



StrucPy.RCFA- Object-Oriented Python Package for Structural Analysis of Reinforced Concrete Frames

SOFTWARE METAPAPER

TABISH IZHAR 

SYED AQEEL AHMAD 

TASNEEM AHMED 

NEHA MUMTAZ 

*Author affiliations can be found in the back matter of this article

]u[ubiquity press

ABSTRACT

StrucPy.RCFA is an open-source object-oriented Python package, capable of performing structural analysis on reinforced concrete (RC) members and 2D/3D reinforced concrete frames. With the increasing necessity of skill development among young engineers in developing countries, StrucPy.RCFA provides a prodigious open-source solution to civil engineering students and researchers. Students can visualize two- and three-dimensional reinforced concrete frame behaviour with a few simple lines of code without depending upon any commercial software. This package provides a simple solution to the laborious task of input without extensive programming to help users from non-programming backgrounds. This package provides a range of features, including the auto-generation of frames and floors. The package has the potential to assist reinforced concrete design optimization to be more practicable by providing complete control over the analysis data and to provide an interactive learning experience to structural engineering students to achieve sustainable development goal 4 (SDG4) of equal access to quality education.

CORRESPONDING AUTHOR:

Tabish Izhar

Department of Civil Engineering,
Integral University, Kursi Road,
Lucknow-226020, India

tizhar@iul.ac.in

KEYWORDS:

Civil Engineering; Reinforced
Concrete Member; Reinforced
Concrete Frame; Structural
Analysis; StrucPy; Open-Source

TO CITE THIS ARTICLE:

Izhar T, Ahmad SA, Ahmed
T, Mumtaz N 2024 StrucPy.
RCFA- Object-Oriented Python
Package for Structural Analysis
of Reinforced Concrete Frames.
*Journal of Open Research
Software*, 12: 10. DOI: [https://
doi.org/10.5334/jors.489](https://doi.org/10.5334/jors.489)

(1) OVERVIEW

1. INTRODUCTION

Structural analysis is the process of calculating and determining how loads and internal forces will affect the behaviour of a structure. It involves the application of engineering principles and mathematical techniques to understand how structures behave under different conditions [1]. Reinforced concrete (RC) frames are one of the most used and common structural systems used in the construction of buildings. The combination of concrete and steel reinforcement allows the structure to withstand significant loads. Thus, in developing nations, reinforced concrete frame structures were and are still being frequently used for building due to access to cheap labour [2]. The analysis of reinforced concrete frame structures is a bulky and complex task, that involves structural analysis with load calculations, and adherence to building codes and regulations. Engineers, aspiring structural engineering students, and researchers must have access to tools capable of performing structural analysis to ensure the structure can safely support the intended loads throughout its design life. Almost all structural engineering students recommend the use of software for structural analysis problems [3]. Yet, students and researchers mainly depend upon institutions to provide commercial/student licenses to software like SAP2000, ETABS, and STAAD Pro to perform structural analysis [4]. Developing nations especially India have seen rapid growth of private institutions in engineering education with the majority lacking access to quality education due to lack of funding [5], thus being unable to provide expensive tools to students. Despite structural analysis of reinforced concrete frames being significant in the curriculum, most students lack the ability to visualize the behaviour of structures, which is crucial for learning [6]. The institutions are still practising the conservative method despite being inefficient and obsolete [7]. The ineffectiveness of traditional teaching techniques lies in the fact that it puts a lot of effort into analysing individual members and places too little emphasis on comprehending how the overall structure behaves in three dimensions [8]. This has created an inability among young engineers to comprehend the behaviour of 3D RC frames resulting in low employability. Further, sole dependency on commercial software without knowing the background creates a “black-box syndrome” resulting in the absence of decision-making among young engineers [9].

Structural analysis is the first step in the design optimization of reinforced concrete frames [10] which has been one of the prominent areas of research in the field of civil engineering. Despite having a large database, little work has been done on three-dimensional frames that represent realistic scenarios [11] due to the complexity of the structure, where getting the analysis results in

a loop becomes difficult and cumbersome. Structural analysis provides data for further actions like design and optimization. A collaboration between some structural analysis software and programming platforms is required, where results can be exchanged for optimization. The work on a three-dimensional reinforced concrete frame lags due to the unavailability of automation and a platform to exchange computer data and information [12]. SAP2000-MATLAB platform has been used a few times for the exchange of structural analysis data [13], [14] along with ETAB(OAPI)-MATLAB [15–17]. ANSYS-MATLAB [18, 19], OpenSees-MATLAB [20, 21], and STAAD Pro-MATLAB [22]. Apart from OpenSees [23], open-source software and packages have hardly been used due to the lack of user-friendliness and low reliability [24]. Open-source software for structural analysis is still an unexplored territory in the field of structural engineering due to its complexity [25] and lack of programming skills among civil engineers [26]. Students and researchers not only need an open-sourced software/package capable of performing structural analysis accurately but also require it to be more user friendly i.e., requiring less programming skills and transparency in calculations.

Python [27] being open source and easy has grown tremendously in the scientific community [28]. The simple learning curve of Python makes it a perfect choice for students from civil engineering backgrounds to get a grasp on it [29]. Git-hub repositories were explored for the relevant Python libraries and packages, many of which were found to be inactive. The repository without any examples, documentation, and readme files was discarded. General finite element solvers like SfePy, PyFEA, xfeMm, FEniCS, etc. are not considered because of their requirement of a high understanding of finite element methods and programming. Several features of the libraries were explored especially those that affect the realistic behaviour of reinforced concrete framed structures such as 3D visualization of behaviour, seismic analysis, load combinations, and load envelopes. The amount of information and data provided by the libraries to users were also investigated under user control over data. Additionally, the amount of programming/coding skills required to perform structural analysis is categorized as low, medium, and high. Some of the notable libraries that are useful purely for the purpose of structural analysis are shown in Table 1 along with a detailed study on the capability of handling reinforced concrete framed structures.

PyNite, OpenSeespy, structural_analysis, adapy, and pyframe3DD libraries were found to be capable of handling three-dimensional structures, yet none is being greatly used for the research and visualizing 3D reinforced concrete structures in the civil engineering curriculum. All the libraries require a lot of coding from modelling to structural analysis except pyframe3dd, as it provides a simple solution to large input data. None of the open-

Table 1 Python Libraries for Structural Analysis.

LIBRARY/PACKAGE	DOCUMENTATION AND EXAMPLES	3D REINFORCED CONCRETE FRAMES	3D VISUALIZATION	CODING	SEISMIC ANALYSIS	LOAD COMBINATIONS	LOAD ENVELOPE GENERATION	USER CONTROL OVER DATA	COMMENTS
anastruct [30]	Yes	No	No	High	No	Yes	No	Medium	-----
Adapy [31]	No	Yes	Yes	High	Yes	Yes	No	Low	<ul style="list-style-type: none"> • Dependent on other software to perform analysis. • Load data and steps are not available.
feastruct [32]	Partial	No	No	High	No	No	No	Low	-----
nusa [33]	Yes	No	No	High	No	No	No	Low	-----
openseespy [34]	Yes	Yes	No	High	Yes	Yes	No	High	<ul style="list-style-type: none"> • The load combination for the analysis of structure is not very clear.
pyframe3dd [35]	Yes*	Yes	Yes	Low	Yes	Yes	No	High	<ul style="list-style-type: none"> • Python binder to Frame3DD software. • *Frame3DD python framework lacks proper documentation.
PyNite [36]	Yes	Yes	Yes	High	No	Yes	No	Medium	-----
StructPy [37]	No	No	No	High	No	No	No	Low	-----
structural_analysis [38]	No	Yes	Yes	High	No	No	No	Medium	-----
Structural-Python [39]	No	No	No	High	No	No	No	High	-----

source libraries were found to be capable of generating load envelopes which simplifies and organizes data for further usage. The amount of control over data is very limited in the available open-source libraries, providing access to only basic results. There are very few libraries that offer access to the stiffness matrix of the entire structure, seismic shears and details, story drifts, and complete information on members' forces. None of the explored open-source libraries offers output in formats that are as easy to understand as that of commercial software. One of the major requirements of a structural analysis package is to generate a structural model effortlessly with fully interactive graphics and to provide a pictorial interpretation of analysis results [40], which most open-source software and packages are lacking.

The present paper elucidates the functionality, features, and advantages of the RCFA module, which is an integral component of the 'StrucPy', a novel Python package designed especially for the structural analysis of reinforced concrete framed structures. RCFA module can perform structural analysis of the RC members and frames, be it simple 2D beams or intricate 3D RC frames. The hallmark of RCFA lies in its user-friendliness and the capacity to perform comprehensive analysis with minimal lines of code without in-depth knowledge of programming. The paper covers the theoretical background, implementation details, quality control, application examples, and result validation of the package. The presented package is originally developed for students and researchers working in the field of civil engineering to study the behaviour of reinforced concrete framed structures.

2. IMPLEMENTATION AND ARCHITECTURE

2.1. Theoretical Background

The StrucPy.RCFA package adopts the direct stiffness method to perform the structural analysis of RC members/frames where every structural element is represented by its 12×12 stiffness matrix $[k]$ in the local coordinate system. The direct stiffness method is grounded on three principles: Equilibrium, Compatibility, and Constitutive Behaviour [41]. Each structural element is represented by its 12×12 stiffness matrix $[k]$ in the local coordinate system [42], which quantifies its resistance to deformation. Every local stiffness matrix in the local coordinate system is then transformed into the global coordinate system using the transformation matrix $[T]$ [43]. By assembling these matrices, a global stiffness matrix $[K]$ for the entire structure is obtained by recognizing the shared nodal degrees of freedom between elements [44]. The unknown displacements are found by invoking matrix algebra, particularly matrix inversion techniques [45]. With the known forces and the inverse of the appropriate submatrix of $[K]$, the nodal displacements are determined. With the displacements known, the elemental force-displacement relationship

is used to retrieve internal forces for each member considering the load distribution within the structure. The finite difference method (FDM) is used in this package to calculate the displacements at various points within the RC elements. It operates by effectively discretizing the continuous domain into a series of nodal points [46]. The differential equation is then transformed into algebraic equations, which are solved simultaneously to attain the solution [47].

RCFA module performs static seismic analysis of RC frames as per IS 1893:2016 [48]. The analysis is performed using the elastic approach, and the structure's response is evaluated under the action of the design earthquake by applying the equivalent lateral forces. The equivalent lateral load distribution is determined based on the seismic zone factor, importance factor, and response reduction factor specified in the code. The calculated forces are then distributed along the height of the structure using appropriate lateral load distribution mechanisms as suggested by the code. The structure's dynamic characteristics, such as time periods, seismic shear, and story drift, are obtained through analysis. The stability index Q is a parameter used to quantify the relative lateral stiffness of a column and determine its behaviour as either a sway or non-sway column. The calculation of stability index Q is carried out as per annex E of IS456:2000 [49]. The effective length of the columns is also determined as per IS456 which is based on the work of wood [50]. The calculated stiffness ratio is then used to determine the effective length factor, separately for sway and non-sway frames.

2.2. Architecture

RCFA is an object-oriented module of the 'StrucPy' library purely written in Python, that encapsulates reinforced concrete elements/frames into an object. The RCFA module exhibits a sophisticated interaction with a group of other Python libraries, capitalizing on their individual strengths to provide an effective toolkit for the structural analysis of reinforced concrete members/frames. One of the foundational dependencies is 'numpy' [51], an essential library in the scientific computing realm. Within RCFA, numpy controls complex matrix manipulations and operations, providing both speed and precision. This numerical efficiency is particularly vital when dealing with large datasets of complex reinforced concrete frames. Another dependency is 'pandas' [52], which provide efficient data structures for data manipulation and analysis. RCFA leverages pandas dataframe for structured data storage, enabling easy querying, filtering, and aggregation. The third dependency is 'plotly' [53], a versatile graphing library. Through plotly, RCFA offers intuitive and interactive visualizations with features like zooming, panning, and hovering over data points to display additional information. This interactivity enables users to explore data more dynamically and engagingly.

Lastly, the ‘ray’ [54] library enhances the scalability and performance of RCFA. Ray provides a framework for parallel and distributed computing. Intensive tasks are distributed across multiple cores, dramatically speeding up computations. In essence, each of these libraries contributes unique capabilities that, when combined, make RCFA a potent tool for structural analysis of reinforced concrete members/frames.

Currently, two Python classes are being used in the module, making it simple to manage and expand. Each of the classes contains several methods, capable of performing various operations. A complete list of classes along with their methods of the RCFA modules is shown in Figure 1. RCF class of RCFA deals uniquely with single load combinations and contains numerous useful methods capable of collecting specific information from RC member/frame objects. Another class RCFenv of RCFA handles multiple load combinations and generates an envelope for load combinations. All the methods of RCF are accessible from RCFenv to retrieve complete data related to any specific load combination using ray actors.

RCFA’s classes accept model input data as instance variables in the form of values, boolean operators, and dataframes. In the context of RC frames where input parameters can vary to thousands, a dataframe is utilized to feed various parameters and properties of the frame in an organized manner. RCFA requires some mandatory model parameters to carry out structural analysis like frame nodal details, member details, and boundary conditions. These mandatory arguments are only accepted in the form of a dataframe. The dataframes

consist of columns representing various detail of instance variables. Each row corresponds to a specific member of the frame with a unique identification (ID) number. The IDs of all the members and nodes should be a natural unique number assigned to them as the dataframe’s index. Users can populate the dataframe with relevant data extracted from architectural drawings and other requirements. This approach not only streamlines the input process but also enhances the reusability of data and facilitates easy modification of input parameters. Currently, RCFA offers various input arguments, which are presented in Table 2 along with its data type and utility. The templates of all the dataframe instance variables can be accessed in the repository folder ‘inputfiles’.

2.3. Features

RCFA is a powerful package specifically for the structural analysis of RC members and frames. It is capable of generating visualization for 3D models, deflected shapes in 3D space, shear force diagrams, bending moment diagrams, and deflected shapes of members using various package methods. The module is equipped with automated generation of frames and floors which allows to perform complex structural analysis with very few lines of Python code. Perhaps one of the most promising features of the module is the high level of user control. Users can control every aspect of the model input, model generation, and model data, allowing for great flexibility in terms of usage. Currently, RCFA is capable of handling structural analysis from single beam elements to complex RC frames. A detail of all the module features

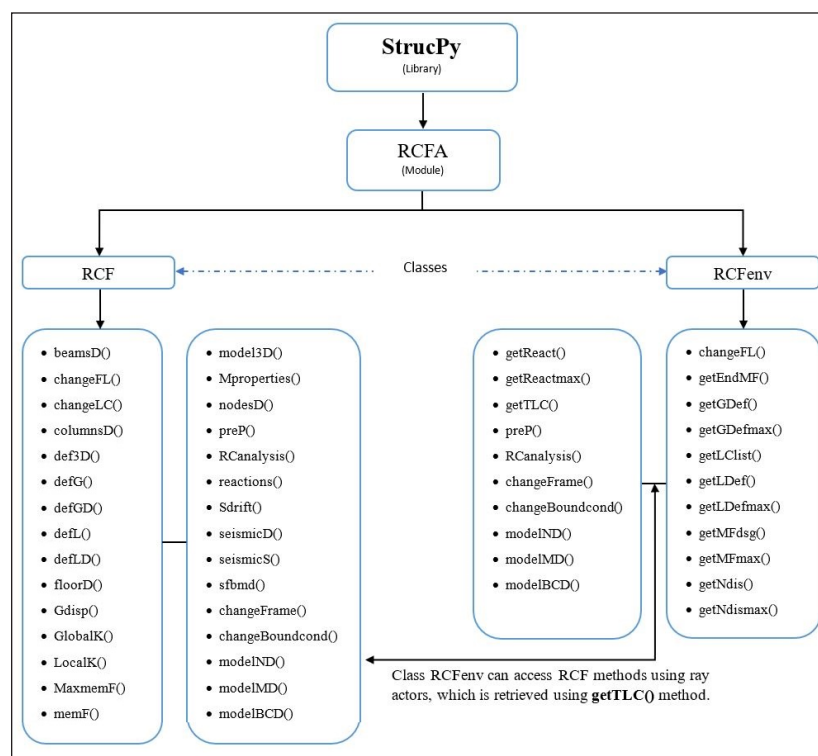


Figure 1 Constituents of RCFA package.

S.NO.	INSTANCE VARIABLE	TYPE	REQUIREMENT	DESCRIPTION
1	nodes_details	DataFrame	Mandatory	Collects the coordinates of nodes in space
2	member_details	DataFrame	Mandatory	Collects the details of members of the structure such as: <ul style="list-style-type: none"> • Id of two nodes connecting member • Width and depth of a member • Uniformly distributed load in local x, y and z directions.
3	boundrycondition	DataFrame	Mandatory	Recognizes the type of restraints at various nodes. It's mandatory to define the base support for reinforced concrete frames.
4	framegen	DataFrame	Optional	Automatically generates frames based on the input of frame bays detail.
5	forcesnodal	DataFrame	Optional	Collects information on any nodal forces acting on the member or frame in a global coordinate system.
6	slab_details	DataFrame	Optional Default: None	Identify the floor's location, thickness and distributed loads as dead and live loads.
7	load_combo	DataFrame	Optional	Represents the load combination to be used for analysis. By default, it considers only dead load with factor 1.
8	seismic_def	DataFrame	Optional	Carry information on seismic considerations to be used in seismic analysis. By default, the module follows IS1893:2016. Note: Only a time period is required if the user intends to use any other guidelines.
9	self_weight	Boolean	Optional	By default, RCFA considers self-weight. If set to False, self-weight is ignored.
10	infillwall	Boolean	Optional	To be used only in case of seismic analysis as per IS1893. It carries information on whether to consider an infill wall or not in order to determine the time period.
11	autoflooring	Boolean	Optional	It automatically applies floors along with loads on every story of the frame. It reduces the user's work of defining floors on complex frames.
12	properties	DataFrame	Optional	Users can assign new properties to beams, columns and slabs. By default, RCFA considers concrete of Grade 25MPa and the rest of the properties are calculated as per IS456:2000.
13	grade_conc	Float/Int	Optional	It allows users to change the grade of concrete.
14	point_loads	DataFrame	Optional	Identifies multiple point loads on the members.
15	col_stability_index	Float	Optional	Value is used to classify columns as sway or non-sway. By default, the value is 0.04 as per IS 456:2000

Table 2 Instance variables of the RCFA package.

is presented in [Table 3](#). There are a few limitations to the feature of floor assignment and floor load distribution. Presently, the floor is considered a flexible diaphragm with zero in-plane stiffness, and floor loads are distributed according to the tributary area. Moreover, the floor should be surrounded by beam elements on all four sides to distribute loads efficiently and correctly.

RCFA's error management capabilities make it very user-friendly and easy for early programmers. RCFA requires a large number of input details for analysis in various formats. Further, some methods of RCFA are required to be called sequentially. To address these challenges and to assist users, the RCFA library has been equipped with various exceptions. Input validation is quite vigorous as it prevents incorrect or malicious data from causing problems. RCFA offers an extensive range of input validation, from data types to dataframes

column arrangement details; ensuring that the input adheres to the expected formats and columns of the dataframe. For instance, RCFA.RCF.member_details raises an exception if the input value is not a dataframe as shown in [Figure 2\(a\)](#). Similarly, if the IDs of members and nodes are incorrect it raises an exception as presented in [Figure 2\(b\)](#). Columns of the dataframes are positional arguments and cannot be left out. The index of the boundary condition must be a valid node ID as it represents the joint conditions. In order to tackle the correct input of columns and valid IDs, RCFA identifies the invalid IDs and correct order of the columns through raising exceptions as shown in [Figure 2\(c\), \(d\)](#). To ensure the correct order of methods execution an exception is raised, for example, if the analysis is called before the pre-processing of model data an exception is raised as shown in [Figure 2\(e\)](#).

FEATURES	DESCRIPTION
Structure	Beam and column elements.
	2D geometrically linear reinforced concrete frames
	3D geometrically linear reinforced concrete frames
Restrains	Fixed Joint
	Hinged Joint
	Roller Joint
Loading	Multiple point loads on a 1-dimensional elements
	Uniformly distributed loads on elements
	Self-weight
	Floor dead load
	Floor live load
	Static seismic load
	Nodal loads
	Load combinations
	Load envelopes
Element Sections	Rectangular
	Square
	Circular
Auto Flooring	Applies floor and it loads automatically on every floor of 3D RC frames
Material Properties	Material properties of RC frame members can change as per the need
	Different member categories can have different properties
Post Analysis	Reactions data
	Nodal displacement data
	Global and local stiffness of the RC frame and members respectively
	Shear force and bending moment data on every element of the RC frame
	Global and local deflection data on every element of the RC frame
	RC elements design data
	Stiffness data on beams, columns, and nodes
	The effective length of columns
	Categorize sway and non-sway frames
	Seismic shear data
Story drift data	
Lumped mass data	
Seismic Definition	Allows users to pass seismic definition as per the requirement.
Stability Index	Allows users to control the parameter to categorize sway and non-sway frames.

Table 3 Features of RCFA package.

```

TypeError                                 Traceback (most recent call last)
c:\Users\Civil Lab\Desktop\tabish\Phd\Git_Repository\StrucPy\Test4.py in line 18
TypeError: Type of the argument 'member_details' must be DataFrame
(a)

TypeError                                 Traceback (most recent call last)
c:\Users\Civil Lab\Desktop\tabish\Phd\Git_Repository\StrucPy\Test4.py in line 17
TypeError: Node Number(Index) in 'nodes_details' must be 'int' type
(b)

Exception                                 Traceback (most recent call last)
c:\Users\Civil Lab\Desktop\tabish\Phd\Git_Repository\StrucPy\Test4.py in line 17
Exception: MEMBER DETAILS must have 7 columns: ['Node1', 'Node2', 'b', 'd', 'xUDL', 'yUDL', 'zUDL'].
First 4 columns are mandatory argument while last three columns presenting member udl can
be left empty or with zero, but all column must be there.
(c)

Exception                                 Traceback (most recent call last)
c:\Users\Civil Lab\Desktop\tabish\Phd\Git_Repository\StrucPy\Test4.py in line 17
Exception: ('These nodes in boundary condition does not exist: ', [79])
(d)

Exception                                 Traceback (most recent call last)
c:\Users\Civil Lab\Desktop\tabish\Phd\Git_Repository\StrucPy\Test4.py in line 20
Exception: Perform Pre-processing of the structure using method 'preP' before performing analysis.
(e)

```

Figure 2 Screenshot of error handling in the RCFA module.

2.4. Usage

The package's documentation and GitHub repository include comprehensive examples in its examples section, illustrating the structural analysis of different RC members and frames through the RCFA package. An example is illustrated in Listing 1, dealing with class 'RCF' which demonstrates the use of RCF class to perform seismic analysis of a 6-story RC ordinary moment resisting frame (OMRF). The frame is auto-generated and the library repository's 'inputfiles' section contains input files for the given example. In the listing, the model's input files containing the information for auto frame generation, seismic definition, and load combination are imported in lines 3–5. An object representing the RC model with all the model input information is created using the RCF class in line 6. The instance variable 'autoflooring' as True generates floors automatically on every story of the RC frame during the pre-processing. The method 'preP' performs pre-processing to arrange and acquire member's information in line 7. The model can also be edited as per the user requirements using the method 'changeFrame' as shown in lines 12–13. In this example, a few members are deleted from the initial model, and all members are assigned rectangular cross-sections in lines 10–11. Further, a uniformly distributed load of 10 kN/m is applied to all the beam members in the direction of gravity (line 12). The method 'changeFL' allows the user to the change floor's thickness and load on all the stories of the frame at once. In this example, the floors are edited in line 14 with a thickness of 100 mm and a uniformly distributed dead load of 5 kN/m², and a live load of 3 kN/m² is applied to every floor. Uniformly distributed loads on members and all floor loads except the live load are considered dead loads in this package. Structural analysis of the generated RC frame is performed in line 15 using the method 'RCanalysis'. Further, various other information can be extracted from the model after performing the structural analysis as shown in lines 16–19.

The variable 'reactions' represents the base joint forces of the RC frame structure, while the variable 'ndisp' contains the joint displacements of the frame. The variation of shear forces and bending moments along the length of the member is obtained in the variable 'sfbmd'. The property of the material used in the structural analysis is obtained through the method 'Mproperties'. The initial RC model and edited model are presented in Figure 3. All the generated plots, interactive visualizations, and output of the presented example is available in the package's repository. Complete examples along with all interactive visualization and plots, and output in Excel are available in the repository. The presented example can be executed on any integrated development environment (IDE) for Python, but Jupiter Notebook is preferred for better visualization. The model can be visualized at any time after pre-processing in order to check the nodes, members, and floor IDs. The visualization of the model allows the selection of single components interactively for better understanding. The initially generated RC model using 'framegen' and the edited model used in the example are presented in Figure 3. The initial frame was generated automatically, and then the members and loads of the generated frame were edited using the method 'changeFrame'.

2.5. Validation

This section of the paper presents and validates the results obtained from the StrucPy.RCFA package. It demonstrates the accuracy and reliability of the RCFA for structural analysis and provides insights into the compatibility of the results with industry-standard software like STAAD PRO connect edition, a widely used commercial structural analysis software. The RCFA module of the StrucPy library and STAAD PRO are employed to analyze the structure presented in the example of this paper. All the details of the model such as the structure's geometry, material properties, loading conditions, boundary conditions, and

```

1  from StrucPy.RCFA import RCF
2  import pandas as pd
3  framegen = pd.read_excel('./InputFiles/Example1.xlsx', 'framegen', header=0, index_col=0)
4  seismic_definition = pd.read_excel('./InputFiles/Example1.xlsx', 'Seismic_Definition', header=0, index_col=0)
5  load_combo = pd.read_excel('./InputFiles/Example1.xlsx', 'load_combinations', header=0, index_col=0)
6  r1 = RCF(nodes_details=None, member_details=None, boundarycondition= None, framegen=framegen, seismic_def=seismic_definition,
7  load_combo=load_combo, autoflooring=True) # Creating RC frame object for analysis
8  r1.preP() # Pre-processing the model
9  model = r1.model3D() # Initial model in 3D
10 mem_list = [121,122,123,125,126,127,129,130,131,111,112,113,114,115,116,138,144,150,156,162,168,174,180,186,99, 100,103,104, 107,
108,89,90, 91,92,137,143,149,155,161,167,67,68,77,81,85,136,148,142]
11 r1.changeFrame(member= mem_list, delete=True) # Deleting few members
12 r1.changeFrame(member='all', width=400, depth=400) #Changing width and depth of all members to 400 X 400
13 r1.changeFrame(member='beam', yudl=-10) #Changing udl of -10kN/m in y-direction on all members
14 r1.preP() # Pre-processing is required after changing the model
15 r1.changeFL(thickness=100, LL=-3, FF=-5, WP=0) #Changing the floor thickness and loads
16 r1.RCanalysis() # Performing Analysis
17 Reactions = r1.reactions() # Getting Reactions
18 Ndisp = r1.Gdisp() # Getting nodal displacement
19 Sfbmd = r1.sfbmd(47) # Getting Shear Force and Bending Moment Diagram of Member with ID- 47
20 material_properties = r1.Mproperties() # Getting Material Properties

```

Listing 1.

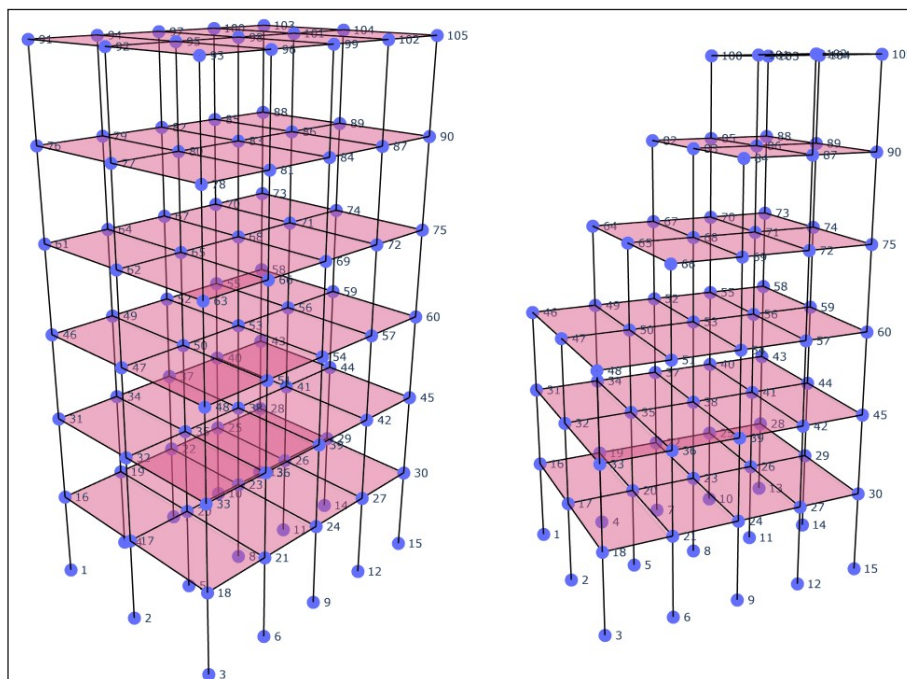


Figure 3 Initial 3D Generated RC Model (Left) and Edited 3D RC Model in RCFA module (Right).

	TIME PERIOD (sec)	SEISMIC ACCELERATION (Ah)	SEISMIC WEIGHT (kN)	BASE SHEAR (kN)
RCFA	0.9613958	0.0565844	14976.00	847.4079
STAAD Pro	0.9614000	0.0566000	14976.00	847.4100

Table 4 Comparison of seismic calculation from RCFA and STAAD Pro.

seismic definition were kept similar. Table 4 shows the comparison of the seismic calculation by RCFA and STAAD Pro respectively. The values are found to be the same and the only difference is that the RCFA package does not round off the values until the final analysis is performed while STAAD Pro rounds off all the values as evaluated.

The comparison of the seismic story shear by RCFA and STAAD Pro is presented in Table 5. In comparing the results of seismic calculations and seismic story shear between the RCFA Python package and STAAD Pro, it is observed that the values are perfectly colinear to each other. The RCFA package does not round off the values until the final analysis is performed. This approach maintains the precision of the calculations throughout the analysis, ensuring that no significant figures are lost prematurely. On the other hand, STAAD Pro rounds off all the values as they are evaluated, which can lead to minor differences in the final results compared to the RCFA package.

This distinction in rounding off values has little impact on the overall accuracy of the RC-framed structure. The RCFA package’s approach of preserving precision until the final analysis results in a slight variation in story forces and displacement in the applied seismic direction compared to STAAD Pro. The difference in the values of the results is classified as an error. To quantify this error,

FLOOR	SEISMIC STORY SHEAR (kN)	
	RCFA RESULT	STAAD PRO RESULT
1	17.14555	17.146
2	68.58222	68.582
3	152.8865	152.886
4	208.0243	208.023
5	225.3913	225.390
6	175.3781	175.379

Table 5 Comparison of seismic story shear from RCFA and STAAD Pro.

the base reactions, maximum member force, nodal displacements, and deflection in local coordinates were calculated for all relevant nodes and members. The chart representing the distribution and variation of errors is depicted in Figures 4 and 5 for forces and displacements respectively. Despite the differences, the maximum error identified is 4 kN in the y-direction force, with an overall percentage error below 0.01% in Figure 4. This suggests that while there are discrepancies between the RCFA packages and STAAD PRO, they are relatively minor in the context of the structural analysis, and the overall

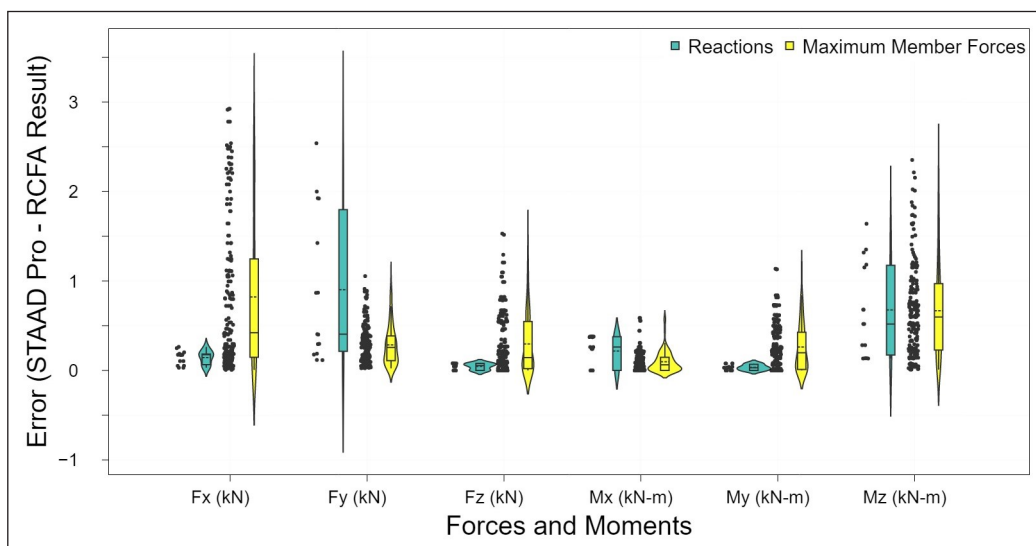


Figure 4 Absolute error distribution of base reactions and maximum member forces.

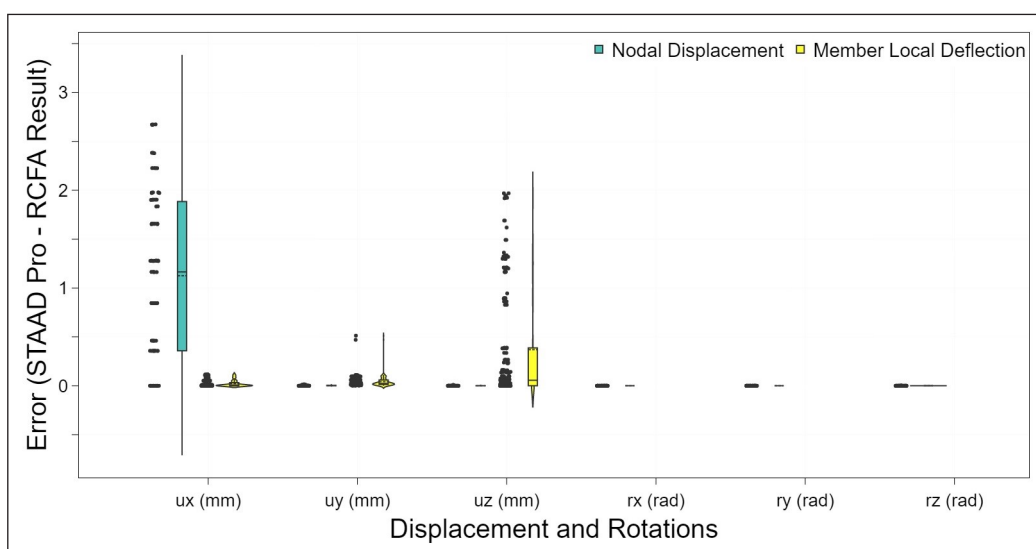


Figure 5 Absolute error distribution of nodal displacements and maximum member deflections.

accuracy of the RC-framed structure is not significantly affected. Additionally, the maximum difference in nodal displacement was found to be 3.6 mm in the direction of the seismic force in Figure 5. These discrepancies are attributed to the rounding off of values in the calculation process. However, it is noted that these differences are minimal when compared to the original values and have almost zero effect on the behaviour of the RC frame. Overall visualization of errors in the figures provides a clear understanding of how the errors are distributed throughout the structure and helps in identifying patterns. Overall, while there are differences in the results between the RCFA package and STAAD Pro due to their respective approaches, these variances have negligible effects on the structural behaviour of the RC frame.

RCFA is also capable of generating the shear force and the bending moment diagram in local and global coordinate systems for all the members of the frames. RCFA can generate shear force and bending moment

diagrams in all three directions. Figure 6 showcases the shear force diagram in the y-direction and the bending moment diagram in the z-direction, as generated by RCFA for member 47 from a given example. It is noted that these diagrams closely resemble those generated by STAAD Pro, displaying an almost negligible variation. These diagrams are produced utilizing the Plotly library, which allows for saving in various formats offered by Plotly and renders them interactive when saved with an HTML extension. In addition to the graphical representation, RCFA furnishes data regarding the variation of shear force and bending moment values along the length of the member, with a precision of 0.01 m. This feature provides students and researchers with detailed insights into structural forces, aiding in the efficient analysis and design of reinforced concrete elements and frames.

RCFA offers the functionality to generate the deflected shape of frame members in both local and global coordinate systems, providing engineers with comprehensive insights

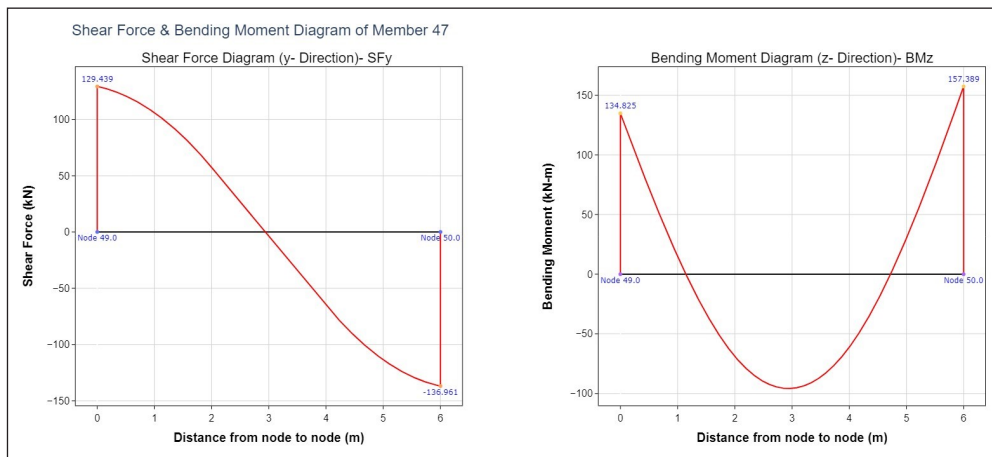


Figure 6 Shear force and bending moment diagram of member 47 from example.

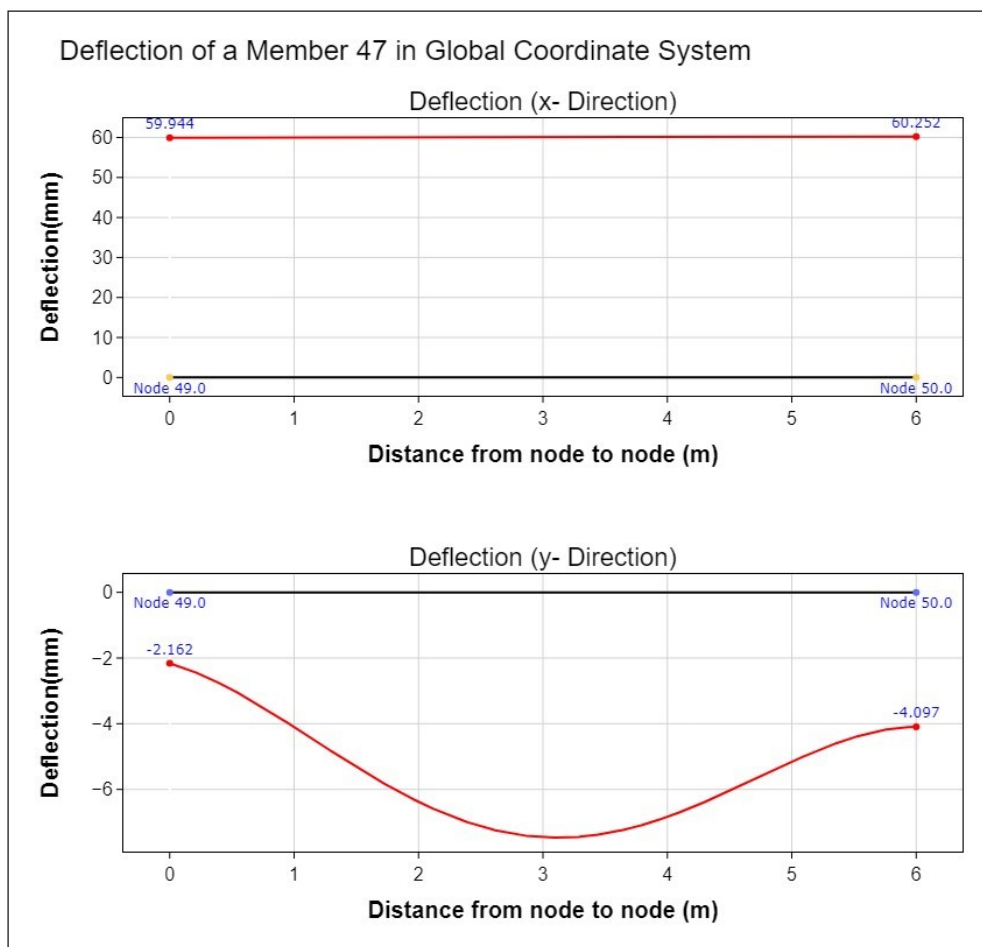


Figure 7 The deflected shape of member 47 from Example.

into structural behaviour. Figure 7 illustrates the deflected shape in the x and y directions for member 47 in the global coordinate system. Notably, these diagrams closely resemble those produced by STAAD Pro, exhibiting minimal variation. RCFA’s capability extends beyond just plotting deflected shapes and diagrams; it also offers access to precise data on deflection in every direction for all members in local and global coordinate systems with the least count of 0.01 m along the length of the members. Furthermore, this deflection data provides engineers with the necessary information to ensure that structural elements meet safety and performance requirements.

2.6. Scalability

Scalability analysis is a crucial aspect of any software package, especially in the field of structural analysis. It helps in determining, how well a software package performs as the size of the problem increases and how sensitive results are to the changes in large-scale RC models. In the case of the RCFA package, scalability is evaluated by analyzing four different structures: a single RC member, an 8-member RC frame, a 64-member RC frame, and a 1070-member RC frame. The error, along with computational time, was assessed for each structure, and the variation in the error was plotted in Figure 8 to

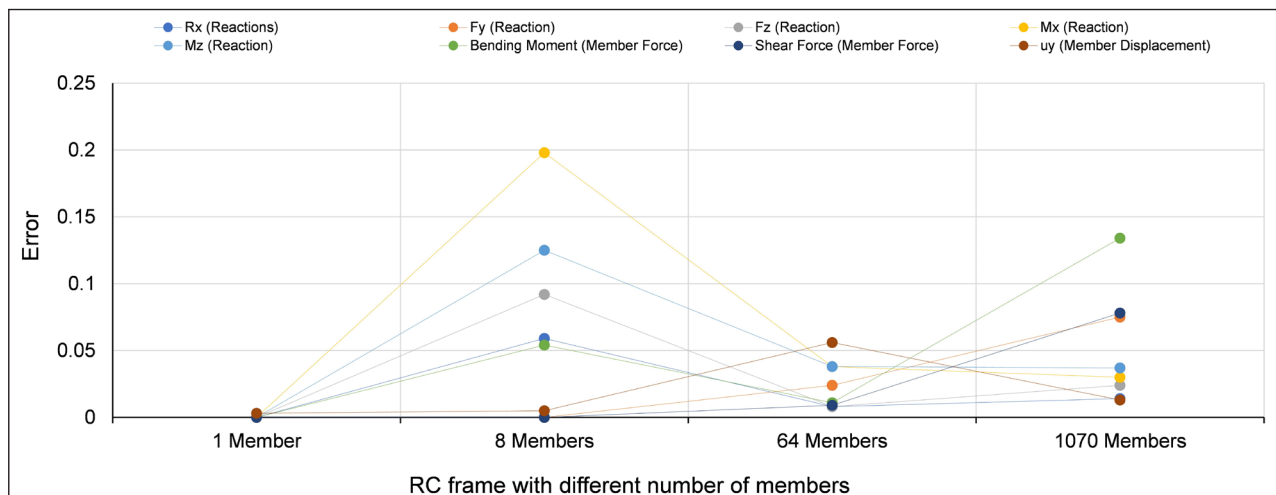


Figure 8 Error in the results with increasing complexity of RC Frame.

evaluate the sensitivity to scalability of the RCFA package. It is noticed that the error variation is below 0.2, which is very low with the scalability of the RC frame structure. The RCFA package is found to exhibit low sensitivity with the increasing complexity of the RC frame structure.

For the single RC member, the analysis serves as a baseline for comparison with more complex structures. This simple structure allows for a quick evaluation of the RCFA package's performance. The computational time for the structural analysis of a single member was found to be 5.022 seconds. The error and computational time are expected to increase compared to the single member, but the rate of increase is a critical factor in determining scalability. If the error and computational time increase significantly faster than the size of the structure, it could indicate a scalability issue. However, if the increase is relatively consistent with the increase in structure size, it suggests good scalability. The RC frame with 8 members was analyzed by RCFA in 5.36 seconds, the computational time was similar to that of a single member with very little error in the result. Similarly, RCFA completed the structural analysis of the RC frame with 64 members in 9.92 seconds with no increase in error. Next, the 1070-member RC frame provides a more challenging test for the RCFA package's scalability. This structure is much larger and more complex than the previous ones, the number of interactions and calculations also increases exponentially. The RCFA package efficiently handles these calculations to maintain a reasonable computational time of 84.26 seconds while minimizing the errors. In conclusion, this section of the paper ensures that the RCFA package can reliably analyze structures of all sizes and provide accurate results for engineering analysis.

2.7. Technical Constraints and Extendibility

The StrucPy package, which currently focuses on structural analysis of reinforced concrete framed structures via the RCFA module, has few technical constraints that shape its current capabilities and future development.

These constraints are crucial for users to understand when considering its application in their projects. Firstly, StrucPy.RCFA's current design assumes that floors are surrounded by beams on all sides. This design constraint is essential for effectively transferring loads to the columns and ensuring structural stability. However, it limits the applicability of the package to structures that adhere to this specific framing configuration. Future versions may address this constraint to support a wider range of structural configurations such as cantilever floors. Secondly, StrucPy lacks certain analysis features such as P-delta analysis, dynamic seismic analysis, and wind load considerations. These features can boost the capability for assessing the behaviour of structures under various other loading conditions. While these capabilities are absent in the current version, it is being actively worked on to incorporate them in the next version. Despite these constraints, StrucPy's modular and object-oriented design makes it relatively easy to extend and modify. This flexibility enables users to customize the package to suit their specific needs and integrate it with other Python libraries seamlessly. Furthermore, the active development and planned enhancements indicate that StrucPy is continuously improving to meet the needs of structural engineers, students, and researchers.

The object-oriented nature of the StrucPy.RCFA module offers a flexible and modular approach to structural analysis, making it easier to extend or modify its functionality. The modular structure of the RCFA allows you to modify existing classes or methods to improve performance, accuracy, or usability. For instance, users can optimize algorithms for better computational efficiency or enhance error handling to improve the robustness of the package. Currently, RCFA allows input to be very flexible to consider various loading and support conditions. The user can further extend the RCFA module by adding new classes or methods to handle additional types of structural elements and post-processing functionalities. Further, the RCFA module can be easily

integrated with other Python libraries to leverage their capabilities. StrucPy.RCFA has been successfully used to retrieve beam force data and to collaborate with package PyRCFD for the design optimization of RC beam [55].

3. QUALITY CONTROL

StrucPy comes with a large test suite that covers most of its functionality. Tests are implemented using the Python library `pytest` [56]. Tests are also included that compare StrucPy's results with other software such as STAAD PRO. Tests are implemented for every class with unit tests and integration tests with coverage over 80%. Tests check whether the model in StrucPy can be processed and solved for a set of default inputs. Users can and do report bugs by raising issues in the GitHub repository. The main StrucPy repository contains an examples folder that provides useful examples on how to use StrucPy for different tasks and using various parameters. In order to run the test all dependency packages along with 'pytest' have to be installed. In order to check the coverage of the tests, the installation of 'pytest-cov' [57] is also required. The tests can be performed by running the command:

```
pytest
```

To check the coverage, the '.coveragerc' file must be added to the directory to avoid considering unwanted files in the test report, which is available on the StrucPy repository. Coverage can be obtained by running the command:

```
pytest --cov
```

All the unit and integration tests are being automatically run via GitHub Actions. At the time of writing, the StrucPy tests run on Ubuntu, macOS, and Windows systems with Python 3.9–3.11. Further details are available on `.github/workflows/Tests.yml` on the StrucPy repository. Tests and documentation badges are also available on the README.md, which will automatically redirect to tests and documentation.

(2) AVAILABILITY

OPERATING SYSTEM

All the operating systems capable of running Python.

PROGRAMMING LANGUAGE

Python 3.9 or higher.

ADDITIONAL SYSTEM REQUIREMENTS

No extra requirement.

DEPENDENCIES

The current version of StrucPy.RCFA uses:

- `numpy` == 1.23.3
- `pandas` == 1.4.4
- `plotly` == 5.10.0
- `ray` == 2.6.1
- `openpyxl` == 3.0.10

SOFTWARE LOCATION

Archive

Name: Zenodo

Persistent identifier: <https://doi.org/10.5281/zenodo.12592773>

Licence: GNU LGPLv2.1

Publisher: Tabish Izhar

Version published: 0.0.2

Date published: 29/06/24

Code repository

Name: Github

Identifier: <https://github.com/TabishIzhar/StrucPy.git>

Licence: GNU LGPLv2.1

Date published: 11/08/23

LANGUAGE

English

(3) REUSE POTENTIAL

The StrucPy.RCFA module provides a comprehensive open-source toolset for structural analysis of RC framed structures. With its capabilities ranging from 3D model generation to providing in-depth structural analysis data, the package stands as a promising tool for civil engineering students, researchers, and professionals. Its user-friendly approach, combined with the power of its automated functions and methods, makes it an easy tool for structural analysis of RC members/frames. The RCFA package offers a simple solution to civil engineering students in developing countries. All the plots and visualization generated by RCFA can be saved in various formats offered by plotly. RCFA can enhance the understanding of engineering students towards the behaviour of framed buildings under various load combinations. It removes the steep programming learning curves and process of taking bulky inputs using automated functions. RCFA package provides a compact open-source solution to attain sustainable development goal 4 (SDG4) of equal access to quality education especially in developing countries where institutes cannot afford to subscribe to commercial licenses. This can help students fill the skill gap surrounding the visualization and understanding of RC frame behaviour.

StrucPy's RCFA module can provide a platform to attain generalized realistic design optimization of complex RC frames by generating analysis data and communicating seamlessly with optimization algorithms. RCFA has been

used to retrieve various beam forces for design optimization of RC beams under market constraints by Izhar et al. [55], where RCFA was used as structural analysis software to perform continuous reanalysis of optimal design sections. Further, it classifies the columns into different types based on effective length, making column design less cumbersome. This functionality enables users to utilize the generated data by StrucPy for more effective, accurate, and realistic design of column members, thus opening up new possibilities in design optimization through the consideration of detailing and buildability.

DATA ACCESSIBILITY STATEMENT

The presented package does not utilize any external data. All data supporting the findings of this study are freely accessible in the package repository. The “Examples” folder includes all scripts featured in the online documentation, along with the example provided. The “InputFiles” folder contains all input data, while the “OutputFiles” folder holds the outputs for the presented example.

ACKNOWLEDGEMENTS

The authors are grateful to all individuals who contributed to RCFA in any way possible and to Integral University, Lucknow for providing the facilities to conduct this study. Manuscript Communication Number: IU/R&D/2023-MCN0001945.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR CONTRIBUTIONS

TI developed the entire software package, formulated the methodology, created the online documentation, and drafted the paper. SAA guided problem formulations and verified the results. NM tested the software and cross-checked the results. TA proofread the paper and assisted in developing the online documentation.

AUTHOR AFFILIATIONS

Tabish Izhar  orcid.org/0000-0002-8082-5373

Department of Civil Engineering, Integral University, Kursi Road, Lucknow-226020, India

Syed Aqeel Ahmad  orcid.org/0000-0002-0769-5226

Department of Civil Engineering, Integral University, Kursi Road, Lucknow-226020, India

Tasneem Ahmed  orcid.org/0000-0003-2702-3168

Department of Computer Application, Integral University, Kursi Road, Lucknow-226020, India

Neha Mumtaz  orcid.org/0000-0003-0532-9107

Department of Civil Engineering, Integral University, Kursi Road, Lucknow-226020, India

REFERENCES

1. **Yuan XF, Teng JG.** Interactive Web-Based Package for Computer-Aided Learning of Structural Behavior. *Comput. Appl. Eng. Educ.*, 2002; 10(3): 121–136. DOI: <https://doi.org/10.1002/cae.10020>
2. **Tappin S.** The Early Use of Reinforced Concrete in India. *Constr. Hist.*, 2002; 18: 79–98.
3. **Ashur S, Alhassan M, Chen D, Gunawardena S.** Integrating Professional Software in Undergraduate Civil Engineering Education. In *Conference: The 4th International Multi-Conference on Engineering and Technological Innovation: IMETI*, Orlando, Florida, USA; 2011. URL: https://www.iiis.org/CDs2011/CD2011SCI/IMETI_2011/PapersPdf/FA819CR.pdf.
4. **Verduzco LF, Horta J, Hernández MAPLy, Hernández JB.** CALRECOD — A software for Computed Aided Learning of Reinforced Concrete structural Design. *Adv. Eng. Softw.*, Oct. 2022; 172: 103189. DOI: <https://doi.org/10.1016/j.advengsoft.2022.103189>
5. **Tilak JBG.** *Economics of Engineering Education in India*. 1st Edition. Routledge; 2023. DOI: <https://doi.org/10.4324/9781003430223>
6. **Bowman DA,** et al. Virtual-SAP: an immersive tool for visualizing the response of building structures to environmental conditions. In *IEEE Virtual Reality*, Los Angeles, CA, USA: IEEE Computer Society. 2003; 243–250. DOI: <https://doi.org/10.1109/VR.2003.1191146>
7. **Shirazi A, Behzadan AH.** Design and Assessment of a Mobile Augmented Reality-Based Information Delivery Tool for Construction and Civil Engineering Curriculum. *J. Prof. Issues Eng. Educ. Pract.*, 2015; 141(3). DOI: [https://doi.org/10.1061/\(ASCE\)EI.1943-5541.0000229](https://doi.org/10.1061/(ASCE)EI.1943-5541.0000229)
8. **Turkan Y, Radkowski R, Karabulut-Ilgu A, Behzadan AH, Chen A.** Mobile augmented reality for teaching structural analysis. *Adv. Eng. Informatics*, 2017; 34, 90–100. DOI: <https://doi.org/10.1016/j.aei.2017.09.005>
9. **Mtenga PV, Spainhour LK.** Applications of Mathematical Software Packages in Structural Engineering Education and Practice. *J. Comput. Civ. Eng.*, 2000; 14(4): 273–278. DOI: [https://doi.org/10.1061/\(ASCE\)0887-3801\(2000\)14:4\(273\)](https://doi.org/10.1061/(ASCE)0887-3801(2000)14:4(273))
10. **Ahmad SA, Izhar T, Ahmed T, Mumtaz N.** Transformative role of machine learning in design optimization of reinforced concrete frames. *Int. J. Adv. Technol. Eng. Explor.*, 2024; 11(112): 437–458. DOI: <https://doi.org/10.19101/IJATEE.2023.10102336>
11. **Mangal M, Li M, Gan VJL, Cheng JCP.** Automated clash-free optimization of steel reinforcement in RC frame structures

- using building information modeling and two-stage genetic algorithm. *Autom. Constr.*, 2021; 126: 103676. DOI: <https://doi.org/10.1016/j.autcon.2021.103676>
12. **Arayici Y, Fernando T, Munoz V, Bassanino M.** Interoperability specification development for integrated BIM use in performance based design. *Autom. Constr.*, 2018; 85(October 2017): 167–181. DOI: <https://doi.org/10.1016/j.autcon.2017.10.018>
 13. **Negrin IA, Chagoyén EL.** Economic and environmental design optimisation of reinforced concrete frame buildings: A comparative study. *Structures*, 2022; 38: 64–75. DOI: <https://doi.org/10.1016/j.istruc.2022.01.090>
 14. **Mergos PE.** Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. *J. Build. Eng.*, 2021; 44: 102935. DOI: <https://doi.org/10.1016/j.job.2021.102935>
 15. **Aslay SE, Dede T.** 3D cost optimization of 3 story RC constructional building using Jaya algorithm. *Structures*, 2022; 40: 803–811. DOI: <https://doi.org/10.1016/j.istruc.2022.04.055>
 16. **Calledda C, Montisci A, Porcu MC.** Optimal design of earthquake-resistant buildings based on neural network inversion. *Appl. Sci.*, 2021; 11(10): 1–14. DOI: <https://doi.org/10.3390/app11104654>
 17. **Elhegazy H, Ebid AM, Mahdi IM, Aboul Haggag SY, Rashid IA.** Selecting optimum structural system for R.C. multi-story buildings considering direct cost. *Structures*, 2020; 24: 296–303. DOI: <https://doi.org/10.1016/j.istruc.2020.01.039>
 18. **Li Y, Duan RB, Li QS, Li YG, Huang X.** Wind-resistant optimal design of tall buildings based on improved genetic algorithm. *Structures*, 2020; 27: 2182–2191. DOI: <https://doi.org/10.1016/j.istruc.2020.08.036>
 19. **Wang S, Ma F.** Reinforced concrete frame structure based on ANSYS optimization analysis and material properties. *Adv. Mater. Res.*, 2013; 788: 571–574. DOI: <https://doi.org/10.4028/www.scientific.net/AMR.788.571>
 20. **Dehnavipour H, Mehrabani M, Fakhriyat A, Jakubczyk-Gałączyńska A.** Optimization-based design of 3d reinforced concrete structures. *J. Soft Comput. Civ. Eng.*, 2019; 3(3): 95–106. DOI: <https://doi.org/10.22115/scce.2020.211509.1145>
 21. **Park HS, Hwang JW, Oh BK.** Integrated analysis model for assessing CO₂ emissions, seismic performance, and costs of buildings through performance-based optimal seismic design with sustainability. *Energy Build.*, 2018; 158: 761–775. DOI: <https://doi.org/10.1016/j.enbuild.2017.10.070>
 22. **Chen C, et al.** Optimum cost design of frames using genetic algorithms. *Steel Compos. Struct.*, 2019; 30(3): 293–304. DOI: <https://doi.org/10.12989/scs.2019.30.3.293>
 23. **McKenna F, Scott MH, Fenves GL.** Nonlinear Finite-Element Analysis Software Architecture Using Object Composition. *J. Comput. Civ. Eng.*, 2010; 24(1): 95–107. DOI: [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000002](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000002)
 24. **Srivastava DK, Sharma A.** Open Source Software (OSS): A new Venue for Civil Engineering Professionals. In *Proceeding of National Conference on Recent Trends and Innovation Civil Engineering*. 2013; 266–269.
 25. **Tsvetkova T, Kerenchev N.** Review of open source software for modelling and analysis of structures. In *IOP Conference Series: Materials Science and Engineering*, 2020; 951: 012003. DOI: <https://doi.org/10.1088/1757-899X/951/1/012003>
 26. **Abudayyeh O, Cai H, Fenves SJ, Law K, Robert O, Rasdorf W.** Assessment of the Computing Component of Civil Engineering Education. *J. Comput. Civ. Eng.*, 2004; 18: 187–195. DOI: [https://doi.org/10.1061/\(ASCE\)0887-3801\(2004\)18:3\(187\)](https://doi.org/10.1061/(ASCE)0887-3801(2004)18:3(187))
 27. **van Rossum G.** *The Python Language Reference*, Release 3.10.12. Python Software Foundation.
 28. **Pérez F, Granger BE, Hunter JD.** Python: An ecosystem for scientific computing. *Comput. Sci. Eng.*, 2011; 13(2): 13–21. DOI: <https://doi.org/10.1109/MCSE.2010.119>
 29. **Cao Q, Lim L, Dale V, Tasler N.** Experiences in python programming laboratory for civil engineering students with online collaborative programming platform. In *14th annual International Conference of Education, Research and Innovation: ICERI2021 Proceedings*. 2021; 5784–5791. DOI: <https://doi.org/10.21125/iceri.2021.1305>
 30. **Vink R, Smith B.** anaStruct. 2023. <https://github.com/ritchie46/anaStruct.git> (accessed Aug. 19, 2023).
 31. **Andersen K, Kristiansen H.** adapy. 2023. <https://github.com/Krande/adapy.git> (accessed Aug. 21, 2023).
 32. **van Leeuwen R.** feastruct.pdf. 2023. <https://github.com/robbievanleeuwen/feastruct.git> (accessed Aug. 23, 2023).
 33. **Santos PJDL.** nusa. 2023. <https://github.com/JorgeDeLosSantos/nusa.git> (accessed Aug. 23, 2023).
 34. **Zhu M, McKenna F, Scott MH.** OpenSeesPy: Python library for the OpenSees finite element framework. *SoftwareX*, 2018; 7: 6–11. DOI: <https://doi.org/10.1016/j.softx.2017.10.009>
 35. **Barter G, Ning A.** pyFrame3DD. <https://github.com/WISDEM/pyFrame3DD.git> (accessed Aug. 25, 2023).
 36. **Craig, Ferster C, Franklin FJ.** PyNite. <https://github.com/JWock82/PyNite.git> (accessed Aug. 29, 2023).
 37. **Chevalier B.** StructPy. <https://github.com/BrianChevalier/StructPy.git> (accessed Sep. 05, 2023).
 38. **Kassab H.** structural_analysis. https://github.com/Hazem-Kassab/structural_analysis.git (accessed Sep. 05, 2023).
 39. **Khan S.** Structural-Python. <https://github.com/shakirkhan1/Structural-Python.git> (accessed Sep. 05, 2023).
 40. **Anwar N, Kanok-Nukulchai W, Batanov DN.** Component-Based, Information Oriented Structural Engineering Applications. *J. Comput. Civ. Eng.*, 2005; 19(1): 45–57. DOI: [https://doi.org/10.1061/\(ASCE\)0887-3801\(2005\)19:1\(45\)](https://doi.org/10.1061/(ASCE)0887-3801(2005)19:1(45))
 41. **Okereke M, Keates S.** Direct Stiffness Method. In *Finite Element Applications: A Practical Guide to the FEM Process*. Cham: Springer; 2018. 47–106. DOI: https://doi.org/10.1007/978-3-319-67125-3_3
 42. **Murín J, Kutiš V, Královič V, Sedlár T.** 3D beam finite element including nonuniform torsion. *Procedia Eng.*, 2012; 48: 436–444. DOI: <https://doi.org/10.1016/j.proeng.2012.09.537>
 43. **Ferreira AJM.** *MATLAB Codes for Finite Element Analysis: Solids and Structures (Solid Mechanics and Its Applications)*, 1st ed. 2008. DOI: <https://doi.org/10.1007/978-1-4020-9200-8>

44. **Papadrakakis M, Sapountzakis EJ.** Introduction to Matrix Methods of Structural Analysis. In *Matrix Methods for Advanced Structural Analysis*. Butterworth-Heinemann, 2018; 1–15. DOI: <https://doi.org/10.1016/b978-0-12-811708-8.00001-5>
45. **Papadrakakis M, Sapountzakis EJ.** Rigid Joints. In *Matrix Methods for Advanced Structural Analysis*. Butterworth-Heinemann, 2018; 227–256. DOI: <https://doi.org/10.1016/b978-0-12-811708-8.00006-4>
46. **Moczo P, Kristek J, Halada L.** *The Finite-Difference Method for Seismologists: An Introduction*. Comenius University, Bratislava; 2004. URL: <https://www.nuquake.eu/Publications/FDM-Intro-SPICE.pdf>.
47. **Fukuchi T.** Finite difference method and algebraic polynomial interpolation for numerically solving Poisson's equation over arbitrary domains. *AIP Adv.*, 2014; 4(060701). DOI: <https://doi.org/10.1063/1.4885555>
48. **(BIS) Bureau Indian Standards.** IS 1893-Part-1-(2016): Criteria for Earthquake resistant design of structures, Part 1: General Provisions and buildings (Sixth Revision). *BIS, New Delhi*, 2016.
49. **(BIS) Bureau Indian Standards.** IS 456: 2000 - Plain Concrete and Reinforced- Code of Practice (fourth revision). *BIS, New Delhi*, 2000.
50. **WOOD RH.** Effective lengths columns in multi-storey buildings. *Struct. Eng.*, 1974; 52(7): 235–244.
51. **Harris CR,** et al. Array programming with NumPy. *Nature*, 2020; 585(7825): 357–362. DOI: <https://doi.org/10.1038/s41586-020-2649-2>
52. **McKinney W.** Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference (SCIPY)*, 2010; 56–61. DOI: <https://doi.org/10.25080/majora-92bf1922-00a>
53. **Plotly Technologies Inc.** *Collaborative data science*. Montreal, QC: Plotly Technologies Inc., 2015. [Online]. Available: <https://plot.ly>.
54. **Moritz P,** et al. Ray: A distributed framework for emerging AI applications. In *13th USENIX Symposium on Operating Systems Design and Implementation*, 2018; 561–577. DOI: <https://doi.org/10.48550/arXiv.1712.05889>
55. **Izhar T, Ahmad SA, Mumtaz N.** Multiobjective design optimization of reinforced concrete beam coupled with market practice based constructability function using simulated annealing. *Asian J. Civ. Eng.*, 2024; 1–14. DOI: <https://doi.org/10.1007/s42107-024-01019-7>
56. **Krekel H.** Pytest. 2017. [Online]. Available: <https://docs.pytest.org/>.
57. **Lawley R, Mills J, Krekel H.** pyetst-cov. 2018. [Online]. Available: <https://docs.pytest.org/>.

TO CITE THIS ARTICLE:

Izhar T, Ahmad SA, Ahmed T, Mumtaz N 2024 StrucPy.RCFA- Object-Oriented Python Package for Structural Analysis of Reinforced Concrete Frames. *Journal of Open Research Software*, 12: 10. DOI: <https://doi.org/10.5334/jors.489>

Submitted: 03 November 2023 **Accepted:** 03 July 2024 **Published:** 24 July 2024

COPYRIGHT:

© 2024 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.