## Title

PyARPES: An analysis framework for multimodal angle-resolved photoemission spectroscopies



## Authors

Stansbury, Conrad
Lanzara, Alessandra

## Publication Date

2020

## DOI

10.1016/j.softx.2020.100472

Peer reviewed

SoftwareX 11 (2020) 100472

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

# PyARPES: An analysis framework for multimodal angle-resolved photoemission spectroscopies

Conrad Stansbury *, Alessandra Lanzara

*Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
*Department of Physics, University of California, Berkeley, CA 94720, USA*

ARTICLE INFO

*Article history:*
Received 24 April 2019
Received in revised form 29 February 2020
Accepted 1 April 2020

*Keywords:*
ARPES
NanoARPES
Pump-probe ARPES
Photoemission
Python
Qt
Jupyter

ABSTRACT

The advent of higher resolution and throughput photoemission spectroscopy experiments has made angle-resolved photoemission spectroscopy (ARPES) a critical tool for the study of quantum materials. The simultaneous development of novel ARPES techniques, including nano/$\mu$-ARPES, spin-resolved ARPES, and pump-probe ARPES mirrors the expansion in scanning modes for scanning tunneling microscopy, which made scanning probe methods the gold standard for driving insights into surface physics. In this paper, we introduce PyARPES, an open source and modular data analysis framework for angle-resolved photoemission spectroscopies. We highlight PyARPES' current capabilities for the analysis of the large photoemission datasets created by nano-ARPES and pump-probe ARPES experiments, show how PyARPES fulfills current data analysis needs for ARPES analysis, and discuss prospects for the future of analysis techniques enabled by PyARPES.



## Code metadata

| | | |
|---|---|---|
| C2 | Permanent link to code/repository used for this code version | https://github.com/ElsevierSoftwareX/SOFTX_2019_160 |
| C3 | Legal Code License | GPLv3 |
| C4 | Code versioning system used | git |
| C5 | Software code languages, tools, and services used | Python, JavaScript, Qt5 |
| C6 | Compilation requirements, operating environments & dependencies | H5, NetCDF4, SciPy, NumPy, xarray, Description of Dependencies |
| C7 | If available Link to developer documentation/manual | https://arpes.netlify.com/ |
| C8 | Support email for questions | chstan@berkeley.edu, GitLab Issues |

## Software metadata

| | | |
|---|---|---|
| S1 | Current software version | 2.3.0 |
| S2 | Permanent link to executables of this version | PyPI and Anaconda Cloud |
| S3 | Legal Software License | GPLv3 |
| S4 | Computing platforms/Operating Systems | Linux, OS X, Microsoft Windows, Unix-like, Jupyter/web-based |
| S5 | Installation requirements & dependencies | Python $\geq$ 3.5, conda or pip |
| S6 | If available, link to user manual - if formally published include a reference to the publication in the reference list | https://arpes.netlify.com/ |
| S7 | Support email for questions | chstan@berkeley.edu, GitLab Issues |

* Corresponding author at: Department of Physics, University of California, Berkeley, CA 94720, USA.
*E-mail address:* chstan@berkeley.edu (C. Stansbury).
https://doi.org/10.1016/j.softx.2020.100472

## 1. Motivation and significance

The development of bright, laser-based photoemission sources, improvements in the flux of synchrotron X-ray sources, and rapid improvements in the resolving power of hemispherical electron analyzers in the 1990s and 2000s have made angle resolved photoemission spectroscopy (ARPES) a quantitative spectroscopy, rather than just a tool for band mapping. This development enabled novel insights into the material properties of graphene [1–3] the cuprate and pnictide families of high-temperature superconductors [4–6]. More recently ARPES has been brought to bear on the physics of low dimensional materials, surfaces, and interfaces [7–9] and emerging quantum materials [10]. Recent trends indicate a second consequence of improvements in ARPES spectrometers and light sources: they also contributed to the development of new ARPES techniques, where the angle-resolved photocurrent can be further resolved by the electron spin (SARPES) [11–13], spatially in a scanning beam setup with sub-micron resolution (nano-ARPES) [14], or temporally in pump-probe configuration to study material properties out of equilibrium (Tr-ARPES) [15–18].

Modern ARPES datasets are higher dimensional and typically sparser than would be desirable for band mapping applications, a consequence of the much greater number of experimental degrees of freedom. This problem is only exacerbated by limited instrumental availability, short surface lifetimes in imperfect vacuums, and the damaging effects of the X-ray and ultraviolet radiation required by the photoemission process. A final complication associated with the use and interpretation of the large datasets created by state-of-the-art ARPES experiments is that to date, there have been only limited efforts to develop open and modular data analysis frameworks convenient for high dimensional photoemission data. Despite this, the neighboring disciplines of scanning probe microscopies [19–21], super-resolution microscopy [22], and X-ray science [23] have demonstrated the viability and usefulness of this approach in advancing the scientific impacts of their respective techniques, and signal an opportunity for further insight from ARPES at the nanoscale and sub-picosecond.

## 2. Software description

In an effort to meet the opportunity provided by these large new datasets, we have developed PyARPES, a Python data analysis framework structured around typical photoemission spectroscopy analyses and workflows. PyARPES is distributed as open source, and relies heavily on the PyData software stack as a backbone, allowing integration into a large variety of cross-discipline analysis tools. These tools include robust image processing and basic machine learning support provided by scikit-image and PyOpenCV [24]. Open source and standard tools for data science, numerics, and plotting are used internally by PyARPES to make working in the environment familiar and productive. These dependencies include SciPy, numpy, xarray, pandas, and Matplotlib [25,26]. With the advent of density functional theory packages and physics libraries in Python, including the Atomic Simulation Environment and Python Materials Genomics [27,28], PyARPES offers a data analysis toolkit for experimentalists complementing those for theory and simulation. PyARPES is more than just a collection of analysis routines however, it is also a platform to perform interactive data exploration and analysis with minimal friction, enabling the sophisticated analysis of photoemission data during data acquisition which can guide and constrain ongoing experiments.

A few guiding constraints in the purpose and requirements of PyARPES are helpful in considering its structure and flexibility. Providing a platform with enough depth to perform state-of-the-art photoemission analyses requires confronting that many photoemission experiments combine measurements at synchrotrons scattered globally, at lab-based laser and discharge lamp sources, and at facilities providing auxiliary measurements such as X-ray diffraction, low energy electron microscopy, low energy electron diffraction, or atomic force microscopy and scanning tunneling microscopy.

In addition to a uniform data model, the most important design constraint in the development of PyARPES have been made around satisfying the conflicting needs for rapid and interactive data exploration, and that for the careful analyses and corrections required to manipulate ARPES data into a format appropriate for scientific insight. Where possible, PyARPES has attempted to merge these goals, by allowing interactive tool development, parameter exploration, and data visualization that can be used either to explore data or to explore the behavior of a new analysis routine; both topics will be explored in more depth to follow. As a rule of thumb throughout, modular coupling of functionality and flexible interactive tools is preferred over a full user interface (an approach explored elsewhere [23]), as the latter model places too many restrictions on practical use.

### 2.1. Software architecture

PyARPES is structured as a loosely coupled collection of Python modules comprising common ARPES analysis techniques, providing primitives for building data analysis routines based on volumetric data selection and iteration, and creating data visualizations (see Fig. 1). These decoupled modules are linked by a common data model for spectroscopic data, built on top of the xarray library [26] and providing an object-oriented interface both to raw photoemission data and associated metadata attached by the data acquisition software or experimenter. In addition to independent modules for analysis and data preparation, PyARPES contains an extensible data ingestion package, a spectral function simulation package, a package for designing and exporting DAQ sequences to guide ARPES experiments, and a large number of interactive applications that can be run inside the Jupyter analysis environment, as interactive Matplotlib visualizations, or natively on a desktop as Qt applications.

### 2.2. Data model

Data intake and normalization are provided via a system of plugins, allowing for the extension of PyARPES to additional sources, facilities, and experimental techniques and modes. By normalizing all data and metadata to a conventional format PyARPES, provides a convenient and consistent experience at analysis-time regardless of the original source of data or type of experimental apparatus.

This extends even to different experiments including X-ray backscattering, where PyARPES supports the data formats used by the NorthStar Laue backscattering data acquisition software. At the time of publication, most photoemission facilities at the Advanced Light Source are supported, with some support for other facilities and NeXus, an emerging data standard for X-ray science [29]. Data interchange between PyARPES and Igor Pro–a popular analysis framework inside the photoemission community–is possible directly. Data formats based on the Igor Pro binary format, such as those produced by Scienta Omicron electron spectrometers, can be loaded directly in this way. Additionally, PyARPES can directly load many packed experiment files and Igor Pro binary wave files for analysis. Plugins are being added for additional photoemission facilities as they are required or contributed by users of the software, with the intention that PyARPES will eventually provide a uniform interface exposing all relevant experimental data and information regardless of where data originated.

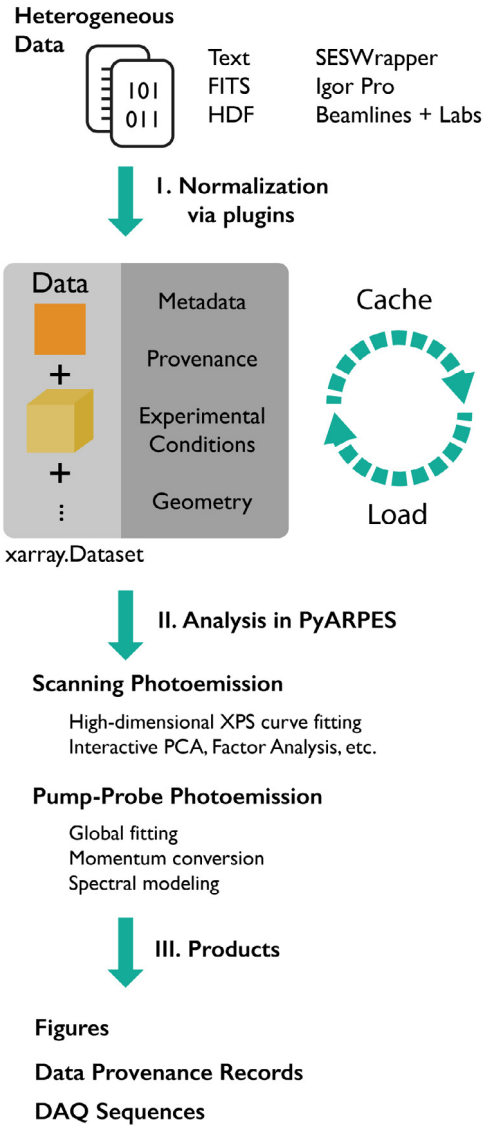


**Fig. 1.** Overview of the data analysis process in PyARPES showing (I) the data loading and supported formats, (II) selected examples of analysis facilities for nano-ARPES and pump-probe ARPES data and (III) work products that PyARPES can generate.

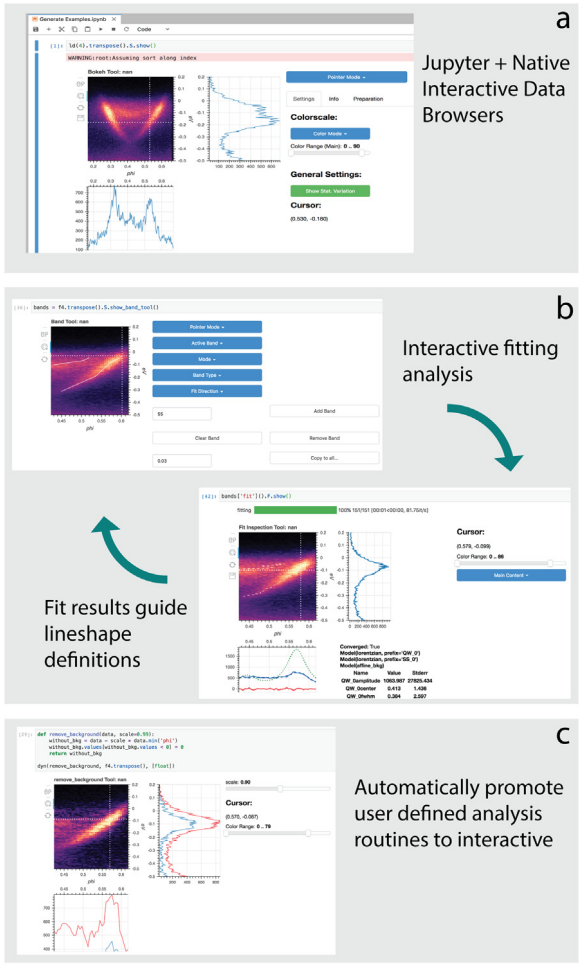


**Fig. 2.** Screenshots of selected interactive analysis and data exploration tools. (a) A data browser for 2D, 3D, and 4D data. Projected marginals at the cursor location are shown. A native browser built on PyQt5 is also provided. (b) A pair of interactive curve-fitting tools allowing the experimenter to lay approximate lineshapes for fitting, and to interactively view the results of multidimensional curve-fitting. (c) PyARPES also features the ability to turn any user defined analysis function into an interactive tool, with standard widgets for viewing data like color saturation, and providing UI elements for all analysis parameters. Here, a background subtraction method is tested on a $Bi_2Se_3$ spectrum.

### 2.2.1. *PyARPES metadata*

The most important metadata associated to ARPES data are the coordinates that annotate the recorded spectrum. Regardless of data dimensionality, PyARPES associates two energy coordinates, the binding energy and the photon energy, three spatial coordinates of the sample manipulator, and six angular coordinates to each piece of data loaded in the software. Three angular coordinates each are used to record the sample goniometer position $(\theta, \beta, \chi)$ and the cut or solid angle observed by the analyzer $(\phi, \psi, \alpha)$ in order to provide a consistent experience across hemispherical electron analyzers and time-of-flight or energy filter systems which observe a finite solid angle. These coordinates are normalized from whatever conventional form they take as recorded during the experiment into a consistent and uniform format in PyARPES. This allows PyARPES to guarantee that any piece of data loaded through the supported plugins will immediately support conversion to momentum space, and is compatible with the analysis builtins.

PyARPES also provides a programmatic interface to all the relevant metadata recorded by the data acquisition software or annotated by the experimenter. In all cases this includes the DAQ settings. Principal among these are the fine configuration of the electron analyzer such as the pass energy, center energy, slit information, and lens table, if they are provided. The acquisition and integration mode, as well as the used aberration correction strategies are also recorded. Essential experimental details such as the sample and cryostat temperatures and information about the sample are also present.

Specialized ARPES disciplines carry with them other important associated metadata required for analysis. In pump-probe studies, PyARPES provides access to the pump and probe wavelength, pulse energies, fluence, full polarization information, bandwidth, and temporal duration. In this way, the metadata model in PyARPES provides an analysis-time complement to storage formats such as NeXus currently under the process of adoption at several synchrotrons.

### 2.2.2. *Data provenance*

This data model allows the analysis modules in PyARPES to address concerns about the reproducibility of scientific analysis.

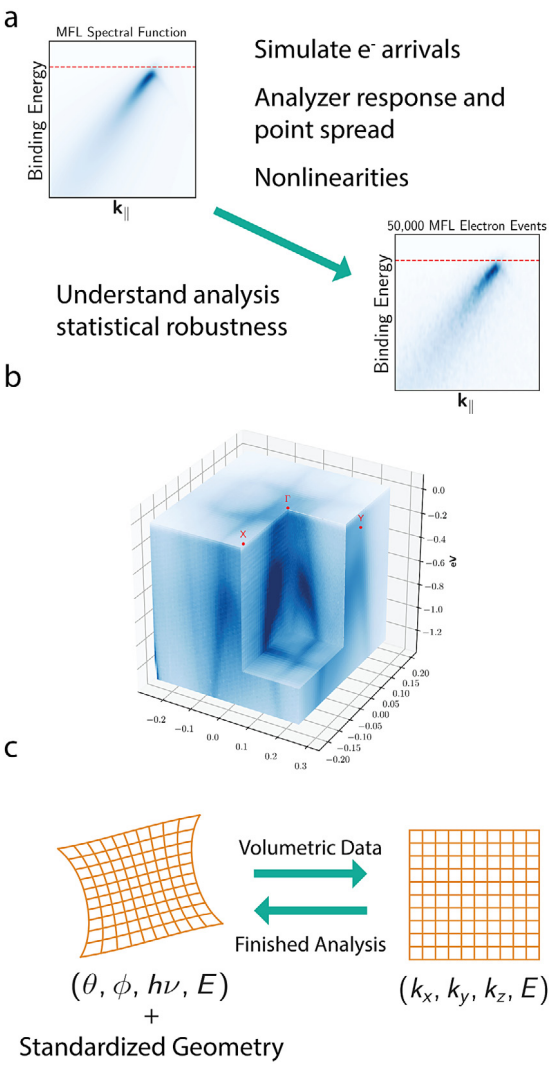


**Fig. 3.** Selected examples of feature from PyARPES: (a) PyARPES provides support for modeling ARPES spectra from theoretical or phenomenological models of the self energy, including by simulating electron events and detector point spread and non-linearity. (b) A variety of common and sophisticated plotting schemes for ARPES data, including cut surfaces through volumetric data (shown), are supported. (c) PyARPES provides an automated coordinate transformation scheme, based on the geometry defined by plugins in the data ingestion system. A uniform API is provided regardless of detector type or geometry. Coordinate meshes indicate the inverse transform coordinate warping over a roughly $\pi/3$ range for the detector geometry inverted in Eqs. (5) and (6) with $\chi = \pi/12$.

Using the built-in analysis routines in PyARPES creates an attached record of data provenance, comprising linked sequences of data and operations that are recorded in JSON format with every plot produced using the PyARPES plotting modules or using the provided wrapper for Matplotlib's IO functions. Together with an experimenter's analysis scripts or Jupyter notebooks, the PyARPES data provenance provides a robust analysis record for ARPES. The default provenance format includes an English language description of the changes performed to the data, the calling function name, and the value of scalar function arguments are recorded in the record. In PyARPES, array-like data is labeled internally by a UUID format ID which is used in place of the full data's value in the data provenance record. Two serious concerns must be addressed when discussing tracking data provenance in a dynamic language like Python, however: code can change and data is mutable.

PyARPES addresses the issue of changes to the analysis code by annotating every element of the data provenance record with the software version in use at the time of analysis and the time of the change.

While the second problem cannot be addressed at a fundamental level in a language like Python, PyARPES offers strong support for users who wish to use data provenance. Essentially all of the data analysis facilities in PyARPES are provenance aware, and will update data provenance appropriately. More importantly, PyARPES offers an API for appending to the data provenance record together with a pair of Python decorators that can be used to transform analysis functions and plotting functions into ones that automatically update the data provenance record without further user intervention. Like the rest of the metadata offered by PyARPES, programmatic access to the lineage of a piece of data is made available through the metadata interface.

### 2.3. Interactive analysis tools

PyARPES contains many interactive analysis tools for masking, data integration, sub-selection along paths and volumes, data slicing, and data exploration among others (Fig. 2). Interactivity is available by way of Bokeh applications hosted inside the Jupyter work environment for Python and native applications that can be launched from any interpreter, built with Qt5 and PyQtGraph. PyARPES is built on the premise that interactive is not only an essential part of the analysis but also of the development of analysis tools, and so furthermore contains a module providing the ability to make essentially any analysis procedure interactive. This allows the experimenter to explore the available parameter space, behavior, and constraints of a new analysis technique, and was used extensively in the internal development of PyARPES.

### 2.4. Figure creation

A final high-level component of the PyARPES analysis framework is a library of figure creation primitives and plotting utilities that supplement popular plotting libraries and provide tailored support for photoemission data analyses (example in Fig. 3(b)).

## 3. Software functionalities

### 3.1. Momentum space conversion

One of the principal physical insights that ARPES provides is direct access into the momentum-resolved electronic structure. In all but hard X-ray experiments the photon involved in the photoemission process carries much less momentum than the photoelectron and the momentum transfer can be neglected. As a result one can directly relate the incident photon frequency $\nu$, the material work function $W$, and the photoelectron's parameters (the kinetic energy $E_\mathrm{k}$ and emission angle $\zeta$) to the energy and parallel crystal momentum.

$$|E_B| = h\nu - W - E_\mathrm{k} \tag{1}$$

$$\hbar\mathbf{k}_\parallel = \sqrt{2mE_\mathrm{k}}\sin\zeta \tag{2}$$

In terms of the physics, this gives a total definition of the problem at hand. For an experimenter though, the situation is more complicated: while these equations can be directly solved in the case of a single spectrum or energy distribution curve at a single point, calculating the emission angle is not always straightforward because of the number of degrees of freedom on the six-axis sample goniometers which have become the standard for photoemission. Even worse, essentially all photoemission

experiments now record very high dimensional datasets either due to capable analyzers, or by scanning goniometer angles. As a result, to transform evenly-spaced volumetric data in angle to evenly-spaced data in momentum, it is in fact the inverse transforms from momentum to angle that must be used in order to leverage grid interpolation schemes.

In the most general case, a rotatable hemispherical analyzer with deflectors or a rotatable angle-resolved time-of-flight spectrometer records the photoelectron velocity with three angles: $\phi$, the angle along the analyzer slit in the case of hemispheres, $\psi$ the angle perpendicular to a hemispherical analyzer's slit, and $\alpha$ a rotation angle about the spectrometer axis. In our convention, horizontal slit hemispherical analyzers measure always with $\alpha = 0$, while vertical slit analyzers measure with $\alpha = \pi/2$. The recorded velocity $\mathbf{v}_a$ is given through

$$\mathbf{v}_a = \begin{bmatrix} \cos\alpha\cos\psi\sin\phi - \sin\alpha\sin\psi \\ \sin\alpha\cos\psi\sin\phi + \cos\alpha\sin\psi \\ \cos\phi\cos\psi \end{bmatrix} \tag{3}$$

Ultimately, we need the velocity in the sample coordinate system, as these are the ones that can be related to the crystal momentum in the same manner as Eq. (2). A six-axis ARPES goniometer implements three rotations, one rotation $\mathrm{R}\left(-\chi, \hat{\mathbf{z}}\right)$ by an angle $\chi$ about the sample normal, another perpendicular to the cryostat axis by an angle $\beta$, and a final rotation around the cryostat axis by an angle $\theta$. Depending on the design of the manipulator, the order of the $\beta$ and $\theta$ rotations may be reversed. Finally, rotations from the cryostat to sample normal coordinates must be performed if they arise due to unintended or intentional offsets of the crystal and cryostat normal vectors. Typically though these are small and may be absorbed into the above angles by small angle approximation.

$$\mathbf{v}_s = \mathrm{R}\left(\chi, \hat{\mathbf{z}}\right)\mathrm{R}\left(\beta, \hat{\mathbf{x}}\right)\mathrm{R}\left(\theta, \hat{\mathbf{y}}\right)\mathbf{v}_a \tag{4}$$

Finally, depending on the coordinates underlying the recorded ARPES data, these equations must be inverted from the appropriate velocities to the scanned analyzer and sample angles with care taken to branch cuts. As an example, this can be done directly after small angle approximation in the case of a hemispherical analyzer which has perpendicular electron deflectors with $\alpha = 0$:

$$\phi - \theta \approx \arcsin\left(\frac{\left(\mathrm{R}\left(\chi, \hat{\mathbf{z}}\right)\right)\cdot\hat{\mathbf{x}}}{\sqrt{1 - \left(\left(\mathrm{R}\left(\chi, \hat{\mathbf{z}}\right)\hat{\mathbf{k}}\right)\cdot\hat{\mathbf{y}}\right)^2}}\right) \tag{5}$$

$$\psi - \beta \approx \arcsin\left(\left(\mathrm{R}\left(\chi, \hat{\mathbf{z}}\right)\hat{\mathbf{k}}\right)\cdot\hat{\mathbf{y}}\right) \tag{6}$$

The inverse transforms can be calculated for the remainder of the geometries, using one or both in-plane equations for fixed photon energy and using the out-of-plane expression when the incident photon energy is varied. So long as the hardware for a particular experiment does not change dramatically, these equations once calculated can be applied directly to data.

#### 3.1.1. Support for momentum conversions in PyARPES

By leveraging information provided by data ingestion plugins on the experimental geometry of each facility, PyARPES automates the process of defining and inverting the appropriate coordinate transforms. In practice, a user needs only provide the coordinate offset in a dataset corresponding to normal emission from the sample, after which PyARPES can coordinate convert the entire data set to momentum, including appropriate choices for the resolution in momentum space.

In addition to volumetric coordinate change via interpolation, PyARPES supports forward coordinate conversion of one-dimensional data, as might be produced by a curve fitting analysis on a single ARPES cut, to momentum space, allowing flexibility on whether data analysis is performed in momentum-space, or in angle-space before being ultimately converted (Fig. 3(c)). This facility is especially advantageous in order to avoid artifacts and resolution loss inherent in the interpolation of gridded data between coordinate representations.

### 3.2. Data corrections + preparation

Quantitative interpretation of ARPES data requires assessment of the experimental apparatus on the data collected. PyARPES contains implementations of semi-automated corrections and data preparation tasks that make these quantitative analyses simpler and less error-prone. These include symmetrizing data across an axis, merging multiple datasets with common coordinates, applying corrections to the chemical potential across the spectrometer window or as a result of miscalibration of beam energy, normalizing data by the recorded angle-integrated photocurrent, background estimation, removal of the trapezoidal edge of an image recorded on a hemispherical analyzer and aberrations, and converting time-of-flight data from flight time to electron kinetic energy.

### 3.3. Common analyses

PyARPES implements the second derivative, image curvature, and minimum gradient [30] methods for band visualization, in addition to wrappers for common filters and data smoothing tools such as the Savitzky-Golay filter. Modules are also included for the analysis of band dispersion created by curve-fitting. To analyze Fermi surface structures and low energy phenomena, a module provides facilities to unwrap and select datasets around paths that follow the Fermi momentum across the Fermi surface. Meanwhile, robust support for Spin-ARPES data together with common Spin-ARPES plots is provided.

Finally, because curve-fitting of lineshapes is an essential component both of understanding the chemical information available in nano-XPS and also in understanding gap and valence band dynamics for pump-probe ARPES, interactive and tools are provided as part of PyARPES both for defining lineshapes that can be used in curve-fitting, and for inspecting the results of multi-dimensional curve-fitting analyses (Fig. 2). Once lineshapes are defined, PyARPES can perform curve fitting conveniently and automatically over large and high dimensional datasets, allowing for efficient analysis of large amounts of photoemission data.

### 3.4. Pump-probe ARPES

The proliferation of pump-probe or ultrafast angle-resolved photoemission (Tr-ARPES) in the 2000s opened a new scientific frontier for ARPES as a probe of the low energy physics in quantum materials. It was quickly realized that extracting meaningful information from Tr-ARPES data requires a higher degree of care to avoid experimental pitfalls including deconvolution of the detector response and incident ultraviolet linewidth, detector response calibrations, corrections to the chemical potential, space charge [31], and experimental nonlinearities [32]. As the community moves towards novel detector and light source schemes, including pump probe experiments at free electron lasers, these challenges will only intensify further [33,34]. As ideas for the study of Tr-ARPES data emerged, it also became apparent that tools for studying gap dynamics and size in superconductors and charge density wave materials produced conflicting results. This

contraindicates their use without comparison to other methods and, ideally, a study of their performance on data drawn from theoretical models [35].

PyARPES provides most standard tools for understanding materials out of equilibrium, including the sophisticated interactive curve fitting analysis tools already described, appropriate and composable lineshape definitions built on top of the nonlinear optimization package LMFIT [36], and facilities to apply common data corrections. Furthermore, PyARPES provides a growing library to simulate realistic photoemission spectra including detector response, point spread, and finite statistics from definitions of the self-energy and bare band dispersions (demonstrated for a marginal Fermi liquid in Fig. 3(a)).

Statistics support exists in PyARPES and allows for bootstrapping analysis procedures that consume volumetric data. For appropriate applications, these tools allow for the attachment of statistically meaningful uncertainties on quantities that in ARPES analyses have generally only been conservatively estimated.

## 4. Sample code snippets analysis

The PyARPES documentation site contains a growing library of examples, illustrating how to perform the most common analyses on a variety of photoemission datasets.

## 5. Illustrative example — Quasiparticle dynamics in a cuprate superconductor

PyARPES comprises different data analysis tools and techniques, so an example analysis will only touch on a few aspects of the software. In the current example, PyARPES is used to load and collate a photoemission dataset produced using a Phoibos 150 electron analyzer for a variety of pump fluences and several locations on the Fermi surface of an optimally doped cuprate superconductor.

The data selection utilities inside PyARPES and xarray are used to integrate the photoemission signal above the chemical potential, revealing the hot quasiparticle dynamics in the sample. To obtain the region for analysis, the Fermi angle $\phi_{\mathrm{d}}(\mu)$ is identified by fitting an equal-energy portion of the photocurrent $\mathrm{I}(\phi, E_B = \mu)$ to a Lorentzian lineshape (Eq. (7)) at the chemical potential.

In subsequent analysis, the total scattering rate given by $1/\tau$ is extracted across the dataset by fitting the excited quasiparticle population, normalized by the equilibrium population, to an exponential decay (Eq. (8)). Estimates of the statistical uncertainty are made by resampling the data across independent observations of the photoemission signal and collected from the same sample. The full example is included as a supplementary notebook, with input dataset structure and results indicated in Fig. 4.

$$\mathrm{I}(\phi, E_B = \mu) = \frac{\mathrm{I}_0 \Delta\phi_{\mathrm{d}}}{[\phi - \phi_{\mathrm{d}}(\mu)]^2 + [\Delta\phi_{\mathrm{d}}]^2} + \phi \mathrm{I}_{\mathrm{bkg}} + \mathrm{I}_{\mathrm{const}} \tag{7}$$

$$\mathrm{I}(t) = \Theta\,(t - t_0)\,\mathrm{I}_0 \exp\left(\frac{t_0 - t}{\tau}\right) + \mathrm{I}_{\mathrm{const}} \tag{8}$$

## 6. Impact

Seamless and immediate access to data enables the experimenter both to develop varied analysis approaches and to make informed decisions during the course of an experiment. With the release of PyARPES we aim to improve the state of angle-resolved photoemission on both accounts. Because of the consistent data interface and plugins, it allows an experimenter to apply preferred preparation and analysis routines while conducting experiments at synchrotron light sources, allowing interactive assessment of data quality, statistics, and experimental geometry before valuable time is lost.

Later at analysis time, in leveraging simple APIs to common ARPES corrections and analysis routines, it offers to the experimenter a friendly work environment that encourages best practices in the analysis of a variety of photoemission datasets. By building on and providing tight integration with scientific computing tools in Python, PyARPES enables rapid development and trial of new analysis techniques that would otherwise be prohibitively expensive to implement in environments with more meager scientific computing support. This structure also allows for its use in cooperation with existing theory and simulation codes without shuttling data between separate environments and formats.

While PyARPES was developed with the ARPES community in mind, its structure, utilities, and plugins for data ingestion make PyARPES appropriate for any chemical or physical spectroscopy, providing fertile ground for the development of analysis tools in synchrotron-based science and optical spectroscopies.

## 7. Conclusions and future directions

Through PyARPES, scientists practicing ARPES are provided simple access to their data, tools to explore it, and sophisticated analysis and modeling modules. However, PyARPES is by no means a finished product. Future releases will expand the scope of the analysis capabilities of the software and refine the experience of data analysis conducted inside PyARPES. In order to cover existing photoemission apparatus, more robust support for multidimensional time-of-flight detectors will eventually be needed. At the same time, broader support for data formats at other synchrotron sources will help fulfill the premise of universal access to data across instruments while the community moves to adopt common data formats. As appropriate practices solidify around applications for machine learning in ARPES, PyARPES hopes to make available inferential tools for chemical analysis of nano-XPS, transient band dynamics of pump-probe ARPES, and background estimation and data preparation procedures.

## Acknowledgments

The authors acknowledge the support of the ARPES group at the Advanced Light Source. We are grateful especially for fruitful discussions with Eli Rotenberg and Jonathan Denlinger.

We also thank the members of the Lanzara group for their feedback and patience on early versions of the software. Daniel Eilbott–a user of the software in the Lanzara group–also contributed the deconvolution code and a spectral function simulation.

This work was supported by the Director, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division, of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231, as part of the Ultrafast Materials Science Program (KC2203).

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

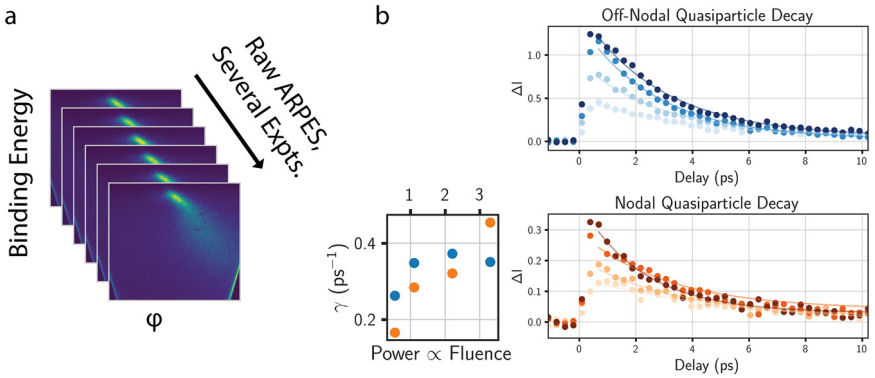


**Fig. 4.** Example tr-ARPES dataset and results from simultaneous analysis of collated data. (a) Illustration of Tr-ARPES dataset comprising many files, requiring collation and treatment. (b) Results of a simple analysis (supplementary notebook) of the quasiparticle dynamics in an optimally doped cuprate superconductor. For two locations on the Fermi surface, the fluence dependent spectral weight above the chemical potential is resolved, and a scattering rate is extracted from an exponential decay model.

## References


[1] Zhou SY, Gweon G-H, Fedorov AV, First PN, de Heer WA, Lee D-H, et al. Substrate-induced bandgap opening in epitaxial graphene. Nature Mater 2007;6(10):770–5. http://dx.doi.org/10.1038/nmat2003, URL https://www.nature.com/articles/nmat2003.

[2] Sprinkle M, Siegel D, Hu Y, Hicks J, Tejeda A, Taleb-Ibrahimi A, et al. First direct observation of a nearly ideal graphene band structure. Phys Rev Lett 2009;103(22). 226803. http://dx.doi.org/10.1103/PhysRevLett.103.226803, URL https://link.aps.org/doi/10.1103/PhysRevLett.103.226803.

[3] Bostwick A, Ohta T, Seyller T, Horn K, Rotenberg E. Quasiparticle dynamics in graphene. Nat Phys 2007;3(1):36–40. http://dx.doi.org/10.1038/nphys477, URL https://www.nature.com/articles/nphys477.

[4] Kaminski A, Fretwell HM. On the extraction of the self-energy from angle-resolved photoemission spectroscopy. New J Phys 2005;7:98. http://dx.doi.org/10.1088/1367-2630/7/1/098.

[5] Lanzara A, Bogdanov PV, Zhou XJ, Kellar SA, Feng DL, Lu ED, et al. Evidence for ubiquitous strong electron–phonon coupling in high-temperature superconductors. Nature 2001;412(6846):510. http://dx.doi.org/10.1038/35087518, URL https://www.nature.com/articles/35087518.

[6] Devereaux TP, Cuk T, Shen Z-X, Nagaosa N. Anisotropic electron-phonon interaction in the cuprates. Phys Rev Lett 2004;93. 117004. http://dx.doi.org/10.1103/PhysRevLett.93.117004, URL https://link.aps.org/doi/10.1103/PhysRevLett.93.117004.

[7] Wang Z, McKeown Walker S, Tamai A, Wang Y, Ristic Z, Bruno FY, et al. Tailoring the nature and strength of electron–phonon interactions in the $SrTiO_3$(001) 2D electron liquid. Nature Mater 2016;15(8):835–9. http://dx.doi.org/10.1038/nmat4623, URL https://www.nature.com/articles/nmat4623.

[8] Chen C, Avila J, Frantzeskakis E, Levy A, Asensio MC. Observation of a two-dimensional liquid of Fröhlich polarons at the bare $SrTiO_3$ surface. Nature Commun 2015;6:8585. http://dx.doi.org/10.1038/ncomms9585, URL https://www.nature.com/articles/ncomms9585.

[9] Arango YC, Huang L, Chen C, Avila J, Asensio MC, Grützmacher D, et al. Quantum transport and nano angle-resolved photoemission spectroscopy on the topological surface states of single $Sb_2Te_3$ nanowires. Sci Rep 2016;6. 29493. http://dx.doi.org/10.1038/srep29493, URL https://www.nature.com/articles/srep29493.

[10] Lu D, Vishik IM, Yi M, Chen Y, Moore RG, Shen Z-X. Angle-resolved photoemission studies of quantum materials. Annu Rev Condens Matter Phys 2012;3(1):129–67. http://dx.doi.org/10.1146/annurev-conmatphys-020911-125027, URL https://doi.org/10.1146/annurev-conmatphys-020911-125027.

[11] Jozwiak C, Park C-H, Gotlieb K, Hwang C, Lee D-H, Louie SG, et al. Photoelectron spin-flipping and texture manipulation in a topological insulator. Nat Phys 2013;9(5):293–8. http://dx.doi.org/10.1038/nphys2572, URL https://www.nature.com/articles/nphys2572.

[12] Jozwiak C, Sobota JA, Gotlieb K, Kemper AF, Rotundu CR, Birgeneau RJ, et al. Spin-polarized surface resonances accompanying topological surface state formation. Nature Commun 2016;7. 13143. http://dx.doi.org/10.1038/ncomms13143, URL https://www.nature.com/articles/ncomms13143.

[13] Santander-Syro AF, Fortuna F, Bareille C, Rödel TC, Landolt G, Plumb NC, et al. Giant spin splitting of the two-dimensional electron gas at the surface of $SrTiO_3$. Nature Mater 2014;13(12):1085–90. http://dx.doi.org/10.1038/nmat4107, URL https://www.nature.com/articles/nmat4107.

[14] Rotenberg E, Bostwick A. MicroARPES and nanoARPES at diffraction-limited light sources: opportunities and performance gains. J Synchrotron Radiat 2014;21(5):1048–56. http://dx.doi.org/10.1107/S1600577514015409, URL http://scripts.iucr.org/cgi-bin/paper?S1600577514015409.

[15] Graf J, Jozwiak C, Smallwood CL, Eisaki H, Kaindl RA, Lee D-H, et al. Nodal quasiparticle meltdown in ultrahigh-resolution pump–probe angle-resolved photoemission. Nat Phys 2011;7(10):805–9. http://dx.doi.org/10.1038/nphys2027, URL https://www.nature.com/articles/nphys2027.

[16] Smallwood CL, Kaindl RA, Lanzara A. Ultrafast angle-resolved photoemission spectroscopy of quantum materials. Europhys Lett 2016;115(2). 27001. http://dx.doi.org/10.1209/0295-5075/115/27001.

[17] Cilento F, Manzoni G, Sterzi A, Peli S, Ronchi A, Crepaldi A, et al. Dynamics of correlation-frozen antinodal quasiparticles in superconducting cuprates. Sci Adv 2018;4(2):eaar1998. http://dx.doi.org/10.1126/sciadv.aar1998, URL https://advances.sciencemag.org/content/4/2/eaar1998.

[18] Boschini F, Neto EHdS, Razzoli E, Zonno M, Peli S, Day RP, et al. Collapse of superconductivity in cuprates via ultrafast quenching of phase coherence. Nature Mater 2018;17(5):416–20. http://dx.doi.org/10.1038/s41563-018-0045-1, URL http://arxiv.org/abs/1707.02305.

[19] Horcas I, Fernández R, Gómez-Rodrí guez JM, Colchero J, Gómez-Herrero J, Baro AM. WSXM: A software for scanning probe microscopy and a tool for nanotechnology. Rev Sci Instrum 2007;78(1). 013705. http://dx.doi.org/10.1063/1.2432410, URL http://aip.scitation.org/doi/10.1063/1.2432410.

[20] Zahl P, Bierkandt M, Schröder S, Klust A. The flexible and modern open source scanning probe microscopy software package GXSM. Rev Sci Instrum 2003;74(3):1222–7. http://dx.doi.org/10.1063/1.1540718, URL http://aip.scitation.org/doi/10.1063/1.1540718.

[21] Nečas D, Klapetek P. Gwyddion: an open-source software for SPM data analysis. Cent Eur J Phys 2012;10(1):181–8. http://dx.doi.org/10.2478/s11534-011-0096-2, URL https://doi.org/10.2478/s11534-011-0096-2.

[22] Laine RF, Tosheva KL, Gustafsson N, Gray RDM, Almada P, Albrecht D, et al. NanoJ: a high-performance open-source super-resolution microscopy toolbox. J Phys D: Appl Phys 2019;52(16). 163001. http://dx.doi.org/10.1088/1361-6463/ab0261.

[23] Basham M, Filik J, Wharmby MT, Chang PCY, El Kassaby B, Gerring M, et al. Data analysis workbench (DAWN). J Synchrotron Radiat 2015;22(3):853–8. http://dx.doi.org/10.1107/S1600577515002283, URL http://scripts.iucr.org/cgi-bin/paper?S1600577515002283.

[24] Walt Svd, Schönberger JL, Nunez-Iglesias J, Boulogne Fc, Warner JD, Yager N, et al. Scikit-image: image processing in Python. PeerJ 2014;2. e453. http://dx.doi.org/10.7717/peerj.453, URL https://peerj.com/articles/453.

[25] Walt Svd, Colbert SC, Varoquaux G. The NumPy array: A structure for efficient numerical computation. Comput Sci Eng 2011;13(2):22–30. http://dx.doi.org/10.1109/MCSE.2011.37, URL https://ieeexplore.ieee.org/document/5725236.

[26] Hoyer S, Hamman J. Xarray: N-D labeled arrays and datasets in Python. J Open Res Softw 2017;5(1):10. http://dx.doi.org/10.5334/jors.148, URL http://openresearchsoftware.metajnl.com/articles/10.5334/jors.148/.

[27] Larsen AH, Mortensen JJ, Blomqvist J, Castelli IE, Christensen R, an MD, et al. The atomic simulation environment—a Python library for working with atoms. J Phys: Condens Matter 2017;29(27). 273002, URL http://stacks.iop.org/0953-8984/29/i=27/a=273002.

[28] Ong SP, Richards WD, Jain A, Hautier G, Kocher M, Cholia S, et al. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. Comput Mater Sci 2013;68:314–9. http://dx.doi.org/10.1016/j.commatsci.2012.10.028, URL http://www.sciencedirect.com/science/article/pii/S0927025612006295.

[29] Könnecke M, Akeroyd FA, Bernstein HJ, Brewster AS, Campbell SI, Clausen B, et al. The NeXus data format. J Appl Crystallogr 2015;48(1):301–5. http://dx.doi.org/10.1107/S1600576714027575, URL http://scripts.iucr.org/cgi-bin/paper?po5029.

[30] He Y, Wang Y, Shen Z-X. Visualizing dispersive features in 2D image via minimum gradient method. Rev Sci Instrum 2017;88(7). 073903. http://dx.doi.org/10.1063/1.4993919, URL https://aip.scitation.org/doi/10.1063/1.4993919.

[31] Graf J, Hellmann S, Jozwiak C, Smallwood CL, Hussain Z, Kaindl RA, et al. Vacuum space charge effect in laser-based solid-state photoemission spectroscopy. J Appl Phys 2010;107(1). 014912. http://dx.doi.org/10.1063/1.3273487, URL https://aip.scitation.org/doi/10.1063/1.3273487.

[32] Reber TJ, Plumb NC, Waugh JA, Dessau DS. Effects, determination, and correction of count rate nonlinearity in multi-channel analog electron detectors. Rev Sci Instrum 2014;85(4). 043907. http://dx.doi.org/10.1063/1.4870283, URL http://aip.scitation.org/doi/10.1063/1.4870283.

[33] Verna A, Greco G, Lollobrigida V, Offi F, Stefani G. Space-charge effects in high-energy photoemission. J Electron Spectrosc Relat Phenom 2016;209:14–25. http://dx.doi.org/10.1016/j.elspec.2016.03.001, URL http://www.sciencedirect.com/science/article/pii/S0368204816300147.

[34] Lollobrigida V, Greco G, Simeone D, Offi F, Verna A, Stefani G. Electron trajectory simulations of time-of-flight spectrometers for core level high-energy photoelectron spectroscopy at pulsed X-ray sources. J Electron Spectrosc Relat Phenom 2015;205:98–105. http://dx.doi.org/10.1016/j.elspec.2015.09.005, URL http://www.sciencedirect.com/science/article/pii/S0368204815002455.

[35] Kordyuk AA, Borisenko SV, Knupfer M, Fink J. Measuring the gap in angle-resolved photoemission experiments on cuprates. Phys Rev B 2003;67(6). http://dx.doi.org/10.1103/PhysRevB.67.064504, URL https://link.aps.org/doi/10.1103/PhysRevB.67.064504.

[36] Newville M, Stensitzki T, Allen DB, Ingargiola A. LMFIT: Non-linear least-square minimization and curve-fitting for Python. Zenodo; 2014, http://dx.doi.org/10.5281/zenodo.11813, URL https://zenodo.org/record/11813#.XMAHnpNKj8s.