Integrated Framework for Automating the Structural Design Iteration

P. Mujumdar[a] and J.U. Maheswari[b]

[a,b]Department of Civil Engineering, Indian Institute of Technology Delhi, India
E-mail: purvamujumdar@gmail.com[a], umapaul@civil.iitd.ac.in[b]

ABSTRACT

Structural designers especially in India use STAAD (Structural Analysis And Design) software to execute the structural analysis and design. This analysis and design follow typical steps in sequence starting from model preparation by defining nodes and elements, material properties, support conditions, loads along their combinations, analyses, design options, etc. During this design process in STAAD, iterations generally occur either due to changes in structural loads and their combinations; or to obtain the optimal design section. In this entire design process, when any of the information exchange occurs along the cycles/loops it is termed as iteration. Automating these structural iterations can speed up the design evolution on STAAD. Till date, negligible steps have been taken towards automating these structural design iterations.

DSM (Design Structure Matrix) is identified as a promising tool for representing and modelling iterative loops. It is a matrix-based tool with the same list of entities along the rows and columns encapsulated with relationships along the off-diagonal cells. Re-sequencing the DSM rows/columns can identify the iterations as blocks. OpenSTAAD is the API (Application Programming Interface) functionality in STAAD which is proposed to automate the iterative structural design in STAAD.

The objective of the paper is to model and automate the structural design iterations using DSM and OpenSTAAD respectively. This proposed framework was applied to a case illustration of six-storey building. Preliminary results of the model are presented herewith.

Keywords -
Automation; DSM; Iteration; OpenSTAAD

1 Introduction

A typical design process of a construction project involves various participants from different design disciplines. Most often, these stakeholders interact with each other in an ad hoc manner that enforces a situation of ill-structured information exchanges. When these exchanges of information occur continuously in cycles/loops, it is referred as design iterations [6]. Iteration is unavoidable in any design project [2]. As the project evolves in complexity, iterations become more challenging.

In construction projects, structural design is perceived as the most crucial component of the overall design phase of a project. The other disciplines such as MEP (Mechanical, Electrical, and Plumbing), HVAC (Heating, Ventilation, and Air conditioning) and so on play contributory role to structural design [9]. In India, structural designers perform the analysis and design in STAAD (STructural Analysis And Design) [1] software by following several steps such as defining nodes and incidences, material properties, support conditions, loads and their combinations, analyses, design options, etc. Structural designers undergo mental and manual iterations several times during this design tenure. Some of these iterations are performed either due to changes in structural loads, combinations or to obtain most suitable and optimized design section.

Iterations emerging for optimizing member sections are repetitive and time-consuming since the designer needs to run the analysis and design in STAAD frequently after each update. In addition, design disciplines other than structure also influence these iterations by introducing changes in their input. These iterations if automated can benefit structural designers by saving a considerable amount of time and effort.

Thus, the objective of this paper is to automate the structural design iterations. To achieve this objective, a framework comprising of STAAD and DSM (Design Structure Matrix) was planned. Researchers had investigated and proved DSM as a potential matrix-based, compact tool for modelling iterations [3, 4, 8, 10]. OpenSTAAD was planned for automating the iterations. It is basically API (Application Programming Interface) functionality with VBA (Visual Basic for Applications) [5] editor to STAAD engines. The entire concept was modelled with Excel VBA. This framework was applied to a case illustration of six-storey building and the initial results were quite convincing.

Henceforth, the reminder of the paper is organized as follows: the next section discusses the structural design iteration in STAAD followed by modelling of those iterations using DSM. Then, automation of iterations using OpenSTAAD was elaborated along with a case illustration. The paper ends at last with results and discussions.

2 Structural design iterations

A structural design process is characterized by the iterative steps of problem understanding and synthesis where individual design solutions are tried and optimized. Iteration is a process that occurs during the entire planning and design phase of any project. During the planning stages of design, structural designers are responsible for selecting a configuration of material and members to resist a set of applied loads and they focus on optimizing the solutions within the design requirements boundary.

Figure 1 illustrates a generic structural design process that starts with identifying use of structure. As seen in the figure, once the criterion for design is met then the drawings and specifications prepared at the end of design process are sent for verification. The fabrication and erection follows the design verification. Alternately, if the design does not satisfy the criteria, member sections will be redesigned and these iterations are highlighted using dotted lines in the figure.

Having presented the prevailing structural design iteration in STAAD, the next section discusses the modelling of the iteration using DSM.

3 DSM iteration model

As stated earlier, researchers had investigated that DSM is potential methodology for modelling and evaluating design iterations [8]. This matrix-based tool has gained popularity in the last decade because of simplicity to represent and model any iteration. DSM is a square matrix with the same elements (elements signify teams, activities, components etc.) in the rows and columns. The basic DSM operations such as partitioning and tearing enable the user to evaluate the feasible sequence of execution (interested readers can refer to [7] as partitioning and tearing are not in the scope of this paper).

To develop a DSM, the typical structural design process illustrated in Figure 1 was considered. Table 1 represents the list of activities comprising the entire design. The activity DSM of this design is presented in Figure 2 (a). The 'X' marks in the off-diagonal cells represent the dependencies. The 'X' mark shown in circle represents that structural geometry is dependent on architectural drawings for its execution. After rearranging the activity DSM to determine the feasible sequence of execution, partitioned DSM along with the block formation is shown in Figure 2 (b). An iterative loop within the block implies that these activities are interdependent. As mentioned earlier, this iteration may be due to changes in loads/combinations and while achieving optimal design section.

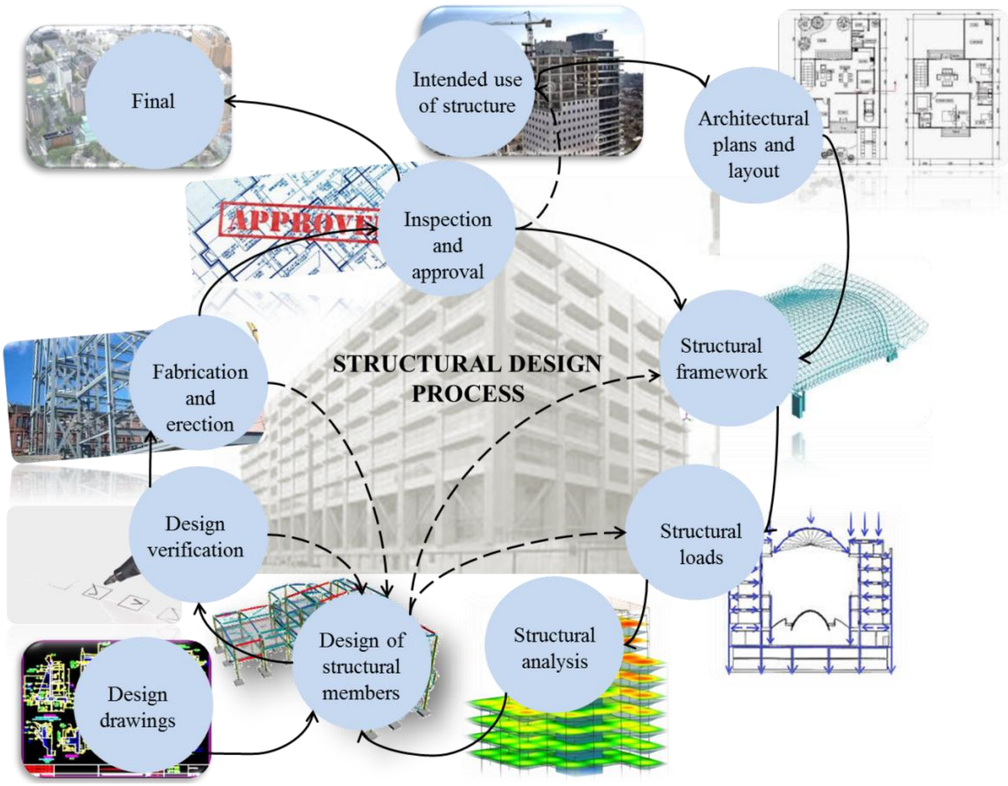


Figure 1. Structural design flow

Table 1 List of activities for the structural design flow

A	Architectural drawings
B	Structural geometry
C	Structural framework
D	Structural design
E	Design verification
F	Structural loads
G	Analysis of structure
H	Fabrication and erection
I	Inspection and approval

	A	B	C	D	E	F	G	H	I
A									
B	X			X					X
C	X	X		X					X
D					X		X	X	
E				X					
F			X	X					
G						X			
H					X				
I								X	

(a) Activity DSM for the structural design flow

Block

	A	B	C	F	G	D	E	H	I
A									
B	X					X			X
C	X	X				X			X
F			X			X			
G				X					
D					X		X	X	
E						X			
H							X		
I								X	

(b) Partitioned DSM for the structural design

Figure 2. Structural design iterations using DSM

4 Automation of iterations using DSM and OpenSTAAD

As explained in the previous section, iterations in structural design are unavoidable and automating them can speed up the design process. In this paper, OpenSTAAD was proposed to automate the STAAD iteration. OpenSTAAD provides a common platform to both STAAD and DSM in order to perform the structural analysis, design and also to automate the identified iterations. Further, it can export the design output after each iteration.

As seen in Figure 3, STAAD is responsible for executing structural analysis and design while Excel can model the DSM and identify iterations. At present, the refinement and finalization of section sizes in STAAD are done manually. Moreover, after each and every information update in design such as load combinations or a new section property to be assigned, etc. structural analysis and design needs to be performed manually for every iteration.

When the two models (i.e. STAAD and DSM) are integrated using OpenSTAAD as depicted in the Figure 3, it facilitates the constant customization and upgradation of structural framework in STAAD GUI (Graphical User Interface) by triggering the design iterations automatically. Few instances that are worthy to mention are alteration in the structural geometry, introducing a new member to framework or continuous refinements in member section sizes to achieve optimization, etc. STAAD includes a huge in-built database of various section sizes which is prompted by OpenSTAAD and runs the analysis and design after trying the specified section type. With the integrated VBA editor of OpenSTAAD, designers can identify the need for iteration or rework from the DSM model in Excel and direct the command to STAAD GUI to incorporate the necessary changes and re-run the structural analysis and design. The next section presents the detailed methodology for this integrated framework.

5 Proposed methodology and case illustration

Having elaborated the objective of the study, methodology for integrated framework for modelling and automating the iteration is elaborated under the following headings -1) Solution methodology and 2) Case illustration.

5.1 Solution methodology

For the integrated model explained in the earlier section and in Figure 3, the detailed solution methodology adopted in the present study is shown in Figure 4.

In simple terms, the methodology of the entire study was elaborated under 1) identification of design activities involved in the structural design phase and their information interdependencies, 2) formation of a structural design model in STAAD and activity DSM iteration model in MS (Microsoft) Excel, 3) DSM-STAAD integration and automation of structural design iteration using OpenSTAAD. The possible iteration loops emerging in structural design are also highlighted in Figure 4. OpenSTAAD provides the design outputs and results after each iteration as well as the final design reports once all the iterations are undergone and optimized section sizes are obtained.

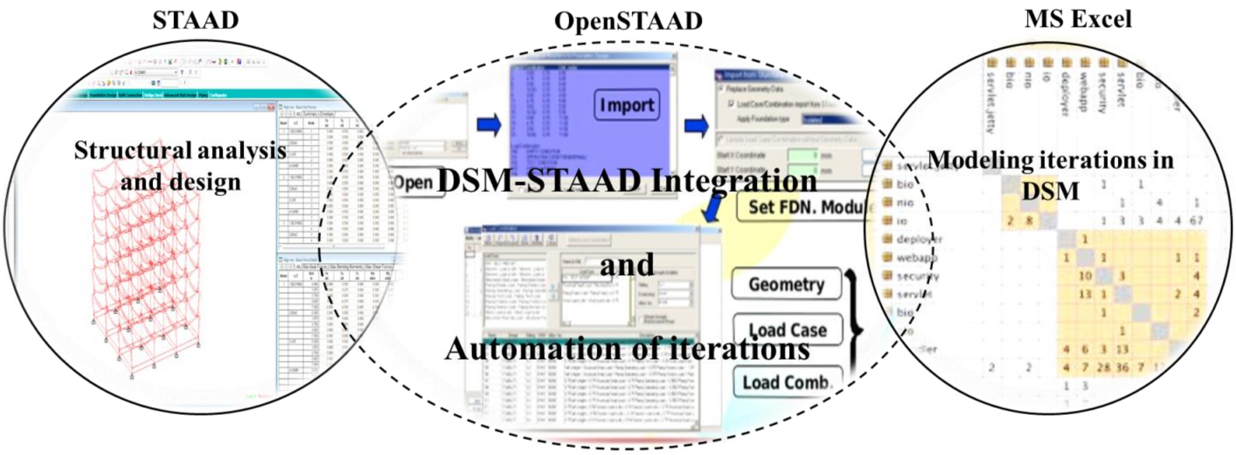


Figure 3. Proposed integration framework

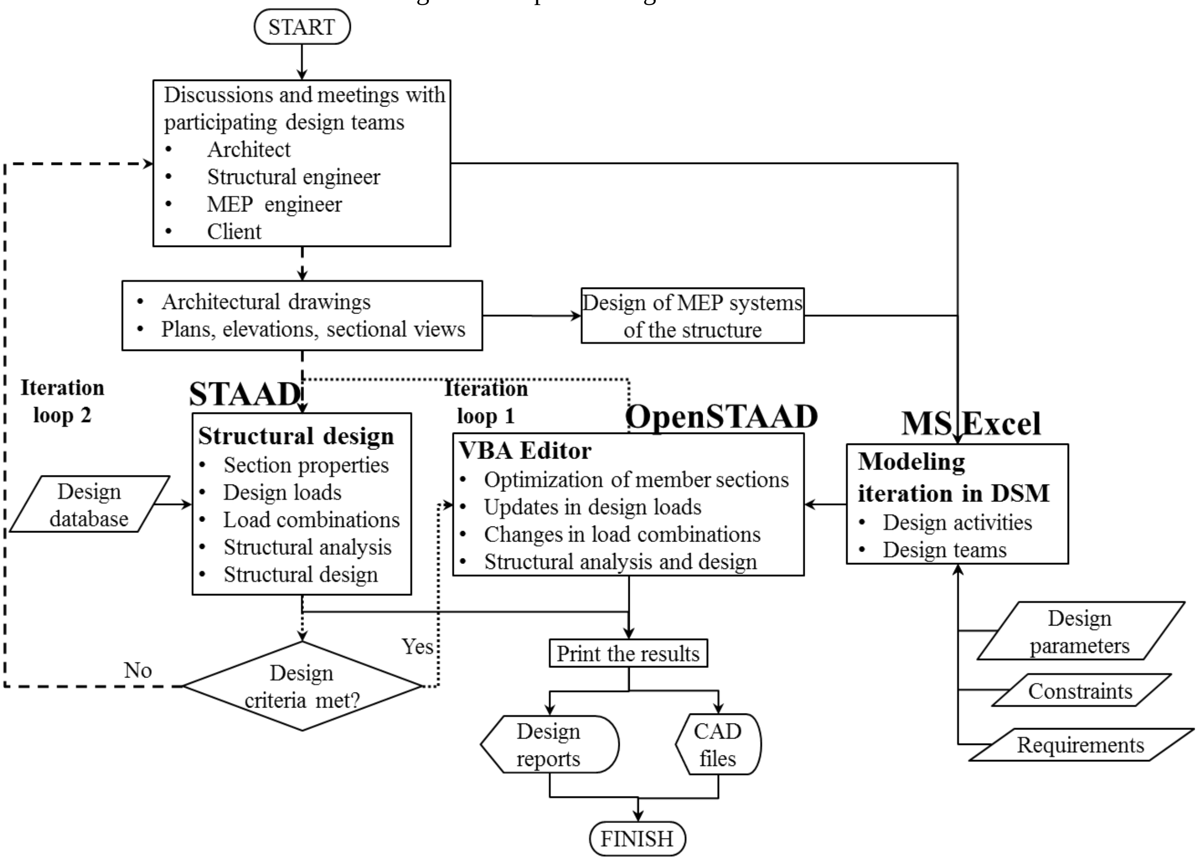
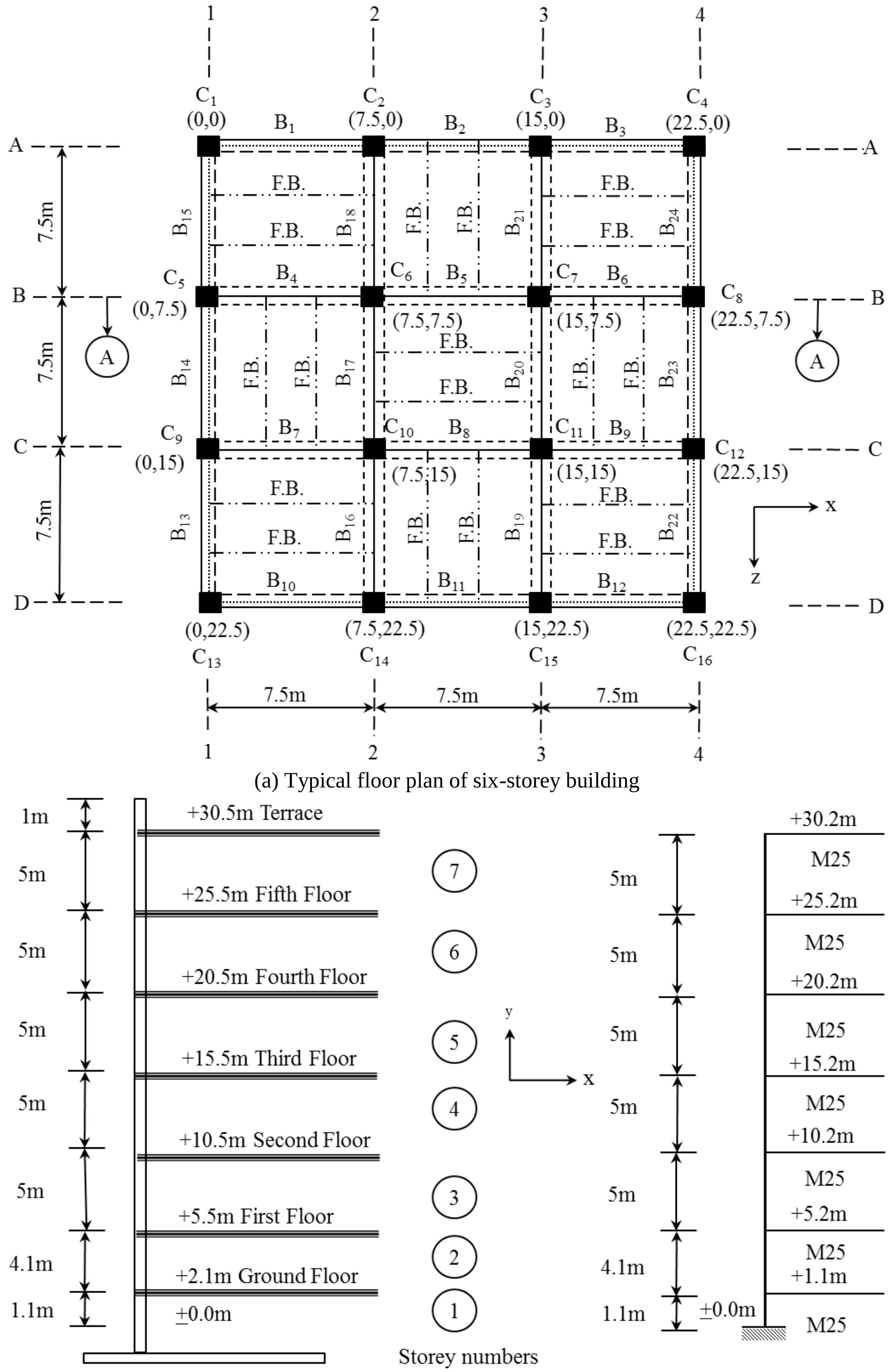


Figure 4. Solution methodology

5.2 Case illustration & Results

A six-storey building has been considered as a hypothetical case to perform structural analysis and design and to automate the iterations. The surface area of the structure is 22.5m x 22.5 m and includes six storeys each with a height of 5m as shown in the Figure 5. Fixed support conditions were considered for this structure and material properties were taken as steel. Once the structure is modelled in STAAD, changes in design as well as further refinement of member sections is done by undergoing iterations introduced by DSM using trial and error approach until the optimized sections are obtained. The trial section sizes of the members are taken from the in-built design database of STAAD as seen in Figure 4.



(a) Typical floor plan of six-storey building

(b) Part section A-A

(c) Part frame section of the six-storey building

Figure 5. General layout of the six-storey building

Once the STAAD is initiated, structural geometry is established by importing the architectural drawings and layouts as shown in steps 1-4 of Figure 6. Then, the structural framework is modelled by assigning supports, member properties, and loads/load combinations that are created as seen in steps 5-8 of Figure 6.

The analysis and design are performed on the structural framework developed as in steps 9 & 10. If there is a necessity to redesign or optimize, the DSM iteration model can reveal the upcoming iterations to be performed. At this juncture, OpenSTAAD starts executing the design iterations and optimization in the STAAD design model. As illustrated in the Figure 6, OpenSTAAD gathers input from DSM about the upcoming iterations such as changes in section properties of members, revising design loads to be considered or introducing new load combinations and automates them by directing STAAD to undergo necessary updates in design.

OpenSTAAD prints the design results every time once the analysis is done after incorporating each iteration. The results shown in Figure 7 represent the changes occurring in design output due to input provided by DSM iteration model. Figure 7a represents the first analysis output. The 'X' mark in the partitioned DSM (Figure 2b) depicting the iteration among the activities such as "structural geometry" and "structural design" prompts OpenSTAAD to choose the next section size (repeating step 6 of Figure 6) from STAAD design database. Then, the analysis is performed again (this implies steps 7 to 10 are repeated again) and the output is shown as Figure 7b. This process continues till the design is optimized and STAAD returns the design results that meet the established design criteria. Similarly, the output of iterations represented using DSM for update in design loads or combinations (retracing step 7 and/or 8 as seen in Figure 6), is shown as Figure 7c. Thus, the iterations are automated using the proposed integration framework.

6 Summary

The proposed integration framework comprising of STAAD-OpenSTAAD-DSM enabled the modelling and automation of iterations. It was evident from the case illustration that iterations were observed while revising design loads, introducing new load combinations and updating the section properties of members. Through OpenSTAAD it was easy to automate the iteration, export the design output and print the design results after each update that helped structural designers to track the information updates/changes.

The developed model in Excel VBA can automate all the iteration and summarize the results in less time which provides designers more time to evaluate the design rather than undergoing iterations. Estimation of the time taken by designers to execute the whole structural design process manually and automatically was computed during this case experiment. The duration taken by designers to manually incorporate the iterations for revising and assigning loads and combinations generally takes 10 to 15 minutes depending on the structure and the size of the structural members available. The optimization for the members sections required approximately 2 to 4 hours for manual execution. With the proposed Excel VBA model for the integration framework, the total time was drastically reduced to less than 5 minutes for this above-mentioned case.

Acknowledgement

The authors would like to acknowledge DST (Department of Science and Technology, Government of India, New Delhi) for providing the financial support to execute the work in this area.

References

[1] Bentley. STAAD Pro v8. Online: http://www.bentley.com, Accessed: 03/02/2015.

[2] Cho, S.H. and Eppinger, S.D. A simulation based process model for managing complex design projects. *IEEE Transactions on Engineering Management*, 52(3):316–328, 2005.

[3] Eppinger, S.D. and Browning, T.R. Design structure matrix methods and applications, The MIT Press, 2012.

[4] Lindemann, U. Design Structure Matrix. Online: http://www.dsmweb.org, Accessed: 29/01/2015.

[5] Microsoft Excel. Visual Basic Applications for Excel. Online: http://www.office.microsoft.com, Accessed: 03/02/2015.

[6] Mujumdar, P. and Maheswari, J.U. A design iteration framework for construction projects. *RICS Cobra conference*, pages 1–6, New Delhi, India, 2013.

[7] Mujumdar, P., Bhattacharya, S. and Maheswari, J.U. A study on DSM partitioning through case study approach. *International Conference on Research into Design*, pages 653-662. Bangalore, India, 2015.

[8] Steward, D.V. The design structure system: a method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 28: 71-74, 1981.

[9] Tuholski, S.J. Transformation, flow and value constellations in AEC projects, Doctoral thesis, University of California, Berkeley, 2008.

[10] Yassine, A.A., "An introduction to modeling and analyzing complex product development processes using the design structure matrix method", Quaderni di Management (Italian Management Review), 9, 1-17, 2004.

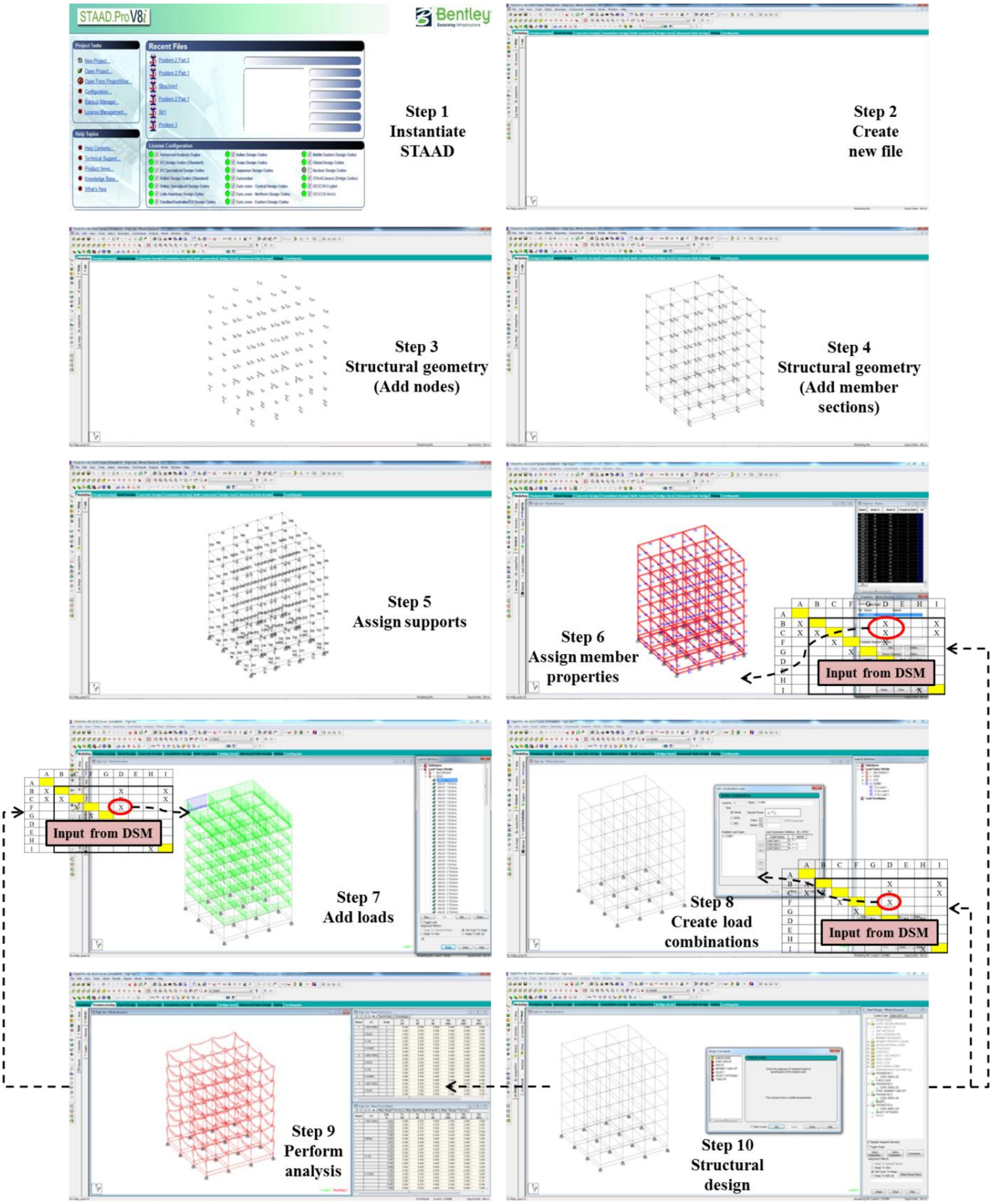


Figure 6. Automation of iterations using OpenSTAAD

	A	B	C	D	E	F	G	H	I
1	Beam	L/C	Node	Fx kN	Fy kN	Fz kN	Mx kNm	My kNm	Mz kNm
82	11	1 SELFWEIGHT	14	0	0.55	0	0	0	0.688
83			15	0	0.55	0	0	0	-0.688
84		2 DEAD	14	0	0	0	0	0	0
85			15	0	0	0	0	0	0
86		3 LIVE	14	0	0	0	0	0	0
87			15	0	0	0	0	0	0
88		4 COMB1	14	0	0.55	0	0	0	0.688
89			15	0	0.55	0	0	0	-0.688
90	12	1 SELFWEIGHT	15	0	0.55	0	0	0	0.688
91			16	0	0.55	0	0	0	-0.688
92		2 DEAD	15	0	0	0	0	0	0
93			16	0	0	0	0	0	0
94		3 LIVE	15	0	0	0	0	0	0
95			16	0	0	0	0	0	0
96		4 COMB1	15	0	0.55	0	0	0	0.688
97			16	0	0.55	0	0	0	-0.688
98	13	1 SELFWEIGHT	17	0.317	0.536	0	0	0	0.618
99			18	-0.317	0.564	0	0	0	-0.721
100		2 DEAD	17	-4.31	0.095	0	0	0	0.482
101			18	4.31	-0.095	0	0	0	0.233
102		3 LIVE	17	-17.249	0.37	0	0	0	1.886
103			18	17.249	-0.37	0	0	0	0.889
104		4 COMB1	17	-30.729	1.206	0	0	0	4.025
105			18	30.729	-0.105	0	0	0	0.892
106	14	1 SELFWEIGHT	18	0.286	0.55	0	0	0	0.688
107			19	-0.286	0.55	0	0	0	-0.688
108		2 DEAD	18	-3.465	0	0	0	0	-0.02
109			19	3.465	0	0	0	0	0.02
110		3 LIVE	18	-13.851	0	0	0	0	-0.081
111			19	13.851	0	0	0	0	0.081
112		4 COMB1	18	-24.649	0.55	0	0	0	0.542
113			19	24.649	0.55	0	0	0	-0.542
114	15	1 SELFWEIGHT	19	0.317	0.564	0	0	0	0.721
115			20	-0.317	0.536	0	0	0	-0.618
116		2 DEAD	19	-4.31	-0.095	0	0	0	-0.233
117			20	4.31	0.095	0	0	0	-0.482
118		3 LIVE	19	-17.249	-0.37	0	0	0	-0.889
119			20	17.249	0.37	0	0	0	-1.886

Input | DSM | Results | Revision 1 | Revision 2

(a) Output results after first time analysis

98	13	1 SELFWEIGHT	17	0.413	0.863	0	0	0	0.986
99			18	-0.413	0.911	0	0	0	-1.166
100		2 DEAD	17	-3.724	0.116	0	0	0	0.573
101			18	3.724	-0.116	0	0	0	0.295
102		3 LIVE	17	-14.892	0.443	0	0	0	2.217
103			18	14.892	-0.443	0	0	0	1.103
104		4 COMB1	17	-26.394	1.666	0	0	0	5
105			18	26.394	0.108	0	0	0	0.842

(b) Output results after first iteration – Update in member sections sizes

98	13	1 SELFWEIGHT	17	0.413	0.863	0	0	0	0.986
99			18	-0.413	0.911	0	0	0	-1.166
100		2 DEAD	17	-3.724	0.116	0	0	0	0.573
101			18	3.724	-0.116	0	0	0	0.295
102		3 LIVE	17	-18.615	0.552	0	0	0	2.767
103			18	18.615	0.552	0	0	0	1.374
104		4 COMB1	17	-31.978	1.83	0	0	0	5.825
105			18	31.978	-0.056	0	0	0	1.249

(c) Output results after second iteration – Update in assigned loads

Figure 7. Design results after iterations