Quantum Metrology with Photoelectrons Vol. 3 *Analysis methodologies*

Paul Hockett

Nov 23, 2022

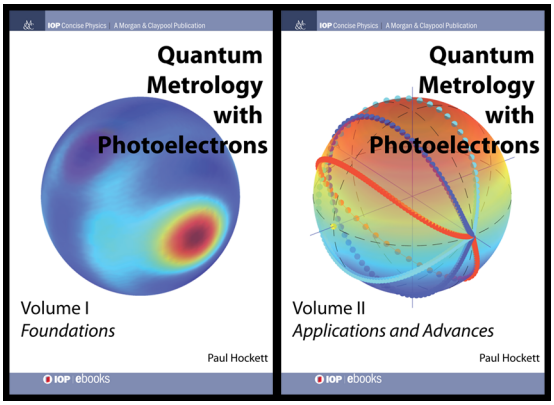
CONTENTS

Quantum Metrology with Photoelectrons Volume 3: *Analysis methodologies*, an open source executable book. This repository contains the source documents (mainly Jupyter Notebooks in Python) and notes for the book, as of Jan 2022 writing is in progress, and the current HTML build can be found online. The book is due to be finished in 2023, and will be published by IOP Press - see below for more details.

Series abstract

Photoionization is an interferometric process, in which multiple paths can contribute to the final continuum photoelectron wavefunction. At the simplest level, interferences between different final angular momentum states are manifest in the energy and angle resolved photoelectron spectra: metrology schemes making use of these interferograms are thus phase-sensitive, and provide a powerful route to detailed understanding of photoionization. In these cases, the continuum wavefunction (and underlying scattering dynamics) can be characterised. At a more complex level, such measurements can also provide a powerful probe for other processes of interest, leading to a more general class of quantum metrology built on phase-sensitive photoelectron imaging. Since the turn of the century, the increasing availability of photoelectron imaging experiments, along with the increasing sophistication of experimental techniques, and the availability of computational resources for analysis and numerics, has allowed for significant developments in such photoelectron metrology.

About the books



- Volume I covers the core physics of photoionization, including a range of computational examples. The material is presented as both reference and tutorial, and should appeal to readers of all levels. ISBN 978-1-6817-4684-5, http://iopscience.iop.org/book/978-1-6817-4684-5 (IOP Press, 2018)
- Volume II explores applications, and the development of quantum metrology schemes based on photoelectron measurements. The material is more technical, and will appeal more to the specialist reader. ISBN 978-1-6817-4688-3, http://iopscience.iop.org/book/978-1-6817-4688-3 (IOP Press, 2018)

Additional online resources for Vols. I & II can be found on OSF and Github.

- Volume III in the series will continue this exploration, with a focus on numerical analysis techniques, forging a closer link between experimental and theoretical results, and making the methodologies discussed directly accessible via new software. The book is due for publication by IOP due in 2023; this volume is also open-source, with a live HTML version at https://phockett.github.io/Quantum-Metrology-with-Photoelectrons-Vol3/ and source available at https://github.com/phockett/Quantum-Metrology-with-Photoelectrons-Vol3.

For some additional details and motivations (including topical video), see the ePSdata project.

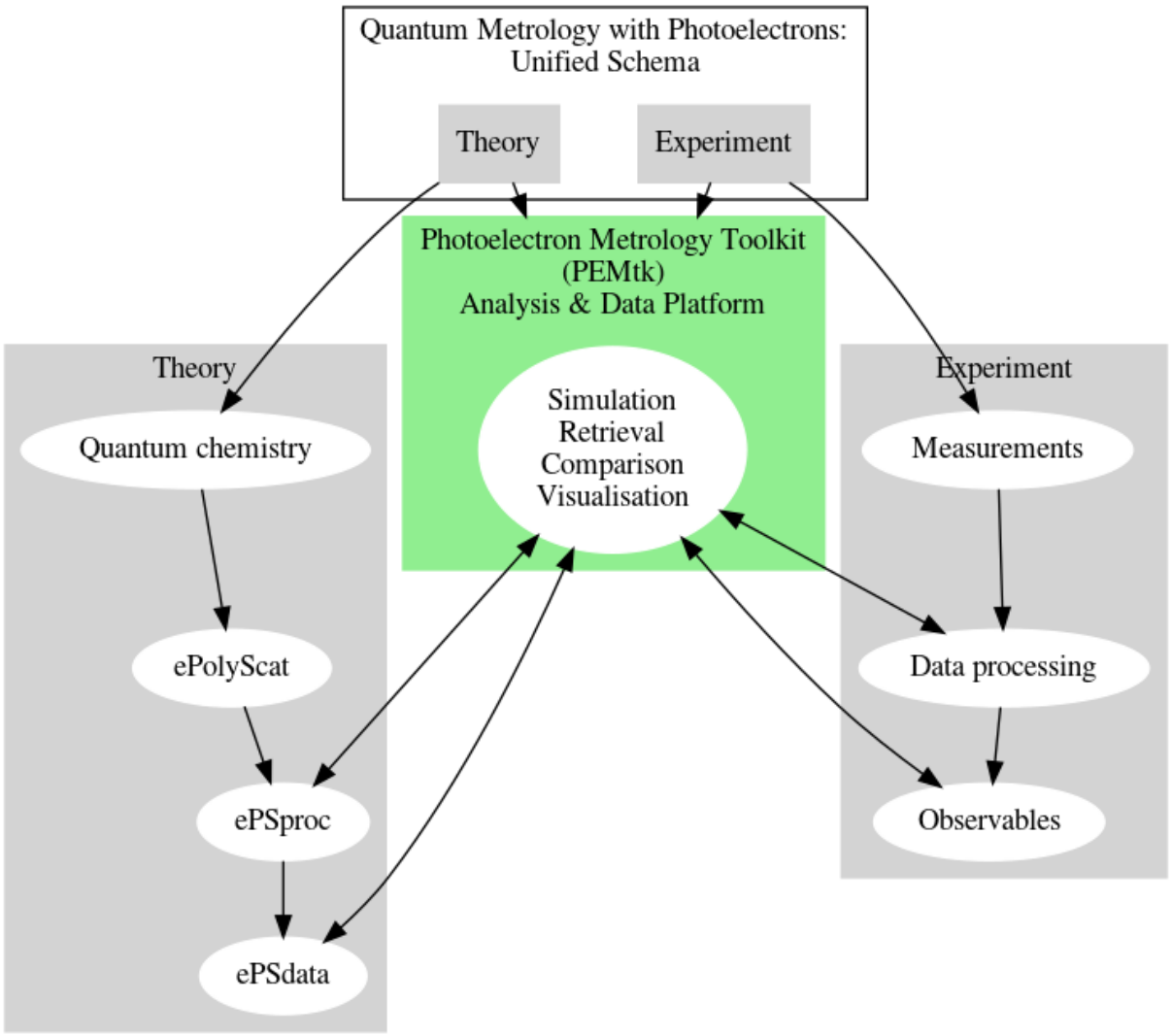
Technical details

This repository contains:

- `doc-source`: the source documents (mainly Jupyter Notebooks in Python)
- `notes`: additional notes for the book,
- the `gh-pages` branch contains the current HTML build, also available at https://phockett.github.io/Quantum-Metrology-with-Photoelectrons-Vol3/

The project has been setup to use the Jupyter Book build-chain (which uses Sphinx on the back-end) to generate HTML and Latex outputs for publication from source Jupyter notebooks & markdown files.

The work *within* the book will make use of the Photoelectron Metrology Toolkit platform for working with experimental & theoretical data.



Running code examples

Each Jupyter notebook (`*.ipynb`) can be treated as a stand-alone computational document. These can be run/used/modified independently with an appropriately setup python environment (details to follow).

Building the book

The full book can also be built from source:

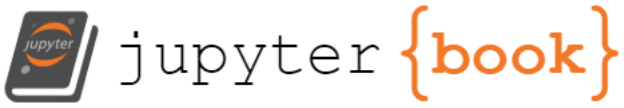
1. Clone this repository
2. Run `pip install -r requirements.txt` (it is recommended you do this within a virtual environment)
3. (Optional) Edit the books source files located in the `doc-source/` directory
4. Run `jupyter-book clean doc-source/` to remove any existing builds
5. For an HTML build:
    - Run `jupyter-book build doc-source/`
    - A fully-rendered HTML version of the book will be built in `doc-source/_build/html/`.
6. For a LaTex & PDF build:
    - Run `jupyter-book build doc-source/ --builder pdflatex`
    - A fully-rendered HTML version of the book will be built in `doc-source/_build/latex/`.

See https://jupyterbook.org/basics/building/index.html for more information.

Credits

This project is created using the open source Jupyter Book project and the executablebooks/cookiecutter-jupyter-book template.

To add: build env & main software packages (see automation for this…)



Part I

Frontmatter

CHAPTER

ONE

OVERVIEW

1.1 General overview

Vol. 3. will focus on analysis techniques for quantum metrology with photoelectrons, including:

- Interpreting experimental data.
- Extraction/reconstruction/determination of quantum mechanical properties (matrix elements, wavefunctions, density matrices) from experimental data.
- Comparison of experimental and theoretical data.
- New analysis methodologies & techniques.
- Introduction to newly-developed software platform (see below).

1.2 Provisional contents

1.2.1 Part 1: theory & software

General review & update of the topic, including recent theory developments.

1. Introduction

   a. Topic overview.

   b. Context of vol. 3 (following vols. 1 & 2).

   c. Aims: Vol. 3 in the series will continue the exploration of quantum metrology with photoelectrons, with a focus on numerical analysis techniques, forging a closer link between experimental and theoretical results, and making the methodologies discussed directly accessible via a new software platform/ecosystem.

2. Quantum metrology software platform/ecosystem overview

   a. Introduction to python packages for simulation, data analysis, and open-data.

   b. Photoelectron metrology toolkit (PEMtk) package/platform for experimental data processing & analysis. (See *pemtk.readthedocs.io*.)

   c. ePSproc package for theory & simulation. (See *epsproc.readthedocs.io*.)

   d. ePSdata platform for data/results library (see *ePSdata motivations*).

3. General method development: geometric tensor treatment of photoionization, fitting & matrix-inversion techniques

   a. Theory development overview - tensor methods (e.g. *ePSproc tensor methods*)

b. Direct molecular frame reconstruction via matrix-inversion methods (see Gregory, Margaret, Paul Hockett, Albert Stolow, and Varun Makhija. "Towards Molecular Frame Photoelectron Angular Distributions in Polyatomic Molecules from Lab Frame Coherent Rotational Wavepacket Evolution." *Journal of Physics B: Atomic, Molecular and Optical Physics* 54, no. 14 (July 2021): 145601.*DOI: 10.1088/1361-6455/ac135f*.)

4. Numerical implementation & analysis platform tools

   a. Tensor methods implementation in ePSproc/PEMtk.

   b. Information content analysis (inc. basis-set exploration, e.g. *PEMtk fitting demo*), see also vol. 2, sect. 12.1.

   c. Density matrix analysis. (e.g. *ePSproc density matrix method dev notes*)

   d. Generalised bootstrapping implementation in PEMtk (see vol. 2, sects. 11.3 & 12.3)

1.2.2 Part 2: numerical examples

Open-source worked examples using the new software platform.

1. Quantum metrology example: generalised bootstrapping for a homonuclear diatomic scattering system (N2)*

   a. Experimental data overview & simulation.

   b. Matrix element extraction (bootstrap protocol, see vol. 2, sects. 11.3 & 12.3) & statistical analysis.

   c. Direct molecular frame reconstruction via matrix-inversion methods.

   d. Comparison of methods.

   e. Information content/quantum information analysis. (See vol. 2, sect. 12.1.)

2. Quantum metrology example: generalised bootstrapping for a heteronuclear scattering system (CO)*

   a. Experimental data overview & simulation.

   b. Matrix element extraction (bootstrap protocol, see vol. 2, sects. 11.3 & 12.3) & statistical analysis.

   c. Direct molecular frame reconstruction via matrix-inversion methods.

   d. Comparison of methods.

   e. Information content/quantum information analysis. (See vol. 2, sect. 12.1.)

3. Quantum metrology example: generalised bootstrapping and matrix-inversion methods for a complex/general asymmetric top scattering system (C2H4 (ethylene))*

   a. Experimental data overview & simulation.

   b. Matrix element extraction (bootstrap protocol, see vol. 2, sects. 11.3 & 12.3) & statistical analysis.

   c. Direct molecular frame reconstruction via matrix-inversion methods.

   d. Comparison of methods.

   e. Information content/quantum information analysis.

4. Future directions & outlook

5. Summary & conclusions

* Exact choice of "simple" and "complex" systems may change, but should include a homonuclear diatomic and/or heteronuclear diatomic, and symmetric and asymmetric top polyatomic systems. May also include an atomic example.

Part II

Theory & software

CHAPTER

TWO

INTRODUCTION

The overall aim of Vol. 3 is to expand, explore, and illustrate, quantum metrology with photoelectrons: specifically, the application of new python-based tools to tackle problems in matrix element retrieval. The book itself is written as a set of Jupyter Notebooks, hence all the material herein is available directly to readers, and can be run locally to further explore the topic, or adapt the methodology to new problems

Whilst this volume aims to provide a self-contained text, and computational examples which may be used without extensive background knowledge, a brief introduction to the core physics and some recent extensions is presented herein. The unfamiliar reader is referred to Volume 1 of the series for a more detailed presentation, and gateway to the literature [1]. Following the topical introduction, the remainder of Part I introduces the main computational and software tools, recent theory developments, and concludes with a general overview for approaching matrix element retrieval numerically.

Part II details the application of these tools to a few specific cases, starting with a (relatively) simple homonuclear diatomic example, then escalating to a polyatomic asymetric top case.

2.1 Topical introduction

2.2 Context & aims for Vol. 3

As noted previously, Vol. 3 is somewhat distinct from the previous volumes in the series; although involving computational elements, Vols. 1 & 2 [1, 2] are more traditional publications. The material presented in this volume aims to continue the exploration of quantum metrology with photoelectrons, with a focus on numerical analysis techniques, forging a closer link between experimental and theoretical results, and making the methodologies discussed directly accessible via a new software platform/ecosystem. In order to fulfil this aim, Vol. 3 is a computational/computable document, with code directly available to readers. Each chapter or section is composed of a Jupyter Notebook (`.ipynb`), each of which can be modified and used independently.

To facilitate code transparency and reuse, the book is available via a Github repository, Quantum Metrology Vol. 3. An HTML version is also available, which includes interactive figures. A full introduction to the relevant tool-chain, including installation instructions, can be found in Chapter **??**: *Quantum metrology software platform/ecosystem overview*.

CHAPTER

THREE

QUANTUM METROLOGY SOFTWARE PLATFORM/ECOSYSTEM OVERVIEW

STUB

In recent years, a unified Python codebase/ecosystem/platform has been in development to tackle various aspects of photoionization problems, including *ab initio* computations and experimental data handling, and (generalised) matrix element retrieval methods. The eponymous *Quantum Metrology with Photoelectrons* platform is introduced here, and is used for the analysis herein. The main aim of the platform is to provide a unifying data platform, and analysis routines, for photoelectron metrology, including new methods and tools, as well as a unifying bridge between these and existing tools. Fig. **??** provides a general overview of some of the main tools and tasks/layers.

As of late 2022, the new parts of the platform - primarily the Photoelectron Metrology Toolkit [3] library - implement general data handling (although not a full experimental analysis toolchain), matrix element handling and retrieval, which will be the main topic of this volume. In the future, it is hoped that the platform will be extended to other theoretical and experimental methods, including full experimental data handling.

3.1 Analysis components

The two main components of the platform for analysis tasks, as used herein, are:

- The Photoelectron Metrology Toolkit [3] (PEMtk) codebase aims to provide various general data handling routines for photoionization problems. At the time of writing, simulation of observables and fitting routines are implemented, along with some basic utility functions. Much of this is detailed herein, and more technical details and ongoing documentation case be found in the PEMtk documentation [4].
- The ePSproc codebase [5, 6, 7] aims to provide methods for post-processing with *ab initio* radial dipole matrix elements from ePolyScat (ePS) [8, 9, 10, 11], or equivalent matrix elements from other sources (dedicated support for R-matrix results from the RMT suite [12, 13] is in development). The core functionality includes the computation of AF and MF observables. Manual computation without known matrix elements is also possible, e.g. for investigating limiting cases, or data analysis and fitting - hence these routines also provide the backend functionality for PEMtk fitting routines. Again more technical details can be found in the ePSproc documentation [7].

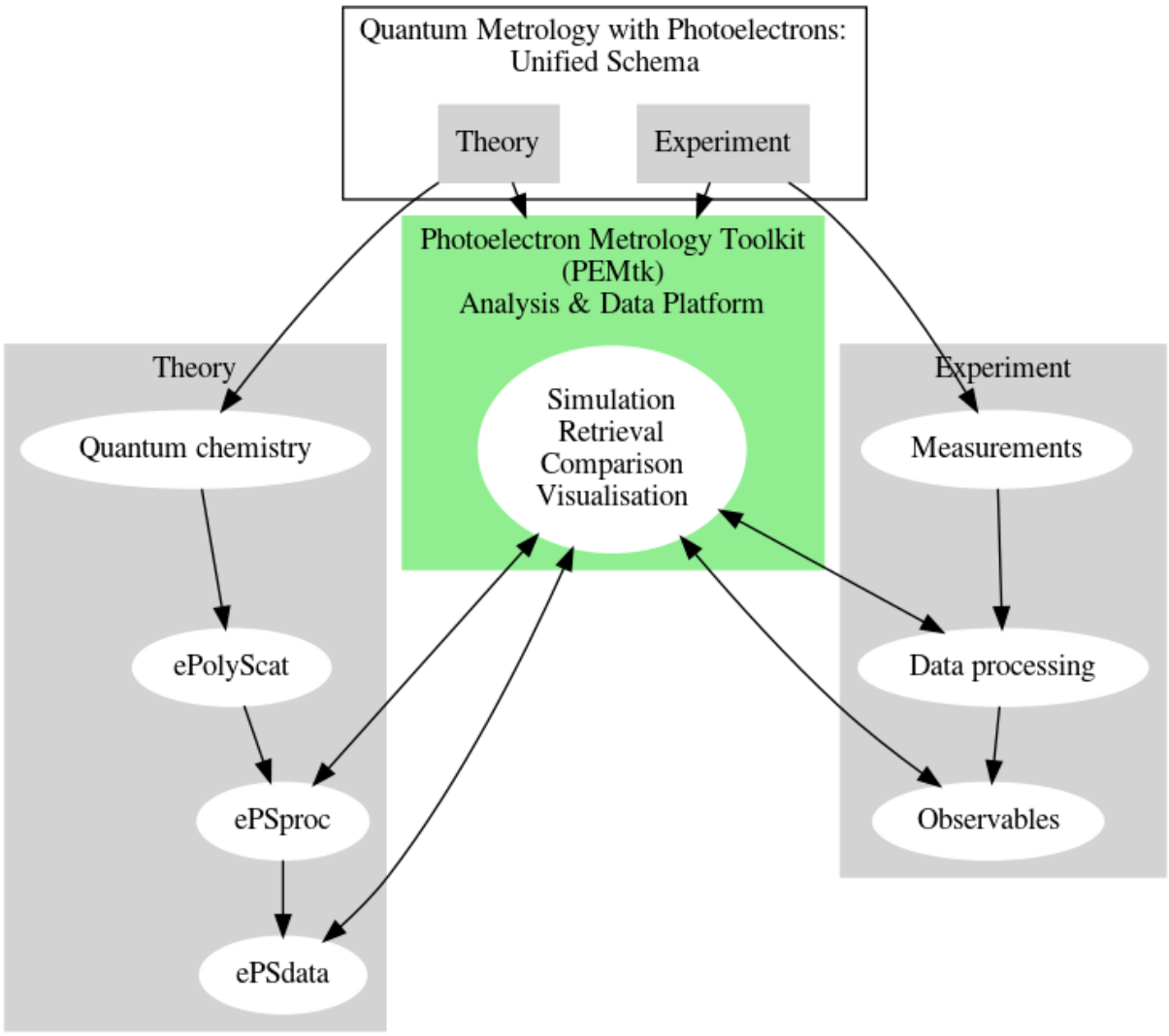


Fig. 3.1: Quantum metrology with photoelectrons ecosystem overview.

3.2 Additional tools

Other tools listed in Fig. **??** include:

- Quantum chemistry layer. The starting point for *ab initio* computations. For the examples herein, all computations made use of Gamess ("The General Atomic and Molecular Electronic Structure System") [14, 15] for electronic structure computations, and inputs to ePolyScat.
- ePolyScat (ePS) [8, 9, 10, 11] is an open-source tool for numerical computation of electron-molecule scattering & photoionization by Lucchese & coworkers. All matrix elements used herein were obtained via ePS calculations. For more details see ePolyScat website and manual [11] and Refs. [8, 9, 10].
- ePSdata [16] is an open-data/open-science collection of ePS + ePSproc results.
  - ePSdata collects ePS datasets, post-processed via ePSproc (Python) in Jupyter notebooks, for a full open-data/open-science transparent pipeline.
  - Source notebooks are available on the ePSdata [16] Github project repository, and notebooks + datasets via ePSdata Zenodo [17]. Each notebook + dataset is given a Zenodo DOI for full traceability, and notebooks are versioned on Github.
  - Note: ePSdata may also be linked or mirrored on the existing ePolyScat Collected Results OSF project, but will effectively supercede those pages.
  - All results are released under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 (CC BY-NC-SA 4.0) license, and are part of an ongoing Open Science initiative.

3.3 Docker deployments

A Docker-based distribution of various codes for tackling photoionization problems is also available from the Open Photoionization Docker Stacks [18] project, which aims to make a range of these tools more accessible to interested researchers, and fully cross-platform/portable. The project currently includes Docker builds for `ePS`, `ePSproc` and `PEMtk`.

3.4 General discussion

Note that, at the time of writing, rotational wavepacket simulation is not yet implemented in the PEMtk suite, and these must be obtained via other codes. An intial build of the `limapack` suite for rotational wavepacket simulations is currently part of the Open Photoionization Docker Stacks [18], but has yet to be tested.

CHAPTER

FOUR

THEORY

- *Observables: photoelectron flux in the LF and MF*
- *Photoionization dynamics*
- *Tensor formulation of photoionization*
- *Information content*

4.1 Observables: photoelectron flux in the LF and MF

The observables of interest - the photoelectron flux as a function of energy, ejection angle, and time - can be written quite generally as expansions in radial and angular basis functions. Various types and definitions are given in this section, including worked numerical examples.

4.1.1 Spherical harmonics

The photoelectron flux as a function of energy, ejection angle, and time, can be written generally as an expansion in spherical harmonics:

$$\bar{I}(\epsilon,t,\theta,\phi)=\sum_{L=0}^{2n}\sum_{M=-L}^{L}\bar{\beta}_{L,M}(\epsilon,t)Y_{L,M}(\theta,\phi) \tag{4.1}$$

Here the flux in the laboratory frame (*LF*) or aligned frame (*AF*) is denoted $\bar{I}(\epsilon,t,\theta,\phi)$, with the bar signifying ensemble averaging, and the molecular frame flux by $I(\epsilon,t,\theta,\phi)$. Similarly, the expansion parameters $\bar{\beta}_{L,M}(\epsilon,t)$ include a bar for the LF/AF case. These observables are generally termed photoelectron angular distributions (*PADs*), often with a prefix denoting the reference frame, e.g. LFPADs, MFPADs, and the associated expansion parameters $\bar{\beta}_{L,M}(\epsilon,t)$ are generically termed *anisotropy paramters*. The polar coordinate system (θ,ϕ) is referenced to an experimentally-defined axis in the *LF*/*AF* case (usually defined by the laser polarization), and the molecular symmetry axis in the *MF*. Some arbitrary examples are given in Fig. **??**, which illustrates both a range of distributions of increasing complexity, and some basic code to set $\beta_{L,M}$ parameters and visualise them; the values used as tabulated in Fig. **??**.

```
# Plot some distributions from specified BLMs

# Set specific LM coeffs by list with setBLMs, items are [l,m,value]
from epsproc.sphCalc import setBLMs

# BLM = setBLMs([[0,0,1],[1,1,1],[2,2,1]])
# BLM = setBLMs([[0,0,1,1,1],[1,1,1,0.5,0.2],[2,2,1,1,0.2]])   # Note different index
BLM = setBLMs([[0,0,1,1,1,1],[1,1,0,0.5,0.8,1],[2,0,1,0.5,0,0],
```

(continues on next page)

(continued from previous page)

```
                [4,2,0,0,0,0.5],[4,-2,0,0,0,0.5]])

# Set the backend to 'pl' for an interactive surface plot with Plotly
# NOTE PL FIG RETURN BROKEN FOR THIS CASE (ePSproc v1.3.1), so run sphSumPlotX too.
dataPlot, figObj = ep.sphFromBLMPlot(BLM, facetDim='t', plotFlag = False, backend =␣
 ↪plotBackend);
figObj = ep.sphSumPlotX(dataPlot,facetDim='t', plotFlag = False, backend =␣
 ↪plotBackend);

# And GLUE for display later with caption
# from myst_nb import glue
# glue("padExamplePlot", figObj[0], display=False);
# Glue with Plotly wrapper.
# gluePlotly("padExamplePlot", figObj[0])    # Working in Render test notebook, but␣
 ↪not here? Issue with subplots?

# Test in separate cell...
gluePlotly("padExamplePlot", figObj[0])    # Working in Render test notebook, but not␣
 ↪here? Issue with subplots?
```

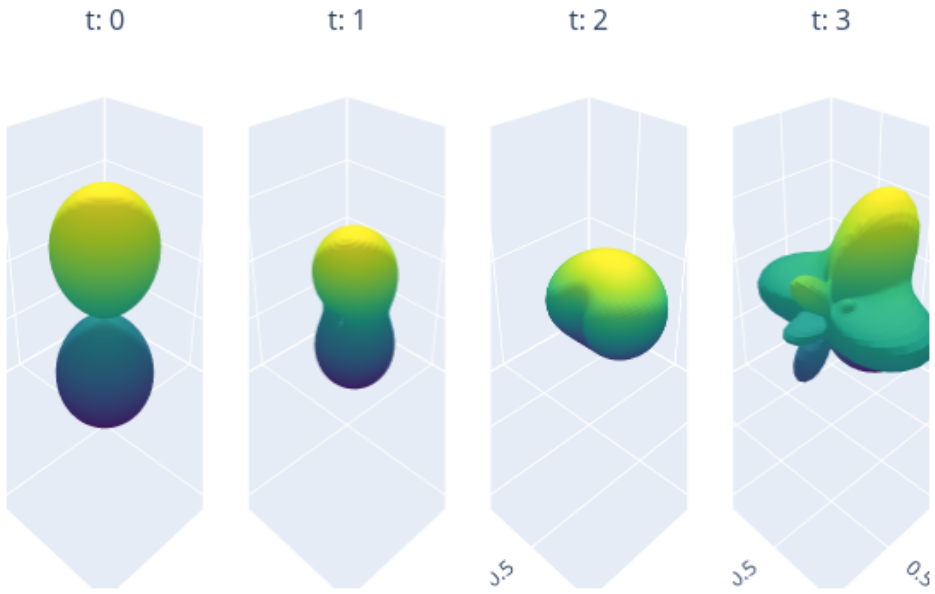


Fig. 4.1: Examples of angular distributions (expansions in spherical harmonics $Y_{L,M}$), for a range of cases. Note that up-down asymmetry is associated with odd-l contributions, and breaking of cylindrical symmetry with $m \neq 0$ terms.

In general, the spherical harmonic rank and order (L, M) of Eq. (**??**) are constrained by experimental factors in the *LF* or *AF*, and n is effectively limited by the molecular alignment (which is correlated with the photon-order for gas phase experiments, or conservation of angular momentum in the *LF* more generally [19]), but in the *MF* is defined by the

t		0	1	2	3
l	m				
0	0	1.0	1.0	1.0	1.0
1	0	0.0	0.5	0.8	1.0
2	0	1.0	0.5	0.0	0.0
4	-2	0.0	0.0	0.0	0.5
	2	0.0	0.0	0.0	0.5

Fig. 4.2: Values used for the plots in Fig. **??**

maximum continuum angular momentum $n = l_{max}$ imparted by the scattering event [20] (note lower-case l here refers specifically to the continuum photoelectron wavefunction, see Eq. (**??**)).

For basic cases these limits may be low: for instance, a simple 1-photon photoionization event ($n = 1$) from an isotropic ensemble (zero net ensemble angular momentum) defines $L_{max} = 2$; for cylindrically symmetric cases (i.e. $D_{\infty h}$ symmetry) $M = 0$ only. For *MF* cases, $l_{max} = 4$ is often given as a reasonable rule-of-thumb for the continuum - hence $L_{max} = 8$ - although in practice higher-l may be populated. Some realistic example cases are discussed later (**PART II**), see also ref. [1] for more discussion and complex examples.

In general, these observables may also be dependent on various other parameters; in Eq. (**??**) two such parameters, (ϵ, t), are included, as the usual variables of interest. Usually ϵ denotes the photoelectron energy, and t is used in the case of time-dependent (usually pump-probe) measurements. As discussed below (Sect. **??**), the origin of such dependencies may be complicated but, in general, the associated photoionization matrix elements are energy-dependent, and time-dependence may also appear for a number of intrinsic or extrinsic (experimental) reasons, e.g. electronic or nuclear dynamics, rotational (alignment) dynamics, electric field dynamics etc. In many cases only one particular aspect may be of interest, so t can be used as a generic label to index changes as per Fig. **??**.

4.1.2 Symmetrized harmonics

Symmetrized (or generalised) harmonics, which essentially provide correctly symmetrized expansions of spherical harmonics (Y_{LM}) functions for a given irreducible representation, Γ, of the molecular point-group can be defined by linear combinations of spherical harmonics (Refs. [21, 22, 23] as below):

$$X_{hl}^{\Gamma\mu*}(\theta,\phi) = \sum_{\lambda} b_{hl\lambda}^{\Gamma\mu} Y_{l,\lambda}(\theta,\phi) \tag{4.1}$$

where:

- Γ is an irreducible representation;
- (l, λ) define the usual spherical harmonic indices (rank, order), but note the use of (l, λ) by convention, since these harmonics are usually referenced to the *MF*;
- $b_{hl\lambda}^{\Gamma\mu}$ are symmetrization coefficients;
- index μ allows for indexing of degenerate components;
- h indexes cases where multiple components are required with all other quantum numbers identical.

Analogously to Eq. (**??**), a general expansion of an observable in the symmetrized harmonic basis set can then be defined as:

$$\bar{I}^{\Gamma}(\epsilon,t,\theta,\phi) = \sum_{\Gamma\mu hl} \bar{\beta}_{hl}^{\Gamma\mu}(\epsilon,t) X_{hl}^{\Gamma\mu*}(\theta,\phi) \tag{4.2}$$

Alternatively, by substitution into Eq. (**??**), and assigning $l = L$ and $\lambda = M$, a general symmetrized expansion may also be defined as:

$$\bar{I}(\epsilon,t,\theta,\phi) = \sum_{\Gamma\mu h}\sum_{L=0}^{2n}\sum_{M=-L}^{L} b_{hLM}^{\Gamma\mu}\bar{\beta}_{L,M}(\epsilon,t)Y_{L,M}(\theta,\phi) \tag{4.3}$$

However, in many cases the symmetrization coefficients are subsumed into the $\beta_{L,M}$ terms (or underlying matrix elements); in this case a simplified symmetrized expansion can be defined as:

$$\bar{I}^{\Gamma}(\epsilon,t,\theta,\phi) = \sum_{L=0}^{2n}\sum_{M=-L}^{L} \bar{\beta}_{L,M}^{\Gamma}(\epsilon,t)Y_{L,M}(\theta,\phi) \tag{4.3}$$

Where the expansion is defined for a given symmetry and irreducible representation with the shorthand Γ; in many systems a single label may be sufficient here, since allowed (L, M) terms will be defined uniquely by irreducible representation, although multiple quantum numbers may be required for unique definition in the most general cases as per Eq. (**??**) (e.g. for cases with degenerate components). Further details and usage in relation to channel functions are also discussed in Sect. **??** (see, in particular, Eq. (**??**) for a similar general case), and in relation to fitting for specific cases in **PART II**.

The exact form of these coefficients will depend on the point-group of the system, see, e.g. Refs. [23, 24]. Numerical routines for the generation of symmetrized harmonics are implemented in Photoelectron Metrology Toolkit [3]: point-groups, character table generation and symmetrization (computing $b_{hl\lambda}^{\Gamma\mu}$ parameters) is handled by libmsym [25, 26]; additional handling also makes use of pySHtools [27, 28].

A brief numerical example is given below, for Td symmetry (l_{max} =4), and more details can be found in the PEMtk documentation [4]. In this case, full tabulations of the parameters lists all $b_{hLM}^{\Gamma\mu}$ for each irreducible representation, and the corresponding PADs are illustrated in Fig. **??**.

Note: Full tabulations of the parameters available in HTML or notebook formats only.

```
# Import class
from pemtk.sym.symHarm import symHarm

# Compute hamronics for Td, lmax=4
sym = 'Td'
lmax=4

symObj = symHarm(sym,lmax)

# Character tables can be displayed
symObj.printCharacterTable()

# Glue items for later
glue("symHarmPG2", f"${sym}$", display=False)
glue("symHarmPG", sym, display=False)
glue("symHarmLmax", lmax, display=False)
```

```
# The full set of expansion parameters can be tabulated
# pd.set_option('display.max_rows', 1)
symObj.displayXlm()  # Display values (note this defaults to REAL harmonics)
# symObj.displayXlm(YlmType='comp')   # Display values for COMPLEX harmonic expansion.
```

```
# To plot using ePSproc/PEMtk class, these values can be converted to ePSproc BLM␣
↪data type...

# Run conversion - the default is to set the coeffs to the 'BLM' data type
symObj.toePSproc()

# Set to new key in data class
data.data['symHarm'] = {}

for dataType in ['BLM']:  #['matE','BLM']:
    data.data['symHarm'][dataType] = symObj.coeffs[dataType]['b (comp)']  # Select␣
↪expansion in complex harmonics
    data.data['symHarm'][dataType].attrs = symObj.coeffs[dataType].attrs

# Plot full harmonics expansions, plots by symmetry
# Note 'squeeze=True' to force drop of singleton dims may be required.
# data.padPlot(keys='symHarm',dataType='BLM', facetDims = ['Cont'], squeeze = True,␣
↪backend=plotBackend)

data.padPlot(keys='symHarm',dataType='BLM', facetDims = ['Cont'], squeeze = True,␣
↪backend=plotBackend, plotFlag=False, returnFlag=True)  # Working
figObj = data.data['symHarm']['plots']['BLM']['polar'][0]

# And GLUE for display later with caption
# from myst_nb import glue
# glue("padExamplePlot2", figObj, display=False);
gluePlotly("symHarmPADs", figObj)
```

4.1.3 Real harmonics

4.1.4 Legendre polynomials

4.2 Photoionization dynamics

The core physics of photoionization has been covered extensively in the literature, and only a very brief overview is provided here with sufficient detail to introduce the metrology/reconstruction/retrieval problem; the reader is referred to Vol. 1 [1] (and refs. therein) for further details and general discussion.

Photoionization can be described by the coupling of an initial state of the system to a particular final state (photoion(s) plus free photoelectron(s)), coupled by an electric field/photon. Very generically, this can be written as a matrix element $\langle\Psi_i|\hat{\Gamma}(\mathbf{E})|\Psi_f\rangle$, where $\hat{\Gamma}(\mathbf{E})$ defines the light-matter coupling operator (depending on the electric field $\mathbf{E}$), and Ψ_i, Ψ_f the total wavefunctions of the initial and final states respectively.

There are many flavours of this fundamental light-matter interaction, depending on system and coupling. For metrology, the focus is currently on the simplest case of single-photon absorption, in the weak field (or purturbative), dipolar regime, resulting in a single photoelectron. (For more discussion of various approximations in photoionzation, see Refs. [29, 30].) In this case the core physics is well defined, and tractable (albeit non-trivial), via the separation of matrix elements into radial (energy) and angular-momentum (geometric) terms pertaining to couplings between various elements of the problem; the retrieval of such matrix elements is a well-defined problem, making use of analytic terms in combination with fitting methodologies as explored herein. Again, more extensive background and discussion can be found in *Quantum Metrology* Vol. 1 [1], and references therein. % [TODO: Add some more refs here?]

The basic case also provides a strong foundation for extension into more complex light-matter interactions, in particular cases with shaped laser-fields (i.e. a time-dependent coupling $\hat{\Gamma}(\mathbf{E},\mathbf{t})$) and multi-photon processes (which require mul-

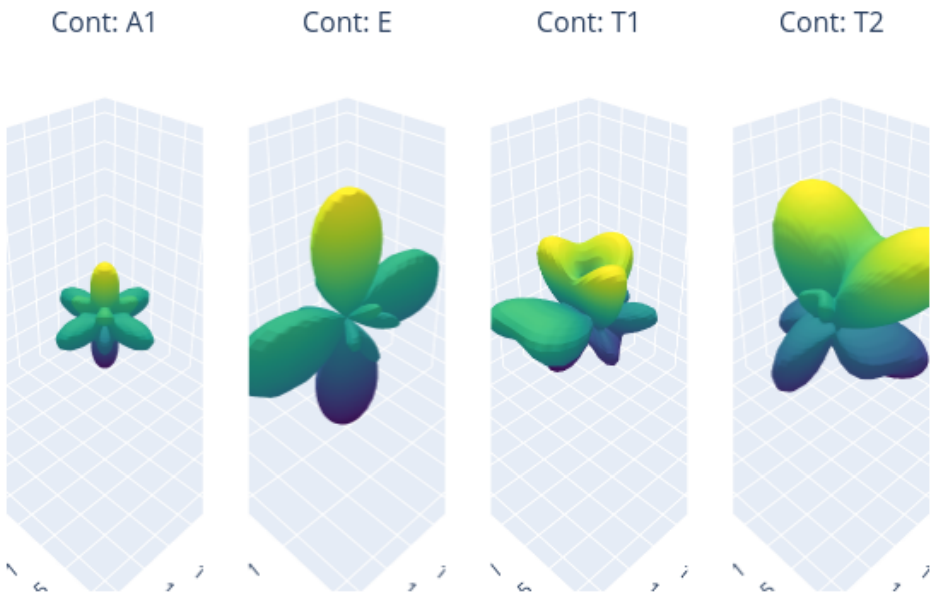


Fig. 4.3: Examples of angular distributions from expansions in symmetrized harmonics $X_{hl}^{\Gamma\mu*}(\theta,\phi)$, for all irreducible representations in Td symmetry (l_{max} =4)

tiple matrix elements, and/or different approximations). Note, however, that non-perturbative (strong field) light-matter interactions are, typically, not amenable to description in a separable picture in this manner. In such cases the laser field, molecular and continuum properties are strongly coupled, and are typically treated numerically in a fully time-dependent manner (although some separation of terms may work in some cases, depending on the system and interaction(s) at hand).

Underlying the photoelecton observables is the photoelectron continuum state $|\mathbf{k}\rangle$, prepared via photoionization. The photoelectron momentum vector is denoted generally by $\mathbf{k} = k\hat{\mathbf{k}}$, in the molecular frame (*MF*). The ionization matrix elements associated with this transition provide the set of quantum amplitudes completely defining the final continuum scattering state,

$$\left|\Psi_f\right\rangle = \sum \int |\Psi_+; \mathbf{k}\rangle \left\langle \Psi_+; \mathbf{k}|\Psi_f\right\rangle d\mathbf{k}, \tag{4.3}$$

where the sum is over states of the molecular ion $|\Psi_+\rangle$. The number of ionic states accessed depends on the nature of the ionizing pulse and interaction. For the dipolar case,

$$\hat{\Gamma}(\mathbf{E}) = \hat{\mu}.\mathbf{E} \tag{4.4}$$

Hence,

$$\left\langle \Psi_+; \mathbf{k}|\Psi_f\right\rangle = \langle \Psi_+;\, \mathbf{k}|\hat{\mu}.\mathbf{E}|\Psi_i\rangle \tag{4.5}$$

Where the notation implies a perturbative photoionization event from an initial state i to a particular ion plus electron state following absorption of a photon $h\nu$, $|\Psi_i\rangle + h\nu \rightarrow |\Psi_+; \mathbf{k}\rangle$, and $\hat{\mu}.\mathbf{E}$ is the usual dipole interaction term [31], which includes a sum over all electrons s defined in position space as $\mathbf{r_s}$:

$$\hat{\mu} = -e \sum_s \mathbf{r_s} \tag{4.6}$$

The position space photoelectron wavefunction is typically expressed in the "partial wave" basis, expanded as (asymptotic) continuum eignstates of orbital angular momentum, with angular momentum components (l, m) (note lower case notation for the partial wave components, distinct from upper-case for the similar terms (L, M) in the observables),

$$\Psi_{\mathbf{k}}(r) \equiv \langle r|\mathbf{k}\rangle = \sum_{lm} Y_{lm}(\hat{\mathbf{k}})\psi_{lm}(r, k) \tag{4.7}$$

where r are MF electronic coordinates and $Y_{lm}(\hat{\mathbf{k}})$ are the spherical harmonics.

Similarly, the ionization dipole matrix elements can be separated generally into radial (energy-dependent or 'dynamical' terms) and geometric (angular momentum) parts (this separation is essentially the Wigner-Eckart Theorem, see Ref. [32] for general discussion), and written generally as (using notation similar to [33]):

$$\langle \Psi_+;\, \mathbf{k}|\hat{\mu}.\mathbf{E}|\Psi_i\rangle = \sum_{lm} \gamma_{l,m}\mathbf{r}_{k,l,m} \tag{4.8}$$

Provided that the geometric part of the matrix elements $\gamma_{l,m}$ - which includes the geometric rotations into the LF arising from the dot product in Eq. (**??**) and other angular-momentum coupling terms - are know, knowledge of the so-called radial (or reduced) dipole matrix elements, at a given k thus equates to a full description of the system dynamics (and, hence, the observables).

For the simplest treatment, the radial matrix element can be approximated as a 1-electron integral involving the initial electronic state (orbital), and final continuum photoelectron wavefunction:

$$\mathbf{r}_{k,l,m} = \int \psi_{lm}(r, k) r \Psi_i(r) dr \tag{4.9}$$

As noted above, the geometric terms $\gamma_{l,m}$ are analytical functions which can be computed for a given case - minimally requiring knowledge of the molecular symmetry and polarization geometry, although other factors may also play a role (see Sect. **??** for details).

The photoelectron angular distribution (PAD) at a given (ϵ, t) can then be determined by the squared projection of $\left|\Psi_f\right\rangle$ onto a specific state $\left|\Psi_+;\mathbf{k}\right\rangle$,

and therefore the amplitudes in Eq. (**??**) also determine the observable anisotropy parameters $\beta_{L,M}(\epsilon,t)$ (Eqn. (**??**)). (Note that the photoelectron energy ϵ and momentum k are used somewhat interchangeably herein, with the former usually preferred in reference to observables.) Note, also, that in the treatment above there is no time-dependence incorporated in the notation; however, a time-dependent treatment readily follows, and may be incorporated either as explicit time-dependent modulations in the expansion of the wavefunctions for a given case, or implicitly in the radial matrix elements. Examples of the former include, e.g. a rotational or vibrational wavepacket, or a time-dependent laser field. The rotational wavepacket case is discussed herein (see Sect. **??**). The radial matrix elements are a sensitive function of molecular geometry and electronic configuration in general, hence may be considered to be responsive to molecular dynamics, although they are formally time-independent in a Born-Oppenheimer basis - for further general discussion and examples see Ref. [34]; discussions of more complex cases with electronic and nuclear dynamics can be found in Refs. [30, 35, 36, 37].

Typically, for reconstruction experiments, a given measurement will be selected to simplify this as much as possible by, e.g., populating only a single ionic state (or states for which the corresponding observables are experimentally energetically-resolvable), and with a bandwidth $d\mathbf{k}$ which is small enough such that the matrix elements can be assumed constant. Importantly, the angle-resolved observables are sensitive to the magnitudes and (relative) phases of these matrix elements, and can be considered as angular interferograms

4.3 Tensor formulation of photoionization

A number of authors have treated MFPADs and related problems [REFS]; herein, a geometric tensor based formalism is developed, which is close in spirit to the treatments given by Underwood and co-workers [37, 38, 39], but further separates various sets of physical parameters into dedicated tensors; this allows for a unified theoretical and numerical treatment, where the latter computes properties as tensor variables which can be further manipulated and investigated. Furthermore, the tensors can readily be converted to a density matrix representation [32, 40], which is more natural for some quantities, and also emphasizes the link to quantum state tomography and other quantum information techniques. Much of the theoretical background, as well as application to aspects of the current problem, can be found in the textbooks of Blum [40] and Zare [32].

Within this treatment, the observables can be defined in a series of simplified forms, emphasizing the quantities of interest for a given problem. Some details are defined in the following subsections,

4.3.1 Channel functions

A simple form of the equations, amenable to fitting, is to write the observables in terms of "channel functions", which define the ionization continuum for a given case and set of parameters u (e.g. defined for the MF, or defined for a specific experimental configuration),

$$\beta^{u}_{L,M} = \sum_{\zeta,\zeta'} \Upsilon^{u,\zeta\zeta'}_{L,M} \, \mathbb{I}^{\zeta\zeta'} \tag{4.10}$$

Where ζ, ζ' collect all the required quantum numbers, and define all (coherent) pairs of components. The term $\mathbb{I}^{\zeta\zeta'}$ denotes the coherent square of the ionization matrix elements:

$$\mathbb{I}^{\zeta,\zeta} = I^{\zeta}(\epsilon) I^{\zeta'*}(\epsilon) \tag{4.11}$$

This is effectively a convolution equation (cf. refs. [38, 41]) with channel functions, for a given "experiment" u, summed over all terms ζ, ζ'. Aside from the change in notation (which is here chosen to match the formalism of Refs. [8, 9, 10]),

these matrix elements are essentially identical to the simplified (radial) forms $\mathbf{r}_{k,l,m}$ defined in Eqn. (**??**), in the case where $\zeta = k, l, m$. These complex matrix elements can also be equivalently defined in a magnitude, phase form:

$$I^{\zeta}(\epsilon) \equiv \mathbf{r}_{\zeta} \equiv r_{\zeta} e^{i\phi_{\zeta}} \tag{4.12}$$

This tensorial form is numerically implemented in the ePSproc [6] codebase, and is in contradistinction to standard numerical routines in which the requisite terms are usually computed from vectorial and/or nested summations, which can be somewhat opaque to detailed interpretation, and typically implement the full computation of the observables in one monolithic computational routine. The Photoelectron Metrology Toolkit [3] codebase implements matrix element retrieval based on the tensor formalism, with pre-computation of all the geometric tensor components (channel functions) prior to a fitting protocol for matrix element analysis, essentially a fit to Eqn. (**??**), with terms $I^{\zeta}(\epsilon)$ as the unknowns (in magnitude, phase form per (**??**)). The main computational cost of a tensor-based approach is that more RAM is required to store the full set of tensor variables; however, the method is computationally efficient since it is inherently parallel (as compared to a traditional, serial loop-based solution), hence may lead to significantly faster evaluation of observables. Furthermore, the method allows for the computational routines to match the formalism quite closely, and investigation of the properties of the channel functions for a given problem in general terms, as well as for specific experimental cases.

TODO: numerical examples here or below. TODO: benchmarks, or link to them (see test fitting notebooks…?).

4.3.2 Full tensor expansion

In more detail, the channel functions can be given as a set of tensors, defining each aspect of the problem.

For the MF:

$$\begin{aligned}\beta^{\mu_i,\mu_f}_{L,-M}(\epsilon) = & (-1)^M \sum_{P,R',R} (2P+1)^{\frac{1}{2}} E_{P-R}(\hat{e};\mu_0) \\ \times & \sum_{l,m,\mu} \sum_{l',m',\mu'} (-1)^{(\mu'-\mu_0)} \Lambda_{R',R}(R_{\hat{n}};\mu,P,R,R') B_{L,-M}(l,l',m,m') \\ \times & I^{p_i\mu_i,p_f\mu_f}_{l,m,\mu}(\epsilon) I^{p_i\mu_i,p_f\mu_f *}_{l',m',\mu'}(\epsilon)\end{aligned} \tag{4.13}$$

And the LF/AF as:

$$\begin{aligned}\bar{\beta}^{\mu_i,\mu_f}_{L,-M}(E,t) = & (-1)^M \sum_{P,R',R} [P]^{\frac{1}{2}} E_{P-R}(\hat{e};\mu_0) \\ \times & \sum_{l,m,\mu} \sum_{l',m',\mu'} (-1)^{(\mu'-\mu_0)} \Lambda_{R'}(\mu,P,R') B_{L,S-R'}(l,l',m,m') \\ \times & I^{p_i\mu_i,p_f\mu_f}_{l,m,\mu}(\epsilon) I^{p_i\mu_i,p_f\mu_f *}_{l',m',\mu'}(\epsilon) \sum_{K,Q,S} \Delta_{L,M}(K,Q,S) A^K_{Q,S}(t)\end{aligned} \tag{4.14}$$

In both cases a set of geometric tensor terms are required,

these terms provide details of:

- $E_{P-R}(\hat{e};\mu_0)$: polarization geometry & coupling with the electric field.
- $B_{L,S-R'}(l,l',m,m')$: geometric coupling of the partial waves into the $\beta_{L,M}$ terms (spherical tensors).
- $\Lambda_{R'}(\mu,P,R')$: frame couplings and rotations.
- $\Delta_{L,M}(K,Q,S)$: alignment frame coupling.
- $A^K_{Q,S}(t)$: ensemble alignment described as a set of axis distribution moments (ADMs).

And $I^{p_i\mu_i,p_f\mu_f}_{l,m,\mu}(\epsilon)$ are the (radial) dipole ionization matrix elements, as a function of energy ϵ. These matrix elements are essentially identical to the simplified forms $r_{k,l,m}$ defined in Eqn. (**??**), except with additional indices to label symmetry and polarization components defined by a set of partial-waves $\{l,m\}$, for polarization component μ (denoting the photon

angular momentum components) and channels (symmetries) labelled by initial and final state indexes $p_i\mu_i, p_f\mu_f$. The notation here follows that used by ePolyScat (ePS) [8, 9, 10, 11], and these matrix elements again represent the quantities to be obtained numerically from data analysis, or from an ePolyScat (or similar) calculation.

[Numerical example here, or already included above somewhere]

Note that, in this case as given, time-dependence arises purely from the $A^K_{Q,S}(t)$ terms in the AF case, and the electric field term currently describes only the photon angular momentum coupling, although can in principle also describe time-dependent/shaped fields. Similarly, a time-dependent initial state (e.g. a vibrational wavepacket) could also describe a time-dependent MF case.

It should be emphasized, however, that the underlying physical quantities are essentially identical in all the theoretical approaches, with a set of coupled angular-momenta defining the geometrical part of the photoionization problem, despite these differences in the details of the theory and notation.

4.3.3 Density matrix representation

The density operator associated with the continuum state in Eq. (**??**) is easily written as $\hat{\rho} = |\Psi_c\rangle\langle\Psi_c|$. In the channel function basis, this leads to a density matrix given by the radial matrix elements:

$$\rho^{\zeta\zeta'} = \mathbb{I}^{\zeta,\zeta'} \tag{4.15}$$

Since the matrix elements characterise the scattering event, the density matrix provides an equivalent characterisation of the scattering event.

Further discussion can also be found in the literature, see, e.g., Ref. [40] for general discussion, Ref. [33] for application in pump-probe schemes.

TODO: numerical examples here

4.4 Information content

As discussed in *Quantum Metrology* Vol. 2 [2], the information content of a single observable might be regarded as simply the number of contributing $\beta_{L,M}$ parameters. In set notation:

$$M = \mathrm{n}\{\beta_{L,M}\} \tag{4.16}$$

where M is the information content of the measurement, defined as $\mathrm{n}\{...\}$ the cardinality (number of elements) of the set of contributing parameters. A set of measurements, made for some experimental variable u, will then have a total information content:

$$M_u = \sum_u \mathrm{n}\{\beta^u_{L,M}\}$$

In the case where a single measurement contains multiple $\beta_{L,M}$, e.g. as a function of energy ϵ or time t, the information content will naturally be larger:

$$\begin{aligned} M_u = & \sum_{u,k,t} \mathrm{n}\{\beta^u_{L,M}(\epsilon,t)\} \\ = & \quad M_u \times M \end{aligned}$$

where the second line pertains if each measurement has the same native information content, independent of u. It may be that the variable k is continuous (e.g. photoelectron energy), but in practice it will usually be discretized in some fashion by the measurement.

In terms of purely experimental methodologies, a larger M_u clearly defines a richer experimental measurement which explores more of the total measurement space spanned by the full set of $\{\beta^u_{L,M}(k,t)\}$. However, in this basic definition

a larger M_u does not necessarily indicate a higher information content for quantum retrieval applications. The reason for this is simply down to the complexity of the problem (cf. Eq. (**??**)), in which many couplings define the sensitivity of the observable to the underlying system properties of interest. In this sense, more measurements, and larger M, may only add redundancy, rather than new information.

A more complete accounting of information content would, therefore, also include the channel couplings, i.e. sensitivity/dependence of the observable to a given system property, in some manner. For the case of a time-dependent measurement, arising from a rotational wavepacket, this can be written as:

$$M_u = \mathrm{n}\{\Upsilon^u_{L,M}(\epsilon, t)\}$$

In this case, each (ϵ, t) is treated as an independent measurement with unique information content, although there may be redundancy as a function of t depending on the nature of the rotational wavepacket and channel functions.

(Note this is in distinction to previously demonstrated cases where the time-dependence was created from a shaped laser-field, and was integrated over in the measurements, which provided a coherently-multiplexed case, see refs. [42, 43, 44] for details.)

CHAPTER

FIVE

NUMERICAL IMPLEMENTATION

Further details of the numerical implementations - base packages, conventions and refs - to go here or in appendix?

Part III

Backmatter

CHAPTER

SIX

BIBLIOGRAPHY

CHAPTER

SEVEN

GLOSSARY

MF Molecular frame (MF) - coordinate system referenced to the molecule, usually with the z-axis corresponding to the highest symmetry axis.

LF Laboratory or lab frame (LF) - coordinate system referenced to the laboratory frame, usually with the z-axis corresponding to the laser field polarization. For circularly or elliptically polarized light the propagation direction is conventionally used for the z-axis. In some cases a different z-axis may be chosen, e.g. as defined by a detector.

AF Aligned frame (AF) - coordinate system referenced to molecular alignment axis or axes. For 1D alignment, the z-axis usually corresponds to the alignment field polarization, and hence may be identical to the standard *LF* definition. For high degrees of (3D) alignment the AF may approach the *MF* in the ideal case, although will usually be limited by the symmetry of the system.

PADs Photoelectron angular distributions (PADs), often with a prefix denoting the reference frame, e.g. LFPADs, MFPADs (sometimes also hypenated, e.g. LF-PADs). Usage is often synonymous with the associated *anisotropy paramters* (or "betas").

anisotropy paramters Expansion parameters $\beta_{L,M}$ for an expansion in spherical harmonics (or similar basis sets of angular momentum functions in polar coordinates), e.g. Eq. (**??**). Often referred to simply as "beta parameters", and may be dependent on various properties, e.g. $\beta_{L,M}(\epsilon, t...)$. Herein upper-case L, M usually refer to observables or the general case, whilst lower-case (l, m) usually refer specifically to the photoelectron wavefunction partial waves, and (l, λ) usually denote these terms referenced specifically to the molecular frame.

Part IV

Test pages

CHAPTER

EIGHT

BUILD VERSIONS AND CONFIG TESTS

8.1 Versions

```
import scooby
scooby.Report(additional=['pemtk','epsproc','xarray', 'jupyterlab','plotly','holoviews
↪'])
```

```
* sparse not found, sparse matrix forms not available.
* natsort not found, some sorting functions not available.
* Setting plotter defaults with epsproc.basicPlotters.setPlotters(). Run directly␣
↪to modify, or change options in local env.
* Set Holoviews with bokeh.
* pyevtk not found, VTK export not available.
```

```
OMP: Info #271: omp_set_nested routine deprecated, please use omp_set_max_active_
↪levels instead.
```

```
--------------------------------------------------------------------------------
  Date: Wed Nov 23 15:22:57 2022 EST

                OS : Linux
            CPU(s) : 64
           Machine : x86_64
      Architecture : 64bit
               RAM : 62.8 GiB
       Environment : Jupyter
       File system : btrfs

  Python 3.9.7 | packaged by conda-forge | (default, Sep 29 2021, 19:20:46)
  [GCC 9.4.0]

             pemtk : 0.0.1
           epsproc : 1.3.2-dev
            xarray : 2022.3.0
        jupyterlab : 3.2.1
            plotly : 5.11.0
         holoviews : 1.15.2
             numpy : 1.20.3
             scipy : 1.7.1
           IPython : 7.28.0
        matplotlib : 3.4.3
```

(continues on next page)

(continued from previous page)

```
            scooby : 0.7.0
--------------------------------------------------------------------------------
```

```
!jupyter-book --version
```

```
Jupyter Book      : 0.13.1
External ToC      : 0.2.4
MyST-Parser       : 0.15.2
MyST-NB           : 0.13.2
Sphinx Book Theme : 0.3.3
Jupyter-Cache     : 0.4.3
NbClient          : 0.5.4
```

8.2 Docker build env

To do

8.3 Book versions

```
QMpath = '/home/jovyan/QM3'
!git -C {QMpath} branch
!git -C {QMpath} log --format="%H" -n 1
```

```
  gh-pages
* main
0e92235dfa44c7b48350e3920623b4cda4d51137
```

```
# Check current remote commits
!git ls-remote --heads https://github.com/phockett/Quantum-Metrology-with-
 ↪Photoelectrons-Vol3
```

```
061d09ff7c97b77e00230c81159c372cbf784b3b        refs/heads/gh-pages
9a70adcd4b3a66f0c8c1e24bc5556543b8bd84ba        refs/heads/main
```

8.4 Github pkg versions

Note - can't get versions for local pip installs from repo(?).

```
from pathlib import Path
import epsproc as ep
ep.__file__
```

```
'/opt/conda/lib/python3.9/site-packages/epsproc/__init__.py'
```

```
# Check current Git commit for local ePSproc version - NOTE THIS ONLY WORKS FOR␣
↪INSTALLED FROM GIT CLONES
# from pathlib import Path
# import epsproc as ep
!git -C {Path(ep.__file__).parent} branch
!git -C {Path(ep.__file__).parent} log --format="%H" -n 1
```

```
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
```

```
# Check current remote commits
!git ls-remote --heads https://github.com/phockett/ePSproc
```

```
92c661789a7d2927f2b53d7266f57de70b3834fa        refs/heads/dependabot/pip/notes/
↪envs/envs-versioned/mistune-2.0.3
fe1e9540c7b91fe571f60562acd31d8e489d491e        refs/heads/dependabot/pip/notes/
↪envs/envs-versioned/nbconvert-6.5.1
70b80a1e3a54de91c2bfe3b6be82d611fcfd5f43        refs/heads/dependabot/pip/notes/
↪envs/envs-versioned/pillow-9.3.0
b8d23a0273a78fd0a24deb6803d8ad048124c501        refs/heads/dev
1c0b8fd409648f07c85f4f20628b5ea7627e0c4e        refs/heads/master
69cd89ce5bc0ad6d465a4bd8df6fba15d3fd1aee        refs/heads/numba-tests
ea30878c842f09d525fbf39fa269fa2302a13b57        refs/heads/revert-9-master
baf0be0c962e8ab3c3df57c8f70f0e939f99cbd7        refs/heads/testDev
```

```
# Check current remote commits
!git ls-remote --heads https://github.com/phockett/PEMtk
```

```
bb51d8b601eee2486a5a8757f081549398e56e10        refs/heads/master
3f4686dffdbb310f15692f978ba36d6a3d15e8d3        refs/heads/mfFittingDev
```

8.5 Full conda env

```
!conda list
```

```
# packages in environment at /opt/conda:
#
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                 conda_forge    conda-forge
_openmp_mutex             4.5                       1_llvm    conda-forge
alabaster                 0.7.12                   pypi_0    pypi
alembic                   1.7.4                pyhd8ed1ab_0    conda-forge
alsa-lib                  1.2.3.2                h166bdaf_0    conda-forge
altair                    4.1.0                      py_1    conda-forge
ansi2html                 1.8.0              py39hf3d152e_1    conda-forge
anyio                     3.3.4              py39hf3d152e_0    conda-forge
appdirs                   1.4.4                pyh9f0ad1d_0    conda-forge
argon2-cffi               21.1.0             py39h3811e60_0    conda-forge
asteval                   0.9.27               pyhd8ed1ab_0    conda-forge
```

(continues on next page)

(continued from previous page)

```
astropy                       5.0.4           py39hd257fcd_0    conda-forge
async_generator               1.10                 py_0    conda-forge
atk-1.0                       2.36.0           h3371d22_4    conda-forge
attrs                         21.2.0         pyhd8ed1ab_0    conda-forge
babel                         2.9.1          pyh44b312d_0    conda-forge
backcall                      0.2.0          pyh9f0ad1d_0    conda-forge
backports                     1.0                  py_2    conda-forge
backports.functools_lru_cache 1.6.4             pyhd8ed1ab_0    conda-forge
beautifulsoup4                4.10.0         pyha770c72_0    conda-forge
blas                          2.112             openblas    conda-forge
blas-devel                    3.9.0          12_linux64_openblas    conda-forge
bleach                        4.1.0          pyhd8ed1ab_0    conda-forge
blinker                       1.4                  py_1    conda-forge
blosc                         1.21.0           h9c3ff4c_0    conda-forge
bokeh                         2.4.1          py39hf3d152e_1    conda-forge
boost-cpp                     1.74.0           h312852a_4    conda-forge
bottleneck                    1.3.2          py39hce5d2b2_4    conda-forge
brotli                        1.0.9            h7f98852_5    conda-forge
brotli-bin                    1.0.9            h7f98852_5    conda-forge
brotli-python                 1.0.9          py39h5a03fae_7    conda-forge
brotlipy                      0.7.0          py39h3811e60_1001    conda-forge
brunsli                       0.1              h9c3ff4c_0    conda-forge
bzip2                         1.0.8            h7f98852_4    conda-forge
c-ares                        1.18.1           h7f98852_0    conda-forge
c-blosc2                      2.0.4            h5f21a17_1    conda-forge
ca-certificates               2022.9.24        ha878542_0    conda-forge
cached-property               1.5.2            hd8ed1ab_1    conda-forge
cached_property               1.5.2          pyha770c72_1    conda-forge
cairo                         1.16.0          h6cf1ce9_1008    conda-forge
cartopy                       0.20.1         py39he7aa91e_2    conda-forge
certifi                       2022.9.24        pyhd8ed1ab_0    conda-forge
certipy                       0.1.3                py_0    conda-forge
cffi                          1.14.6         py39h4bc2ebd_1    conda-forge
cfitsio                       4.0.0            h9a35b8e_0    conda-forge
cftime                        1.6.0          py39hd257fcd_1    conda-forge
chardet                       4.0.0          py39hf3d152e_1    conda-forge
charls                        2.2.0            h9c3ff4c_0    conda-forge
charset-normalizer            2.0.0          pyhd8ed1ab_0    conda-forge
chrpath                       0.16            h7f98852_1002    conda-forge
click                         8.0.3          py39hf3d152e_0    conda-forge
cloudpickle                   2.0.0          pyhd8ed1ab_0    conda-forge
colorama                      0.4.4          pyh9f0ad1d_0    conda-forge
colorcet                      3.0.1          pyhd8ed1ab_0    conda-forge
conda                         4.10.3         py39hf3d152e_2    conda-forge
conda-package-handling        1.7.3          py39h3811e60_0    conda-forge
configurable-http-proxy       4.5.0          node15_he6ea98c_0    conda-forge
cryptography                  35.0.0         py39h95dcef6_1    conda-forge
curl                          7.79.1           h2574ce0_1    conda-forge
cycler                        0.10.0               py_2    conda-forge
cython                        0.29.24        py39he80948d_0    conda-forge
cytoolz                       0.11.0         py39h3811e60_3    conda-forge
dash                          2.7.0          pyhd8ed1ab_0    conda-forge
dask                          2021.10.0      pyhd8ed1ab_0    conda-forge
dask-core                     2021.10.0      pyhd8ed1ab_0    conda-forge
dbus                          1.13.6           h5008d03_3    conda-forge
dcw-gmt                       2.1.1            ha770c72_0    conda-forge
```

(continues on next page)

(continued from previous page)

```
debugpy                    1.4.1            py39he80948d_0    conda-forge
decorator                  5.1.0              pyhd8ed1ab_0    conda-forge
defusedxml                 0.7.1              pyhd8ed1ab_0    conda-forge
dill                       0.3.4              pyhd8ed1ab_0    conda-forge
distributed                2021.10.0        py39hf3d152e_0    conda-forge
docutils                   0.17.1                   pypi_0    pypi
ducc0                      0.23.0           py39h5a03fae_0    conda-forge
entrypoints                0.3            py39hde42818_1002    conda-forge
epsproc                    1.3.2.dev0               pypi_0    pypi
expat                      2.4.8                h27087fc_0    conda-forge
ffmpeg                     4.4.0                h6987444_4    conda-forge
fftw                       3.3.10         nompi_h77c792f_102    conda-forge
flask                      2.1.3              pyhd8ed1ab_0    conda-forge
flask-compress             1.13               pyhd8ed1ab_0    conda-forge
font-ttf-dejavu-sans-mono  2.37                 hab24e00_0    conda-forge
font-ttf-inconsolata       3.000                h77eed37_0    conda-forge
font-ttf-source-code-pro   2.038                h77eed37_0    conda-forge
font-ttf-ubuntu            0.83                 hab24e00_0    conda-forge
fontconfig                 2.14.0               h8e229c2_0    conda-forge
fonts-conda-ecosystem      1                             0    conda-forge
fonts-conda-forge          1                             0    conda-forge
freetype                   2.10.4               h0708190_1    conda-forge
freexl                     1.0.6                h7f98852_0    conda-forge
fribidi                    1.0.10               h36c2ea0_0    conda-forge
fsspec                     2021.10.1          pyhd8ed1ab_0    conda-forge
future                     0.18.2             pyhd8ed1ab_6    conda-forge
gdal                       3.3.3            py39h0494519_2    conda-forge
gdk-pixbuf                 2.42.6               h04a7f16_0    conda-forge
geos                       3.10.0               h9c3ff4c_0    conda-forge
geotiff                    1.7.0                hcfb7246_3    conda-forge
gettext                    0.19.8.1           h73d1719_1008    conda-forge
ghostscript                9.54.0               h27087fc_2    conda-forge
ghp-import                 2.1.0                    pypi_0    pypi
giflib                     5.2.1                h36c2ea0_2    conda-forge
gitdb                      4.0.9                    pypi_0    pypi
gitpython                  3.1.29                   pypi_0    pypi
glib                       2.70.2               h780b84a_4    conda-forge
glib-tools                 2.70.2               h780b84a_4    conda-forge
gmp                        6.2.1                h58526e2_0    conda-forge
gmpy2                      2.1.0b5          py39h78fa15d_0    conda-forge
gmt                        6.2.0                h7e416ca_3    conda-forge
gnuplot                    5.4.3                hedd7fda_0    conda-forge
gnutls                     3.6.13               h85f3911_1    conda-forge
graphicsmagick             1.3.37               hdc87540_0    conda-forge
graphite2                  1.3.13             h58526e2_1001    conda-forge
greenlet                   1.1.2            py39he80948d_0    conda-forge
gshhg-gmt                  2.3.7              ha770c72_1003    conda-forge
gst-plugins-base           1.18.5               hf529b03_3    conda-forge
gstreamer                  1.18.5               h9f60fe5_3    conda-forge
gtk2                       2.24.33              h539f30e_1    conda-forge
h5py                       3.4.0          nompi_py39h7e08c79_101    conda-forge
harfbuzz                   3.1.1                h83ec7ef_0    conda-forge
hdf4                       4.2.15               h10796ff_3    conda-forge
hdf5                       1.12.1         nompi_h2750804_101    conda-forge
heapdict                   1.0.1                      py_0    conda-forge
holoviews                  1.15.2             pyhd8ed1ab_0    conda-forge
```

(continues on next page)

(continued from previous page)

```
hvplot                    0.8.1                    py_0    pyviz
icu                       68.2                h9c3ff4c_0    conda-forge
idna                      3.1               pyhd3deb0d_0    conda-forge
imagecodecs               2021.8.26      py39h571908b_2    conda-forge
imageio                   2.9.0                    py_0    conda-forge
imagesize                 1.4.1                  pypi_0    pypi
importlib-metadata        4.8.1           py39hf3d152e_0    conda-forge
importlib_metadata        4.8.1               hd8ed1ab_1    conda-forge
importlib_resources       5.3.0             pyhd8ed1ab_0    conda-forge
ipykernel                 6.4.2           py39hef51801_0    conda-forge
ipympl                    0.8.2             pyhd8ed1ab_0    conda-forge
ipython                   7.28.0          py39hef51801_0    conda-forge
ipython_genutils          0.2.0                    py_1    conda-forge
ipywidgets                7.6.5             pyhd8ed1ab_0    conda-forge
itsdangerous              2.1.2             pyhd8ed1ab_0    conda-forge
jbig                      2.1               h7f98852_2003    conda-forge
jedi                      0.18.0          py39hf3d152e_2    conda-forge
jinja2                    3.0.2             pyhd8ed1ab_0    conda-forge
joblib                    1.1.0             pyhd8ed1ab_0    conda-forge
jpeg                      9d                  h36c2ea0_0    conda-forge
json-c                    0.15                h98cffda_0    conda-forge
json5                     0.9.5             pyh9f0ad1d_0    conda-forge
jsonschema                3.2.0                  pypi_0    pypi
jupyter-book              0.13.1                 pypi_0    pypi
jupyter-cache             0.4.3                  pypi_0    pypi
jupyter-dash              0.4.2             pyhd8ed1ab_1    conda-forge
jupyter-server-mathjax    0.2.6                  pypi_0    pypi
jupyter-sphinx            0.3.2                  pypi_0    pypi
jupyter_client            7.0.6             pyhd8ed1ab_0    conda-forge
jupyter_core              4.9.1           py39hf3d152e_0    conda-forge
jupyter_server            1.11.1            pyhd8ed1ab_0    conda-forge
jupyter_telemetry         0.1.0             pyhd8ed1ab_1    conda-forge
jupyterhub                1.4.2           py39hf3d152e_0    conda-forge
jupyterhub-base           1.4.2           py39hf3d152e_0    conda-forge
jupyterlab                3.2.1             pyhd8ed1ab_0    conda-forge
jupyterlab-spellchecker   0.7.2                  pypi_0    pypi
jupyterlab_pygments       0.1.2             pyh9f0ad1d_0    conda-forge
jupyterlab_server         2.8.2             pyhd8ed1ab_0    conda-forge
jupyterlab_widgets        1.0.2             pyhd8ed1ab_0    conda-forge
jupytext                  1.14.0            pyheef035f_0    conda-forge
jxrlib                    1.1                 h7f98852_2    conda-forge
kaleido-core              0.2.1               h3644ca4_0    conda-forge
kealib                    1.4.14              h87e4c3c_3    conda-forge
kiwisolver                1.3.2           py39h1a9c180_0    conda-forge
krb5                      1.19.2              hcc1bbae_2    conda-forge
lame                      3.100            h7f98852_1001    conda-forge
latexcodec                2.0.1                  pypi_0    pypi
lcms2                     2.12                hddcbb42_0    conda-forge
ld_impl_linux-64          2.36.1              hea4e1c9_2    conda-forge
lerc                      3.0                 h9c3ff4c_0    conda-forge
libaec                    1.0.6               h9c3ff4c_0    conda-forge
libarchive                3.5.2               hccf745f_1    conda-forge
libblas                   3.9.0          12_linux64_openblas    conda-forge
libbrotlicommon           1.0.9               h7f98852_5    conda-forge
libbrotlidec              1.0.9               h7f98852_5    conda-forge
libbrotlienc              1.0.9               h7f98852_5    conda-forge
```

(continues on next page)

(continued from previous page)

```
libcblas                  3.9.0           12_linux64_openblas    conda-forge
libclang                  11.1.0          default_ha53f305_1    conda-forge
libcurl                   7.79.1               h2574ce0_1    conda-forge
libdap4                   3.20.6               hd7c4107_2    conda-forge
libdeflate                1.8                  h7f98852_0    conda-forge
libedit                   3.1.20191231         he28a2e2_2    conda-forge
libev                     4.33                 h516909a_1    conda-forge
libevent                  2.1.10               h9b69904_4    conda-forge
libffi                    3.4.2                h9c3ff4c_4    conda-forge
libgcc-ng                 11.2.0              h1d223b6_11    conda-forge
libgd                     2.3.3                h6ad9fb6_0    conda-forge
libgdal                   3.3.3                h18e3bf0_2    conda-forge
libgfortran-ng            11.2.0              h69a702a_11    conda-forge
libgfortran5              11.2.0              h5c6108e_11    conda-forge
libglib                   2.70.2               h174f98d_4    conda-forge
libgomp                   11.2.0              h1d223b6_11    conda-forge
libiconv                  1.16                 h516909a_0    conda-forge
libkml                    1.3.0              h238a007_1014    conda-forge
liblapack                 3.9.0           12_linux64_openblas    conda-forge
liblapacke                3.9.0           12_linux64_openblas    conda-forge
libllvm11                 11.1.0               hf817b99_2    conda-forge
libmsym                   0.2.4                    pypi_0    pypi
libnetcdf                 4.8.1           nompi_hb3fd0d9_101    conda-forge
libnghttp2                1.43.0               h812cca2_1    conda-forge
libnsl                    2.0.0                h7f98852_0    conda-forge
libogg                    1.3.4                h7f98852_1    conda-forge
libopenblas               0.3.18          pthreads_h8fe5266_0    conda-forge
libopus                   1.3.1                h7f98852_1    conda-forge
libpng                    1.6.37               h21135ba_2    conda-forge
libpq                     13.5                 hd57d9b9_1    conda-forge
libprotobuf               3.18.1               h780b84a_0    conda-forge
librttopo                 1.1.0                h0ad649c_7    conda-forge
libsodium                 1.0.18               h36c2ea0_1    conda-forge
libsolv                   0.7.19               h780b84a_5    conda-forge
libspatialite             5.0.1               h1d9e4f1_10    conda-forge
libssh2                   1.10.0               ha56f1ee_2    conda-forge
libstdcxx-ng              11.2.0              he4da1e4_11    conda-forge
libtiff                   4.3.0                h6f004c6_2    conda-forge
libuuid                   2.32.1             h7f98852_1000    conda-forge
libuv                     1.41.1               h7f98852_0    conda-forge
libvorbis                 1.3.7                h9c3ff4c_0    conda-forge
libvpx                    1.11.0               h9c3ff4c_3    conda-forge
libwebp                   1.2.1                h3452ae3_0    conda-forge
libwebp-base              1.2.1                h7f98852_0    conda-forge
libxcb                    1.13               h7f98852_1004    conda-forge
libxkbcommon              1.0.3                he3ba5ed_0    conda-forge
libxml2                   2.9.12               h72842e0_0    conda-forge
libzip                    1.8.0                h4de3113_1    conda-forge
libzlib                   1.2.11             h36c2ea0_1013    conda-forge
libzopfli                 1.0.3                h9c3ff4c_0    conda-forge
linkify-it-py             1.0.3                    pypi_0    pypi
llvm-openmp               12.0.1               h4bd325d_1    conda-forge
llvmlite                  0.37.0            py39h1bbdace_0    conda-forge
lmfit                     1.0.3               pyhd8ed1ab_0    conda-forge
locket                    0.2.0                      py_2    conda-forge
lz4-c                     1.9.3                h9c3ff4c_1    conda-forge
```

(continues on next page)

(continued from previous page)

```
lzo                       2.10            h516909a_1000    conda-forge
mako                      1.1.5             pyhd8ed1ab_0    conda-forge
mamba                     0.17.0          py39h951de11_0    conda-forge
markdown                  3.4.1             pyhd8ed1ab_0    conda-forge
markdown-it-py            1.1.0                   pypi_0    pypi
markupsafe                2.0.1           py39h3811e60_0    conda-forge
mathjax                   2.7.7                ha770c72_3    conda-forge
matplotlib-base           3.4.3           py39h2fa2bec_1    conda-forge
matplotlib-inline         0.1.3             pyhd8ed1ab_0    conda-forge
mdit-py-plugins           0.2.8                   pypi_0    pypi
mdurl                     0.1.0             pyhd8ed1ab_0    conda-forge
mistune                   0.8.4           py39h3811e60_1004    conda-forge
mock                      4.0.3           py39hf3d152e_1    conda-forge
mpc                       1.2.1                h9f54685_0    conda-forge
mpfr                      4.1.0                h9202a9a_1    conda-forge
mpmath                    1.2.1             pyhd8ed1ab_0    conda-forge
msgpack-python            1.0.2           py39h1a9c180_1    conda-forge
mysql-common              8.0.27               ha770c72_3    conda-forge
mysql-libs                8.0.27               hfa10184_3    conda-forge
myst-nb                   0.13.2                  pypi_0    pypi
myst-parser               0.15.2                  pypi_0    pypi
nbclassic                 0.3.3             pyhd8ed1ab_0    conda-forge
nbclient                  0.5.4             pyhd8ed1ab_0    conda-forge
nbconvert                 6.2.0           py39hf3d152e_0    conda-forge
nbdime                    3.1.1                   pypi_0    pypi
nbformat                  5.1.3             pyhd8ed1ab_0    conda-forge
ncurses                   6.2                  h58526e2_4    conda-forge
nest-asyncio              1.5.1             pyhd8ed1ab_0    conda-forge
netcdf4                   1.5.8          nompi_py39h64b754b_101    conda-forge
nettle                    3.6                  he412f7d_0    conda-forge
networkx                  2.6.3             pyhd8ed1ab_1    conda-forge
nodejs                    15.14.0              h92b4a50_0    conda-forge
notebook                  6.4.5             pyha770c72_0    conda-forge
nspr                      4.32                 h9c3ff4c_1    conda-forge
nss                       3.72                 hb5efdd6_0    conda-forge
numba                     0.54.1          py39h56b8d98_0    conda-forge
numexpr                   2.7.3           py39hde0f152_0    conda-forge
numpy                     1.20.3          py39hdbf815f_1    conda-forge
oauthlib                  3.1.1             pyhd8ed1ab_0    conda-forge
olefile                   0.46              pyh9f0ad1d_1    conda-forge
openblas                  0.3.18         pthreads_h4748800_0    conda-forge
openh264                  2.1.1                h780b84a_0    conda-forge
openjpeg                  2.4.0                hb52868f_1    conda-forge
openssl                   1.1.1o               h166bdaf_0    conda-forge
packaging                 21.0              pyhd8ed1ab_0    conda-forge
pamela                    1.0.0                     py_0    conda-forge
pandas                    1.3.4           py39hde0f152_0    conda-forge
pandoc                    2.15                 h7f98852_0    conda-forge
pandocfilters             1.5.0             pyhd8ed1ab_0    conda-forge
panel                     0.14.1            pyhd8ed1ab_0    conda-forge
pango                     1.48.10              h54213e6_2    conda-forge
param                     1.12.2            pyh6c4a22f_0    conda-forge
parso                     0.8.2             pyhd8ed1ab_0    conda-forge
partd                     1.2.0             pyhd8ed1ab_0    conda-forge
patsy                     0.5.2             pyhd8ed1ab_0    conda-forge
pcre                      8.45                 h9c3ff4c_0    conda-forge
```

(continues on next page)

(continued from previous page)

```
pemtk                     0.0.1                    pypi_0    pypi
pexpect                   4.8.0              pyh9f0ad1d_2    conda-forge
pickleshare               0.7.5            py39hde42818_1002   conda-forge
pillow                    8.3.2             py39ha612740_0    conda-forge
pip                       21.3.1              pyhd8ed1ab_0    conda-forge
pixman                    0.40.0                h36c2ea0_0    conda-forge
plotly                    5.11.0              pyhd8ed1ab_0    conda-forge
pooch                     1.5.2               pyhd8ed1ab_0    conda-forge
poppler                   21.09.0               ha39eefc_3    conda-forge
poppler-data              0.4.11                hd8ed1ab_0    conda-forge
postgresql                13.5                  h2510834_1    conda-forge
proj                      8.1.1                 h277dcde_2    conda-forge
prometheus_client         0.11.0              pyhd8ed1ab_0    conda-forge
prompt-toolkit            3.0.21              pyha770c72_0    conda-forge
protobuf                  3.18.1            py39he80948d_0    conda-forge
psutil                    5.8.0             py39h3811e60_1    conda-forge
pthread-stubs             0.4                h36c2ea0_1001    conda-forge
ptyprocess                0.7.0               pyhd3deb0d_0    conda-forge
pybtex                    0.24.0                    pypi_0    pypi
pybtex-docutils           1.0.2                     pypi_0    pypi
pycosat                   0.6.3            py39h3811e60_1006   conda-forge
pycparser                 2.20                pyh9f0ad1d_2    conda-forge
pyct                      0.4.6                      py_0    conda-forge
pyct-core                 0.4.6                      py_0    conda-forge
pycurl                    7.44.1            py39h72e3413_0    conda-forge
pydata-sphinx-theme       0.8.1                     pypi_0    pypi
pyerfa                    2.0.0.1           py39hd257fcd_2    conda-forge
pygments                  2.10.0              pyhd8ed1ab_0    conda-forge
pygmt                     0.5.0               pyhd8ed1ab_1    conda-forge
pyjwt                     2.3.0               pyhd8ed1ab_0    conda-forge
pyopenssl                 21.0.0              pyhd8ed1ab_0    conda-forge
pyparsing                 3.0.3               pyhd8ed1ab_0    conda-forge
pyproj                    3.2.1             py39ha81a305_2    conda-forge
pyrsistent                0.17.3            py39h3811e60_2    conda-forge
pyshp                     2.3.1               pyhd8ed1ab_0    conda-forge
pyshtools                 4.10              py39hd777142_0    conda-forge
pysocks                   1.7.1             py39hf3d152e_3    conda-forge
pytables                  3.6.1             py39h2669a42_4    conda-forge
python                    3.9.7           hb7a2778_3_cpython    conda-forge
python-dateutil           2.8.2               pyhd8ed1ab_0    conda-forge
python-json-logger        2.0.1               pyh9f0ad1d_0    conda-forge
python-kaleido            0.2.1               pyhd8ed1ab_0    conda-forge
python_abi                3.9                       2_cp39    conda-forge
pytz                      2021.3              pyhd8ed1ab_0    conda-forge
pyviz_comms               2.2.1               pyhd8ed1ab_1    conda-forge
pywavelets                1.1.1             py39hce5d2b2_3    conda-forge
pyyaml                    6.0               py39h3811e60_0    conda-forge
pyzmq                     22.3.0            py39h37b5a0c_0    conda-forge
qt                        5.12.9                hda022c4_4    conda-forge
quaternion                2022.4.2          py39hd257fcd_0    conda-forge
readline                  8.1                   h46c0cb4_0    conda-forge
reproc                    14.2.3                h7f98852_0    conda-forge
reproc-cpp                14.2.3                h9c3ff4c_0    conda-forge
requests                  2.26.0              pyhd8ed1ab_0    conda-forge
requests-unixsocket       0.2.0                      py_0    conda-forge
retrying                  1.3.3                      py_2    conda-forge
```

(continues on next page)

(continued from previous page)

```
ruamel.yaml                0.17.16          py39h3811e60_0    conda-forge
ruamel.yaml.clib           0.2.2            py39h3811e60_2    conda-forge
ruamel_yaml                0.15.80        py39h3811e60_1004    conda-forge
scikit-image               0.18.3           py39hde0f152_0    conda-forge
scikit-learn               1.0.1            py39h7c5d8c9_1    conda-forge
scipy                      1.7.1            py39hee8e79c_0    conda-forge
scooby                     0.7.0              pyhd8ed1ab_1    conda-forge
seaborn                    0.9.0                     py_2    conda-forge
seaborn-base               0.11.2             pyhd8ed1ab_0    conda-forge
send2trash                 1.8.0              pyhd8ed1ab_0    conda-forge
setuptools                 58.2.0           py39hf3d152e_0    conda-forge
setuptools-scm             6.4.2              pyhd8ed1ab_0    conda-forge
shapely                    1.8.0            py39hc7dd4e9_2    conda-forge
six                        1.16.0             pyh6c4a22f_0    conda-forge
smmap                      5.0.0                    pypi_0    pypi
snappy                     1.1.8                 he1b5a44_3    conda-forge
sniffio                    1.2.0            py39hf3d152e_1    conda-forge
snowballstemmer            2.2.0                    pypi_0    pypi
sortedcontainers           2.4.0              pyhd8ed1ab_0    conda-forge
soupsieve                  2.0.1                     py_1    conda-forge
spherical_functions        2022.4.1           pyhd8ed1ab_1    conda-forge
sphinx                     4.5.0                    pypi_0    pypi
sphinx-book-theme          0.3.3                    pypi_0    pypi
sphinx-comments            0.0.3                    pypi_0    pypi
sphinx-copybutton          0.5.0                    pypi_0    pypi
sphinx-design              0.1.0                    pypi_0    pypi
sphinx-external-toc        0.2.4                    pypi_0    pypi
sphinx-jupyterbook-latex   0.4.7                    pypi_0    pypi
sphinx-multitoc-numbering  0.1.3                    pypi_0    pypi
sphinx-thebe               0.1.2                    pypi_0    pypi
sphinx-togglebutton        0.3.2                    pypi_0    pypi
sphinxcontrib-applehelp    1.0.2                    pypi_0    pypi
sphinxcontrib-bibtex       2.5.0                    pypi_0    pypi
sphinxcontrib-devhelp      1.0.2                    pypi_0    pypi
sphinxcontrib-htmlhelp     2.0.0                    pypi_0    pypi
sphinxcontrib-jsmath       1.0.1                    pypi_0    pypi
sphinxcontrib-qthelp       1.0.3                    pypi_0    pypi
sphinxcontrib-serializinghtml 1.1.5                    pypi_0    pypi
spinsfast                  2022.4.1         py39hc8e5db4_2    conda-forge
sqlalchemy                 1.4.26           py39h3811e60_0    conda-forge
sqlite                     3.36.0               h9cd32fc_2    conda-forge
statsmodels                0.13.0           py39hce5d2b2_0    conda-forge
sympy                      1.9              py39hf3d152e_0    conda-forge
tblib                      1.7.0              pyhd8ed1ab_0    conda-forge
tenacity                   8.1.0              pyhd8ed1ab_0    conda-forge
terminado                  0.12.1           py39hf3d152e_0    conda-forge
testpath                   0.5.0              pyhd8ed1ab_0    conda-forge
threadpoolctl              3.0.0              pyh8a188c0_0    conda-forge
tifffile                   2021.10.12         pyhd8ed1ab_0    conda-forge
tiledb                     2.3.4                 he87e0bf_0    conda-forge
tk                         8.6.11                h27826a3_1    conda-forge
toml                       0.10.2             pyhd8ed1ab_0    conda-forge
tomli                      2.0.1              pyhd8ed1ab_0    conda-forge
toolz                      0.11.1                    py_0    conda-forge
tornado                    6.1              py39h3811e60_1    conda-forge
tqdm                       4.62.3             pyhd8ed1ab_0    conda-forge
```

(continues on next page)

(continued from previous page)

```
traitlets                 5.1.1              pyhd8ed1ab_0    conda-forge
typing_extensions         3.10.0.2           pyha770c72_0    conda-forge
tzcode                    2022a                h166bdaf_0    conda-forge
tzdata                    2021e                he74cb21_0    conda-forge
uc-micro-py               1.0.1                    pypi_0    pypi
uncertainties             3.1.7              pyhd8ed1ab_0    conda-forge
urllib3                   1.26.7             pyhd8ed1ab_0    conda-forge
wcwidth                   0.2.5              pyh9f0ad1d_2    conda-forge
webencodings              0.5.1                      py_1    conda-forge
websocket-client          0.57.0           py39hf3d152e_4    conda-forge
werkzeug                  2.1.2              pyhd8ed1ab_1    conda-forge
wget                      3.2                      pypi_0    pypi
wheel                     0.37.0             pyhd8ed1ab_1    conda-forge
widgetsnbextension        3.5.1            py39hf3d152e_4    conda-forge
x264                      1!161.3030           h7f98852_1    conda-forge
x265                      3.5                  h924138e_3    conda-forge
xarray                    2022.3.0           pyhd8ed1ab_0    conda-forge
xerces-c                  3.2.3                h9d8b166_3    conda-forge
xlrd                      2.0.1              pyhd8ed1ab_3    conda-forge
xorg-kbproto              1.0.7             h7f98852_1002    conda-forge
xorg-libice               1.0.10               h7f98852_0    conda-forge
xorg-libsm                1.2.3             hd9c2040_1000    conda-forge
xorg-libx11               1.7.2                h7f98852_0    conda-forge
xorg-libxau               1.0.9                h7f98852_0    conda-forge
xorg-libxdmcp             1.1.3                h7f98852_0    conda-forge
xorg-libxext              1.3.4                h7f98852_1    conda-forge
xorg-libxrender           0.9.10            h7f98852_1003    conda-forge
xorg-libxt                1.2.1                h7f98852_2    conda-forge
xorg-renderproto          0.11.1            h7f98852_1002    conda-forge
xorg-xextproto            7.3.0             h7f98852_1002    conda-forge
xorg-xproto               7.0.31            h7f98852_1007    conda-forge
xz                        5.2.5                h516909a_1    conda-forge
yaml                      0.2.5                h516909a_0    conda-forge
zeromq                    4.3.4                h9c3ff4c_1    conda-forge
zfp                       0.5.5                h9c3ff4c_7    conda-forge
zict                      2.0.0                      py_0    conda-forge
zipp                      3.6.0              pyhd8ed1ab_0    conda-forge
zlib                      1.2.11            h36c2ea0_1013    conda-forge
zstd                      1.5.0                ha95c52a_0    conda-forge
```

BIBLIOGRAPHY

[1] Paul Hockett. *Quantum Metrology with Photoelectrons, Volume 1: Foundations*. IOP Publishing, 2018. ISBN 978-1-68174-684-5. doi:10.1088/978-1-6817-4684-5.

[2] Paul Hockett. *Quantum Metrology with Photoelectrons, Volume 2: Applications and Advances*. IOP Publishing, 2018. ISBN 978-1-68174-688-3. doi:10.1088/978-1-6817-4688-3.

[3] Paul Hockett. Photoelectron Metrology Toolkit (PEMtk) Github repository. 2021. URL: https://github.com/phockett/PEMtk (visited on 2022-02-18).

[4] Paul Hockett. PEMtk - the Photoelectron Metrology Toolkit - documentation. 2021. URL: https://pemtk.readthedocs.io (visited on 2022-02-18).

[5] Paul Hockett. ePSproc: Post-processing suite for ePolyScat electron-molecule scattering calculations. *Authorea*, 2016. doi:10.6084/m9.figshare.3545639.

[6] Paul Hockett. ePSproc: Post-processing for ePolyScat (Github repository). Github, 2016. doi:10.6084/m9.figshare.3545639.

[7] Paul Hockett. ePSproc: Post-processing for ePolyScat documentation. 2020. URL: https://epsproc.readthedocs.io (visited on 2022-02-18).

[8] Robert R. Lucchese, Kazuo Takatsuka, and Vincent McKoy. Applications of the Schwinger variational principle to electron-molecule collisions and molecular photoionization. *Physics Reports*, 131(3):147–221, January 1986. doi:10.1016/0370-1573(86)90147-X.

[9] F. A. Gianturco, R. R. Lucchese, and N. Sanna. Calculation of low-energy elastic cross sections for electron-CF4 scattering. *The Journal of Chemical Physics*, 100(9):6464, May 1994. doi:10.1063/1.467237.

[10] Alexandra P P Natalense and Robert R Lucchese. Cross section and asymmetry parameter calculation for sulfur 1s photoionization of SF[sub 6]. *The Journal of Chemical Physics*, 111(12):5344, 1999. doi:10.1063/1.479794.

[11] R R Lucchese. ePolyScat User's Manual. URL: https://epolyscat.droppages.com/ (visited on 2022-04-26).

[12] Andrew C. Brown, Gregory S. J. Armstrong, Jakub Benda, Daniel D. A. Clarke, Jack Wragg, Kathryn R. Hamilton, Zdeněk Mašín, Jimena D. Gorfinkiel, and Hugo W. van der Hart. RMT: R-matrix with time-dependence. Solving the semi-relativistic, time-dependent Schrödinger equation for general, multielectron atoms and molecules in intense, ultrashort, arbitrarily polarized laser pulses. *Computer Physics Communications*, 250:107062, May 2020. arXiv:1905.06156, doi:10.1016/j.cpc.2019.107062.

[13] Andrew C. Brown, Gregory S. J. Armstrong, Jakub Benda, Daniel D. A. Clarke, Jack Wragg, Kathryn R. Hamilton, Zdeněk Mašín, Jimena D. Gorfinkiel, and Hugo W. van der Hart. RMT: R-matrix with time-dependence (repository). May 2020. URL: https://gitlab.com/Uk-amor/RMT/rmt (visited on 2022-11-09), arXiv:1905.06156.

[14] Michael W Schmidt, Kim K Baldridge, Jerry A Boatz, Steven T Elbert, Mark S Gordon, Jan H Jensen, Shiro Koseki, Nikita Matsunaga, Kiet A Nguyen, Shujun Su, Theresa L Windus, Michel Dupuis, and John A Montgomery. General

atomic and molecular electronic structure system. *Journal of Computational Chemistry*, 14(11):1347–1363, 1993. doi:10.1002/jcc.540141112.

[15] Mark S. Gordon. Gamess website. URL: http://www.msg.ameslab.gov/gamess/.

[16] Paul Hockett. ePS data: Photoionization calculations archive. 2019. URL: https://phockett.github.io/ePSdata/ (visited on 2022-02-16).

[17] Paul Hockett. ePSdata repositories on Zenodo. 2019. URL: https://zenodo.org/search?page=1&size=20&q=hockett&keywords=Data.

[18] Paul Hockett. Open Photoionization Docker Stacks. URL: https://github.com/phockett/open-photoionization-docker-stacks (visited on 2022-08-04).

[19] C. Yang. On the Angular Distribution in Nuclear Reactions and Coincidence Measurements. *Physical Review*, 74(7):764–772, October 1948. doi:10.1103/PhysRev.74.764.

[20] D Dill. Fixed-molecule photoelectron angular distributions. *The Journal of Chemical Physics*, 65(3):1130–1133, 1976. doi:10.1063/1.433187.

[21] S. L. Altmann and C. J. Bradley. On the Symmetries of Spherical Harmonics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 255(1054):199–215, January 1963. doi:10.1098/rsta.1963.0002.

[22] SL Altmann and AP Cracknell. Lattice harmonics I. Cubic groups. *Reviews of Modern Physics*, 37(1):19–32, 1965. URL: http://rmp.aps.org/abstract/RMP/v37/i1/p19_1 (visited on 2013-06-12).

[23] N Chandra. Photoelectron spectroscopic studies of polyatomic molecules. I. Theory. *Journal of Physics B: Atomic and Molecular Physics*, 20(14):3405–3415, July 1987. doi:10.1088/0022-3700/20/14/013.

[24] Katharine L. Reid and Ivan Powis. Symmetry considerations in molecular photoionization: Fixed molecule photoelectron angular distributions in C3v molecules as observed in photoelectron–photoion coincidence experiments. *The Journal of Chemical Physics*, 100(2):1066, 1994. doi:10.1063/1.466638.

[25] Marcus Johansson and Valera Veryazov. Automatic procedure for generating symmetry adapted wavefunctions. *Journal of Cheminformatics*, 9(1):8, February 2017. doi:10.1186/s13321-017-0193-3.

[26] Marcus Johansson. Libmsym Github. July 2022. URL: https://github.com/mcodev31/libmsym (visited on 2022-08-03).

[27] Mark A. Wieczorek and Matthias Meschede. SHtools Github. SHTOOLS, August 2022. URL: https://github.com/SHTOOLS/SHTOOLS (visited on 2022-08-03).

[28] Mark A. Wieczorek and Matthias Meschede. SHTools: Tools for Working with Spherical Harmonics. *Geochemistry, Geophysics, Geosystems*, 19(8):2574–2592, August 2018. doi:10.1029/2018GC007529.

[29] Tamar Seideman. Time-resolved photoelectron angular distributions: concepts, applications, and directions. *Annual review of physical chemistry*, 53:41–65, January 2002. doi:10.1146/annurev.physchem.53.082101.130051.

[30] Tamar Seideman. Time-resolved photoelectron angular distributions as a probe of coupled polyatomic dynamics. *Physical Review A*, 64(4):042504, September 2001. doi:10.1103/PhysRevA.64.042504.

[31] Christopher Gerry and Peter Knight. *Introductory Quantum Optics*. Cambridge University Press, 2005.

[32] Richard N Zare. *Angular Momentum: Understanding Spatial Aspects in Chemistry and Physics*. John Wiley & Sons, 1988.

[33] Katharine L. Reid, David J. Leahy, and Richard N. Zare. Effect of breaking cylindrical symmetry on photoelectron angular distributions resulting from resonance-enhanced two-photon ionization. *The Journal of Chemical Physics*, 95(3):1746, 1991. doi:10.1063/1.461023.

[34] Guorong Wu, Paul Hockett, and Albert Stolow. Time-resolved photoelectron spectroscopy: from wavepackets to observables. *Physical chemistry chemical physics : PCCP*, 13(41):18447–67, November 2011. doi:10.1039/c1cp22031d.

[35] Yasuki Arasaki, Kazuo Takatsuka, Kwanghsi Wang, and Vincent McKoy. Probing wavepacket dynamics with femtosecond energy- and angle-resolved photoelectron spectroscopy. *Journal of Electron Spectroscopy and Related Phenomena*, 108(1-3):89–98, 2000. doi:DOI: 10.1016/S0368-2048(00)00148-1.

[36] Toshinori Suzuki and Benjamin J. Whitaker. Non-adiabatic effects in chemistry revealed by time-resolved charged-particle imaging. *International Reviews in Physical Chemistry*, 20(3):313–356, July 2001. doi:10.1080/01442350110045046.

[37] Albert Stolow and Jonathan G. Underwood. Time-Resolved Photoelectron Spectroscopy of Non-Adiabatic Dynamics in Polyatomic Molecules. In Stuart A. Rice, editor, *Advances in Chemical Physics*, volume 139, pages 497–584. John Wiley & Sons, Inc., Hoboken, NJ, USA, March 2008. doi:10.1002/9780470259498.ch6.

[38] Katharine L. Reid and Jonathan G. Underwood. Extracting molecular axis alignment from photoelectron angular distributions. *The Journal of Chemical Physics*, 112(8):3643, 2000. doi:10.1063/1.480517.

[39] Jonathan G. Underwood and Katharine L. Reid. Time-resolved photoelectron angular distributions as a probe of intramolecular dynamics: Connecting the molecular frame and the laboratory frame. *The Journal of Chemical Physics*, 113(3):1067, 2000. doi:10.1063/1.481918.

[40] Karl Blum. *Density Matrix Theory and Applications*. Volume 64. Springer Berlin Heidelberg, Berlin, Heidelberg, 3rd editio edition, 2012. ISBN 978-3-642-20560-6. doi:10.1007/978-3-642-20561-3.

[41] Margaret Gregory, Paul Hockett, Albert Stolow, and Varun Makhija. Towards molecular frame photoelectron angular distributions in polyatomic molecules from lab frame coherent rotational wavepacket evolution. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 54(14):145601, July 2021. arXiv:2012.04561, doi:10.1088/1361-6455/ac135f.

[42] P. Hockett, M. Wollenhaupt, C. Lux, and T. Baumert. Complete Photoionization Experiments via Ultrafast Coherent Control with Polarization Multiplexing. *Physical Review Letters*, 112(22):223001, June 2014. doi:10.1103/PhysRevLett.112.223001.

[43] Paul Hockett, Matthias Wollenhaupt, Christian Lux, and Thomas Baumert. Complete photoionization experiments via ultrafast coherent control with polarization multiplexing. II. Numerics and analysis methodologies. *Physical Review A*, 92(1):013411, July 2015. doi:10.1103/PhysRevA.92.013411.

[44] P Hockett, M Wollenhaupt, and T Baumert. Coherent control of photoelectron wavepacket angular interferograms. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 48(21):214004, November 2015. doi:10.1088/0953-4075/48/21/214004.