

BraineryWiz OpenSeesPy Users Guideline

BraineryWiz is package for plotting OpenSees models and also finite element models and is basically is independent of the element. In this document you will find out the method of installation and usage of BraineryWiz **OpenSeesPy** package for plotting OpenSees models. BraineryWiz is also available for TCL version.

BraineryWiz is using an element independent approach that do not need to know the type of elements to plot them. So, it makes this package more powerful to plot any type of element without any need of defining them. Also, this package uses GPU to create more fluent and faster. This package is under develop and users using following addresses can find out the latest updates and new features. Also, users can send us any bug if they encounter with and also any requirement if they need to be considered in their plots.

A program by:

- Silvia Mazzoni
- Bijan Sayyafzadeh

Sources

You can find the source of this package at the below addresses:

- BraineryWiz Main WebSite: <https://www.silviasbrainery.com/brainerywiz>
- BraineryWiz GitHub Page: <https://github.com/Silvia-s-Brainery/BraineryWiz>
- Email Address: BraineryWiz@Gmail.com

Installation

- Using command prompt using the following command:

```
pip install BraineryWiz
```

- For anaconda users:

```
conda install BraineryWiz
```

How to use:

User should import package in their codes:

```
import BraineryWiz as bz
```

Then by referring to the assigned name for the package, you will have access to the commands. In the following part commands are described.

BraineryWiz commands and options:

Currently available commands are:

- Model Plotting command: `PlotModel`
- Deformation plotting command: `PlotDefo`
- ModeShape plotting command: `PlotModeShape`
- Realtime plotting commands: `RealTimeObj`, `RealTimeUpdate`
- Animation plotting commands: `RecorderReset`, `Record`, `PlotAnime`, `PlotAnimeGif`

Model Plotting command

PlotModel is the command that should be used to plot the model. The structure of the command is in this way:

```
PlotModel (plotmode=3, **kwargs):
```

There are plenty of options that user can add to this command in the kwargs part. They are described in the following:

- ***plotmode=1***
 - For different python editors there is a need to consider some settings to plot the figure. There are different modes to plot figures and if your editor does not show the figure change the number of the mode! (1,2,3,4,5,6)
- ***draw_nodes=True***
 - Boolean that determines does nodes be drawn or not
- ***show_nodes_tag=False***
 - Boolean that determines does the nodes tag be shown or not
- ***show_elemens_tag=False***
 - Boolean that determines does the elements tag be shown or not
- ***onhover_message=False***
 - Boolean that says Do the hovers data be shown or not
- ***title='BraineryWiz'***
 - Title of the figure
- ***fig_width=1000***
 - determines the width of the Figure
- ***fig_height=800***
 - determines the height or the figure
- ***elements_data_tag=[]***
 - List of the elements tag that you want to add some data to be shown on them
- ***elements_data=[]***
 - List of the data corresponding to the elements that you want to be shown on each element
- ***nodes_data_tag=[]***
 - List of the nodes tag that you want to add some data to be shown on them
- ***nodes_data=[]***

- List of the data corresponding to the nodes that you want to be shown on each element
- **image_type=None**
 - (Types: 'png' | 'jpeg' | 'svg' | 'webp') This parameter sets the format of the image to be downloaded, if we choose to download an image.
- **image_filename='BraineryWz'**
 - Filename of the download image
- **quivers_size=None**
 - If None means do not plot quivers else plot quivers with entered size by the user
- **plot_fibers=False**
 - Boolean that shows plot Fiber points or not

Deformation plotting command

PlotDefo is for plotting current step deformation of the model. The structure of the command is in this way:

```
PlotDefo (plotmode=3, **kwargs):
```

There are plenty of options that user can add to this command in the kwargs part. They are described in the following:

- **plotmode=1**
 - For different python editors there is a need to consider some settings to plot the figure. There are different modes to plot figures and if your editor does not show the figure change the number of the mode! (1,2,3,4,5,6)
- **scale_factor=1**
 - determines scale factor
- **draw_wire_shadow=True**
 - Boolean that determines does Shadow structure be drawn or not
- **draw_nodes=True**
 - Boolean that determines does nodes be drawn or not
- **show_nodes_tag=False**
 - Boolean that determines does the nodes tag be shown or not
- **show_elements_tag=False**
 - Boolean that determines does the elements tag be shown or not
- **onhover_message=False**
 - Boolean that says Do the hovers data be shown or not
- **title='BraineryWiz Deformation'**
 - Title of the figure
- **fig_width=1000**
 - determines the width of the Figure
- **fig_height=800**

- determines the height of the figure
- **elements_data_tag=[]**
 - List of the elements tag that you want to add some data to be shown on them
- **elements_data=[]**
 - List of the data corresponding to the elements that you want to be shown on each element
- **nodes_data_tag=[]**
 - List of the nodes tag that you want to add some data to be shown on them
- **nodes_data=[]**
 - List of the data corresponding to the nodes that you want to be shown on each element
- **image_type=None**
 - (Types: 'png' | 'jpeg' | 'svg' | 'webp') This parameter sets the format of the image to be downloaded, if we choose to download an image.
- **image_filename='BraineryWz'**
 - Filename of the download image

ModeShape plotting command

PlotModeShape is for plotting current mode shape of the model. The structure of the command is in this way:

```
PlotModeShape (plotmode=3, **kwargs):
```

There are plenty of options that user can add to this command in the kwargs part. They are described in the following:

- **plotmode=1**
 - For different python editors there is a need to consider some settings to plot the figure. There are different modes to plot figures and if your editor does not show the figure change the number of the mode! (1,2,3,4,5,6)
- **scale_factor=1**
 - determines scale factor
- **mode_number=1**
 - number identifying which mode you would like to display
- **round_digit=4**
 - Number of the digits that wants to be shown
- **draw_wire_shadow=True**
 - Boolean that determines does Shadow structure be drawn or not
- **draw_nodes=True**
 - Boolean that determines does nodes be drawn or not
- **show_nodes_tag=False**
 - Boolean that determines does the nodes tag be shown or not
- **show_elements_tag=False**
 - Boolean that determines does the elements tag be shown or not

- ***onhover_message***=False
 - Boolean that says Do the hovers data be shown or not
- ***title***='BraineryWiz ModeShape'
 - Title of the figure
- ***fig_width***=1000
 - determines the width of the Figure
- ***fig_height***=800
 - determines the height of the Figure
- ***elements_data_tag***=[]
 - List of the elements tag that you want to add some data to be shown on them
- ***elements_data***=[]
 - List of the data corresponding to the elements that you want to be shown on each element
- ***nodes_data_tag***=[]
 - List of the nodes tag that you want to add some data to be shown on them
- ***nodes_data***=[]
 - List of the data corresponding to the nodes that you want to be shown on each element
- ***image_type***=None
 - (Types: 'png' | 'jpeg' | 'svg' | 'webp') This parameter sets the format of the image to be downloaded, if we choose to download an image.
- ***image_filename***='BraineryWz'
 - Filename of the download image

Realtime plotting commands

RealTimeObj, **RealTimeUpdate** are commands for plotting the deformation of the model in each analysis step. Using **RealTimeObj** command, the code creates a figure of the model and then by each time calling **RealTimeUpdate** command the figure of the model will be update to the current deformation of the model.

ATTENTION: This feature currently is just available only on the JupyterNotebook editor.

RealTimeObj

By this command a figure of the model will be plotted and by calling the **RealTimeUpdate** command will be update to the current deformation shape. The structure of the command is as the following:

```
RealTimeObj (plotmode=7, **kwargs)
```

There are plenty of options that user can add to this command in the kwargs part. They are described in the following:

plotmode=1

For different python editors there is a need to consider some settings to plot the figure. There are different modes to plot figures and if your editor does not show the figure change the number of the mode! (1,2,3,4,5,6,7,8)

- **draw_wire_shadow=True**
 - Boolean that determines does Shadow structure be drawn or not
- **draw_nodes=True**
 - Boolean that determines does nodes be drawn or not
- **show_nodes_tag=False**
 - Boolean that determines does the nodes tag be shown or not
- **show_elements_tag=False**
 - Boolean that determines does the elements tag be shown or not
- **onhover_message=False**
 - Boolean that says Do the hovers data be shown or not
- **title='BraineryWiz RealTime'**
 - Title of the figure
- **fig_width=1000**
 - determines the width of the Figure
- **fig_height=800**
 - determines the height of the figure
- **image_type=None**
 - (Types: 'png' | 'jpeg' | 'svg' | 'webp') This parameter sets the format of the image to be downloaded, if we choose to download an image.
- **image_filename='BraineryWiz'**
 - Filename of the download image

RealTimeUpdate

This command causes an update to figure that has been plotted by **RealTimeObj** command.

The structure of the command is as the following:

```
RealTimeUpdate(**kwargs):
```

There are some options that user can add to this command in the kwargs part. They are described in the following:

- **scale_factor=1**
 - determines scale factor
- **title='BraineryWiz Deformation'**
 - Title of the figure
- **auto_zoom=True**
 - Boolean that determines that program set zoom automatically or not

Animation plotting commands

RecorderReset, Record, PlotAnime, PlotAnimeGif are commands for plotting the animations of the model. User using **Record** command records current step of deformation data and when the

recording finished then by calling **PlotAnime** command can watch the animation or using **PlotAnimeGif** can create a gif file from the recorded model animation frames. Also, to clear recorded frames, user can use **RecorderReset** command to start recording from zero.

RecorderReset, Record

These commands are with no arguments and need no extra description. Just know the **Record()** records current model frame or deformation data and by repeating this command new frames will be added to the previous recorded frames and also using **RecorderReset()** command all the recorded frames will be clear.

PlotAnime

This command shows the animation of the model according to the recorded frames that user recorded in each step by **Record ()** command. The structure of the command is as the following:

```
PlotAnime (plotmode=3, **kwargs)
```

There are some options that user can add to this command in the kwargs part. They are described in the following:

- **plotmode=1**
 - For different python editors there is a need to consider some settings to plot the figure. There are different modes to plot figures and if your editor does not show the figure change the number of the mode! (1,2,3,4,5,6)
- **dt=0.01**
 - Time interval of each step
- **scale_factor=1**
 - determines scale factor
- **draw_wire_shadow=True**
 - Boolean that determines does Shadow structure be drawn or not
- **draw_nodes=True**
 - Boolean that determines does nodes be drawn or not
- **show_nodes_tag=False**
 - Boolean that determines does the nodes tag be shown or not
- **show_elements_tag=False**
 - Boolean that determines does the elements tag be shown or not
- **onhover_message=False**
 - Boolean that says Do the hovers data be shown or not
- **title='BraineryWiz Deformation'**
 - Title of the figure
- **fig_width=1000**
 - determines the width of the Figure
- **fig_height=800**
 - determines the height of the figure
- **elements_data_tag=[]**
 - List of the elements tag that you want to add some data to be shown on them

- ***elements_data***=[]
 - List of the data corresponding to the elements that you want to be shown on each element
- ***nodes_data_tag***=[]
 - List of the nodes tag that you want to add some data to be shown on them
- ***nodes_data***=[]
 - List of the data corresponding to the nodes that you want to be shown on each element
- ***image_type***=None
 - (Types: 'png' | 'jpeg' | 'svg' | 'webp') This parameter sets the format of the image to be downloaded, if we choose to download an image.
- ***image_filename***='BraineryWiz'
 - Filename of the download image
- ***recorderfilename***='BrainRecorder.txt'
 - recorderfilename of the recordedData

PlotAnimeGif

This command act exactly like **PlotAnime** command but instead of plotting the animation, it creates a gif file of animation. The structure of the command is as the following:

```
PlotAnimeGif(**kwargs)
```

There are some options that user can add to this command in the kwargs part. They are described in the following:

- ***dt***=0.01
 - Time interval of each step
- ***scale_factor***=1
 - determines scale factor
- ***draw_wire_shadow***=True
 - Boolean that determines does Shadow structure be drawn or not
- ***draw_nodes***=True
 - Boolean that determines does nodes be drawn or not
- ***show_nodes_tag***=False
 - Boolean that determines does the nodes tag be shown or not
- ***show_elemens_tag***=False
 - Boolean that determines does the elements tag be shown or not
- ***title***='BraineryWiz Deformation'
 - Title of the figure
- ***fig_width***=1000
 - determines the width of the Figure
- ***fig_height***=800
 - determines the height of the figure
- ***image_filename***='BraineryWiz'

The logo for Brainery WIZ is located in the top-left corner. It consists of the word "Brainery" in a purple, cursive-style font, with "WIZ" in a bold, purple, sans-serif font below it. The text is enclosed within a light purple, hand-drawn oval shape.

- Filename of the download image
- ***recorderfilename***='BrainRecorder.txt'
 - recorderfilename of the recordedData