PAPER • OPEN ACCESS

Process automation of simulation using Toolkit/Tool Command Language (TK/TCL) scripting

View the article online for updates and enhancements.

Process automation of simulation using Toolkit/Tool Command Language (TK/TCL) scripting

R Bala Anand[1] and Pankaj Kumar[1*]

[1]Department of Mechanical Engineering, SRM Institute of Science and Technology, India.
*Corresponding Author: pankajkumar.r@ktr.srmuniv.ac.in

Abstract: In the advancement of the technology, the validation of prototype can be in the virtual environment. It is a forerunner to the physical validation. Through virtual-validation, many parameter changes can be made. In current practice of development process, designing and validation with standards are vital primitive steps. Since the beginning, the designs are validated using physical prototyping, which consumes lot of time and energy. Virtual validation reduces the lead-time for the manufacturing processes. The process of virtual validation involves three steps, (i) Pre-processing (importing the models, meshing, applying suitable boundary conditions), (ii) Solving (analysis types and respective solvers). (iii) Post processing (generating the results and documenting). But, even when using computer software can lead to time and energy consumption due to some of the repeated tasks. When these tasks can be automated using computer programs, further time and energy can be saved. A set of computer programs are written to automate the modal analysis simulation process in ALTAIR HYPERMESH 14.0 software using Toolkit/Tool command Language (TK/TCL) scripting. It is found from the present simulation that, it is helpful in many ways like, persons who are new to the HYPERMESH software can perform the pre-processing setup without going much deep into learning the background process. This will also make the user get used to the software at much faster rate and the simulation set up process is made easier for beginners. Secondly, human Errors can be avoided due to automation work by reducing the number of manual steps that has to be done. If the number of manual operation is decreased, automatically the chances to do an error is reduced. This is a great advantage in the industry as it saves a lot of work force to find and rectify the errors. Apart from this, time consumption for the creation of deck file is reduced since most of the operations are automated. This can be accounted in terms of a number of button clicks. By scripting, the number of button clicks is reduced, which is directly related to the time consumed for the file creation process. Hence, the process automation is one of the developing field and has a promising future with many advantages both personally and commercially.
Keywords: Automation; Scripting; Toolbox; Computer aided engineering; TK/TCL; Modal analysis.

1. Introduction
A product which is under development has many steps in its life cycle. The first step is collecting data by feedback, brainstorming, and other techniques. Then with the advent of computer and technology, the Computer Aided Design (CAD) models are developed. Prior to the manufacturing, the design validation step has been traditionally done through physical validation. This involves manufacturing a

set of sample prototypes and testing them under the desired working conditions. Once the test is done the results are taken and compared with each other. Based on the engineering knowledge of the tester and the results obtained, the decision is made whether the design is valid or not. But this process consumes a lot of time and energy which leads to the high cost. Taking this in to consideration and to make use of the computers, virtual validations are done. Virtual validation follows the procedure of creating the Three-Dimensional (3D) model of the component using any CAD software. Once the 3D model is done, the model data is imported into any analysis software (Computer Aided Engineering (CAE)). Then the required set up is done which might include various analyses like structural, thermal, dynamics etc. Then the results are obtained which is in the form of soft data which are used for validating the designs. Again, if we take the virtual validation part, we need to use some CAE software which has a standard set of inbuilt functions and macros running in it. We can make use of those macros to perform our analysis. Sometimes this software can be in very much generalized form i.e. the software will have all the options and steps to use, in which some of them may be a repeated one. While performing these steps repeatedly, once again there is wastage of time and energy. These repeated tasks are automated by using the scripting language.

The process of virtual validation involves (i) Preprocessing (importing the models, meshing, applying suitable boundary conditions), (ii) Solving (analysis types and respective solvers), (iii) Post processing (generating the results and documenting). The preprocessing stage involves, (i) Importing the models (collecting CAD data from CAD packages and other source) (ii) Meshing (Discretization of the CAD model into Finite Element (FE) model data) (iii) Applying suitable boundary conditions (The actual loading conditions in the real world are applied as a virtual load to the FE model). The solving stage involves selecting the analysis types and its respective solvers. Various types of solvers like Optistruct, RADIOSS, etc. are available which serve their own type of purposes. The post processing stage involves generating the results (Once the model is solved, various results based on the requirements are generated) and Documentation (Obtained results are documented for the future use). The software package chosen for analysis is ALTAIR HYPERMESH 14.0, which supports scripting with the help of TK/TCL. Hence scripts are written for automating the process of simulation. The simulation process is the modal analysis of a simple simply supported beam with the circular cross-section. This analysis deck file creation procedure is automated using scripts.

2. Literature Survey

Rao and Krishna [1] created a macro to assign the material data and property data to the components for large number of components and achieved to reduce the time factor from several hours to seconds. Mukherjee et al. [2] automated the strain gauges location identification process in CAE. The manual check and creation of strain gauge report are replaced by execution of scripts which will compute and provide the required data in hours compared to a week's job, eliminating the manual error. Sonavane et al. [3] did the automation of post-processing using TCL scripts. The method of Post-Processing discussed is highly accurate and speedy as compared to any other existing manual method. This script eliminates human intervention and thus increasing the accuracy of result interpretation. One CAE engineer can save up to 360 hours by using this script in one year. Chaudhari et al. [4] did automation tool for "Fatigue life estimation of chassis frame and biw" using Altair Process Manager. The total time saving achieved by Process Automation (PA) tool is 85 percentage. Defining a structured process for the fatigue analysis was a challenge as standardization of processes was the key requirement for automation. Pathak et al. [5] did automation tool for "Strength analysis of chassis frame" using Altair Process Manager.

Topalkatti et al. [6] did automation to eliminate penetrations caused by zero set concepts in Ls-DYNA. The main challenge in this project is to select the required free edges of the surface, which was done easily using HyperMesh Scale command. These macros trim all the free edges of surface and this script can be modified in such a way that only selected edges can be split. Ghosh and Sengupta [7] automated the complete simulation process of a blast furnace. They found that the main challenge is the TK/TCL language which is not an object-oriented language. Programmers with big projects mostly

prefer for object-oriented languages like Java, C++. More recently multi-paradigm, functional languages like Python and JavaScript are also preferred. Even after development, big projects using other families of languages are difficult to maintain, extend and test. Moreover, Script View, the Integrated Development Environment (IDE) they were using, was not mature like Visual studio or eclipse for debugging of TK/TCL Code.

Jha [8] did automation of engineering process for Noise, Vibration and Harshness (NVH) model preparation. The key challenges faced during the setup of the methodology is, TK/TCL is a new programming language, is difficult to understand. Mane et al. [9] did process automation of passenger vehicle body structural torsion stiffness optimization. The use of PA Tool in CAE field can reduce the vehicle development time, weight and cost (man-hour's reduction) significantly. The developed PA Tool is implemented as standard practice for torsion stiffness optimization analysis across CAE team of the organization. Gole et al. [10] worked on auto spot weld FE connection using TCL script for process automation. The script greatly depends upon the geometry of supplied spots. Any discrepancy between the spot data and the component geometry might lead to inaccurate spot creation. The future plan is to make a comprehensive tool for connections by extending similar approach for creation of other connections like adhesive, seam weld and bolt joints.

Modal analysis has become a vital means of finding the modes of vibration of a structure. The discrete dynamic equation used in Finite Element Analysis (FEA) is the following (1),

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [k]\{u\} = \{F\} \qquad (1)$$

Where [M] is the mass matrix, [C] is the damping matrix and [K] the stiffness matrix. All three matrices are constant in linear dynamics. The damping is neglected as it is nearly equal to zero. Solving this we get ([K]- ω^2[M]) {U} = {0}. In which ω^2 is the eigenvalue and {U} the eigenvector. Then we get (2),

$$\omega = \left(\frac{K}{M}\right)^{1/2} \qquad (2)$$

TCL is a general purpose multi-paradigm programming language. It is a scripting language which aims at providing the path or ability for applications to communicate with each other. TCL can be written once and can run anywhere. It runs on Windows OS, Mac OS X, Disk Operating System (DOS), and almost on all UNIX platform. [11]

TK refers to Toolkit which provides a cross platform GUI widgets and helps in building a Graphical User Interface. It was developed to extend to TCL scripting language for the GUI purpose by John Ousterhout. TK remained in development independently from TCL. It is available as open source. It is customizable and configurable to a good extent. It provides many widgets which can be used in the interactive programs. GUI looks identical across all platforms [11].

HyperWorks is a comprehensive and open architecture CAE simulation tool that is widely used in the industry. It offers great technologies for designing and optimizing high performance and innovative products. HyperWorks includes tools like modeler, linear and nonlinear analyzing tool, structural and system-level optimization tool, fluid dynamics simulation tool, multi-physics analysis tool, and data management solutions. HyperMesh is a finite element pre-processor and post-processor for FE solvers, which allows engineers to perform CAE tasks in an interactive and visual environment. HyperMesh has an easy to learn GUI and it can directly import CAD geometry and existing finite element models, thus it can provide robust interoperability and high efficiency. Advanced automation tools which are packaged within HyperMesh allow its users to optimize the mesh from a set of quality criteria, change the existing meshes through morphing and other techniques, and generate mid-surfaces from thin models which may even have varying thickness. It is one of widely used pre-processor in the industries today [12].

So, by this literature review, it is found that there is a huge focus on automation of simulation process to reduce the lead time. This can be used to save a huge amount of time and energy. So, this topic has been chosen as the area of improvement. Here the scripts are written, which will be used to automate the simulation process of modal analysis. Taking it to a further level, USER PAGE, which is a GUI page is formed. This USERPAGE is placed on the home screen. This can be used for making

the simulation process easier by placing all the buttons in a linear fashion. This is used by person who are new to HyperMesh for performing the modal analysis without the need to understand the background process of the simulation process. As said previously, the TK/TCL scripts are used for the process.

3. Modal Analysis

The procedure to perform a modal analysis in HYPERMESH is explained through figure 1.
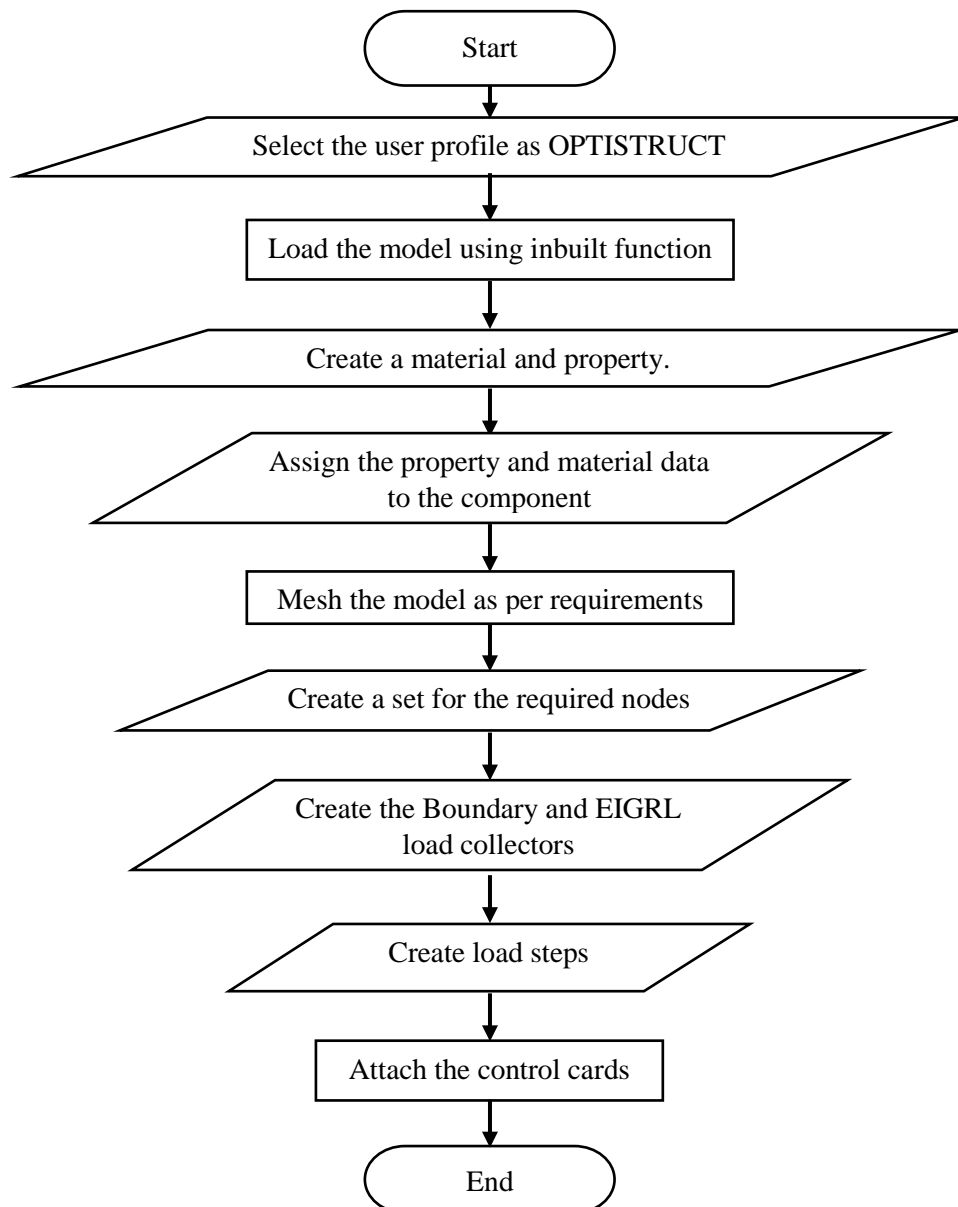
```
                    ┌─────────────┐
                    │    Start    │
                    └──────┬──────┘
                           ↓
          ╱──────────────────────────────────╲
          │  Select the user profile as OPTISTRUCT  │
          ╲──────────────────────────────────╱
                           ↓
              ┌─────────────────────────────┐
              │ Load the model using inbuilt function │
              └─────────────────────────────┘
                           ↓
          ╱──────────────────────────────────╲
          │      Create a material and property.    │
          ╲──────────────────────────────────╱
                           ↓
          ╱──────────────────────────────────╲
          │   Assign the property and material data  │
          │           to the component              │
          ╲──────────────────────────────────╱
                           ↓
              ┌─────────────────────────────┐
              │  Mesh the model as per requirements │
              └─────────────────────────────┘
                           ↓
          ╱──────────────────────────────────╲
          │     Create a set for the required nodes │
          ╲──────────────────────────────────╱
                           ↓
          ╱──────────────────────────────────╲
          │    Create the Boundary and EIGRL       │
          │           load collectors              │
          ╲──────────────────────────────────╱
                           ↓
          ╱──────────────────────────────────╲
          │          Create load steps             │
          ╲──────────────────────────────────╱
                           ↓
              ┌─────────────────────────────┐
              │      Attach the control cards       │
              └─────────────────────────────┘
                           ↓
                    ┌─────────────┐
                    │     End     │
                    └─────────────┘
```

Figure 1. Modal Analysis procedure.

3.1. Modelling

The CAD model is created using CATIA V5 which is one of the advanced CAD modeler in the industry. The dimensions are chosen as an arbitrary value as the focus is on the automation of the process irrespective of the dimension. Dimensions of the circular cross section beam are as follows, which is shown in the figure 2. The length of the beam = 100 mm and diameter of circular section = 25 mm.
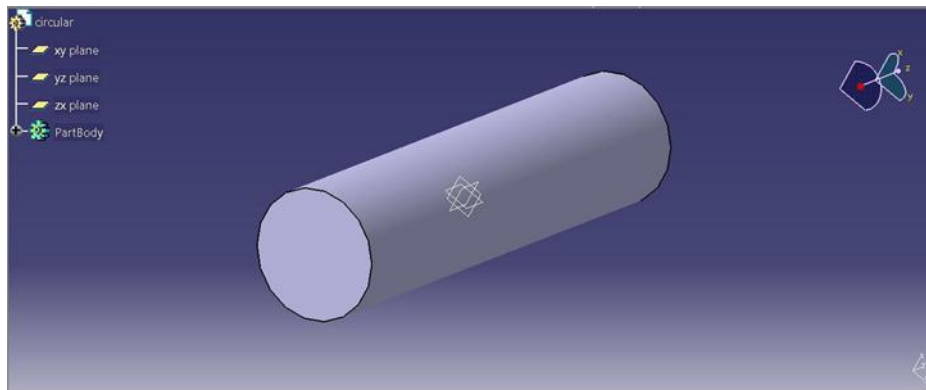

Figure 2. CATIA model.

3.2. Exporting

The CAD model is natively saved with **.catpart* file format extension. HyperMesh allows its user to import some of the CAD data in its native file format. In this case, HyperMesh allows the CATIA V5 part file data to be imported directly. Hence the CAD data is directly imported into the Hyper Mesh (HM) environment.

3.3. Scripting

The scripts are written for each of the steps that are involved in the model analysis deck file creation procedure. The process is explained step by step.

3.3.1. User Profile. User profiles are solver interfaces in which the pre-processed model is going to be solved. Here the OPTISTRUCT user profile is selected. It stands for (Structural + Optimization). The script is written to select the OPTISTRUCT user profile as in the figure 3. This is done by including two files. First a template file named Optistruct and a macro file named opti.mac.

3.3.2. Importing Geometry. The model geometry is imported in to the software using the import options available. Since the importing GUI is quite advanced and serves its purpose well, the default GUI is used for importing the geometric model.

3.3.3. Creating Material. In this step the script is written to create the material property. It is explained using the figure 4. The material property is created by providing a GUI dialog box. The dialog box is as shown in the figure 5.

The inputs asked are the material name, material id, Young's modulus, Poisson's ratio and density. By getting these values, the material property is created by the script which is seen in the model browser as shown in the figure 6.
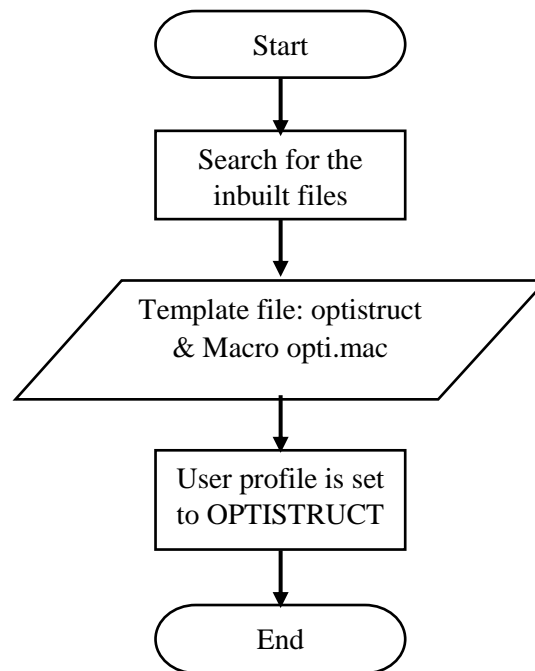
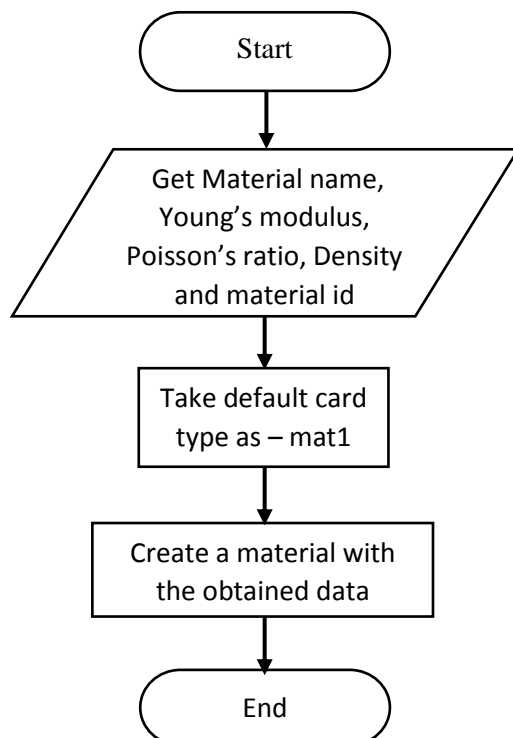**Figure 3.** OPTISTRUCT user profile script flow process.

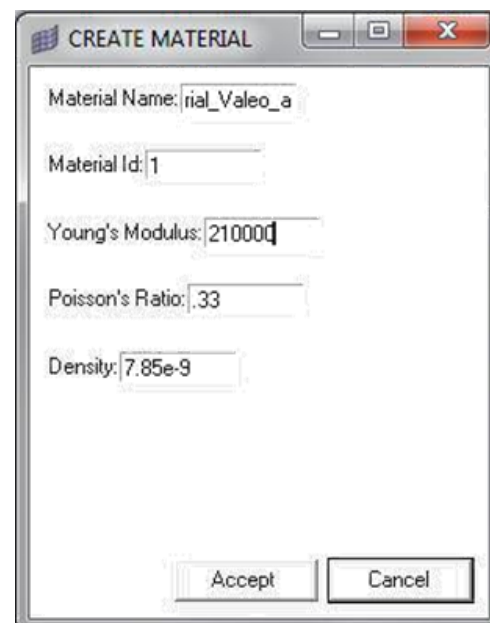**Figure 4.** Material creation script flow process.   **Figure 5.** Material creation dialog box.
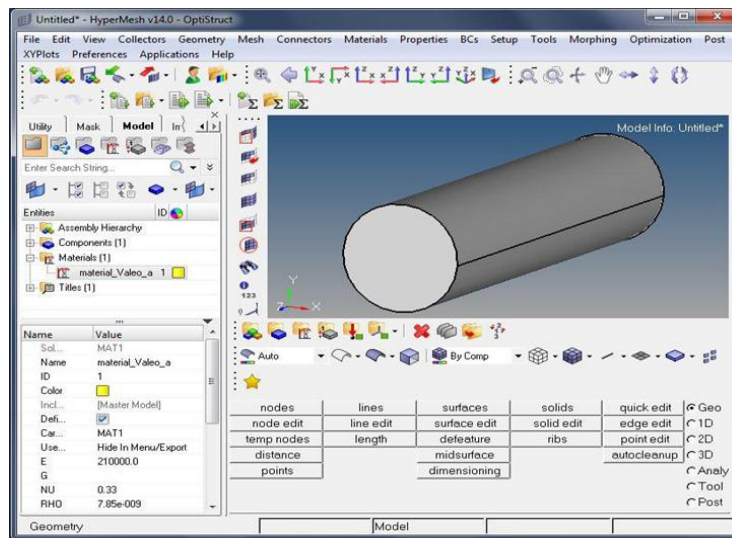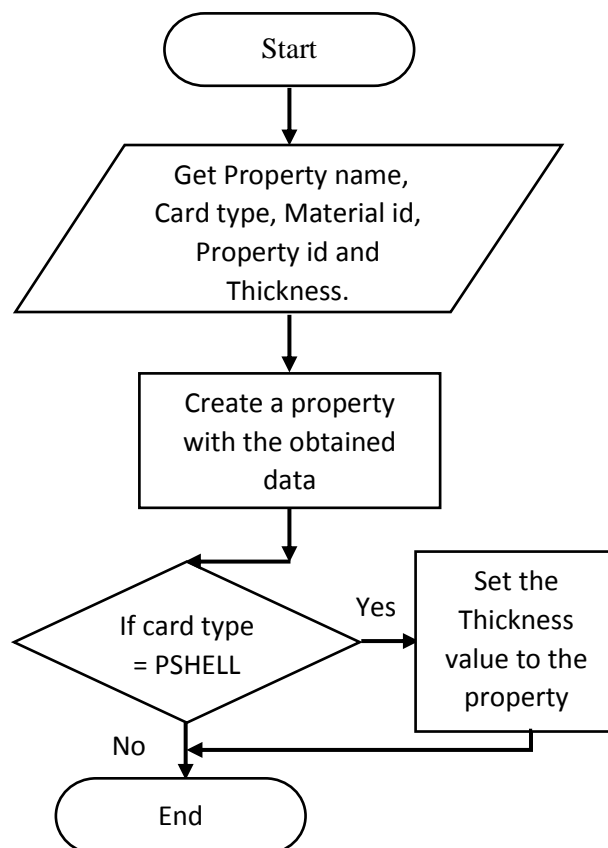
Figure 6. Material created.

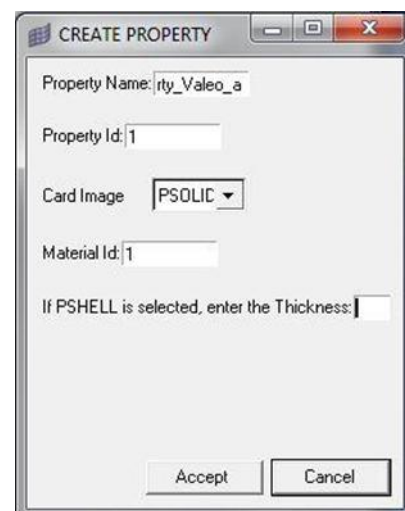**Figure 7.** Property creation script flow process.

Figure 8. Property creation dialog box.

3.3.4. Creating Property. The selected user profile is designed by the software vendor in such a way that the material data are assigned to a property first and then the property is then assigned to the

component. In this step the script is written to create the property entity. It is explained using the figure 7. The property is created by providing a GUI dialog box. The dialog box is as shown in the figure 8.

The inputs asked are the property name, property id, its corresponding material id, the card image. There are two cards imaged used, PSOLID and PSHELL. PSOLID card image is used for components which are in the form of solids, whereas PSHELL card image is used for components which are in the form of surfaces. If PSHELL card image is selected due to surface type components, its thickness must be defined. By getting these values, the property is created.

3.3.5. Assigning Property Data To Components. In this step the script is written to assign the property data to the selected component. It gets the component id and the property id to set the values.

3.3.6. Meshing. Meshing is discretization of the geometric model in to required number of elements. The HyperMesh has many options for its user for effectively meshing the model. The given model is meshed by the user as per the requirement. As meshing needs manual supervision, it is left with the user to carry-on. The meshed model is as shown in the figure 9.
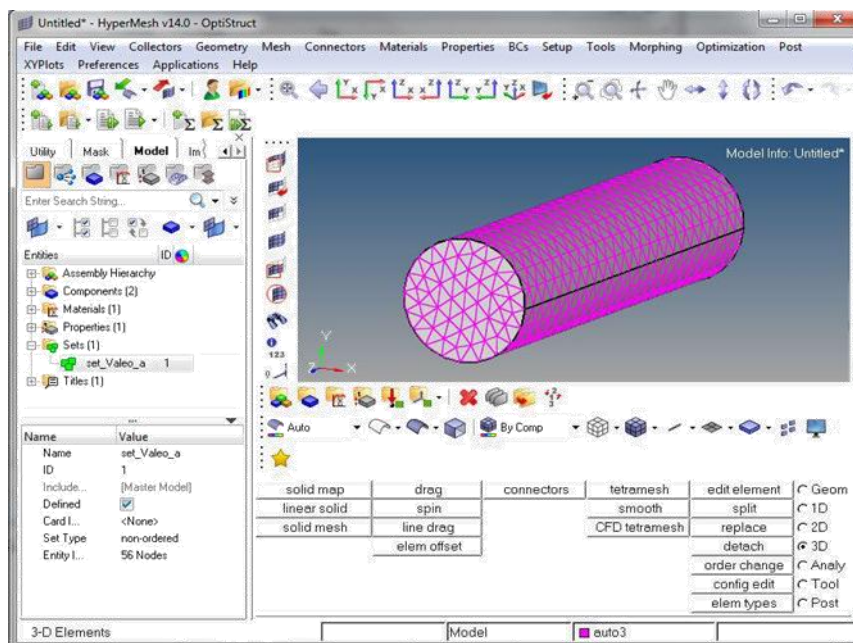

Figure 9. Meshed Model.

3.3.7. Creation of Sets. Sets are a group of entities. In this step the script is written to create a set of selected entity "nodes".

3.3.8. Load Collector (Boundary). A load collector is used to assign a set of constraints or equation to the model. In the modal analysis of the simple beam the nodes on one of the faces of the model is constrained with all Degrees Of Freedom (DOF) fixed. This is done by making a set of selected nodes which is explained in the previous section. The load collector for the boundary condition is created by providing a GUI dialog box. The dialog box is as shown in the figure 10.

The inputs asked are the required DOF. By getting these values, it prompts the user to select the desired set which is already created to assign the boundary loading conditions to it. Once the loading condition constraints are assigned to the set, it is seen in the model browser.
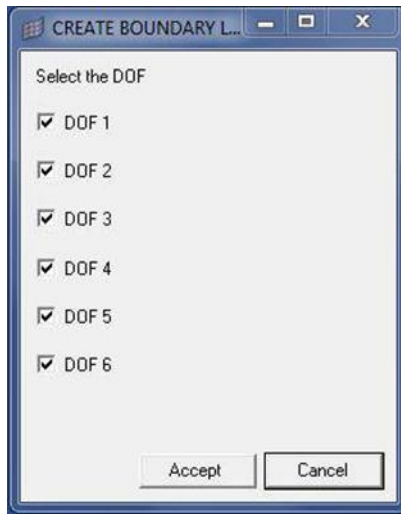
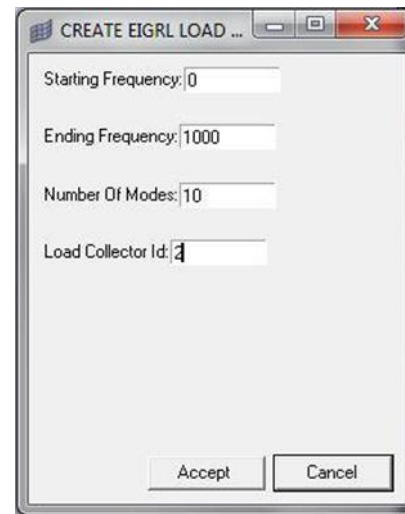**Figure 10.** Boundary load collector creation dialog box.

Figure 11. EIGRL load collector creation dialog box.

3.3.9. Load Collector (EIGRL). This load collector is used to setup the analysis. It specifies the number of mode shapes to be extracted and the frequency range in which it is to be extracted. The card type chosen here is "EIGRL" which is for modal analysis. It provides suitable section to input the required values for the analysis. In this step the script is written to form EIGRL load collector. The EIGRL load collector is created by providing a GUI dialog box. The dialog box is as shown in the figure 11.

The inputs asked are the starting and ending frequency for which the mode shapes must be extracted and the number of mode shapes to be extracted. By getting these values, the load collector is created. Once the values are assigned to the load collector, it is seen in the model browser as shown in the figure 12.
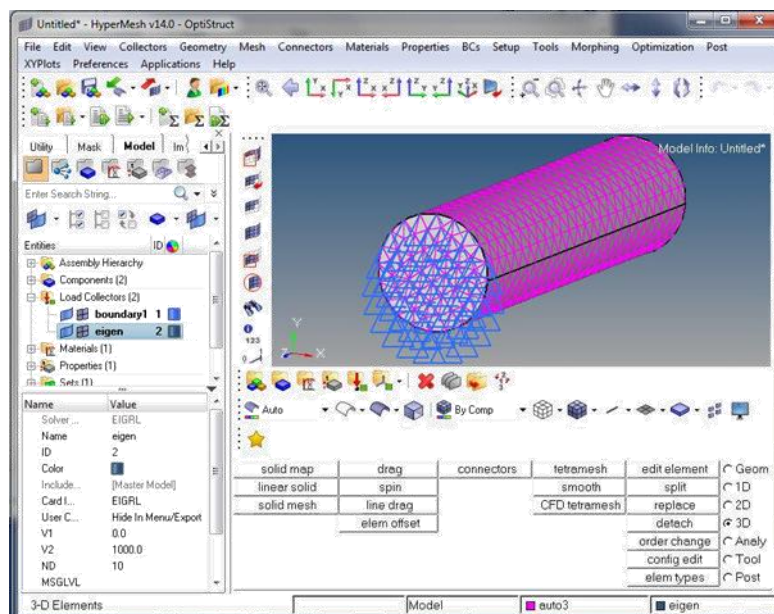

Figure 12. EIGRL load collector created.

3.3.10. Load Step. The load step is used to group the required number of load collector and form an analysis loading setup. Since we are performing a modal analysis, the analysis type is set to "NORMAL MODES" by the script itself. The boundary and the EIGRL load collector which are created already are assigned to "SPC" and "Method (Struct)" options. These actions are also performed automatically by the script. Once the load step is created and the load collectors are assigned to the load step, it is seen in the model browser as shown in the figure 13.
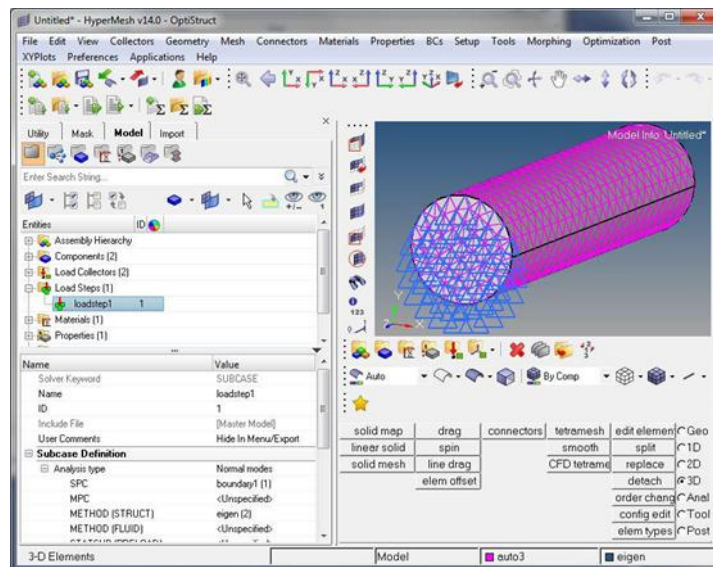

Figure 13. Load step created.

3.3.11. Control Cards. Control card entities are used to create solver control cards such as results file In-put/Output options, Central Processing Unit (CPU) and memory limits, and other inputs that are required to complete the analysis. The three control cards used here are SCREEN, PARAM and FORMAT. The FORMAT control card is used to control the type of output that the analysis setup must generate. Here the selected types of output format are "HM" and "H3D". The HM is used as a standard HyperMesh file format and the H3D is a binary type result file. The SCREEN control card is set to "OUT" value which is used to control the display of the solver data while it is running. The PARAM control card is selected. In that the "CHECKEL" option is set to "NO". It makes the solver to skip from checking the element quality before the run starts. In this step the script is written to attach the control cards to the deck file. It is explained using the figure 14.

All the mentioned the cards are created automatically using scripting and user do not need to perform any action for that. Once the control cards are attached, it is seen in the model browser.

3.3.12. User Page. The User page is a macro that is used to create buttons for the TK/TCL scripts files. These buttons are placed in the "USER" page in the browser section. It is placed in a linear fashion, such that the user can perform the modal analysis by pressing the buttons one after the other. This makes it easier for the user to perform the modal analysis and reduces the time consumption and errors in performance. Once the User Page macro is created, it is seen in the "USER" browser as shown in the figure 15.
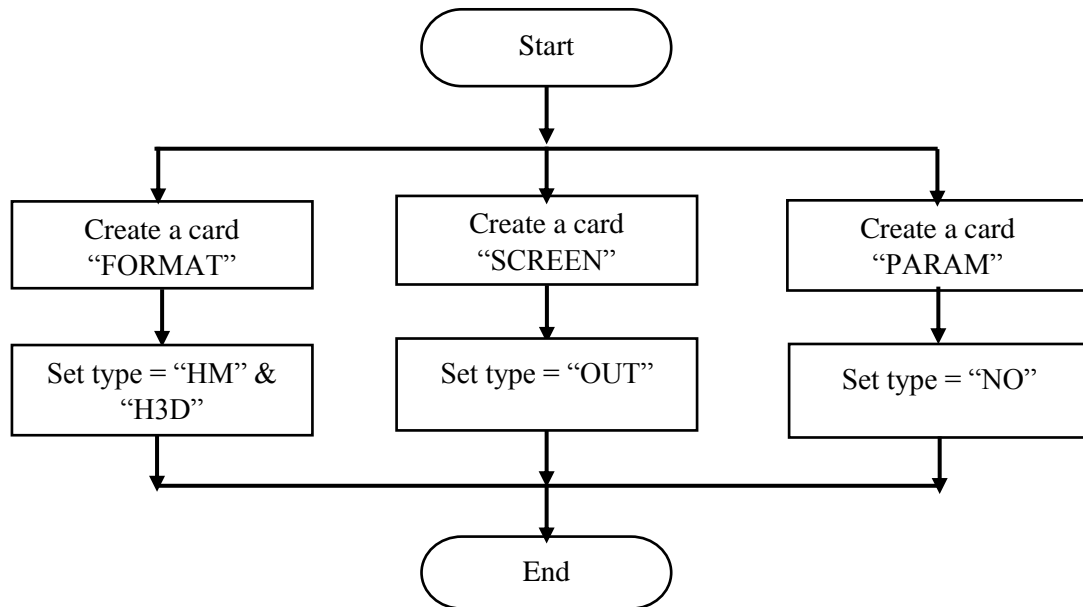
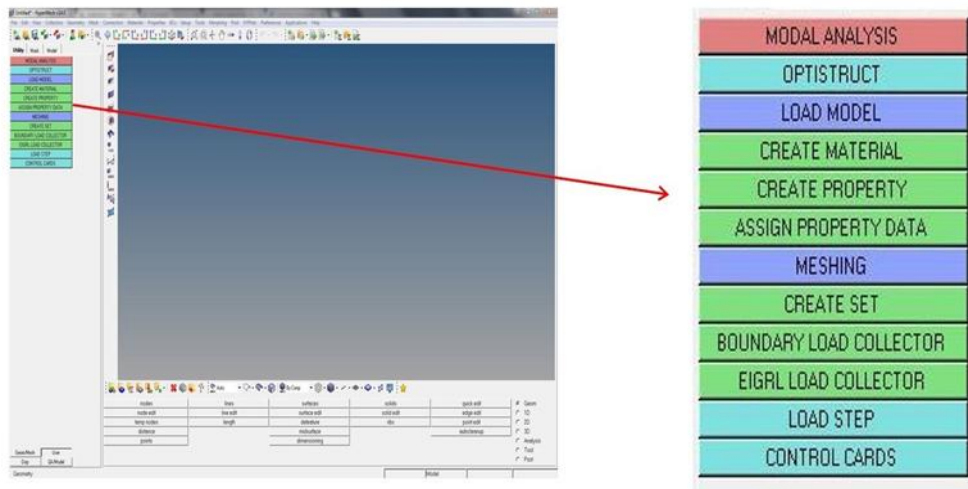**Figure 14.** Control card creation script flow process.
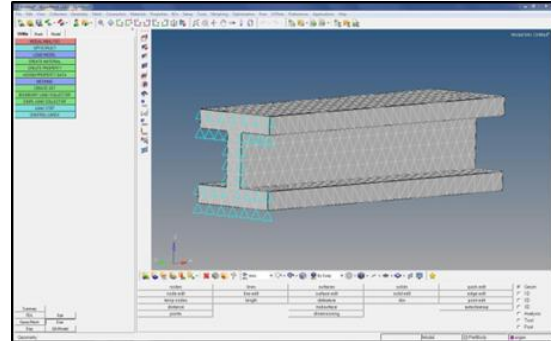

Figure 15. USER page.

4. Results and Discussion

The code is tested with few simple models to check the integrity. The simulation deck setup is made for these simple models. The models selected for the testing purpose are a simple beam with square cross section and a "I" section. These models are also made using CATIA V5. Dimensions of the square cross section beam are, length of the beam = 100 mm and side of the square of the beam = 25 mm. Dimensions of the "I" section beam are, length of the beam = 100 mm, side of the of the beam = 25 mm, thickness of the flange = 10 mm. Then for these models, the modal analysis deck setup is made using the created scripts. The scripts ran successfully, and the deck setup is made for the models. Once the deck is created, they are seen as shown in the figure 16.

Hence it is found that the scripts are successful in creating modal analysis deck setup for any similar type of model that is given as input.
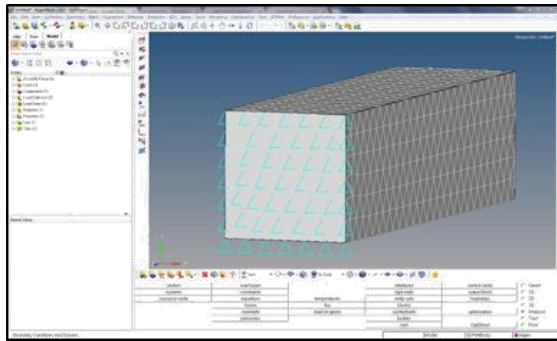
(a) Deck of square section beam model.   (b) Deck of "I" beam model.

Figure 16. Testing the codes with cad models.

5. Conclusion

The process automation of simulation using (TK/TCL) scripting is done to perform a modal analysis setup for a 3D CAD model. This will be helpful in ways such as, (i) a new person can also handle HYPERMESH software and perform the pre-processing setup. This will make the user friendly to work at much faster rate. (ii) Human errors are reduced due to automation work by reducing the number of manual steps that has to be done. This is a great advantage in the industry as it saves a lot of work force to find and rectify the errors. (iii) Time consumption for creation of deck file is reduced since most of the operations are automated.

Hence the process automation is one of the developing and has a promising future with lot of advantages both personally for the automation engineer and commercially for the organization.

6. References

[1]   Rao P and Krishna N V 2012 "Automation for material and property values in HyperMesh using TCL." *Altair India HTC 2012*.
[2]   Mukherjee A, Manikandan and Karthik 2017 "Strain gauge automation." *Altair Technology Conference.*
[3]   Sonavane P, Patel S, and Bhirud P 2017 "Automation of post-processing techniques for iterative design procedure using TCL script." *Altair Technology Conference.*
[4]   Chaudhari A, Hange A S, and Girase D 2017 "Automation tool for fatigue life estimation of chassis frame and biw, using Altair process manager." *Altair Technology Conference*.
[5]   Pathak V S, Patil J and Shah V 2017 "Automation tool for strength analysis of chassis frame, using Altair process manager." *Altair Technology conference*.
[6]   Topalkatti P, Buddhe H, and Patil D 2017 "Automation to eliminate penetrations caused by zero set concept in ls-dyna." *Altair Technology Conference.*
[7]   Ghosh D and Sengupta B 2014 "Design automation of blast furnace using TCL/TK (HyperMesh)." *Larsen & Toubro Technology Conclave -III.*
[8]   Mane R, Patil J, Chavan P, Patil S and Durgule P 2017 "Process automation of passenger vehicle body structure torsion stiffness optimization." *Altair Technology Conference.*
[9]   Gole V, Patel S J and Sahu S 2017 "Auto spot weld FE connection using TCL script for process automation." *Altair Technology conference*.
[10] Jha M 2017 "Engineering process automation for nvh model preparation." *Altair Technology Conference.*
[11] https://www.tcl.tk and https://www.tutorialspoint.com
[12] Altair Engineering, Inc 2013 *Altair Hyperworks Documentation 13.0.*