Proceedings of the 23rd Python in Science Conference

Proceedings of the 23rd Python in Science Conference

Edited by Meghann Agarwal, Amey Ambade, Chris Calloway, Rowan Cockett, Sanhita Joshi, Charles Lindsey, and Hongsup Shin

SciPy 2024
Tacoma, Washington
July 8 - July 14, 2024

ISSN:2575-9752
https://doi.org/10.25080/DTVR3553

Organization

Conference Chairs

- Alexandre Chabot-Leclerc, *Enthought, Inc.*
- Julie Krugler Hollek, *Mozilla*

Program Chairs

- Paul Ivanov, *Citadel*
- Madicken Munk, *University of Illinois at Urbana-Champaign*
- Guen Prawiroatmodjo, *MotherDuck*
- Matthew Feickert, *University of Wisconsin-Madison*
- Anna Haensch, *Tufts University*

Communications

- Matt Davis, *Open Source Maintainer*
- Juanita Gomez, *University of California, Santa Cruz*
- Cam Gerlach, *Python Core/Spyder*

Birds of a Feather

- Michael Akerman, *Novonesis*
- Mike Droettboom, *Microsoft Corporation*

Proceedings

- Meghann Agarwal, *GDI*
- Amey Ambade, *SLB*
- Chris Calloway, *University of North Carolina*
- Sanhita Joshi, *Deloitte*
- Charles Lindsey, *Aptos*
- Hongsup Shin, *Arm*
- Rowan Cockett, *Curvenote*

Financial Aid

- Scott Collis, *Argonne National Laboratory*
- Eric Ma, *Moderna*
- Nadia Tahiri, *University of Sherbrooke*

Tutorials

- Tetsuo Koyama, *PyVista Community*
- Logan Thomas, *Pattern Bioscience*
- Benoit Hamelin, *Tutte Institute for Mathematics and Computing*
- Inessa Pawson, *Albus Code/OpenTeams*

Sprints

- James Lamb, *NVIDIA*
- Brigitta Sipőcz, *Caltech/IPAC*

Diversity

- Sarah Kaiser, *Microsoft*
- Meekail Zain, *Quantsight*

Activities

- Paul Anzel, *Rev.com*
- Ed Rogers, *Majesco*
- Ana Comesana, *Lawrence Berkeley National Laboratory*

Sponsors/Financial/Logistics

- Jim Weiss, *NumFOCUS*

Hybrid

- David Nicholson, *Independent Researcher*
- Rebecca BurWei, *Mozilla*
- Allen Harvey Jr, *Applied Research Associates, Inc.*
- Neelima Pulagam, *Ford Motor Company*

Proceedings Reviewers

- Abhay Dutt Paroha
- Amadi Gabriel Udu
- Andrei Paleyes
- Andrew James
- Andy Terrel
- Angus Hollands
- Ankur Ankan
- Ashwin Hegde
- Blaine Mooers
- Bobby Jackson
- Chong Shen Ng
- Chuchu Wu
- Cliff Kerr
- Conrad Koziol
- Drew Camron
- Dr. Kirtan Dave
- Franklin Koch
- James Lamb
- Jane Adams
- Jennifer E. Yoon
- John Drake
- Juan Cabanela
- Katelyn FitzGerald
- Katie Wetstone
- Kalyan Prasad
- Kevin Lacaille
- Kuntao Zhao
- Lindsey Heagy
- Marcus Hill
- Matt Craig
- Matthew Feickert
- Mihai Maruseac
- Mike Sarahan

- Nadia Tahiri
- Nate Jacobs
- Nicole Brewer
- Paul Wright
- Pranoy Ray
- Rithwik Tom
- Rohit Goswami
- Stefan van der Walt
- Steve Purves
- Sumit Kumar
- Suzana Șerboi
- Talha Irfan
- Tek Kshetri
- Tetsuo Koyama
- Tolulade Ademisoye
- Veronica Gathoni
- Victoria Adesoba
- Wu-Jung Lee

Posters and Slides

Keynote Presentations

Explainable AI for Climate Science: Opening the Black Box to Reveal Planet Earth

Earth’s climate is chaotic and noisy. Finding usable signals amidst all of the noise can be challenging: be it predicting if it will rain, knowing which direction a hurricane will go, understanding the implications of melting Arctic ice, or detecting the impacts of humans on the earth’s surface. Here, I will demonstrate how explainable artificial intelligence (XAI) techniques can sift through vast amounts of climate data and push the bounds of scientific discovery: allowing scientists to ask “why?” but now with the power of machine learning.

Elizabeth A. Barnes

https://doi.org/10.25080/yhec5334

Particles, People, and Pull Requests

I will tell the story of how the statistical challenges in the search for the Higgs boson and exotic new physics at the Large Hadron Collider led to new approaches to collaborative, open science. The story centers around computational and sociological challenges where software and cyberinfrastructure play a key role. I will highlight a few important changes in perspective that were critical for progress including embracing declarative specifications, pivoting from reproducibility to reuse, and the abstraction that led to the field of simulation-based inference.

Kyle Cranmer

https://doi.org/10.25080/rkcg9834

The Right Tool for the Job

There are many programming languages that we might choose for scientific computing, and we each bring a complex set of preferences and experiences to such a decision. There are significant barriers to learning about other programming languages outside our comfort zone, and seeing another person or community make a different choice can be baffling. In this talk, hear about the costs that arise from exploring or using multiple programming languages, what we can gain by being open to different languages, and how curiosity and interest in other programming languages supports sharing across communities. We’ll explore these three points with practical examples from software built for flexible storage and model deployment, as well as a brand new project for scientific computing.

Julia Silge

https://doi.org/10.25080/xwen4438

Accepted Talks

No-Code-Change GPU Acceleration for Your Pandas and NetworkX Workflows

This talk describes new open-source GPU accelerators from the NVIDIA RAPIDS project for Pandas and NetworkX and will demonstrate how you can enable them for your workflows to experience significant speedups without code changes.

Rick Ratzel, Vyas Ramasubramani

https://doi.org/10.25080/jyef9727

Python for early-stage design of sustainable aviation fuels

We develop a multi-objective, multi-parameter optimization methodology applied to designing novel sustainable aviation fuels

A.M. Martz, A.E. Comesana, V.H. Rapp, K.E. Niemeyer

https://doi.org/10.25080/afjf2467

Introduction to Causal Inference Using pgmpy

In the domain of data science, a significant number of questions are aimed at understanding and quantifying the effects of interventions, such as assessing the efficacy of a vaccine or the impact of price adjustments on the sales volume of a product. Traditional association based methods machine learning methods, predominantly utilized for predictive analytics, prove inadequate for answering these causal questions from observational data, necessitating the use of causal inference methodologies. This talk aims to introduce the audience to the Directed Acyclic Graph (DAG) framework for causal inference. The presentation has two main objectives: firstly, to provide an insight into the types of questions where causal inference methods can be applied; and secondly, to demonstrate a walkthrough of causal analysis on a real dataset, highlighting the various steps of causal analysis and showcasing the use of the pgmpy package.

Ankur Ankan

https://doi.org/10.25080/kvta3223

Coming Online: Enabling Real-Time and AI-Ready Scientific Discovery

A framework for building real-time and AI enabled sensor processing applications

Adam Thompson, Luigi Cruz

https://doi.org/10.25080/juet4542

Expanding the OME ecosystem for imaging data management

OMERO is an open-source solution for image data management which can be customized and hosted by individual institutions, based on the widely used OME data model for microscopy data. Multiple OMERO deployments might be used to provide core delivery, facilitate internal research, or serve as a public data repository. The omero-cli-transfer package facilitates data transfer between these OMERO instances and provides new methods for importing datasets. Another open-source package, ezomero, improves the usability of OMERO in a research environment by providing easier access to OMERO's Python interface. Along with existing OMERO plugins built for other analysis and viewing software, this positions OMERO to be a hub for image storage, analysis, and sharing.

Erick Ratamero

https://doi.org/10.25080/wdya6338

Free, public, standardized Zarr stores of geospatial data in the cloud for all! Now in Beta.

At the NASA Goddard Earth Sciences (GES) Data and Information Services Center (DISC), we're doing the heavy lifting to make large geospatial datasets easily accessible from the cloud. No more downloading data. No more worrying about quirky metadata or missing dimensions. No more concatenating hundreds or thousands of files together. Just fire up your Jupyter notebook somewhere in Amazon Web Services (AWS)'s US-West-2 region, get some free temporary AWS credentials, open our Zarr stores, and start doing your science.

Christine Smit, Hailiang Zhang, Brianna Pagan, Dieu My Nguyen, James Acker, Ashley Heath, Mahabaleshwara Hegde, Long Pham

https://doi.org/10.25080/majr5893

My NumPy year: From no CPython C API experience to shipping a new DType in NumPy 2.0

Support for string data in NumPy has long been a sore spot for the community. At the beginning of 2023 I was given the task to solve that problem by writing a new UTF-8 variable-length string DType leveraging the new NumPy DType API. I will offer my personal narrative of how I accomplished that goal over the course of 2023 and offer my experience as a model for others to take on difficult projects in the scientific python ecosystem, offering tips for how to get help when needed and contribute productively to an established open source community.

Nathan Goldbaum

https://doi.org/10.25080/wdug2226

Introduction to Causal Inference with Machine Learning

Causal inference has traditionally been used in fields such as economics, health studies, and social sciences. In recent years, algorithms combining causal inference and machine learning have been a hot topic. Libraries like EconML and CausalML, for instance, are good Python tools that facilitate the easy execution of causal analysis in areas like economics, human behavior, and marketing. In this talk, I will explain key concepts of causal inference with machine learning, show practical examples, and offer some practical tips. Attendees will learn how to apply machine learning to causal analysis effectively, boosting their research and decision-making.

Hajime Takeda

https://doi.org/10.25080/tufy8474

HyperSpy: Your Multidimensional Data Analysis Toolbox

HyperSpy is a community-developed open-source library providing a framework to facilitate interactive and reproducible analyses of multidimensional datasets. Born out of the electron microscopy scientific community and building on the extensive scientific Python environment, HyperSpy provides tools to efficiently explore, manipulate, and visualize complex datasets of arbitrary dimensionality, including those larger than a system’s memory. After 14 years of development, HyperSpy recently celebrated its 2.0 version release. This presentation (re)introduces HyperSpy’s features and community, with a focus on recent efforts paring the library into a domain-agnostic core and a robust ecosystem of extensions providing specific scientific functionality.

Joshua Taillon

https://doi.org/10.25080/gpna7794

Ibis and interfaces

This talk lays out the current database / data landscape as it relates to the SciPy stack, and explores how Ibis (an open-source, pure Python, dataframe interface library) can help decouple interfaces from engines, to improve both performance and portability.

Gil Forsyth

https://doi.org/10.25080/fdck7886

Using Satellite Imagery to Identify Harmful Algal Blooms and Protect Public Health

This talk illustrates how machine learning models to detect harmful algal blooms from satellite imagery can help water quality managers make informed decisions around public health warnings for lakes and reservoirs. Rooted in the development of the open source package CyFi, this talk includes insights around identifying when your model is getting the right answer for the wrong reasons, the upsides of using decision tree models with satellite imagery, and how to help non-technical users build confidence in machine learning models.

Emily Dorne

https://doi.org/10.25080/ghpx3574

An Introduction to Impact Charts

Impact charts, as implemented in the impactchart package, make it easy to take a data set and visualize the impact of one variable on another in ways that techniques like scatter plots and linear regression can't, especially when there are other variables involved. In this talk, we introduce impact charts, demonstrate how they find easter-egg impacts we embed in synthetic data, show how you can create your first impact chart with just a few lines of code, and show how impact charts can find hidden impacts in a real-world use case.

Darren Vengroff, Ph.D.

https://doi.org/10.25080/tfaj6588

ITK-Wasm: Universal spatial analysis and visualization

How WebAssembly makes scientific computing accessible, sustainable, and reproducible

Matthew McCormick

https://doi.org/10.25080/pghc3745

Making Research Data Flow with Python

Telescopes exist in remote environments, and yet produce huge amounts of data. This presentation is about building the data transfer tool Librarian for the Simons Observatory, which enables seamless shifting between internet-enabled transfers and hand-carrying disks down mountains.

Josh Borrow

https://doi.org/10.25080/ttdf6694

Monte Carlo/Dynamic Code: Performant and Portable High-Performance Computing at Scale via Python and Numba

Monte Carlo / Dynamic Code (MC/DC) is a Monte Carlo neutron transport solver targeting high performance computing. MC/DC is acclerated using the Numba compiler and has the capibility to run on CPUs and GPUs. This talk describes the development of MC/DC

Joanna Piper Morgan, Kyle E. Niemeyer

https://doi.org/10.25080/cdrf9272

Starsim: A flexible framework for agent-based modeling of health and disease

Starsim is an open-source agent-based modeling framework for simulating the spread of diseases among agents via dynamic transmission networks. This talk describes the Starsim package and gives an example of how it can be used to model HIV and syphilis.

Cliff Kerr, Robyn Stuart, Romesh Abeysuriya, Paula Sanz-Leon, Jamie Cohen, Daniel Klein

https://doi.org/10.25080/ukpu4584

anywidget: custom Jupyter Widgets made easy

anywidget simplifies the creation and distribution of Jupyter Widgets by providing a portable and reusable specification and toolset. It ensures cross-platform compatibility with notebook platforms, lowers the barrier to entry, and improves reusability and interoperability in interactive computing environments. This talk highlights the motivation to bring the web and Python ecosystems closer together, showcasing community-driven anywidgets and new widgets that push the boundaries of these platforms.

Trevor Manz

https://doi.org/10.25080/wdhm9848

Pooch: A friend to fetch your data files

Easily download and cache data files from the web.

Santiago Soler

https://doi.org/10.25080/frkj7844

scikit-build-core: A modern build-backend for CPython C/C++/Fortran/Cython extensions

A presentation about scikit-build-core exploring how it modernizes Python extension building by integrating CMake with Python packaging standards, enabling seamless cross-compilation and multi-platform support. The slides highlight key features such as simplified configuration, support for multiple languages like C++, Fortran, and Cython, and the transition from the classic scikit-build. They also provide insights into how scikit-build-core enhances the development experience and streamlines Python module creation for diverse environments.

Jean-Christophe Fillion-Robin, Henry Schreiner, Matt McCormick

https://doi.org/10.25080/xjvg7399

Sparse Arrays in scipy.sparse

The shift from sparse matrices to sparse arrays in SciPy. What’s coming, migration and API decisions.

Dan Schult

https://doi.org/10.25080/ejft5676

Building a modular simulation platform for magnetic resonance force microscopy (MRFM) experiments (mrfmsim and mmodel)

We present mrfmsim, an open-source framework that not only facilitates the design, simulation, and signal validation of magnetic resonance force microscopy experiments, but also significantly speeds up the development process. In the talk, we present the challenges in building simulation packages for experiments undergoing continuous development in a graduate research setting. We show how we designed mrfmsim and its backend mmodel for modularity, extendibility, and readability, and how these design principles translate into practical benefits for researchers in the field.

Peter Sun, John Marohn

https://doi.org/10.25080/xpnp9684

Vector space embeddings and data maps for cyber defense

Cyber defense, and in particular threat detection, requires gaining insight from very large amounts of telemetry data, using unsupervised learning. This talk presents a method for embedding such data in vector spaces and exploring it through interactive visualization and labeling, using *data maps*. We demonstrate this method by looking at the baseline behaviours of hosts in the open ACME3 dataset of host-based telemetry.

Benoit Hamelin, John Healy

https://doi.org/10.25080/uykm3443

Simplifying analysis of hierarchical HDF5 and NetCDF4 files with xarray-datatree

Xarray-datatree, is a Python package that supports HDFs (Hierarchical Data Format) with hierarchical group structures by creating a tree-like hierarchical data structure in xarray.

Eniola Awowale, Tom Nicholas, Lucas Sterzinger, Nick Lenssen

https://doi.org/10.25080/xfex7842

Accepted Posters

Training a Supervised Cilia Segmentation Model from Self-Supervision

Understanding cilia behavior is essential in diagnosing and treating such diseases. But, the tasks of automatically analysing cilia are often a labor and time-intensive since there is a lack of automated segmentation. In this work we overcome this bottleneck by developing a robust, self-supervised framework exploiting the visual similarity of normal and dysfunctional cilia. This framework generates pseudolabels from optical flow motion vectors, which serve as training data for a semi-supervised neural network. Our approach eliminates the need for manual annotations, enabling accurate and efficient segmentation of both motile and immotile cilia.

Seyed Alireza Vaezi, Shannon Quinn

https://doi.org/10.25080/hfew4757

Domovyk: Multilingual Transliteration for Cyrillic Text

The Domovyk package provides transliteration to and from Cyrillic alphabets in a way that addresses some limitations in existing packages, providing multilingual functionality, support for composite Unicode characters, and support for languages not addressed in other packages, such as Church Slavonic and Carpatho-Rusyn. Domovyk aims to increase the accessibility of transliteration technologies for users working in these languages, focusing on use cases that require thorough and accurate transliteration.

Ian Goodale

https://doi.org/10.25080/udkt5322

ncompare: A Python Package for Comparing netCDF Structures

As netCDF (Network Common Data Form) files are widely used in Earth science — with climate models, oceanographic or atmospheric reanalyses, and observational data — improved means of evaluating netCDF files can help enable a wide range of applications. We have developed a reusable open source approach through `ncompare`, which is a Python package for comparing netCDF structures. `ncompare` facilitates rapid comparisons by generating a formatted display of the matching and non-matching groups, variables, and associated metadata between two NetCDF datasets. The user has the option to colorize the terminal output for ease of viewing, and `ncompare` can optionally save comparison reports in text, comma-separated value (CSV), and/or Microsoft Excel formats.

Daniel E. Kaufman, Walter E. Baskin, Julia S. Lowndes

https://doi.org/10.25080/etnj4973

Development and Application of CWGID: the California Wildfire GeoImaging Dataset for Deep Learning Driven Forest Wildfire Detection

This poster presents the development and application of the CWGID (California Wildfire GeoImaging Dataset), a comprehensive dataset for deep learning-driven forest wildfire detection. The study explores the dataset creation process, its application in wildfire detection using deep learning techniques, and the results obtained.

Valeria Martin, K. Brent Venable, Derek Morgan

https://doi.org/10.25080/kgpe7737

Fast and Easy Graph Analytics with the NetworkX Ecosystem of Backends

Poster describing the function dispatching features of NetworkX and the various backends currently available.

Rick Ratzel, Dan Schult

https://doi.org/10.25080/ryga7653

RoughPy: Streaming data is rarely smooth

RoughPy is a library that aims to connect data science with the mathematics of rough paths to provide a new perspective for working with streamed data. The Stream object provided by the library is an abstraction of streamed data so that it can be viewed through the lens of rough path theory. This makes the high order representation of the data (the signature) available as a tool to be used in data science and machine learning applications. This poster outlines the mathematics, the applications and data, and how RoughPy brings both sides together.

Sam Morley

https://doi.org/10.25080/yfnx8796

Mamba Models:A Potential Replacement for Transformers?

Mamba models leverage State Space Models and the HiPPO framework to efficiently handle long-range dependencies, reducing computational complexity compared to traditional transformers.

Suvrakamal Das

https://doi.org/10.25080/txwe5647

Employing the strengths of Generative AI supports the execution of time series analysis and forecasting

This poster explores the use of Generative AI models for time series analysis and forecasting, specifically in the context of energy consumption. It compares traditional statistical methods, such as ARIMA, with advanced AI-based techniques like AutoGluon-TimeSeries, xLSTM, and TimeGPT. The study aims to demonstrate the efficiency and accuracy of these methods using real-world energy data from the PJM Interconnection LLC. Results indicate that AI-based models, particularly xLSTM and AutoGluon-TimeSeries, outperform traditional models in forecasting accuracy, showcasing their potential for better resource management and decision-making in climate change mitigation.

Ying-Jung Chen

https://doi.org/10.25080/xtng2642

Parallel Graph Algorithms and Building Backends with Entry Points

Hi! Have you ever wished your pure Python libraries were faster? Or wanted to fundamentally improve a Python library by rewriting everything in a faster language like C or Rust? Well, wish no more… NetworkX's backend dispatching mechanism redirects your plain old NetworkX function calls to a FASTER implementation present in a separate backend package by leveraging Python's `entry_point` specification! NetworkX is a popular, pure Python library used for graph (aka network) analysis. But when the graph size increases (like a network of everyone in the world), NetworkX algorithms could take days to solve a simple graph analysis problem. To address these performance issues, this backend dispatching mechanism was recently developed. This poster explores NetworkX's parallel backend that utilizes Joblib to run graph algorithms on multiple CPU cores and how we can use it just by specifying a `backend` keyword argument or by passing the backend graph object (type-based dispatching). It also goes over some of the future ToDos for the nx-parallel backend, the speedups obtained, and some important notes to ponder about. Last but not at all the least, it depicts the ideal pipeline starting from networkx, going on to nx-parallel, then to joblib, and then towards the various parallel libraries. (nx-parallel GitHub repo - https://github.com/networkx/nx-parallel) Thank you :)

Aditi Juneja

https://doi.org/10.25080/ypkc2577

Aeromancy: Towards More Reproducible AI and Machine Learning

We present Aeromancy, an opinionated philosophy and open-sourced framework that closely tracks experimental runtime environments for more reproducible machine learning. In existing experiment trackers, it's easy to miss important details about how an experiment was run, e.g., which version of a dataset was used as input or the exact versions of library dependencies. Missing these details can make replicability more difficult. Aeromancy aims to make this process smoother by providing both new infrastructure (a more comprehensive versioning scheme including both system runtimes and external datasets) and a corresponding set of best practices to ensure experiments are maximally trackable.

David McClosky

https://doi.org/10.25080/yyvd5799

Leveraging FAIR principles for efficient management of meteorological radar data

Radars are crucial in meteorology for their precise spatio-temporal resolution, enabling early detection and tracking of severe weather. This capability aids meteorologists in issuing timely alerts, thus safeguarding lives and reducing property damage. Radar data also supports offline applications like cloud and precipitation analysis, climatology, and insurance risk assessment, all relying on its time-series nature. However, storing radar data traditionally involves proprietary formats with high I/O demands, leading to slow computations and resource-intensive requirements.

To address these challenges, a new data model is proposed using the CF format-based FM301 hierarchical tree structure and ARCO formats. This model efficiently organizes radar data into cloud-storage buckets using Python libraries like Xarray, Xradar, Wradlib, and Zarr. Demonstrated with Carimagua, Colombia radar data, the model shows faster processing times than legacy methods on standard hardware. Emphasizing FAIR principles (Findable, Accessible, Interoperable, Reusable), this approach enhances accessibility to radar data on cloud platforms, promoting open science and wider societal benefit.

Alfonso Ladino, Maxwell Grover, Stephen Nesbitt, Kai Mühlbauer

https://doi.org/10.25080/wnaf9823

Building Quantum Bridges: Advancing Drug Discovery with QAOA and Explaining Quantum Computing with Building Blocks

Using quantum computers to solve combinatorial optimization problems

B. Maurice Benson

https://doi.org/10.25080/mtdn2862

Building sustainability and community in a small project: lessons from working on SaltProc

SaltProc is an open source tool for simulating batch-wise reprocessing of fuel in nuclear reactors developed around an export controlled depedency. This limited the size of the userbase. I contributed features to SaltProc to support an open-source alternative, and found that this change attracted new users.

Oleksandr R. Yardas, Madicken Munk

https://doi.org/10.25080/ercj5799

Open Source Farm to Open Science Table: Project Pythia’s Cook-off Hackathons

This poster describes Project Pythia’s annual community hackathons, aka Cook-offs. These summer sprints blur the lines between scientist and software developer at an individual and group level, and seed excitement and commitment to the open source, open science community.

M. Drew Camron, Kevin Tyle

https://doi.org/10.25080/rrdv2565

Accelerating the use of Lagrangian data with Clouddrift

clouddrift is a python library built to accelerate and simplify the use of lagrangian datasets in science. To achieve this the library adapts datasets into cloud optimized ragged arrays, provides analysis and query methods to work with lagrangian datasets as ragged arrays.

Santana, Kevin, Elipot, Shane, Miron, Philippe, Curcic, Milan

https://doi.org/10.25080/ftyc2662

Geist: a multimodal data transformation, query, and reporting language

Geist is a new templating language for declarative data manipulation, query, and report generation. Building on the Jinja template engine, Geist is designed to support diverse data backends and query engines via predefined tags and filters, and may be extended with custom tags. A single Geist template may include multiple queries expressed in different languages, e.g. SQL and SPARQL, to leverage the strengths of each for clarity and ease of maintenance. Because Geist both can generate reports in diverse formats and perform inserts and updates on new or existing databases during template expansion, Geist templates may orchestrate data extraction, transformation, and load operations spanning multiple tools and data storage systems. Geist also enables modularity in query languages and eliminates messy procedural programs. Geist aims to enable developers to use whatever language or tools they like regardless of where the data is stored.

Meng Li, Timothy McPhillips, Bertram Ludäscher

https://doi.org/10.25080/geaj2635

zfit: scalable pythonic likelihood fitting

zfit is a highly scalable and customizable model manipulation and likelihood fitting library. It uses the same computational backend as TensorFlow and is optimised for simple and direct manipulation of probability density functions.

Jonas Eschle, Albert Puig Navarro, Rafael Silva Coutinho, Matthieu Marinangeli, Nicola Serra, Iason Krommydas

https://doi.org/10.25080/xwwd2556

Convolutional Autoencoders for Denoising Solar Images

Using scientific packages in Python, we trained convolutional autoencoders to improve the quality of solar images taken from the Atmospheric Imaging Array (AIA).

Jimmy Lynch

https://doi.org/10.25080/fpju4745

Facilitating scientific investigations from long-tail data with Python

This presentation walks through our work on creating a non-nanosecond datetime for Pandas and the development of Python toolboxes to query databases and analyze datasets using Pandas objects for interoperability.

Deborah Khider, Varun Ratnakar, Julien Emile-Geay, Kim Pevey, Marco Gorelli, Nicholas McKay, Alexander James, Jordan Landers

https://doi.org/10.25080/jfcn6576

Climatic and Geographic Influences on Cumacea Genetics in the Northern North Atlantic

Cumacea crustaceans serve as vital indicators of benthic health in marine ecosystems. This study investigates the influence of environmental parameters on their genetic makeup in the North Atlantic, focusing on Icelandic waters. We analyze mitochondrial 16S rRNA gene sequences from 62 Cumacea specimens collected across varying depths.

Justin Gagnon, Nadia Tahiri

https://doi.org/10.25080/eage2654

From concept to compute: accelerating research with workflow management in `pyiron`

`pyiron_workflow` is a python framework for developing research workflows based on composing individual function nodes together into computational graphs. Each node class can be defined by simply applying a decorator to a regular python function, allowing user-developers to extend `pyiron` functionality even without deep knowledge of Object-Oriented Programming. Nodes can be grouped together into macro nodes using the same simple function-and-decorator approach, such that complex workflows can be built up and simply represented by composing and nesting these graphs. In contrast to user workflows being defined via a series of Jupyter notebook cells, this approach rigorously defines workflows by their graph topology and allows them to be easily shared – and incorporated into new contexts – by sharing/importing the workflow as a macro node. Interoperability can be controlled with (optional) type checking on data connections between nodes.

Liam Huber, Joerg Neugebauer

https://doi.org/10.25080/xwwj3999

3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)

VTK implements an object-oriented approach to 3D visualization, and PyVista adheres to that underlying structure to provide an API that expands on VTK's data types. These expanded, wrapped types hold methods and attributes for quickly accessing scalar arrays, inspecting properties of the dataset, or using filtering algorithms to transform datasets. PyVista wrapped objects have a suite of common filters ready for immediate use directly on the objects. These filters are commonly used algorithms in the VTK library that have been made more accessible by binding a method to control that algorithm directly onto all PyVista datasets, providing a shared set of functionality. Through the use of these bound filtering methods, powerful VTK algorithms can be leveraged and controlled via keyword arguments designed to be intuitive for novice users.

Tetsuo Koyama

https://doi.org/10.25080/vxvf4964

Vectorized Quadrature, Series Summation, Differentiation, Optimization, and Rootfinding in SciPy

Until recently, SciPy's best options for scalar quadrature, minimization, and root finding called compiled code, which could not take advantage of a vectorized Python integrand, objective function, or residual function; SciPy offered no functions for accurate numerical differentiation or series summation. These gaps are being filled with a family of pure-Python, array API compatible functions for dramatically faster vectorized calculation of scalar integrals, infinite sums, derivatives, minimizers, and roots.

Matt Haberland, Albert Steppi, Pamphile T. Roy

https://doi.org/10.25080/uyyk2727

Spyder and the NumFOCUS SDG program: better UI/UX, improved code completion and lessons learned

Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. Thanks to the NumFOCUS Small Development Grants (SDG) program, from late 2022 through the beginning of 2024 we've made many improvements to Spyder's UI/UX and to its support for external code completion plugins.

We’d like to share the work we’ve done, and explain the ideas and execution behind two of our proposals, from both a technical and project management perspective.

Daniel Althviz Moré, Juan Sebastian Bautista Rojas

https://doi.org/10.25080/gdhp7664

Scientific Publishing with MyST Markdown

Supercharge your scientific writing with MyST Markdown! MyST is designed for technical communication and research publication. Sprinkling in all the typesetting power that you’d get from LaTeX but with the pleasure of lightweight content focussed writing that you get from Markdown, create interactive papers or documentation that link with data, have embedded computation and provide explorable notebooks to your readers. Get your work out to a templated PDF for traditional publication from exactly the same material, all from the command line or with continuous integration. In this talk, we’ll show how MyST works, how to get started and some examples of awesome MyST based publications.

Steve Purves, Rowan Cockett

https://doi.org/10.25080/yrmn7235

gravitational lensing simulations made user friendly with Caustics’ three interface levels

We present Caustics, a tool to accelerate the analysis of gravitational lensing systems for the next generation of astronomical data. Caustics will enable precision measurements of dark matter properties, the expansion rate of the Universe, lensed black holes, the first stars, and more. In this talk I will the benefits and challenges of how we used PyTorch (a differentiable and GPU accelerated scientific python package) to allow for fast development without sacrificing numerical performance. I will detail our development process as well as how we encourage users of all skill levels to engage with our documentation/tools.

Connor Stone

https://doi.org/10.25080/gwcm5548

Seamless integration across developer ecosystems with python and wasm in VTK

Visualizing large-scale simulation datasets is vital for research in scientific and engineering fields. To this end, VTK, an open-source C++ visualization library, and its Python counterparts VTK Python, PyVista and Vedo are streamlining 3D visualization, though shortcomings remain. We significantly improved VTK’s automated wrappers and enabled VTK WebAssembly thus simplifying creation of visualizations in Python or on the web using Trame. This also made it easier to execute VTK native code directly in a web browser. We discuss challenges and lessons learned in extending large C++ software to diverse languages through API wrapping and cross-compilation to ultimately benefit the Python community.

Jaswant Panchumarti, Sebastien Jourdain, Berk Geveci, Aashish Chaudhary

https://doi.org/10.25080/fyvp8946

Venturial: Generating CFD Workflows in Python

Venturial is an open-source suite of interactive Python tools for Computational Fluid Dynamics (CFD). Venturial, envisions to benefit the scientific python community by providing an easy-to-understand CFD application building workflow within the Python environment.

Rajdeep Adak, Janani Srree Murallidharan, Prabhu Ramachandran https://doi.org/10.25080/tpwg2365

SciPy Tools Plenaries

SciPy Tools Plenary on the Journal of Open Source Software (JOSS)

Updates from the Journal of Open Source Software (JOSS) on new work and improvements to the journal in 2023 and 2024.

Matthew Feickert

https://doi.org/10.25080/pcna4769

SciPy Tools Plenary on Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. This presentation summarizes changes over the past year, new features, and future plans.

Elliott Sales de Andrade

https://doi.org/10.25080/guwd9846

NumPy annual update

Comprehensive overview of the latest release and major milestones for the NumPy library and its contributor community

Inessa Pawson

https://doi.org/10.25080/dpha2486

SciPy: not just a conference

Besides a conference, ‘SciPy’ is also a library that provides fundamental building blocks for modeling and solving scientific problems. SciPy includes algorithms for optimization, interpolation, statistics, fast Fourier transforms, and many other classes of problems; it also provides specialized data structures, such as k-dimensional trees and sparse matrices. This presentation summarizes the current status and future plans of the SciPy library.

Matt Haberland

https://doi.org/10.25080/mtvj6294

Sphinx-gallery and Sphinx-tags

Sphinx-gallery and sphinx-tags features and updates

Hannah Aizenman

https://doi.org/10.25080/ddee5226

Lightning Talks

Deploying Python environments on top of Mt. Rainier

A process for building preparing artifacts that enable the deployment of a Python computing environment in a place where there is no Internet access.

Benoit Hamelin

https://doi.org/10.25080/pyyu4582

Hello Project!

tips for onboarding into contributing to open source

Hannah Aizenman

https://doi.org/10.25080/hxxk4957

swiftascmaps: A colour map library for swifties

A hopefully comedic talk about the colour map library swiftascmaps.

Josh Borrow

https://doi.org/10.25080/uncn2995

Sciris: Simplifying scientific Python

Sciris aims to streamline the development of scientific Python code by making it easier to perform common tasks. This example illustrates how the same block of fairly typical scientific Python code – which performs tasks like collecting data from a function running in parallel, saving and loading files, and 3D plotting – looks like when written in ‘vanilla Python’ compared to using Sciris.

Cliff Kerr, Paula Sanz-Leon, Romesh Abeysuriya

https://doi.org/10.25080/knjj9332

Plotting Slides in Matplotlib

Matplotlib makes easy things easy and hard things possible, like this silly idea of making slides in it.

Elliott Sales de Andrade

https://doi.org/10.25080/wrvp6756

For the python evangelist

Because clearly python needs to make music.

Christine Smit

https://doi.org/10.25080/hcya9443

Renovate: Automating Dependency Management

A brief introduction to Renovate, a tool for automating dependency management in software projects. The slides can be accessed at https://paddyroddy.github.io/talks/renovate-automating-dependency-management. Two example parent configurations I maintain are available at https://github.com/paddyroddy/.github/tree/main/renovate and https://github.com/UCL-ARC/.github/tree/main/renovate.

Patrick J. Roddy

https://doi.org/10.25080/tkky5633

Sponsored Students

Scholarship Recipients

- Amadi Gabriel Udu, *University of Leicester*
- Ankur Ankan, *Radboud University*
- Ayush Nag, *University of Washington*
- C.A.M. Gerlach, *University of Alabama Huntsville*
- Elliot Salisbury, *University of Southampton*
- Erick Martins Ratamero, *The Jackson Laboratory*
- JT Thielen, *Colorado State University*
- Mohamed El Shorbagy, *Ain Shams University*
- Sheku Shafeie, *TechPoint*
- Tek Kshetri, *University of Calgary*
- Willy Menacho, *Universidad Técnica Federico Santa María*
- Ying-Jung Chen, *University of Washington eScience*

NumFOCUS Diversity Scholarship Recipients

- Atharva Rasane, *DaSouk / GDG Belgaum*
- Hannah Aizenman, *City College of New York*
- Jacqui Levy, *Environment and Climate Change Canada*
- Juanita Gomez, *University of California Santa Cruz*
- Juliana Ferreira Alves, *Itaú Unibanco*
- Meng Li, *Rice University*
- Tetsuo Koyama, *ARK Information Systems*

Table of Contents