial team (e.g. [scipy-conference/scipy_proceedings#915](https://scipy-conference.github.io/scipy-proceedings#915) added CRediT roles, ORCIDs abbreviations, and DOIs to get a passing check (See Figure 6)). This is a low-friction way of improving metadata at the time of authoring or submission to elevate content to standards the standards that are required for a certain type of publication (e.g. proceedings vs blog post vs peer-reviewed journal).

Example 3 (Structural Checks and Metadata Checks)

Specific checks can be setup for any kind of content being submitted with MyST, for example, a "Short Report" might have different length requirements or metadata standards than a "Research Article" or "Editorial". Using Curvenote, these checks can be configured for a specific venue or kind of article, for example, to check specifically for SciPy Proceedings articles, `curvenote check scipy --kind Article --collection 2024` can be run, where the list of checks is configured remotely by journal administrators.

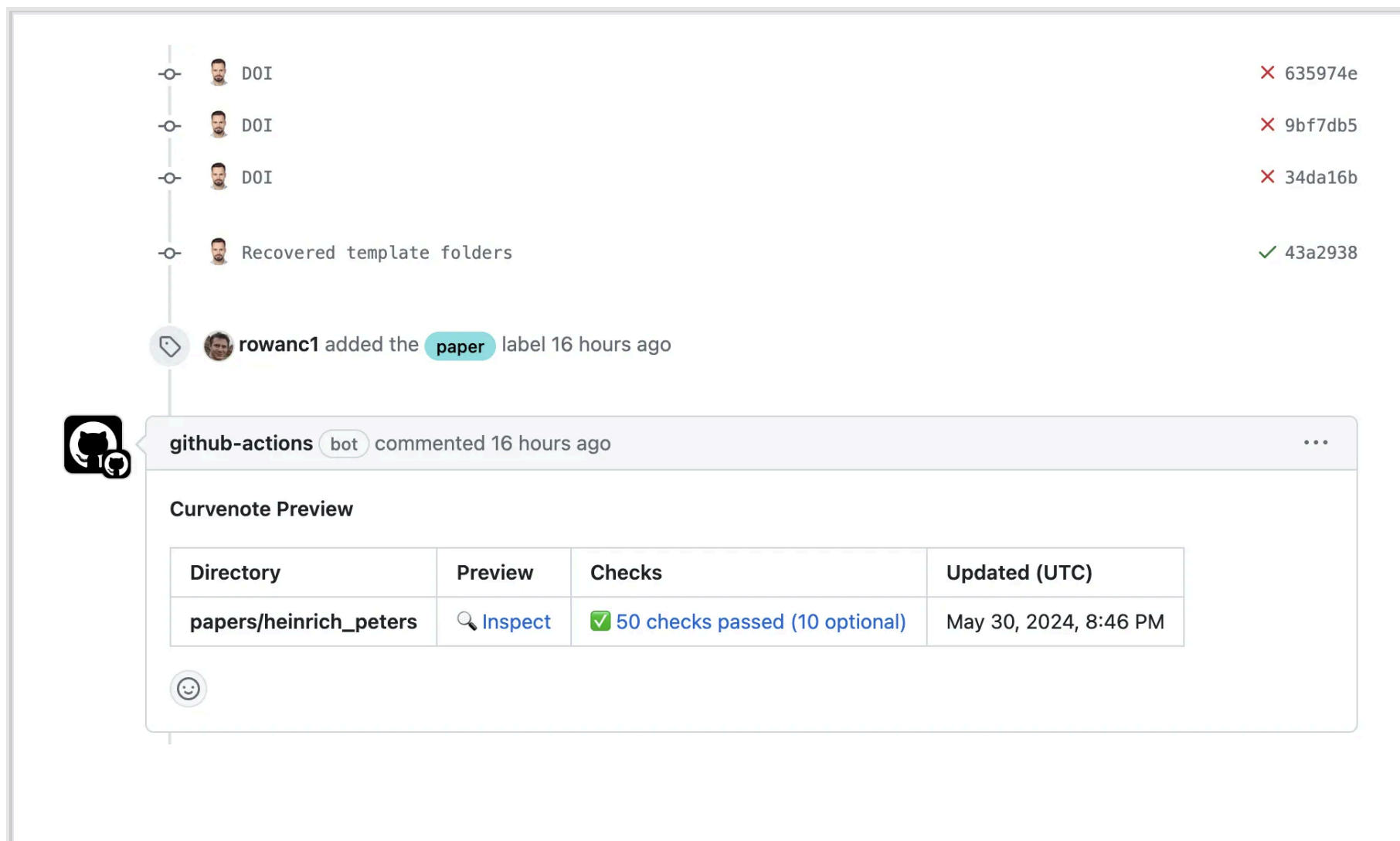
A terminal window with a dark background and light text. The window title is "...gs/papers/aaron_meurer". The terminal shows a git commit command: "base ~/git/journals/scipy/scipy_proceedings/papers/aaron_meurer git:(2023-myst)±21". A mouse cursor is pointing at the end of the command line.

```
base ~/git/journals/scipy/scipy_proceedings/papers/aaron_meurer git:(2023-myst)±21
```

Figure 5: Running `curvenote check` on a SciPy Proceedings article to check for missing DOIs, author ORCIDs, word-count and specific sections (e.g. abstract). These can be run in less than a few seconds both locally and through GitHub actions ([Figure 6](#)), which provides a user interface on the checks.

3.2 Automated Actions

Upon submission of there are a number of checks that are run and a submission or draft is deployed to Curvenote's platform, which stores the structured MyST project. We utilized GitHub Actions to automate initial checks and generate previews of submissions (via <https://github.com/curvenote/actions>). The actions automatically generate a preview of the manuscript exactly as it would appear in publication using MyST Markdown, and these are linked within the pull request comments for easy access by reviewers (Figure 6).



The screenshot shows a GitHub pull request interface. At the top, a commit history is visible with four entries: three by user 'DOI' (commit hashes 635974e, 9bf7db5, and 34da16b) and one by 'Recovered template folders' (commit hash 43a2938). Below the commit history, a comment from user 'rowanc1' states 'added the paper label 16 hours ago'. A comment from the 'github-actions bot' is highlighted, containing a 'Curvenote Preview' table. The table has four columns: 'Directory', 'Preview', 'Checks', and 'Updated (UTC)'. The data row shows the directory 'papers/heinrich_peters', a preview link 'Inspect', a status '50 checks passed (10 optional)', and the update time 'May 30, 2024, 8:46 PM'.

| Directory | Preview | Checks | Updated (UTC) |
|------------------------|-------------------------|----------------------------------|-----------------------|
| papers/heinrich_peters | Inspect | ✓ 50 checks passed (10 optional) | May 30, 2024, 8:46 PM |

Figure 6: An example of a comment by a GitHub action, which shows the checks and preview of the article directly. The checks in this example have promoted the author to improve metadata, see [scipy-conference/scipy_proceedings#911](https://github.com/scipy-conference/scipy_proceedings#911).

3.3 Enabling Researchers

Although SciPy submissions are made in a central repository, the Curvenote checks can be setup on any repository. As MyST Markdown bring a researcher's writing back into a simple text based format, and if they are working with Jupyter notebooks, integrates directly with their computational analysis and results, a researcher's manuscript can be easily created and maintained as an integral part of their research code base and updated in lock step with it. Imagine writing your next paper for submission to SciPy in a way where your draft was automatically rebuilt with the latest figures and data tables on every push to `main`, and any issues that you would hit on submission are flagged as they happen and so are more easily fixed, in context. This is just one of the ways that continuous science practices we spoke about in (Section 1.3) can lead to radically improved efficiencies at scale.

Curvenote's tools builds on this theme of enabling researchers to do more and to have more visibility into how their work will appear when submitted and published. The previews generated during the journal submission process, or by the actions on a researchers own repo can also be generated by an author at any time. By running `curvenote submit <journal> --draft` researchers can get a set of check results and a preview in the style of that venue, and where submissions of Computational Articles are permitted, authors and reviewers will even get feedback on reproducible environment check that any interactive figures and notebooks execute as expected.

3.4 Use Cases

The rapid feedback in authoring (Section 2) coupled with structural checks (Section 3.1), automation (Section 3.2), and archiving of the content opens workflows for varied sizes of teams to adopt these *continuous science* practices. Whether it is individuals publishing static MyST Markdown sites, lab-groups creating a better archive to highlight the research contributions of their team (e.g. [Applied Geophysics](#)), conference proceedings (e.g. [SciPy Proceedings](#)), or more formalized society journals (e.g. [Physiome](#), [American Geophysical Union](#), [Elemental Microscopy](#)) (Figure 7).

grammar is bit off. This is still part of the previous sentence (even if long)

Program Abstracts Sessions Posters Workshops

Agrogeo 24 Abstracts

Agriculture and geophysics. Illuminating the subsurface!

Agrogeophysics is an emerging research field. What are the current, potential and, future applications research needs? Come and discover with us!

[Book of Abstracts](#)

[Download book of abstracts in PDF format.](#)

31/01 19:00 Optional Icebreaker drinks: <https://www.atelierbar.ch/>

Day 1: Thursday February 1st, 2024

| | | |
|---|---|----------------|
| 8:45 | Welcome | A. Romero-Ruiz |
| 9:15-12:45 | Oral presentations | |
| Non-invasive assessment of root zone processes | | |
| Convener: G. Blachy | | |
| 9:15 | Monitoring and modelling root-zone processes with geoelectrical methods (invited) | B. Miry |
| 9:35 | High-throughput crop root phenotyping through electromagnetic induction method | D. Zhao |

Research Mission Open Science

Opportunities for open-source software to accelerate research in applied geophysics

Lindsey J. Heagy , Seog Kang , Joseph Caporini , Dominique Fournier , Steven Cocke , Douglas W. Oldenburg

December 18, 2023 - <https://doi.org/10.1002/ae4.43232084.1>

[Download PDF](#)

Abstract

This article considers the potential for open-source software and open-science practices to accelerate research in applied geophysics and thereby contribute to solutions of geoscientific problems impacting society. We provide context on the definition of open-source and give a brief history of open-source software in applied geophysics. Drawing from our experience in the SimPEG project, which develops software for simulation and inversion of geophysical data, we provide two examples where research was accelerated because of open-source software. These include the re-use of regularization methods for different geophysical problems (magnetics and time-domain electromagnetics), and the combination of multiple geophysical data types in joint inversions. We also provide an example where research code was re-purposed for education and humanitarian projects. Each of these examples was made possible because of the availability of code and the practices adopted by the community of collaborators involved in the project. We conclude with our perspective on how practices adopted by open-source communities that enable collaboration amongst researchers with different backgrounds, skills, and interests can be applied more broadly in research. This will ultimately increase the use and effectiveness of geophysics in helping solve applied problems.

Keywords: numerical, inversion, magnetics, electromagnetics, gravity

Introduction and Motivation

This special issue discusses several pressing applications where geophysics has an important role to play in non-
research involving the subsurface. These include problems such as locating natural resources, for example, to en-

IN THIS ARTICLE

- Introduction and Motivation
- Background: Open-source software
- Brief overview of open-source in applied geophysics
- Impacts of open-source software in applied geophysics
- Accelerated research, education, and outreach in the SimPEG community
- Regularization (variational and compact norms)
- Joint inversions
- Education and outreach
- Some takeaways
- Looking to the future
- Acknowledgment

Home Articles Archive News

Python in Science Conferences

THE PROCEEDINGS

The SciPy Conference is a cross-disciplinary gathering focused on the use and development of the Python language in scientific research. This event strives to bring together both users and developers of scientific tools, as well as academic research and state of the art industry.

Latest Articles

ARTICLE
Jul 10, 2023

Data Reduction Network

Multidimensional categorical data is widespread but not easily visualized using standard methods. For example, questionnaire data generally consists of questions with categorical responses. Popular methods of handling categorical data include one-hot encoding and enumeration, which applies an unwarranted and potentially misleading notional order to the data. To address this, we introduce a novel visualization method named Data Reduction Network.

Maoyin Xu, Hwan-min Lu, José Linggoco

Latest Research About Submit Editorial Board

Reproducible, reusable models for physiological research

Physica publishes mathematical models of physiological processes where the reproduction of details have been published or accepted for publication in a recognized primary peer-reviewed journal in the field of physiological modeling.

A Physica article provides a citable link between the published model and its implementation.

Latest Articles

ORIGINAL RESEARCH
Feb 12, 2024

Reproducibility Study on a PBPK Model of FcRn-Mediated Recycling for Large Molecules

The PBPK model of FcRn-mediated recycling of large molecules was developed and studied by deWitte et al. (2023) to characterize and predict immunoglobulin G (IgG) disposition in plasma and tissues. This study investigated the large molecule model in PK-Sim and its applicability to molecules with FcRn binding affinity in plasma.

Soroush Saïtes, Veronique De Brabandere, Wilhelmus E. A. de Witte, Lindsay B. Arntz, Tom Van Droogenbroeck, Géraldine Sargemont-Korbeel

ORIGINAL RESEARCH
Dec 6, 2023

Steady-state Approximations for Hodgkin-Huxley Cell Models: Towards Multi-scale Models of Uterine Smooth Muscle

An effort at reducing the Tong et al. (2011) model into a smaller number of equations for computational efficiency – the ‘reduced’ Tong model – aims at reproducing the overall behaviour without excessive detail.

Shawn A. Meers, Alisa M. Clark, Leo K. Cheng

ORIGINAL RESEARCH
May 21, 2023

Reproducibility study of the modular and reusable model of epithelial transport in the proximal convoluted tubule

Figure 7: MyST Markdown and Curvenote tools can help lower the barrier to entry for scientific publishing workflows for journals, research institutes, conferences, private consortiums, universities, and lab groups.

4 Conclusions

Scientific publishing infrastructure is disconnected from the day-to-day workflows of scientists, and at scale this slows the progress of science. These misalignments are particularly pronounced in computational disciplines, where rapid evolution of methodologies, software, and data demands equally dynamic and interconnected platforms. This gap — between the authoring/doing of research and the communicating/publishing of the research — slows the speed of research dissemination, reuse, and uptake and completely impedes “networked knowledge” and importing/reusing work in a structured way. For example, “importing” visualizations, equations or any other deeply-linked content – including provenance information – into new research articles, documentation or educational sites is completely impossible in today’s research ecosystem. As a metaphor, compare open-access science to open-source programming: it would be a world without package managers to share, version, reuse and rapidly build upon other peoples work in a structured way. The open-source ecosystem would not exist without this infrastructure. *punchline? -> these ideas help motivate MyST/Curvenote infrastructure*

I think this is the first time this idea is mentioned. I like this technology but would recommend introducing the term and idea earlier.

Open infrastructure for communicating science also has to be easy to integrate into existing tools, support computational, interactive components, be archivable for the long term, and be adopted by our existing sociotechnical system of societies, journals, and institutions. There are two interconnected problems that need to be solved: (1) upgrade existing scientific authoring tools, ensuring these are integrated into both scientific and data-science ecosystems; and (2) develop radically better ways to share content as individuals, small groups, preprints, and formalized, traditional journals with existing societies and institutions. The two problems are connected, in that the authoring tools should be able to deeply integrate with publishing mediums (e.g. referencing a figure from a publication should be able to show you that figure directly as you are authoring, including all interactivity and computation).

In this article we have presented some of the goals behind features of MyST Markdown and Curvenote, to support new open-science infrastructure including authoring tools as well as publishing workflows that support checks and automation. To support every next-generation research tool on top of open access knowledge, we need access to **high-quality, structured content**, data and software — not just the scholarly metadata and citation graph. *but also ...?* Our goal of contributing to MyST Markdown is to make the processes behind creating this structured data more accessible and affordable. These tools

in the hands of researchers can also enable process changes and **continuous science** practices: where checking and automation can support rapid iterations and feedback. The analogies between continuous delivery of software and continuous science give us an opportunity to peek ahead a decade to an analogous future and draw on many learnings on how to organize and focus infrastructure to get the best out of our scientific community.

There is, of course, an *enormity* of work ahead of these tools to transform science publishing at scale. We are grateful to our society partners who are changing community practices around publishing to support HTML-first publishing, experimenting with computational articles, and implementing new peer-review workflows. Tools on their own do not make change, but can help to enable it. Improvements to scientific publishing requires many diverse community efforts to improve the quality and speed of how we communicate knowledge, and ultimately to accelerate scientific progress.

Acknowledgments

Portions of this manuscript have been previously shared in the Curvenote and MyST Markdown documentation. The authors would like to thank Lindsey Heagy, Arfon Smith, Chris Holdgraf, Greg Caporaso, Jim Colliander, Fernando Perez, J.J. Allaire, and Kristen Ratan who have provided input on versions of these ideas over the last few years. Funding for portions of this work have come from the Sloan Foundation (for MyST Markdown and *Notebooks Now!*) and Alberta Innovates (for the initial version of the Curvenote CLI, which became `mystmd`). Thank you to the growing list of MyST Markdown contributors who continue to make MyST a fantastic community project.

References

Show All

1. Bourne, P. E., Clark, T. W., Dale, R., de Waard, A., Herman, I., Hovy, E. H., & Shotton, D. (2012). *Improving The Future of Research Communications and e-Scholarship (Dagstuhl Perspectives Workshop 11331)*. [10.4230/DAGMAN.1.1.41](#)
2. Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1). [10.1038/sdata.2016.18](#)
3. Grossmann, A., & Brembs, B. (2021). Current market rates for scholarly publishing services. *F1000Research*, 10, 20. [10.12688/f1000research.27468.2](#)
4. Hagve, M. (2020). Pengene bak vitenskapelig publisering. *Tidsskrift for Den Norske Legeforening*. [10.4045/tidsskr.20.0118](#)

5. Clotworthy, A., Davies, M., Cadman, T. J., Bengtsson, J., Andersen, T. O., Kadawathagedara, M., Vinther, J. L., Nguyen, T.-L., & Varga, T. V. (2023). Saving time and money in biomedical publishing: the case for free-format submissions with minimal requirements. *BMC Medicine*, 21(1). [10.1186/s12916-023-02882-y](https://doi.org/10.1186/s12916-023-02882-y).

Show all 26 references

↑
while the world adjusts, having these shown by
default would be appreciated. ↓