User Guide

Pyth-MPG version 1.0.0

June 2026

Andrea Urru,[1] Turan Birol,[2], Trey Cole[3], and David Vanderbilt[3]

[1] Department of Physics, University of Cagliari, Italy

[2] Department of Chemical Engineering and Materials Science, University of Minnesota, USA

[3] Department of Physics and Astronomy, Rutgers University, USA

1 General Information

The Pyth-MPG project provides two components:

- A selection of spreadsheets allowing the user to identify which of the 122 magnetic point groups (MPGs) allow certain physical properties to be present based on symmetry considerations. Several spreadsheets are provided with pre-selected sets of physical properties.
- A Python package PYTHMPG that allows the user to create similar spreadsheets that cover a modified or expanded set of physical properties. Any scalar, vector, or tensor property that can be described by a Jahn symbol (see below) can be included at the option of the user.

Distribution

- The available pre-constructed spreadsheets and the present User Guide are provided in the Zenodo repository zenodo.org/records/18672613. If you wish to share the spreadsheets with co-workers and colleagues, we kindly ask you to do so by sharing the link to the Zenodo repository, rather than by sharing the materials directly.
- The Python package PYTHMPG itself is available at github.com/pythmpg/pythmpg. A zipped copy of the distribution is also available on the Zenodo repository.

Citation

If you find the spreadsheets or the PYTHMPG package useful, please cite the project by referring to doi.org/10.5281/zenodo.18672613. A sample bibtex file is as follows.

```
@software{Python-MPG-2026,
author = {Urru, Andrea and Birol, Turan and Cole, Trey and Vanderbilt, David},
doi = {10.5281/zenodo.18672613},
license = {GPL-3.0-or-later},
month = jun,
title = {{Python Magnetic Point Group (PythMPG)}},
url = {https://zenodo.org/records/18672613},
version = {2.0.0},
year = {2026}}
```

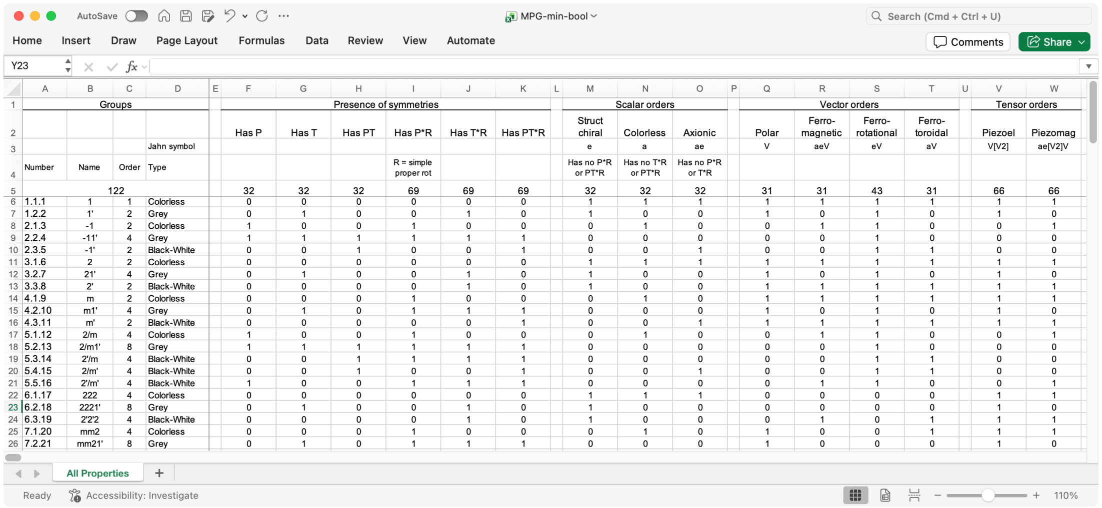
Contact

If you have any questions or wish to report an issue, our contact information is as follows.

- Andrea Urru <aurru@dsf.unica.it>
- Turan Birol <tbirol@umn.edu>
- Trey Cole <bc889@physics.rutgers.edu>
- David Vanderbilt <dhv@physics.rutgers.edu>

Acknowledgments

The project is inspired by the MTENSOR utility [1] of the Bilbao Crystallographic Server [2, 3, 4]. Support was provided by NSF Grant DMR-2421895.



	A	B	C	D	F	G	H	I	J	K	M	N	O	Q	R	S	T	V	W
1	Groups				Presence of symmetries						Scalar orders			Vector orders				Tensor orders	
2					Has P	Has T	Has PT	Has P*R	Has T*R	Has PT*R	Struct chiral	Colorless	Axionic	Polar	Ferro-magnetic	Ferro-rotational	Ferro-toroidal	Piezoel	Piezomag
3				Jahn symbol							e	a	ae	V	aeV	eV	aV	V[V2]	ae[V2]V
4	Number	Name	Order	Type				R = simple proper rot			Has no P*R or PT*R	Has no T*R or PT*R	Has no P*R or T*R						
5		122			32	32	32	69	69	69	32	32	32	31	31	43	31	66	66
6	1.1.1	1	1	Colorless	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
7	1.2.2	1'	2	Grey	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0
8	2.1.3	-1	2	Colorless	1	0	0	1	0	0	0	1	0	0	1	1	0	0	1
9	2.2.4	-11'	4	Grey	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0
10	2.3.5	-1'	2	Black-White	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0
11	3.1.6	2	2	Colorless	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
12	3.2.7	21'	4	Grey	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0
13	3.3.8	2'	2	Black-White	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1
14	4.1.9	m	2	Colorless	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1
15	4.2.10	m1'	4	Grey	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0
16	4.3.11	m'	2	Black-White	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1
17	5.1.12	2/m	4	Colorless	1	0	0	1	0	0	0	1	0	0	1	1	0	0	1
18	5.2.13	2/m1'	8	Grey	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0
19	5.3.14	2'/m	4	Black-White	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0
20	5.4.15	2/m'	4	Black-White	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0
21	5.5.16	2'/m'	4	Black-White	1	0	0	1	1	1	0	0	0	0	1	1	0	0	1
22	6.1.17	222	4	Colorless	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
23	6.2.18	2221'	8	Grey	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0
24	6.3.19	2'2'2	4	Black-White	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1
25	7.1.20	mm2	4	Colorless	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1
26	7.2.21	mm21'	8	Grey	0	1	0	1	1	1	0	0	0	1	0	0	0	1	0

Figure 1: Spreadsheet `MPG-min-bool`. Only a subset of rows is shown for clarity.

2 Introduction

2.1 Physical properties as described by Cartesian tensors

Suppose we would like to know which magnetic point groups (MPGs) allow for materials that are non-polar and lack a net magnetization, but are nevertheless piezomagnetic. This is the kind of question that the present package of tools is designed to answer. Note that piezomagnetism is a third-rank tensor, describing the linear response of magnetization (a 3-vector, i.e., a first-rank tensor) to strain (a 3×3 matrix, i.e., a second-rank tensor).[1]

We can find the answer in the spreadsheet reproduced in Fig. 1. Columns A-D show some information about each MPG, and a '1' in Columns F-K indicates the *presence* of a symmetry, as described below. Instead, starting in Column M, a '1' implies enough *absence* of symmetry to allow for the existence of a physical quantity (observable, order parameter, or response property). So, we look for a '0' in Columns Q and R to rule out materials that are polar or have a net magnetization, but insist on a '1' in Column W for piezomagnetism. After sorting on columns and deleting rejected rows, we find a list of 27 MPGs that satisfy these criteria. The results are shown in Fig. 2 on the next page. There we also see, by scanning Row 5, that of these, 8 are structurally chiral, 16 are piezoelectric, etc.

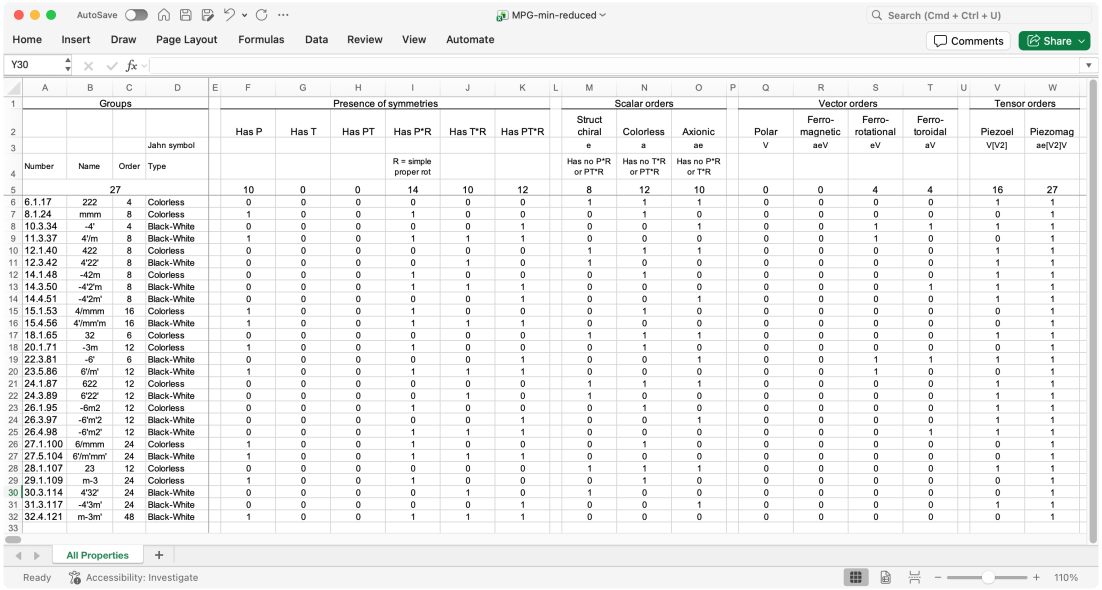
Tensors may be of any integer rank. Those of interest usually extend no higher than rank six, but the present package has no limit on the allowed rank. At the other extreme are vectors (rank 1) and scalars (rank 0), which are also understood to be tensors in our terminology.

2.2 Jahn symbols

The rank of a tensor, together with its symmetry properties, are identified with a compact notation, known as the *Jahn symbol* [5]. The elements that make up a Jahn symbol are the following:

- A number of "V" characters equal to the rank of the tensor. For tensors of rank higher than 1, the "V" characters are written in a compact way using a power-like notation.

[1] Eqivalently, by reciprocity, it also corresponds to the response of stress to applied magnetic field.



Groups				Presence of symmetries						Scalar orders			Vector orders				Tensor orders	
				Has P	Has T	Has PT	Has P*R	Has T*R	Has PT*R	Struct chiral	Colorless	Axionic	Polar	Ferro-magnetic	Ferro-rotational	Ferro-toroidal	Piezoel	Piezomag
			Jahn symbol							e	a	ae	V	aeV	eV	aV	V[V2]	ae[V2]V
Number	Name	Order	Type				R = simple proper rot			Has no P*R or PT*R	Has no T*R or PT*R	Has no P*R or T*R						
27				10	0	0	14	10	12	8	12	10	0	0	4	4	16	27
6.1.17	222	4	Colorless	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
8.1.24	mmm	8	Colorless	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
10.3.34	-4'	4	Black-White	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1
11.3.37	4'/m	8	Black-White	1	0	0	1	1	1	0	0	0	0	0	1	0	0	1
12.1.40	422	8	Colorless	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
12.3.42	4'22'	8	Black-White	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1
14.1.48	-42m	8	Colorless	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1
14.3.50	-4'2'm	8	Black-White	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1
14.4.51	-4'2m'	8	Black-White	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1
15.1.53	4/mmm	16	Colorless	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
15.4.56	4'/mm'm	16	Black-White	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1
18.1.65	32	6	Colorless	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
20.1.71	-3m	12	Colorless	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
22.3.81	-6'	6	Black-White	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1
23.5.86	6'/m'	12	Black-White	1	0	0	1	1	1	0	0	0	0	0	1	0	0	1
24.1.87	622	12	Colorless	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
24.3.89	6'22'	12	Black-White	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1
26.1.95	-6m2	12	Colorless	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1
26.3.97	-6'm'2	12	Black-White	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1
26.4.98	-6'm2'	12	Black-White	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1
27.1.100	6/mmm	24	Colorless	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
27.5.104	6'/m'mm'	24	Black-White	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1
28.1.107	23	12	Colorless	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
29.1.109	m-3	24	Colorless	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
30.3.114	4'32'	24	Black-White	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
31.3.117	-4'3m'	24	Black-White	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1
32.4.121	m-3m'	48	Black-White	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1

Figure 2: Spreadsheet `MPG-min-bool` after sorting and keeping only MPGs with piezomagnetism in the absence of polarization and magnetization.

For example, V^2 is a rank-2 tensor, V^3 is a rank-3 tensor, and so on. More complicated cases exist when additional characters appear, such as for the rank-3 tensor $a[V^2]V$, as explained below.

- Square brackets "[" and curly brackets "{" identify symmetrization and antisymmetrization under permutation of the enclosed group of indices, respectively. (Note that this convention is reversed with respect to quantum commutators and anticommutators.) As an example, $[V^2]$ and $\{V^2\}$ are rank-2 tensors symmetric ($T_{ij} = T_{ji}$) and antisymmetric ($T_{ij} = -T_{ji}$) under exchange of their two indices, respectively. Similarly, Jahn symbol $[V^2]V$ identifies a rank-3 tensor symmetric under exchange of its first two indices, i.e., $T_{ijk} = T_{jik}$. Symmetrization and antisymmetrization can also apply to larger sets of indices. For example, $[V^4]$ identifies a rank-4 tensor that is symmetric under exchange of any pair of its four indices.

- Characters "a" and "e" describe the transformation properties of a tensor under time-reversed and improper rotations respectively. Specifically:

  - "a" identifies a tensor that is odd under time reversal. If no "a" is present, the tensor is even under time reversal.
  - "e" identifies an *axial* tensor. If no "e" is present, the tensor is *proper*. The terms proper and axial refer to how the tensor transforms under improper point symmetry operations $\mathcal{S}$, i.e., those with $\det \mathcal{S} = -1$. In general, a proper rank-n tensor transforms under $\mathcal{S}$ as

$$T_{ijk\dots z} = \mathcal{S}_{ii'}\mathcal{S}_{jj'}\mathcal{S}_{kk'}\dots\mathcal{S}_{zz'}T_{i'j'k'\dots z'}, \quad (1)$$

(implied sum over indices) whereas an axial tensor transforms as

$$T_{ijk\dots z} = (\det \mathcal{S})\mathcal{S}_{ii'}\mathcal{S}_{jj'}\mathcal{S}_{kk'}\dots\mathcal{S}_{zz'}T_{i'j'k'\dots z'}. \quad (2)$$

The two expressions are the same for proper symmetry operations ($\det \mathcal{S} = 1$), but differ in sign for improper symmetry operations ($\det \mathcal{S} = -1$).

As an example, V is a time-even proper vector (e.g., position $\mathbf{r}$), aV is a time-odd proper vector (e.g., velocity $\mathbf{v}$), eV is a time-even axial vector (e.g., torque $\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}$), and aeV is a time-odd axial vector (e.g., magnetic field $\mathbf{B}$).

Jahn symbols are given for many tensor properties of interest in the Tables in Ref. [1], and others can be found by navigating to the `MTENSOR` utility of the Bilbao Crystallographic Server and clicking on the links for "equilibrium," "optical," "nonlinear optical susceptibility," and "transport" tensors. `MTENSOR` is found under "Magnetic Symmetry and Applications" in the Bilbao Crystallographic Server.

2.3 Spreadsheet structure

This package contains three minimal spreadsheets, named `MPG-min-bool`, `MPG-min-num`, and `MPG-exten`. Fig. 1 shows, as an example, a screenshot of `MPG-min-bool`. All three spreadsheets have a set of columns (A-K) in common, which report general information about the MPG and the presence of specific symmetries. In particular:

- Columns A-D: MPG information. Column A $\rightarrow$ BNS (Belov-Neronova-Smirnova) notation [6]; Column B $\rightarrow$ HM (Hermann–Mauguin) notation; Column C $\rightarrow$ group order (i.e., number of symmetry elements in the MPG); Column D $\rightarrow$ type (colorless, grey, black-white).
- Columns F-K: Presence or absence of specified symmetry elements in the MPG. The presence of a symmetry is identified by a boolean variable (0 = symmetry is absent, 1 = symmetry is present). The first three of these columns indicate the existence or not of $\mathcal{P}$, $\mathcal{T}$, and $\mathcal{PT}$ in columns F, G, and H respectively, where $\mathcal{P}$ is parity (inversion) and $\mathcal{T}$ is time reversal. The next three columns indicate the presence of any symmetry of the form $\mathcal{PR}$ (column I), $\mathcal{TR}$ (column J), or $\mathcal{PTR}$ (column K), where the additional $\mathcal{R}$ indicates composition with a simple (proper, not time-reversed) rotation.

From column M onwards the spreadsheets report, for each MPG, information about the presence of specific orders and material responses. Each order (or response) is identified by its Jahn symbol, which we introduced in the previous Section. In particular, the "minimal" spreadsheets `MPG-min-bool` and `MPG-min-num` consider the following orders (the Jahn symbol for each order is given inside parentheses):

- Scalar orders:
  - Chiral order (e, i.e., odd under inversion). Such order is present in MPGs that lack both $\mathcal{PR}$ and $\mathcal{PTR}$ symmetries;[2]
  - "colorless" order (a, odd under time-reversal). Such order is present in MPGs that lack both $\mathcal{TR}$ and $\mathcal{PTR}$ symmetries, i.e., Colorless MPGs;
  - Axionic order (ae, odd under both inversion and time-reversal, hence even under $\mathcal{PT}$). Such order is present in MPGs that lack both $\mathcal{PR}$ and $\mathcal{TR}$ symmetries.
- Vector orders:

[2] The chiral order reported in this column corresponds to "structural" (or "geometric") chirality, which is determined by the absence of improper rotations when ignoring any spin order that may be present. This excludes 21 MPGs that have purely magnetic chirality, i.e., those for which columns IJK are 001 respectively.

– Polar order (V);
– Ferromagnetic order (aeV);
– Ferrorotational order (eV);
– Ferrotoroidal order (aV).

• Higher-rank orders:

– Piezoelectric response ($V[V^2]$);
– Piezomagnetic response ($ae[V^2]V$).

Instead, the spreadsheet `MPG_exten` considers the same vector orders listed above, but with an extended set of higher-rank responses, specifically:

• Higher-rank orders:

– Piezomagnetic response and linear magnetoresistance ($ae[V^2]V$);
– Natural optical activity ($\{V^2\}V$);
– Gyrotropic birefringence ($a[V^2]V$);
– Ordinary Hall and ordinary Faraday responses ($e\{V^2\}V$);
– Spontaneous Ettingshausen effect ($a\{V^2\}$).

Note that `MPG-min-bool` reports a boolean variable to identify whether a given order / response is present (0 = absent, 1 = present). `MPG-min-num` is identical to `MPG-min-bool` except that it reports the number of independent symmetry-allowed components for each tensor. `MPG-exten` reports the number of independent symmetry-allowed components, similarly to `MPG-min-num`.

Finally, also note that each spreadsheet contains metadata that can be accessed by going to File → Properties → Description. This metadata includes the name of the project, the authors, the Zenodo link location, and the conditions of use. Please ensure that this metadata is not lost in the process of modifying the spreadsheets.

2.4 Source of the data

The information in the cells of these spreadsheets is obtained from the PYTHMPG Python code package described in Sec. 3. No knowledge of Python, or of the inner workings of the PYTHMPG package, is required for ordinary use of the spreadsheets provided.

2.5 Hints for using the spreadsheets

Because the information is presented in the form of an Excel spreadsheet, all of the usual tools associated with such spreadsheets can be applied, including sorting on a row or column, hiding some rows or columns, and the like.

We assume here that the user is employing one of the standard applications for manipulating Excel spreadsheets such as Microsoft Excel itself, Apple Numbers, or LibreOffice Calc. We avoid giving detailed instructions that require the use of a particular application in what follows.

In each spreadsheet provided, note that Rows 1-5 and Columns A-D are "frozen," meaning that the sliders at the bottom and right edges of the window do not affect these rows and columns. Also, note that the Name Box (just above Column A, showing the current range of selected cells) has a pull-down menu containing "named ranges." One of these is "Contents," which selects and highlights all rows below the frozen rows. This is particularly useful when sorting by column.

Sorting on columns. To sort on Column N ("Ferromagnetic"), for example, select "Contents" from the Name Box, Select "Data," select "Sort," and choose to sort increasing or decreasing on Column N. Optionally, you can then hide or delete all rows containing zero or nonzero entries in Column N, and then hide or delete Column N. Repeat to sort on multiple columns, or use advanced sorting methods to sort on multiple columns at once.

Hiding or deleting columns or rows. To hide or delete a column or range of columns, select the column(s) by clicking at the very top of the column (above Row 1), then right-click and choose "delete" or "hide."

To hide or delete a row or range of rows, select the row(s) by clicking at the very left of the row (before Column A), then right-click and choose "delete" or "hide." Note: Row 5 contains formulas that count the number of nonzero entries below. These numbers conveniently update if rows are deleted, but not if they are only hidden, in which case the hidden rows still contribute to the totals shown in Row 5.

Save personalized spreadsheet. After condensing the spreadsheet by deleting or hiding columns and rows with a particular set of applications in mind, you may wish to rename and save your spreadsheet for future use.

3 The PYTHMPG package

The PYTHMPG python package provides tools for enumerating symmetry properties of all 122 magnetic point groups (MPGs) and to count the number of independent components of arbitrary-rank tensors under those symmetries, as classified by their Jahn symbol.

A major feature of the package is its ability to export data in the form of a `.csv` file that can be used to build a spreadsheet capable of screening for MPGs based on whether they have certain symmetries or support specied tensor properties. A broader community of users can then use standard spreadsheet tools, such as sorting on columns and hiding columns and rows, to achieve similar ends, without the need to access the python codes themselves.

3.1 Installation and documentation

The PYTHMPG package is available at PyPI, so that it can be installed just by issuing

```
pip install pythmpg
```

The full code package is available from its GitHub repository github.com/pythmpg/pythmpg, which is also available in the Zenodo repository zenodo.org/records/18672613 in the form of a `.zip` archive. More detailed installation instructions are provided in the repository. Documentation of the code package is hosted at pythmpg.readthedocs.io.

3.2 User-level python script

The standard way of accessing the PYTHMPG tools to create a spreadsheet is to write a user-level script that imports the `Spreadsheet` class from the PYTHMPG package. A few example scripts are provided; here we begin by working through the `mpg_min.py` script, which is intended as a minimal example. This script begins:

```
#!/usr/bin/env python3

# ------------------------------------------------------
# Example user program to create a minimal spreadsheet
# ------------------------------------------------------

# import Spreadsheet class
from pythmpg.spreadsheet import Spreadsheet

# Initialize my spreadsheet
my_sheet = Spreadsheet()
```

This imports the `Spreadsheet` class from the PYTHMPG package and creates `my_sheet` as an object belonging to this class. At the end of the script, this object will be used to write a `.csv` file that can be cut-and-pasted into a regular spreadsheet (Excel or similar). Logically, the first four rows correspond to header information, and the remaining rows run over the 122 MPGs.

The columns are structured to be divided into "sections," with several columns devoted to each section. (Sections will be separated by narrow empty columns in the eventual spreadsheet for clarity of presentation.) Upon creation, `my_sheet` already has four default sections in place, as summarized by the comment lines that come next:

```
# Default sections are
#   'groups'
#     Identification info on MPGs
#   'symm'
#     Presence or absence of six types of symmetry
#   'scalar'
#     Chiral              e
#     Colorless           a
#     Axionic             ae
#   'vector'
#     Polar               V
#     Ferro- magnetic     aeV
#     Ferro- rotational   eV
#     Ferro- toroidal     aV

# Print the default sections and their headers
# my_sheet.header_report()
```

The "groups" section identifies the MPG in two notations, gives the order of the MPG (number of operations), and MPG type ("black-white," "grey," or "colorless"), while the "symm" section specifies the presence or absence of six types of symmetries ($\mathcal{P}$, $\mathcal{T}$, and $\mathcal{PT}$, without or with a proper rotation). These two sections should generally not be modified.

Next come one or more groups of tensor properties that can be specified by Jahn symbols appearing to the right of the names. (Here "tensor"is a general term that includes rank-zero scalars, rank-1 vectors, and tensors of rank 2 and higher.) The default configuration, initialized when the `Spreadsheet` object `my_sheet` is created, has two sections, one for the three scalar Jahn symbols and one for the four vector Jahn symbols.

The `# my_sheet.header_report()` line is commented above, but if uncommented, it prints a summary of the defaults that looks like this:

```
=============
Header Report
=============

Header Sections: ['groups', 'symm', 'scalar', 'vector']
-------------------------------------------------------------
```

```
groups:
-----------------------------------------------------------
                                      Number
                                      Name
                                      Order
                        Jahn symbol   Type
-----------------------------------------------------------
symm:
-----------------------------------------------------------
  Has P
  Has T
  Has PT
  Has P*R                             R = simple proper rot
  Has T*R
  Has PT*R
-----------------------------------------------------------
scalar:
-----------------------------------------------------------
  Struct chiral        e              Has no P*R or PT*R
  Colorless            a              Has no T*R or PT*R
  Axionic              ae             Has no P*R or T*R
-----------------------------------------------------------
vector:
-----------------------------------------------------------
  Polar                V
  Ferro- magnetic      aeV
  Ferro- rotational    eV
  Ferro- toroidal      aV
-----------------------------------------------------------
```

Next, the example script modifies the defaults by adding a new section called 'user' which includes some second- and third-rank tensors.

```
# Add a new section of tensor orders
tensor_section = [
    ('Piezoel','V[V2]'),
    ('Piezomag','ae[V2]V') ]

my_sheet.add('tensor',tensor_section)

# Print the modified sections and their headers
my_sheet.header_report()
```

In each tuple enclosed in parentheses, the first item is a tensor property name, the second item is the Jahn symbol, and the optional third item is a comment or the name of another tensor property sharing the same Jahn symbol. The header report now appears as above except that the new 'tensor' section has been added:

```
-----------------------------------------------------------
tensor:
-----------------------------------------------------------
  Piezoel              V[V2]
  Piezomag             ae[V2]V
-----------------------------------------------------------
```

Finally, the remainder of the script is

```
# Print the modified sections and their headers
my_sheet.header_report()
```

```
# Print the details of the parsing of Jahn symbols
# my_sheet.jahn_report()

# Analyze tensors for each MPG
my_sheet.build_csv()

# Write csv files to disk
my_sheet.write_csv('mpg_min_bool.csv')
my_sheet.write_csv('mpg_min_num.csv',kind='num')
```

Here an option to print details of the parsing of the Jahn symbols is bypassed, the spreadsheet is built using the `build_csv` method, and the `.csv` file is written to the local directory with an entry for each combination of MPG and Jahn symbol, first with Boolean entries (0 or 1) and then again with numerical entries (number of independent tensors).

The `Spreadsheet` class also has methods for deleting or modifying sections of the spreadsheet. See the online documentation of the `spreadsheet` module for details. Any tensor that can be described by a valid Jahn symbol can be included in a section.

3.3 Constructing the spreadsheet from the `.csv` file

These directions apply to creating `MPG-min-bool.xlsx` starting from `mpg_min_bool.csv`, but the same approach should apply to other cases. (Note that we prefer underscores and lower-case letters in `python` names, including input and output files, while switching to dashes and some upper-case letters at the spreadsheet level.)

Before you begin, you should be aware of a feature in `MPG-template.xlsx` which will also be inherited by `MPG-min-bool.xlsx`. Columns beyond Column D, and rows under Row 5, have had the "Freeze Panes" feature applied. This means that when scrolling up-and-down or left-and-right in the data area of the window, Columns A-D and Rows 1-5 remain visible.

Follow these steps, saving periodically:

1. Copy `MPG-template.xlsx` to become `MPG-min-bool.xlsx`.

2. In Excel, open `mpg_min_bool.csv` and `MPG-min-bool.xlsx`.

3. To create the header rows, copy Rows 1-4 from `mpg_min_bool.csv` and paste using `Paste and Match Formatting`[3] into Rows 1-4 of `MPG-min-bool.xlsx`. The special paste operation is needed to preserve font and row height information in the target document.

   It is also recommended to add a bottom border line to each merged cell.

4. Again using special paste operation, copy Rows 5-126 from `mpg_min_bool.csv` and paste into Rows 6-127 of `MPG-min-bool.xlsx`, leaving Row 5 untouched. (You should be able to just highlight Row 6 in the target and proceed with the paste.)

5. In Row 1, merge cells belonging to each section to become a single wider cell with a centered label of your choice. For example, merge cells A1-D1 and leave a centered label "Groups"; merge F1-K1 and label as "Presence of symmetry"; merge M1-O1 and label as "Scalar orders"; etc.

6. Make adjustments to Row 5, where each cell contains a formula summing over the number of nonzero entries in the rows below. If there is an entry in Row 5 for an empty column, such as for L, P, and U in the present case, delete it. Also delete the extra ones if the

[3] This applies when working in Excel. In LibreOffice, use `Paste Special` and select `Values Only`. In Numbers, use `Paste and Match Style`.

template had too many; while if it had too few, copy and paste the last one into the cells to its right as needed.

7. Reduce the width of any empty columns to be the same narrow width as Column E. These are spacer columns for visual aesthetics. (You can delete them if you prefer.)

8. In Excel, the Name Box (located just above the column label 'A' and containing the location of the currently highlighted cell) can be used to give names to cell ranges. We have predefined two: 'Contents' covering only data rows (below Row 5); and 'Print Area' covering headers as well as data. To find these, click on the right end of the Name Box, and if needed, adjust these predefined ranges to cover the correct number of columns for your spreadsheet. (Range 'Contents' is useful when sorting a spreadsheet, as discussed in Sec. 2.5 above.)

3.4 Description of the PYTHMPG package

In brief, the code starts with the list of 122 MPGs and parses them to create a list of generators for each group (and to record the BNS number). It then uses that information to fill the data in Columns A-D and F-K. Furthermore, it parses the Jahn symbols given in Row 3 beginning in Column M, and determines the number of independent degrees of freedom allowed for a tensor of that form in each MPG. In the case of Boolean spreadsheets, this is converted to a 1 or 0 depending on whether or not the tensor is allowed to have any nonzero components.

The package is composed of five modules as follows:

- `spreadsheet`
  Interface with user-level scripts for generating MPG spreadsheets. This is usually the only module that needs to be accessed by the user when constructing spreadsheets. This module calls the lower-level modules enumerated below.

- `mpg_tools`
  Provide functions for querying magnetic point group properties for given Jahn symbols. It is called from `pythmpg.spreadsheet`, but some of its functions can also be queried directly from a user script via "`from pythmpg.mpg_tools import get_mpg_info`" or "`from pythmpg.mpg_tools import get_num_indep`." This module calls lower-level modules `parse_jahn` and `mpg_dicts`.

- `parse_jahn`
  Parse Jahn symbols by converting each one into an instruction set for applying symmetrization and antisymmetrization over indices, as well as keeping track of parity under space and time.

- `mpg_dicts`
  Construct dictionaries describing the operations and multiplicative structure of magnetic point groups. This module calls lower-level module `pg_elements`.

- `pg_elements`
  Define crystallographic proper rotations and build their multiplication tables.

For details on the code, see the documentation API posted at pythmpg.readthedocs.io.

References

[1] S. V. Gallego, J. Etxebarria, L. Elcoro, E. S. Tasci, and J. M. Perez-Mato. Automatic calculation of symmetry-adapted tensors in magnetic and non-magnetic materials: a new tool of the Bilbao Crystallographic Server. *Acta Crystallographica Section A*, 75(3):438–447, May 2019.

[2] M. I. Aroyo, J. M. Perez-Mato, D. Orobengoa, E. Tasci, G. De La Flor, and A. Kirov. Crystallography online: Bilbao crystallographic server. *Bulg. Chem. Commun.*, 43(2):183 – 197, 2011.

[3] M. I. Aroyo, J. M. Perez-Mato, C. Capillas, E. Kroumova, S. Ivantchev, G. Madariaga, A. Kirov, and H. Wondratschek. Bilbao crystallographic server: I. databases and crystallographic computing programs. *Z. Kristallogr.*, 221(1):15–27, 2006.

[4] M. I. Aroyo, A. Kirov, C. Capillas, J. M. Perez-Mato, and H. Wondratschek. Bilbao Crystallographic Server. II. Representations of crystallographic point groups and space groups. *Acta Crystallogr. Sect. A*, 62(2):115–128, Mar 2006.

[5] H. A. Jahn. Note on the Bhagavantam–Suranarayana method of enumerating the physical constants of crystals. *Acta Crystallographica*, 2:30–33, March 1949.

[6] N. V. Belov, K. N. Neronova, and T. S. Smirnova. Note on the Bhagavantam–Suranarayana method of enumerating the physical constants of crystals. *Soviet Physics Crystallography*, 2:311–322, 1957.