Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2023) 1188-1205

ECCOMAS
Proceedia

COMPDYN 2023
9th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Athens, Greece, 12-14 June 2023

# DEVELOPING FUNDAMENTAL PERIOD FORMULAE FOR STEEL FRAMED STRUCTURES THROUGH MACHINE LEARNING AND AUTOMATED ALGORITHMS

**Duan Calitz[1*], George Markou[1], Nikolaos Bakas[2], Manolis Papadrakakis[3]**

[1]University of Pretoria
Civil Engineering Department
Hatfield Campus, South Africa
duancalitz@pm.me, george.markou@up.ac.za

[2] National Infrastructures for Research and Technology – GRNET
7 Kifisias Avenue, 11523, Athens, Greece
nibas@grnet.gr

[3] National Technical University of Athens
Civil Engineering Department
Zografou Campus, Greece
mpapdra@central.ntua.gr

**Abstract.** *The fundamental period of a structure is an important intrinsic dynamic property of the structural system response. Current design codes use simplified empirical formulae to estimate the fundamental period, that generally do not account for the influence of plan geometry, cross-sectional properties of the specific structural elements, nor the properties of the soil the structures are founded on, and thus the effects of soil-structure interaction (SSI) are not accounted for.*

*This research work investigates the fundamental period of unbraced steel framed structures of various geometric configurations, and multiple different steel column members, as well as the SSI effect. A new algorithm is also developed for the automated construction and analysis of finite element models. The proposed automated procedure is used to create a dataset that is used to train and test the predictive formulae for computing the fundamental period of steel structures with and without accounting for the SSI effect. The proposed 40-feature formula was found to derive an optimum coefficient of determination ($R^2$) with a value of 99.976% and an error that is less than 2%.*

**Keywords:** Earthquake Engineering, Finite Element Method, Fundamental Period, Soil-Structure Interaction, Steel-Framed Structure, Structural Dynamics.

doi: 10.7712/120123.10468.20613

## 1 INTRODUCTION

The fundamental period of a structure is a dynamic property of the structural system [1] that is used to estimate seismic loads when designing earthquake-resistant structures. Therefore, having an objective, reliable, and efficient method that will allow Civil Engineers to evaluate the fundamental period of structures is of great importance. Currently, the proposed formulae found within international design codes, usually consider the height of the structure as the only parameter that determines the fundamental period of building-like structures [2, 3]. The best alternative to the design code formulae thus far is the use of the finite element method (FEM) and modal analysis. Nevertheless, this procedure is time-consuming when the objective is to compute the fundamental period of the under-study building. For this reason, the development of a formula that will be able to predict the fundamental period of buildings [2, 3], and in this case, steel framed structures, is necessary. According to [2], a new approach was presented in the development of fundamental period formulae, which foresaw the use of artificial intelligence (AI) and machine learning (ML) to train, test and validate with datasets that are generated through finite element analysis [13]. This approach, as it was also initially discussed in [4], and thereafter utilized to develop the first fundamental formula for steel framed structures [3], it is used herein for the development of a new improved predictive model that also accounts for the stiffness of the structural columns. It is important to note here that this approach was used to develop predictive models for other Civil Engineering related problems and can be found in [4, 8, 2, 5, 6, 7].

## 2 DATASET DEVELOPMENT

To develop the dataset that was used to derive the proposed formulae, multiple modal analyses were performed by using Reconan FEA [13]. The finite element (FE) software Femap was used to construct the initial models as shown in Figure 1[*]. For each initial model, there exists six (6) ”sub-models” of a varying number of storeys. By using the automated procedure that is going to be presented below, a total of 10,368 fundamental period results were generated by using a standard computer. This highlights the importance of using automated processes in developing datasets for training with ML algorithms.

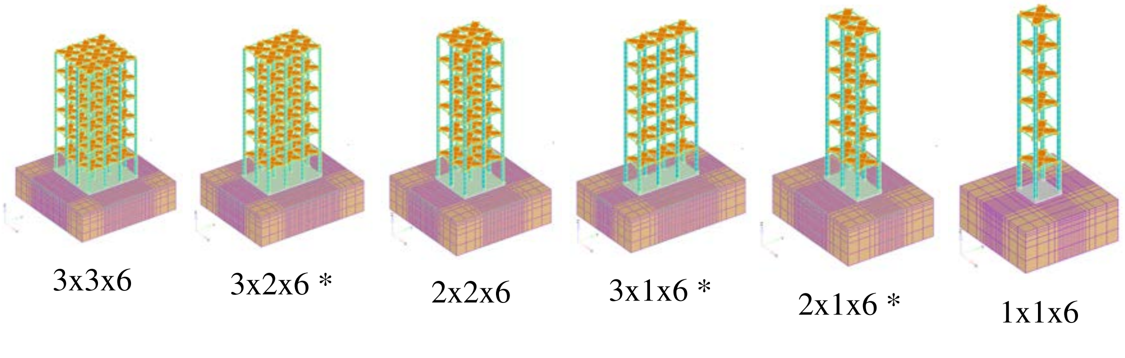

Figure 1: Frames with 6-storeys and different plan views.

[*] Some models could be modified by changing the orientation of the columns by $90^o$ in order to deliver a second set of results.

## 2.1 Steel frames and modal analysis

Modal analysis was performed on various finite element models, in order to obtain the fundamental frequencies of the understudy frames. The largest period that is numerically obtained through the FEA is known as the fundamental period of oscillation and will henceforth be referred to as 'the analysis result(s)'. As it was shown in Figure 1, the models are composed of a raft foundation, columns, beams, and rigid diagonal members that represent the diaphragmatic behaviour of the slab. The foundations were modeled as a concrete raft slab with $E_r$ = 30 GPa; $\nu_r$ = 0.2; $f_{cu}$ = 30 MPa and mass density = 25 kN/m$^3$.

The columns and beams of the steel frames were modeled by assuming a steel material with $E_{col}$ = 200 GPa $\rho_{col}$ = 7850 kg/m$^3$, $\nu_{col}$ = 0.3 and $f_y$ = 355 MPa. For simplicity purposes, the beams were assumed to use one section which was that of the IPE200. It is also important to note here that a total of 64 different column cross-sections were used during the generation of the dataset.

The soil domain was modelled to a depth of 5m with the first two layers having a thickness of 0.3 m, the third layer a thickness of 0.6 m, the fourth 0.8 m the fifth and sixth 1.5 m (see Figure 4). This discretization approach ensured the decrease of the number of hexahedral elements needed without any accuracy loss during the modal analysis [9] Three different Young moduli were used as the properties of the soil ($E_s$ = 65 MPa,300 MPa, and 700 MPa), and the Poisson ratio was $\nu_s$ = 0.2, with a uniaxial compressive strength equal to $f_{cu}$ = 30 MPa.

Figures 2-4 show the different domains of the model that include the raft slab, the steel frame and the soil domain. It is noteworthy to state at this point that the raft slab and soil domains were discretized with the 8-noded isoparametric FE, while the frame was discretized through the use of the Natural Beam Column Flexibility Based (NBCFB) element [10]. It is also evident that the use of different dimensionality FEs requires special handling for connecting the columns with the raft slab. This is done by applying the HYMOD approach (Hybrid Modeling), as it is described in [10].

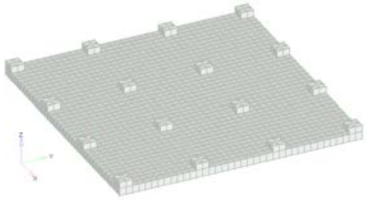

Figure 2: Foundation of a 9mx9m 3x3 bay structure.

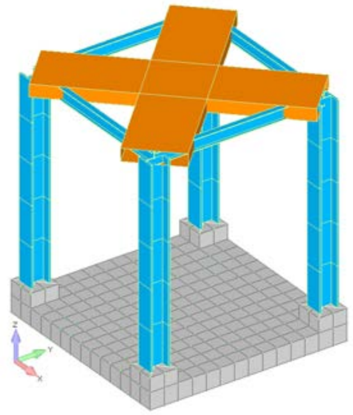

Figure 3: 1x1x1 frame with rigid diagonal members (orange).

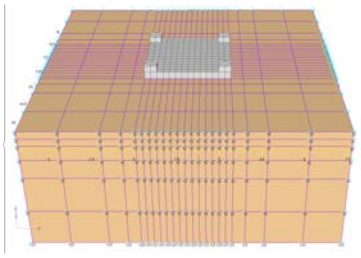

Figure 4: Soil domain - perspective

Table 1 shows the upper and lower bounds for the different simulation parameters.

| Input Parameter | Lower Bound | Upper Bound | Input Units |
|---|---|---|---|
| $H$ | 3.5 | 21 | m |
| $B$ | 3 | 9 | m |
| $L$ | 3 | 9 | m |
| $E$ | 65 | 700 | MPa |
| $I_{col}$ | 126x10$^3$ | 861x10$^6$ | mm$^4$ |
| $A_{col}$ | 856 | 25200 | mm$^2$ |

Table 1: Upper and lower bounds for simulation parameters.

## 2.2 Column stiffness

To account for the column size as an input during the training and testing procedure through the use of the developed dataset, the stiffness had to be introduced as an input variable. This led to the development of three specific input variables: $\xi$, $\Psi$, and $\Phi$, which will be described in this section.

Assume the behaviour of the oscillating column members to be the same as that of a beam element with similar orientation as shown in Figures 5 and 6. Let $M_1$ and $M_2$ be the fixed end moments at either end of a finite beam element subject to displacements and rotations at its ends, and where $\Delta = \delta_2 - \delta_1$ - be the distance between the two displaced ends. The value of $M_1$ can be calculated as:

$$M_1 = \frac{6EI\Delta}{dL^2} - \frac{4EI\theta_1}{dL} - \frac{2EI\theta_2}{dL} \tag{1}$$

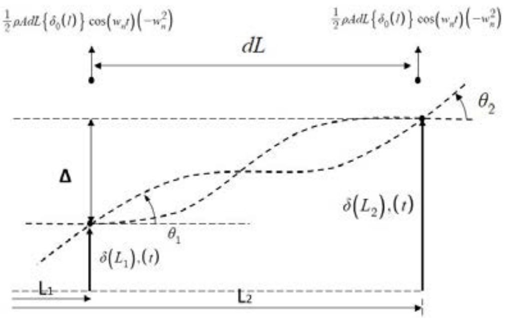
$\frac{1}{2}\rho A dL\{\delta_0(l)\}\cos(w_n t)(-w_n^2)$
$\frac{1}{2}\rho A dL\{\delta_0(l)\}\cos(w_n t)(-w_n^2)$
$dL$
$\theta_2$
Δ
$\delta(L_2).(t)$
$\theta_1$
$\delta(L_1).(t)$
L1
L2

Figure 5: Displacement, and dynamic loading of the finite beam element.

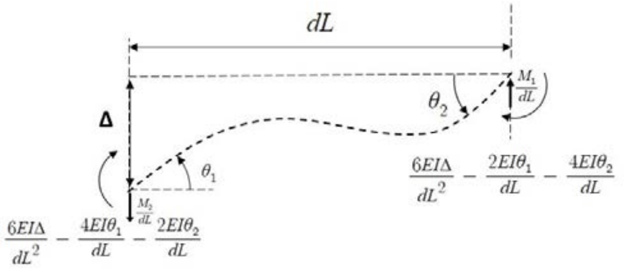
$dL$
$\frac{M_1}{dL}$
$\theta_2$
Δ
$\theta_1$
$\frac{6EI\Delta}{dL^2} - \frac{2EI\theta_1}{dL} - \frac{4EI\theta_2}{dL}$
$\frac{M_2}{dL}$
$\frac{6EI\Delta}{dL^2} - \frac{4EI\theta_1}{dL} - \frac{2EI\theta_2}{dL}$

Figure 6: Fixed end moments of a beam subject to displacement and rotation at both ends.

It follows that the shear force at either end will be equal to the moment at the other end divided by the length of the beam element expressed as: $V_1 = \frac{M_2}{dL}$ and $V_2 = \frac{M_1}{dL}$. Assuming that the oscillating beam element depicted in Figures 5 and 6 is at a point of maximum displacement and that at such a displacement $t = 0$ or $2\pi$.

$$\frac{d\delta}{dt} = 0$$

$$\delta(l,t) = X_0(l)cos(w_n t) \tag{2}$$

It follows that the acceleration is:

$$a = \frac{\partial^2 \delta(l,t)}{\partial t^2} = -X_0 w_n^2 cos(w_n t) \tag{3}$$

Where $X_0$ is the function describing the maximum displacement. This function describes the maximum displacement of the beam element at a point $l$ along the beam. Then by the beam equation it follows that the deflection, rotation, moments, and shear forces can be calculated as:

$$\delta(l) = \delta(l,0) = X_0 cos(w_n) \tag{4}$$

$$\theta(l) = \frac{\partial \delta(l)}{\partial l} = \frac{\partial X_0(l)}{\partial l} cos(w_n) \tag{5}$$

$$\frac{M(l)}{EI} = \frac{\partial \theta(l)}{\partial l} = \frac{\partial^2 X_0(l)}{\partial^2} cos(w_n) \tag{6}$$

$$\frac{V(l)}{EI} = \frac{\partial M(l)}{\partial l} = \frac{\partial^3 X_0(l)}{\partial l^3} cos(w_n) \tag{7}$$

Furthermore let the mass at an end of such a beam element be equal to half the mass of the element so that:

$$m = \frac{1}{2} \rho A \partial l \tag{8}$$

$$F_{net}(l,t) = ma = -\frac{1}{2} \rho A X_0(l) w_n^2 cos(w_n t) \partial l \tag{9}$$

Let the work done by the external inertial forces due to the induced curvature in the $y$ direction be equal to $F_{net}$ times the displacement at the finite beam element. Let the external inertial forces be situated at the ends so that the work done by the external inertial forces is equal to:

$$F_{net}(l_1) X_0(l_1) + F_{net}(l_2) X_0(l_2) \tag{10}$$

$$\sum_{y}^{ma} = -\frac{\rho A \partial l}{2} w_n^2 cos(w_n t) \left( X_0(l_1)^2 + X_0(l_2)^2 \right) \tag{11}$$

Let the work done by the internal forces ($V_1$ and $V_2$) in the direction $y$ due to the induced curvature be:

$$\sum_{y}^{kx} = V_2 X_0(l_2) - V_1 X_0(l_1)$$

$$\sum_{y}^{kx} = cos(w_n t)\left(\frac{M_1}{dl}X_0(l_2) - \frac{M_2}{dl}X_0(l_1)\right) \tag{12}$$

$$\sum_{y}^{kx} = \frac{cos(w_n t)}{dl}\left(\left[\frac{6EI\Delta}{dL^2} - \frac{4EI\theta_1}{dL} - \frac{2EI\theta_2}{dL}\right]X_0(l_2) - \left[\frac{6EI\Delta}{dL^2} - \frac{4EI\theta_2}{dL} - \frac{2EI\theta_1}{dL}\right]X_0(l_1)\right) \tag{13}$$

as $dl \to 0$:

$$l_2 = l_1 + \partial l$$

$$X_0(l_2) = X_0(l_1) + \partial\delta$$

$$\theta_2 = \theta_1 + \partial\theta$$

So that equation 13 becomes:

$$\left(\frac{dl}{cos(w_n t)}\right)\sum_{y}^{kx} = \left[\frac{6EI\Delta}{dL^2} - \frac{4EI\theta_1}{dL} - \frac{2EI\theta_2}{dL}\right]X_0(l_1) + \left[\frac{6EI\Delta}{dL^2} - \frac{4EI\theta_1}{dL} - \frac{2EI\theta_2}{dL}\right]\partial\delta \\ - \left[\frac{6EI\Delta}{dL^2} - \frac{4EI\theta_2}{dL} - \frac{2EI\theta_1}{dL}\right]X_0(l_1) \tag{14}$$

$$\sum_{y}^{kx} = \frac{cos(w_n t)}{dl}\left(\left[\frac{2EI\theta_2}{dL} - \frac{2EI\theta_1}{dL}\right]X_0(l_1) + \left[\frac{6EI(\partial\delta)}{dL^2} - \frac{4EI\theta_1}{dL} - \frac{2EI\theta_2}{dL}\right]\partial\delta\right) \tag{15}$$

$$\sum_{y}^{kx} = \frac{cos(w_n t)}{dl}\left(\left[\frac{2EI\partial\theta}{dL}\right]X_0(l_1) + \left[\frac{6EI(\partial\delta)}{dL^2} - \frac{6EI\theta_1}{dL} - \frac{2EI\partial\theta}{dL}\right]\partial\delta\right) \tag{16}$$

By substituting various representative displacement functions for $X_0(l)$, iterative functions containing new specific inputs and the fundamental period as an iterative variable were developed. The specific inputs were identified and gathered for use during the nonlinear regression process. These specific inputs are:
$\xi$ (xi) – The primary specific input variable:

$$\xi = \frac{3E_{col}I_{col}}{H^3 h \times 10^3}\left(h\left[\frac{(10H^2 - 2ZH - Z^2)(ln|h - Z|)}{10H^2 - 3Z^2} - ln(h)\right] + 2H\right) \tag{17}$$

Where:

$$Z = \sqrt[3]{\frac{-20H^3}{2} + \sqrt{\left(\frac{20H^3}{2}\right)^2 + \left(\frac{10H^2}{3}\right)^3}} - \sqrt[3]{\frac{20H^3}{2} + \sqrt{\left(\frac{20H^3}{2}\right)^2 + \left(\frac{10H^2}{3}\right)^3}} \tag{18}$$

$\Psi$ (Psi) – The secondary specific input:

$$\Psi = (ln|h^3 - 10H^2 h + 20H^3|) + 2ln(h) \tag{19}$$

$\Phi$ (Phi) – The tertiary specific input:

$$\frac{\rho_{col} A_{col} \times 10^3}{120 E_{col} I_{col} H}(-h^5 + 10h^3H^2 - 20h^2H^3) \tag{20}$$

Where:
$H$ – the height of the building in *metres*. (See Table 1)
$h$ – the inter-storey dimension in *metres*. Measured between storey levels. (h=3.5m)
$I_{col}$ – the moment of inertia of the steel column sections about the axis perpendicular to the direction of motion in *mm*$^4$. (See Table 1)
$A_{col}$ – the cross sectional area of the steel column sections in *m*$^2$. (See Table 1)
$E_{col}$ – the elastic modulus of the steel column sections in *GPa*. ($E_{col}$ = 200GPa)
$\rho_{col}$ – the density of the steel column sections in *kg/m*$^3$. ($\rho_{col}$ = 7850kg/m$^3$)

$K_{col}$ – Additional input - the beam stiffness of the column sections in the direction of motion:

$$K_{col} = \frac{E_{col} I_{col}}{H} \tag{21}$$

### 2.3 Geometry and soil structure interaction

In order to account for the SSI phenomenon, both the equivalent soil stiffness ($E_s$) and the unit-less normalized raft-soil stiffness ($K_{rs}$) [11] were investigated.

$K_{rs}$ – the unitless normalised raft-soil stiffness [11]:

$$K_{rs} = 5.57 \left(\frac{E_r}{E_s}\right)\left(\frac{1-\nu_s^2}{1-\nu_r^2}\right)\left(\frac{B}{L}\right)^{0.5}\left(\frac{t_r}{L}\right)^3 \tag{22}$$

Where:

$$E_r = \frac{12E_c}{t_r^3} \tag{23}$$

$E_s$ – the equivalent soil stiffness in *MPa*. (See Table 1)
$E_c$ – the elastic modulus of the concrete raft foundation in *kPa*. ($E_c$ = 30MPa)
$B$ – the width of the structure parallel to the direction of motion in *metres*. Measured between column centres. (See Table 1)
$L$ – the width of the structure perpendicular to the direction of motion in *metres*. Measured between column centres. (See Table 1)
$t_r$ – thickness of the raft foundation in *metres*. ($t_r$ = 0.3m)
$v_s$ and $v_r$ – the Poisson's ratio of the soil and raft respectively. ($v_s$ = 0.3 and $v_r$ = 0.2)
$T_\alpha$ and $T_\beta$ – Additional inputs - modified variations of the fundamental period formula for dynamic systems:

$$T_\alpha = \sqrt{\frac{\rho_{col} H}{K_{col}}} \tag{24}$$

$$T_\beta = 2\pi\sqrt{\frac{\rho_{col} H}{K_{col} \times 10^3}} \tag{25}$$

## 3 MACHINE LEARNING

A dataset containing the analysis results and various input parameters (see Table 2) for the 10,368 models was developed. The dataset was then subject to an algorithmic analysis procedure in order to produce the predictive formulae.

Given the nature of the adopted ML algorithm, it is found to perform optimally when the input values are within the boundaries of the spectrum it was trained for (see Table 1). However, values well out of this spectrum could yield drastic results, thus the proposed formulae should only be used for values within the spectrum it is trained for, as shown in Table 1). The ML algorithm that was used for the needs of this research work can be seen below.

| | |
|---|---|
| Simulation parameters | $H, B, L, E_s, I_{col},$ and $A_{col}$ |
| Dependent input variables | $\xi, \Psi, \Phi, K_{col}, K_{rs}T_\alpha, T_\beta$ |

Table 2: Input parameters used as independent variables by the machine learning algorithm.

Higher Order Regression Algorithm [2][4][12]:

*Input:* $[X]$ (matrix of independent variables - See Table 2), $\{Y\}$ (vector of dependent variable), and $N$ (number of nonlinear features to be kept in the model).

*Output:* Closed form predictive formula.

```
1. Create all nonlinear features* ([a.n])
2. for i from 1 to N do:
3.     for j from 1 to [a.n]
4.         Add j^th feature to the model
5.         Calculate the MAPE_j†
6.     End
7.     In the model, keep the jth feature which yields the minimum prediction error
8. End
```

*Return:* Predictive model (closed form formula)

### 3.1 Automation of the model generation procedure and analysis

Various scripts were developed as part of this research work in order to automate and enhance the model generation procedure, automate the model analysis input process, and automate the data-result extraction process. All developed algorithms were tested and verified through benchmark problems.

The first algorithm See Appendix A which provides the code that is written in Visual Basic and makes use of the Femap API [*]. It takes a base-model and parameter list as input and modifies various element properties within the base-model to match the parameter list. It then exports the modified file as an analysis-ready file in a text format. For the needs of this research work, adopting this approach allowed each base-model to be transformed into multiple unique models ready for automated analysis.

The second script (See Appendix B) developed herein, takes the list of the automatically generated input files and runs the analysis process through the use of Reconan FEA (2020) [13]. Furthermore, it parses the files generated by each analysis run and stores the required output values

[*] with all inter-item multiplication combinations, up to the third degree.

[†] Mean Absolute Percentage Error (MAPE)

[*] Femap is the Finite Element Modeling software used.

in a comprehensive data set that also contains the information of the input variable parameters for each model. The development of these automated algorithms allowed for the analysis of more than 10,000 numerical models through the use of a standard computer in a matter of days. This is ideal when working with any ML algorithm that requires a large dataset.

## 4 PROPOSED FORMULAE

This section will present the proposed formulae as they are derived through the training and testing of the ML algorithm that was presented in Section 3. The coefficient of determination and the Mean Absolute Percentage Error (MAPE) that were numerically obtained through the training and test datasets will be discussed below.

### 4.1 10-Feature formula R$^2$ – 99.833%

The 10-feature formula, which can be seen in Equation 26, exhibits a coefficient R$^2$ of 99.833% on the test dataset. The MAPE for this model was equal to 3.209%. When compared to the other formulae, the 10-feature formula tends to slightly overestimate the value of the fundamental period, which led to resulting the highest error as is going to be shown below. Figures 7 and 8 show the correlation between the predicted and numerical values for the training (Figure 7) and test data (Figure 8).

$$\begin{aligned} T_{10} = &+1.10932(\Phi\xi)^{0.5}\Psi^{-1}T_\beta + 7.01915\times10^{-5}(\Phi\xi)^{0.5}H^2 + 1.13990\times10^{4}\xi^{-0.5}\Psi^{-1} \\ &- 1.12553\xi^{-0.5}LB - 1.07498\times10^{-9}\xi^{-0.5}I_{col}E_s + 6.19508\times10^{-2}T_\beta LB \\ &- 2.01361\times10^{-4}LBH + 3.51950\times10^{-8}\Phi I_{col}H - 3.44050T_\alpha \\ &- 3.51514\times10^{-7}T_\alpha E_s K_{rs} - 7.30737\times10^{-3} \end{aligned} \tag{26}$$

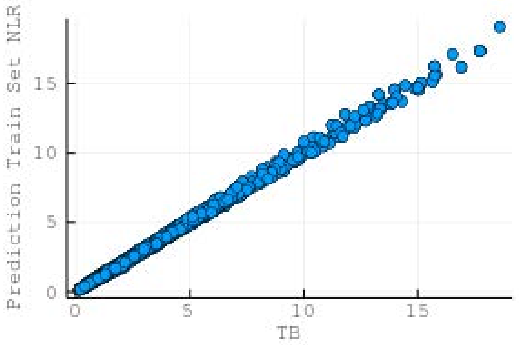

Figure 7: Performance of the 10-feature formula on the training data.

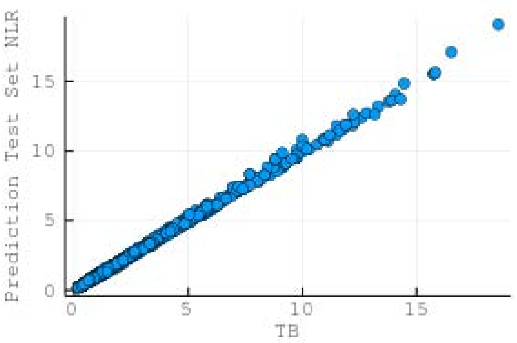

Figure 8: Performance of the 10-feature formula on the test data.

### 4.2 15-Feature formula R$^2$ – 99.945%

In an attempt to further investigate the development of a more accurate model, the number of features of the predictive formula was increased. The 15-feature formula is shown through Equation 27, where it can be seen that it exhibits a, R$^2$ coefficient of 99.945% and a MAPE value of 2.926%. Figures 9 and 10 show the graphical representation between the predicted and numerically obtained values. It is easy to observe that the values are found on a straight line for both the training (Figure 9) and test data (Figure 10). This indicates an improved predictive capability given that the number of equation features was increased.

$$\begin{aligned}T_{15} = &+1.61920 \times 10^{-1}(\Phi\xi)^{0.5}\Psi^{-1}T_{\alpha} + 4.73325 \times 10^{-5}(\Phi\xi)^{0.5}H^{2} + 1.16994 \times 10^{4}\xi^{-0.5}\Psi^{-1}\\ &- 3.55651 \times 10^{-3}(\Phi\xi)^{0.5}T_{\beta}L - 1.91979 \times 10^{1}T_{\beta} - 9.72228 \times 10^{-1}\xi^{-0.5}LB\\ &- 9.87587 \times 10^{-10}\xi^{-0.5}I_{col}E_{s} - 6.24396 \times 10^{-3}\xi^{0.5}\Psi\Phi + 1.02424 \times 10^{-1}\Psi^{-0.5}LT_{\alpha}\\ &- 5.60425 \times 10^{-6}T_{\beta}E_{s}K_{rs} + 5.82434 \times 10^{-3}LBT_{\alpha} - 1.50127 \times 10^{-4}LBH\\ &+ 8.89981 \times 10^{-2}BT_{\alpha} - 4.15783 \times 10^{-3}B^{2}T_{\alpha} + 2.57546 \times 10^{-8}\Phi I_{col}H\\ &+ 4.68534 \times 10^{-3} \end{aligned} \qquad (27)$$

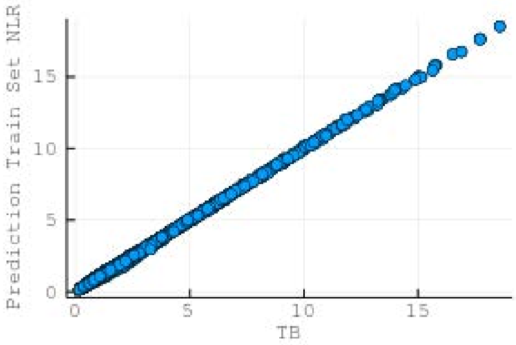

Figure 9: Performance of the 15-feature formula on the training data.

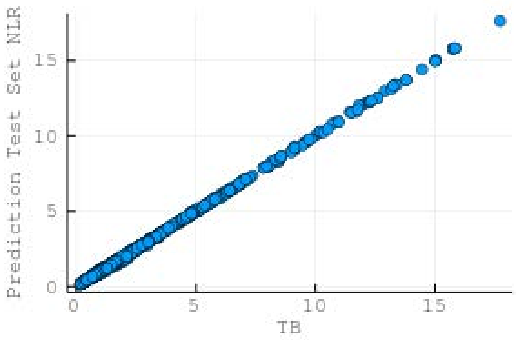

Figure 10: Performance of the 15-feature formula on the test data.

### 4.3 20-Feature formula R$^2$ – 99.957%

The 20-feature formula (see Equation 28) exhibited an R$^2$ of 99.957% and a respective MAPE value of 2.558%. Figures 11 and 12 show the numerically obtained results from the proposed formula for both the training (Figure 11) and test data (Figure 12). The MAPE decreased by 15.6%, according to the numerical results that were obtained from the testing dataset.

$$\begin{aligned}T_{20} = &+8.31846 \times 10^{-1}(\Phi\xi)^{0.5}\Psi^{-1}T_{\beta} - 3.94210 \times 10^{-6}(\Phi\xi)^{0.5}H^{2} - 3.95863 \times 10^{-1}\Phi^{0.5}B\\ &- 8.27183 \times 10^{-3}(\Phi\xi)^{0.5}T_{\beta}L + 1.20957 \times 10^{4}\xi^{-0.5}\Psi^{-1} - 8.13929 \times 10^{-1}\xi^{-0.5}LB\\ &- 6.53561 \times 10^{-10}\xi^{-0.5}I_{col}E_{s} - 2.04532 \times 10^{-2}\xi^{0.5}\Psi\Phi + 2.44129 \times 10^{-5}\xi^{2}\Phi^{3}\\ &+ 3.75710 \times 10^{-10}\xi^{-0.5}I_{col}K_{rs} + 1.40357 \times 10^{-1}\Psi^{-0.5}LT_{\alpha} + 2.00712 \times 10^{-2}T_{\beta}LB\\ &- 2.03899 \times 10^{1}T_{\beta} - 5.65784 \times 10^{-6}T_{\beta}E_{s}K_{rs} + 8.35009 \times 10^{-5}LBH\\ &+ 3.10127 \times 10^{-8}L\Phi I_{col} + 1.11280 \times 10^{-1}BT_{\alpha} - 4.20939 \times 10^{-3}B^{2}T_{\alpha}\\ &- 1.66545 \times 10^{-8}\Phi I_{col}H - 3.00367 \times 10^{-5} - 1.78431 \times 10^{-2} \end{aligned} \qquad (28)$$

### 4.4 30-Feature formula R$^2$ – 99.969%

The number of features was then doubled compared to the 15-feature formula and the retraining led to a new predictive formula, shown in Equation 29. The 30-feature formula derived

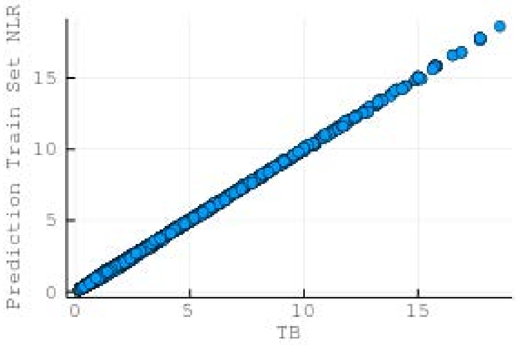

Figure 11: Performance of the 20-feature formula on the training data.

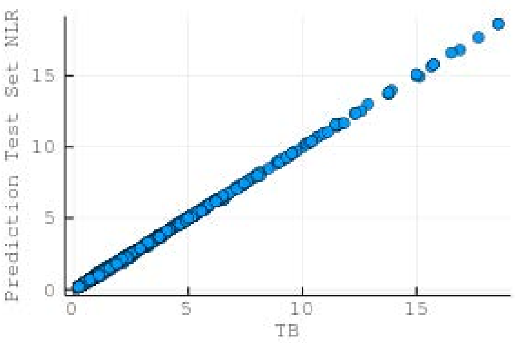

Figure 12: Performance of the 20-feature formula on the test data.

an R$^2$ coefficient of 99.969% and a corresponding MAPE value of 2.412%. Figures 13 and 14 show the correlation between the predicted and numerically computed values for both the training and test data, respectively. The MAPE improved by 17.6% compared to the 15-feature formula, thus it was decided to increase the number of features one final time.

$$\begin{aligned}
T_{30} = {} & -3.98533\times10^{-2}(\Phi\xi)^{0.5}\Psi^{-1}T_{\alpha} + 4.05686\times10^{-5}(\Phi\xi)^{0.5}H^{2} + 9.29522\times10^{3}\xi^{-0.5}\Psi^{-1} \\
& - 2.75683\times10^{-3}(\Phi\xi)^{0.5}T_{\beta}L - 1.45399\times10^{1}T_{\beta} - 6.73233\times10^{-10}\xi^{-0.5}I_{col}E_{s} \\
& - 1.42192\times10^{-1}\xi^{-0.5}LB - 2.28822\times10^{-2}\xi^{0.5}\Psi\Phi + 2.06584\times10^{-2}\xi^{0.5}B\Phi \\
& + 3.69051\times10^{-10}\xi^{-0.5}I_{col}K_{rs} + 4.79394\times10^{-5}\xi^{2}\Phi^{3} + 1.29294\times10^{-1}\xi^{-2}I_{col} \\
& + 3.52721\times10^{-3}\xi B\Phi^{2} - 6.25620\times10^{-3}\xi\Psi^{-1}\Phi T_{\alpha} + 5.17736\times10^{-1}\xi^{-0.5}L^{2} \\
& - 1.41444\xi^{-1}L\Phi^{-1} + 1.31352\Psi^{-0.5}T_{\beta}L - 3.51887\times10^{-4}\Psi BH + 1.70309\times10^{3}\Psi^{-2}\Phi \\
& - 1.52247\times10^{-4}\Psi LH + 1.43678\Psi^{-1}T_{\beta}B + 4.44329\times10^{-1}T_{\beta}B \\
& - 2.63405\times10^{-2}T_{\beta}B^{2} + 2.09772\times10^{-4}LBT_{\alpha} + 3.18239\times10^{-4}LBH \\
& - 5.70132\times10^{-3}L^{2}T_{\alpha} + 2.60988\times10^{-4}B^{2}H + 5.23462\times10^{-9}\Phi I_{col}H \\
& + 4.23547\times10^{-8}HE_{s}K_{rs} - 7.06876\times10^{-7}T_{\alpha}E_{s}K_{rs} - 2.55387\times10^{-2} \qquad (29)
\end{aligned}$$

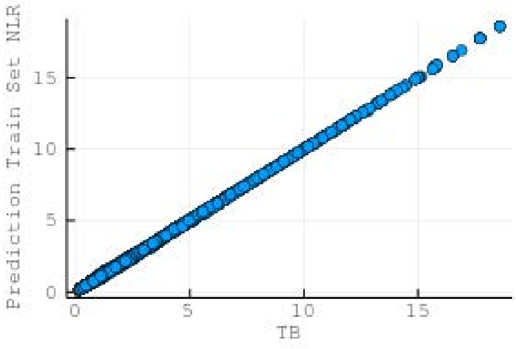

Figure 13: Performance of the 30-feature formula on the training data.

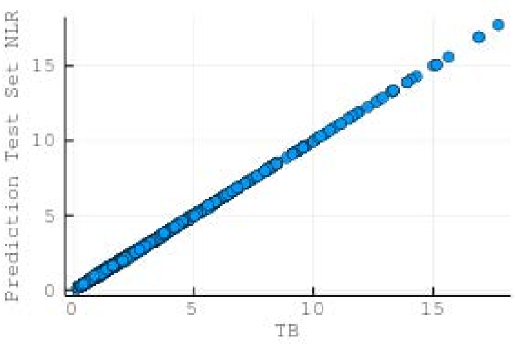

Figure 14: Performance of the 30-feature formula on the test data.

### 4.5 40-Feature formula R$^2$ – 99.976%

The 40-feature formula that is shown in Equation 30, derived an R$^2$ coefficient equal to 99.976% and the lowest MAPE that had a value of 1.969%. Figures 15 and 16 depict the graphical correlation between the predicted and numerically obtained values for both the training (Figure 15) and test data (Figure 16), respectively. It is evident that the proposed 40-feature formula derives outperforms the other models, decreasing the 15-feature formula's MAPE by almost 33%. It is important to note here that the proposed formulae are suitable to be used after programming them in an Excel sheet or a user-friendly application that will allow the user to input the parameters of their building and soil, and compute the fundamental period by just clicking on a calculate button.

$$\begin{aligned}
T_{40} = &+4.10179(\Phi\xi)^{0.5}\Psi^{-1}T_\beta - 5.13549\times10^{-5}(\Phi\xi)^{0.5}H^2 + 1.68757\times10^{-2}\Phi^{0.5}B \\
&- 2.72183\times10^{2}\Phi^{0.5}\Psi^{-1} - 4.61855\times10^{-2}(\Phi\xi)^{0.5}T_\beta^2 + 4.46782\Phi^{0.5}\xi^{-0.5}L \\
&- 1.65486\times10^{-8}(\Phi\xi)^{0.5}T_\beta K_{col} + 4.67885\times10^{3}\xi^{-0.5}\Psi^{-1} - 2.19230\times10^{-4}\xi\Psi\Phi T_\alpha \\
&+ 2.27157\times10^{-1}\xi^{-0.5}LB + 6.41720\times10^{-4}\xi^{-1}E_sK_{rs} - 8.44523\times10^{-8}\xi\Phi^2K_{col} \\
&+ 3.59580\times10^{-10}\xi^{-0.5}I_{col}K_{rs} - 5.88334\times10^{-8}\xi^{-0.5}BI_{col} - 3.52377\xi^{-1}B\Phi^{-1} \\
&+ 6.50545\times10^{-6}\xi^{-0.5}E_sK_{rs} + 7.34562\times10^{-4}\xi\Psi\Phi^2 + 6.39769\times10^{-15}\xi^{-0.5}I_{col}^2 \\
&- 1.54766\times10^{-5}\xi^{-1}BI_{col} + 3.89106\times10^{-1}\Psi^{-0.5}LT_\alpha + 3.74808\times10^{-4}\Psi^2H \\
&+ 6.22854\times10^{-10}\Psi^{-0.5}I_{col}H - 3.83306\times10^{-3}T_\beta LB - 4.83023T_\beta \\
&+ 1.05512\times10^{-5}LBH + 4.53940\times10^{-6}L\Phi K_{col} + 1.75930\times10^{-11}LI_{col}H \\
&- 5.57552\times10^{-10}LI_{col}T_\alpha - 6.25482\times10^{-3}L^2T_\alpha - 1.45786\times10^{-2}LH \\
&+ 9.77408\times10^{-4}L^2H + 9.05671\times10^{-2}BT_\alpha - 5.11814\times10^{-3}B^2T_\alpha \\
&+ 6.38713\times10^{-12}BI_{col}H - 3.39867\times10^{-3}BH + 2.34654\times10^{-4}B^2H \\
&- 6.58079\times10^{-9}\Phi I_{col}H + 2.58301\times10^{-8}\Phi E_sK_{col} - 1.53957\times10^{-7}HE_sK_{rs} \\
&- 1.89967\times10^{-7}T_\alpha E_sK_{rs} + 8.62800\times10^{-2}
\end{aligned} \tag{30}$$

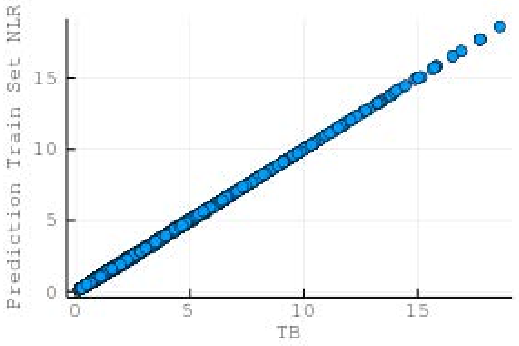

Figure 15: Performance of the 40-feature formula on the training data.

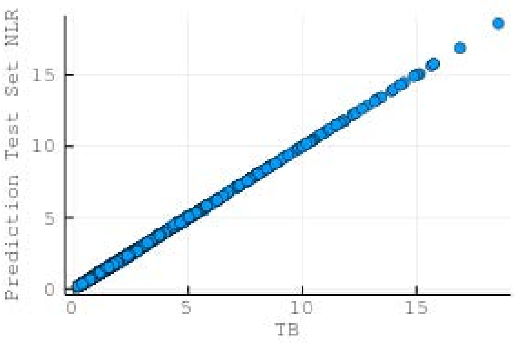

Figure 16: Performance of the 40-feature formula on the test data.

The MAPE and R$^2$ values for each formula were calculated on the test dataset and presented in a summarised manner in Table 3. As can be seen, the MAPE decreases as the number of features increases, a phenomenon that is not always present when performing this type of analysis. This phenomenon is controlled by the characteristics and type of the dataset, whereas the ML algorithm used to derive the relationship between the input and output values also determines the overall numerical response of the different predictive models.

| | $MAPE$ | $R^2$ |
|---|---|---|
| T10 | 3.21% | 99.83% |
| T15 | 2.93% | 99.95% |
| T20 | 2.56% | 99.96% |
| T30 | 2.41% | 99.97% |
| T40 | 1.97% | 99.98% |

Table 3: Performance summary of the proposed formulae on the test dataset (MAPE, and R$^2$ values)

## 5 CONCLUSION AND RECOMMENDATIONS

This research work presented a newly developed automated algorithm that is responsible to handle a base input file, generating new input files, and then analyzing them through the use of Reconan FEA (2020). The ability to automatically generate multiple input files and analyze them allowed for the generation of a dataset that consisted of 10,368 data points. This was performed through the use of a standard PC demonstrating the efficiency of the proposed algorithm.

Furthermore, 5 new formulae for predicting the fundamental period of steel framed structures with or without the SSI effect, were proposed through the use of an advanced nonlinear regression algorithm. The new formulae are capable of accounting for various aspects of the structural system, including the soil elasticity, structural geometry, as well as the cross-sectional properties of the column members of the structures. The proposed predictive formulae are to be used only for buildings that have geometries within the geometrical limits of the building models used to train and test the predictive models.

Future research work foresees the increase of the dataset through the use of the proposed automated algorithms and high-performance computers in developing a dataset that will consist of more than one million data points. This will allow future engineers to compute the fundamental period of steel structures through the use of ML-generated models.

## 6 ACKNOWLEDGMENTS

The financial support from the EuroCC project (GA 951732) and EuroCC 2 project (101101903) of the European Commission is acknowledged. Parts of the runs were performed on the MeluXina (https://docs.lxp.lu/) as well as Cyclone (https://hpcf.cyi.ac.cy/) Supercomputers.

## REFERENCES

[1] Ch. Mourlas, N. Khabele, H.A. Bark, D. Karamitros, F. Taddei, G.Markou, and M. Papadrakakis, *The Effect of Soil-Structure Interaction on the Nonlinear Dynamic Response of Reinforced Concrete Structures*, International Journal of Structural Stability and Dynamics, 20(13), 2020, 2041013

[2] Z.D. Gravett, C. Mourlas, V.L. Taljaard, P.N. Bakas, G. Markou, and M. Papadrakakis, *Fundamental Period Formulae for Soil-Reinforced Concrete Structures Interaction Using*

*Machine Learning Fundamental Period Formulae for Soil-Reinforced Concrete Structures Interaction Using Machine Learning*, Soil Dynamics and Earthquake Engineering, 144, 2021, 106656

[3] A. van der Westhuizen, G. Markou, and N. Bakas, *Development of a New Fundamental Period Formula for Steel Structures Considering the Soil-structure Interaction with the Use of Machine Learning Algorithms*, ICAART 2022, February 3-5, 2022

[4] G. Markou and N.P. Bakas, *Prediction of the shear capacity of reinforced concrete slender beams without stirrups by applying artificial intelligence algorithms in a big database of beams generated by 3D nonlinear finite element analysis*, Computers and Concrete, 28(6), 2021, 533-547

[5] N. Carstens, G. Markou, and N. Bakas, *Improved Predictive Fundamental Period Formula for Reinforced Concrete Structures through the Use of Machine Learning Algorithms*, ICAART 2022, February 3-5, 2022

[6] E. Ababu, G. Markou, and N. Bakas, *Using Machine Learning and Finite Element Modelling to Develop a Formula to Determine the Deflection of Horizontally Curved Steel I-beams*, ICAART 2022, February 3-5, 2022

[7] M. AlHamaydeh, G. Markou, N.P. Bakas, and M. Papadrakakis, *AI-based shear capacity of FRP-reinforced concrete deep beams without stirrups*, Engineering Structures, 264, 2022, 114441

[8] Z. Spijkerman, N.P. Bakas, G. Markou, and M. Papadrakakis, *Predicting the Shear Capacity of Reinforced Concrete Slender Beams Without Stirrups by Applying Artificial Intelligence Algorithms*, COMPDYN 2021

[9] Z.D. Gravett, and G. Markou, *State-of-the-art Investigation of Wind Turbine Structures Founded on Soft Clay by Considering the Soil-Foundation-Structure Interaction Phenomenon – Optimization of Battered RC Piles*, Engineering Structures, 235, 2019, 112013

[10] G. Markou, and M. Papadrakakis, *A simplified and efficient hybrid finite element model (HYMOD) for non-linear 3D simulation of RC structures*, Engineering Computations, 32(5), 2015, 1477-1524

[11] J. Knappett, R. Craig, *Craig's Soil Mechanics (8th ed.)*, pg.372, CRC Press, 2012.

[12] G. Markou, N.P. Bakas, S.A. Chatzichristofis, and M. Papadrakakis, *High-Performance Machine Learning Algorithms for Structural Engineering*, Submited for publication.

[13] Reconan FEA v2.00, *User's Manual*, 2020.

[14] D. Gravett, C. Mourlas, V.L. Taljaard, N. Bakas, G. Markou, M. Papadrakakis, *New Fundamental Period Formulae for Soil-Reinforced Concrete Structures Interaction using Machine Learning Algorithms And Anns*

## APPENDIX A - Automated Visual Basic algorithm for model generation

```
Sub Generate()
    Dim App As Object
    Set App = GetObject(, "femap.model")
    Dim Pr As Object
    Set Pr = App.feProp
    Row = 2
    ID = Worksheets(1).Cells(Row, 1).Value
    SName = Worksheets(1).Cells(Row, 2).Value
    FName = Worksheets(1).Cells(Row, 7).Value
    h = Worksheets(1).Cells(Row, 3).Value
    b = Worksheets(1).Cells(Row, 4).Value
    tf = Worksheets(1).Cells(Row, 5).Value
    tw = Worksheets(1).Cells(Row, 6).Value
    Sum = 0
    While ID $>$ 0
        Pr.Get (3)
        Pr.matlID = 1
        Pr.Type = 5
        Pr.flagI(1) = 9
        Pr.flagI(3) = 101
        Pr.pval(40) = h
        Pr.pval(41) = b
        Pr.pval(42) = b
        Pr.pval(43) = tf
        Pr.pval(44) = tf
        Pr.pval(45) = tw
        Pr.Title = SName
        Dim Dr(6) As Double
        Dr(0) = Pr.pval(40)
        Dr(1) = Pr.pval(41)
        Dr(2) = Pr.pval(42)
        Dr(3) = Pr.pval(43)
        Dr(4) = Pr.pval(44)
        Dr(5) = Pr.pval(45)
        rc = Pr.ComputeStdShape(9, Dr, 0, 2, False, True, True)
        Pr.pval(46) = 400
        Pr.pval(47) = 2
        Pr.pval(48) = 0
        Pr.pval(49) = 0
        Pr.pval(50) = 0
        Pr.pval(51) = 0
        Pr.pval(52) = 200000000
        Pr.pval(53) = 0.3
        Pr.pval(54) = 0
        Pr.pval(55) = 355500
        Pr.pval(56):Pr.pval(67) = 0
        Pr.pval(68) = 5
        Pr.pval(69) = 0
        Pr.pval(70) = 0.3
        Pr.pval(71) = 1
        Pr.pval(72) = 2
        Pr.pval(73) = 1
        Pr.Put (3)
        rc = App.feFileWriteNeutral2(0, FName, False, True, False, True,
             True, False, False, False, True, True,8, 9, 0)
        Row = Row + 1
        ID = Worksheets(1).Cells(Row, 1).Value
        SName = Worksheets(1).Cells(Row, 2).Value
        FName = Worksheets(1).Cells(Row, 7).Value
        h = Worksheets(1).Cells(Row, 3).Value
        b = Worksheets(1).Cells(Row, 4).Value
        tf = Worksheets(1).Cells(Row, 5).Value
        tw = Worksheets(1).Cells(Row, 6).Value
        Sum = Sum + 1
    Wend
    rc = App.feAppMessage(0, "Import Done of " Sum " Section(s)")
    App.feFileClose (False)
End Sub
```

## APPENDIX B - Python Script for the Automation of the Analysis and Data Collection Procedure

**Description:** Takes an ”INPUT” array containing all input files appended with ”\n”, and checks if the corresponding ”OUTPUT” file exists in the corresponding array. If it doesn’t exist, it runs the FEA software Reconan FEA (2020), creating the file upon completion of the analysis. Once all ”OUTPUT” files exist, it creates a ”RESULTS.xlsx” - excel workbook and records the specific parts of interest from the ”OUTPUT” files in the database. INPUT files were named to be unique, by containing the variable information as part of their names, which aids in later filling of the database with more information.

```
from subprocess import Popen, PIPE
import os, xlsxwriter
rowstart, RESULTS, exename = 3, ’RESULTS.xlsx’,’\\\\ReconanFEA.exe’
#Gets current working folder location as a string object with:
direct = (os.path.dirname(__file__))
mypath = direct.replace(’/’,’\\\\’)
exe_path = f"{mypath}{exename}"
#ARRAY/LIST INPUT AND CRITERIA OUTPUT
INPUT  = [’INPUT001.neu\n’,’INPUT001.neu\n’,etc.]
OUTPUT = [’OUTPUT001.dat’ ,’OUTPUT001.dat’ ,etc.]
i=0
while i <len(INPUT):
  if os.path.exists(OUTPUT[i]) == False:
     CHNG = CHNG+1
     print("Using/Running:",INPUT[i],len(INPUT)-i,"left")
     process = Popen([exe_path], stdin=PIPE, stdout=PIPE, text=True)
     try:
      print(process.communicate(input=INPUT[i]))[0]
     except:
       continue
  else:
    print(OUTPUT[i]," exists")
    i=i+1
#WRITES CERTAIN LINES FROM CREATED {OUTPUT} FILES TO SINGLE[***NEW***]
#EXCEL DOC
WB = xlsxwriter.Workbook(RESULTS)
WS = WB.add_worksheet()
#WRITES HEADINGS IN COUNTABLE ROW = ROWSTART (I.E. N = N-1)
WS.write(rowstart-1,1,"Sections")
for col in range (2,11):
  WS.write(rowstart-1,col,col-1)
#DATAFILL
for ROW in range(rowstart,len(INPUT)+rowstart):
  N = ROW-rowstart
  WS.write(ROW,1,INPUT[N])
  dat = open(OUTPUT[N])
  content = dat.readlines()
  for line in range (10,20):
    Txt=content[line]
    val = Txt[len(Txt)-8:len(Txt)]
    WS.write(ROW,line-8,val)
  dat.close()
WB.close()
```