Project Summary

EarthCube Data Capabilities: Collaborative Proposal: Jupyter meets the Earth: Enabling discovery in geoscience through interactive computing at scale

Our project seeks to advance interactive computing capabilities on the cloud and high performance computing (HPC) centers to better serve the needs of geoscience researchers. The proposed work will: (a) improve access to data sources and data catalogs by exposing them to users in the same interface where they conduct their computational work, (b) empower researchers to seamlessly utilize and combine cloud and HPC resources, (c) accelerate research by simplifying the process for scientists to create and deploy custom, interactive applications for their research question, and (d) facilitate dissemination of research findings to decision-makers, stakeholders, and the general public. We will achieve these goals by extending the current capabilities of Jupyter technologies for interactive computing.

Tools in the Jupyter ecosystem are designed in a modular fashion, and behave similarly on a researcher's laptop, a high-performance computing center, or the cloud. As a result, Jupyter technologies have been widely adopted across a spectrum of scientific disciplines. In the geosciences, Jupyter is one of the enabling technologies for the Pangeo project, an ongoing EarthCube-funded project that has reduced the barrier for geoscientists who enter the realm of Big Data.

In this project, we take a holistic view of the scientific discovery process, from initial data discovery, through computational analysis to the dissemination of findings. We have identified several high-impact areas where we are uniquely positioned to reduce pain-points and make technological improvements that serve scientists. These developments will be driven by the specific needs of domain use-cases in the geosciences. Our team is composed of both developers of Jupyter technologies and geoscientists who use and contribute to open-source tools. We have ample experience building tools that first meet concrete user needs, and then generalizing them to work across related fields and usage patterns. This approach ensures that real problems are solved first (avoiding the "build it and they will come" trap), and we have the necessary high-level architectural understanding to then extract generic components that can be reused and offered to a broader community.

Intellectual Merit: The Big Data era in the geosciences offers immense opportunities for transformative scientific discoveries, but this promise is often cut short by technical barriers arising from extraneous complexity. The proposed work aims to simplify this path by expanding successful open-source projects and elevating proven open-science concepts. By jointly solving problems in data management, computation, infrastructure, and geoscience, this project will develop novel approaches that neither side alone (technologists or geoscientists) typically achieves. Seamless interactive access to petabytes of data and cloud-scale resources, in an environment that can cover the lifecycle of research ideas from scientist to public consumer, is a lofty yet achievable goal; this project will make substantial inroads towards this scenario in multiple dimensions.

Broader Impacts: The tools from Project Jupyter are already used by millions of people worldwide in research, education, industry, government, and the media. They are core tools for researchers across virtually all scientific disciplines and the humanities. The challenges this project addresses in the geosciences exist for all of those disciplines as well: by working directly within Jupyter, we will ensure that the outcomes from this project have large-scale societal impacts and benefits. All our outcomes will be, in the tradition of Project Jupyter, developed in open partnership with our community of stakeholders and made available under liberal licensing terms.

Contents

1 Introduction **1**
- 1.1 Motivation: interactive computing at scale . . . 1
- 1.2 Building on success: Jupyter and Pangeo . . . 2
- 1.3 Project Team . . . 2

2 The open-source geoscience landscape **3**
- 2.1 Data . . . 3
- 2.2 Scientific software . . . 3
- 2.3 Project Jupyter . . . 3
- 2.4 Pangeo . . . 5

3 Geoscience communities of practice **5**
- 3.1 Large-Scale Hydrologic Modeling (Larsen) . . . 5
- 3.2 CMIP6 climate data analysis (Hamman) . . . 6
- 3.3 Geophysical inversions (Heagy) . . . 7

4 Technical contributions **8**
- 4.1 Data discovery . . . 8
- 4.2 Scientific discovery through interactive computing . . . 9
- 4.3 Established tools and data visualization . . . 10
- 4.4 Using and managing shared computational infrastructure . . . 10
- 4.5 An open foundation for the future . . . 11

5 EarthCube participation **11**

6 Measuring effectiveness **12**

7 Intellectual merit, scientific advances, and community growth **12**

8 Broader impacts **13**

9 Management plan **13**
- 9.1 Roles of different institutions and investigators . . . 13
- 9.2 Coordination and communication . . . 14
- 9.3 Timeline . . . 14

10 Sustainability Plan **14**

11 Results from Prior NSF Support **15**

EarthCube Data Capabilities: Collaborative Proposal: Jupyter meets the Earth: Enabling discovery in geoscience through interactive computing at scale

1 Introduction

This project revolves around the following key goals: (1) Facilitate the discovery, integration, and effective use of the diverse sources of data in the geosciences. By integrating data sources and catalogs in the shared analysis environment used by scientists, we will lower the cost of entry to research and extract value currently locked in hard-to-access datasets. (2) Empower researchers to utilize modern, scalable compute resources. By streamlining the process for scientists to transition from small-scale prototyping to large-scale computing, we will enable scientists to take advantage of shared infrastructure in a cost-effective manner. (3) Accelerate the process of discovery by enabling researchers to rapidly create and deploy custom interactive applications tailored to the research question at hand. (4) Make it possible to communicate scientific results in a manner that is tailored to the final consumers of research – be they other scientists, policy makers, students, or the general public.

We will achieve these goals by extending the current capabilities of Jupyter technologies for interactive computing. These developments will be driven by three geoscience research avenues and will be conducted in partnership with the Pangeo community.

1.1 Motivation: interactive computing at scale

Geoscientific workflows are increasingly moving beyond the capabilities of local computing hardware. Whether this is large datasets of varying types and spatiotemporal scales, or computationally-intensive algorithms that require immense computing power, it is becoming common to rely on shared infrastructure such as a cloud platform or a high-performance computing (HPC) system. In the past several years, new standards, tools, and services around shared infrastructure have made it easier explore vary large datasets, perform more complex analyses, and pursue new avenues of research.

Unfortunately, the barrier-to-entry for these tools is still too high for most individual scientists. Effectively using this shared infrastructure often requires learning entirely new software as well as developing new skills in scalable computation. Furthermore, while it has become easier to do scientific analysis on shared infrastructure, these platforms still do not readily support the iterative, collaborative nature of scientific work. Often, scientists must perform their experimentation and iteration locally, and then re-write their software so that it can be run on shared infrastructure.

In addition, it is common for geoscience datasets (particularly large, complex, interesting data) to be stored in multiple remote locations. These data have different procedures for discovering, accessing, and retrieving them, and it is impractical for each scientist to do this on their own machine. This adds extra complexity to the research discovery process, and slows down the iterative nature of exploring data and testing hypotheses. While communities such as EarthCube have made significant progress in defining community-wide standards in storing data, as well as providing common Application Programming Interfaces (APIs) for accessing that data, there is much more that we can do to empower researchers to ask important questions about the earth.

Jupyter and Pangeo are both open communities with a shared goal in developing tools and practices that make interactive scientific workflows realizable for research avenues that involve big data and big compute. In this project, we will build upon the successes of these two projects to build open-source tools that empower geoscience researchers to do their work on shared, scalable infrastructure. To avoid adding a new technical burden on scientists, these tools must abstract

away the complexity of working with shared and distributed infrastructure, giving scientists the ability to experiment and leverage these resources more easily. We will use our combined expertise in the geosciences and in building open tools for interactive computing in order to build technology that serves the geosciences community. Our project team includes members from the Jupyter and Pangeo communities, with representation across the geosciences including climate modeling, water resource applications, and geophysics.

1.2 Building on success: Jupyter and Pangeo

This is not an isolated project to develop new tools, but a proposal to advance an ecosystem of open tools that has already proven its value for geoscience researchers. The open-source scientific software stack has grown significantly in recent years, with modern, open languages such as Python, Julia, and R, becoming commonplace in the geoscience toolbox. Utilizing this stack makes it easier to build on the work of others and to create more modular, powerful tools for the geosciences. Connecting these tools is the Jupyter ecosystem, which serves as the connective fabric that joins open tools, data, and computational resources in an interactive session. The Jupyter Notebook has been adopted as a medium for interactive computing by millions of users worldwide due to its ability to support the iterative, human-in-the-loop workflows of scientists. Jupyter is language-, workflow-, and platform-agnostic, making it well-suited to bridge the gap between working locally and on shared and scalable infrastructure.

The NSF-EarthCube and NASA-ACCESS -funded Pangeo project integrates Python tools for efficiently working with large multidimensional datasets (`Xarray`) and performing parallel computation (`Dask`) with JupyterHub deployments on cloud resources and HPC centers to create computational environments that are tailored to the workflows of geoscientists. In addition, the Pangeo project has successfully advocated for cloud-friendly storage formats of large datasets. Many of these datasets are now on the Google or Amazon cloud platforms, meaning they can readily be accessed and incorporated into computational workflows performed on these clouds. As a result, the Pangeo project has enabled Big Data, software tools for analysis, and compute resources to be brought together by researchers in a manner that was previously impossible.

Pushing the boundaries of any tool-set unveils areas for improvement and opportunities for developments that streamline and upgrade the user-experience. This project takes a holistic perspective of a geoscience research workflow that leverages these state-of-the-art tools and focuses development on high-impact areas along the research life-cycle. All developments will be contributed to the thriving Jupyter and Pangeo ecosystems. This benefits the diverse community of researchers who rely on these tools and is critical to our sustainability strategy which includes making strong social and technical connections between developers of Jupyter with geoscience researchers.

1.3 Project Team

Our team is an interdisciplinary collaboration that brings together software developers, geoscientists, and statisticians from both the University of California at Berkeley and the National Center for Atmospheric Research. Within the geosciences, we have experience in solving large-scale computational problems with open-source tools and working with big data. The team includes active participants in the Pangeo project and researchers who are new to the project, providing the opportunity to extend the Pangeo community to other domains of the geosciences. We are well-positioned to understand the needs and challenges facing the geoscience community, and through both the Jupyter and Pangeo projects, we have demonstrated the merit of our development philosophy: build tools that are first designed to solve specific problems, and then generalized to other domains across the sciences.

2 The open-source geoscience landscape

The work we describe in this proposal is primarily about increasing the efficacy of tools that are already used by researchers in order to have maximum impact. In this section, we provide an overview of several tools that we have identified as relevant to the EarthCube community; we will work to improve access and interoperability between these open resources so that they can be effective in serving a wide community of geoscientists.

2.1 Data

One of the biggest challenges to data sharing and (re)use is in adopting standards for data formats and necessary metadata. Fortunately, several EarthCube projects have improved the accessibility and use of data across a number of geoscientic applications. Data standards such as NetCDF, THREDDS, and geoJSON [4, 8, 2] have an ecosystem of domain-specific tools and infrastructure that facilitate the use of these data. Where data is centrally-hosted, standard data formats and APIs have made it easier to access these datasets programmatically. Emerging data storage formats, like Cloud Optimized GeoTIFF (COG) and Zarr, allow scientists to access parts of large data collections stored on remote resources (like cloud object store) without having to first download the complete data archive. Ongoing work in the Pangeo project, supported by the NASA ACCESS program (#4200677983), is improving the discovery and performance of these cloud optimized data formats for a range of geoscience use cases.

2.2 Scientific software

High-level, open-source languages such as Julia, Python, and R have gained popularity across scientific disciplines. These languages are open and free (facilitating reproducibility and collaboration), and each language has a growing ecosystem of interoperable software packages for scientific computation. Within the Python ecosystem, tools such as Numpy[51] enable efficient matrix algebra. Higher-level packages like Scipy[39] contain tools for sparse linear algebra, optimization routines, and a range of other analysis tools common in scientific computing. Xarray[23] and Dask[18] offer the scientific community a way to scale existing scientific workflows to larger applications and to take advantage of distributed computing resources.

On top of these numerical computation libraries is a well-developed layer of domain-specific tools for the geosciences. These packages try to make it easy to perform particular computations and visualizations that are relevant to geoscience (e.g. MetPy [35] in meteorology and SimPEG[14, 22] for geophysical inversions), and interface with the standard data formats described above.

2.3 Project Jupyter

Project Jupyter creates open-source tools that that facilitate interactive computing; these span the spectrum from low-level tools such as the specification for how code is run interactively up to the web-based user interface that a researcher uses. Jupyter is agnostic of programming language, enabling researchers to choose the language and software packages that are most suited for the task at hand. It also aims to be agnostic about what back-end computational resources are employed. It supports both local and web-based (e.g., cloud or HPC) workflows on a wide range of hardware and providers, facilitating a consistent experience for a researcher through prototyping, iterating, analyzing, and communicating science across many scales of computation. While there are dozens of tools in the Jupyter ecosystem, below we briefly describe a few tools that are core to this proposal.

The **Jupyter Notebook document** specification is a JSON-based way for storing narrative text (in markdown), code (in any language), code outputs such as plots, and interactive components such as widgets. It also tracks metadata about the document. In a scientific workflow, it allows researchers to conduct an analysis in an iterative manner where they can tune parameters and

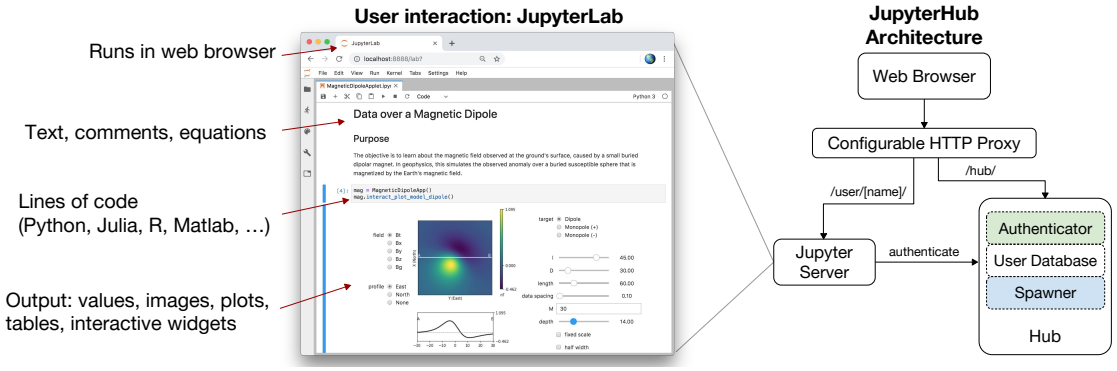

Figure 1: Schematic of Jupyter deployed on shared computational infrastructure. A researcher interacts with a Notebook document through the JupyterLab interface. The document can contain text, code, plots and interactive widgets. A JupyterHub is deployed to manage user-authentication, resource-allocation (e.g. CPU, RAM), and serves the JupyterLab interface to users over the web.

examine intermediate results before moving on to the next step.

JupyterLab is a next-generation user-interface for interactive computing. It is modular and provides a flexible, customizable user-interface for doing computational work. Not only can users run notebooks through JupyterLab, they can access a terminal or text editor, and connect custom extensions to the computation (e.g. for monitoring the progress of a parallel computation or for generating interactive data visualizations). It is a web-based tool, and offers a consistent user interface whether you are working locally on a laptop, or connecting remotely to a distributed computing system.

Jupyter Widgets is a specification for how users can create interactive visualizations and user-interfaces using Python code. The resulting widgets can be used in user interfaces such as JupyterLab in order to give scientists the ability to quickly create rich and interactive experiences with their analyses. They can also be used to create dashboards that let scientists quickly select parameters or options or an analysis and see the results interactively. They are particularly useful for communicating ideas that are backed by computation, but that don't explicitly require individuals to see any code.

JupyterHub is a tool for managing users and Jupyter sessions on shared infrastructure. It provides a central location for users to log-in and authenticate themselves, and manages the computational infrastructure needed to provide users with interactive sessions on the shared infrastructure. JupyterHub works on many scales – it can be used on a single, local machine (for example, as a central computational hub for a small team of scientists), as well as in a large-scale and distributed deployment (for example, to teach a course of 1,500 students). It is also highly customizable, and can serve a number of user-interfaces and scientific software environments. A related project called **Binder** is built on JupyterHub. It is a deployable service that lets scientists share their scientific workflows and results as open code repositories. Binder will build the computational environment needed to run the code in these workflows, and provides a a URL that scientists can share with others to immediately run their code, reproduce results, and interact with their findings. An example of one such deployment is at `https://mybinder.org`. In the past 12 months, 3.5 million users have visited mybinder.org. In addition, as a mechanism for sharing training material and on-boarding new community members, a BinderHub was launched in August 2018 at `https://binder.pangeo.io` and has since had 4300 user-sessions launched.

2.4 Pangeo

The Pangeo project is a distributed community effort to improve open-source software and computational infrastructure for Big Data applications in the geosciences. Originally funded through an EarthCube integration project (NSF award #1740648), Pangeo has focused on solving general problems and improving the integration of open-source tools in the scientific Python ecosystem. The *Pangeo Platform* is simply a modular composition of these individual projects. Deployed on HPC and cloud systems alike, the Pangeo Platform includes the following components: a browser-based user interface (Jupyter), a data model and analytics toolkit (Xarray), a parallel job distribution system (Dask), a resource management system (either Kubernetes or a job queuing system such as PBS), and a storage system (either cloud object store or traditional HPC file system). The modular composition of the platform allows for individual components to be readily exchanged and the system to be applied in new use cases.

3 Geoscience communities of practice

Our proposal is driven by the needs of scientists, and our strategy is to build tools with researchers who are making progress on areas of active research, and iterate with the researcher to refine those tools. We specifically focus on the needs of three communities of practice that will drive the technical developments in this project. Scientists on the project team span the fields of Large-Scale Hydrologic Modeling (Larsen), climate data analysis (Hamman) and Geophysical inversions (Heagy). Hamman (NCAR PI) and Paul (NCAR senior personnel) are core to the Pangeo project and will provide perspective on the needs of current users of Pangeo, as will our unfunded collaborators Dr. Ryan Abernathey at Columbia and Dr. Rich Signell at USGS. Larsen and Heagy are new to the Pangeo community and will help us extend Pangeo to their domains and networks of researchers in the earth sciences.

We will use the research applications of these scientists as a vehicle for delivering project developments to broader communities. This includes demonstrating open-science best-practices and the new capabilities our project enables through conference and journal publications, as well as through the dissemination of educational material via tutorials and workshops (enumerated in Sec 6). We will work to see these materials adopted in various undergraduate courses taught by Drs. Larsen and Pérez at UC Berkeley and in a course taught by Dr. Abernathey at Columbia that already uses the Pangeo platform.

3.1 Large-Scale Hydrologic Modeling (Larsen)

Although large-scale climate models have now been in development and use for decades, large-scale hydrologic models have only recently come on the scene, with NOAA launching its national-extent National Water Model (NWM) in 2016 [34] and the US Geological Survey launching its National Hydrologic Model (NHM) in 2018 [45]. These models are designed to produce projections of streamflow from the timescale of operational forecasts (e.g., one to 30 days ahead; NWM) to longer-timescale planning periods (e.g., monthly, yearly; NHM), using inputs from climate and land-surface models. Because of the newness of these products, and compelled by their planned use for flood hazard mitigation and water management operations, there is a need for tools that enable model evaluation and benchmarking against observational datasets and that facilitate visualization of both model outputs and hydrometeorological observations.

To first order, a salient need is that of bringing together model-generated streamflow forecasts or hindcasts with sensor-based observations of discharge and available hydrometeorological forcing factors, such as precipitation, temperature, relative humidity, and snow-water equivalent. To do so, several practical challenges must be overcome. First, while NWM outputs are provided in standard NetCDF format through the NOAA Operational Model Archive and Distribution System

(NOMADS), sensor data from hydrologic observatories are provided in diverse formats, with different variable naming schemes, different conventions (for example, for recording data gaps), and different quality control protocols. Second, there is often a mismatch in spatial scale between the outputs of the national-extent hydrologic models and data from hydrologic observatories, which tend to be situated in small, headwater catchments.

To address these two sets of challenges, we will leverage Larsen's ongoing efforts as lead PI of a USGS Powell Center for Synthesis project on watershed storage and controls. Co-PIs on that project include David Gochis, one of the NWM developers at NCAR, and Jessica Driscoll, one of the NHM developers at USGS. As part of that project, we have identified a set of pilot watersheds (examples include the East River, CO and HJ Andrews Experimental Watershed, OR) for which model output is comparable in scale to the scale of observations. We are also engaging in a synthesis of a much broader set of sensor data and model outputs, with the idea that as regionalization efforts enable downscaling of model outputs, direct comparison of modeled and observational data will be possible for a much larger set of watersheds. While that synthesis will make these distinct datasets available in a common format (NetCDF), here we propose to develop a series of workflow and visualization tools in Jupyter to allow users to locate available sensor and model-generated data and customize the data quality assurance/quality control protocol. Customization tools will include user-selectable options for identifying outliers and data gaps, filling data gaps, removing seasonal or diel cycles, and evaluating and/or transforming the data's distribution.

3.2 CMIP6 climate data analysis (Hamman)

The World Climate Research Program's Coupled Model Intercomparison Project is now in its sixth phase [CMIP6; 19]. When the archive is complete, expected by mid-2019, CMIP6 will include 287 coordinated climate model experiments from dozens of modeling centers around the world. These simulations are expected to provide the most comprehensive and robust projections of future climate ever produced. In the coming years, these simulations will serve as the basis for thousands of fundamental research and climate change adaptation studies.

The complete CMIP6 archive, which is expected to exceed 18 PB in size, will be distributed via the Earth System Grid Federation [ESGF; 13]. ESGF is a decentralized peer-to-peer database system for storing and distributing large volumes of scientific data. In the case of CMIP6, the shear size and high-dimensionality of the dataset present significant challenges to the scientific user community; challenges that ESGF will likely struggle to cope with on its own [10]. Beyond the challenges associated with CMIP6's data volumes, other significant obstacles stand between scientists and ground breaking research. Some of these obstacles include complex archive structures, lack of co-located compute and storage, and rigid data access patterns.

Scientists in the climate science domains are increasingly turning to advanced statistical methods and machine learning as tools to help make sense of the inherent complexities in large climate model archives such as CMIP6 [46]. These approaches require advanced pipelines for data discovery, acquisition, transformation, computation, and visualization that can be easily tailored to a range of specific use cases. In the case of data discovery and acquisition, we envision a well-integrated set of modular tools for data cataloging (e.g. the Spatio-temporal Asset Catalog) and data access (e.g. OpenDAP) that abstract unnecessary complexity away from scientists and support more intuitive interactions between scientists and their data. For large datasets like CMIP6, we expect data proximate computing to be of paramount importance to facilitate time and cost effective use of the archive.

The Jupyter and Pangeo projects are well positioned to address some of the key computational challenges that currently exist in common climate analysis and machine learning work-

flows. Working with the Pangeo community and integrating specific developments from this project, we will produce a series of topical tutorials that demonstrate the use of new Jupyter-based tools for data discovery, data visualization, and interactive computing with CMIP6 datasets.

3.3 Geophysical inversions (Heagy)

Three-dimensional models of the subsurface are critical for characterizing, managing, and monitoring natural resources such as groundwater, as well as for understanding the risk that natural hazards such as volcanoes pose to surrounding communities and to air-traffic routes worldwide. In both of these examples, electrical conductivity is a diagnostic physical property which can be used to construct a 3D model of the subsurface. For groundwater applications, clay-layers, which act as aquitards that prevent flow, are typically more electrically conductive than the sedimentary units which host the aquifer. The hazard that a volcano poses is largely governed by the characteristics and composition of its magma chamber; the mineralogical content and melt fraction both influence the electrical conductivity of the magma chamber. For these examples, an electromagnetic survey can be employed to collect measurements that are sensitive to the contrasts in electrical conductivity. To recover a 3D model of the subsurface, we pose the inverse problem as an optimization problem. It can be solved by minimizing an objective function that consists of a data misfit and a regularization term [50, 41, 16]. In the case of electromagnetics, the posed optimization problem is non-linear and one approach is to solve it using a gradient-descent algorithm. This requires many evaluations of the 3D forward simulation of Maxwell's equations and thus the algorithm must be efficient.

There are established proprietary codes [21, 26, 12] and open-source[24] codes written in low-level languages [32, 27] which can efficiently solve the electromagnetic inverse problem. However, as more complex questions are being asked of the data, there is a need to explore new methodologies and approaches for integrating multiple data types. Statistical and machine learning techniques for including geologic or petrophysical information are active areas of research (e.g. [48, 11]), as are joint inversion approaches which include multiple types of geophysical data, each of which is sensitive to different physical properties (e.g. [15, 49]). To support sustained, extensible research in these directions, a modular set of open, interoperable tools are needed – this is the motivation for the open-source Python project SimPEG (for Simulation and Parameter Estimation in Geophysics) [14, 22]. The growing SimPEG community encompasses researchers at universities including UC Berkeley, Stanford, Colorado School of Mines, and the University of British Columbia, as well as at geologic surveys around the world (USGS, New Zealand, Canada).

There are two aspects of efficiency we will work to improve: researcher efficiency and computational efficiency. Setting up and solving the inverse problem requires researcher input and often tuning and iterative re-evaluation at many steps: from creating a simulation mesh to designing a survey that will produce data sensitive to the target of interest to selecting tuning parameters in the stated inverse problem. To improve researcher-efficiency, we will develop custom Jupyter widgets and dashboards that facilitate exploration and decision making at each of these steps. To improve computational efficiency, we will work with the Pangeo team to incorporate Xarray and Dask into SimPEG so that it can scale to tackle large 3D problems on shared computational infrastructure. As motivation for these developments, we will examine electromagnetic data collected over the Okmok volcano in the Aleutian Islands with collaborator Dr. Paul Bedrosian (USGS). Additionally, we will invite members of the SimPEG community to test-drive the developments. To facilitate adoption by the wider community, we will develop and deploy educational resources for inverting a range of geophysical data types (e.g. magnetics, gravity, electromagnetics).

4 Technical contributions

The lifecycle of research in geoscience presents some common motifs (that are shared with other data-intensive disciplines): search for relevant data; exploratory data analysis (EDA); in-depth modeling with appropriate tools ranging from ODE- and PDE-based forward models to optimization and machine learning techniques; complex visualization of intermediate results (often with 3D tools), and collaborative discussion with colleagues. This process is typically iterative. Results drive new questions, which drive new analyses, which ideally lead to the publication of findings. It may also include communicating results to decision-makers, students, and the general public. This is especially true for geoscience questions that have a broader societal impact, such as resource management and climate science.

Throughout this process, components of the geoscientist's workflow often require access to big data and significant computational resources. Pangeo and EarthCube have successfully advocated for standards around earth sciences datasets that are stored in the cloud. Pangeo facilitates the use of these datasets by deploying JupyterHubs in the cloud that provide the environment and necessary hardware to move their workflows into the cloud and interact with this data. However, there is still significant friction in this transition. In the subsequent sections we identify several "pain-points" along the research cycle that our project will address, along with opportunities to leverage and adapt emerging open-source technologies to accelerate geoscience research. At each step of the way, our goals will be to improve exsisting open-source technology, create new technology where necessary, document our approach and the tools

4.1 Data discovery

An early step of any data-intensive scientific workflow is to find a dataset and load it into memory. Data catalogs provide a way to expose datasets to the community in a way that is structured and easier to access. For example, Unidata's Thematic Real-time Environmental Distributed Data Services (THREDDS) [8] and the Spatio Temporal Asset Catalog (STAC) [7] have both improved the discoverability and accessibility of valuable datasets in the geosciences. Moreover, open source tools such as Anaconda's Intake package make it easy for researchers to build user-friendly programmatic interfaces to these datasets [3].

Traditional scientific data analysis has been oriented around datasets stored on file systems. Such pipelines implicitly assume that the data is locally available, and can be loaded into memory (either all at once or in chunks). The Jupyter Notebook Server has followed this model by providing a contents REST API for loading whole datasets into the client. As both data and the compute necessary to analyze it has grown, these assumptions have begun to break. Datasets may no longer be available on local file systems, or even able to fit on a single hard drive. They may be hosted in traditional hierarchical array-based formats on HPC systems (like NetCDF or HDF5), or in cloud-friendly formats on Amazon S3 or Google GCS (like Zarr or Cloud-Optimized-GeoTIFF). Our project will will work to expose data catalogs through the JupyterLab interface which researchers are using to perform their computational analysis.

Development task: JupyterLab extensions for data catalogs. There are a number of emerging community standards for data catalogs, and we will build a user interface template to expose these to researchers. This user interface will be available as a JupyterLab extension which can be installed by the user or included as a component of a custom institutional deployment. Despite there being multiple different catalogs available, there are a number of common features that will be useful to provide. These may include: (a) Metadata for the dataset, including sizes, licenses, names, authors, and timestamps. (b) Drag-and-droppable code snippets for ingesting and analyzing data in popular programming languages. (c) Previews of the data, such as the first few rows of tabular datasets, or downsampled imagery for satellite data. (d) Collaboration tools for

annotating and commenting on datasets within a multi-user environment.

We will use the THREDDS, STAC, and Intake catalogs as a test cases for this development. We will build appropriate documentation and general programmatic components for connecting data catalogs to JupyterLab so that this model can be replicated across other data catalogs and service in the earth sciences.

4.2 Scientific discovery through interactive computing

The Jupyter Notebook has been adopted by many scientists because it supports an iterative, exploratory workflow combining code and narrative. Beyond code, text, and images, Jupyter supports the creation of Graphical User Interfaces with minimal programming effort on the part of the scientist. The `ipywidgets` framework lets scientists create a minimal Graphical User Interface (GUI) with a single line of code, while still allowing for extensive customization and more complex interfaces when required. These "Research GUIs" provide access to interactive elements such as sliders, buttons and menus while still living in the Notebook, side by side with regular code and narrative text. This "Just in Time" model gives scientists the ability to seamlessly change their mode of interaction from writing and executing code (e.g. "researcher-programmer" mode) to exploring results through a user-interface ("researcher-interpreter" mode). Since they live in the data and code context of the rest of a Notebook-based exploration, these GUIs can be developed only with the necessary functionality for the problem at hand. They avoid the complex design and development cycle typically associated with the development of standalone GUIs and foster instead the development of modular libraries of mini-GUIs ready to be deployed and used when needed. Furthermore, these are used within the mental flow of the rest of the analysis and narrative, without requiring a disruption to open a separate standalone data management or visualization GUI.

Development task: Widgets and dashboards. In this project, we will develop custom widgets tailored at the specific scientific needs of each of our driving use cases, as detailed in §§3.1, 3.2 and 3.3. These will be contributed as new modules in existing libraries (e.g. for SimPEG [14, 22]) or as new standalone tools, as appropriate. As part of this effort we will collaborate with our partners at Kitware Inc. on the development of custom widgets to support interactive 2d and 3d visualization in JupyterLab based on their PyGeoJS, JupyterLab GeoJS and ITK Jupyter Widget tools [29, 28, 25].

A Jupyter Notebook document typically contains both the code for an analysis and its companion narrative. This makes it an ideal medium to expose results to audiences who are interested in these outcomes and may want to explore further, but without the interest or skill set for writing code. These audiences can be students, policy makers or the media, for example. The (now unmaintained) Jupyter Dashboards Layout Extension [17] prototype showed that a Notebook could be presented to such an end user as a live dashboard without the need for re-creating the analysis, thus closing the gap between development and interpretation. In this project, we will partner with our colleagues at QuantStack, who lead the development of Voila [44]. Voila provides a secure framework to render notebooks as interactive web applications with fine-grained control over security, elements to be displayed (such as hiding code), layout and more. We will develop Voila-based dashboarding tools that can expose the analysis workflows of our domain use cases to third-party users, e.g. interactive dashboards for groundwater managers or farmers to access hydrological predictions and subsurface inversion models, and for policy makers and agency managers to explore, compare and contrast the wealth of data in the CMIP6 model archive.

We will contribute to the Voila codebase all generic improvements, and will provide demonstrations of how to deploy such dashboards securely against large datasets and with substantial computational resources (either cloud- or HPC-based).

4.3 Established tools and data visualization

Once a dataset has been identified, it is then necessary to understand its internal structure and basic characteristics. Currently this often involves writing custom code to explore the contents of a dataset, or using desktop-based data viewers such as Ncview [42], a widely used tool for visualizing the contents of NetCDF files. Like many fit-for-purpose research GUIs (graphical user interfaces), this work-horse tool is a desktop application and is not easily incorporated into a cloud or HPC based workflow. This is also true of many of the widely-used tools for exploring 3D datasets (such as Paraview). When a researcher wants to visualize these types of datasets, they are then faced with the choice of either learning a new set of modern software tools (that may or may not yet have functionality equivalent to the desktop software), or downloading the dataset to their local machine. In cases where the dataset is large, the latter option is either completely impractical or extremely inefficient.

Development task: Running desktop applications on JupyterHub. JupyterHub can readily serve non-Jupyter web-native software applications such as RStudio, Shiny applications, and Stencila, to users. Under this project we will extend the types of software that can be delivered to users to include desktop-native applications that run on Linux distributions, such as Ncview and Paraview. We will use proven Virtual Network Computing[9] technologies to allow users to access traditional desktop applications without having to modify the applications themselves. Beyond visualization software, these developments will provide a trajectory for useful legacy software to be adopted into Cloud and HPC workflows.

4.4 Using and managing shared computational infrastructure

JupyterHub makes it possible to manage computing resources, user accounts, and provide access to computational environments online. However, managing this shared infrastructure is not trivial, and successfully doing so can have a significant impact on the scientist's ability to explore, analyze, and share their work. Currently, JupyterHubs in the Pangeo project are deployed and maintained using the Zero to JupyterHub guide [43] alongside the HubPloy library [40]. Together, these libraries have simplified the initial setup and automated upgrades to the Hubs. However, there are many improvements that can make JupyterHub more suitable for larger, more complex organizations, giving the hub maintainer more insight into the activity patterns and resources that are being used. Our efforts will focus on two core areas: user management and resource usage. To improve the use of JupyterHubs in large organizations that provide access to different groups of users, we will develop more fine-grained information and controls about user identity and permissions. For example, most universities have complex accounting structures that are tied to lab-specific grants. In order to make a JupyterHub deployment sustainable for an institution (or a distributed project such as Pangeo), it is necessary to link user identity with account management information; this supports payments for the use of computational resources. Keeping track of users' resource use in a form that is useful to both users and administrators is a challenge. When done effectively, it lets users be aware of what resources are currently available to them, enabling them to use these resources as they see fit with minimal cognitive overhead. Administrators can also use this to understand the long term usage pattern of their users, and help tailor the deployment to the community it serves. This can be addressed by both collecting better usage metrics and presenting these data in clear dashboards for users and administrators.

Development task: Tracking and exposing usage information. To enable more efficient usage and administration of JupyterHubs on shared computational infrastructure, we will build tools to: (1) collect actionable metrics about resource usage including CPU and Memory usage, available Disk space, or hardware and software failures within JupyterHub deployments, (2) expose these metrics to users so they can adapt their computational work accordingly, (3) expose these metrics

to administrators so they can plan & customize the deployment to better meet user's needs. We will work with institutions and users to figure out what metrics would be most useful to them and leverage existing software tools such as Grafana or Prometheus to display this information to users and administrators. The outcome of this work will be a customizable template that can be included in JupyterHub deployments that follow the Zero to JupyterHub guide.

Development task: Tools for managing complex JupyterHub deployments. Continuous deployment tools are required for administrators to maintain large and complex JupyterHub deployments effectively. They provide reliable and repeatable installation and upgrade processes, and make it possible for a small number of administrators to manage a large number of complex deployments. We will work on improving HubPloy and similar tooling to help continuous deployment be adoptable by a wide community of deployment maintainers, including those who maintain Pangeo deployments. We will also work to create a community of practice around managing complex JupyterHub deployments so that the number of resources available to administrators can continue to grow and best practices can be spread. These tools will be validated at some of the nation's HPC facilities. In addition to our work at NCAR, we will collaborate with Rollin Thomas and his team at NERSC to develop and test these tools in a heavily used HPC environment. NERSC is a DOE HPC facility that has provided JupyterHub access to its supercomputers for the last few years and has seen broad and satisfied uptake of these tools by its user base.

4.5 An open foundation for the future

All outcomes from this project will remain as generic as possible on two technical dimensions: computational platform (cloud vs HPC) and cloud vendor. The Cloud-or-HPC debate is an active one in the sciences, and arguments continue to be made for workloads that are better suited for either option under suitable assumptions. We do not aim in this project to resolve that question, but rather to offer scientists an interactive computation and data access experience that is as agnostic to these differences as possible. Furthermore, while all cloud vendors offer at first sight similar technologies (Linux-based virtual machines and containers, block and object storage, databases, networks, etc.), in practice they all work hard to gain competitive advantages by presenting their specific flavor of each of these. Even if science-enabling software is licensed as open-source, if it relies on an API or set of tools specific to one vendor, then users are locked into that vendor's proprietary paradigm. Vendor lock-in poses a significant threat to scientific reproducibility and long term sustainability of projects, so an expressed goal of this project is to provide a layer of tools that avoids it. We will demonstrate this by partnering with Ryan Abernathey and the broader Pangeo project to test deployments on diverse infrastructure (currently Pangeo has deployments on Google, Amazon, and on the Cheyenne HPC System at NCAR). Additionally, PI Pérez, is co-PI on a submitted proposal for the NSF INFEWS program, that if funded, would also result in a JupyterHub deployment and can serve as an additional test case. To accomplish this, we will maintain all JupyterHub deployment tools independent of any cloud vendor-specific APIs. In cases where connecting to vendor APIs is necessary, we will develop versions for at least two commercial cloud providers as well as for the OpenStack environment, a vendor-agnostic project that exposes fully open abstractions for cloud and datacenter management and deployment.

5 EarthCube participation

Our team of domain-specialists in the geosciences, along with members of the Jupyter and scientific Python communities, provides an opportunity to create a strong bridge between the EarthCube community and the broader open source science community. We believe that this will yield meaningful contributions to the EarthCube mission. Our knowledge of building user-friendly, interactive workflows that facilitate data analytics will benefit many in the EarthCube program,

and our focus on scalability and bridging the gap with cloud- and HPC-based computation will help the community ask and answer more complex questions in the geosciences. We will also be able to expand the Pangeo community with representation in new fields of geoscience, such as hydrology and geophysical inversions.

Moreover, because Jupyter's primary goal is in facilitating *pre-existing* tools and workflows in the scientific community, we are well-positioned to increase the adoption and value of other tools in the EarthCube community. For example, EarthCube has championed many community standards around data formats and metadata (such as NetCDF, GeoJSON, and THREDDS). Our team's goal is to ease the interoperability of these formats so that scientists can leverage the extensive amount of data that is already available to them.

We will participate fully in EarthCube governance, planning, demonstration, and assessment activities. Science users from UC Berkeley will volunteer for the the Science Committee, and the team members from NCAR will volunteer for the Technology & Architecture Committee. Annual travel for all project PIs to the EarthCube All-Hands meeting has been budgeted. We also intend to help foster increased EarthCube presence on Github, which is the central tool our team uses for communication, project management, and code version control. Our team's well-developed practices for continuous integration and testing could serve as a template for future EarthCube software-related activities.

6 Measuring effectiveness

We will measure the effectiveness of our work through the following metrics. All target numbers are three year totals unless otherwise specified. **Community workshops**: 2 AGU workshop for $\sim$50 participants each, 2 webinar for $\sim$50 participants each, with video recordings available afterwards. **Documentation and tutorials**: (a) connecting data catalogs to JupyterLab, (b) deploying desktop apps through JupyterHub, (c) deploying a "Research App", (d) accessing user metrics on JupyterHub, (e) continuous deployment for JupyterHub deployments using Hubploy. **New Pangeo communities:** minimum of 2 (hydrology, geophysical inversions). **Pangeo deployments** (5 in total) that adopt project technologies and practices. **"Research Apps"** (minimum of 3) that are accessible to the general public, one for each geoscience community-of-practice. **Journal publication:** 5. **Conference presentations:** 10.

7 Intellectual merit, scientific advances, and community growth

This is an interdisciplinary effort at its core, drawing on the expertise of domain geoscientists working on specific questions to build widely applicable computational tools that will serve a broad community, in geoscience and beyond. Its intellectual merit comes from this intersection: often domain scientists will solve only the computational problems strictly required to answer their own questions, losing the opportunity to reach broad and lasting impact and accelerate discovery in the long run. This is due to understandable and natural constraints on time, effort, software engineering and computational expertise. Conversely, scientific computing is littered with technology developed with overly generic architectural considerations that appears compelling at first, yet ultimately proves ill-suited for extensive real-world usage and ends up abandoned.

Instead, we use the domain questions as drivers of a user-centered software design process, for which the Jupyter (formerly IPython) project has over 15 years of proven experience. Through this process, we will build tools that close the gap between interactive, scientist- and question-driven exploratory computation and the analysis of heterogeneous and rich data at scale. A key contribution here will be the co-design of backend data-access tools in JupyterHub deployed either in the cloud or HPC systems, along with the front-end components in JupyterLab that will make these data accessible interactively. This co-design process enables us to make tradeoffs between

latency and bandwidth necessary to provide interactivity on large-scale data and platforms.

A further contribution is the development of tools ranging from data access and visualization to cluster resource management that are aimed at the domain scientists, reducing the need for specialized data center staff to assist scientists on a number of tasks. The promise of the cloud for science won't be fully realized if working scientists need a lot of dedicated and highly specialized engineering support (typically not available) or if they need to effectively become such engineers themselves (not viable nor a good use of their time for most). Instead, our tools will be easy to adopt by teams with only minimal support thanks to a combination of ease of use, modularity, documentation, and direct community engagement activities we will pursue as part of this project.

The domain questions outlined in Sec 3 not only serve as context for technical developments in this project, they are also drivers for advancing open science in each of those communities. We will produce state-of-the-art demonstrations of open, interactive geoscience research on shared computational infrastructure. These will be accompanied by tutorials, documentation, and workshops that will enable scientists, including the >1000 researchers who have engaged with the Pangeo project already, to adopt the developed tools in their own work.

8 Broader impacts

Pangeo has shown that there is an appetite for adopting open-source technologies and practices for the next generation of geoscience research. These have had rippling effects, resulting in new applications in the earth sciences and beyond. For example, ocean.pangeo.org now serves a JupyterHub for the oceanography community, and there is currently work on a "panneuro" deployment for neuroimaging. This project further develops domain-agnostic tools in the Jupyter ecosystem. The millions of users who currently incorporate Jupyter into their day-to-day work will have access to all advancements made during the course of this project.

Since 2016 Jupyter has played a significant-enough role to be explicitly mentioned by name in the title or the project summary of 31 NSF-funded projects (Award numbers: #1735234, #1615848, #1615001, #1740229, #1739657, #1812786, #1730170, #1550588, #1712282, #1541450, #1712354, #1661497, #1740315, #1550476, #1550475, #1837661, #1550562, #1738975, #1738979, #1639722, #1822351, #1835661, #1616709, #1829622, #1550528, #1835791, #1822336, #1835692, #1835566, #1816388, #1639648). These represent an investment of more than $20 million by the NSF, spanning education initiatives (e.g. #1735234, #1712282), efforts to improve reproducibility (e.g. #1541450), and domain research applications from biology (e.g. #1661497), to chemistry (e.g. #1738975) to earth sciences (e.g. #1740315, #1639722). The outcomes of our proposed research will directly benefit all of these projects and the communities of researchers they serve.

9 Management plan

9.1 Roles of different institutions and investigators

University of California, Berkeley. P.I. Pérez's group (Heagy, Holdgraf, Panda, Pérez + software engineer + postdoc) consists of Jupyter contributors and domain-scientists. They will lead the technical developments within the Jupyter ecosystem (see Sec 4). Heagy (Postdoc with Pérez) will be responsible for the geophysics use-case and will contribute to scientific software development (e.g. SimPEG) with unfunded collaborator Dr. Bedrosian (USGS). P.I. Larsen and Dr. Moges will primarily focus on the Hydrology use-case and will coordinate with Pérez's group to guide software-development decisions and priorities.

National Center for Atmospheric Research. P.I. Hamman is a disciplinary scientist and is a technical contributor to the Pangeo and Jupyter projects. Hamman and a junior software will lead the integration with the climate data analysis subject area and will provide perspective on the needs of the current Pangeo community with unfunded collaborators Dr. Abernathey (Columbia)

and Dr. Signell (USGS). The junior software engineer will integrate developments from this project into HPC and cloud JupyterHub deployments at NCAR and the broader Pangeo project.

All project members will contribute to outreach. This includes contributing to documentation and tutorials for geoscientists looking to adopt these tools in their workflows as well as connecting with geoscientists in their networks and encouraging them to try out new developments and provide feedback.

9.2 Coordination and communication

Technical communication will be conducted on GitHub, which is already where Jupyter, Pangeo, and many related scientific Python communities (e.g. Xarray, Dask, SimPEG) coordinate. GitHub has tools for versioning software, peer-reviewing reviewing changes, tracking issues, and it interfaces to tools for testing and continuous integration; these features make it one of the most widely used platforms for developing and maintaining software. For more informal communication, we will establish a Slack channel. We will also host regular video-conference calls and participate in the regular Pangeo meetings. Finally, we will hold annual face-to-face meetings at UC Berkeley; travel has been budgeted for this. These meetings will provide the opportunity to review the progress of the project, discuss challenges, and prioritize new developments.

9.3 Timeline

In **year 1**, the scientists will focus on incorporating Jupyter throughout their scientific use-cases, contributing to underlying scientific software (e.g. Xarray, SimPEG) as needed. Software developments in JupyterHub (see Sec 4.3 4.4) and JupyterLab (see Sec 4.1, 4.2) in this year will be "prototype"-phase developments. We will focus on iterating on these tools with the team scientists and their close collaborators on small deployments. Throughout, we will generate API and user documentation. At the end of year 1, we will demonstrate the prototyped interface for data catalogs, tools for managing JupyterHubs, and progress on interactive widgets & dashboards at the EarthCube AHM. In **year 2**, we will focus on testing and growing the use of the developed tools. This will include expanding user documentation. In collaboration with the Pangeo project, we will incorporate tools for managing JupyterHubs and running desktop applications on the Cloud and HPC centers through JupyterHub into exsisting deployments. In addition, we will expose the user-interface for accessing data catalogs to Pangeo users. Along with the continued feedback from our project scientists, we will continue to refine these tools with feedback from the wider Pangeo community. At the EarthCube AHM, we will provide a production-scale demonstration. In **year 3**, we will continue refining the developed tools, and will focus on dissemination to the wider geoscience community. We will complete the geoscience use-cases by publishing the interactive workflows and develop geoscience-motivated tutorials to support the documentation. In each of **years 2 & 3**, we will host a webinar and a workshop at the AGU meeting to inform the wider community about the developments and discuss how they can be contribute to the continual improvements of these tools. Final demonstrations will be presented at the EarthCube AHM.

10 Sustainability Plan

This proposal has several strategies for ensuring the sustainability of the tools we build. First, our team has extensive experience working on technical projects with long-lasting and sustainable impact. The flagship tool in the Jupyter Project has seen sustained development and growth in its community of developers and users since 2001. Moreover, Pangeo has seen a steady growth of scientists both using the platform and extending it with new tools.

The key to sustainability for Jupyter, Pangeo, and this proposal is in the modular nature of the tools we build. By building open source tools in partnership with the broader open science community, we can build off of the extensive offerings of the open source community, only build-

ing more specialized tools only when necessary. This keeps tools more well-scoped, technically-simple, and adaptable to new use-cases. As a team with extensive experience in the open source community, we know how to make technical choices in order to ensure that a community can be built around a project, as opposed to optimizing solely for computational elegance or efficiency. This is crucial to building a sustainable open community, and we will apply these principles to all tools built as a part of this project.

Finally, this project is driven by use cases in the geosciences, meaning that any new tool development will be performed in close consideration with scientists working on their domain-specific problems. This ensures that the kinds of tools created as a part of this proposal are primarily useful to the geosciences and the Pangeo community. This will increase the likelihood of adoption of these tools, and will make it easier to build a sustainable developer community around them.

11 Results from Prior NSF Support

Pérez has one concluded NSF award: *BIGDATA: Small: DA: Classification Platform for Novel Scientific Insight on Time-Series Data*, NSF 1251274, 08/01/13-07/31/17. (PI Bloom; Pérez is Co-PI) Intellectual Merit: Implementation of novel and efficient feature extraction algorithms on irregularly sampled time-series data, supporting feature engineering for machine-learning applications in the domain sciences. Broader Impacts: The development of open-source tools implementing these ideas (cesium-ml) as well as a modern web framework to adapt them to various scientific workflows (cesium-web). These tools are now being adapted for use in the Zwicky Transient Facility (ZTF), a next-generation time-domain astronomical survey. Publications: [37, 38]. Research Products: Cesium project [5], core algorithms library [6] and open-source web platform [1]. Pérez is also co-PI on *TRIPODS: Berkeley Institute on the Foundations of Data Analysis*, NSF 1740855, a currently active project with a focus on methodological development for data science.

Larsen holds a CAREER grant (NSF-EAR 1455362; 2015-2020) focused on understanding interactions between organic sediment and vegetation in deltaic environments. Resulting peer-reviewed publications include [30, 36, 31]. This award has resulted in one student dissertation [33] and one senior honors thesis (Nghiem, forthcoming), research experiences for nine undergraduate students, and currently funds one postdoctoral scholar. It produced major new research and teaching infrastructure (the Ecogeomorphology Flume), used as a teaching resource in two of Larsen's undergraduate and graduate classes. Two short-courses were developed through the National Center for Earth Dynamics Summer Institute, and an exhibit was developed for the San Francisco Exploratorium (opening date December 2017).

Hamman has one active NSF award. *Collaborative Proposal: EarthCube Integration: Pangeo: An Open Source Big Data Climate Science Platform*, NSF 1740633, 09/01/2017-08/31/2020. (PI Paul; Hamman is Co-PI). This EarthCube Integration project combines a suite of open-source software tools to form the Pangeo Platform, a tool collection can tackle petabyte-scale ESM datasets. The culmination of the project will be a robust new software toolkit for climate science at scale. Intellectual Merit: Some of the most exciting and ambitious ideas in climate science are currently impossible to realize due to the computational burden of processing petabyte-scale datasets. The Pangeo Platform will enable a new Big Data era in climate science in which disciplinary scientists can realize their most ambitious goals. Broader Impacts: Pangeo's core components are already widely used in the scientific python community, and the resulting development of these tools from our proposed work will greatly benefit this upstream community. Additionally, training and educational materials for these tools will be developed, distributed widely online, and integrated into existing educational curricula. Research Products & Their Availability: The developments in this project have resulted in 13 blog posts, multiple software releases, and 4 hands-on tutorials. The project's main Pangeo Platform has engaged more than 1000 scientists.

References

[1] Cesium web frontend. https://github.com/cesium-ml/cesium_web.

[2] Geojson. http://geojson.org/.

[3] Intake: A general interface for loading data. https://github.com/intake/intake.

[4] Network common data form (netcdf). https://www.unidata.ucar.edu/software/netcdf.

[5] Open-source machine learning for time series analysis. http://cesium.ml/.

[6] Open-source platform for time series inference. https://github.com/cesium-ml/cesium.

[7] Spatiotemporal asset catalog (stac). https://github.com/radiantearth/stac-spec.

[8] Thredds data server (tds). https://www.unidata.ucar.edu/software/thredds/current/tds/.

[9] Virtual network computing. https://en.wikipedia.org/wiki/VNC.

[10] R. Abernathey, N. Henderson, N. Robinson, J. Tomlinson, K. Paul, J. Hamman, J. Zhuang, D. Rothenberg, and M. Rocklin. Interactive comment on "requirements for a global data infrastructure in support of cmip6" by venkatramani balaji et al. *Geoscientific Model Development*, 2018.

[11] Thibaut Astic and Douglas W. Oldenburg. *Petrophysically guided geophysical inversion using a dynamic Gaussian mixture model prior*, pages 2312–2316. 2018.

[12] Esben Auken, Anders Vest Christiansen, Casper Kirkegaard, Gianluca Fiandaca, Cyril Schamper, Ahmad Ali Behroozmand, Andrew Binley, Emil Nielsen, Flemming Effersø, Niels Bøie Christensen, Kurt Sørensen, Nikolaj Foged, and Giulio Vignoli. An overview of a highly versatile forward and stable inverse algorithm for airborne, ground-based and borehole electromagnetic and electric data, 2015.

[13] V. Balaji, K. E. Taylor, M. Juckes, B. N. Lawrence, P. J. Durack, M. Lautenschlager, C. Blanton, L. Cinquini, S. Denvil, M. Elkington, F. Guglielmo, E. Guilyardi, D. Hassell, S. Kharin, S. Kindermann, S. Nikonov, A. Radhakrishnan, M. Stockhause, T. Weigel, and D. Williams. Requirements for a global data infrastructure in support of cmip6. *Geoscientific Model Development*, 11(9):3659–3680, 2018.

[14] Rowan Cockett, Seogi Kang, Lindsey J. Heagy, Adam Pidlisecky, and Douglas W. Oldenburg. SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications. *Computers & Geosciences*, 85:142 – 154, 2015.

[15] Daniele Colombo, Gary McNeice, Diego Rovetta, Ernesto Sandoval-Curiel, Ersan Turkoglu, and Armando Sena. High-resolution velocity modeling by seismic-airborne tem joint inversion: A new perspective for near-surface characterization. *The Leading Edge*, 35(11):977–985, 2016.

[16] Steven C. Constable, Robert L Parker, and Catherine G Constable. Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, 52(3):289–300, mar 1987.

[17] Dashboards Development Team. Jupyter dashboards layout extension, 2018. https://jupyter-dashboards-layout.readthedocs.io.

[18] Dask Development Team. Dask: Library for dynamic task scheduling, 2016. https://dask.org.

[19] V. Eyring, S. Bony, G. A. Meehl, C. A. Senior, B. Stevens, R. J. Stouffer, and K. E. Taylor. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.

[20] GeoSci.xyz contributors, Douglas Oldenburg, Lindsey Heagy, and Seogi Kang. Geosci.xyz: Open source resources for the geosciences. https://geosci.xyz.

[21] Eldad Haber and Uri M Ascher. Fast Finite Volume Simulation of 3D Electromagnetic Problems With Highly Discontinuous Coefficients. *SIAM Journal on Scientific Computing*, 22(6):1943–1961, 2001.

[22] Lindsey J. Heagy, Rowan Cockett, Seogi Kang, Gudni K. Rosenkjaer, and Douglas W. Oldenburg. A framework for simulation and inversion in electromagnetics. *Computers & Geosciences*, 107:1 – 19, 2017.

[23] S. Hoyer and J. Hamman. xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, 5(1), 2017.

[24] Open Source Initiative. The open source definition (annotated), 2019. https://opensource.org/osd-annotated.

[25] Insight Software Consortium. itk-jupyter-widgets: Interactive Jupyter widgets to visualize images in 2D and 3D, 2018. https://github.com/InsightSoftwareConsortium/itk-jupyter-widgets.

[26] Anna Kelbert, Naser Meqbel, Gary D. Egbert, and Kush Tandon. ModEM: A modular system for inversion of electromagnetic geophysical data. *Computers and Geosciences*, 66:40–53, 2014.

[27] Kerry Key and Jeffrey Ovall. A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling. *Geophysical Journal International*, 186(1):137–154, jul 2011.

[28] Inc. Kitware. jupyterlab_geojs: A jupyterlab notebook extension for rendering geospatial data using GeoJS, 2018. https://github.com/OpenGeoscience/jupyterlab_geojs.

[29] Inc. Kitware. pygeojs: ipywidget wrapper for GeoJS, 2018. https://github.com/OpenGeoscience/pygeojs.

[30] L. G. Larsen. Multiscale flow-vegetation-sediment feedbacks in low-gradient landscapes. *Geomorphology*, Minor revisions, 2019.

[31] L. G. Larsen, J. Ma, and D. Kaplan. How important is connectivity for surface-water fluxes? a generalized expression for flow through heterogeneous landscapes. *Geophysical Research Letters*, 44:10349–10358, 2017.

[32] Y Li and K Key. 2D marine controlled-source electromagnetic modeling: Part 1 — An adaptive finite-element algorithm. *GEOPHYSICS*, 72(2):WA51–WA62, 2007.

[33] Hongxu Ma. *Data-driven approaches to resolving feedback processes driving the earth system over multi- spatial and temporal scales*. PhD thesis, University of California, Berkeley, 2019.

[34] D. R. Maidment. Conceptual framework for the national flood interoperability experiment. *Journal of the American Water Resources Association*, 53(2):245 – 257, 2017.

[35] Ryan May, Sean Arms, Patrick Marsh, Eric Bruning, and John Leeman. Metpy: A Python package for meteorological data, 2008 - 2017. https://github.com/Unidata/MetPy.

[36] W. Nardin, L. G. Larsen, S. Fagherazzi, and P. Wiberg. How does vegetation community shape geomorphological evolution? tradeoffs among hydrodynamics, sediment fluxes, and vegetation in the virginia coastal reserve. *Estuarine, Coastal and Shelf Science*, 210:98–108, 2018.

[37] Brett Naul, Joshua S. Bloom, Fernando Pérez, and Stéfan van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2(2):151–155, 2018.

[38] Brett Naul, Stéfan van der Walt, Arien Crellin-Quick, Joshua S. Bloom, and Fernando Pérez. cesium: Open-source platform for time-series inference. *CoRR*, abs/1609.04504, 2016.

[39] Travis E Oliphant. Python for Scientific Computing. *Computing in Science Engineering*, 9(3):10–20, may 2007.

[40] Yuvi Panda. HubPloy: The jupyterhub kubernetes deployer, 2019. https://hubploy.readthedocs.io/en/latest/.

[41] Robert L Parker. The inverse problem of electromagnetic induction: Existence and construction of solutions based on incomplete data. *Journal of Geophysical Research: Solid Earth*, 85(B8):4421–4428, aug 1980.

[42] David W. Pierce. Ncview: a netcdf visual browser, 2016. http://meteora.ucsd.edu/~pierce/ncview_home_page.html.

[43] Project Jupyter Developers. Zero to jupyterhub with kubernetes, 2019. https://zero-to-jupyterhub.readthedocs.io/en/latest/.

[44] QuantStack. Voila: Interactive renderer for Jupyter notebooks, 2017. https://github.com/QuantStack/voila.

[45] R. S. Regan, S. L Marskstrom, L. E. Hay, R. J. Viger, P. A. Norton, J. M. Driscoll, and J. H. LaFontaine. *Description of the National Hydrologic Model for use with the Precipitation-Runoff Modeling System (PRMS)*, chapter B9, page 38 pp. 2018.

[46] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195, 2019.

[47] Victoria Stodden. Enabling reproducible research: Open licensing for scientific innovation. *International Journal of Communications Law and Policy, Forthcoming*, 2009.

[48] Jiajia Sun and Yaoguo Li. Multidomain petrophysically constrained inversion and geology differentiation using guided fuzzy c-means clustering. *GEOPHYSICS*, 80(4):ID1–ID18, 2015.

[49] Jiajia Sun and Yaoguo Li. Joint inversion of multiple geophysical data using guided fuzzy c-means clustering. *Geophysics*, 81(3):ID37–ID57, 2016.

[50] A. N. Tikhonov and V. Y. Arsenin. Solutions of Ill-Posed Problems. *SIAM Review*, 32:1320–1322, 1977.

[51] S van der Walt, S C Colbert, and G Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering*, 13(2):22–30, mar 2011.