

Design Note

Project

Piles Stiffness Calibration Tool

Subject

Zero Code Tools

Project no

[Project no]

Date

11 November 2024

| Revision | Description | Issued by | Date | Approved (signature) |
|----------|--|-----------|------------|----------------------|
| P01 | User Guide for the "Piles Stiffness Calibration" Zero Tool | GA | 11/11/2024 | |

Description

The Piles Stiffness Calibration Tool is a CSI ETABS Plugin developed by Buro Happold to assist structural and geotechnical engineers to automate the calibration of existing/proposed piles stiffnesses under existing/proposed loads.

The tool is directly accessible in the ETABS UI and allows to run in parallel alternate analyses both in ETABS (structural software) and in Oasys PDisp (geotechnical software): the first to compute the structural loads applied onto the piles and the second to determine their stiffness.

The process continues until the convergence criteria are met and the documentation on the outputs is produced automatically for the engineers.

Piles Stiffness Calibration Tool

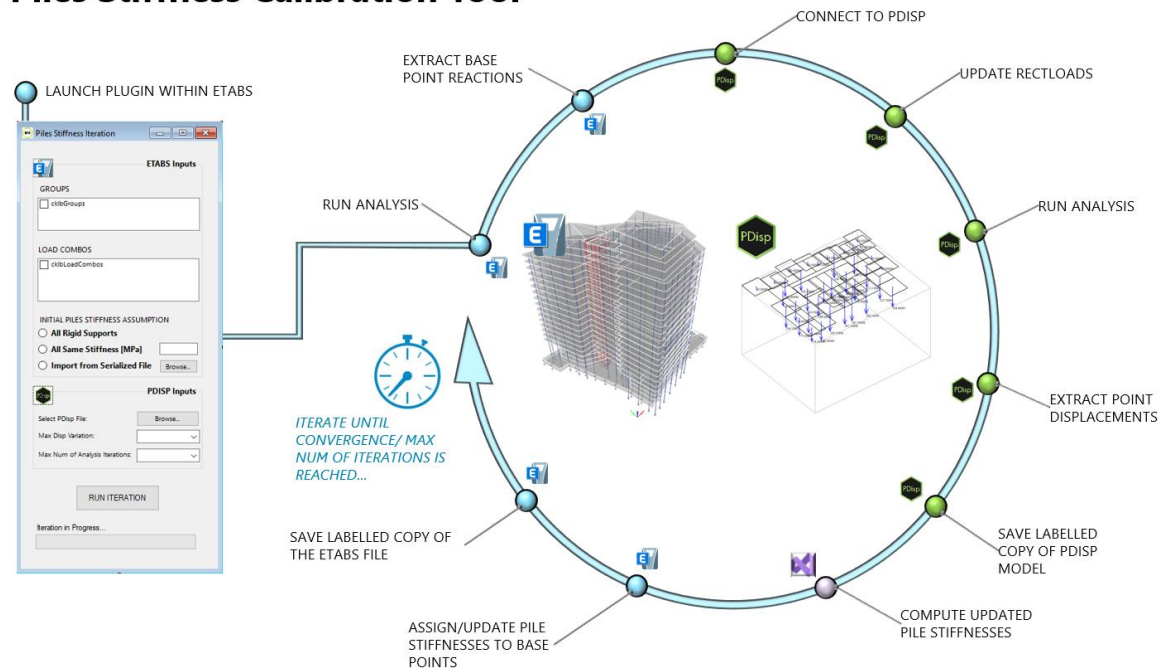
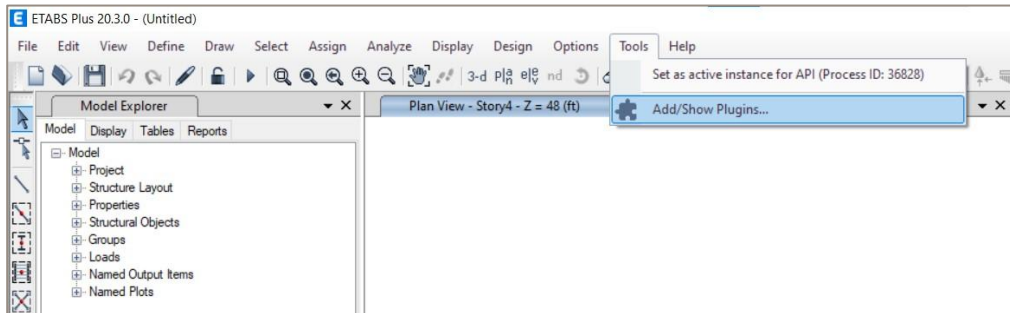


Figure 1: Overview on main steps of the tool's alogrithm

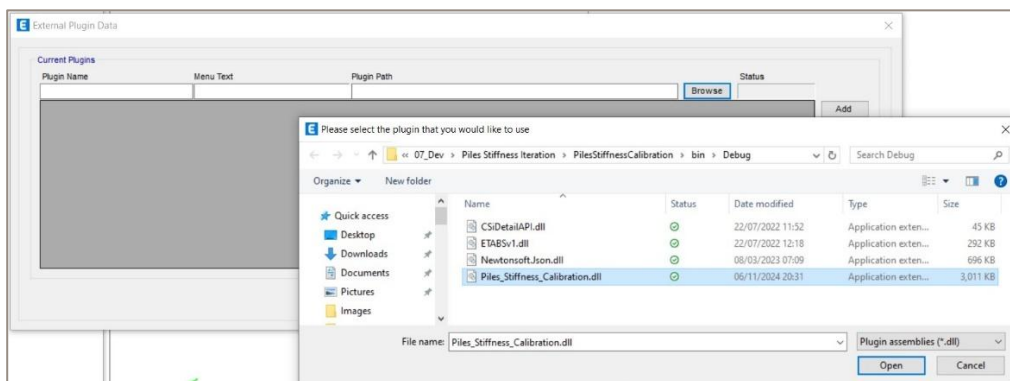
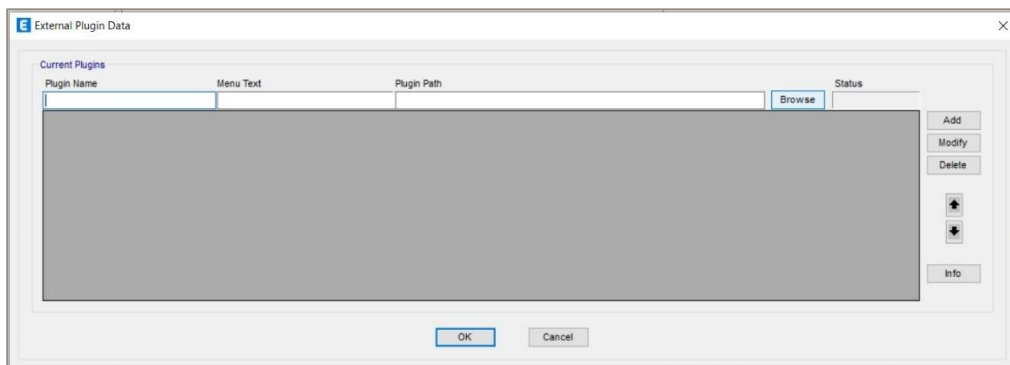
Installation

The process of installation of the plugin is quite immediate and can be summarized in the following steps.

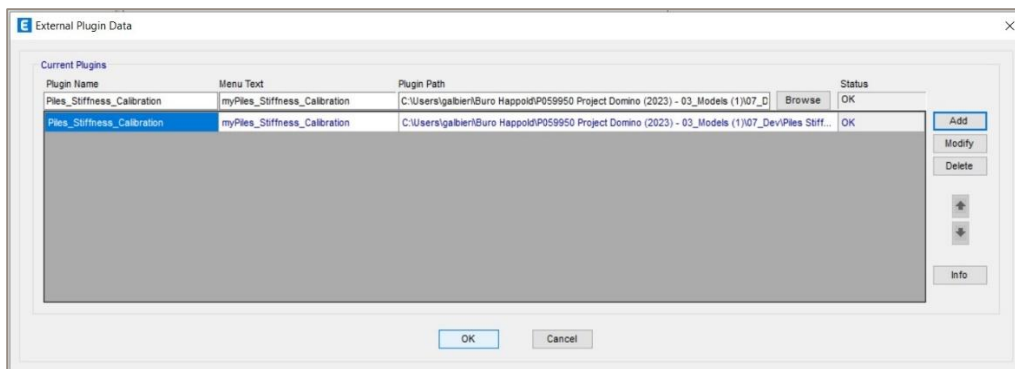
1. Open ETABS and create a new model
2. Left Click on "Tools" -> "Add/Show Plugins". The "External Plugin Data" window pops up.



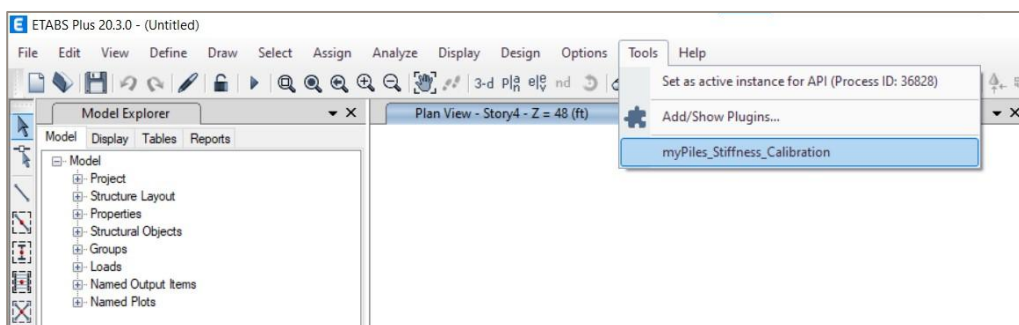
3. In the "External Plugin Data" Window, left click on the "Browse" button and select the assembly Piles_Stiffness_Calibration.dll file.



4. Back in the "External Plugin Data" Window, the Name, Menu Text and path of the plugin will now be displayed in the corresponding textboxes. Left click on the "Add" button to load the selected plugin into ETABS.



5. Left click on the "OK" button and all is done. Now, when getting back into the "Tools" ribbon button, it will be possible to see the uploaded plugin available for use. Just click on it to run it whenever you want and in whatever future session of ETABS.



User Interface

ETABS Tools UI

In order to run the Plugin, it's just enough to access the Tools Menu in ETABS and left click on the name of the plugin in the drop-down window coming up.

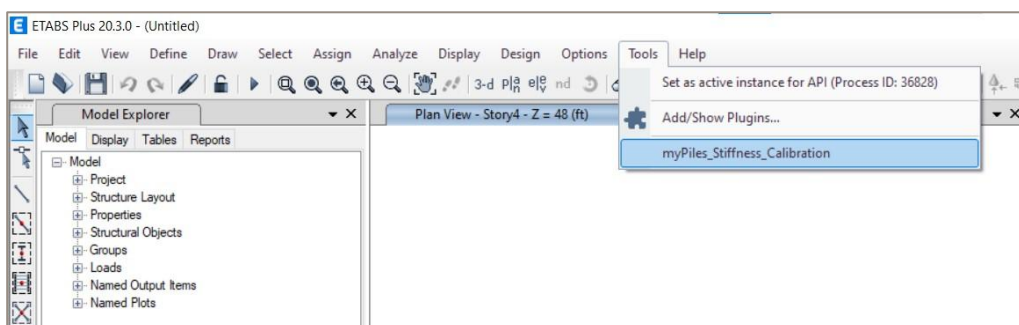


Figure 2: Access to the Tool via the ETABS UI

SplashScreen

As soon as the Plugin starts running, a Splashscreen window will appear on the screen. This contains the logo of the tool, the name, the version and the copyright. After few seconds it will disappear leaving space to the AboutBox window.

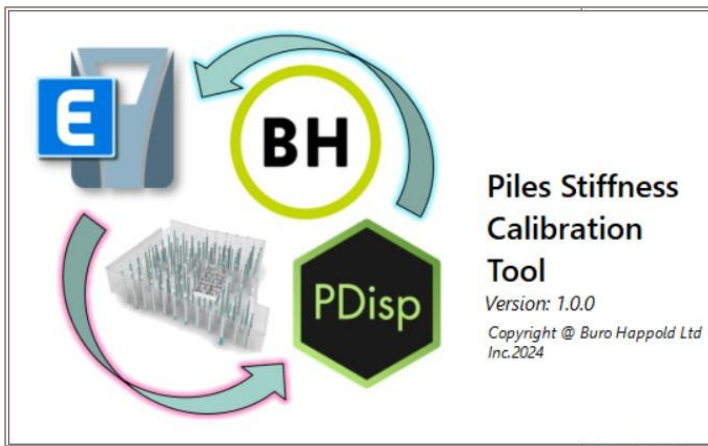


Figure 3: Tool's Splashscreen Window

AboutBox

The AboutBox window is a very important part of the tool's UI as it contains relevant information about the use of the plugin. In addition to the logo and the versioning/copyright information, it contains a textbox where the strengths and limitations of the tool are reported and regularly updated.

It is strongly recommended that, whenever a new version of the tool gets released, the user takes the chance to read all this information before moving on using the tool.

The information reported is subdivided in the following sections:

- DESCRIPTION
 - High level description of the scope of the tool.
- ALGORITHM
 - Step-by-step process followed by the tool to compute the outputs.
- UI FEATURES
 - List of main inputs to be collected from the user via the User Interface
- LIMITATIONS
 - Current limitations the user has to be aware of before using the tool.

Once finished reading, left click on the "OK" button in the bottom right corner of the window to move on to the Inputs Form.

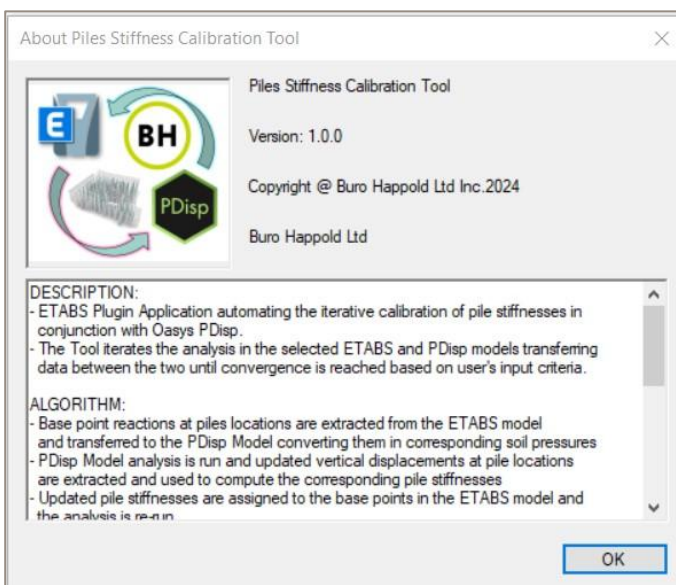


Figure 4: Tool's AboutBox Window

Inputs Form

The Inputs Form is the main window of the tool. It's the window that reads data from the ETABS model currently running, allows the user to select the inputs required, runs the iterative process and informs the user when the process finishes.

The Form is designed to be as simple as possible and is subdivided into 3 main sections:

- The Inputs Section
 - Subdivided into "ETABS Inputs" and "PDISP Inputs" groups.
- The Run Iteration Button
 - To be clicked when the user is ready to launch the iteration.
- The Progress Bar
 - Filling gradually while the iteration is running up to completion.

A detailed description of both Inputs and Outputs is reported in the following paragraphs of this document.

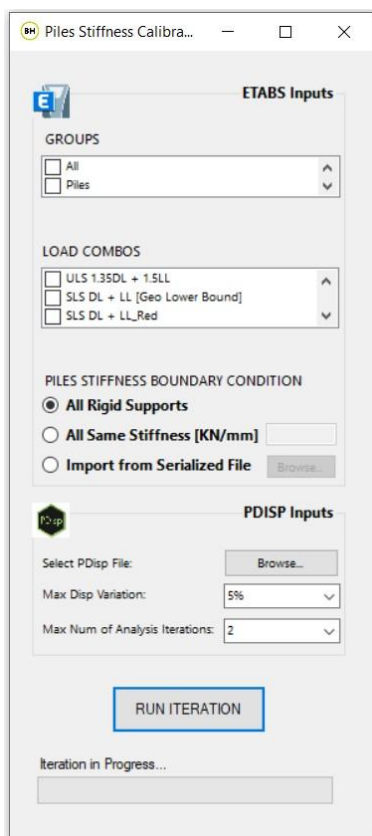


Figure 5: Tool's Inputs Form Window

Inputs

ETABS Inputs

The inputs that the tool requires the user to choose/select are the following ones:

- Groups
- Load Combos
- Piles Stiffness Boundary Condition

Groups

The Groups appearing within this List-Box (see Figure 5) are directly extracted by the tool from the ETABS model currently running.

These groups have to be set up by the user in the ETABS model prior to running the tool.

It is important that all the nodes representing piles in the ETABS model are assigned with a specific group (e.g. "Piles"). The tool is designed to select the pile nodes based on the group selected by the user.

Load Combinations

The Load Combinations displayed in this List-Box (see Figure 5) are directly extracted by the tool from the ETABS model currently running.

Similarly to what happens with the Groups, the user is requested to select one single Load Combination.

The Load Combination is the one that is used by the tool to compute the pile loads in ETABS and then determine the corresponding pile stiffnesses in PDisp.

Piles Stiffness Boundary Condition

The piles stiffness boundary condition stands for the initial value of the stiffness of the piles that the iteration will start running from.

Three different options are available:

- All Rigid Supports
- All Same Stiffness [KN/mm]
- Import from Serialized File

All Rigid Supports

This is the most basic and probably most frequent option to be used.

All the pile supports will be initially set as rigid pinned supports.

This can be possibly the best choice when no data nor estimate about the stiffness of the piles is currently available.

Therefore, the pile loads computed by ETABS in the first iteration of the process will be determined assuming all the piles to have the same infinite stiffness.

Moving to the following iterations, the pile supports will be assigned by the tool with specific stiffness values that will be calculated in the PDisp model that is running in parallel.

All Same Stiffness [KN/mm]

This is the intermediate option and consists in assigning to all the piles the same vertical stiffness value expressed in KN/mm.

This option is to be used when, for example, the geotechnical engineers can calculate an initial estimate of the stiffness value for all the piles.

This represents a good starting point (more refined and accurate compared to the "All Rigid Supports" assumption) for running the iterative calibration of the piles stiffnesses.

Import from Serialized File

This is the most advanced option and it is especially useful when designing new structures on top of existing piles.

In such case, the existing piles will have already experienced a certain amount of load due to the pre-existing structure being supported on them.

Therefore, in the moment when the existing building gets demolished allowing for constructing the new one, the piles will have reached a certain value of stiffness that can increase only if the new proposed loads applied onto them are higher than the ones from the existing scenario (as long as the piles don't fail).

This means that, prior to running the piles stiffness calibration on the proposed model, we need to have run it on the existing (to be demolished) model. This will allow to get the converged stiffness value for all the existing piles as well as the information regarding the maximum load they have been subjected to during the life of the existing building.

This will be the starting point for the analysis of the proposed scenario (whenever, in fact, an existing pile is subjected to a proposed load lower than the existing one, the pile won't settle. The tool will then assign to the pile a rigid support that will re-create the absence of vertical settlement at that specific location).

All this information is stored within a .json file that is produced by the tool itself at the end of the calibration of the existing model.

Loading this .json file, the tool will be able to read all the data calculated for all the existing piles, apply them to the piles in the proposed model and then start the calibration process from this start point (piles in the existing and in the proposed model will have to be assigned with the same names).

PDisp Inputs

The inputs that the tool requires the user to choose/select are the following ones:

- PDisp File
- Maximum Displacement Variation
- Maximum Number of Analysis Iterations

PDisp File

The user is requested to select the Oasys PDisp file that will be run by the tool in parallel with the ETABS model of the building (see Figure 5).

It is strongly recommended that the Oasys PDisp model is set up by the geotechnical engineers working on the project. Geotechnical and structural engineers will have anyway to work together to make sure that all the pile nodes modelled in the PDisp file are assigned with the same name as the corresponding ones present in the ETABS model.

The Tool needs in fact to match the pile nodes in the two software packages in order to be able to transfer loads/stiffnesses from one to the other iteratively. The matching of the piles is not achieved based on the location of the pile nodes but based on their names.

Before running the tool, it's important that the matching of the names of all pile nodes in the two software packages is checked carefully.



Figure 6: PDisp Model: 3D View (left), Pile Points List (right)

Max Disp Variation

Together with the maximum number of analysis iterations, this is one of the convergence criteria for the calibration of the piles stiffnesses (see Figure 5).

The value is defined as a pre-defined percentage (i.e. 5%, 10%, 15% or 20%).

Assuming that the value of 5% is chosen, that means that the iterative calibration process run by the tool will continue until the increase/decrease of stiffness/displacement from one iteration to the other is smaller than 5% on all piles.

That means that we've reached the convergence of our analysis and, therefore, it can be stopped.

Whatever convergence criterion gets reached first between this one and the other, the tool will stop the iterative process and will output the results reached at that stage.

Max Num of Analysis Iterations

Together with the maximum displacement variation, this is one of the convergence criteria for the calibration of the piles stiffnesses (see Figure 5).

The maximum number of iterations the tool can run can be chosen from the following pre-set list: 2, 3, 4, 5, 10, 20, 50, 100.

Whatever convergence criterion gets reached first between this one and the other, the tool will stop the iterative process and will output the results reached at that stage.

Outputs

Data set serialized in JSON

The main output of the tool is a list of data sets, serialized in json, containing all the results of each iteration.

For each iteration, up to the last one, the corresponding data set consists essentially in a list of records (one for each pile object) that contains all the relevant identifying information (name, location, diameter, length...) as well as analysis outputs (loads, displacements, stiffnesses) relevant for that pile.

Serializing this information in JSON allows to minimize the size of the output (thus allowing to use the tool on models of significant size without the risk of producing heavy output files) as well as maximize its readability making it retrievable in the widest range of software packages possible.

```

{
  "name": "IP1",
  "location": {
    "name": "IP1",
    "x": 15000.0,
    "y": 30000.0,
    "z": 6095.999999999999
  },
  "status": 0,
  "loads": {
    "values_N": [
      538.0,
      7148.0,
      1068488.0
    ]
  },
  "displacements": {
    "values_mm": [
      0.0,
      0.0,
      36.2
    ]
  },
  "stiffness": {
    "name": "Spring IP1",
    "values_N_mm": [
      0.0,
      0.0,
      29516.2
    ]
  },
  "diameter_m": 0.0,
  "length_m": 0.0
},
{
  "name": "IP14",
  "location": {
    "name": "IP14",
    "x": 6749.999999999999,
    "y": 26250.0,
    "z": 6095.999999999999
  },
  "status": 0,
  "loads": {
    "values_N": [
      -19956.0,
      29422.0,
      727221.0
    ]
  },
  "displacements": {
    "values_mm": [
      0.0,
      0.0,
      25.2
    ]
  },
  "stiffness": {
    "name": "Spring IP11",
    "values_N_mm": [
      0.0,
      0.0,
      28858.0
    ]
  },
  "diameter_m": 0.0,
  "length_m": 0.0
},
{
  "name": "IP14",
  "location": {
    "name": "IP14",
    "x": 25581.914306180919,
    "y": 28111.471199224048,
    "z": 6095.999999999999
  },
  "status": 0,
  "loads": {
    "values_N": [
      -176.0,
      1862.0,
      7693680.0
    ]
  },
  "displacements": {
    "values_mm": [
      0.0,
      0.0,
      46.7
    ]
  },
  "stiffness": {
    "name": "Spring IP14",
    "values_N_mm": [
      0.0,
      0.0,
      164746.9
    ]
  },
  "diameter_m": 0.0,
  "length_m": 0.0
},

```

Figure 7: Sample of JSON File Content

For instance, once the JSON file has been produced by the tool, it is possible to visualize the outputs graphically in Rhino via Grasshopper just simply using a specific C# gh component de-serializing json files.

The C# component and a sample script showing how to do it are saved at the following link [\[GH Template Files\]](#)

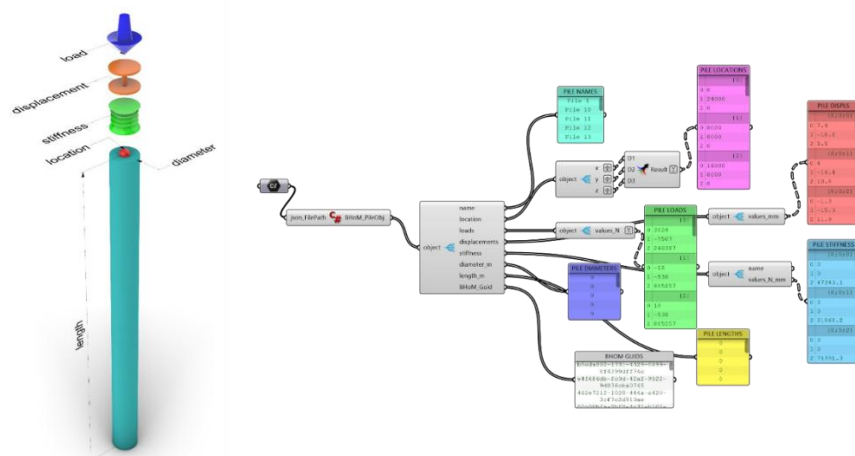


Figure 8: C# Component for deserializing the output JSON File and Data Extraction in Grasshopper

The data imported in the Grasshopper environment can then be visualized, manipulated and be used in whatever computational workflow the user might need.

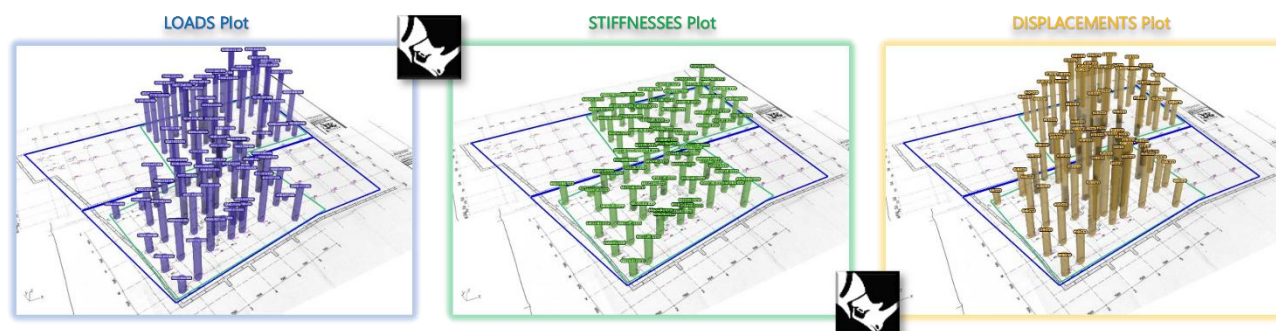


Figure 9: 3D Visualization of the outputs in the Rhino Environment via Grasshopper

Summary Excel Spreadsheet

In addition to the json file and the possibility of reading it in grasshopper, also an excel spreadsheet is produced containing all the relevant outputs computed by the tool.

The excel spreadsheet is built by the tool automatically reading the json data sets behind the scenes and reporting them in excel tables and bar charts properly formatted.

In this way, the user has got the chance to have immediately a full report of the outputs and of the trend of the iteration step-by-step in a simple and universally readable format.

| Iteration00 | | | | | Iteration01 | | | | | Iteration02 | | | | | Iteration03 | | | | | Iteration04 | | | | |
|-------------|--------|---------|------------|---------|-------------|--------|---------|------------|---------|-------------|--------|---------|------------|---------|-------------|--------|---------|------------|---------|-------------|--------|---------|------------|---------|
| Pile Name | Status | Fz [kN] | Kz [kN/mm] | dz [mm] | Pile Name | Status | Fz [kN] | Kz [kN/mm] | dz [mm] | Pile Name | Status | Fz [kN] | Kz [kN/mm] | dz [mm] | Pile Name | Status | Fz [kN] | Kz [kN/mm] | dz [mm] | Pile Name | Status | Fz [kN] | Kz [kN/mm] | dz [mm] |
| 1P1 | 0 | 1068.5 | 29.5162 | 36.2 | 1P1 | 0 | 1190.1 | 32.8752 | 36.2 | 1P1 | 0 | 1280 | 35.6536 | 35.9 | 1P1 | 0 | 1378.4 | 38.5016 | 35.8 | 1P1 | 0 | 1481.5 | 41.267 | 35.9 |
| 1P10 | 0 | 611.78 | 66.4973 | 9.2 | 1P10 | 0 | 1264.3 | 110.9031 | 11.4 | 1P10 | 0 | 1766.7 | 136.952 | 12.9 | 1P10 | 0 | 2019.5 | 147.4055 | 13.7 | 1P10 | 0 | 2104.9 | 151.4343 | 13.9 |
| 1P11 | 0 | 727.22 | 28.858 | 25.2 | 1P11 | 0 | 923.27 | 34.7092 | 26.6 | 1P11 | 0 | 1030.3 | 38.4427 | 26.8 | 1P11 | 0 | 1115.1 | 41.4552 | 26.9 | 1P11 | 0 | 1191.6 | 44.2975 | 26.9 |
| 1P12 | 0 | 6322.6 | 165.9463 | 38.1 | 1P12 | 0 | 6516 | 167.0769 | 39 | 1P12 | 0 | 6301.4 | 163.6723 | 38.5 | 1P12 | 0 | 6127.6 | 160.4091 | 38.2 | 1P12 | 0 | 5992.3 | 158.1076 | 37.9 |
| 1P13 | 0 | 8375.4 | 174.4868 | 48 | 1P13 | 0 | 8133.5 | 172.3192 | 47.2 | 1P13 | 0 | 7891.3 | 169.3402 | 46.6 | 1P13 | 0 | 7751.3 | 167.4144 | 46.3 | 1P13 | 0 | 7659.1 | 165.7806 | 46.2 |
| 1P14 | 0 | 7693.7 | 164.7469 | 46.7 | 1P14 | 0 | 7731.1 | 167.3403 | 46.2 | 1P14 | 0 | 7713 | 166.9482 | 46.2 | 1P14 | 0 | 7678.2 | 166.5553 | 46.1 | 1P14 | 0 | 7648.2 | 166.2644 | 46 |
| 1P15 | 0 | 5646.4 | 149.7706 | 37.7 | 1P15 | 0 | 5878.9 | 155.5259 | 37.8 | 1P15 | 0 | 5962.9 | 156.0971 | 38.2 | 1P15 | 0 | 5948.7 | 155.3187 | 38.3 | 1P15 | 0 | 5900 | 154.0459 | 38.3 |
| 1P16 | 0 | 4729.6 | 141.1824 | 33.5 | 1P16 | 0 | 5136 | 147.1636 | 34.9 | 1P16 | 0 | 5077.8 | 146.3344 | 34.7 | 1P16 | 0 | 4984.2 | 144.8896 | 34.4 | 1P16 | 0 | 4911.8 | 143.6206 | 34.2 |
| 1P17 | 0 | 4979.8 | 138.327 | 36 | 1P17 | 0 | 4657 | 138.6024 | 33.6 | 1P17 | 0 | 4636.2 | 138.3931 | 33.5 | 1P17 | 0 | 4617.6 | 137.8396 | 33.5 | 1P17 | 0 | 4593.7 | 137.5361 | 33.4 |
| 1P18 | 0 | 945.99 | 38.9295 | 24.3 | 1P18 | 0 | 718.09 | 34.0326 | 21.1 | 1P18 | 0 | 636.23 | 30.8848 | 20.6 | 1P18 | 0 | 580.37 | 28.4495 | 20.4 | 1P18 | 0 | 536.41 | 26.5549 | 20.2 |
| 1P2 | 0 | 1196 | 27.6224 | 43.3 | 1P2 | 0 | 1235.8 | 28.8066 | 42.9 | 1P2 | 0 | 1260.1 | 29.442 | 42.8 | 1P2 | 0 | 1284.1 | 30.1435 | 42.6 | 1P2 | 0 | 1311.3 | 30.7818 | 42.6 |
| 1P22 | 0 | 8186.9 | 152.4555 | 53.7 | 1P22 | 0 | 8183 | 154.6878 | 52.9 | 1P22 | 0 | 8192.6 | 154.8702 | 52.9 | 1P22 | 0 | 8175.2 | 154.5408 | 52.9 | 1P22 | 0 | 8138.7 | 154.1416 | 52.8 |
| 1P23 | 0 | 7435.9 | 117.1005 | 63.5 | 1P23 | 0 | 6812.9 | 115.0832 | 59.2 | 1P23 | 0 | 6669.1 | 113.613 | 58.7 | 1P23 | 0 | 6576.8 | 112.8089 | 58.3 | 1P23 | 0 | 6521.9 | 112.2535 | 58.1 |
| 1P24 | 0 | 7655 | 118.8661 | 64.4 | 1P24 | 0 | 6971.6 | 116.7778 | 59.7 | 1P24 | 0 | 6829.4 | 115.3621 | 59.2 | 1P24 | 0 | 6735.8 | 114.5539 | 58.8 | 1P24 | 0 | 6678.2 | 113.9627 | 58.6 |

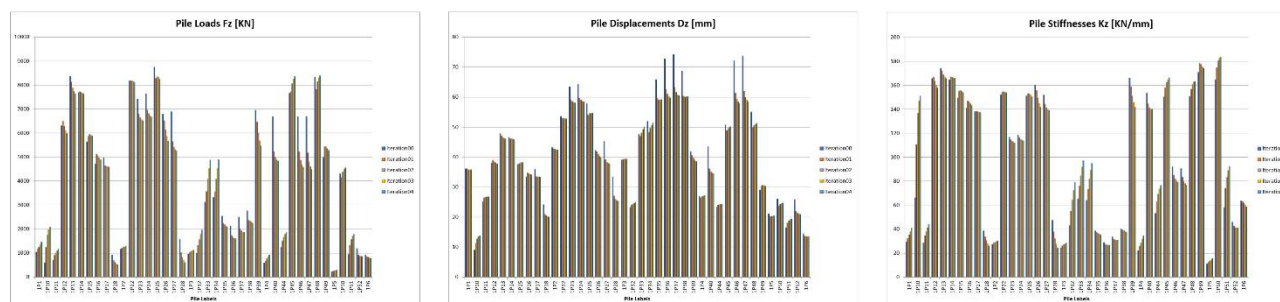


Figure 10: Pile Calibration Outputs in Excel: Table (above) and Bar Charts (below)

ETABS and PDisp Models

In order to allow both the structural and the geotechnical engineers to have a full track of records of all the steps the tool has gone through during the iteration, the tool is designed to save automatically and store all the ETABS and PDisp models created during the process.

For each iteration, the corresponding ETABS and PDisp models used are saved with a proper label with a unique reference to the iteration number they correspond to.

No models overwriting occurs and all files are saved so that the engineers can time-travel through the iteration steps, thus allowing for maximum openness and reliability.

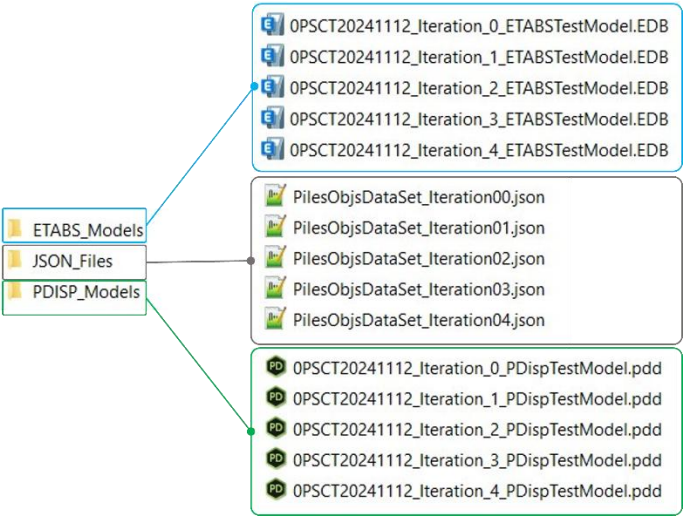


Figure 11: Output Models and JSON Files for each Iteration