

Taylor & Francis
Taylor & Francis Group

Tapuya: Latin American Science, Technology and Society

ISSN: 2572-9861 (Online) Journal homepage: www.tandfonline.com/journals/ttap20

Problems of code: ethnographic encounters with the Amazon Forest as a computational model

Felipe Mammoli & Marko Monteiro

To cite this article: Felipe Mammoli & Marko Monteiro (16 Oct 2026): Problems of code: ethnographic encounters with the Amazon Forest as a computational model, Tapuya: Latin American Science, Technology and Society, DOI: 10.1080/25729861.2026.2700795

To link to this article: https://doi.org/10.1080/25729861.2026.2700795

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

Published online: 16 Oct 2026.

Submit your article to this journal

Article views: 213

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ttap20

TAPUYA: LATIN AMERICAN SCIENCE, TECHNOLOGY AND SOCIETY
https://doi.org/10.1080/25729861.2026.2700795

THEMATIC CLUSTER: TRACING THE PAST, MAPPING THE PRESENT: DIGITAL METHODS IN LATIN AMERICAN CULTURES

OPEN ACCESS Check for updates

Problems of code: ethnographic encounters with the Amazon Forest as a computational model

Felipe Mammoli [a] and Marko Monteiro [b]

[a]Technologies in Practice, IT University Copenhagen, Copenhagen, Denmark; [b]Department of Science and Technology Policy, University of Campinas (Unicamp), Campinas, Brazil

ABSTRACT

As environmental issues become increasingly datafied, computer code is becoming more important to the quotidian work of earth system scientists. Not only has code become central to their knowledge practices, it has become a way in which social relations travel through global knowledge infrastructures, connecting knowledge, data, and people. This article explores the use of code as material for ethnographic inquiry through the study of how a predictive model about climate change in the Amazon is built at the AmazonFACE modeling laboratory. Based on ethnographic fieldwork, we show that computer code has an ambiguous nature: as both text and process. This duality involving matters of knowledge (how code connects data) and matters of interpersonal relationships (how code connects people) enables it to function as an ethnographic fieldsite. We show that access to source code is a symmetrical problem for ethnographers and modelers alike, producing partial forms of understanding. Based on reading code fragments, we examine how reading, writing, and maintaining code sustain scientific collaborations while simultaneously separating "scientific" contributions from "technical" labor. By following how new and old code are assembled in the model's source code, we show how careers, knowledge, and relations are made and maintained through computational practices.

Problemas de Código: Encontros Etnográficos com a Floresta Amazônica como Modelo Computacional

RESUMO

Conforme as questões ambientais têm se tornado cada vez mais datificadas, o código computacional tem assumido um papel cada vez mais importante no trabalho cotidiano dos cientistas do sistema terrestre. Não apenas o código se tornou central para sua prática de conhecimento, mas também se tornou uma via pela qual as relações sociais transitam pelas infraestruturas globais de conhecimento sobre o clima, conectando dados e pessoas. Esse artigo explora o uso de código computacional como material etnográfico no estudo de um laboratório que desenvolve um modelo computacional da Amazônia e que faz parte do programa de pesquisa AmazonFACE. Com base em trabalho de campo etnográfico realizado no laboratório, mostramos a natureza ambígua do código computacional: ora texto, ora processo. Uma dualidade que, ao envolver questões de conhecimento (como o código conecta dados) e questões de relações interpessoais (como o código conecta pessoas), permite que o código funcione como um campo etnográfico. Mostramos ainda que o problema de acesso ocorre de forma simétrica tanto para modeladores quanto para etnógrafos, possibilitando apenas compreensões parciais. Reaproveitamos também fragmentos de código para examinar como a leitura, a escrita e a manutenção do código sustentam determinadas colaborações científicas enquanto separam contribuições "científicas" de contribuições "técnicas". Ao acompanhar como linhas de código novas e antigas são combinadas no código-

ARTICLE HISTORY
Received 18 December 2025
Accepted 3 July 2026

KEYWORDS
Algorithms; code; ethnography; digital methods; climate change

PALAVRAS-CHAVE
Algoritmos; código; etnografia; métodos digitais; mudanças climáticas

PALABRAS CLAVE
Algoritmos; código; etnografía; métodos digitales; cambio climático

CONTACT Felipe Mammoli femama@itu.dk

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

fonte do modelo, mostramos como carreiras, conhecimentos e relações são produzidos e mantidos por meio de práticas computacionais.

Problemas del Código: Encuentros Etnográficos con la Amazonía como Modelo Computacional

RESUMEN

A medida que las cuestiones ambientales se vuelven más datificadas, el código de computadora adquiere mayor importancia en el trabajo cotidiano de los investigadores del clima. El código no solo se ha vuelto central para sus prácticas de conocimiento, sino también una vía por la cual las relaciones sociales transitan por infraestructuras globales, conectando conocimiento, datos y personas. Este artículo explora el código como material etnográfico en el estudio de un modelo predictivo sobre el cambio climático en la Amazonía, desarrollado en el laboratorio de AmazonFACE. Basándonos en un trabajo de campo etnográfico, mostramos que el código posee una naturaleza ambigua: como texto y como proceso. Esta dualidad, que involucra cuestiones de conocimiento (cómo el código conecta datos) y de relaciones interpersonales (cómo el código conecta personas), le permite funcionar como campo etnográfico. Mostramos que el acceso al código fuente es un problema simétrico tanto para etnógrafos como para modeladores, produciendo comprensiones parciales. A partir de fragmentos de código, examinamos cómo leer, escribir y mantener código sostienen colaboraciones científicas, mientras separan contribuciones "científicas" del trabajo "técnico". Al seguir cómo códigos nuevos y antiguos se combinan en el modelo, mostramos cómo carreras, conocimientos y relaciones se producen y mantienen mediante prácticas computacionales.

1. Introduction

The notion that knowledge can be detached from those who produced it and made to circulate freely as information, without reference to those people, is a particularly prevalent assumption in the modern sciences (Leach 2012a, 255; 2012b, 80; Strathern 2020, 104). However, the history of science shows that knowledge not only connects concepts, but it also connects people (Strathern 2020). Knowledge, in this sense, can be understood as analogous to the modern notion of kinship, as the substance that links people into close relations (Leach 2009, 182; Strathern 2020, 104; 2021). This debate over the inseparability of knowledge and the people involved in producing it can contribute to understanding how close relations are formed in the modern sciences and how digital technologies participate in sustaining them.

In this article, we examine how data and people get connected through the scientific practice of computer modeling in climate change research. In particular, we focus on a modeling laboratory in Brazil, where a group of ecologists is developing a computer model to study the effects of climate change on the vegetation of the Amazon Forest. We show that computer code, often obscured as a technical layer, can be explored as an ethnographic site that registers not only how different data get connected in the making of scientific knowledge, but also how this process entails connecting people.

This article draws on ethnographic fieldwork conducted between 2018 and 2021 within the modeling laboratory of the AmazonFACE program, a large international climate change research initiative hosted in Brazil to study the effects of climate change in the Amazon Forest. Our analysis builds on the interfaces between Digital Methods (Rogers 2013, 2024) and Science and Technology Studies (STS) research on digital technologies (Christin 2020; Coleman 2010; Jaton 2020; Knox and Nafus 2018; Mackenzie 2018; Seaver, 2017) as identified in Marres and Gerlitz (2016) and articulated in Carlson et al. (2021) through the proposal of including code in ethnographic analysis by treating it as a process. Methodologically, we build on their converging suggestions of repurposing artifacts native to digital infrastructures as tools for researching broader social processes.

Thus, we repurpose computer code as a methodological entry point into the contemporary knowledge practices of computer modeling. By examining snippets of code and the ways they circulate among researchers, we show how code becomes a site through which knowledge, people, and data are continually brought into relation. This analytical work, we argue, is relevant for both Digital Methods and STS studies on code. By understanding code and its material relationships to knowledge production, we enable novel methodological and empirical questions about how to approach code and coding as a knowledge practice, research site, and social process.

The article proceeds as follows. First, we review how code has been discussed in Digital Methods and STS, showing that despite being central to digital culture, it remains understudied as an empirical object. We then propose approaching code as an ethnographic fieldsite through which computational, conceptual, and interpersonal relations become observable. By engaging with code as modelers do, reading, writing, and running it, we argue that code not only structures ecological knowledge but also laboratory life itself. Finally, through three code stories, we show that: (1) code connects not only datasets but also people through situated practices of knowledge sharing; (2) access to code is not a binary condition but an ongoing process shaped by the coexistence of multiple versions and partial forms of understanding; and (3) code has an ambiguous materiality, operating simultaneously as text and process and sustaining different relations between people, infrastructure, and knowledge.

Through this analysis, we aim to contribute to ongoing debates in STS and Digital Methods about the role of digital infrastructure in knowledge-making by showing how computer code renders visible the entanglement between technical and social relations in the making of environmental knowledge about the Amazon Forest.

2. Code in digital methods and STS

Digital Methods are typically described as the repurposing of mechanisms, techniques, and data native to digital platforms for research on society (Omena 2019; Rogers 2013). According to Rogers (2013), it differentiates itself from other digital research strategies by thinking with digital devices themselves, their limits, and opportunities to develop research questions and novel methods. Omena (2019) notes that the interrelatedness between research design and mainstream digital technologies requires researchers to build a minimum level of practical and technical knowledge. This, coupled with reflexivity, enables an understanding of how the digital medium is part of the social phenomena being researched (Venturini et al. 2018).

As Rogers (2013, 2024) shows, research in Digital Methods can take many forms. In Brazil, for example, studies have applied these methods to large datasets, such as Scheren et al. (2024) in their analysis of millions of far-right political messages on Telegram, and Regattieri's (2023) investigation of disinformation bots on Twitter targeting Amazon conservation. Silva et al. (2020), in turn, repurposed computer vision APIs to investigate how different classification systems interpret image datasets and reproduce cultural stereotypes. Silva (2022), in his study of algorithmic racism, highlights that computational systems are not only classification machines but also participate in producing and scaling harmful social distinctions. Nemer (2022), through an ethnographic engagement with digital life in Brazilian favelas, shifts attention from technologies in their abstract universalism to the situated ways in which they are adapted and made usable as tools for emancipation, as well as the frictions and inequalities that follow these transformations. These suggest the importance of some "literacy" (Rosa 2022) to avoid mistaking the workings of the platform for those of the social phenomena under study (Rogers 2013).

Despite being constitutive of both the study sites and methods of Digital Methods, code is not often engaged as a material for reflexive and critical inquiry, as noted by Carlson et al. (2021) and Alcaras and Larribeau (2022). Our effort in this article to bring code to the forefront is not to suggest that everyone should learn how to write it or that all digital research should be interdisciplinary,[1] quite the contrary. It is a response to the apparent asymmetry between how much code is written in contemporary life and how little it is critically examined (Couture 2012).

Alcaras and Larribeau (2022) suggest that approaching code in its materiality is a way to avoid what Ziewitz (2016) has called "the algorithm drama," where opacity enhances the mystique and perceived power of algorithmic systems. Gillespie (2014) also mentions this dilemma, calling it "algorithmic objectivity." But as Seaver (2017) observes, what computer scientists refer to as algorithms is often different from what social scientists mean by algorithms. For computer scientists, an algorithm may refer to a very specific section of code. For social scientists, algorithms are usually systems that encompass code, people and institutions. As Alcaras and Larribeau (2022) note, algorithms are only one tool in computer science's varied toolset for making software. A focus on code in its materiality and as a process is a way to avoid what

[1]Disciplinarity, of course, has a lot to offer, as noted by Strathern (2000; 2007).

they call "algorithmic reduction" (2), or the obscuring of the many other computational activities involved in making software by the all-encompassing notion of algorithm.

Alcaras and Larribeau (2022) suggest "studying software as it is written" (3) to detach it from the mythical figures and discourses surrounding code – the algorithm – and reframe it "in its ordinary, routine and vulnerable aspects" (4). This would enable attending to other computational work, such as the interplay between writing new code and maintaining old code, caring for infrastructure, constructing identities through professional practices, the material conditions of work, the forms of knowledge involved in programming, and the relations between people that software development creates. As the authors conclude, if attention to algorithms enables critical research on the conceptual devices mobilized to make software, then attention to code provides an entry point that drives research toward the concrete practices of software making, the writing of code to be read and interpreted by machines and people.

Critical Code Studies (CCS), as proposed by Marino (2020), stems from a similar interest in treating code as a cultural object (Marino and Douglass 2023). Through close reading, CCS shows how assumptions, values, and social relations become inscribed in code. CCS thus opens space for studying software not only in terms of what it does, but also in what it says, both explicitly through textual elements and implicitly through the conventions and structures by which code is produced and shared.

This attention resonates with Riles's (2006) approach to documents as active ethnographic objects that mediate social life (Lowenkron and Ferreira 2014). Like documents, code enacts relations by structuring knowledge, coordinating labor, and circulating across institutional settings. Borrowing from Walford's (2021) insight that it is not only people who work on data, but data also works on people, we might also say that code similarly shapes the people who write, maintain, and interpret it. By leveraging the generative approximation between STS and Digital Methods observed in Marres (2012), we suggest that approaching code as a cultural and infrastructural artifact enables an ethnographic understanding of how scientific and computational practices are organized, negotiated, and sustained.

3. Code as fieldsite

The idea of an ethnographic fieldsite as a bounded space has long been problematized in anthropology (Candea 2007; Marcus 1995). Ethnography of digital media makes particularly clear that fieldsites are not naturally occurring, as if they were places waiting to be discovered, but something produced through specific relations, practices, and movements sustained by ethnographers. Particularly in digital anthropology, the ethnographic site has been described as distributed across space, mediated by communication technologies, and often embedded in digital infrastructure (Hine 2015; Knox and Nafus 2018; Miller and Horst 2021). This description is particularly relevant to the ethnographic work conducted at the modeling laboratory, since much of the modeling occurred outside the laboratory's physical space and either within the digital medium of code, charts, and messaging groups or through them.

Fieldwork at the modeling laboratory was conducted from 2018 to 2021.[2] During this period, the lab consisted of eight researchers and one IT technician: three PhD students, one professor (lab coordinator), two MSc students, and two undergraduate students. All of them were ecologists with a background in biology and little programming experience beyond their current research on modeling, except for the lab coordinator, who had extensive experience with computer modeling through their MSc, PhD, and current research agenda. The IT technician had a part-time contract and was responsible for managing the entire AmazonFACE IT infrastructure, not just the modeling lab's IT needs.

The main purpose of the laboratory was to develop a new vegetation model of the Amazon Forest to study how it will change by the end of the twenty-first century with increasing atmospheric CO2, a central aspect of climate change. Their argument for the need of a new model is that current Digital Global Vegetation Models (DGVMs[3]) do not capture the Amazon's diversity and how it could affect the

[2]This fieldwork was part of the first author's PhD research. The first author conducted the ethnographic fieldwork, contributed with analysis and writing. The second author acted as the first author's PhD supervisor, followed the research process closely and contributed to analysis, discussions and writing.

[3]Digital Global Vegetation Models are computer models intended to predict vegetation distribution according to climate conditions. Their development began in the 1990's as part of the Earth System Science research agenda to create models that are able to be coupled with Global Climate Models to increase their precision and capabilities.

forest's capacity to adapt to new climate conditions. A model developed by Global South researchers, experts in the Amazon ecology, would make a significant contribution to the debates about the future of the Amazon Forest and add nuance to the often catastrophic predictions for the future of the Amazon (Cox et al. 2000; Lapola, Oyama, and Nobre 2009; Nobre, Sellers, and Shukla 1991).

This modeling effort is part of a larger international science program called AmazonFACE, the objective of which is to investigate how Amazon Forest vegetation will respond to elevated atmospheric CO_2 expected by the end of the twenty-first century due to Climate Change. The modeling laboratory is the digital counterpart of its experimental system and works alongside it to guide data collection and to propose hypotheses for testing (Lapola 2018; Lapola and Norby 2014; Lapola et al. 2024). An important difference is that while the experimental plot is located 70 km north of the city of Manaus, the modeling laboratory is at a university in southeastern Brazil, located 1500 km away from the experimental plots.

Fieldwork included attending weekly lab meetings, attending events, participating in the group chat, and conducting in-depth interviews. By leveraging the first author's background in computer science, participant observation also included helping members configure their programming environments, engaging with the model's source code, and writing documentation. This became a strategy to gain access to activities and interactions that would otherwise be hard to observe but were nonetheless deemed by modelers as the practical side of modeling. Despite the lab being empty most days of the week, news kept appearing at meetings and accumulating in the model's code.

An interview with one of the MSc students provides a succinct description of what the modeling activity entails and the place that code has in it:

> This [showing charts about her results] is what is most enchanting in modeling, at least for me. I really like it for the results it can bring, but that is only 10% of the work. The other 90% is just trying to solve problems of code; it is trying to write what you want, trying to make sense [of the code]. Only after that can you run everything and see the results, see the changes in the climate, see how the plants are responding, etc. In modeling, you have endless possibilities in your hands, but you have to know programming and ecology to understand what is coming out [as a result] of everything that you write. (Interview conducted in 2021)

Although it is probably a figure of speech, this commentary illustrates how modelers understand their own knowledge practice and the centrality of code. Encoded in this comment is also a negative feeling toward code, as if it were the means that insists on getting in the way of proper ecology, the interpretation of results. This uneasiness is an important aspect of how these ecology researchers relate to modeling and sets an important distinction from other works that discuss people who code and their code. While in studies of Hacker communities, such as Coleman (2013), code is presented as a medium of expression and self-actualization, in this modeling laboratory formed by ecologists, code appears as an obstacle that must be overcome for real ecology to happen, much like the obstacles encountered in the forest in field ecology.[4] We use this image of code as an obstacle to highlight a certain spatiality that modeling as a practice unfolds across: the lab, the meeting, the house, and the code.

Our methodological approach of treating code as a fieldsite builds on the understanding that the field is not only the physical places where interlocutors convene, but also the materials, digital or not, through which relations and knowledge are produced (Boellstorff 2016; 2021; Marres 2012). Code becomes a site of co-presence not because ethnographers and modelers inhabit the same room, but because they share a mode of engagement with the same evolving text, whether by reading, modifying, running, and interpreting it. In this sense, code becomes a location, a kind of digital environment, where ethnographic attention can follow the sociotechnical stories that bind people and data. Here, Mackenzie's (2018) notion of operative ethnography is relevant as a methodological insight of working with code as a way of inhabiting the practices that constitute computational systems and understanding code as "a constant weaving of human and non-human entities, entangled in socio-material networks," (17) and repurposing ethnographic methods into an "anthropology of digital life" (Carlson et al. 2021, 17).

The first author's background in computer science was crucial. Gaining access to the model's code opened entry points into aspects of laboratory life that would otherwise remain opaque. By helping new lab members configure their computers, set up development environments, install software, or use version-

[4]It remains to be seen what other kinds of relations researchers will create with code as it becomes increasingly written and read through Generative AI tools.

control tools such as Git and GitHub, the ethnographer became involved in what the lab members often called "technical" work, in contrast to the "scientific" work of interpreting ecological patterns resulting from the model's execution. This provided opportunities to observe how researchers translated ecological processes into computational procedures and navigated the tension between learning ecology and programming simultaneously. And while this focus on code-centered interactions might produce a bias toward code in structuring laboratory life, it is the result of a situated encounter between the ethnographer's background and the moment when the laboratory was on the agenda. What we aim to do by suggesting code as a fieldsite is precisely to reconfigure this technical artifact into something that can be revisited and observed ethnographically for more than just its formalism.

Code as fieldsite is the operationalization of one of Digital Methods' insights, repurposing of digital-native objects to study broader social processes, to show how it can yield insight into how scientific collaboration is mediated by digital technology. Through this arrangement, it was not so much that the meetings were the context for what was written in code. By reading and writing code, we were able to understand the context in which many of the modelers' interactions were embedded. Code, in this sense, becomes a registry that contains the relations, either among people or among data, that become partially visible during meetings. Treating code as fieldsite, thus, shifts attention from the digital tools used in the sciences as a means of data extraction to considering digital infrastructures as a space of social practice where social life unfolds.

4. Relations in code, coding relations

Roughly speaking, code is written as text files composed of two types of lines, one intended to be executed by the computer, and another intended to be read by people. The latter are called comments: textual notes left by programmers that are ignored during execution but help readers make sense of how information flows through the many lines that compose a project. Yet programming, and modeling in this case, is not only about writing new code. It also involves maintaining older lines, adapting them, and ensuring that additions do not disrupt what is already in place. As they persist, these older lines register stories of collaboration and make visible how knowledge moves not only across lines of code but also across research projects.

The history of collaboration in the model's development becomes clearer when reading one of the model's main source files, *func.f90*, which contains 1,546 lines of Fortran 90 code. According to the modelers themselves, this file contains the most important sections of code, as it is where much of the process of photosynthesis is calculated – the main process that connects the biosphere to the atmosphere by describing how plants absorb atmospheric carbon.

Figure 1 is an entry point into this history of collaboration. It shows the first lines of the *func.f90* file. Each line is numbered on the left. It contains two types of lines: those beginning with "!" are comments. Line 19, in contrast, is code that the computer will execute: it creates an organizational structure, names it "photo" (short for photosynthesis), and contains all the logic related to this phenomenon. Line 16 lists the people who wrote the code, while line 17 notes that the file is based on the INPE-CPTEC-PVM2 model, an older model created by the AmazonFACE modeling lab coordinator during his MSc in Brazil's National Institute for Space Research (INPE), one of Brazil's main centers for climate modeling. Line 22 further specifies the role this earlier model played in shaping the code's logic.

The INPE-CPTEC-PVM2 is a Potential Vegetation Model designed to calculate the vegetation most suited to a specific climate condition (Lapola, Oyama, and Nobre 2009). It was an improvement over the earlier CPTEC-PVM mode (Oyama and Nobre 2004), used as the basis for Nobre et al.'s (2004) evaluation of possible climate scenarios for the Amazon in the twenty-first century. The main improvement was the addition of the carbon cycle to the potential vegetation calculation, expanding its applicability. Lapola's (2018) observation that rising CO_2 levels are a key factor to consider when modeling the Amazon's future would later become the central argument for justifying the AmazonFACE experiment. These lines show how knowledge from the lab coordinator's MSc was not simply transferred to his current project through teaching, use of model results, or tacit knowledge, but by sharing the very section of code with students' research projects. A fragment of code that traversed two different projects with almost 15 years in between them.

Figure 2 provides a different situation. In the portion of the code calculating maintenance respiration, the metabolic process that plants use to keep their tissues functioning, the comment identifies one of the lab's

```
1   ! Copyright 2017- LabTerra
2
3   !     This program is free software: you can redistribute it and/or modify
4   !     it under the terms of the GNU General Public License as published by
5   !     the Free Software Foundation, either version 3 of the License, or
6   !     (at your option) any later version.
7
8   !     This program is distributed in the hope that it will be useful,
9   !     but WITHOUT ANY WARRANTY; without even the implied warranty of
10  !     MERCHANTABILITY or FITNESS FOR A PARTICULAR 2PURPOSE.  See the
11  !     GNU General Public License for more details.
12
13  !     You should have received a copy of the GNU General Public License
14  !     along with this program.  If not, see <http://www.gnu.org/licenses/>.
15
16  ! AUTHORS: *, JP Darela, Bianca Rius, Helena do Prado, David Lapola
17  ! *This program is based on the work of those that gave us the INPE-CPTEC-PVM2 model
18
19  module photo
20
21     ! Module defining functions related with CO2 assimilation and other processes in CAETE
22     ! Some of these functions are based in CPTEC-PVM2, others are new features
23     use types
24     implicit none
25     private
```

Figure 1. The comment section where the process of photosynthesis is calculated, a description of the code's license, its authors, and some of its history.

```
1224      !=====================================================================
1225      !=====================================================================
1226       !> This function calculates the maintenance respiration
1227       !> @param temp Temperature in °C
1228       !> @param ts Soil temperature in °C
1229       !> @param cl1_mr Carbon content in leaf pool (kgC/m2)
1230       !> @param cf1_mr Carbon content in fine root pool (kgC/m2)
1231       !> @param ca1_mr Carbon content in aboveground woody biomass pool (kgC/m2)
1232       !> @param n2cl Nitrogen content in leaf pool (kgN/m2)
1233       !> @param n2cw Nitrogen content in sapwood pool (kgN/m2)
1234       !> @param n2cf Nitrogen content in fine root pool (kgN/m2)
1235       !> @param aawood_mr Carbon content in aboveground woody biomass pool (kgC/m2)
1236       !> @return rm Maintenance respiration in kgC/m2/yr
1237       !> @author: JPdarela Adapted from LPJ-GUESS code
1238       !> @date 2023-10-01
1239       !> @version 1.0
1240       !=====================================================================
1241       function m_resp(temp, ts,cl1_mr,cf1_mr,ca1_mr,&
1242            & n2cl,n2cw,n2cf,aawood_mr) result(rm)
1243
1244          use global_par, only: sapwood
1245          !implicit none
1246
```

Figure 2. A portion of code showing that the logic was adapted from the LPJ-GUESS model.

older PhD students as the author and notes, in line 1237, that the code section was adapted from the LPJ-GUESS (Lund–Potsdam–Jena General Ecosystem Simulator) model described in Sitch et al. (2003). Its inclusion resulted from a period when the lab's PhD student spent a semester as a visiting student at the University of Munich, where one of the LPJ-GUESS maintainers worked at the time.

It can be argued that the main outcome of this collaboration was the 2025 publication (Dantas de Paula et al. 2025) on the inclusion of phosphorus cycle in LPJ-GUESS, coauthored by two PhD students from the modeling lab and other researchers. Yet the article does not fully account for how the knowledge exchange during that period fed back into the AmazonFACE model not only as information but in the form of a code fragment. Through sharing these code fragments, kinship is formed not only among modelers through sustained scientific collaboration, but also among models themselves by the flux of fragments from one to another. Mackenzie and Vurdubakis (2011) refer to these effects of code, usually unintended by the code's author, as the excess that remains after code has been written and executed, a sort of durable effect extended by its performative capacity.

These code stories and their code fragments show that code carries far more than just instructions to be executed by a computer. Each line, whether a mathematical operation or a comment, is part of a network of relations that encompasses earlier models and newer ones, students and senior researchers, and Brazilian institutions and international collaborators. Code, in this sense, is not only text to be read or a process to be executed, but also a medium through which kinship is sustained, an ongoing practice of inheritance, adaptation, and exchange. Or the substance that links people in close relations. To read the source code of the AmazonFACE model is thus to encounter a history of scientific collaboration written into the very structure of the program itself.

5. No code or too much code: elusive access

Critical studies of algorithms and software systems frequently emphasize the lack of access to source code and develop methodological strategies to engage with systems indirectly, laterally, or partially (Bonini and Gandini 2020; Christin 2020). Less discussed, however, is what access means once source code becomes available. This section explores that question ethnographically by showing how source code complicates the notion of access. Rather than a binary condition, we show that access is a continuous process shaped by the collective work of writing, maintaining, and interpreting code (Seaver 2017). The following vignette presents access as both a symmetrical problem for ethnographers and modelers and a paradox: once achieved, code proliferates into multiple versions containing minor to significant variations, fragmenting the very object that access was meant to stabilize.

In a previous meeting, the senior PhD students presented the newcomers (one PhD student, two MSc students, and two undergraduates who had been attending lab meetings for nearly three months) with the ecological theories on which the model was based. That meeting was full of diagrams of nutrient, carbon, and water cycles from Earth System Science, discussing ecological theories and representations of how energy and matter circulate between plants and the environment. This new meeting promised to reveal how those theories became computational processes.

The meeting began with one newcomer expressing relief: "I'm happy to finally meet the model. You're always talking about the model, the model, the model, but I kept wondering, what is it you're actually talking about?"

After nearly three hours of opening and closing more than a dozen files, many with over 200 lines, the newcomers seemed overwhelmed by the model's source code complexity and uneasy about what they would need to learn to advance their research. To apparently lighten the mood, one of the presenters pointed to a specific line of code "ph = 0.012*31557600.0*f1*f4sum*f4shade" (Figure 3, line 72) and said:

> This is just math, that's all. What the model does is connect these variables, the data, to each other – it's nothing out of this world. Here you have photosynthesis, here leaf resistance, specific leaf area, water stress, etc.

All the new members had a background in biology, but no prior programming experience. The comment was meant to bridge the cryptic text on the screen with ecological knowledge they already had, linking lines of code to photosynthesis formulas they had studied since their undergraduate studies. The dialogue continued with a question from one of the new PhD students:

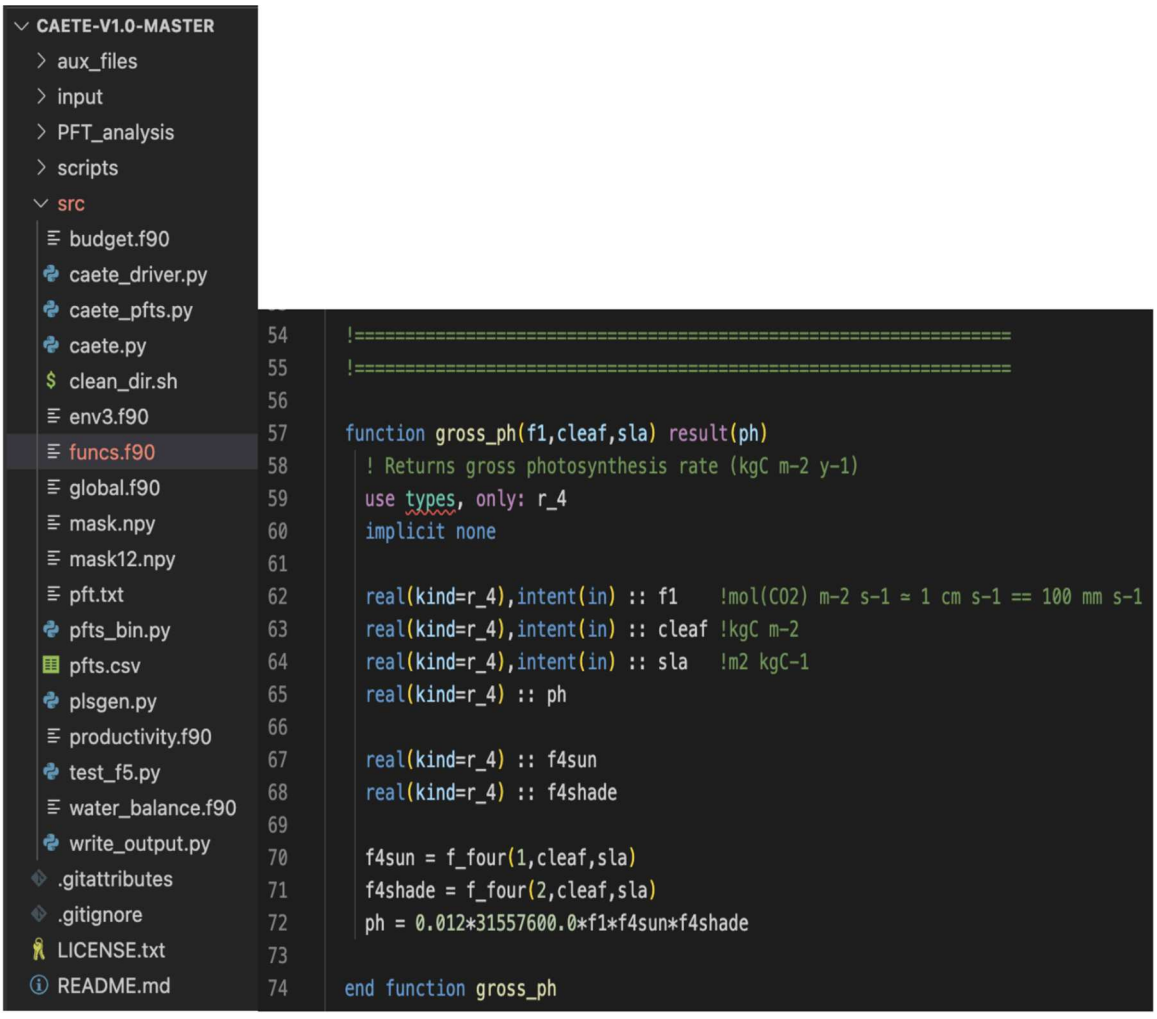


Figure 3. Left side: Some of the folders and files that are part of the source code of the AmazonFACE model. Right Side: The gross photosynthesis calculation in the funcs.f90 file. The numbers column lists the specific line numbers to which each command corresponds.

> New PhD lab member: Very interesting, but how long do you think it would take for someone to understand that [what the model's source does]? How long did you take?
>
> Older PhD student: I don't know how long it took, but it was a process. The first thing I did was try to understand the model's structure, how the logic flowed through the code. But I think I only really understood when I had to change something, when I tried something, and an error popped up. Then I had to figure out the error, fix it, and try again. And even today, there are things I don't understand, for example, the code optimizations that he [the other PhD student] did to improve performance.

The answer was not particularly motivating for the new lab members. Still, it highlighted two crucial aspects of how the model is built in code: (1) knowledge comes through hands-on engagement, and (2) modeling is a collective endeavor, with each member contributing small parts to the model's whole. The first point highlights the relationship between knowledge and access: understanding only emerges through direct engagement, which requires code to be accessible for tinkering. The second underscores that knowing the model does not mean knowing everything, but rather knowing enough to contribute to the collective work.

These observations point to a broader implication for studying code ethnographically. If understanding code comes through intervention rather than pure inspection, access need not be equated with transparency or complete knowledge. This way, code is less a finished object waiting to be inspected than an environment that must be inhabited through experimentation and failure. As noted, the model only becomes understandable to newcomers once they begin changing it, breaking it, and repairing it. In this

way, code becomes a process through which scientific relations are organized and reproduced, and not simply an object that contains scientific knowledge (Carlson et al. 2021). By taking code in practice, as suggested by Alcaras and Larribeau (2022), following code ethnographically means attending to what it represents, how it acquires its representational capacity, and to the practical work through which people learn to inhabit it, extend it, and preserve it.

A few days later, the model's code was made publicly available through a GitHub[5] repository. Soon, every lab member had downloaded it and begun tailoring the model's code to their research needs, producing at least eight divergent versions, including one of our own. If at first there was eagerness for access, once achieved, the object multiplied, making total access, meaning keeping track of all eight versions, impossible. Just as the model encoded multiple ecological subfields that a single ecologist could not master, it became clear that it would not be possible to follow every version. Here, access did not resolve the issue but multiplied it, producing what Seaver (2017) calls a "texture of access": the idea that access is never total, but an ongoing, uneven process, "a resistance to knowledge that is omnipresent and not always the same" (7). Following the advice of the senior PhD student, the ethnographic aim shifted: not to follow the model in its totality, but to engage with it hands-on, partially learning what had been done, and contributing something new.

The strategy to work on the model and keep periodic contact with every lab member was to create a technical guide for the new lab members on how to download the code, set up their development environment, and run the model, all activities that leveraged the ethnographer's background in computer science. This did not allow for total knowledge of the model or access to every version, but it allowed for following the modelers' own mode of engagement.

As anthropologist Marilyn Strathern (2000) noted in her work on bureaucracy, the assumption that access equals understanding comes from confusing transparency with knowledge. This confusion is problematic in studying algorithms because it portrays them as stable and unified, as if there were a single Facebook algorithm, an Instagram algorithm, or, in this article's case, a model's algorithm. In practice, knowing is a symmetrical problem for modelers and ethnographers alike: there are always multiple versions being worked on, and the ones writing code – in this case, the ecologists – are also trying to make sense of code written by others, and often, by themselves. As Couture (2019) notes, code is not just lines of text or files, but part of a system of relations that connect people and code. What the ethnographic vignette suggests is that gaining access to code does not resolve problems of knowing, it multiplies them. Approaching code ethnographically, thus, shifts attention from whether code is open or closed to how access is itself enacted and sustained (Seaver 2017). Source code, in this sense, becomes neither fully transparent nor inaccessible, recovering the ethnographic experience of partial knowledge, of learning enough to participate.

6. Coding separations

If previous sections showed how code acts as a medium of relation and inheritance, this section reveals that not all relations are preserved equally. Some become stabilized as visible scientific collaboration, while others disappear into infrastructure (Larkin 2013; Star 1999). Code fragments that describe ecological processes might be shared back and forth among researchers and their authorship might be registered in the code itself, but the infrastructural work that enables the code to function often goes unnoted. In a kind of activity usually pointed by the lab members as "technical," as in opposition to "scientific" or "ecological," lies different kinds of lines of code that are responsible for operationalizing the ecological processes described in mathematical and logical formalisms into a process that runs on a computer and outputs numerical results that can be analyzed as a matter of "ecology." These lines have different relational capacities.

To "run the model" is itself a significant activity for the modelers, consisting of having all the code and input data computed and producing outcomes in the form of numerical data that can later be transformed into charts and analyzed as the possible futures of the Amazon Forest. Running the full model usually takes

[5]Unfortunately, there is no time here to discuss the rich ecology of tools that code is usually entangled with. Code is rarely alone and it is usually followed by tools that help work with it, from websites that make their circulations easier to computational environments that help in the reading and writing it for programmers. For git, for example, see Straube (2016).

days of processing on a dedicated high-performance server located in the basement of the lab building. Because it is so computationally intensive, the model cannot be realistically run on a personal computer, except in what they call "test mode." In this mode, instead of simulating vegetation across the whole Amazon, the model calculates only a small 100 km^2 area (the equivalent of a single pixel at the model's spatial resolution). Even then, test mode takes several hours to complete.

The processing server was acquired by the lab's coordinator and configured by the AmazonFACE IT technician, an experienced professional who worked part-time creating the computational infrastructure for the modelers and attending to any other AmazonFACE's IT demands. Configuring the processing server, teaching the modelers how to access it remotely, running their model's version, monitoring, and then collecting the results were all part of the IT technician contract and formed the backbone of the modelers' daily work. Despite that, there is little reference to his contribution beyond the system itself that he left working for a future IT technician to manage after his contract ended, and his name does not appear in the lab's publications.

As Walford (2021) shows about data cleaning in another large scientific project in the Amazon, research authorship is not distributed equally. Some activities are placed under the research infrastructure itself, and their presence is often rendered invisible, casting them outside the network's cut that defines authorship (Strathern 1996). The modeling lab worked by separating activities between "technical" and "scientific." And, it was not uncommon to hear in the meetings that they should not "hang up on technical issues," since their time would be better spent analyzing and interpreting results rather than delving into the specific knowledge required for configuring servers, data storage, parallel computing, performance, etc. Running the model, though, activates all these issues and creates complex entanglements that the modelers themselves should sort out.

Following the senior students' advice, once we gained access to the source code, we tried to run the model and gain practical experience. Despite the explanations offered during the meeting, we could not make it work for several days. As Munk (2012) observed during a flood-modeling workshop, running a model is not straightforward: beyond ecological theory and its translation into code, it requires substantial computational knowledge. To run a model, one must learn to "feed it" correctly. More than making software comply with the user's intentions, becoming a modeler often means learning to comply with the software's demands. As Munk (2012, 144) concludes after running a model for the first time: "It is complying. Or rather, we are complying."

After an "unending stream of bugs and error messages" (Munk 2012, 144), numerous trials and errors, and by leveraging the first author's background in computer science, we eventually managed to get the model to run on our own computer. We traced the problem back to a single line of code, or better, the absence of a single line of code: "mp.set_start_method('fork')" in line 313 in Figure 4 (bottom). The reason the model worked on the senior PhD students' computers but not ours was that they were using the Linux operating system, while we were using MacOS. The PhD student's explanation was made with his experience with Linux, and they had not anticipated the problems that could arise when running the model on other operating systems.

What I eventually understood was that in Linux, when a program requests the creation of a new process, it uses a method called *fork*, which creates a process that starts exactly where the program made the request. In MacOS (and Windows), however, the method is *spawn*, which creates a new process that starts the program from the beginning, duplicating it, and making the model restart its calculations. The section of code in Figure 2 is the entry point of the model, the first thing executed by the computer when the program runs. Line 313 forces MacOS to behave like Linux when creating a new process. This did not solve all cross-platform problems, but it made the model executable on my computer.

This situation recalls Jaton's (2020) observation about programming as a practice: more than solving problems once and for all, programming often means "walking around" issues. I was not the only one to modify the source code in this way; all new lab members eventually made small adaptations to get it working on their own computers. Already, the code was diverging into multiple versions, and with my single-line alteration, I was creating yet another version, enacting the very practice of modeling.

Having the same source code, then, does not imply producing the same outcomes. On the contrary, it was precisely my specific change that allowed me to generate the same behavior as other lab members. This points to what Mackenzie (2003, 5) calls "the code's problematic mode of existence": the ambiguity of its

model_driver.py

src > model_driver.py > ...

```
307
308 if __name__ == "__main__":      João P. Darela Filho - SOMBRERO, 13 m
309
310     output_path = Path("../outputs").resolve()
311
312     if output_path.exists():
313         pass
314     else:
315         from os import mkdir
316         mkdir(output_path)
317
318     import time
319
320     from post_processing import write_h5
321     from h52nc import h52nc
322
323     n_proc = mp.cpu_count() // 2 if not sombrero else mp.cpu_count()
324
325     fh = open('logfile.log', mode='w')
326     print("START: ", time.ctime())
327     fh.writelines(time.ctime(),)
328     fh.writelines("\n\n",)
329     fh.writelines("SOIL SPINUP...\n",)
330     start = time.time()
```

model_driver.py

CAETE-DVM > src > model_driver.py > ...

```
308 if __name__ == "__main__":
309
310     # This line forces the mp module to use spawn method to create a new process.
311     # This is necessary to run in MacOC, since in this systems the default
312     # method to create new process is spawn and linux is fork.
313     mp.set_start_method('fork')
314       You, 5 months ago • force spawn to be fork, needed for macos …
315     output_path = Path("../outputs").resolve()
316
317     if output_path.exists():
318         pass
319     else:
320         from os import mkdir
321         mkdir(output_path)
322
323     import time
324
325     from post_processing import write_h5
326     from h52nc import h52nc
327
328     n_proc = mp.cpu_count() // 2 if not sombrero else mp.cpu_count()
329
330     fh = open('logfile.log', mode='w')
331     print("START: ", time.ctime())
```

Figure 4. Top: The code that the older PhD students originally shared. Bottom: The same code fragment after the ethnographer's alteration. The difference is in line 313 in the bottom image.

materiality, at once text and process. As text, code communicates a set of commands to a computer, and at the same time, as Chun (2008) reminds us, communicates with fellow programmers who might contribute to it. Programming is as much about reading as it is about writing. Yet code as text is of limited use to modelers. What matters for them is putting this text into a machine capable of processing it into numerical results, code as a process (Carlson et al. 2021). As Mackenzie (2003, 6) notes, the purpose of code is to work in a computational environment. Execution, however, is not a transparent translation from text to machine actions. The moment a user instructs the machine to run code, the text undergoes a series of transformations into languages that are increasingly close to sequences of zeros and ones that regulate the electrical impulses flowing through the transistors in the circuits that make up a computer (Kittler 1997).

The execution of code is thus an entanglement between the writing practices of programming, which encode logic into symbols, and the specific workings of processors built by different companies, such as Intel, AMD, or Apple. At lower levels, those accessed when programming complex systems, significant differences emerge in how particular computing functions are executed (Mackenzie 2006). At a smaller scale, the contingencies of this entanglement across multiple layers of the computational environment (Straube 2016) became clear when, despite having the same source code as the senior students, I was unable to reproduce the same effect or outcome by running the model. To achieve the same results, I had to depart from a different origin.

The trouble with this single line of code points to the critical reproducibility issue that climate models tend to fall short on. Despite the code being openly accessible, these programs need powerful computers, such as supercomputers or dedicated servers, to run on vast amounts of data, infrastructure that only a very few research centers can provide. It is no surprise then that the community developed other forms of circulating partial results, such as sharing code fragments between different modeling labs or the very Coupled Model Intercomparison Project (CMIP) by the World Climate Research Programme (WCRP) to compare the workings and the results produced by different climate research centers worldwide.

The line in Figure 2 did not alter the model's ecological assumptions, equations, or outputs. The line is ecologically empty; it does not model the forest. It models the condition under which the forest becomes modelable. In this sense, this intervention belongs to what the lab members often refer to as "technical" rather than "scientific" work. Unlike changes to ecological parameters or the model structure, this kind of change is rarely featured in meetings, cited in articles, or even recognized as a scientific contribution. Yet, without this kind of work, the model would not be able to run and produce results.

This distinction mirrors code's dual materiality observed by Mackenzie (2003). As text, code preserves traces of relations and makes visible the history of collaborations through comments, fragments, and its inherited structure. As a process, code depends on an expanding set of infrastructural adjustments that disappear once the model's execution succeeds. The labor of adapting the computational environments, managing versions, and maintaining compatibility is absorbed as the lab infrastructure and is rendered invisible (Edwards 2010; Star 1999). They only exist in the ephemeral temporality of the model's processing time and not in the long temporality of scientific collaboration.

Thus, code not only connects, but it also separates. It separates forms of labor according to whether they remain legible as knowledge production or dissolve into technical maintenance. The ethnographer's version, alongside the IT technician's work, becomes part of the hidden layer that does not speak to ecological formalisms but to the computational resources that underpin them. By becoming infrastructure, it ceases to appear as authorship while remaining the material condition for the model's scientific production. In a moment when code becomes ever more present in many scientific practices as a means of producing knowledge, following code requires attention not only to the relations it enables but also to the relations that disappear as infrastructure.

7. Conclusion

In this article, we sought to move beyond the conception of computer code as a mere technical intermediary in data processing. By engaging ethnographically with the model's code, we have shown that code works as a site where the quotidian work of climate change research is negotiated, stabilized, transmitted, and archived. We argue that by repurposing digital objects and processes for the study of broader social relations, code becomes more than an outcome of knowledge practices. Code becomes an object that

structures part of the scientific life in the laboratory, tying together data, concepts, people, and careers along its lines.

Methodologically, this study responds to the challenges of doing ethnography for a "data-saturated world" (Knox and Nafus 2018) by proposing that code can be navigated as a site where conceptual and interpersonal relations unfold and get inscribed. We also suggest a strategy for exploring the code of complex systems ethnographically, not by trying to comprehend it in its totality, but by emulating the very way those working with code engage with it: knowing enough to execute it and suggest changes.

This hands-on engagement with its partiality reframes the issue of access to computational systems from a binary event (access or no access) to a texture (Seaver 2017, 2022), a continuous process of collective learning and navigation through the many barriers of knowledge that are present in the fieldwork, not only for the ethnographer but also for their interlocutors. This analytical move allows for the identification of symmetries in the field. When dealing with complex algorithmic systems, access is not just an issue for ethnographic fieldwork, but also for people working within these systems, whether due to security reasons or the sheer complexity of the system at hand. Partial access is not, in itself, a limitation but a practical mode of engagement.

The dual nature of code as both text and process (Mackenzie, 2003) offers an analytical stance for understanding how environmental knowledge is produced in the laboratory. As text, the AmazonFACE model's code operates as a repository of relations, carrying genealogies that connect current students to previous modelers across Brazil and international institutions. As a process, code reveals the material frictions of infrastructure in the climate change sciences, particularly in modeling and how matters of "science" are differentiated from "technical" matters. The specific struggle to harmonize "fork" and "spawn" behaviors across different operating systems highlights that planet-scale research about the Amazon depends on highly situated computational environments and embodied experiences. This duality also produces different forms of relations: some lines connect ecological theory and others computational resources, creating partially overlapping communities organized around matters of "science" and other "technical" matters, communities temporarily brought together by the event of "running" the model.

Treating code as an ethnographic site allows a description of the model not as a disembodied system for predicting the future of the Amazon Forest, but as a living archive of relations and knowledge networks. As climate issues become increasingly datafied, treating code as another ethnographic site becomes ever more pressing to understand how knowledge about the planet is built and sustained by very situated digital infrastructures and the relations therein. By rendering the often-invisible work surrounding code visible, we have shown that the capacity to imagine the future of the Amazon Forest is intimately connected to the capacity to maintain fragile networks of humans and non-humans in which code, concepts, data, and people become participants.

Acknowledgements

We would like to thank the members of the AmazonFACE Modeling Group for their continued generosity in sharing their research with us. We would also like to thank the cluster editors and the two anonymous reviewers for helping to push the argument further. We are also thankful for the Code Ethnography Collective for their contribution on a very early version of the material presented here. This publication was possible thanks to the Brazilian Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) funding of the first author's doctoral research. This article was published Open Access thanks to the Adele Clarke Fund for Early Career Researchers, generously funded by the Society for Social Studies of Science.

Author contributions

Credit: **Felipe Mammoli:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Writing – original draft, Writing – review & editing; **Marko Monteiro:** Formal analysis, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was funded by the Brazilian Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), code 140837/2017-5. It was also supported by FAPESP (grant: 2023/09046-5); and CNPq (grant: 305403/2026-5). The publication of this work was supported by the Society for Social Studies of Science through the Adele Clarke Fund for Early Career Researchers.

Notes on contributors

Felipe Mammoli is a Marie Skłodowska-Curie Actions Postdoctoral Fellow in the Center for Climate IT (CCIT) at the IT University of Copenhagen. His research focuses on the interplay between climate change and digital technologies; the history of data in the Amazon Forest; and the theory and methods of ethnography of science and technology.

Marko Monteiro is an Associate Professor at the Department of Science and Technology Policy, University of Campinas. His research interests include sociotechnical controversies; ethnographies of interdisciplinary scientific practice; and governance of S&T in Brazil. He is the leader of GEICT – Interdisciplinary Research Group in Science and Technology and a member of the Scientific Steering Committee of the AmazonFACE Program.

ORCID

Felipe Mammoli http://orcid.org/0000-0002-1934-3068
Marko Monteiro https://orcid.org/0000-0003-4008-4985

References

Ainsworth, E. A., and S. P. Long. 2021. "30 Years of Free-Air Carbon Dioxide Enrichment (FACE): What Have We Learned about Future Crop Productivity and Its Potential for Adaptation?" *Global Change Biology* 27 (1): 27–49. https://doi.org/10.1111/gcb.15375.

Alcaras, G., and A. Larribeau. 2022. "Writing Code, Making Software." *RESET. Recherches en sciences sociales sur Internet* 11. https://doi.org/10.4000/reset.3944.

Boellstorff, T. 2016. "For Whom the Ontology Turns: Theorizing the Digital Real." *Current Anthropology* 57 (4): 387–407. https://doi.org/10.1086/687362.

Boellstorff, T. 2021. "Rethinking Digital Anthropology." In *Digital Anthropology*, edited by H. Geismer, E. Liber, 39–60. Routledge.

Bonini, T., and A. Gandini. 2020. "The Field as a Black Box: Ethnographic Research in the Age of Platforms." *Social Media + Society* 6 (4). https://doi.org/10.1177/2056305120984477.

Candea, M. 2007. "Arbitrary Locations: In Defence of the Bounded Field-Site." *Journal of the Royal Anthropological Institute* 13 (1): 167–184. https://doi.org/10.1111/j.1467-9655.2007.00419.x.

Carlson, Rebecca, Ruth Dorothea Eggel, Lina Franken, Sarah Thanner, and Libuše Hannah Vepřek. 2021. "Approaching Code as Process: Prototyping Ethnographic Methodologies." *Kuckuck – Notizen zur Alltagskultur* 1 (21): 13–17.

Christin, A. 2020. "The Ethnographer and the Algorithm: Beyond the Black Box." *Theory and Society* 49 (5-6): 897–918. https://doi.org/10.1007/s11186-020-09411-3.

Chun, W. H. K. 2008. "On 'Sourcery,' or Code as Fetish." *Configurations* 16 (3): 299–324. https://doi.org/10.1353/con.0.0064.

Coleman, E. G. 2010. "Ethnographic Approaches to Digital Media." *Annual Review of Anthropology* 39 (1): 487–505. https://doi.org/10.1146/annurev.anthro.012809.104945.

Coleman, E. G. 2013. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton, NJ: Princeton University Press.

Couture, S. 2012. "L'écriture collective du code source informatique." *Revue d'anthropologie des connaissances* 61 (1): 21–42.

Couture, S. 2019. "The Ambiguous Boundaries of Computer Source Code and Some of Its Political Consequences." In *DigitalSTS: A Field-Guide for Science & Technology Studies*, edited by J. Vertesi, D. Ribes, C. DiSalvo, L. Forlano, J. Jackson, Y. Loukissas, D. K. Rosner, H. R. Shel, 136–156. Princeton: Princeton University Press. https://doi.org/10.2307/j.ctvc77mp9.

Cox, P. M., R. A. Betts, C. D. Jones, S. A. Spall, and I. J. Totterdell. 2000. "Acceleration of Global Warming due to Carbon-Cycle Feedbacks in a Coupled Climate Model." *Nature* 408 (6809): 184–187. https://doi.org/10.1038/35041539.

Dantas de Paula, M., M. Forrest, D. Warlind, J. P. Darela Filho, K. Fleischer, A. Rammig, and T. Hickler. 2025. "Including the Phosphorus Cycle into the LPJ-GUESS Dynamic Global Vegetation Model (v4.1, r10994) – Global Patterns and Temporal Trends of N and P Primary Production Limitation." *Geoscientific Model Development* 18 (7): 2249–2274. https://doi.org/10.5194/gmd-18-2249-2025.

Edwards, P. N. 2010. *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*. Cambridge, MA: MIT Press.

Gillespie, T. 2014. "The Relevance of Algorithms." In *Media Technologies: Essays on Communication, Materiality, and Society*, edited by W. Bijker, E. Jones-Imhotep, R. Slayton, 167. Cambirdge, MA. MIT Press.
Hine, C. 2015. *Ethnography for the Internet: Embedded, Embodied and Everyday*. London: Routledge.
Jaton, F. 2020. *The Constitution of Algorithms: Ground-Truthing, Programming, Formulating*. Cambridge, MA: MIT Press.
Kittler, F. 1997. *There Is No*. Software. https://web.stanford.edu/class/history34q/readings/Kittler/There_Is_No_Software.html.
Knox, H., and D. Nafus, eds. 2018. *Ethnography for A Data-Saturated World. 1st ed*. Manchester: Manchester University Press.
Lapola, D. M. 2018. "Bytes and Boots to Understand the Future of the Amazon Forest." *New Phytologist* 219 (3): 845–847. https://doi.org/10.1111/nph.15342.
Lapola, D. M., I. P. Hartley, R. Norby, C. A. N. Quesada, A. Esquivel-Muelbert, A. Rammig, B. T. T. Portela, et al. 2024. *AmazonFACE: Assessing the Effects of Increased Atmospheric CO_2 on the Ecology and Resilience of the Amazon Forest – 2025–2030 Science Plan*. Ministério da Ciência, Tecnologia e Inovação (MCTI).
Lapola, D., and R. Norby, coords. 2014. *AmazonFACE: Assessing the Effects of Increased Atmospheric CO_2 on the Ecology and Resilience of the Amazon Forest. Science Plan and Implementation Strategy*. Brasília: Brazilian Ministry of Science, Technology and Innovation (MCTI) and the Inter-American Development Bank (IDB).
Lapola, D. M., M. D. Oyama, and C. A. Nobre. 2009. "Exploring the Range of Climate Biome Projections for Tropical South America: The Role of CO_2 Fertilization and Seasonality." *Global Biogeochemical Cycles* 23 (3). https://doi.org/10.1029/2008GB003357.
Larkin, B. 2013. "The Politics and Poetics of Infrastructure." *Annual Review of Anthropology* 42 (1): 327–343. https://doi.org/10.1146/annurev-anthro-092412-155522.
Leach, J. 2009. "Knowledge as Kinship: Mutable Essence and the Significance of Transmission on the Rai Coast of Papua New Guinea." In *Kinship and Beyond: The Genealogical Model Reconsidered*, edited by S. Bamford, J. Leach, 175–192. New York, NY: Berghahn Books.
Leach, J. 2012a. "Leaving the Magic out: Knowledge and Effect in Different Places." *Anthropological Forum* 22 (3): 251–270. https://doi.org/10.1080/00664677.2012.723611.
Leach, J. 2012b. "'Step Inside: Knowledge Freely Available': The Politics of (Making) Knowledge-Objects." In *The Politics of Knowledge*, edited by F. D. Rubio, P. Baert, 79–95. Abingdon: Routledge.
Lowenkron, L., and L. Ferreira. 2014. "Anthropological Perspectives on Documents. Ethnographic Dialogues on the Trail of Police Papers." *Vibrant: Virtual Brazilian Anthropology* 11 (2): 76–112. https://doi.org/10.1590/S1809-43412014000200003.
Mackenzie, A. 2006. *Cutting Code: Software and Sociality*. Vol. 30. New York, NY: Peter Lang.
Mackenzie, A. 2003. *The Problem of Computer Code: Leviathan or Common Power?* Institute for Cultural Research, Lancaster University.
Mackenzie, A. 2018. "Operative Ethnographies and Large Numbers." In *Ethnography for a Data-Saturated World*, edited by H. Knox, D. Nafus, 151–180. Manchester University Press.
Mackenzie, A., and T. Vurdubakis. 2011. "Codes and Codings in Crisis: Signification, Performativity and Excess." *Theory, Culture & Society* 28 (6): 3–23. https://doi.org/10.1177/0263276411424761.
Marcus, G. E. 1995. "Ethnography in/of the World System: The Emergence of Multi-sited Ethnography." *Annual Review of Anthropology* 24 (1): 95–117. https://doi.org/10.1146/annurev.an.24.100195.000523.
Marino, M. C. 2020. *Critical Code Studies*. Cambridge, MA: MIT Press.
Marino, M. C., and J. Douglass. 2023. "Introduction: Situating Critical Code Studies in the Digital Humanities." *Digital Humanities Quarterly* 17 (2). https://doi.org/10.63744/y32fgmj37w3y.
Marres, N. 2012. "The Redistribution of Methods: On Intervention in Digital Social Research, Broadly Conceived." *The Sociological Review* 60 (1_suppl.): 139–165. https://doi.org/10.1111/j.1467-954X.2012.02121.x.
Marres, N., and C. Gerlitz. 2016. "Interface Methods: Renegotiating Relations between Digital Social Research, STS and Sociology." *The Sociological Review* 64 (1): 21–46. https://doi.org/10.1111/1467-954X.12314.
Miller, D., and H. A. Horst. 2021. "Six Principles for a Digital Anthropology." In *Digital Anthropology*, edited by H. Geismer, E. Liber, 21–43. Abingdon: Routledge.
Munk, A. K. 2012. "Emancipating Nature: What the Flood Apprentice Learned from a Modelling Tutorial." In *The Social Life of Climate Change Models: Anticipating Nature*, edited by K. Hastrup, M. Skrydstrup, 144–162. Abingdon: Routledge.
Nemer, D. 2022. *Technology of the Oppressed: Inequity and the Digital Mundane in Favelas of Brazil*. Cambridge, MA: The MIT Press. https://doi.org/10.7551/mitpress/14122.001.0001.
Nobre, C. A., M. D. Oyama, G. S. Oliveira, J. A. Marengo, and E. Salati. 2004. "*Impacts of Climate Change Scenarios for 2091–2100 On the Biomes of South America*." Paper presented at First CLIVAR International Conference, Clim. Var. and Predictability, Baltimore, Md., 21 – 25 June.
Nobre, C. A., P. J. Sellers, and J. Shukla. 1991. "Amazonian Deforestation and Regional Climate Change." *Journal of Climate* 4 (10): 957–988. https://doi.org/10.1175/1520-0442(1991)004<0957:ADARCC>2.0.CO;2.
Omena, J. J. 2019. *Métodos Digitais: Teoria-Prática-Crítica*. Boston, MA: Icnova.
Oyama, M. D., and C. A. Nobre. 2004. "Climatic Consequences of a Large-Scale Desertification in Northeast Brazil: A GCM Simulation Study." *Journal of Climate* 17 (16): 3203–3213. https://doi.org/10.1175/1520-0442(2004)017<3203:CCOALD>2.0.CO;2.

Regattieri, L. 2023. "A propaganda desinformativa no projeto de destruição nacional bolsonarista: A desinformação como estratégia de governo na agenda socioambiental durante a presidência de Jair Bolsonaro (PL)." *Revista Eco-Pós* 26 (01): 105–139. https://doi.org/10.29146/eco-ps.v26i01.28005.

Riles, A., ed. 2006. *Documents: Artifacts of Modern Knowledge*. Ann Arbor, MI: University of Michigan Press.

Rogers, R. 2013. *Digital Methods*. Cambridge, MA: The MIT Press.

Rogers, R. 2024. *Doing Digital Methods*. (Vols. 1-0). SAGE Publications Ltd, https://doi.org/10.4135/9781036231514

Rosa, F. R. 2022. "Code Ethnography and the Materiality of Power in Internet Governance." *Qualitative Sociology* 45 (3): 433–455. https://doi.org/10.1007/s11133-022-09517-3.

Scheren, M. L., V. S. Rodrigues, G. D. López Zamora, L. Cesarino, L. F. Nascimento, P. Fonseca, and T. Barreto. 2024. "Métodos mistos para a antropologia digital: um relato de experiência sobre a análise de grupos bolsonaristas na plataforma Telegram." *Horizontes Antropológicos* 68: e680407.

Seaver, N. 2017. "Algorithms as Culture: Some Tactics for the Ethnography of Algorithmic Systems." *Big Data & Society* 4 (2): 205395171773810. https://doi.org/10.1177/2053951717738104.

Seaver, Nick. 2022. *Computing Taste: Algorithms and the Makers of Music Recommendation*. Chicago, IL: University of Chicago Press.

Silva, T. 2022. *Racismo algorítmico: Inteligência artificial e discriminação nas redes digitais*. São Paulo: Edições Sesc SP.

Silva, T., A. Mintz, J. J. Omena, B. Gobbo, T. Oliveira, H. T. Takamitsu, E. Pilipets, and H. Azhar. 2020. "APIs de visão computacional: Investigando mediações algorítmicas a partir de estudo de bancos de imagens." *Logos* 27 (1). https://doi.org/10.12957/logos.2020.51523.

Sitch, S., B. Smith, I. C. Prentice, A. Arneth, A. Bondeau, W. Cramer, J. O. Kaplan, et al. 2003. "Evaluation of Ecosystem Dynamics, Plant Geography and Terrestrial Carbon Cycling in the LPJ Dynamic Global Vegetation Model." *Global Change Biology* 9 (2): 161–185. https://doi.org/10.1046/j.1365-2486.2003.00569.x.

Star, S. L. 1999. "The Ethnography of Infrastructure." *American Behavioral Scientist* 43 (3): 377–391. https://doi.org/10.1177/00027649921955326.

Strathern, M. 1996. "Cutting the Network." *The Journal of the Royal Anthropological Institute* 2 (3): 517–535. https://doi.org/10.2307/3034901.

Strathern, M. 2000. "The Tyranny of Transparency." *British Educational Research Journal* 26 (3): 309–321. https://doi.org/10.1080/713651562.

Strathern, M. 2007. "Interdisciplinarity: Some Models from the Human Sciences." *Interdisciplinary Science Reviews* 32 (2): 123–134. https://doi.org/10.1179/030801807X163562.

Strathern, M. 2020. *Relations: An Anthropological Account*. Durhan, NC. Duke University Press.

Strathern, M. 2021. "What's in an Argument? Reflections on Knowledge Exchanges." *Contributions to Indian Sociology* 55 (1): 7–29. https://doi.org/10.1177/0069966720973168.

Straube, T. 2016. "Stacked Spaces: Mapping Digital Infrastructures." *Big Data & Society* 3 (2): 2053951716642456. https://doi.org/10.1177/2053951716642456.

Venturini, T., L. Bounegru, J. Gray, and R. Rogers. 2018. "A Reality Check(List) for Digital Methods." *New Media & Society* 20 (11): 4195–4217. https://doi.org/10.1177/1461444818769236.

Walford, T. 2021. "Data – Ova – Gene – Data." *Journal of the Royal Anthropological Institute* 27 (S1): 127–141. https://doi.org/10.1111/1467-9655.13484.

Ziewitz, M. 2016. "Governing Algorithms: Myth, Mess, and Methods." *Science, Technology, & Human Values* 41 (1): 3–16. https://doi.org/10.1177/0162243915608948.