

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

Steven Firth

Loughborough University

2021-06-23

# My background

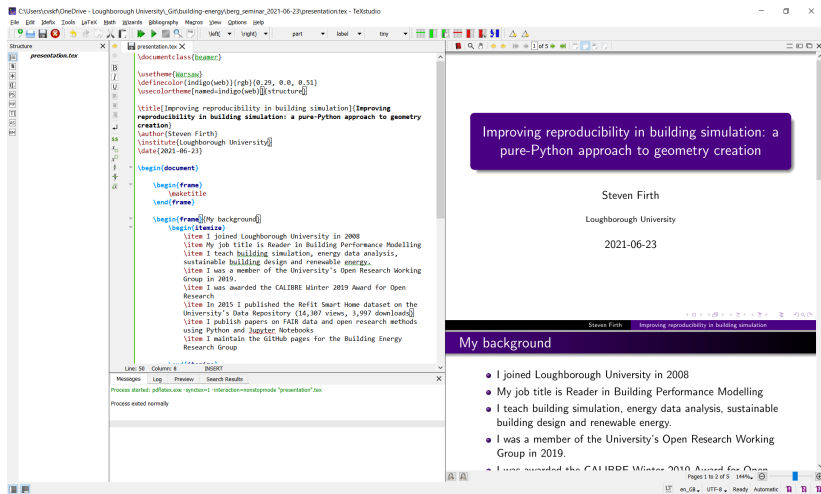
- I joined Loughborough University in 2008
- My job title is Reader in Building Performance Modelling
- I teach building simulation, energy data analysis, sustainable building design and renewable energy.
- I was a member of the University's Open Research Working Group in 2019.
- I was awarded the CALIBRE 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	$\text{\LaTeX}$
Creating slides	Microsoft PowerPoint	$\text{\LaTeX}$
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 screenshot shows the GitHub interface for the repository 'building-energy / berg\_seminar\_2021-06-23'. The repository has 129440 files, 18 seconds ago, and 3 commits. The file list includes:

File	Commit	Time
.gitignore	Initial commit	4 hours ago
LICENSE	Initial commit	4 hours ago
README.md	Initial commit	4 hours ago
latex_example.png	Updates	18 seconds ago
presentation.aux	Updates	32 seconds ago
presentation.nav	Updates	32 seconds ago
presentation.out	Updates	32 seconds ago
presentation.pdf	Updates	32 seconds ago
presentation.snm	Updates	32 seconds ago
presentation.synctex.gz	Updates	32 seconds ago
presentation.tex	Updates	32 seconds ago
presentation.toc	Updates	32 seconds ago
presentation.vrb	Updates	18 seconds ago

The README.md file is displayed below the file list, showing the title 'berg\_seminar\_2021-06-23' and the description: 'Repository with latex source for the presentation 'Improving reproducibility in building simulation: a pure-Python approach to geometry creation'.'

# 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* → data analysis
- *matplotlib* → plotting graphs and figures

Packages can be hosted on *GitHub* → allows others to contribute  
→ open source software.



# Example #2: the SAP2012 package

The screenshot shows the GitHub interface for the repository 'building-energy / sap2012'. The repository has 2 unwatched, 5 stars, and 2 forks. The main content area displays a file tree with the following files and their commit history:


File	Commit History	Time Ago
docs	Updates	4 months ago
notebooks	Updates	4 months ago
sap2012	Updates	4 months ago
tests	Updates	4 months ago
.gitignore	Initial commit	17 months ago
LICENSE	Initial commit	17 months ago
README.md	Updates	4 months ago
setup.py	Updated setup url	4 months ago

The README.md file is selected, showing the title 'sap2012' and the description 'SAP2012 energy calculation method in Python.' The right sidebar contains the 'About' section, which states 'SAP2012 energy calculation method in Python' and lists the 'MIT License'. The 'Releases' section shows '1 tag' and a link to 'Create a new release'. The 'Packages' section states 'No packages published' and a link to 'Publish your first package'. The 'Contributors' section lists 'stevenkfrith' and 'BHalls'.

# Example #2: the SAP2012 package

The screenshot shows the PyPI project page for 'sap2012'. The browser address bar shows 'pypi.org/project/sap2012/'. The page has a blue header with a search bar, navigation links (Help, Sponsors, Log in, Register), and a large section for the current version 'sap2012 0.0.0'. Below this, it says 'Python package implementing the SAP2012 domestic building energy calculations'. The main content area is divided into two columns. The left column contains a 'Navigation' sidebar with links to 'Project description', 'Release history', and 'Download files'. Below this are 'Project links' (Homepage) and 'Statistics' (GitHub statistics: Stars: 5). The right column contains the 'Project description' section, which includes the title 'sap2012', a brief description 'SAP2012 energy calculation method in Python.', an 'Introduction' section, and a paragraph about the package's purpose and origin, including a link to the Building Research Establishment website.

← → ↻ 📑 pypi.org/project/sap2012/ ☆ 🔒 🌐

 Search projects 🔍

Help Sponsors Log in Register

## sap2012 0.0.0

`pip install sap2012` 📄

Released: Feb 16, 2021

Python package implementing the SAP2012 domestic building energy calculations

### Navigation

- Project description
- Release history
- Download files

### Project links

- Homepage

### Statistics

GitHub statistics:

- Stars: 5

### Project description

#### sap2012

SAP2012 energy calculation method in Python.

#### Introduction

SAP2012 is the Standard Assessment Procedure 2012, the UK Government's energy calculation method for the building regulation compliance for new dwellings.

SAP2012 is developed and published by the Building Research Establishment:  
<https://www.bregroup.com/sap/standard-assessment-procedure-sap-2012/>.

This Python package contains the calculation methods and data in the SAP2012 methodology. It allows the user to specify the inputs needed for a SAP calculation and then to run a SAP calculation based on these inputs.

# Example #2: the SAP2012 package

The screenshot shows a web browser displaying the documentation for the 'sap2012' package. The browser's address bar shows 'sap2012.readthedocs.io/en/latest/'. The page has a blue header with the 'sap2012' logo and 'latest' version indicator. A search bar is present. The left sidebar contains a table of contents with sections: 'START HERE:' (Introduction, Approach, Quick Example), 'REFERENCE - SAP WORKSHEET:' (calculate\_worksheet function, 1. Overall dwelling dimensions, 2. Ventilation rates, 3. Heat losses and heat loss parameter, 4. Water heating requirement, 5. Internal gains, 6. Solar gains, 7. Mean internal temperature, 8. Space heating requirement, 9. Energy requirements, 10. Fuel costs, 11. SAP rating, 12. CO2 emissions), and 'REFERENCE - SAP APPENDICES:' (internal\_gains\_appendix\_L function, solar\_gains\_appendix\_U3 function). The main content area has a welcome message, a link to 'Edit on GitHub', a 'Welcome to sap2012's documentation!' heading, a paragraph about SAP2012 energy calculation, a 'Start Here:' section with links to Introduction, Approach, and Quick Example, and a 'Reference - SAP Worksheet:' section with a list of 12 items corresponding to the sidebar. The footer shows 'Read the Docs' and 'v: latest'.

← → ↺ 🏠 sap2012 latest

Search docs

**START HERE:**

- Introduction
- Approach
- Quick Example

**REFERENCE - SAP WORKSHEET:**

- calculate\_worksheet function
- 1. Overall dwelling dimensions
- 2. Ventilation rates
- 3. Heat losses and heat loss parameter
- 4. Water heating requirement
- 5. Internal gains
- 6. Solar gains
- 7. Mean internal temperature
- 8. Space heating requirement
- 9. Energy requirements
- 10. Fuel costs
- 11. SAP rating
- 12. CO2 emissions

**REFERENCE - SAP APPENDICES:**

- internal\_gains\_appendix\_L function
- solar\_gains\_appendix\_U3 function

Read the Docs v: latest ▼

» Welcome to sap2012's documentation! [Edit on GitHub](#)

## Welcome to sap2012's documentation!

SAP2012 energy calculation method in Python.

### Start Here:

- [Introduction](#)
  - [Installation](#)
  - [Issues & feature requests?](#)
  - [Contributions](#)
  - [Contacts](#)
- [Approach](#)
- [Quick Example](#)

### Reference - SAP Worksheet:

- calculate\_worksheet function
- 1. Overall dwelling dimensions
- 2. Ventilation rates
- 3. Heat losses and heat loss parameter
- 4. Water heating requirement
- 5. Internal gains
- 6. Solar gains
- 7. Mean internal temperature
- 8. Space heating requirement
- 9. Energy requirements
- 10. Fuel costs
- 11. SAP rating
- 12. CO2 emissions

# A Python package for geometry calculations

*crossproduct* - 2D and 3D geometry in Python.

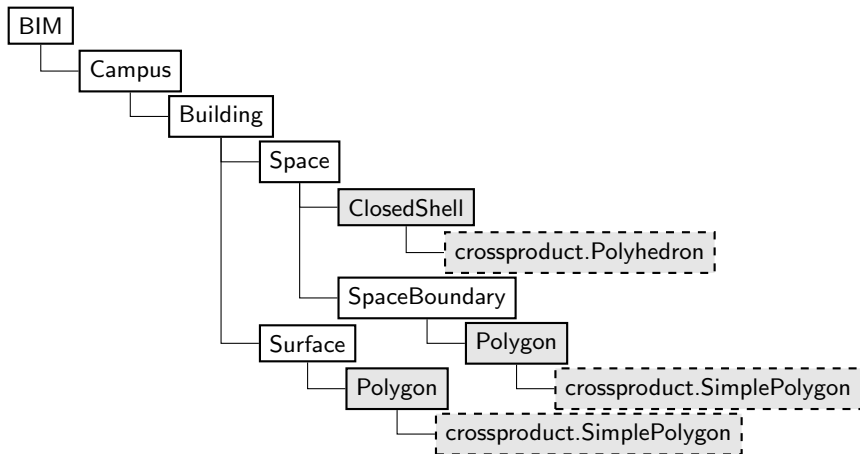
This has classes relating to the major geometric objects:

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:

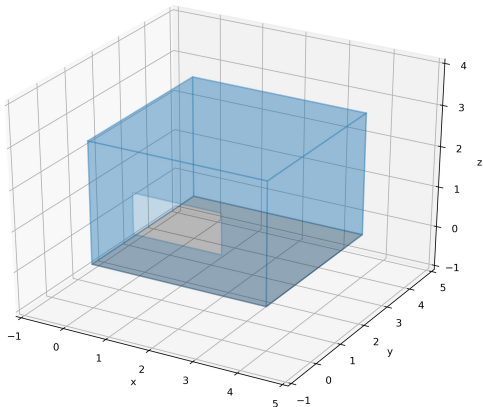


# Creating geometry using Python

```
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 → this is work in progress. . . .
- 5 Questions, comments, please contact me at:  
`s.k.firth@lboro.ac.uk`