Improving reproducibility in building simulation: a pure-Python approach to geometry creation

Steven Firth

Loughborough University

2021-06-23

My background

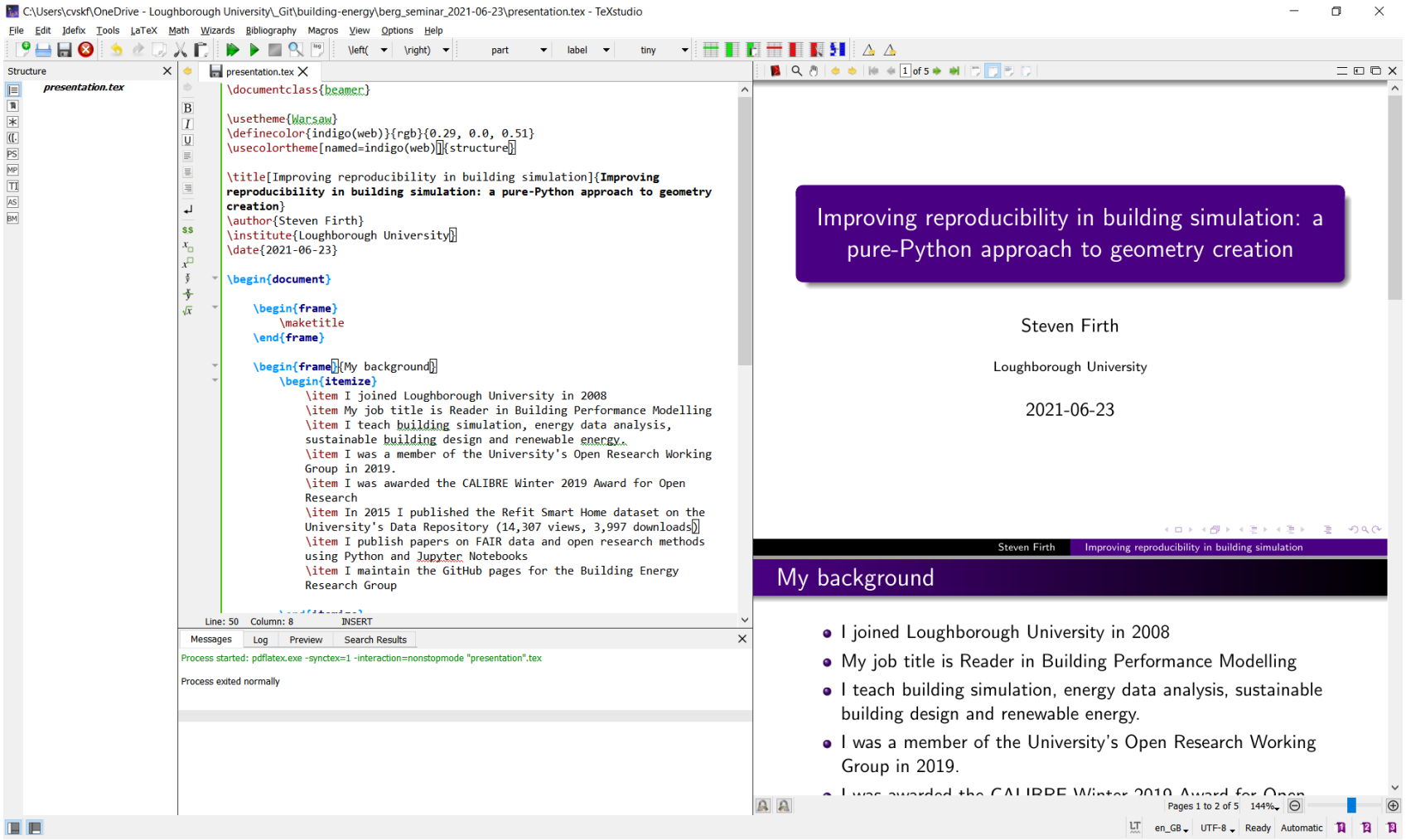
- I joined Loughborough University in 2008
- My job title is Reader in Building Performance Modelling
- I teach building simulation, energy data analysis, sustainable building design and renewable energy.
- I was a member of the University's Open Research Working Group in 2019.
- I was awarded the CALIBRE Winter 2019 Award for Open Research
- In 2015 I published the Refit Smart Home dataset on the University's Data Repository (14,307 views, 3,997 downloads)
- I publish papers on FAIR data and open research methods using Python and Jupyter Notebooks
- I maintain the GitHub pages for the Building Energy Research Group

Point-and-click vs. Text-based commands

Research task	Point-and-Click	Text-based Commands
Writing documents	Microsoft Word	LaTeX
Creating slides	Microsoft PowerPoint	LaTeX
Analysing data	Microsoft Excel	Python and Jupyter Notebooks
Building websites	Adobe Dreamweaver	HTML, Bootstrap, Django

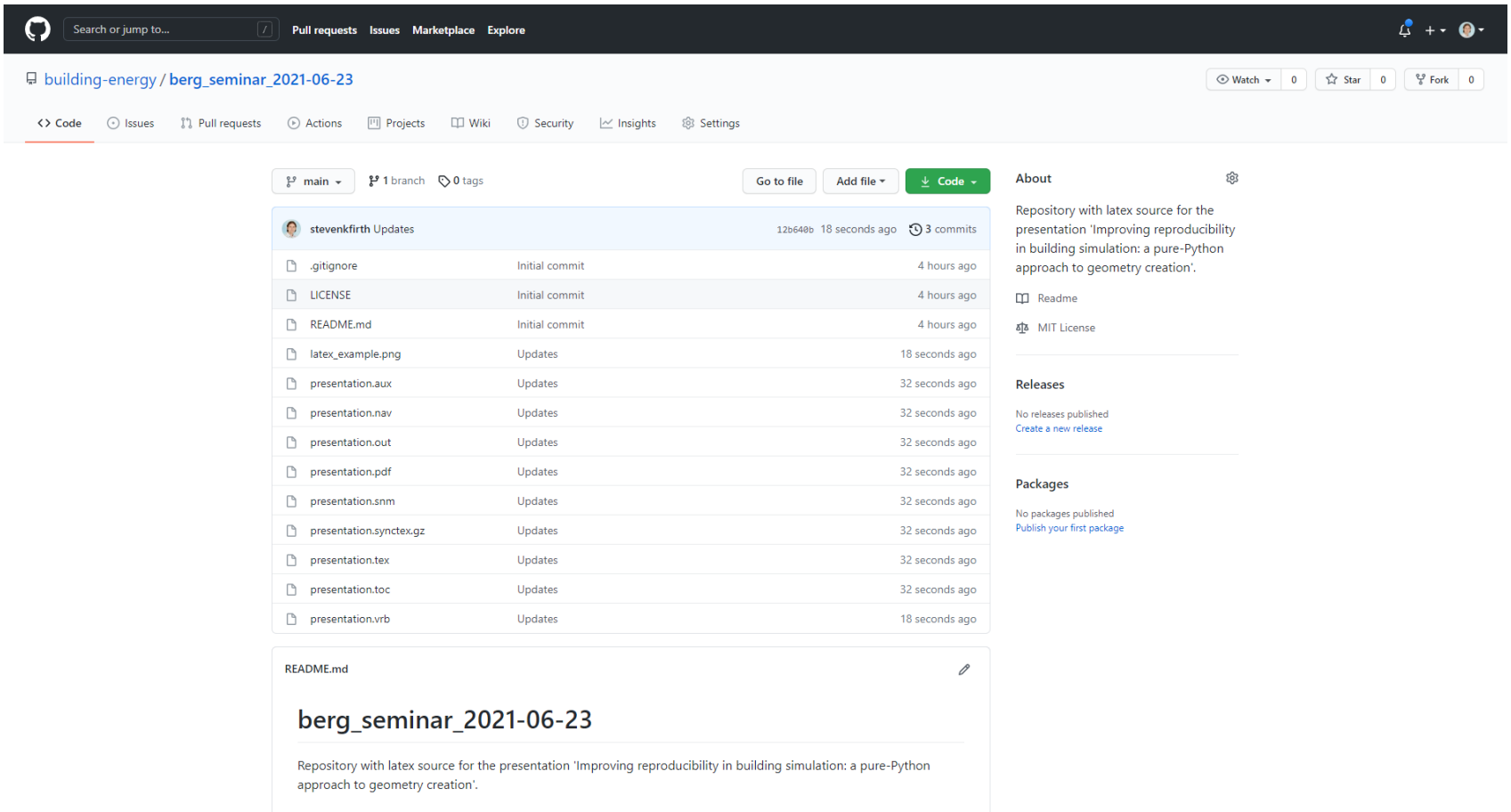
Example #1: A Reproducible Presentation

This presentation is reproducible as it is written in code (Latex)



Example #1: A Reproducible Presentation

This presentation is also open as the code is hosted on the BERG Github repository



The problem I am trying to solve

- **Task:** I would like *to construct a building simulation model* of a building and *to simulate the energy performance* of the building using the EnergyPlus software.
- **Challenge:** I would like to do this in an *open*, *transparent* way so that the whole process is *reproducible*.

I'm going to do this in Python

Python is one of the world's best known computer languages.

Specifically designed to be easy for others to read - this enhances *reproducibility*.

The language of choice for the building simulation community.

- EnergyPlus has a Python plug-in.
- Building simulation software such as IES and DesignBuilder allow Python scripting.
- The eSIM 2021 conference accepted submissions of Jupyter Notebooks written in Python.

I'm going to develop Python packages

Python packages are libraries of reusable code.

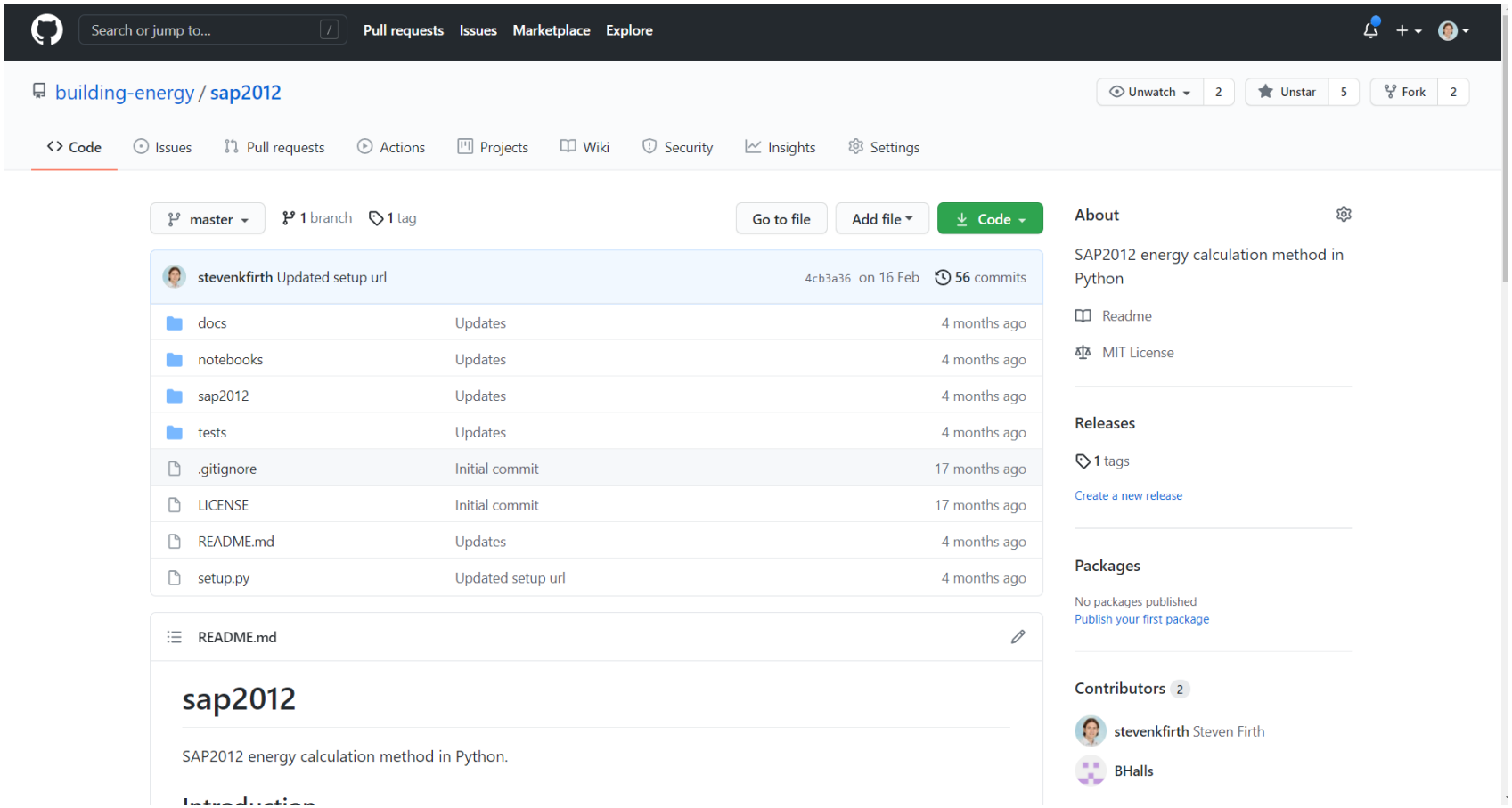
- They have an API which exposes classes and functions.
- The classes provide object instances which can store data and class methods.

Well-known examples of Python packages are:

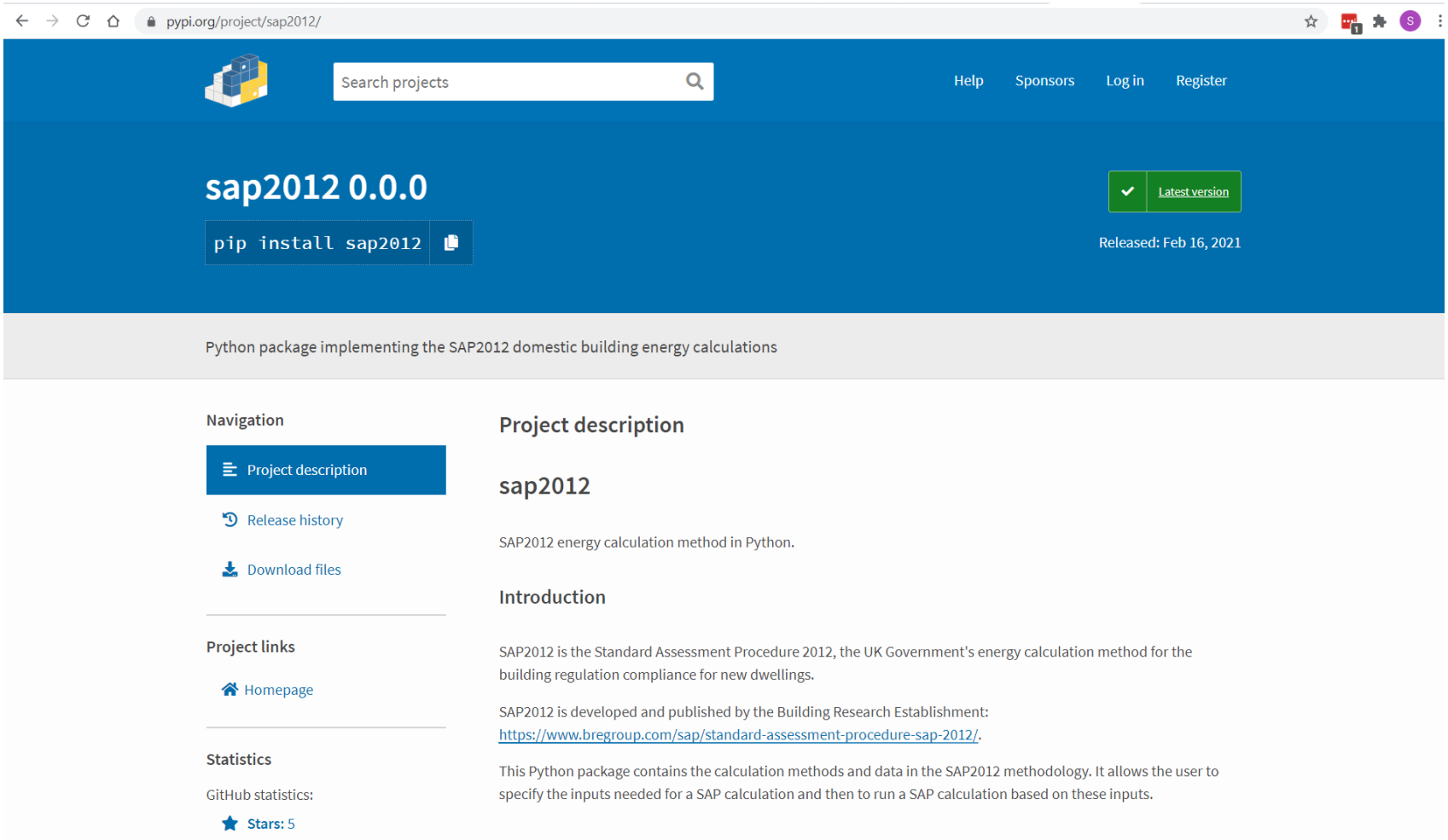
- *pandas* $\rightarrow$ data analysis
- *matplotlib* $\rightarrow$ plotting graphs and figures

Packages can be hosted on *GitHub* $\rightarrow$ allows others to contribute $\rightarrow$ open source software.

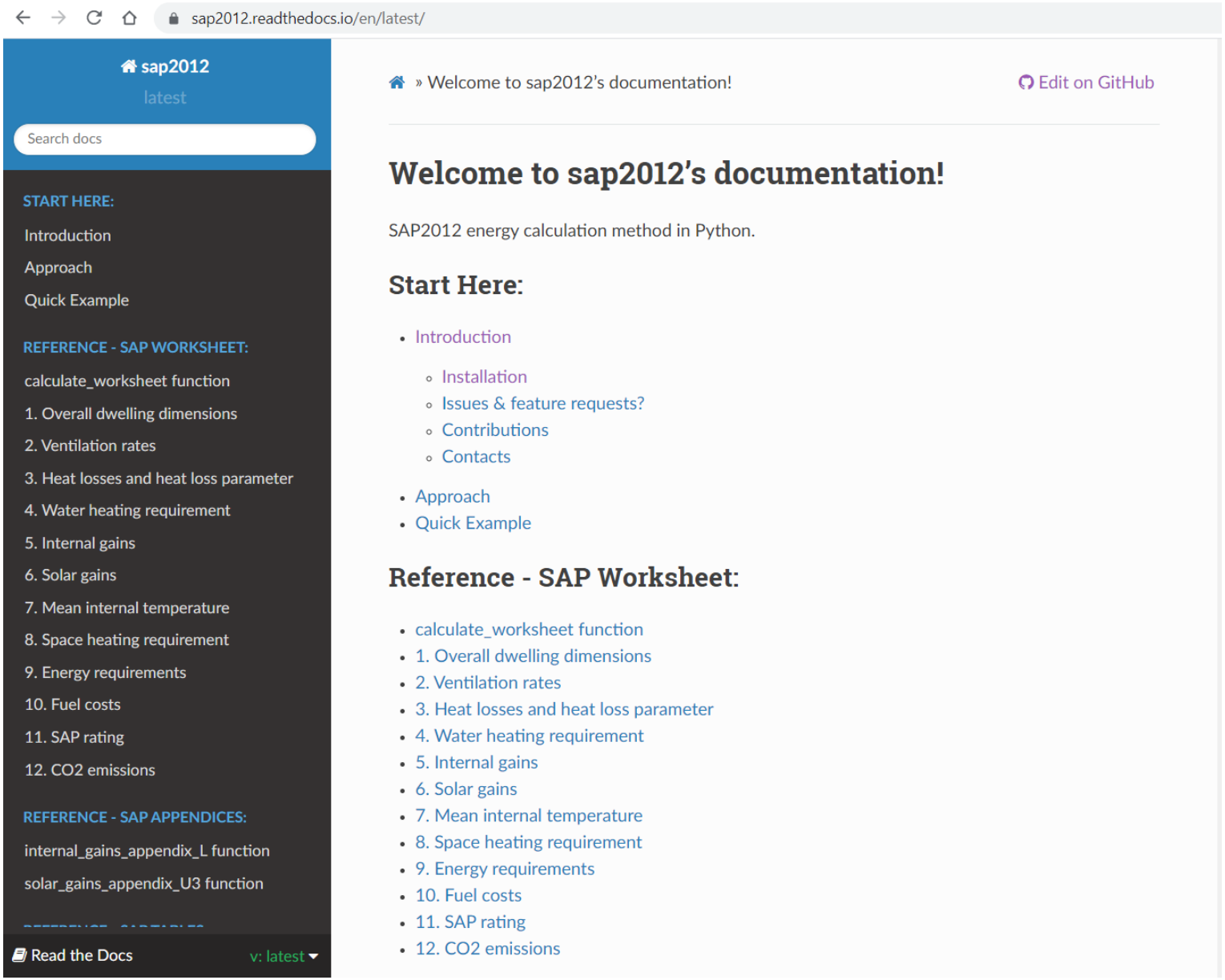
Example #2: the SAP2012 package



Example #2: the SAP2012 package



Example #2: the SAP2012 package



A Python package for geometry calculations

crossproduct - 2D and 3D geometry in Python.

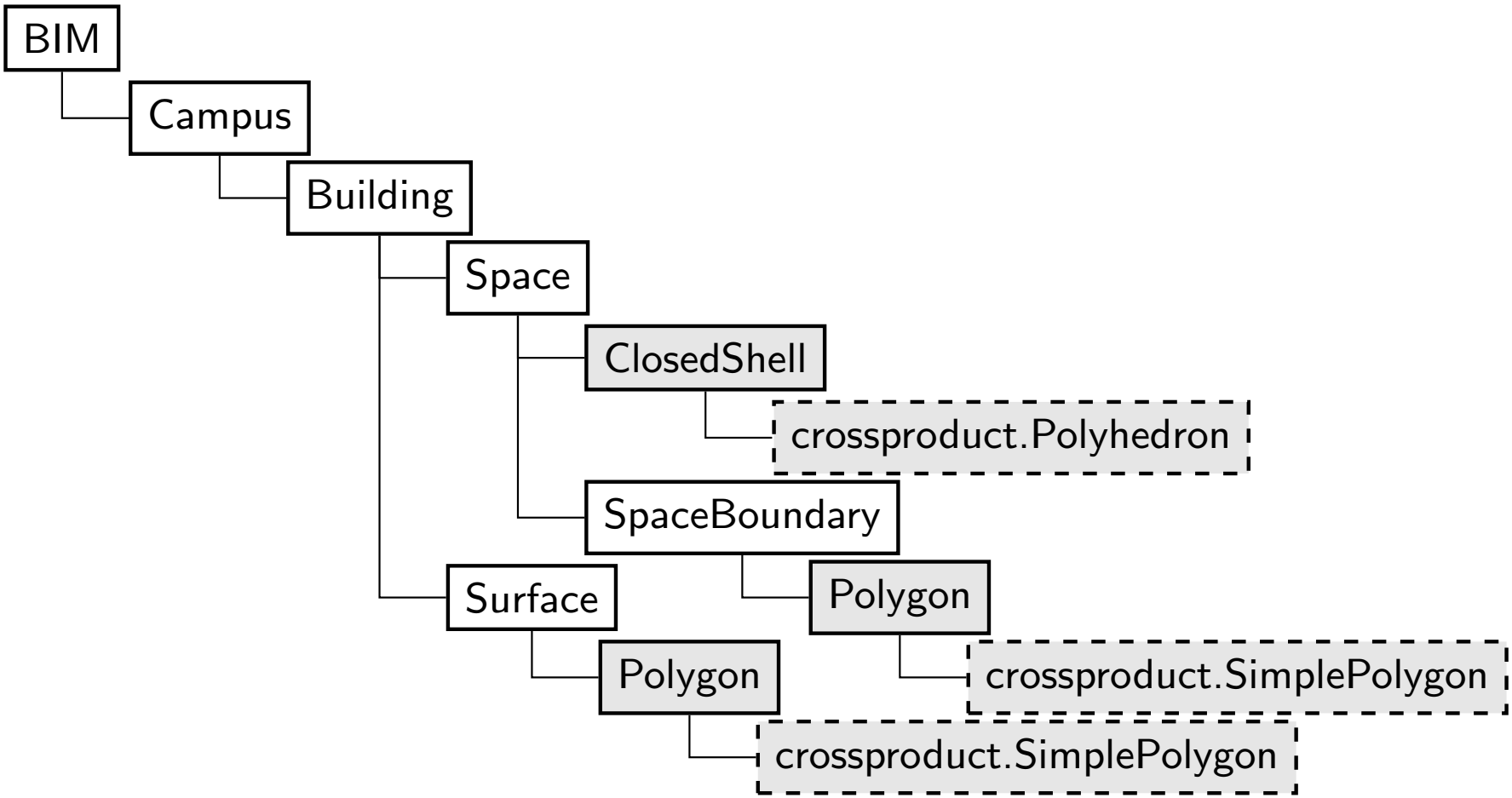
This has classes relating to the major geometric objects:

- **Point** — A 2D or 3D point, as described by xy or xyz coordinates
- **Vector** — A 2D or 3D vector, as described by xy or xyz coordinates
- **Line** — A 2D or 3D line, as described by a Point and a Vector
- **Halfline** — A 2D or 3D halfline, as described by a Point and a Vector
- **Segment** — A 2D or 3D segment, as described by two Points
- **Polyline** — A 2D or 3D polyline, as described by a series of Points
- **Plane** — A 3D plane, as described by a Point and a normal Vector
- **Polygon** — A 2D or 3D polygon, as described by a series of Points
- **SimplePolygon** — A 2D or 3D non-intersecting Polygon
- **Polyhedron** — A 3D polyhedron, as described by a series of Polygons

A Python package for creating 3D BIM models

pybim - The Python Building Information Modelling package

This has a series of linked classes similar to a gbXML file:

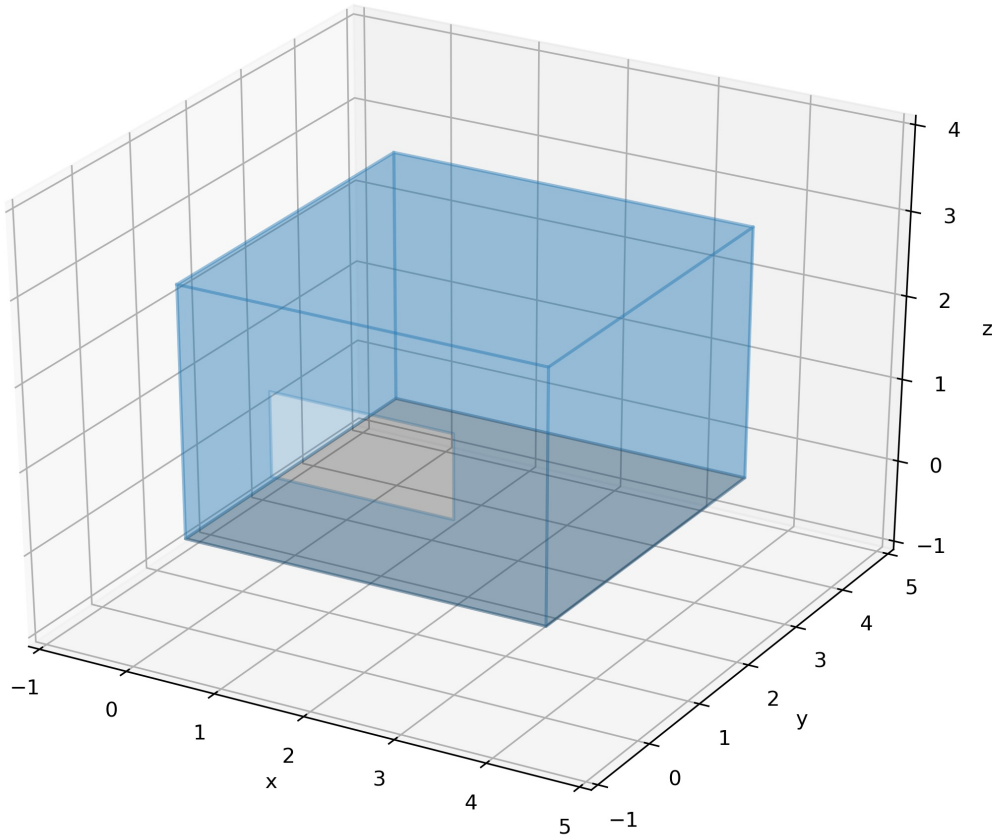


Creating geometry using Python

```
import pybim.gbxml601
bim = pybim.gbxml601.BIM(id='bim1')
campus = bim.add_campus(id='campus1')
building = campus.add_building(id='building1')
space = building.add_space(id='space1',
                           floor_polygon=((0.0, 4.0, 0.0),
                                          (4.0, 4.0, 0.0),
                                          (4.0, 0.0, 0.0),
                                          (0.0, 0.0, 0.0)),
                           extrud_vector=(0.0, 0.0, 3.0)
                           )
surface = campus.surfaces(space_inner='space1',
                          space_outer=None,
                          azimuth=180.0)[0]
opening = surface.add_opening(id='opening1',
                              polygon=((1.0, 1.0),
                                       (3.0, 1.0),
                                       (3.0, 2.0),
                                       (1.0, 2.0))
                              )
```

Creating geometry using Python

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(10,8),dpi=300)
ax = fig.add_subplot(111, projection='3d')
bim.plot(ax)
```



Conclusions

1. Reproducibility involves writing text-based commands, i.e. code, scripts, programming etc.
2. Building simulation can be made reproducible by:
   - Creating the models using programming code.
   - Running the models in EnergyPlus.
   - Analysing the results using programming code.
3. The challenge is the 3D geometry of the building model.
4. New open source libraries and packages are required $\rightarrow$ this is work in progress. . . .
5. Questions, comments, please contact me at: `s.k.firth@lboro.ac.uk`