Introduction to PSA as a Free Structural Analysis Software

Ishwaragouda S. Patil and Dr. Satish A. Annigeri

Abstract--- *Paper introduces the software PSA (Program for Structural Analysis), developed to analyze the behavior of space framed structures for various load combinations. Results obtained include section forces (bending moment, shear force at various sections of member), support reactions, node displacements, etc. in the analysis. The software is intended to generate a detailed report of calculations at various steps of analysis, which is lacking in commercial software. Access to this analysis steps may be used for educational and research purpose. Paper also discusses on various technical and programming implementations adopted in the development of software.*

Keywords--- *Structural Analysis, Software Development, Object Oriented Programming, STAAD.Pro*

I. INTRODUCTION

FOR decades structural engineering community has accepted the use of software applications to perform various civil structure related tasks like modeling, analysis and design of structures. While structural engineers have mainly concentrated on accepting the output from computer programs, very little importance is given to the analysis procedure that the software follows. In commercial software available these days, this facility is lacking. This paper introduces a free software that gives access to several intermediate steps followed during the structural analysis.

Program for Structural Analysis (PSA) was developed (Downloadable at http://sourceforge.net/projects/psafem), intended to give access to intermediate results, helping the user understand the underlying procedures.Instead of adopting the Procedural paradigm, Object Oriented Programming (OOP) was adopted as a better approach for the development of this software, as discussed in the following section. PSA is developed in Java programming language, because of its wide range of advantages, which are later mentioned in this paper.

II. ADVANTAGES OOP OVER PROCEDURAL PROGRAMMING

Prior to the emergence of structured programming, programmers used to write programs as a continuous stretch of code that extended line after line without any organizational structure. Such an unstructured approach led to tedious work in understanding, debugging and maintaining code, thus making the programs unreliable for their purpose[1]. Object-oriented systems are based on objects. An object is a package consisting of both data (the object's instance variables) and procedures (the object's methods). The instance variables define the object's state, and the methods define its behavior[2]. When used in program design, the object-oriented paradigm significantly increases the extensibility, maintainability, and reusability of the resulting code[3]. For three years G. Yu et.al worked on a research project for global-local finite element analysis of composite laminates using the traditional structured programming approach and the FORTRAN language. Then they noticed firsthand the immense difficulty in developing an easily readable, expandable, and maintainable software systems using the traditional approach. That research led them to explore and develop new programming paradigms and architecture[4]. When properly applied Encapsulation, Inheritance and Polymorphism combine to produce a programming environment that supports the development of far more robust and scalable programs than does the process procedure-oriented model[5]. Also, the object oriented programming complies with the standard UML documentation that generalizes the development procedure of any large scale project. Strong Object Oriented programming languages like Java have the feasibility of employing the Model View Controller (MVC) to easily handle the Graphical User Interface (GUI) in the software development process. Relational Database Management System (RDBMS), which is the most widely accepted data storage, retrieval and manipulation model of the Information Technology in the world can be easily mapped with Object Oriented Paradigm. Tools like Hibernate make this type of mapping easy in Java.

III. FEATURES OF JAVA

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.[6]

Ishwaragouda S. Patil, Prof., Civil Engineering, KLE Institute of Technology, Hubballi, India. E-mail:patil.ishwargouda09@gmail.com

Dr. Satish A. Annigeri, Civil Engineering, KLE Technological University, Hubballi, India. E-mail:satish.annigeri@gmail.com

DOI:10.9756/BIJMMI.8167

IV. FEATURES OF PSA

Features of PSA can be categorized into various Types as mentioned below:

A. Modeling Features

PSA accepts structure model representation in '.std' file of STAAD. Pro as its input. Various modeling features that are supported by PSA are:

1) Type of Structure Supported

The type of elements considered are by default based on the type of Structure declared in the beginning of '.std' file. Table 1 details about the different Structure Types

Table 1: Types of Elements PSA Considers for Different Structures Defined

Structure defined in STAAD file	Element considered	D.O.F of element
TRUSS	Plane truss element	4
PLANE	Plane beam	6
TRUSS SPACE	Space truss	6
SPACE	Space beam	12

2) Load Types Supported

In STAAD.Pro, Loads are differentiated on the location of load imposed as follows

a. Nodal/.Joint Load: Load can be directly applied on a predefined node. For example imposing a load of 150N in X direction and 120N in negative Y direction is given by STAAD command

JOINT LOAD

6 FX 150 FY -120

b. Member Load:Loads that are imposed on the beam element are considered to be Member Load(s). These can be of two types based on the configuration of the load

- *Concentrated Load:* To apply Concentrated Load of 80N at distance of 2.5 m from first node of beam number 1 following command is used

MEMBER LOAD

1 CON Y 80 2.5

- *UDL Load:* To apply UDL of 30N/m in interval of 1.0 m to 5.5 m on beam number 1, following command is used

MEMBER LOAD

1 UNI Y 30 1 5.5

Not giving the interval of the load (1 to 1.5m) indicates that UDL is to be imposed on the entire span of beam.

3) Supoort Specification

Supports can be defined as fixed, roller and hinged by releasing the appropriate D.O.F of node. For example if nodes 1,2,3 in structure are to be defined as fixed, roller and hinged respectively, following STAAD.pro commands are used

SUPPORTS

1 FIXED

2 FIXED BUT FX

FIXED BUT MZ

4) Cross Section Shape of the Memeber

In present development stage user can define the cross section property of the member in 2 different ways

- *Rectangular shape:* This is one of the very basic shape that STAAD.pro allows. Following STAAD.Pro command defines rectangular cross section of 0.1m×0.2m to elements 1 and 2

MEMBER PROPERTY AMERICAN

1 2 PRIS YD 0.1 ZD 0.2

- *Custom shape property:*STAAD.Pro and PSA allow users to define custom members, without mentioning the geometry of c/s. Commands below show one such example where members 1 to 3 have known c/s area(AX) and Moment of Inertia (IZ)

TO 3 PRIS AX 0.02 IZ 6.667e-5

5) Multiple Load Cases

A Load Case acts like a wrapper to a particular set of Load configurations, further load combinations can be made making use of the existing Load Cases. For instance, pertaining to Indian Standard code book IS 456 Loads combination to be considered is 1.5(DL+LL) for residential buildings. As of now PSA supports multiple load cases in the model, but 'LoadCombination' feature is not supported.

6) Materials

Like STAAD.pro model, PSA allows multiple Material types to be defined. Which means one can model a structure with members of different material properties like Steel, Aluminium etc. Properties given for the materials should be strictly in the following order

DEFINE MATERIAL START

YOUNGS MODULUS

POISSONS RATIO

MATERIAL DENSITY

ALPHA

DAMPING VALUE

B. Output Features

All the outputs except the intermediate steps of analysis (which is in HTML format) are provided in MS Excel 2007 format. Various results PSA provides are as follows

1) Support Reactions

Loads imposed on the structure are ultimately transferred to the supports. Obtaining the support reactions is essential in structural analysis and design procedure for the substructure design. The program is able to calculate all 6 reaction components (F_x,F_y,F_z,M_x,M_y,M_z) of the support, in space structures.

2) Section Forces

At any section in the beam element there are six forces acting, namely, Axial force, Shear-Y, Shear-Z, Torsion, Moment-Y and Moment-Z. PSA is able to process all the section forces arising due to loads in any 3D directions at various locations of the beam member.

3) Beam Deflections

Deflections and rotations at nodes are directly available from displacement vector processed from stiffness method. However, Area-Moment method is used to find the deflection at any intermediate location in the beam element. And the deflections obtained are in all the possible directions of reference (Global directions GX,GY,GZ and local directions X,Y and Z) as given by STAAD.Pro.

4) Analysis Report

The prime intention of developing PSA was to fill the void present in professional software, which is to give access to the procedures and their details carried out in the analysis of structures. Hence, PSA has the feature of reporting the intermediate steps of analysis in HTML format, so that viewing of large sized matrices is made easy for the user. This is well suited by giving the scrolling bars in the window of webpage as shown in Fig. 1 below.

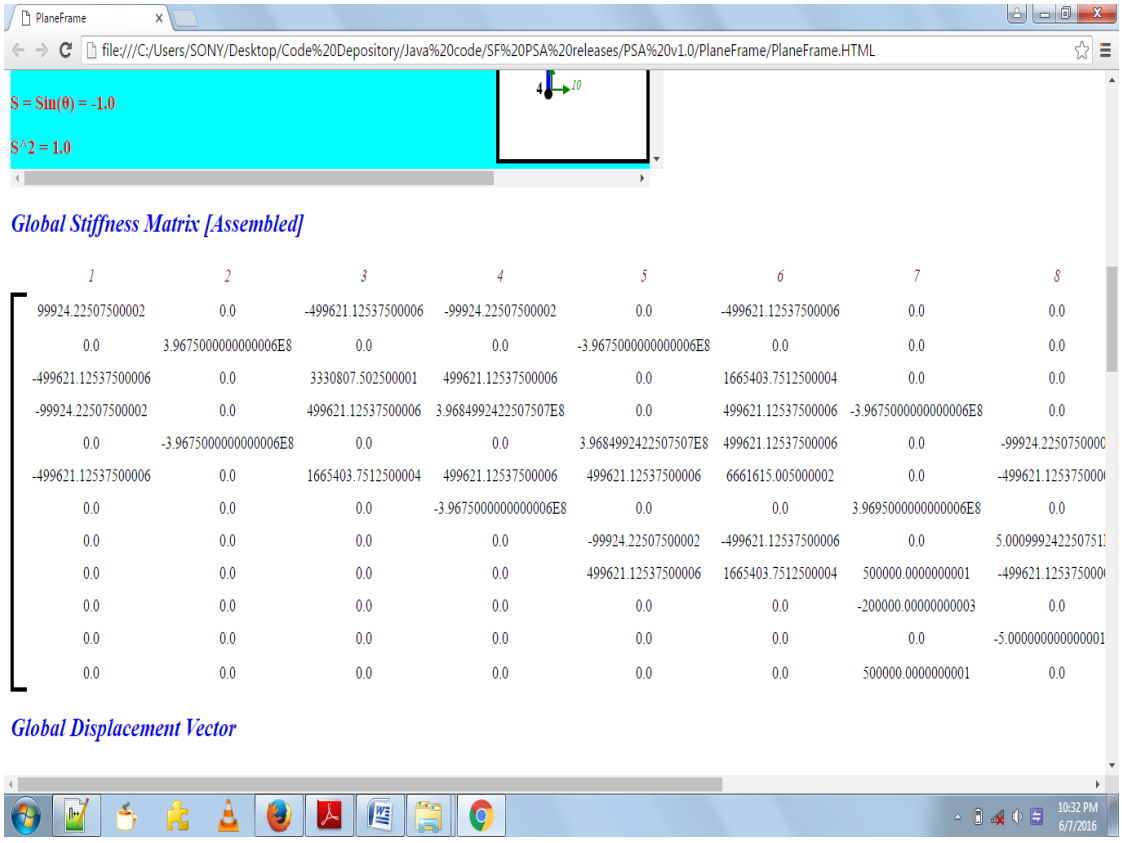


Figure 1: Analysis Report Shown in HTML Format

V. Data Input for PSA

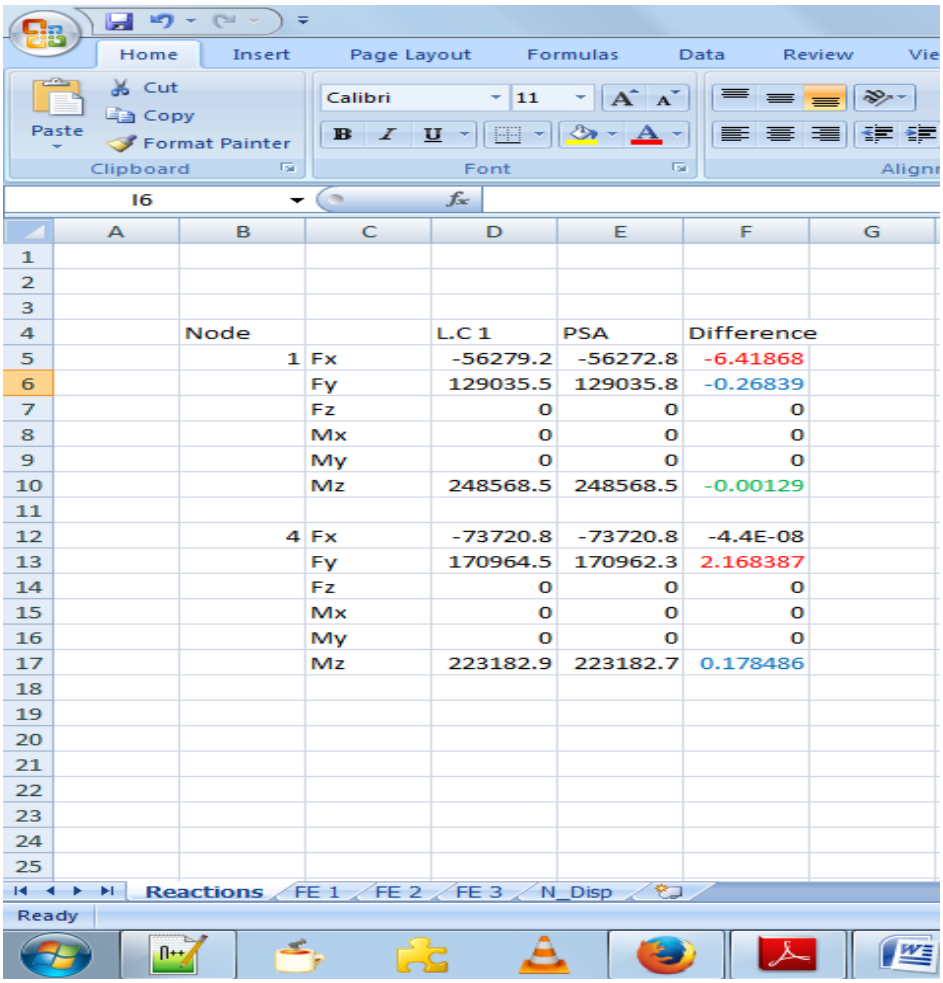
STAAD.Pro has a simple text based input data file with specific keywords that describe the type of input. Knowing this file format and grammar, other programs can parse and interpret the input data and build the structural model. Thus structural models can be built using STAAD.Pro GUI, saved to .std format and then read in by PSA thereby simplifying data input, and storage format. In Software development phase, this gave PSA the additional benefit of using STAAD.pro to visualize the model, thus avoiding GUI development in the development process.

VI. Testing Process

In order to verify the result output of PSA it was required to find a easy means to get the work done. Such automation could be done if the developer has access to the result data of any analyzed structure from a professionally accepted software. 'Open STAAD' is one such facility which helps to do such programming interface with analysis functionalities of STAAD. Open STAAD is a library of exposed functions allowing engineers access to STAAD. Pro's internal functions and routines as well as its graphical commands. With Open STAAD, any user can use practically any programming language (including C, C++, VB, VBA, FORTRAN, Java and Delphi) to tap into STAAD's database and seamlessly link input and output data to third-party applications. Section 4 describes how this facility was leveraged to automate the software testing process.

With Open STAAD facility in hand, two programs were created to perform testing. First program written in Visual Basic(VB), which invokes Open STAAD library to analyze a sample structure, query the results from it and place them in an excel file. The Second program written as integral part of

PSA, analyses the same structure and places its obtained values in the same Excel File, also validating the results against the ones placed by VB program. In its validation process the program also reports the differences of both results, by highlighting the severity of difference in various colours (Green for low severity, Blue for medium, Red Color for High Severity) as shown in figure 2. Further one more program is written to do the batch testing of various structures simultaneously, to understand how the modified code has effected on the results of other bunch of structure files.



	A	B	C	D	E	F	G
1							
2							
3							
4		Node		L.C 1	PSA	Difference	
5		1	Fx	-56279.2	-56272.8	-6.41868	
6			Fy	129035.5	129035.8	-0.26839	
7			Fz	0	0	0	
8			Mx	0	0	0	
9			My	0	0	0	
10			Mz	248568.5	248568.5	-0.00129	
11							
12		4	Fx	-73720.8	-73720.8	-4.4E-08	
13			Fy	170964.5	170962.3	2.168387	
14			Fz	0	0	0	
15			Mx	0	0	0	
16			My	0	0	0	
17			Mz	223182.9	223182.7	0.178486	
18							
19							
20							
21							
22							
23							
24							
25							

Figure 2: Automated Result comparison of Results, Highlighting Severity of Difference in Various Colours

VII. Technical Implementations

Following section discusses some of the key technical concepts that were employed to code the analysis procedures

A. *Node Displacements*

Obtaining the node displacements is very straight forward. All the element Stiffness Matrices are assembled to obtain Global Stiffness Matrix $[K]_g$, and the Global force Vectors $[F]_g$ are formed. It is to be noted that Force vectors may be many, based on the number of Load Cases defined in the model. Whereas the $[K]_g$ remains only one for the structural system. To get the displacements at various D.O.F the following operation is done on these two matrices

$$[d]_g=[K]_g^{-1}*[F]_g$$

B. *Obtaining Support Reactions*

To obtain the support reactions, the row from GSM corresponding to co-ordinate number of support reaction is multiplied with the global displacement vector.

C. *Computation of Forces on Nodes*

To obtain Axial force, SF and BM at ends of a beam, the row from Element Stiffness Matrix in local coordinates are multiplied with member end displacements in local coordinate system.

D. *Sectional Forces*

Sectional forces at any intermediate location of the beam element are obtained by the superposition of effects of the nodal forces, along with the effect of individual member forces imposed on the beam element. For instance to obtain the BM at any location, a linear variation curve (straight line joining the BM values at both the extreme nodes) is hypothesized, then combined with the free moment curves arising from effect of multiple member loads is obtained.

E. Beam Deflection

To obtain the shape of beam deflection, Moment-Area theorem is employed. Out of the two Moment-Area theorems only the second law, which mentions of obtaining the Transverse deflection is used. The first law describes about obtaining the rotational (angular displacement) at any intermediate location of the beam. However to obtain the actual deflection shape rotations at the beam ends is required, which can be directly obtained by the Global Displacement Vector without needing to use the first theorem.

VIII. AREAS OF PSA WITH SOME KNOWN DIFFICULTIES

There are some areas in PSA which require modification to achieve more standards

A. Analysis Accuracy of Members with High Cross section/Length Ratio

Analysis results of plane frames and space frames are differing largely when compared with STAAD.pro results for elements whose cross sections/length ratios are large. This is because shear deformations are predominant in such members and must be included in the stiffness formulation, which is yet to be included in PSA.

Further study is required to find the cause of large result differences in the cases where elements slenderness is low. Then necessary corrections can be implemented in the code.

B. Taking Care of Instability of Structure

The Global Stiffness Matrix becomes non invertible for structures with less supports. Without our specifying adequate kinematic constraints or support conditions, the structure will be free to move as a rigid body and not resist any applied loads. Under these circumstances PSA fails to analyze the structure. But STAAD.pro inverts such matrices and performs the post-processing. Hence, further study is required in this area to analyze the structures with instability issues.

C. Providing the Complete Report of Analysis Procedures

Presently, the program generates report detailing FEM calculations only (Until finding node displacements only). Also report about calculations done for creating SFD, BMD and Beam Deflections need to be generated in the web report by program.

There must be an facility of obtaining only the desired analysis report instead of getting the entire report of analysis. The user must be able to choose, through suitable input data, which reports are to be generated so that the output can be studied easily.

D. Improvement in Parsing the '.std' File

Presently the module that parses '.std' file to PSA model is not fully flexible, as it adheres to particular sequence of commands only. This mandatory sequence is not enforced in the STAAD.pro software. Eventually development will be done to eliminate this stringency in the file format.

IX. CONCLUSION

PSA, a free software capable of performing most of static analysis like any professional software is introduced, along with the technical methods incorporated to obtain those results. Recent programming languages like Java have much to offer in software development process. It is necessary to adopt Object Oriented Programming paradigm for the development of Structural Analysis software, as it handles data and methods of the program in more structured and effective manner.

REFERENCES

[1] A. Madan, "Object Oriented Paradigm in programming for Computer-Aided analysis of structures," Journal of Computing in Civil Engineering , Vol. 18, No. 3, Pp. 226, 2004.

[2] R. Sause, K. Martini and G.H. Powell, "Object-oriented approaches for integrated engineering design systems", Journal of computing in civil engineering, Vol. 6, No. 3, Pp. 248-265, 1992.

[3] S. Moni, and D.W. White, "Object-Oriented visualization system for frame analysis," Journal of Computing in Civil Engineering , Vol. 10, No. 4, Pp. 276-285, 1996.

[4] G. Yu and H. Adeli, "Object-Oriented Finite Element Analysis using EER model," Journal of Structural Engineering , Vol. 119, No. 9, Pp. 2763-2781, 1993.

[5] Herbert Schildt, "Java: the complete reference" McGraw-Hill Education India Pvt. Ltd, 7th Indian Edition.

[6] Website:https://en.wikipedia.org/wiki/Java_(programming_language)