World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:5, 2021

# Using Analytical Hierarchy Process and TOPSIS Approaches in Designing a Finite Element Analysis Automation Program

Ming Wen, Nasim Nezamoddini

***Abstract*****—**Sophisticated numerical simulations like finite element analysis (FEA) involve a complicated process from model setup to post-processing tasks that require replication of time-consuming steps. Utilizing FEA automation program simplifies the complexity of the involved steps while minimizing human errors in analysis set up, calculations, and results processing. One of the main challenges in designing FEA automation programs is to identify user requirements and link them to possible design alternatives. This paper presents a decision-making framework to design a Python based FEA automation program for modal analysis, frequency response analysis, and random vibration fatigue (RVF) analysis procedures. Analytical hierarchy process (AHP) and technique for order preference by similarity to ideal solution (TOPSIS) are applied to evaluate design alternatives considering the feedback received from experts and program users.

***Keywords*****—**FEA, random vibration fatigue, process automation, AHP, TOPSIS, multiple-criteria decision-making, MCDM.

## I. Introduction

FEA is a mathematical approximation method to predict the systems' physical behaviors, such as the distortion of a truck chassis under vibration loading. Various commercial FEA software packages are applied to complete tasks in different areas in industries to assist decision making in product development. FEA requires the user to accurately define and formulate the engineering problem into a computer model and solve it using different software packages.

A standard FEA procedure for structural analysis includes six steps, as shown in Fig. 1. The input of the procedure is the CAD geometry of the structure, as shown in Step 1. In Step 2, parts of the structure are meshed individually using preprocessing software. In this step, the structure is broken down into individual parts, and then each part is discretized into elements with different configurations and mechanical properties. In Step 3, meshed parts are virtually assembled according to the original structural topology. In Step 4, boundary conditions are defined in the model for different analysis purposes. The model is submitted to solvers for calculation in Step 5. In Step 6, the solved model is processed using post-processing software to demonstrate system output and results. Among all six steps, Step 4 and Step 5 directly determine the validation and accuracy of the simulation results. These two steps require professionally trained engineers to execute the model. In many analyses, this includes repetitive model set up tasks where an engineer may make mistakes during that procedure.

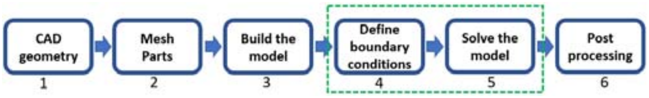


Fig. 1 FEA procedure

In comparison with operating each step manually, using process automation programs to complete Step 4 and Step 5 will reduce the complexity of the analysis procedure. For example, the RVF analysis procedure includes multiple steps, boundary conditions are applied to the FE model, then the model is solved. The results of the solved model are used as input for fatigue analysis. Moreover, an engineer may make mistakes when manually executing this analysis procedure. Using a program that automatically executes routine analysis steps will reduce the chance of calculation mistakes.

In this paper, a program is developed to automate Step 4 and Step 5 in the FEA procedure to improve work efficiency. The program is used for the modal analysis, frequency response analysis, and RVF analysis procedures. Among these three types of analysis, the RVF analysis is the most complex one. It is used to predict fatigue life of parts in a structure when the structure is experiencing random vibration loading which cannot be described as a function of time but can be characterized statistically.

To develop the process automation program, one challenge is to identify and evaluate all technical needs to meet user requirements, which can be expressed as a multiple-criteria decision-making (MCDM) problem. In this paper, to find the best design alternatives and increase efficiency of the system, an AHP is applied to rank user requirements and then they are mapped to design characteristics through correlation matrices. Technique for order preference by similarity to ideal solution (TOPSIS) is applied to evaluate alternatives and find the optimal combination of design characteristic values. Based on the target values of design characteristics, the process automation program is developed using Python. A model is used to demonstrate the RVF FEA procedure.

## II. Literature Review

Karlsson et al. [1] compared six methods for prioritizing software quality requirements, which include performance, design, and adaptation criteria. According to the research AHP

Ming Wen, Ph.D. is with the Oakland University, Rochester, MI 48309 USA (corresponding author, phone: 313-421-3634; e-mail: mingwen@oakland.edu).

Nasim Nezamoddini, Ph.D. is with Oakland University, Rochester, MI 48309 USA. (phone: 248-370-2989; e-mail: nezamoddini@oakland.edu).

is the most promising method because this method yields the most trustworthy results. Melegati et al. [2] built a model to analyze the executing of requirements engineering in software startups, which do not follow a single set of practice, but build a custom process. Sadiq et al. [3] used a web survey approach to elicit the software requirements and used AHP to rank alternatives, which resulted in improving the system. Scacchi [4] studied four open software development communities and investigated the requirements for open software development efforts. Scacchi et al. [4] also studied the development of these requirements different from those traditional software engineering and requirements engineering. Scacchi et al. [4] found eight kinds of software informalisms that play a critical role in the elicitation, analysis, specification, validation and management of requirements for developing open software systems. Lin et al. [5] applied AHP and TOPSIS to analyze customer requirements and design characteristics, achieving an effective evaluation of the final design solution. Reddy et al. [6] applied AHP, integer programming, and goal programming for selection of Software Architecture. A framework was proposed for dealing with multi objective functions. Based on AHP, Benlian [7] proposed a framework to evaluate the relative fulfilment of key selection criteria by traditional, open-source, and on-demand office suites. Agrawal [8] used integrating Fuzzy AHP and Fuzzy TOPSIS based framework to assess web applications' sustainable security.

FEM based RVF analysis is applied to assess the max stress of components by the vibration loading which can only be described as power spectrum density (PSD) [9]. Yu et al. [10], [11] developed an assessment methodology to predict the fatigue life of electronic components under random vibration loading. The assessment methodology includes two steps. The first step is to characterize fatigue properties through a test. The second step is to obtain the response PSD of the critical solder joint and assess the fatigue life using rain-flow cycle counting. Chen et al. [12] developed a methodology that combines the vibration failure test, FEA, and theoretical formulation, to estimate the fatigue life of a specially designed plastic Ball Grid Array (BGA) under vibration loading. In this method, FEA is used to obtain the stress. The S-N curve is constructed by correlating FEA obtained stresses and tested data. Miner's rule is applied to calculate the fatigue damage index. Al-Yafawi et al. [13], [14] developed a fatigue life estimation procedure. In the procedure, material's fatigue curves are characterized, then the FE model is built and validated by correlating the natural frequencies, mode shapes and transmissibility functions from simulation with experimental results. Finally, material fatigue data and damage rules are combined to estimate the fatigue life.

Julien et al. [15] investigated the effects of stiffness of the bolted connection, Young's modulus and density of material and variation of these parameters on the modal analysis results, to identify the most influential parameter. This research provides a method to estimate the probability of system failure.

## III. Analyzing User Requirements and Design Characteristics

AHP is an effective methodology to describe people's subjective judgments. In this paper, the consultative AHP procedure developed by Lin et al. [5] is applied to identify important user requirements and designer characteristics, as shown in Fig. 2.

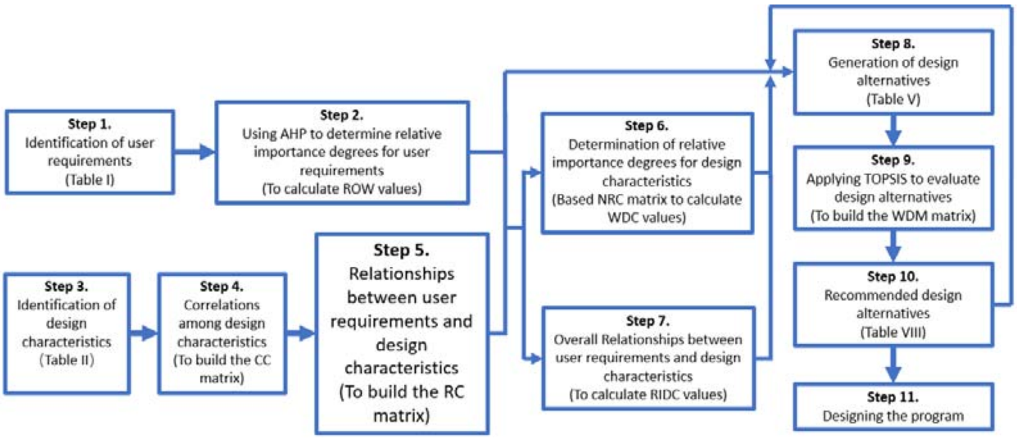

Step 1.
Identification of user requirements
(Table I)
Step 2.
Using AHP to determine relative importance degrees for user requirements
(To calculate ROW values)
Step 3.
Identification of design characteristics
(Table II)
Step 4.
Correlations among design characteristics
(To build the CC matrix)
Step 5.
Relationships between user requirements and design characteristics
(To build the RC matrix)
Step 6.
Determination of relative importance degrees for design characteristics
(Based NRC matrix to calculate WDC values)
Step 7.
Overall Relationships between user requirements and design characteristics
(To calculate RIDC values)
Step 8.
Generation of design alternatives
(Table V)
Step 9.
Applying TOPSIS to evaluate design alternatives
(To build the WDM matrix)
Step 10.
Recommended design alternatives
(Table VIII)
Step 11.
Designing the program

Fig. 2 Consultative AHP procedure steps

The first seven steps are the consultative AHP procedure. This procedure maps user requirements into design characteristics through ranking the overall importance of design characteristics and calculating design characteristics' contribution to each user requirement. AHP is used to evaluate user requirements.

From the eighth step to the tenth step are the design and evaluation process. In this paper, based on previous analysis obtained results and engineering judgments, design alternatives are proposed. TOPSIS is used to evaluate design alternatives.

The last step, the eleventh step, is to implement the design alternative. In this step, the program is built using Python.

In application, before determining the final product, to respond to the changing in user requirements and design characteristics, the consultative AHP procedure and the design and evaluation process might be applied in multiple iterations independently or dependently. In this paper, only one iteration is applied to demonstrate the procedure.

The following are the execution of the 10 steps to design the FEA automation program.

The first step is to identify user requirements, based on descriptions by engineers using brainstorming and discussion. User requirements are classified into three clusters: interface, functions, and others. Under each cluster, detailed requirements are listed in Table I. R$_1$, R$_2$, and R$_3$ are categories of user requirements. For each category, user requirements are denoted as R$_{i,j}$, i is the category number, j represents the detailed user requirement under each category.

TABLE I
HIERARCHICAL LIST OF USER REQUIREMENTS

| $R_n$ and $R_{i,j}$ | User Requirements | Importance degrees of user requirements (W$_i$) | Relative Overall Importance degrees of user requirements (ROW$_i$) |
|---|---|---|---|
| $R_1$ | Cluster I: Interface | 0.18 (Priority value) | |
| $R_{11}$ | Simple/Intuitive | 0.64 | 0.11 |
| $R_{12}$ | Color | 0.10 | 0.02 |
| $R_{13}$ | Graphics/shapes | 0.26 | 0.05 |
| $R_2$ | Cluster II: Functions | 0.75 (Priority value) | |
| $R_{21}$ | Modal Analysis | 0.36 | 0.27 |
| $R_{22}$ | RVF Analysis | 0.44 | 0.33 |
| $R_{23}$ | Installation | 0.07 | 0.06 |
| $R_{24}$ | Robust | 0.12 | 0.09 |
| $R_3$ | Cluster III: Others | 0.07 (Priority value) | |
| $R_{31}$ | Easy to get help | 1 | 0.07 |

The second step is to calculate relative overall importance degrees (ROW) for user requirements. One paired comparison matrix for clusters and two matrices for interface requirements and function requirements are developed.

Equation (1) is used to calculate the importance degree of user requirements (W) under each cluster and priority values of clusters, as listed in the third column of Table I. For each cluster, the element of paired comparison matrices for user requirements is denoted by $a_{i,j}$.

$$w_i = \frac{(\prod_{j=1}^{n} a_{ij})^{1/n}}{\sum_{i=1}^{n}(\prod_{j=1}^{n} a_{ij})^{1/n}} \tag{1}$$

The maximum eigenvalues, $\lambda_{\max}$, for developed three matrices are 3.03, 3.04, 4.09; consistency ratios (CR) are 0.03, 0.03, 0.03. All CR values are below 0.1, which means inconsistency is acceptable.

The ROW values are W values multiplied by each cluster's priority value. Compared to the reference procedure developed by [5], customer performance satisfactions, improvement ratio, and potential sales increments of user requirements are not considered in the calculation. The ROW values are listed in the fourth column of Table I.

The third step is to identify the primary design characteristics. The primary design characteristics of the program are listed in Table II. These characteristics are identified based on the designer's observation on other programs.

The set of design characteristics representing the product features is denoted as [D1, D2, D3, D4, D5], a total of five clusters and their corresponding alternative values are included and structured into a hierarchical list, as shown in Table II. Design characteristics are evaluated using (1), and corresponding priority values are listed in Table II.

TABLE II
DESIGN CHARACTERISTICS AND THEIR ASSOCIATED ALTERNATIVE VALUES

| $D_m$ | Design characteristic | Design alternatives |
|---|---|---|
| $D_1$ | Layout Style | 0.26 (Priority value) |
| | | Basic menu style |
| | | Vertical menu style |
| | | Drop-down menu style |
| | | Secondary menu style |
| $D_2$ | Progress Indicator | 0.14 (Priority value) |
| | | Change color of button |
| | | Progress bar |
| | | Progress circle |
| $D_3$ | Deployment Method | 0.06 (Priority value) |
| | | Copy all files to the user |
| | | Automatically installment |
| | | Guided installment |
| $D_4$ | Feedback | 0.03 (Priority value) |
| | | Send message to an email |
| | | A button to send message |
| | | A "Help" button with a list of FAQ |
| $D_5$ | Software | 0.51 (Priority value) |
| | | OptiStruct |
| | | ABAQUS |
| | | LS-DYNA |
| | | Nastran |

The fourth step is to identify the correlations among design characteristics. This step assists designers to clarify the relationships among design characteristics. In Table III, the rating scales of 1, 3, and 9 are used to represent weak, moderate, and strong positive correlations; corresponding negative correlations using the rating scales -1, -3, and -9. Blank means no relationship exists, as shown in Table III. The element of the design characteristic correlation matrix is denoted by $CC_{k,j}$.

TABLE III
THE DESIGN CHARACTERISTICS CORRELATION MATRIX

| | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|
| $D_1$ | 1 | 9 | - | 9 | 1 |
| $D_2$ | 0.11 | 1 | - | 9 | 3 |
| $D_3$ | - | - | 1 | - | 3 |
| $D_4$ | 0.11 | 0.11 | - | 1 | 3 |
| $D_5$ | 1 | 0.33 | 0.2 | 0.33 | 1 |

The fifth step is to identify the strength of the relationships between user requirements and design characteristics. In Table IV, the rating scales of 1, 3, and 9 are used to represent weak, moderate, and strong positive correlations; corresponding negative correlations using the rating scales -1, -3, and -9. Blank means no relationship exists. The element of the user requirements and design characteristics correlation matrix is denoted by $RC_{k,j}$.

TABLE IV
THE USER REQUIREMENTS AND DESIGN CHARACTERISTICS CORRELATION MATRIX

| | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|
| $R_{11}$ | 9 | 9 | 9 | 9 | - |
| $R_{12}$ | 9 | 3 | - | - | - |
| $R_{13}$ | 9 | 9 | - | 9 | - |
| $R_{21}$ | 3 | 3 | - | - | - |
| $R_{22}$ | 3 | 9 | - | - | - |
| $R_{23}$ | 9 | 9 | 9 | 9 | - |
| $R_{24}$ | - | - | - | 3 | 9 |
| $R_{31}$ | 1 | - | 9 | 9 | 9 |

The sixth step is to determine the relative importance degree of design characteristics (WDC). The normalized correlation coefficients between user requirements and design characteristics (NRC) are calculated according to (2):

$$NRC_{i,j} = \frac{\sum_{k=1}^{m} RC_{i,k} CC_{k,j}}{\sum_{i=1}^{n} \sum_{k=1}^{m} RC_{i,k} CC_{k,j}} \tag{2}$$

The WDC values are calculated using (3):

$$WDC_j = \sum_{i=1}^{n} ROW_i \times NRC_{i,j} \tag{3}$$

The WDC values for each designer characteristics D1, D2, D3, D4 and D5 are 0.05 (layout style), 0.19 (progress indicator), 0.02 (deployment method), 0.51 (feedback), and 0.23 (software). Higher WDC values indicate corresponding design characteristics exerts higher influence on all user requirements, this means a greater potential of the design characteristic to meet user requirements.

The seventh step is to establish relative overall relationships of user requirements to design characteristics (RIDC), which indicates the influence of a specific user requirement on all design characteristics.

The RIDC is calculated using (4) and (5):

$$IDC_i = \sum_{j=1}^{m} \sum_{k=1}^{m} RC_{i,j} CC_{j,k} \text{ , j = 1, 2, …, n} \tag{4}$$

$$RIDC_i = \frac{IDC_i}{\sum_i^n IDC_i} \tag{5}$$

The RIDC for each user requirements are listed in Table V.

TABLE V
RICD VALUES

| | User Requirements | RIDC |
|---|---|---|
| $R_{11}$ | Simple/Intuitive | 0.21 |
| $R_{12}$ | Color | 0.13 |
| $R_{13}$ | Graphics/shapes | 0.19 |
| $R_{21}$ | Modal Analysis | 0.06 |
| $R_{22}$ | RVF Analysis | 0.10 |
| $R_{23}$ | Installation | 0.21 |
| $R_{24}$ | Robust | 0.02 |
| $R_{31}$ | Easy to get help | 0.07 |

A higher RIDC value means that the corresponding user requirement exerts higher influence on all design characteristics.

## IV. PROGRAM CONCEPTUAL DESIGN AND EVALUATION

The eighth step of the process is to generate design alternatives. Based on previous analysis and engineering judgment, four conceptual designs, DA1, DA2, DA3, and DA4, with different combinations of design alternatives are proposed and listed in Table VI.

TABLE VI
THE ATTRIBUTES OF DESIGN ALTERNATIVES

| | $DA_1$ | $DA_2$ | $DA_3$ | $DA_4$ |
|---|---|---|---|---|
| $D_1$ | Vertical menu style | Basic menu style | Drop-down menu style | Secondary menu style |
| $D_2$ | Progress bar | Change color of button | Progress circle | Change color of button |
| $D_3$ | Copy all files to the user | Automatically installment | Guided installment | Automatically installment |
| $D_4$ | Send message to an email | A “Help” button with a list of FAQ | A button to send message | Send message to an email |
| $D_5$ | OptiStruct | Abaqus | LS-Dyna | Nastran |

The ninth step is to use TOPSIS to evaluate design alternatives. TOPSIS is based on the concept that the chosen alternative should have the shortest geometric distance from the positive ideal solution (PIS) and the longest geometric distance from the negative ideal solution (NIS). The TOPSIS procedure is applied in the competitive benchmarking stage to link the mental model of the user and the conceptual model of the designer.

The normalized decision matrix (NDM) is established using (6):

$$NDM_{i,r} = \frac{a_{i,r}}{\sqrt{\sum_{r=1}^{s} a_{i,r}^2}} \tag{6}$$

The NDM values are listed in Table VII.

TABLE VII
THE NDM OF DESIGN ALTERNATIVES

| | $DA_1$ | $DA_2$ | $DA_3$ | $DA_4$ |
|---|---|---|---|---|
| $R_{11}$ | 0.49 | 0.63 | 0.35 | 0.49 |
| $R_{12}$ | 0.46 | 0.59 | 0.33 | 0.59 |
| $R_{13}$ | 0.23 | 0.55 | 0.39 | 0.70 |
| $R_{21}$ | 0.52 | 0.67 | 0.37 | 0.37 |
| $R_{22}$ | 0.52 | 0.67 | 0.37 | 0.37 |
| $R_{23}$ | 0.37 | 0.67 | 0.52 | 0.37 |
| $R_{24}$ | 0.50 | 0.50 | 0.50 | 0.50 |
| $R_{31}$ | 0.52 | 0.67 | 0.37 | 0.37 |

The weighted normalized decision matrix WDM is established using (7):

$$b_{i,r} = W_i \frac{a_{i,r}}{\sqrt{\sum_{r=1}^{s} a_{i,r}^2}} \tag{7}$$

The WDM values are listed in Table VIII. For the same design characteristic, a higher WDM value means the design alternative is closer to the ideal solution.

Based on the results of WDM, the positive ideal solution (PIS) would be {0.01, 0.03, 0.18, 0.22, 0.04, 0.05, 0.05}; the negative ideal solution (NIS) would be {0.01, 0.01, 0.10, 0.12, 0.02, 0.05, 0.03}.

TABLE VIII
THE WDM OF DESIGN ALTERNATIVES

| | DA1 | DA2 | DA3 | DA4 | PIS | NIS |
|---|---|---|---|---|---|---|
| $R_{11}$ | 0.06 | 0.07 | 0.04 | 0.06 | 0.07 | 0.04 |
| $R_{12}$ | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| $R_{13}$ | 0.01 | 0.03 | 0.02 | 0.03 | 0.03 | 0.01 |
| $R_{21}$ | 0.14 | 0.18 | 0.10 | 0.10 | 0.18 | 0.10 |
| $R_{22}$ | 0.17 | 0.22 | 0.12 | 0.12 | 0.22 | 0.12 |
| $R_{23}$ | 0.02 | 0.04 | 0.03 | 0.02 | 0.04 | 0.02 |
| $R_{24}$ | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| $R_{31}$ | 0.04 | 0.05 | 0.03 | 0.03 | 0.05 | 0.03 |
| $b_r^+$ | 0.072 | 0.007 | 0.134 | 0.131 | | |
| $b_r^-$ | 0.067 | 0.135 | 0.011 | 0.028 | | |
| $\mathrm{RC}_r$ | 0.48 | 0.95 | 0.08 | 0.17 | | |

$$PIS = \{\max b_{i,r}\} = (b_1^+, b_2^+, \dots, b_n^+) \tag{8}$$

$$NIS = \{\min b_{i,r}\} = (b_1^-, b_2^-, \dots, b_n^-) \tag{9}$$

$$b_r^+ = \sqrt{\sum_{i=1}^{n}(b_{i,r} - b_i^+)^2} \tag{10}$$

$$b_r^- = \sqrt{\sum_{i=1}^{n}(b_{i,r} - b_i^-)^2} \tag{11}$$

$$RC_r = \frac{b_r^-}{b_r^+ + b_r^-} \tag{12}$$

The separation distance measures, $b_r^+$ and $b_r^-$, of each competitive design alternative from the positive ideal solution are (0.07, 0.007, 0.13, 0.13), and from the negative ideal solutions are (0.065, 0.95, 0.08, 0.15).

The tenth step is to recommend design alternatives. In this step, based on previous search strategy, the target combination of design alternatives of the program is that the program has: basic menu style; a changing color button to show progress; automatic installation on user's computer; and Abaqus as the solver, as listed in Table IX.

TABLE IX
THE ALTERNATIVE VALUES OF DESIGN CHARACTERISTICS

| | Design characteristic | Alternative value |
|---|---|---|
| $D_1$ | Layout Style | Basic menu style |
| $D_2$ | Progress Indicator | Change color of button |
| $D_3$ | Deployment Method | Automatically installment |
| $D_4$ | Feedback | A "Help" button with a list of FAQ |
| $D_5$ | Software | ABAQUS |

## V. PROGRAM DESIGN AND RVF ANALYSIS PROCEDURE

Previous analysis decided the optimal combination of alternatives values for the program, as listed in Table IX. In the eleventh step, all decided alternative values are implemented in the design using Python.

Fig. 3 shows the program's graphical user interface (GUI), which is a basic menu style. Buttons in the GUI, through changing their color, indicate that the program is running, this feature is used to reduce user's uncertainty while waiting for the results. The program can be saved as an executable file and installed into the target computer. The "HELP" button is used to provide more information to the user to use the program.

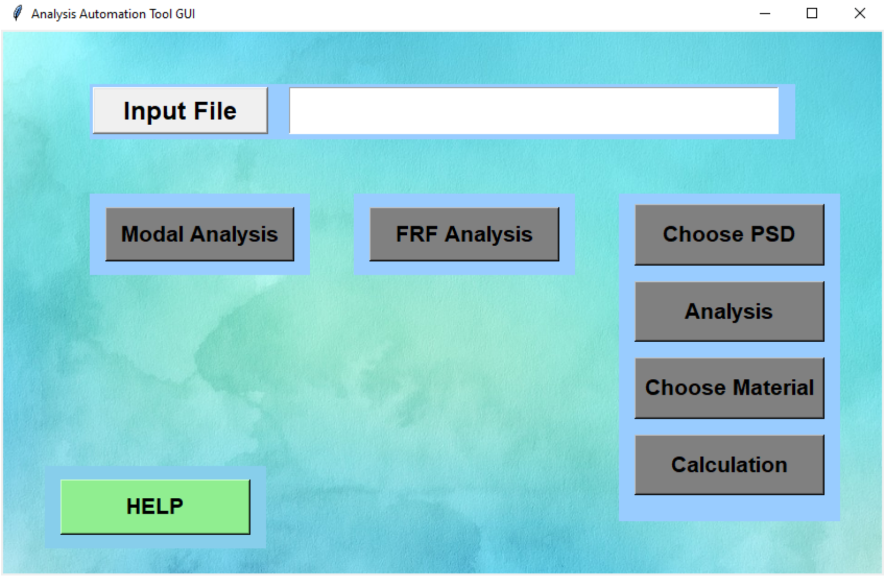


Fig. 3 GUI of the program

The program simplified the procedures into several buttons with limited selections in each step. The "Modal Analysis" button is designed for modal analysis. After clicking the button, the model will be edited and then submitted for solving automatically. The "FRF Analysis" button is design for frequency response analysis. After clicking this button, the model will be edited and then submitted for solving automatically.

The "Choose PSD", "Analysis", "Choose Material", and "Calculation" buttons are designed to complete the RVF analysis. After clicking the "Choose PSD" button, users need to choose a PSD data as the input loading to the model, then click the "Analysis" button to edit and submit the model to the solver. After the simulation is completed, users need to click the "Choose Material" button to choose fatigue material data, then click the "Calculation" button to use the selected fatigue material data and previous analysis results as input to calculate the fatigue life of each part in the model.

In Fig. 4, buttons for analysis on the GUI and the corresponding automated steps are listed.

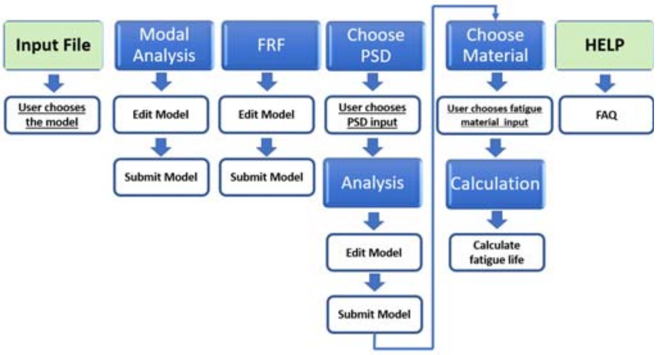


Fig. 4 Automated steps for each button

After clicking each button of the program, the previously manually operated steps are executed automatically. Users are required to make fewer choices than before. Through automating repeating steps, human-introduced errors are reduced, and work efficiency is improved.

A vehicle frame model is used to demonstrate the RVF analysis procedure, which is with more analysis steps than modal analysis and frequency response analysis.

The model used for analysis includes three parts, the frame, front bar, and bumper, as shown in Fig. 5. The bumper and frame are assigned with aluminum material properties. The yield stress of the aluminum is 680 MPa; the Front bar is assigned with steel properties, with yield stress equal to 750 MPa.

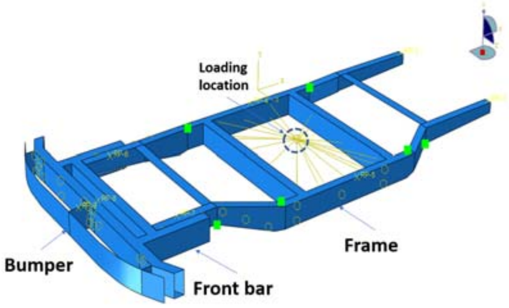
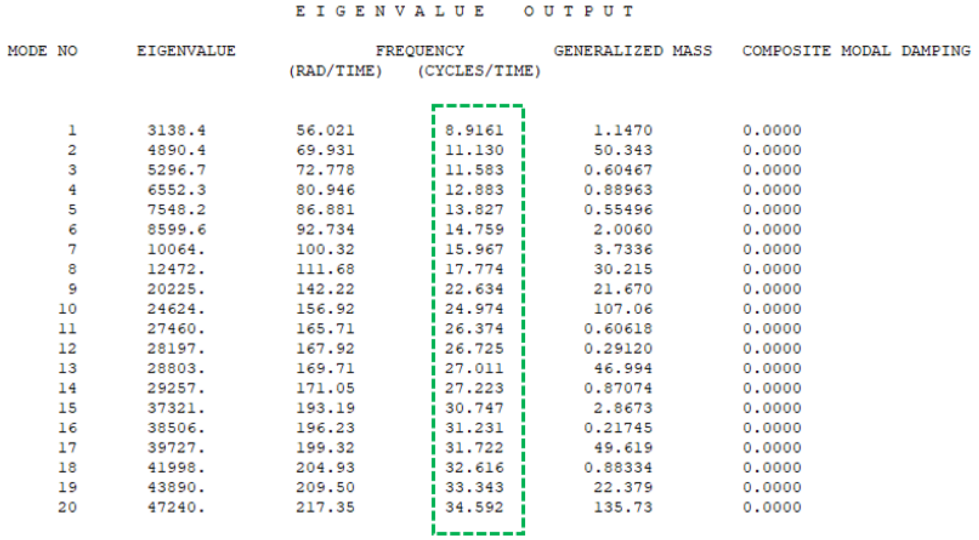


Fig. 5 Model for random vibration analysis

A linear perturbation procedure that gives the linearized dynamic response of the frame to user-defined random excitation and use the set of modes extracted in a previous eigenfrequency extraction step to calculate the power spectral densities (PSDs) of response root mean square (RMS) values of stresses, strain, displacements [16].

The random response analysis predicts the response of the frame that is subject to a nondeterministic continuous excitation that is expressed in a statistical sense by a cross-spectral density matrix. Since the loading is nondeterministic, it can be characterized only in a statistical sense, in Abaqus/Standard the excitation is defined as stationary and ergodic. In Fig. 6, the first 20 mode frequencies are calculated and listed. The fourth column is the mode frequencies of the model.



```
                        E I G E N V A L U E    O U T P U T

 MODE NO      EIGENVALUE              FREQUENCY         GENERALIZED MASS   COMPOSITE MODAL DAMPING
                                (RAD/TIME)   (CYCLES/TIME)


     1        3138.4           56.021        8.9161           1.1470          0.0000
     2        4890.4           69.931        11.130           50.343          0.0000
     3        5296.7           72.778        11.583          0.60467          0.0000
     4        6552.3           80.946        12.883          0.88963          0.0000
     5        7548.2           86.881        13.827          0.55496          0.0000
     6        8599.6           92.734        14.759           2.0060          0.0000
     7        10064.           100.32        15.967           3.7336          0.0000
     8        12472.           111.68        17.774           30.215          0.0000
     9        20225.           142.22        22.634           21.670          0.0000
    10        24624.           156.92        24.974           107.06          0.0000
    11        27460.           165.71        26.374          0.60618          0.0000
    12        28197.           167.92        26.725          0.29120          0.0000
    13        28803.           169.71        27.011           46.994          0.0000
    14        29257.           171.05        27.223          0.87074          0.0000
    15        37321.           193.19        30.747           2.8673          0.0000
    16        38506.           196.23        31.231          0.21745          0.0000
    17        39727.           199.32        31.722           49.619          0.0000
    18        41998.           204.93        32.616          0.88334          0.0000
    19        43890.           209.50        33.343           22.379          0.0000
    20        47240.           217.35        34.592           135.73          0.0000
```

Fig. 6 Modal frequencies

Fig. 7 shows the frequency response of displacement. According to the curve, at about 25 Hz, the system shows the max displacement. According to Fig. 6, it is the tenth order resonant frequency.

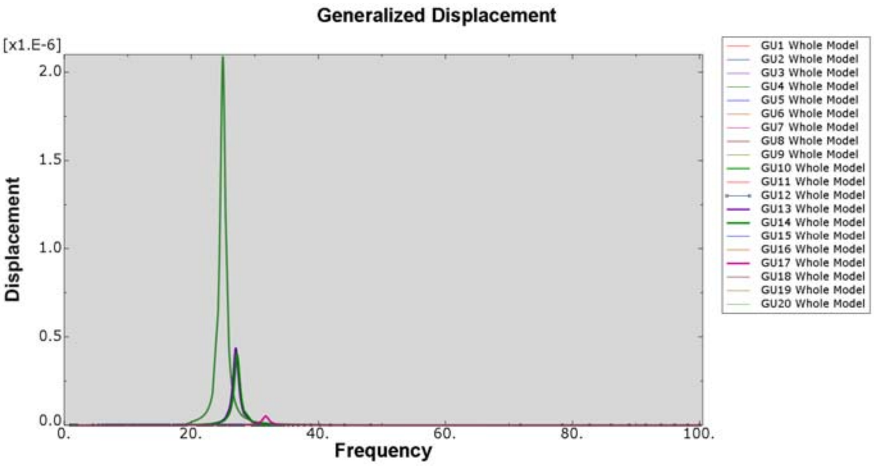


Fig. 7 Frequency response function of displacement

Fig. 8 shows the power spectrum density load chosen by the user for the model.

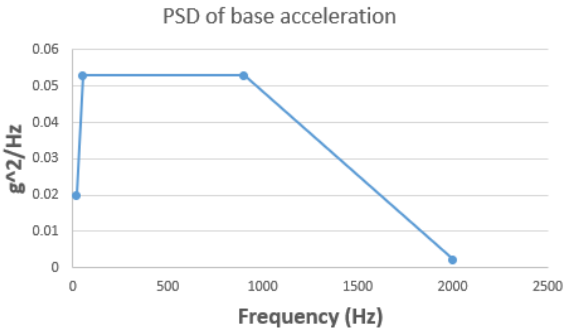


Fig. 8 PSD Load

After completing the random vibration analysis of the model, the RMS contour plot is shown in Fig. 9. It is observed that "hot spots", where the stress levels are higher than other areas and showing red color in the contour plot, are located at the joint locations.

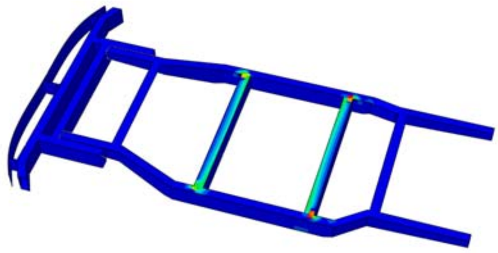


Fig. 9 RMS of Mises stress contour plot

All curves of RMS of Mises stress based on Segalman Rule for the frame are plotted out, the normal RMS stress value is 54.3 MPa, as shown in Fig. 10.

In Fig. 11 are S-N curves of two aluminums and two steels. To demonstrate the fatigue life calculation process, aluminum 2024-T3 is assigned as the frame's material for simulation, and its S-N curve is used for fatigue calculation.

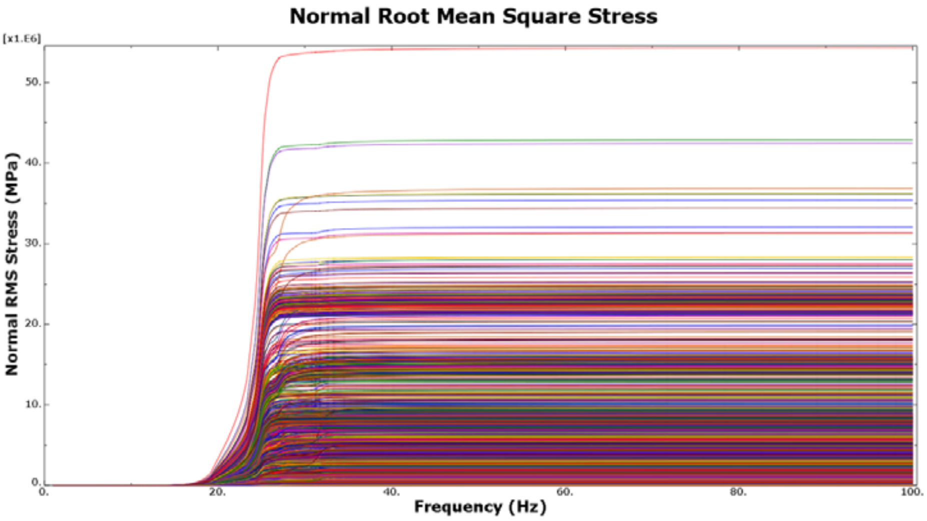


Fig. 10 RMS of frame of the system

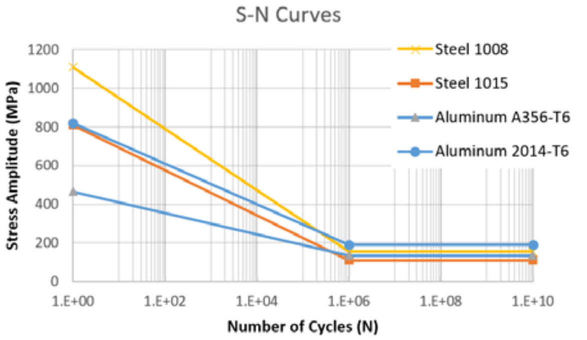


Fig. 11 S-N curve of aluminums and steels

The number of cycles value, $\boldsymbol{N_i}$, is obtained by interpolation according to the RMS of Mises stress value. The number of cycles $\boldsymbol{n_i}$ for the dominant stress is calculated by (13):

$$\boldsymbol{n_i} = T * dominnate\ frequence * Occurrence \qquad (13)$$

where dominant frequency is 25 Hz when the frame's material is aluminum, as shown in Fig. 7. T is the duration time, in this analysis is 20 hours. According to the Gaussian distribution, 1 σ values occurs 68.3% of the time; 2 σ values occurs 27.1% of the time; 3 σ values occurs 4.3% of the time. All values for the fatigue life calculation are listed in Table X.

TABLE X
PARAMETERS TO CALCULATE THE FATIGUE LIFE

| Std | Normal RMS Stress (MPa) | Frequency (Hz) | Values occurrence (%) | $\boldsymbol{N_i}$ (cycles) | $\boldsymbol{n_i}$ (cycles) |
|---|---|---|---|---|---|
| 1 σ | 49.5 | 25 | 68.3 | + ∞ | 1e+6 |
| 2 σ | 99 | 25 | 27.1 | + ∞ | 4.88e+5 |
| 3 σ | 148.5 | 25 | 4.3 | 9.58e+5 | 7.74e+4 |

The damage value is calculated by (14):

$$D = \sum_1^3 D_i = \sum_1^3 \frac{n_\mathrm{i}}{N_\mathrm{i}} = 0 + 0 + 8.08e^{-2} = 0.08 \qquad (14)$$

When assigning steel as the frame's material, the dominant frequency was changed to 38.5 Hz and the normal RMS stress was changed to 70.8 MPa.

Following the same procedure, assigning different materials to the frame in the model, the fatigue life values of the frame with different materials are calculated, and results are listed in Table XI.

TABLE XI
FATIGUE LIFE OF THE FRAME WITH DIFFERENT MATERIAL

| Materials | Aluminum 2014-T6 | Aluminum 2024-T3 | Steel 1015 | Steel 1008 |
|---|---|---|---|---|
| Fatigue Limit (MPa) | 189.7 | 134.4 | 109.9 | 153.8 |
| D values | 0 | 0.09 | 0.93 | 0.13 |

Among these four material alternatives, aluminum 2014-T6 shows the minimum damage value. This means when the frame's material is aluminum 2014-T6, 20 hours Z-direction PSD load causes no damage to the frame, and the fatigue life of the frame is infinity. Providing the simulation results, an engineer could choose the material for the chassis to meet different design targets.

The program can be further expanded to compatible with other fatigue software and adding more analysis steps.

## VI. Conclusion

In this paper, a Python-based program is developed to reduce repeated manual operations and increase work efficiency for vibration and fatigue analysis procedures. Consultative AHP procedure is applied during the software development process to identify critical user requirements and design characteristics. TOPSIS is applied to choose the optimal alternative, which is more appealing to users.

## References

[1] Joachim Karlsson, An evaluation of methods for prioritizing software requirements. Claes Wohlin, Björn Regnell, Information and Software Technology 39 (1998) 939-947.

[2] Jorge Melegati, Alfredo Goldman, Fabio Kon, Xiaofeng Wang, A model of requirements engineering in software startups. Information and Software Technology, 109 (2019) 92–107.

[3] Mohd. Sadiq, Jawed Ahmed, Mohammad Asim, Aslam Qureshi, R. Suman, More on Elicitation of Software Requirements and Prioritization using AHP. 2010 International Conference on Data Storage and Data Engineering.

[4] W. Scacchi, Understanding the requirements for developing open source software systems. IEE Proceedings—Software, Paper number 29840, Accepted for publication with revisions, December 2001.

[5] Ming-Chyuan Lin, Chen-Cheng Wang, Ming-Shi Chen, C. Alec Chang. Using AHP and TOPSIS approaches in customer-driven product design process. Computers in Industry 59 (2008) 17–31.

[6] A Rama Mohan Reddy, Prof. M M Naidu, Prof. P. Govindarajulu. An Integrated approach of Analytical Hierarchy Process Model and Goal Model (AHP-GP Model) for Selection of Software Architecture. IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.10, October 2007.

[7] Alexander Benlian, Is traditional, open-source, or on-demand first choice? Developing an AHP-based framework for the comparison of different software models in office suites selection. European Journal of Information Systems (2011) 20, 542–559.

[8] Alka Agrawal, Mamdouh Alenezi, Rajeev Kumar, Raees Ahmad Khan. Measuring the Sustainable-Security of Web Applications through a Fuzzy-Based Integrated Approach of AHP and TOPSIS. Received September 21, 2019, accepted October 6, 2019, date of publication October 11, 2019, date of current version November 1, 2019.

[9] Andrew Halfpenny. A frequency domain approach for fatigue life estimation from Finite Element Analysis. International Conference on Damage Assessment of Structures (DAMAS 99) Dublin.

[10] Da Yu, Abdullah Al-Yafawi, Tung T. Nguyen, Seungbae Park, Soonwan Chung, High-cycle fatigue life prediction for Pb-free BGA under random vibration loading. Microelectronics Reliability 51 (2011) 649–656.

[11] Da Yu, Abdullah Al-Yafawi, Seungbae Park, Soonwan Chung, Finite Element Based Fatigue Life Prediction for Electronic Components under Random Vibration Loading. 2010 Electronic Components and Technology Conference.

[12] Y.S. Chen, C.S. Wang, Y.J. Yang, Combining vibration test with finite element analysis for the fatigue life estimation of PBGA components. Microelectronics Reliability 48 (2008) 638–644.

[13] Abdullah Al-Yafawi, Saket Patil, Da Yu, Seungbae Park, James Pitarresi, Random vibration test for electronic assemblies fatigue life estimation. 2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems.

[14] Abdullah Al-Yafawi, Da Yu , Seungbae Park, James Pitarresi, Soonwan Chung, Reliability Assessment of Electronic Components under Random Vibration Loading. 2009 Electronic Components and Technology Conference.

[15] Baussaron Julien, Fouchez Bertrand, Yalamas Thierry, Probabilistic Random Vibration Fatigue. Procedia Engineering 66 (2013) 522 – 529. 5th Fatigue Design Conference, Fatigue Design 2013.

[16] https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus-/docs-/v6.6/books/usb/default.htm?startat=pt03ch06s03at15.html