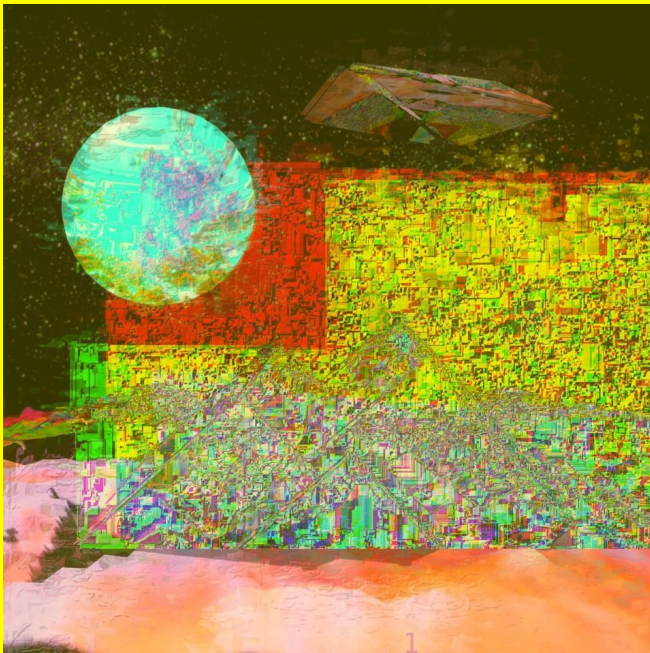


***the title is: (and
then a series of
checkboxes)
(insert here your
suggestions :})
+1+1+1+1+1**

- This is how we do it
- what shall we do it
- ctrl+c, ctrl+v, ctrl+c, ctrl+v
- computational breaking things
- other ...



INDEX

- [1. Introduction: Reflecting on This Mode of Working and Publishing](#)
- [2. Code Aesthetics Remix](#)
- [3. The Machinifestation of Collective Writing at ICLC2023: A Response to Martino Morandi's Constant Padology](#)
- [4. On the Jupyter Notebook](#)
- [5. Reflections on Platform Literacy and Collaborative Knowledge Production](#)
- [6. Expanded Publishing: Is Computational Publishing a Form of Expanded Publishing?](#)
- [7. Bleps are Bleps](#)
- [8. Model Legacy GPT 3.5](#)
- [9. A Need for Computational Writing and Publishing in the Humanities?](#)
- [10. Political Challenges and Questions of Responsibility in Collaborative Writing](#)
- [11. 500 Words Materialised](#)
- [12. Computational Writing and Publishing as Weaving](#)
- [13. Colophon](#)

Introduction: Reflecting on This Mode of Working and Publishing

Once upon a time, we decided to put together a few thoughts about collaborative writing and other stuff relating to padology. We don't want to make total sense. We want to show the unfinished business of writing together. As an example, we talk about: the machinifestation of collective writing at ICLC2023 ([#the-machinifestation-of-collective-writing-at-ICLC2023](#)); the pedagogical virtues of Jupiness notebooks ([#on-the-jupyter-notebook](#)); and others call their section "bambambam". What is the relation between these texts? I don't know. The point is, we utilize Octomode, a collaborative writing program, to actively explore the possibilities and limitations of collaborative programming and writing.

Part of that is practical. Are we actually comfortable with writing this together? Who says what, does what, and how? How do I change a font? Are you nostalgic about

Google Docs? Who are you? (???) Funnily, a lack of rules makes this process a little bit more loose than writing on a pretty UI-ed text editor. Even layouts are out for grabs; we can change them, even when we don't know how. We're messing with the basic parameters of writing.

Another part is conceptual. While it is an opportunity to lower entry barriers and allow for quicker, more efficient editing and publishing of academic texts, avoiding often restrictive structures of traditional academic distribution of texts, it brings along practical obstacles and general questions of ethics and values. Collaborative writing creates opportunities to share knowledge as well as writing or editing habits while simultaneously creating tensions revealing the delicacy of the personal integrity and dignity that comes with exposing personal writing to others. Furthermore, it problematises the concepts of authorship, copyright, temporality, and economical reputation more generally. In the current academic environment that is still largely structured through institutions, global disproportion of allocating resources to the West, and monetisation, collaborative writing and publishing pose opportunities to question fundamental structures and amplify voices that are generally paid attention to less.

Take that, big publishing companies!

Booooooh 🗨️ mic drop 🤘👋🎤

Code Aesthetics Remix

aka collage writing

Creative expression

Computational writing and publishing offer new opportunities for creative expression. Approaching the writing of a book as Soon & Cox do, as an iterative process “in the spirit of how software developers collaborate, host, review, and merge `code`, as well as build software together” (2020, p.18) opens up knowledge production and sharing to **unlimited potentialities!**. By advocating for a deeper understanding of the relationship between `code` and aesthetics, examining the interplay between technology, creativity, and design, Soon and Cox provide valuable insight into how computational writing can be harnessed as a powerful artistic medium.

New ways of publishing/writing or TAKE THAT BIG PUBLISHING COMPANIES!!!

Computational writing and publishing enables us to

question the assumptions and limitations of traditional forms of publishing. It allows us to explore new forms of writing. It is a type of weaving, because it consists of assemblages of different thoughts. Cuts that have been made through the reading or scanning of texts are re-woven in a collage. These are “imaginaries of **code**”: `ctrl+c`, `ctrl+v`, `ctrl+c`, `ctrl+v`, ... It’s not just a way of writing, but a way of thinking: a *way of weaving*. It is multi-directional, in hyperlinks manner, rather than linear and towards one specific end-goal.

Politiczzz

Computational writing and publishing can be a powerful tool for critical reflection. In order to adequately confront the ways in which platforms and algorithms shape our day to day lives the question of programming as literacy becomes crucial. While considering *platform literacy*, the importance of understanding this mode of communication or “way of thinking and doing in the world” (p.14) is related to our unavoidable relationship with the platforms themselves. There is a tension however, between seeing programming and computational language as a natural and essential part of everyday life; to the point of considering it as its own literacy (Vee 2017, Soon and Cox 2020); and **code** most often being hidden from many conversations about this very everyday life – leaving it with specialists to deal with.

The Machinifestation of Collective Writing at ICLC2023

A Response to Martino Morandi's *Constant Padology*

by Olga EFREMOV Utrecht University | RMA MAPS

Martino Morandi's *Constant Padology* published by MARCH in January 2023 is a celebration of Etherpad as an essential component of the Constant's tool-ecology, a simple piece of open-source software that has been integrated into organizational workflow and permeated most activities and practices. Constant Etherpad is a vivid example of so-called "grassroot organizational practices" that sensible and forward-looking organizational leaders should strive to be constantly aware of and use to the maximum benefit of the organization. A collaborative document creation where contributions are anonymized can be a powerful tool of integrating a variety of voices into a single text-space where the shared authorship is not by design attributable to individual contributors. Morandi notes that such practices are becoming an increasingly common way to create common voice experiences and collective identities expressed through the practice of Etherpad-enabled collaborative writing.

Only yesterday I had a privilege of witnessing the creation of *The 2023 Live Coding Machinifesto* during a workshop session that took place as a part of the International Conference of Live Coding in Utrecht. While I was not able to be physically present at the location, a link to disroot.org pad was provided by the organizers for anyone willing to contribute or to observe. It is noteworthy how this writing practice organically reflected the communal aspirations of the group that put “Community is key” as an opening statement of their machinifesto. Even the term “machinifesto” as a collaborative creation, as the title’s colors attest (Figure 1).

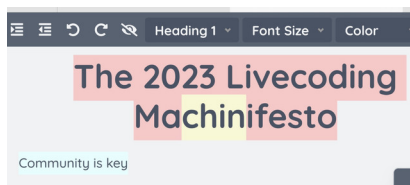
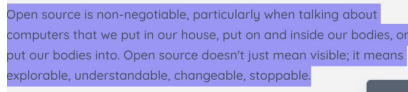


Figure 1. The opening statement of The 2023 Livecoding Machinifesto on pad.disroot.org.

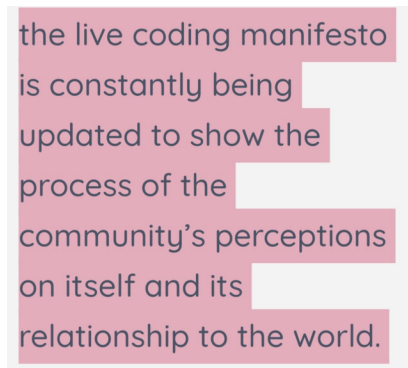
Livecoding community which thrives on openness and nonhierarchical collaboration cites their commitment to open-source code and solutions that are “explorable, understandable, changeable, stoppable” (Figure 2). Collaborative writing of the machinifesto is a prime example of a open-source practice which embeds all these qualities. The creators stress the fact that the

machinifesto is a live document, constantly updated “to show the process of the community’s perception on itself and its relationship to the world” (Figure 3). In that respect, one could trace the parallels between live coders’ collaborative writing practice and constant organisational practices described by Morandi: “Connections arise through the collective practice of writing and through the distributed running of the infrastructure that supports it”. The collective document will stay open to the input even after the end of ICLC2023 event, as an artefact of the connections that have been forged during the conference.

A screenshot of a text document with a light blue background. The text is black and reads: "Open source is non-negotiable, particularly when talking about computers that we put in our house, put on and inside our bodies, or put our bodies into. Open source doesn't just mean visible; it means explorable, understandable, changeable, stoppable." The text is partially obscured by a dark grey bar at the bottom right.

Open source is non-negotiable, particularly when talking about computers that we put in our house, put on and inside our bodies, or put our bodies into. Open source doesn't just mean visible; it means explorable, understandable, changeable, stoppable.

Figure 2. Livecoders’ statement on open source.

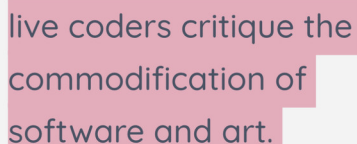
A screenshot of a text document with a light grey background. The text is dark blue and reads: "the live coding manifesto is constantly being updated to show the process of the community’s perceptions on itself and its relationship to the world." The text is partially obscured by a dark grey bar at the bottom right.

the live coding manifesto is constantly being updated to show the process of the community’s perceptions on itself and its relationship to the world.

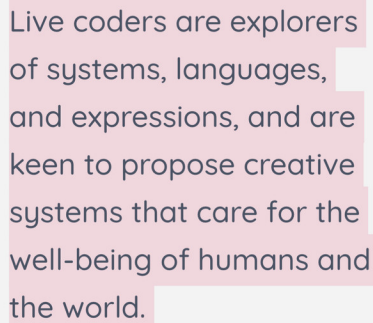
Figure 3. The live coding manifesto as “constant padding”.

Another remarkable parallel is “open source padding” as

a written manifestation of the community’s critical attitudes towards the commodification of software and art (Figure 4). Morandi notes that Etherpad “is rooted in the proposal of a different type of usership in contrast with the one produced on massive corporate platforms”, and thus creates a sense of usersafety and control that is owned by the community. The 2023 Livecoding Machinifesto pad space has turned a safe place for the groups self-reflection and self-exploration, an environment for fostering a collective identity in the same way as live coders explore “systems, languages, and expressions” (Figure 4). In this short document the live coders exemplify their thinking-making practices that are supported by their commitment to open source code and their critique of the coporate ownership of the software.



live coders critique the
commodification of
software and art.



Live coders are explorers of systems, languages, and expressions, and are keen to propose creative systems that care for the well-being of humans and the world.

Figures 4 and 5. Excerpts from the machinifesto.

Reflections on Platform Literacy and Collaborative Knowledge Production

In order to adequately confront the ways in which platforms and algorithms shape our day to day lives the question of programming as literacy becomes crucial. While considering *platform literacy*, the importance of understanding this mode of communication or “way of thinking and doing in the world” (Soon & Cox, 2020, p.14) is related to our unavoidable relationship with the platforms themselves. As programming is considered infrastructural (Vee, 2017), being cognizant of it allows us to confront and be critical of the power behind the knowledge of the existing infrastructure that we interact with daily. As someone who would be considered platform

illiterate, **I often do find myself taking the ignorance is bliss approach when it comes to programming.**

While I am aware of how deeply ingrained this “world-making” is in my own world, the skill has somehow always felt out of my reach. I can see how valuable this form of literacy would be especially when considering programming as a mode of written communication. Those who do not have a natural or self-motivated interest to learn the skill rarely have a chance to become literate throughout their schooling. This seems to lead to a lot of people being left out of the conversation. These perspectives, or lack thereof, can shape our understanding of the cultural production of programming. Considering programming as a literacy allows us to prioritize the teaching and learning of this skill to encourage more voices in the conversations that confront the authority of the infrastructure that governs our digital lives.

Approaching the writing of a book as Soon & Cox do, as an iterative process “in the spirit of how software developers collaborate, host, review, and merge **code**, as well as build software together” (p.18) opens up knowledge production and sharing to unlimited potentialities. Janneke Adema (2021) emphasizes the dynamic materiality of books and in opening them to alterity, the repetitive and conservative aspects of books that privileges the author and book-object over knowledge production is challenged. In this way

knowledge sharing becomes collaborative and dynamic conversations can take place.



My interest in content moderation and platform governance intersects with these practices as highlighted by Martino Morandi (2023) when he shared his experience with being responsible for Etherpad installations. Even though the threshold of entrance is low and there is a concern around anonymity creating a vulnerability to malicious participation or disruption, he witnessed that instances of defacement were extremely low and rarely intentional. This lack of maliciousness within a digital public is not something considered typical of anonymous users. His concept of trust and responsibility in the collective in publically accesible Pads is really inspiring when considering safety and trust in members of a community relating to each other online.

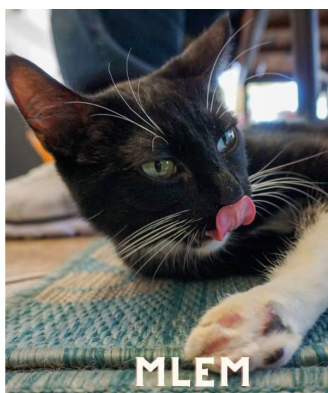
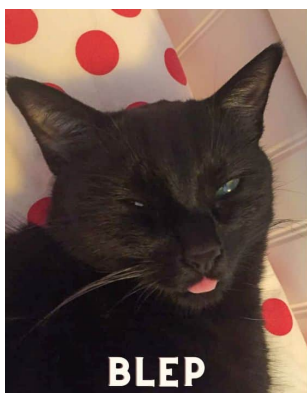
II: Casual studies

Bleps are Bleps

In academia, there appears to be an awkward gap between knowing about the networked nature of ideas and writing, and abiding by a (now) centuries-old practice of publishing still text edited once in every few years. How does one bridge these two writing cultures? Should they be bridged, the scholar's "idea" would be seen as somewhat less original, less unique to an individual genius, because networked writing would transpire the collaborative and somewhat banal aspects of thought — everyone thinks more or less the same thing, at more or less the same time. It would also afford constant updating, in accordance with the constant flow of information that is facilitated by networked environments. Then, textbooks could be wikis, and analytical writing a wiki of networked ideas. So why not make scholarly text more like — modular, building blocks of collective thought? There seems to be an institutional roadblock, with perhaps good enough justifications as to why writing cannot quite change as radically as coding proposes. What are these objections?

At the same time, there appears to be another institutional roadblock in the world of coding: one needs

to be technically literate (in a variety of ways) to participate in a transformation of literacy. One needs to survive massive job market shifts. One needs to be able to critique and intervene in Big Tech software regimenting service economies. Unwillingly, these expectations echo the literacy divide of long ago, when those who read and wrote (in Latin) were few religious and intellectual elites that did not expect any better from lower classes, even when they were humanists. How strange that, today, these classes are not necessarily lower, but part of another institutional setting.



but a blep can also be a mlem

Expanded Publishing: Is Computational Publishing a Form of Expanded Publishing?

Expanded Publishing (EP) in its broader sense relates to going beyond typical publishing, in terms of format and technology. In some similar contexts, self-reflective, extended, hybrid, and urgent publishing have a different name, but they, at least partly, relate to the same point: a publishing practice that is more inclusive than traditional publishing in that it enables broader means of expression, includes more features and inter-format connections, involves readership or peers, open-sources the underlying technology for better access, integrates the metadata of a publication, experiments with non-publishable art or scientific formats (e.g. an exhibition, a conference) etc. In short, it goes beyond (= expands) the traditional.

A tendency in recent media theory has been to qualify or suspend the language of rupture or discontinuity in discussing the relations between “old” media and “new” digital technologies. Instead, older models and arrangements are understood to persist in various forms of hybridity, convergence, remediation or recuperation. At the same time, the old has to compete with the new to remain attractive: for this reason, print is becoming hypermediated, as it incorporates verbal genres and gestures in a self-conscious imitation of and rivalry with electronic media.

EP is a literal expansion of three pre-existing types of publishing: multimedia, hybrid and urgent publishing.

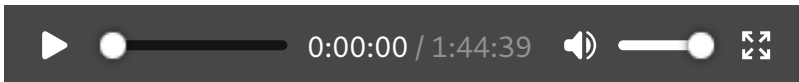
- **Multimedia** has its roots in the 1990s and the early internet culture. It is commonly defined as “communication that uses a combination of different content forms such as text, audio, images, animations, or video into a single interactive presentation, in contrast to traditional mass media, such as printed material or audio recordings, which features little to no interaction between users”. (Wikipedia) In short, multimedia is a synthesis of different types of media formats brought together in a single presentation or on a single page.
- **Hybrid publishing** emerged after the rise of smart devices, such as smartphones and tablets, where the content is published in different medium-specific formats that highlight the distinction between physical and virtual content. Examples are physical books, PDFs, ePUBs, digital long-form essays, websites etc.
- In **urgent publishing**, which emerged over the past years, the notion of time was added to the previous two types of publishing mentioned above. In times of crises and information overload, the ability to make knowledge public at a specific (often urgent) moment in time becomes pressing - aside from formats and quality of the content, the time component of actual publishing becomes hugely important. New strategies such as riso printing and

xerox copy machines grow in importance.
Decentralised approaches are an essential aspect of
urgent publishing, especially in its political
implications.

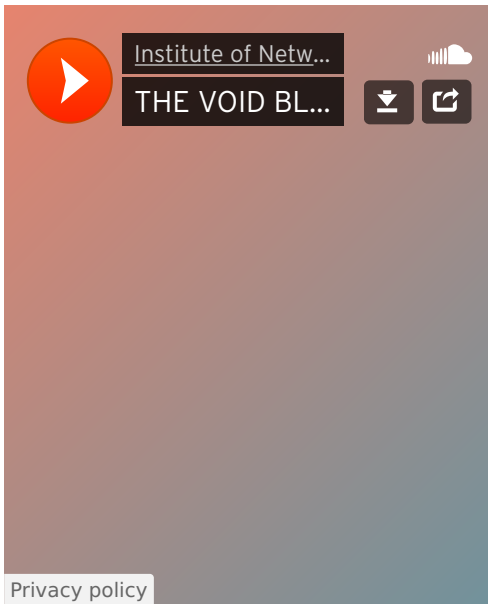
These practices are rooted in independent art, research
and science publishing, artistic research, DIY and
experimental practices, perpetrated by underground and
technology activists and various small publishers who
need to experiment and innovate in order to remain at
the forefront and avoid being overshadowed.

Further multimedial references

**Pagedjs workshop with Julie Blanc and Julien
Tacquet @ XPUB, 13 October 2022**



**THE VOID BLOG | Looking at Expanded Audio-Visual
Publishing for Open Collaborations**



I like this workshop

A Need for Computational Writing and Publishing in the Humanities?

My personal need for computational research as a Humanities student draws from the fact that I am working on a *conjunctural analysis* of the current state in Dutch culture after the power display of the *BoerBurgerBeweging* during this year's elections, when the *BBB* became the biggest party in all of Holland's twelve provinces. A conjuncture is defined by Stuart Hall

as “a period during which the different social, political, economic and ideological contradictions that are at work in society come together to give it a specific and distinctive shape” (Hall, qtd. in Clarke 80). Conjunctural analysis, therefore, stresses “multiplicity, heterogeneity and the condensed dynamics of over-determination” (Clarke 82). Scholars widely agree on the fact that conjunctural analysis is “hard work” (84). The set of questions and possible answers are endless, and thus conjunctural analysis [...] “is not something that should be undertaken alone” (84).

The possibilities that computational writing and publishing offer are vast. Firstly, these digital technologies go beyond the individual author and “offer opportunities to highlight the communal and processual nature of research” and therefore seamlessly fits the mode of communal engagement called for by conjunctural research (Adema). Secondly, the primary goal of conjunctural analysis is to describe the present (and offer alternatives), but if this present is ever-changing, how can we then stay up to date through our conventional methods of publishing? Before a book or an essay is published, a conjunctural shift might already have taken place. And thirdly, the references that point to these different contradictions are not always conventional published books, but often contain newspaper articles, TV- and radioshows, podcasts, blogs, etc., sources that are often stored digitally. These

references scream for a different bibliography that goes beyond the last page of a printed book (with a printed out url...).

Political Challenges and Questions of Responsibility in Collaborative Writing

In broader media studies, the processes and makings behind media and platforms are sometimes neglected, including programming and the general infrastructures of writing. There is a tension between taking programming and computational language as natural and essential of everyday life to the point of considering it as its own literacy (Vee 2017, Soon and Cox 2020), while it is still being hidden from many conversations about this very everyday life and left with specialists to deal with. Vee puts this divide into perspective, arguing that “programming is so infrastructural to everything we say and do now, leaving it to computer science is like leaving writing to English or other language departments.” (Vee 2017, 7)

Making programming more accessible through increasing literacy outside of specialists also opens doors to collaborative work with subversive potential (Morandi 2023, Adema 2022). Collaborative programming, writing, and publishing, in combination with greater literacy rates,

lowers entry barriers and works against elitism and exclusivity within official structures, but also in perceiving reality. This notion introduced by Soon and Cox, of programming as “a means of understanding some of the complex procedures that underwrite and constitute our lived realities, in order to act upon those realities” (Soon and Cox 2020, Preface), is already visible in other media practices, but must always be considered in a wider social context. Here, I want to complicate the concept of collaborative writing as a subversive practice. While collaboration and greater access of technically linguistic knowledge to the public has many benefits and can lead to beautiful projects, it can also carry a risk of further reiteration of elitism and the establishment of politically questionable collectives. As an example of this collaborative effort, I would like to mention projects like [Archive of Our Own](#), which provides a platform of creative writing that ensures the function of a permanent, openly accessible library. Other projects, as mentioned by Adema (2022) which undergo official publishing channels, allow for peer reviewing of academic texts and open accessibility. On the other hand, these collectives are also at times problematic. Thinking about the amateurishly crafted platform 4CHAN that led to the establishment of QAnon and Anonymous, and in further projection to the formation and solidification of the alt-right under Trump, these collectives can breed harmful rhetoric that may reflect in global or local political climates. If collaborative writing and programming relies on peer reviewing, it is

important to ask who this peer review is accessible to, and how it is conducted.

Another question to ask might be that of authorship, and in relation, that of responsibility. While authorship, or literal ownership in context of copyright, may be a reductive notion as no idea or work has ever come into existence by one person alone, a lack of overview of participants in a work may either result in an erasure of acknowledging efforts of certain people or groups, or blur the origins of works that may hold individuals accountable. In the context of the development of AI systems, responsibility and attribution pose major questions. In this case, the infrastructure is in a sense almost more present than the writers, as some sort of autonomy is attributed to the system. In this line of reasoning, it is the AI system that has produced texts with wrong information, not the writers of the system. How could the writers instead be held accountable? And is this question similar if what is written is plain text instead of code? As programming is central to societal infrastructure, it is important to discuss the possibilities of collaborative work, question structures in place, and search for alternative frameworks.

Anything With Nothing Now

Model
Legacy (GPT-3.5) ✓

Default (GPT-3.5)


Legacy (GPT-3.5) ✓


GPT-4


GPT-4

Our most advanced model, available to Plus subscribers.

GPT-4 excels at tasks that require advanced reasoning, complex instruction understanding, and more creativity.

Reasoning 

Speed 

Conciseness 

ChatGPT **PLUS**

You are asked to write a short text (within 500 words) reflecting on computational writing and publishing, drawing upon the references below, and based on your own research interests. Feel free to include images in your text. This will provide material for the workshop to share and rework.

Reading list

Martino Morandi, "Constant Padology," MARCH (2023). <https://march.international/constant-padology/>

Winnie Soon & Geoff Cox, "Preface" + "Design notes", in Aesthetic Programming (Open Humanities

ChatGPT Mar 23 Version. ChatGPT may produce inaccurate information about people, places, or facts.

Model legacy GPT-3.5

Computational writing and publishing have become increasingly important in recent years as technology continues to shape the way we communicate and access information. The traditional model of publishing, which relies on the printed codex, is being challenged by new forms of writing and publishing that are enabled by computational technologies. In this reflection, I will draw upon the references provided to explore the implications of this shift and the opportunities it presents.

In „Constant Padology,” Martino Morandi argues that the rise of digital devices and online platforms has led to a constant state of writing and publishing, in which the distinction between authors and readers has become blurred. This constant state of writing and publishing is facilitated by the ease with which digital devices and online platforms allow us to create and share information. The implications of this shift are significant, as it means that the traditional model of publishing, which relies on a small number of gatekeepers to determine what gets published, is being disrupted.

Winnie Soon and Geoff Cox, in their „Preface” to *Aesthetic Programming*, argue that computational writing and publishing offer new opportunities for creativity and expression. They suggest that computational writing and publishing allow us to explore new forms of writing that are not possible with traditional forms of publishing. They also note that computational writing and publishing can be a powerful tool for critical reflection, as it enables us to question the assumptions and limitations of traditional forms of publishing.

Annette Vee, in „Introduction: Computer Programming as Literacy,” argues that computational writing and publishing should be considered a form of literacy. She notes that coding is a form of communication that enables us to express our ideas and create new forms of knowledge. She also argues that computational writing and publishing should be considered a form of literacy because it requires a certain level of competency in order to participate effectively in the digital world.

Janneke Adema, in „The Processual Book,” argues that the traditional model of publishing, which is based on the printed codex, is becoming increasingly outdated. She notes that the rise of computational writing and publishing has led to a shift away from the printed codex and towards processual books, which are dynamic and interactive forms of writing and publishing. Adema argues that this shift presents new opportunities for writing and publishing, as processual books can be updated and revised in real-time, providing a more fluid and dynamic form of communication.

In conclusion, computational writing and publishing are disrupting the traditional model of publishing and presenting new opportunities for creativity, expression, and critical reflection. As these technologies continue to evolve, they will likely play an increasingly important role in shaping the way we communicate and access information. It is important for us to understand the implications of this shift and to explore the opportunities that computational writing and publishing present.

Model GPT-4

The rapid evolution of digital technologies has had a transformative impact on writing and publishing practices. This short text will reflect on the significance of computational writing and publishing, drawing upon the provided readings and my research interests in the intersection of technology, language, and collaboration.

Martino Morandi's „Constant Padology“ explores the creative possibilities of computational writing through the use of pads. Morandi emphasises the fluidity and flexibility of digital writing, where content can be added, modified, or erased with ease. This dynamic nature allows for the continuous development and adaptation of the written text, fostering a sense of processuality that traditional writing practices lack.

Similarly, Winnie Soon and Geoff Cox's „Aesthetic Programming“ addresses the artistic aspects of computational writing, advocating for a deeper understanding of the relationship between code and aesthetics. By examining the interplay between technology, creativity, and design, Soon and Cox provide valuable insight into how computational writing can be harnessed as a powerful artistic medium.

Annette Vee's „Coding Literacy“ further expands on the idea of computational writing as a form of literacy. Vee argues that computer programming shares similarities with traditional literacy skills, such as reading and writing, and should thus be considered a valuable component of contemporary education. By recognizing the cultural and social implications of computational writing, Vee's work highlights the importance of nurturing coding literacy in an increasingly digital world.

In her article, „The Processual Book,“ Janneke Adema proposes moving beyond the printed codex to embrace the potentials of digital publishing. Adema envisions a future where books become living, evolving entities, where content is continuously updated and expanded by multiple authors. This vision aligns with the concept of „Living Books,“ a project that reimagines the traditional book format by incorporating dynamic, interactive, and collaborative features.

Drawing upon these readings, it becomes clear that computational writing and publishing practices can revolutionize the way we perceive and engage with written text and can transform academic scholarship and challenge the codex of academic writing. By embracing processuality and collaboration, these practices challenge traditional notions of authorship and authority, allowing for the creation of more inclusive, diverse, and dynamic knowledge and can foster more generous approaches to writing and reading.

In conclusion, computational writing and publishing practices have the potential to radically transform our understanding of and engagement with written text. By fostering processuality, collaboration, and diversity, these practices challenge traditional notions of authorship and authority, opening up new possibilities for creativity, new geometries of thinking doing and knowledge production, as well as dissemination.



Dall-E generated image with the prompt “a group of people writing together lines of words and lines of digits on a big computer screen in cyberpunk style”

500 Words Materialised

These 500 words were written using a second hand laptop, Linux, Vim, Pandoc, Latex, Zotero, a pdf reader, a server and git. In order to write and publish in a way that connects to my research, I try to align my tools with the topic I write about, which at this moment is the materiality of the networked image and its environmental impact. Translating this to text and writing, I need to first think through what this materiality consists of.

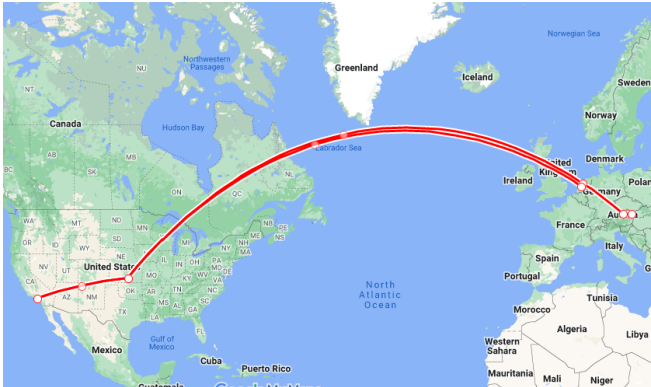
A shopping list of material needed for these 500 words to exist on my laptop and backup server:

- * The hardware needed to produce the text[^{emb}].
- * The demands the software I'm using to write this on that hardware in terms of:
 - * processing power (1.3% CPU on average) and
 - * electricity consumption (the more intense on the CPU, the more electricity is used).
- * The demands of the text on that hardware in terms of file size, i.e. the size or weight of the file (8kb) and the hardware needed to accommodate it.
- * The energy it takes to transmit the file over the Internet to my git server.
- * The infrastructure that circulates the text, meaning the entire network infrastructure from the Asymmetric Digital Subscriber Line (ADSL) making use of phonelines in my street, to the fiberoptic cables, network switches, internet exchange points and depending on the hops, undersea cables, making up the Internet (see fig. 1 for a misguided mapping).
- * The hardware needed to store the text (my server) and the energy it takes for that hardware to make the text readable when needed (servers are usually always on).

The materiality of a networked publication is an

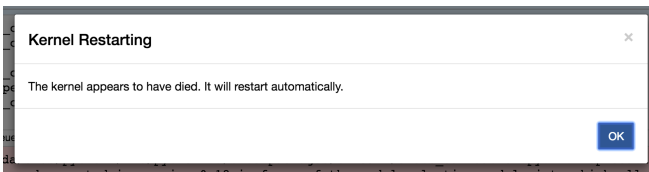
interconnected web of dependencies that all cost energy and resources. Trying to computationally write and publish something while taking this materiality into account becomes a matter of choosing your battles, because we are all positioned on the edge of this enormous infrastructure we have only very limited control over. From the edge of this [behemoth](#) we can still make choices though, as Martino Morandi remarks in *Constant Padology*; by running your own server and services you have the privilege of choosing *some* of your own dependencies (Morandi, 2023).

The choices that can be made perform on different levels. The materiality of networked texts connects to their social context. When do services need to be available and when can they be turned off? How can collaboration and organisation take place in a lightweight way? Materiality also connects to the economic dimension of networked communication, acknowledging the social and environmental cost of tracking people online and the business models underpinning the infrastructures used, which then connects to the political question of how to deal with the dependencies that do damage, are not chosen and require collective action rather than individual choice.



nmap traceroute with geolocation from my laptop in the Netherlands to my server in Austria. The hops to the US are incorrect and a result of nmap geolocation using whois data to determine the route, rather than IP lookup, showing the location of the owner (in this case Hurricane Electric LLC from San Jose, California). A regular traceroute shows the traffic goes through Germany.

On the Jupyter Notebook



A poetic error message showing in the Jupyter Notebook interface.

Ever since the first time I got to use a Jupyter Notebook, the format has interested me. The Notebook originally started as an interactive computing tool for scientists, to

quickly run analyses on their datasets. The linear structure consisting of vertically arranged `code` and plain text blocks was added later, allowing for what the Jupyter development team calls a *computational narrative*^{*}. The format is designed for taking yourself (or someone else) through the thinking steps of a data analysis. `Code` can be run in place such that results and data visualizations become a dynamic part of the story. `Code` blocks alternate with text blocks to argue for choices and reflect on results in the analysis. The Notebook proves at least as useful for personal practices, as for educational purposes. Tutorials written in Notebook files (especially when hosted in an online environment) enable students to have a quick start at playing around with `code`, without having to go through buggy setup procedures. By now, whole courses are taught in Notebooks, as well as textbooks are written in it.

The Jupyter Notebook has shown its use for *writing* and *teaching* `code`, but somehow the possibilities for *publishing* `code` are underexposed. While journals are put out in print less and less and essentially have already been consolidated in the digital sphere, scientists and publishing houses hold on to the trusty old PDF for the communication of all academic works, even though many of these works rely heavily on computational analyses. If we are lucky, some footnote floating around in the PDF refers to a github repository containing the analysis, and if we are very lucky, the link is not broken. After the

invention of the printing press, it took about a hundred years to realize that there was no good reason to stick to the illegible scribbling of typefaces that emulated traditional monks' handwriting, and the first serif typefaces were developed. The invention of the computer gave us as much opportunity to *write*`code` as to *publish* it, so why would we stick to an emulation of the A4 text on paper format that can only be used for *describing* datasets, analyses and results? Sharing these directly improves transparency and reproducibility, two highly valued aspects in scientific research. Show, don't tell.

Suppose the Jupyter Notebook or a similar format would become a standard for publishing computation, would that bring about major changes? Would this, following Annete Vee's introduction on `code` literacy, support a widespread increase in this literacy, or would it merely answer needs for a literacy that is already present? Would it compete with other literacies, threatening to replace them, or merely complement other forms of communication, including the good old (handwritten) text on paper? As project Jupyter evokes these and similar questions, I think the project is much worth discussing in the context of computational writing and publishing.

Computational Writing and Publishing as Weaving

Computational writing highlights how producing and publishing texts has always been about *weaving*. Weaving links between humans, weaving links between writing fragments, weaving links between different spirits and different technics. Writing is connecting, and new forms of writing now need an internet connection and at least once a day, connection of our writing machines to plugs. Digital writing is networked writing, writing for networks.

As with weaving, the process is never fully terminated and needs to be remade, each time a whole appears.

**There are always holes that need to be weaved.
Intellectual, emotional, relational, political holes.**

Yet computational writing is a very peculiar form of weaving: it is made, through electronic simultaneity, *collective*. A negotiation in different colors on an etherpad. What is this negotiation? It is a negotiation on a project – that is, on a collective projectuality, necessary for any form of collective at all. Who is the owner of the text? No one, everyone: the group. Authority and trust are reconfigured through a “computational mentality”: from the representability of the author over the delegating crowd, to the participation of the individual in the collective and of the collective in the individual. The vertical delegation of the past is transformed into diversifying diagonals.

Literacy is also about political organization. The author, proclaimed dead by [Foucault](#) and Barthes in the late 60s, was then truly brought to his grave with the advent of the world wide web 30 years later. With this funeral came the one of old political forms: representative democracy, experts driving societal decisions, ... We need to understand literacy in this way: as a political organization and paradigm in itself, that has shifted with computational writing. One particular parameter in which we observe more acutely this shift is in the *distribution* aspect of texts: that is, in *informational power*.

This is what the “Constant Padology” article also means by “another topology of trust and responsibility”. Not the old white male author-ity, but the distributed one. Allowing for revisioning means trusting, in counter-current to technocracy and patriarchal paternalism.

Computational writing is weaving, because it consists of assemblages of different thoughts. Cuts that have been made through the reading or scanning of texts are re-woven in a collage writing. This is for me the “imaginaries of **code**” that Winnie Soon and Geoff Cox talk about: ctrl+c, ctrl+v, ctrl+c, ctrl+v, ... It’s not just a way of writing, but a way of thinking: a *way of weaving*. It is multi-directional, in hyperlinks manner, rather than linear and towards one specific end-goal.

One way to transform the techno-social is through the

social-text: in which the conversations in the margins are displayed and valorized as an integral part of the text. A transformation of the text in its “inside-out”, through which Walter Benjamin’s reader is more than ever ready to become the writer – from a passive literary subject to an active computational citizen.

The following words have been revealed in the original texts contributing to the present publication:

computational, computation, me, you, I, we, us, when, where, there, here, open, access, process, publishing, temporality, collective, questioning, reading, writing, write, writes, critique, author, authorship, collaborative, collaboration, alternative, production, again, community, communities, dependency, relation, relations

writing publishing
access
writing publishing I computational we publishing

writing publishing
writing publishing us publishing

computational writing publishing
computational writing publishing is publishing writing
computational writing publishing us publishing
computational writing publishing
publishing us computational writing

computational writing publishing
computational writing publishing writing publishing publishing

computational writing publishing
publishing us we access
writing publishing computational

writing publishing computational writing publishing
collaboration
computational writing writing
writing
computational writing
computational writing
computational writing
reading writing computational writing

publishing where where
where
collaborative
writing publishing computational we
collaboration writing
authorship writing reading
computational writing publishing collaboration
authorship production

I
computational writing publishing
when
there
I
when

References

- Martino Morandi, “Constant Padology.” MARCH (2023). <https://march.international/constant-padology/>
- Winnie Soon & Geoff Cox, “Preface” + “Design notes”, in *Aesthetic Programming* (Open Humanities Press, 2020), pp.13-25. <https://aesthetic-programming.net/>
- Annette Vee, “Introduction: Computer Programming as Literacy”, in *Coding Literacy* (The MIT Press, 2017), pp. 1-42. <https://library.memoryoftheworld.org/#/book/ec32fdcc-a48c-444c-837d-f793dd8de667>
- Janneke Adema, “The Processual Book. How Can We Move Beyond the Printed Codex?” (2022), <https://blogs.lse.ac.uk/impactofsocialsciences/2022/01/10/the-processual-book-how-can-we-move-beyond-the-printed-codex/> (Further reading: *Living Books*,

<https://livingbooks.mitpress.mit.edu/pub/35x6zq2o/release/1>

- John Clarke, “Doing the Dirty Work: Challenges of Conjunctural Analysis,” in Stuart Hall - Conversations, Projects and Legacies (Goldsmiths Press, 2017).

Code Commentary

Voices from the octomode repository

<https://git.vvvvvvvaria.org/varia/octomode>

from octomode.py

```
# To sanitize Flask input fields
# To sanitize Markdown input
# To read the Markdown metadat
# ---
# create pad in case it does not yet exist
# Select default template
# Create pad and add the default template
# Convert Markdown to HTML
# Sanitize the Markdown
# Another built-in Flask way to sanitize
# Read the metadata from the Markdown
# ---
# The pads are filled with the default templates
(pad, stylesheet, template)
```

```
# This is when the environment is "created"
# only here we need application root to make all
the URLs work.....
# only here we need application root to make all
the URLs work.....
# //////////////////////////////////
# RENDERED RESOURCES
# //////////////////////////////////
# (These are not saved as a file on the server)
# Insert CSS sanitizer here.
# TO GENERATE THE PREVIEW WEBPAGE
# TO GENERATE THE PAGED.JS WEBPAGE
# //////////////////////////////////
```

```
from settings.py
```

```
# Load environment variables from the .env file
# Bind them to Python variables
# Check if API key is provided
```

```
from templates/base.html
```

```
// Insert the nav buttons, after the page is
loaded
```

```
from templates/default.css
```

```
/* ----- cover */
/* ----- main */
```

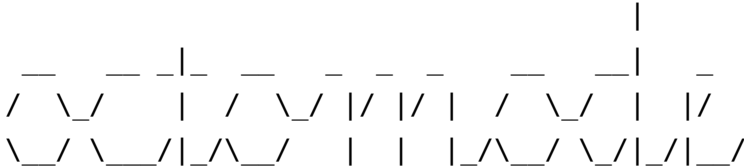
```
from static/main.css
```



```
/* ----- cover */  
/* ----- main */
```

Colophon

With thanks to Varia and especially Manetta Berends for:



Octomode is a collective editing space for PDF making, using Etherpad, Paged.js and Flask developed by Varia.

Inspired by the non-centralised, tentacular cognition capabilities of the octopus, we imagined a space in which the artificial boundaries of writing and designing can be crossed; where writing, editing and designing can be done in one environment simultaneously, allowing the format to influence the matter and vice-versa.

David Geoff Winnie Kate Aga Tommaso Ley Olga Marloes stefan Dave Juliet Erica Victor km0

a little duckling also made it here at some point

[A sleepy duckling](#)

hELLO wσ^{UP}LD hELLO wσ^{UP}LD hELLO wσ^{UP}LD hELLO wσ^{UP}LD
hELLO wσ^{UP}LD hELLO wσ^{UP}LD