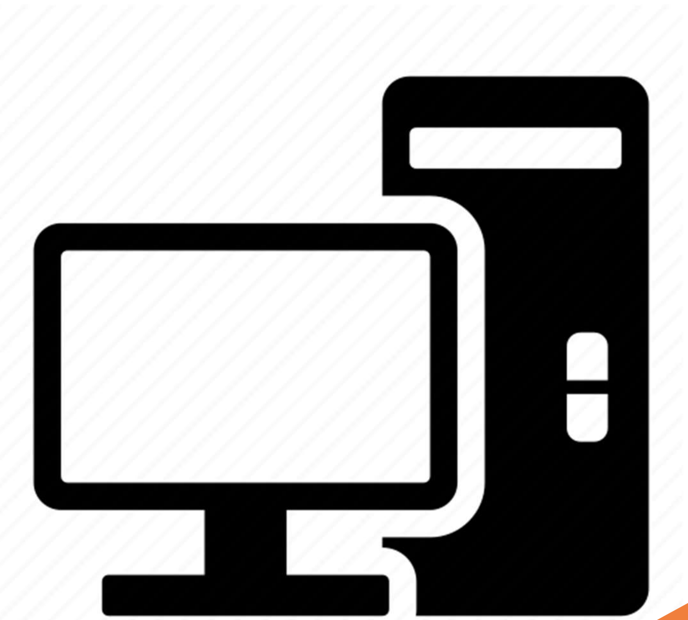
Python Tools for Civil Engineering Applications

Python-based tools for structural and geotechnical engineering analysis, modeling, and visualization

by

Eliott THOMMES



Python Tools for Civil Engineering Applications

Python-based tools for structural and geotechnical engineering analysis, modeling, and visualization

THOMMES Eliott

January 6, 2026

Abstract

Keywords: Python, Civil Engineering.

Résumé

Mots-clés: Python, Génie Civil.

Copyright

Contents

List of Figures

List of Tables

Acronyms

This document is incomplete. The external file associated with the glossary ‘acronym’ (which should be called `ChapterGanttChartLib.acr`) hasn’t been created.

Check the contents of the file `ChapterGanttChartLib.acn`. If it’s empty, that means you haven’t indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can’t be generated. If the file isn’t empty, the document build process hasn’t been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "ChapterGanttChartLib"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "ChapterGanttChartLib"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

1 GanttChartLib

Contents

1.1 Overview

GanttChartLib is a lightweight Python library for creating customizable Gantt charts to visualize project timelines, task progress, and scheduling. It allows users to define a list of tasks with associated start/end dates, completion status, and group coloring, and generates high-quality matplotlib plots to represent the data.

Its goal is to simplify Gantt chart creation for planning, reporting, and presentation purposes while maintaining flexibility in visualization options (such as color grouping, date range control, and weekend shading).

1.2 Dependencies and External Libraries

GanttChartLib depends on the following external libraries:

- `matplotlib` – for plotting the Gantt charts.

- `pandas` – for handling task data and date operations.
- `queue` -
- `datetime` – for manipulating and comparing date/time values.

Additional dependencies (assumed to be in the same project) include:

- `PlotLib.ParamPLT`
- `PlotLib.PLTGrid`
- `PlotLib.PLTLimit`

1.3 Architecture and Organization

The GanttChartLib module is organized into:

- **Core class:** `Tasks` – defines and manages task data and generates Gantt chart visualizations.
- **Helper function:** `DatePlusNumDays` – performs date arithmetic.
- **Plot utilities:** External modules in `PlotLib` manage styling and figure control.

1.4 Core Classes

1.4.1 Class `Tasks`

- **Description:** Manages a collection of tasks and handles Gantt chart plotting. Supports task coloring, completion tracking, and grouped legends.
- **Inherits from:** None
- **Constructor:**

```
def __init__(self):
```

- **Properties:**
  - `getLTask` – Returns the internal list of task dictionaries.
  - `getPandasStruct` – Returns the internal DataFrame storing task information.
  - `SetPandasStruct`: Converts the internal task list into a pandas DataFrame for plotting.
  - `getBColorGroups` – Gets or sets the boolean flag for enabling group-based color mapping.
- **Methods:**
  - `AddTask`: Adds a task to the list with optional color and completion status.
    - **Signature:**

```
def AddTask(title, start, end, completion\_ratio=0, color=None,
↪ BColorGroup=False):
```

  * **Parameters:**
    - `title` (str): Starting date in "YYYY-MM-DD" format.
    - `NumDays` (int): Number of days to add.
  * **Returns:** `str` – New date as a string after adding the specified number of days.

- `RangePLT`: Determines the plotting range from task dates and current date if requested.
  * **Signature:**

```
def AddTask(title, start, end, completion\_ratio=0, color=None,
↪ BColorGroup=False):
```

  * **Parameters:**
    - `x` (typex): x.
    - `x` (typex): x.
  * **Returns:** `x` (typex) : x.

- `PLTTasks`: Plots the Gantt chart using matplotlib. Supports color grouping, legends, weekend shading, and date configuration.
  * **Signature:**

```
def PLTTasks(paramPLT, StartDate=None, EndDate=None, BCurrentDate=False,
↪ BWeekEnds=False, GroupsColors=None, BYTicks=True):
```

  * **Parameters:**
    - `x` (typex): x.
    - `x` (typex): x.
  * **Returns:** `x` (typex) : x.

- **Example Usage:**

```
my_tasks = Tasks()
my_tasks.AddTask("Phase 1", "2024-01-01", "2024-03-01", 0.75, color='IT')
paramPLT = ParamPLT(...)
my_tasks.PLTTasks(paramPLT, BCurrentDate=True)
```

1.5 Functions

1.5.1 Function `DatePlusNumDays`

- **Description:** Adds a given number of days to a specified date and returns the result as a string in "YYYY-MM-DD" format.
- **Signature:**

```
def DatePlusNumDays(Date, NumDays):
```

- **Parameters:**
  - `Date` (str): Starting date in "YYYY-MM-DD" format.
  - `NumDays` (int): Number of days to add.

- **Returns:** `str` – New date as a string after adding the specified number of days.
- **Example:**

```python
new_date = DatePlusNumDays("2024-01-01", 30)
print(new_date)  # Output: "2024-01-31"
```

1.6 Usage Examples

1.7 Development and Planned Improvements

Glossary

This document is incomplete. The external file associated with the glossary 'main' (which should be called `ChapterGanttChartLib.gls`) hasn't been created.

Check the contents of the file `ChapterGanttChartLib.glo`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

If you don't want this glossary, add `nomain` to your package option list when you load `glossaries-extra.sty`. For example:

```
\usepackage[nomain]{glossaries-extra}
```

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "ChapterGanttChartLib"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "ChapterGanttChartLib"
  ```

Then rerun LaTeX on this document.
This message will be removed once the problem has been fixed.