JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DESIGNING A G+2 STRUCTURE USING PYTHON WITH GRAPHICAL USER INTERFACE

[1]Abhishek S. Killedar, [2]Laxman G. Kalurkar

[1] Student M.Tech Structural Engineering, Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India, 431001
[2] Professor M.Tech Structural Engineering, Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India, 431001

Abstract : Python is a programming language that may be used for a variety of tasks, including repetitive, laborious, and complex tasks. In this study, I sought to make things easier by writing a set of tools in Python to build one-way slabs, two-way slabs, beams, columns, and foundations. These tools will assist us in determining the required reinforcement for the structural element we are developing, as well as reducing human error and delivering results promptly. The goal of this study is to assist a person in designing a structure without prior understanding of structural analysis software such as STAAD.Pro, ETABS, SAFE, and others. The programming is done through a user interface rather than a script, because we don't have to start over from the beginning if we want to modify a parameter; instead, we may edit the parameters and instantly see the results for the altered parameter. The Code have been developed using IS456:2000, SP-16 and IS875-Part 2.

***Keywords* – Python, Structural Analysis, Structural Design, Reinforced Concrete Structures, GUI Programming.**

I. INTRODUCTION

Civil engineers are not even introduced to Python throughout their college years, but as they go towards structure design, they are exposed to a wider range of design tools. If you were lucky enough to be taught Staad-pro and Auto-Cad during your engineering years, you'll have a leg up on the competition if you apply for design positions. Now, when you develop bonding with these software's, you'll see that the majority of them were created by Civil Engineers rather than computer scientists. That's not unexpected, given that learning to code doesn't necessitate a software engineering degree. At any age, anyone can begin learning Python, C++, or Java. A logical mind can work wonders with programming languages, and who knows, you could be the next big thing. In this study, we will use Python to automate a daily operation that we perform repeatedly. This will save time, enhance accuracy, and remove human mistake.

II. PYTHON LIBRARIES FOR CIVIL ENGINEERING

Python has a large standard library, and because it is an open source language, many valuable libraries for data science have been built. The Python standard library's import function provides the freedom to use various libraries in programming. Some libraries that are particularly beneficial from a Civil Engineering perspective are covered here.

A. NumPy Library

NumPy is a Python external library for performing sophisticated mathematical operations at a low level. The usage of multi-dimensional array objects in NumPy allows it to overcome slower execution times. It features array manipulation capabilities built-in.

Different algorithms can be converted into functions that can be used on arrays. NumPy has a lot of uses that aren't only for NumPy. It is a very diverse library with numerous applications in other fields.

Data Science, Data Analysis, and Machine Learning can all benefit from Numpy. It also serves as a foundation for other Python libraries. These libraries make advantage of NumPy's features to expand their capabilities.

B. SciPy Library

SciPy is a Python open-source library for addressing issues in mathematics, science, engineering, and technology. It gives users the ability to alter and view data using a variety of high-level Python commands. SciPy is based on the NumPy Python extension. "Sigh Pi" is another way of pronouncing SciPy.

C. Matplotlib Library

Matplotlib is a fantastic Python visualization package for 2D array charts. Matplotlib is a multi-platform data visualization package based on NumPy arrays and intended to operate with the SciPy stack as a whole.

One of the most significant advantages of visualization is that it provides us with visual access to massive volumes of data in simply understandable graphics. Matplotlib has a variety of plots such as line, bar, scatter, histogram, and so on.

III. Tkinter Module

Tkinter is a Python tool that allows programmers to construct user interfaces for their programs. In this the programmer can also include buttons, text, entry boxes, and other elements. Tkinter is a Python library that makes it easy to create graphical user interfaces. Tkinter is the only GUI Framework that comes pre-installed with the Python Standard Library. Below are some advantages of Tkinter module

• Another advantage of Tkinter is that it is cross-platform, which means that the same code may run on Windows, Mac OS X, and Linux.

• Tkinter is a compact module.

• It's easy to use.

IV. Objective

1. The study's main goal is to develop a collection of tools that will allow anyone with no prior knowledge of programming or structural analysis software to design a structure that is confined to G + 3.
2. To demonstrate how programming is used in building.
3. Using Python, create a G + 2 structure.
4. To save the results as a text file or.txt file.

V. Implementation of Python Code

The python code for our research is developed in the same way that we would construct any RCC structure manually, i.e., step by step. This makes programming easier because we will be coding according to the steps required. We've also used graphs from SP-16 for moments in the foundation design section, where the required co-ordinates will be calculated and displayed to the user, and the user will have to look in SP-16 for the value that we'll be getting from the co-ordinates provided. However, the programs have reference diagram buttons that can be used to help with the values to be fed on the spot.

VI. Load Transfer Mechanism in Python

Within a building, multiple parts are used to transmit and resist external loads. The load path, or mechanism of load transfer in a building, is defined by these factors. The load path runs from the roof to the foundation, passing through each structural element. Everyone involved in building design and construction needs to appreciate the fundamental relevance of a complete load path.

Load is transferred from the floor and roof slabs to the beams, then to the columns or walls, down to the foundations, and finally to the supporting earth beneath. However, this is the principle, and the flow chart below shows how we implemented it in our Python software.

The flow chart depicts the method that has been designed in order to transmit load from one structural part to another.

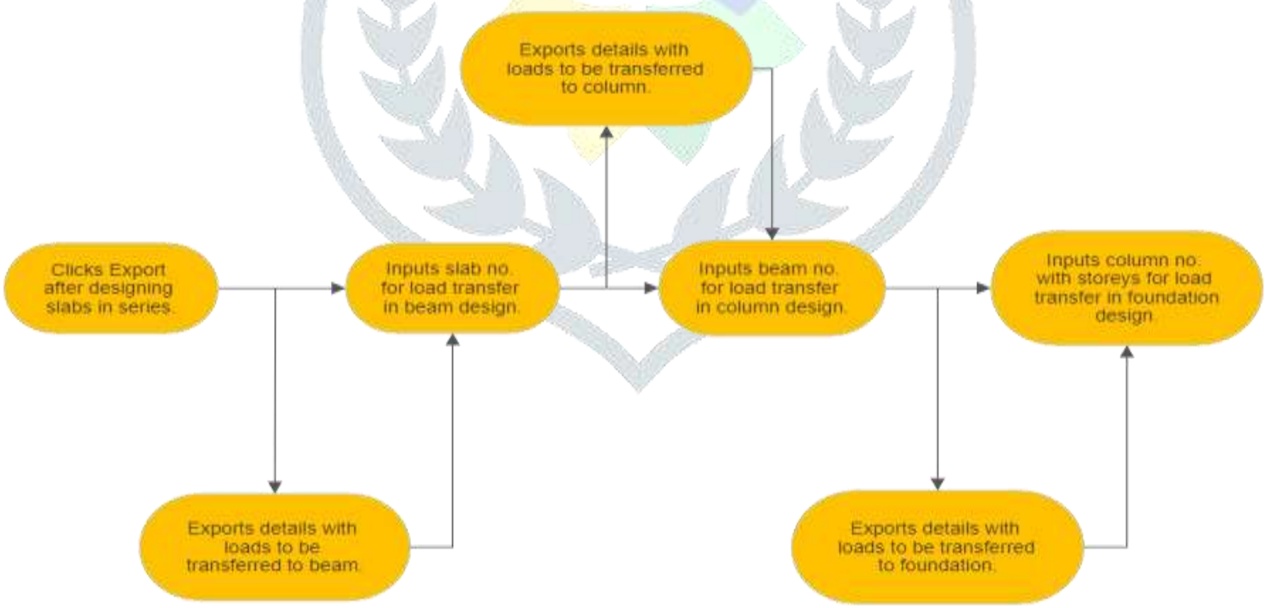

Fig.1 - Load Transfer In Python

VII. Steps to Follow While Using the Program

Below are the steps to be followed in one way slab design however, the steps will remain the same throughout the program with some modifications based on element to be designed.

1. Start with a concrete grade.
2. Fill in the steel grade.
3. Fill in the expected steel percentage.
4. Enter the assumed modification factor using the 'Modification Factor Graph' option, which displays a graph of the modification factor.
5. Use a shorter span (shorter span can also be given data of longer span if you want to distribute load along longer span).

6. Type in a longer span.
7. Fill in the live load, floor finish, and any additional loads that are needed.
8. Decide on the slab type.
9. Fill in the intended bar diameter.
10. Using the button, check effective depth and enter the expected effective depth in the entry field below.
11. Enter the slab cover.
12. Check the slab's entire depth.
13. If necessary, double-check the factor load.
14. In the shorter direction, enter the average depth of the beam.
15. Enter a shorter or longer span for the slab as needed.
16. Enter the width of the first, shorter-span support.
17. Enter the width of the second, shorter-span support.
18. Enter the diameter of the main bar that will be used.
19. Enter the diameter of the distribution bar that will be utilized.
20. Check the new modification factor coordinates and enter the new modification factor in the field below using the same modification graph button.
21. Select Perform Checks to run checks on our slab, such as depth, shear, and so on.
22. To examine the development length, click the development length button.
23. To see all of the reinforcement data required, click the reinforcement button.
24. For the output to be placed under this mark, enter slab marks such as S1, 1, and so on.
25. To export the results, click export.

Fig.2 – One Way Slab GUI

VIII. VALIDATIONS

Our python software was used to assess and design a G+2 building in Jalna, Maharashtra, India. The results of the comparison between hand calculations and automated computer calculations, as well as floor designs, are shown below.

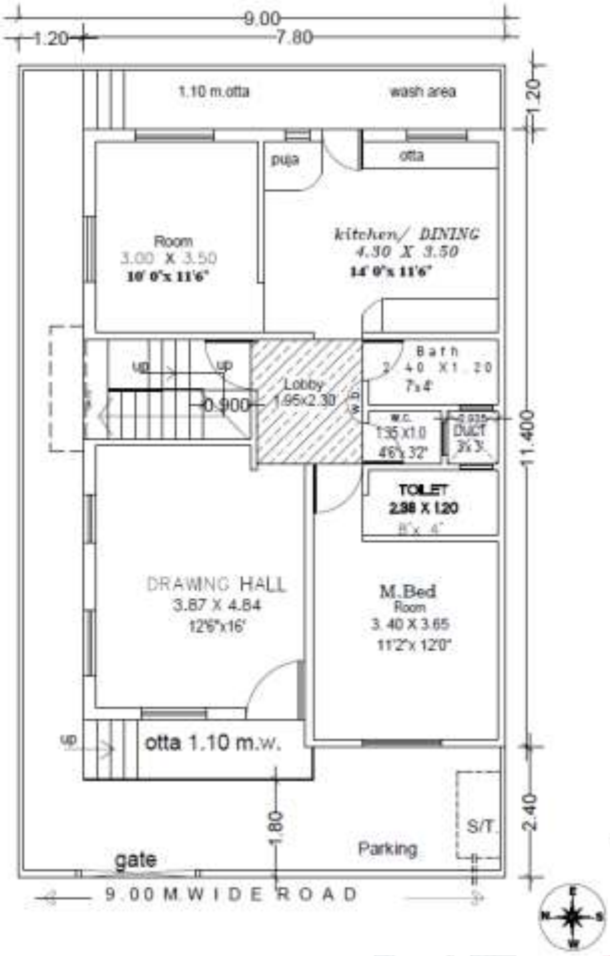

Fig.3 – Ground Floor Plan

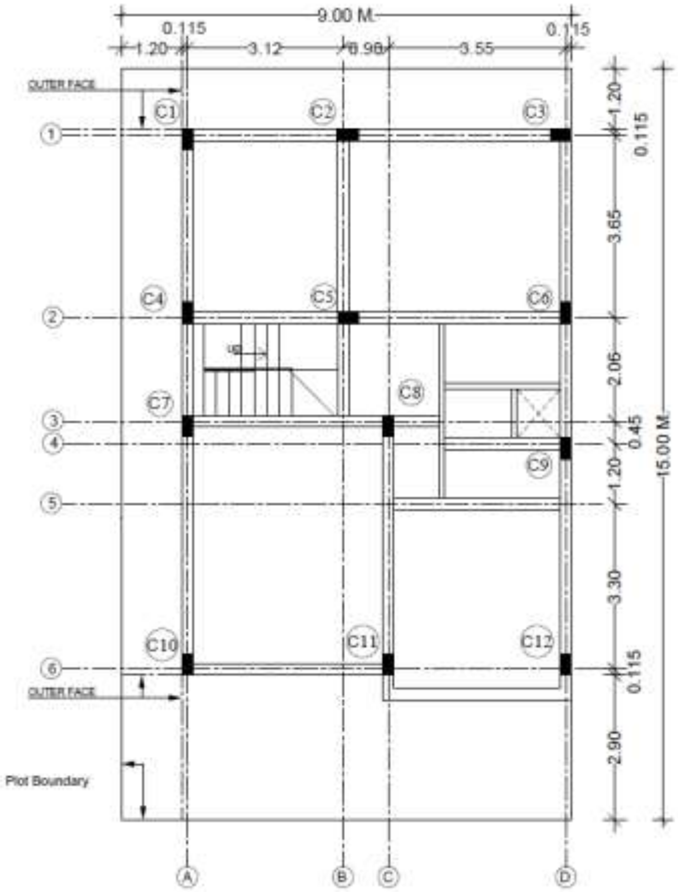

Fig.4 – Centerline Plan

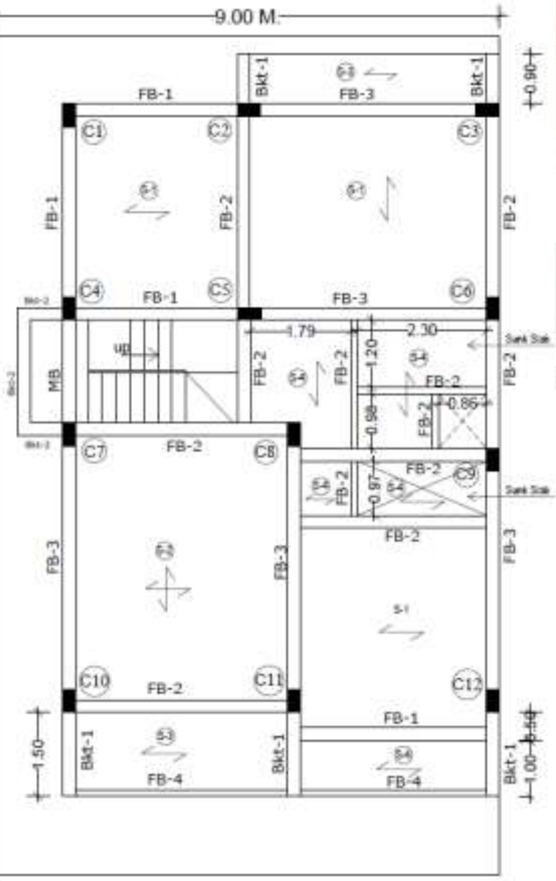

Fig.5 – Ground Floor Roof Plan

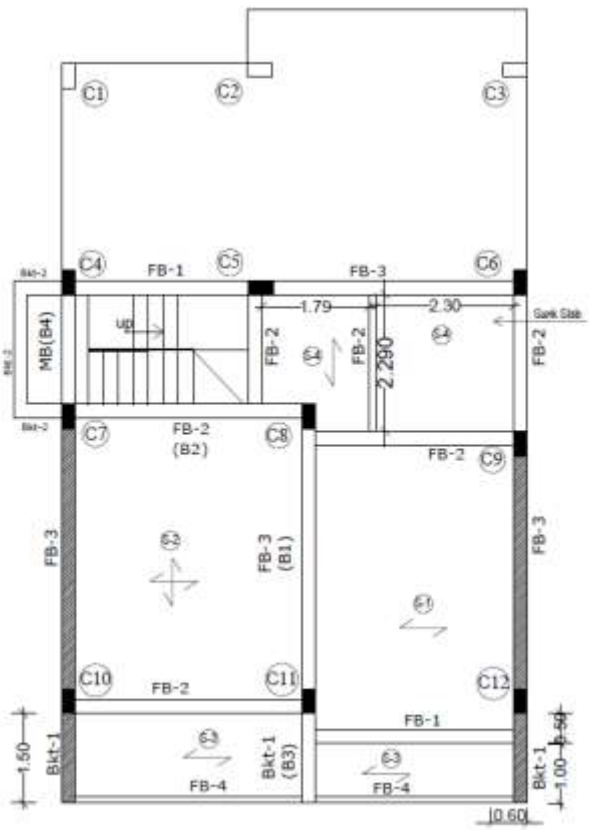

Fig.6 – First Floor Roof Plan

A. Slab validations

SLAB MARK	COVER	DEPTH	MAIN BAR DIA & SPACING	DIST. BAR DIA & SPACING
S1	20 mm	170.83 = 170 mm (approx.)	10 mm @ 260.62 mm C/C (approx. 260 mm)	8 mm @ 245.2 mm C/C (approx. 245 mm)
S2	20 mm	225 mm	10 mm @ 300 mm C/C	8 mm @ 300 mm C/C
S3-1	20 mm	233.33 = 230 mm (approx.)	10 mm @ 132.09 mm C/C (approx. 130 mm)	8 mm @ 179.52 mm C/C (approx. 180 mm)
S3-2	20 mm	191.67 = 190 mm (approx.)	10 mm @ 186.29 mm C/C (approx. 185 mm)	8 mm @ 218.55 mm C/C (approx. 220 mm)
S4	20 mm	129.17 = 130 mm (approx.)	10 mm @ 300 mm C/C	8 mm @ 300 mm C/C

Table 1 – Manual Slab Validations

Table 2 – Python Slab Validations

SLAB MARK	COVER	DEPTH	MAIN BAR DIA & SPACING	DIST. BAR DIA & SPACING
S1	20mm	170.83 = 170 mm (approx.)	10 mm @ 260.18 mm C/C (approx. 260 mm)	8 mm @ 245.22 mm C/C (approx. 245 mm)
S2	20mm	225 mm	10 mm @ 300 mm C/C	8 mm @ 300 mm C/C
S3-1	20mm	233.33 = 230 mm (approx.)	10 mm @ 132.28 mm C/C (approx. 130 mm)	8 mm @ 179.54 mm C/C (approx. 180 mm)
S3-2	20mm	191.67 = 190 mm (approx.)	10 mm @ 186.25 mm C/C (approx. 185 mm)	8 mm @ 218.57 mm C/C (approx. 220 mm)
S4	20mm	129.17 = 130 mm (approx.)	10 mm @ 300 mm C/C	8 mm @ 300 mm C/C

B. Beam validations

Table 3 – Manual Beam Validations

BEAM MARK	COVER	SIZE	TENSION BARS	COMPRESSION BARS	STIRRUPS (TWO LEGGED)
FB-3	50mm	230mm*450mm	2.25 NOS (3 NOS) of 20 mm bars at mid span	2 NOS of 20 mm bars at mid span	8 mm @ 300 mm in centre
			2.45 NOS (3 NOS) of 20 mm bars at support	2 NOS of 20 mm bars at support	8mm @ 173.18 mm (170 mm approx.) upto 1 m from either supports
FB-2	50mm	230mm*450mm	0.58 NOS == 2 NOS of 20 mm bars at mid span	-	8 mm @ 300 mm in centre
			0.556 NOS == 2 NOS of 20 mm bars at support	-	8mm @ 300 mm upto 0.6 m from either supports
BKT-1	50mm	230mm*450mm	0.176 NOS == 2 NOS of 20 mm bars at mid span	-	8 mm @ 300 mm in centre
			2.226 NOS (3 NOS) of 20 mm bars at support	-	8mm @ 248.63 mm (250 mm approx.) upto 0.5 m from either supports
MB	50mm	230mm*380mm	0.328 NOS == 2 NOS of 20 mm bars at mid span	-	8 mm @ 247.5 mm (250 mm approx.) in centre
			0.314 NOS == 2 NOS of 20 mm bars at support	-	8mm @ 300 mm upto 0.6 m from either supports

Table 4 – Python Beam Validations

BEAM MARK	COVER	SIZE	TENSION BARS	COMPRESSION BARS	STIRRUPS (TWO LEGGED)
FB-3	50mm	230mm*450mm	2.74 NOS (3 NOS) of 20 mm bars at mid span	2 NOS of 20 mm bars at mid span	8 mm @ 300 mm in centre
			2.74 NOS (3 NOS) of 20 mm bars at support	2 NOS of 20 mm bars at support	8mm @ 158.77 mm (160 mm approx.) upto 1.08 m from either supports
FB-2	50mm	230mm*450mm	2 NOS of 20 mm bars at mid span	-	8 mm @ 300 mm in centre
			2 NOS of 20 mm bars at support	-	8mm @ 300 mm upto from either supports
BKT-1	50mm	230mm*450mm	2 NOS of 20 mm bars at mid span	-	8 mm @ 300 mm in centre
			2.28 NOS (3 NOS) of 20 mm bars at support	-	8mm @ 300 mm upto 0.34 m from either supports
MB	50mm	230mm*380mm	2 NOS of 20 mm bars at mid span	-	8 mm @ 247.5 mm (250 mm approx.) in centre
			1.26 NOS (2 NOS) of 20 mm bars at support	-	8mm @ 196.18 mm upto 0.42 m from either supports

C. Column validations

Table 5 – Manual Column Validations

COLUMN MARK	COVER	SIZE	NO. OF BARS	STIRRUPS (TWO LEGGED)
C-9 (CATEGORY 1)	25mm	230mm*450mm	1.687 NOS == 4 NOS of 25 mm bars	8 mm @ 230 mm C/C
C-8 (CATEGORY 2)	25mm	230mm*450mm	4.639 NOS (5 NOS) of 25 mm bars	8 mm @ 230 mm C/C

Table 6 – Python Column Validations

COLUMN MARK	COVER	SIZE	NO. OF BARS	STIRRUPS (TWO LEGGED)
C-9 (CATEGORY 1)	25mm	230mm*450mm	4 NOS of 25 mm bars	8 mm @ 192 mm C/C
C-8 (CATEGORY 2)	25mm	230mm*450mm	4.43 NOS (5 NOS) of 25 mm bars	8 mm @ 192 mm C/C

D. Foundation validations

Table 7 – Manual Foundation Validations

COLUMN MARK	COVER	SIZE	DEPTH	X - AXIS REINFORCEMENT	Y - AXIS REINFORCEMENT
F-C9 (CATEGORY 1)	50mm	1897mm (X) * 1677mm (Y)	450 mm	9.995 NOS == 10 NOS of 12 mm bars @ 173.889 mm C/C (170 mm approx.)	11 NOS == 12 NOS of 12 mm bars @ 162.273 mm C/C (160 mm approx.)
F-C8 (CATEGORY 2)	50mm	2214mm (X) * 1994mm (Y)	500 mm	31.074 NOS == 32 NOS of 12 mm bars @ 60.71 mm C/C (60 mm approx.)	12.88 NOS == 13 NOS of 12 mm bars @ 175.167 mm C/C (175 mm approx.)

Table 8 – Python Foundation Validations

COLUMN MARK	COVER	SIZE	DEPTH	X - AXIS REINFORCEMENT	Y - AXIS REINFORCEMENT
F-C9 (CATEGORY 1)	50mm	1920mm (X) * 1700mm (Y)	500 mm	6.56 NOS == 8 NOS of 12 mm bars @ 226.86 mm C/C (225 mm approx.)	7.15 NOS == 8 NOS of 12 mm bars @ 258.29 mm C/C (260 mm approx.)
F-C8 (CATEGORY 2)	50mm	2240mm (X) * 2020mm (Y)	500 mm	25.76 NOS == 26 NOS of 12 mm bars @ 76.32 mm C/C (75 mm approx.)	13.48 NOS == 14 NOS of 12 mm bars @ 163.69 mm C/C (160 mm approx.)

IX. Conclusion

The conclusion of this research is that manual calculations take a long time, measured in hours, whereas using a programming language like Python, we can write our own program and answer the identical problem in a matter of seconds.

Furthermore, the programming has demonstrated that we may easily relate different components of our problems, such as depth, width, and so on.

X. Future Scope

The future scope of this research is that we can use multiple python packages available or create our own package to construct a software in Python that takes into account earthquake loads for high-rise buildings.

References

[1] Mohammed G. Mohammed and Kathleen M. Trauth. 2020. GIS-Based Python Simulation of Infiltration over a Landscape American Society of Civil Engineers.
[2] Dr. P. D. Hiwase, Mr. Shashank Bisen and Mr. Pratik Surana. 2018. Adoption of Programming Codes in the Design of Earth Retaining Wall in Different Backfill Conditions. The International Journal of Engineering and Science (IJES).
[3] Amar Khennane. 2005. Performance Design of Reinforced Concrete Slabs Using Commercial Finite Element Software.
[4] Arshad Quraishi and N. K. Dhapekar. 2021. Applicability of Python in Civil Engineering: Review. International Research Journal of Engineering And Technology.
[5] Aravind N. , Nagajothi S. and Elavenil S. 2021. Machine learning model for predicting the crack detection and pattern recognition of geopolymer concrete beams. Elsevier Journal.
[6] Shivaji M. Sarvade and Sachin M. Pore. 2019. Use of Python Programming for interactive design of Reinforced Concrete Structures. National Conference on Exploring New Dimensions in Teaching Learning for Quality Education.
[7] Petteri Huuhka. 2018. Automation of Construction Planning Of a Reinforced Concrete Retaining Wall. Master Thesis.
[8] Shinya Inazumi, Sutasinee Intui, Apiniti Jotisankasa, Susit Chaiprakaikeow and Kazuhiko Kojima. 2020.Artificial intelligence system for supporting soil classification. Elsevier Journal.
[9] Jesse Bonanno. 2021. InterminateBeam: A Python package for solving 1D interminate Beams. The Journal of Open Source Education.
[10] Wenli Liu, Xianguo Wu, Limao Zhang, Yanyu Wang and Jiaying Teng. 2018. Sensitivity analysis of structural health risk in operational tunnels. Elsevier Journal.
[11] Pablo Martinez, Rafiq Ahmad and Mohamed Al-Hussein. 2019. A vision-based system for pre-inspection of steel frame manufacturing. Elsevier Journal.
[12] Rasmus Rempling, Alexandre Mathern, David Tarazona Ramos and Santiago Luis Fernández. 2019. Automatic structural design by a set-based parametric design method. Elsevier Journal.
[13] Maryam Kouzehgar, Yokhesh Krishnasamy Tamilselvam, Manuel Vega Heredia and Mohan Rajesh Elara. 2019. Self-reconfigurable façade-cleaning robot equipped with deep-learning based crack detection based on convolutional neural networks. Elsevier Journal.
[14] Manishaben Jaiswal, "CRYPTOCURRENCY AN ERA OF DIGITAL CURRENCY", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.8, Issue 1, pp.60-70, January 2020, Available at :http://www.ijcrt.org/papers/IJCRT2001010.pdf
[15] Jun Kang Chow, Zhaoyu Su, Jimmy Wu, Zhaofeng Li, Pin Siang Tan, Kuan-fu Liu, Xin Mao and Yu-Hsing Wang. 2020. Artificial intelligence-empowered pipeline for image-based inspection of concrete structures. Elsevier Journal.
[16] Sumanta Das, Matthew Aguayo, Subramaniam D. Rajan, Gaurav Sant and Narayanan Neithalath. 2018. Microstructure-guided numerical simulations to predict the thermal performance of a hierarchical cement-based composite material. Elsevier Journal.
[17] M. Navarro, S. Ivorra and F.B. Varona. 2020. Parametric finite element analysis of punching shear behavior of RC slabs reinforced with bolts. Elsevier Journal.
[18] Song Ee Park, Seung-Hyun Eem and Haemin Jeon. 2020. Concrete crack detection and quantification using deep learning and structured light. Elsevier Journal.
[19] Tuan Nguyen-Sy, Jad Wakim, Quy-Dong To, Minh-Ngoc Vu, The-Duong Nguyen and Thoi-Trung Nguyen. 2020. Predicting the compressive strength of concrete from its compositions and age using the extreme gradient boosting method. Elsevier Journal.
[20] Federica Rosso, Virgilio Ciancio, Jacopo Dell'Olmo and Ferdinando Salata. 2020. Multi-objective optimization of building retrofit in the Mediterranean climate by means of genetic algorithm application. Elsevier Journal.

[21] Lewei Tong, Bo Liu and Xiao-Ling Zhao. 2017. Numerical study of fatigue behaviour of steel reinforced concrete (SRC) beams. Elsevier Journal.
[22] Sujith Mangalathu, Hansol Jang, Seong-Hoon Hwang and Jong-Su Jeon. 2020. Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls. Elsevier Journal.
[23] Xingquan Guan, Henry Burton and Thomas Sabol. 2020. Python-based computational platform to automate seismic design, nonlinear structural model construction and analysis of steel moment resisting frames. Elsevier Journal.
[24] Jou-Yi Shih, Ilaria Grossoni and Yann Bezin. 2019. Settlement analysis using a generic ballasted track simulation package. Elsevier Journal.
[25] Viet-Linh Tran, Duc-Kien Thai and Duy-Duan Nguyen. 2020. Practical artificial neural network tool for predicting the axial compression capacity of circular concrete-filled steel tube columns with ultra-high-strength concrete. Elsevier Journal.
[26] Katherine R. Barnhart, Eric Hutton and Gregory E. Tucker. 2019. umami: A Python package for Earth surface dynamics objective function construction. The Journal of Open Source Software.
[27] Umesh T. Jagadale, Chittaranjan B. Nayak, Asmita Mankar, Sunil B. Thakare and Wasudeo N. Deulkar. 2020. An experimental-based python programming for structural health monitoring of non-engineered RC frame. Springer Journal.
[28] Sama S. Memari and T. Prabhakar Clement. 2021. PySWR- A Python code for fitting soil water retention functions. Elsevier Journal.
[29] Lakshmi Elangovan, Riddhi Singh and B.V.N.P. Kambhammettu. 2021. REGSim: An open-source framework to estimate recharge and simulate groundwater heads. Elsevier Journal.
[30] Brooke E. Mason , Abhiram Mullapudi and Branko Kerkez. 2021. StormReactor: An open-source Python package for the integrated modeling of urban water quality and water balance. Elsevier Journal.
[31] J.E. Tomlinson, J.H. Arnott and J.J. Harou. 2020. A water resource simulator in Python. Elsevier Journal.
[32] Songling Xue , Ruili Shen , Huajuan Xue , Xiaoxiong Zhu , Qiong Wu and Sheng Zhang. 2021. Failure analysis of high-strength steel wire under random corrosion. Elsevier Journal.
[33] IS456:200, Plain And Reinforced Concrete Code Of Practice.
[34] SP-16, Design Aids For Reinforced Concrete to IS : 456.
[35] IS875 – Part 2, Code Of Practice For Design Loads (Other Than Earthquake) For Buildings And Structures.