XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Development of a Python Program for Analysis of Grid Structures by Finite Element Method

Diogo M. Cardoso[1,2], Maria S. M. Sampaio[1,3]

[1]*Dept. of Civil Engineering, University of Amazonas State*
Darcy Vargas Avenue, 69050-020, Amazonas/Manaus, Brazil
[2]*dmc.eng22@uea.edu.br,* [3]*msampaio@uea.edu.br*

Abstract. Grid structures are commonly used in civil engineering systems, such as floor slabs, bridge decks, etc., where elements are subjected to bending in two orthogonal directions and torsion. Proper analysis of these systems requires numerical techniques capable of accurately capturing their structural behavior. The development of a computational tool in Python for the analysis of planar grid structures based on the Finite Element Method (FEM) helps optimize and solve various problems. Implementation involves calculating local stiffness matrices for each element, transforming coordinates to the global reference system, and assembling the global stiffness matrix. Once the structure's geometry, boundary conditions, and loads are defined, the system of equations is solved to obtain nodal displacements. From these, internal forces, such as bending moments, torsional moments, shear forces, and support reactions, are derived. To verify the program's accuracy and reliability, benchmark problems are used, and the results are compared with analytical solutions and established software. The main objective of this project is to provide a lightweight, open-source, teaching-oriented tool that can assist engineers, students, and researchers in understanding and applying grid structure analysis, especially in academic environments where commercial solutions are not readily available.

Keywords: Grid Structures, Finite Element Method, Structural Analysis, Python Programming.

1 Introduction

Grids are planar models with loads in the direction perpendicular to the plane, including moments about the plane axes. Typically, grid models are used to represent the behavior of a building floor or a bridge deck. Structural grid analysis is fundamental in civil engineering, and the use of computational tools allows for a more efficient and accurate approach to calculating internal forces, optimizing the structural design and sizing process.

The growing demand for fast and accurate solutions motivates the development of accessible and effective computational tools. However, knowledge of the fundamentals of basic structural analysis methods is extremely important, as using computer programs without adequate knowledge of structural analysis can be very dangerous, as erroneous analysis results are the cause of serious accidents in civil construction.

Engineering programs related to structural analysis assist the user in finding solutions to a given problem. Many engineering problems can be expressed in terms of partial differential derivatives and have analytical solutions. However, in real-world problems, the analytical solution is complex, requiring the use of a numerical tool to obtain approximate solutions.

A widely used method for calculating approximate solutions is the Finite Element Method (FEM). Its main feature, as confirmed by Hughes [1], Zienkiewicz and Taylor [2], Oden and Reddy [3], and Reddy [4], is to divide a body into finite elements, connected by nodes, and obtain an approximate solution to the problem under analysis. Structural analysis can now be viewed as a computational simulation of structural behavior. According to Martha [5], "it is important to keep in mind that it is inconceivable, in the 21st century, to perform structural analysis tasks, even in the case of bar structures, without the use of computers and computer graphics." With the advancement of

computer technologies, the need for specialized tools for structural analysis becomes evident. Therefore, the development of this program aims to provide an accessible and educational resource for engineers and students in the field, facilitating structural analysis without the need for expensive commercial software.

2 General formulation of the FEM for analysis in frame structures

A plane grid finite element has length (L), modulus of elasticity (E), bending moment of inertia (I) and shear modulus (G)associated with the torsional moment (J). When subjected to a distributed load (q) along its length, bending moments arise in two orthogonal directions, in addition to corresponding torsional moments and shear reactions.

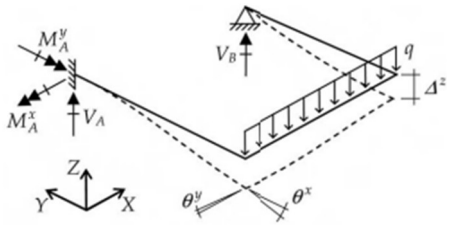


Figure 1. Global axes, loads, reactions, displacements and rotations of a grid (Martha [5]).

Figure 1 shows a grid in the XY plane with a uniformly distributed force transverse to that plane. The support reactions of a grid have only one force component, which is in the vertical Z direction, and two moment components, indicated by double arrows. Assumingly, a grid does not exhibit displacements within its plane. Figure 1 shows the deformed configuration of the grid, which exposes the following displacement and rotation components: Δ^z,$\theta^x$and θ^y. Being the displacement in the direction of the global Z axis, rotation around the global X axis and rotation around the global Y axis, respectively, Martha [5].

2.1 Fundamentals of the stiffness method

Both matrix analysis and FEM lead to the same stiffness matrix for the frame element. While matrix analysis uses the displacement method, FEM can use variational or energy methods to obtain the element's stiffness matrix. However, FEM is more general and can be used to formulate more general 2D and 3D elements, Martha[5], Vaz[6].

The stiffness method consists of subdividing the structure into finite elements, whose endpoints are nodes representing the connections between the members. The force–displacement properties of each element are determined and related through the equilibrium equations at the nodes, forming the global stiffness matrix K. From this, the nodal displacements for an applied load are determined, and, using these displacements, the internal and external forces of the structure are calculated. To do this, we must first establish an element stiffness matrix [k'] shown in Eq. (1) for each frame element, Hibbeler [7].

$$\left[\hat{k}\right]_e = \begin{bmatrix} \frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} \\ 0 & \frac{GJ}{L} & 0 & 0 & -\frac{GJ}{L} & 0 \\ \frac{6EI}{L^2} & 0 & \frac{4EI}{L} & -\frac{6EI}{L^2} & 0 & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} \\ 0 & -\frac{GJ}{L} & 0 & 0 & \frac{GJ}{L} & 0 \\ \frac{6EI}{L^2} & 0 & \frac{2EI}{L} & -\frac{6EI}{L^2} & 0 & \frac{4EI}{L} \end{bmatrix}_e \tag{1}$$

This matrix relates the vector of equivalent nodal forces $\{f_e\}$ with the vector of nodal displacements $\{d_e\}$:

$$\left[\hat{k}\right].\{d_e\} = f_e \tag{2}$$

Subsequently, a coordinate transformation is applied to relate the element's local system to the structure's global system. The matrices of all elements are then combined to form the global stiffness matrix [K]. By applying the boundary conditions and nodal loads, the global system is solved:

$$[K].\{D\} = \{F\} \tag{3}$$

In Eq. (3), {D} is the global displacement vector, and {F} is the vector of applied external forces. Finally, with the known displacements, the internal forces (shear forces, bending moments and torsion) in each element are obtained, as well as the reactions at the supports.

3 General flowchart for the analysis of grilling structures

A general flowchart was created to demonstrate the processing process for solving grid structure analyses. It describes each step in arriving at the solution for the structure's nodal displacements, forces, moments, and torsion, as shown in Figure 2.

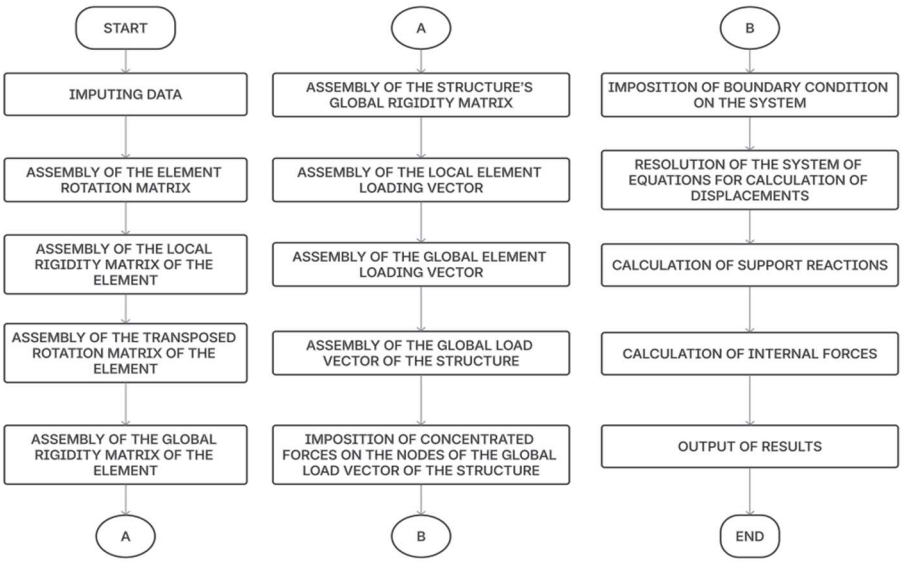


Figure 2. Flowchart of the developed python program.

The flowchart presented in Figure 2 clearly and organized illustrates the computational process adopted for grid analysis. It allows visualization of the logical sequence of operations, from defining the initial parameters to calculating the final displacements and forces. This representation facilitates understanding of the method, highlighting critical steps such as assembling the global stiffness matrix, applying boundary conditions, and solving the system of equations. Furthermore, the flowchart serves as a basis for algorithmic implementation and verification of the analysis procedure.

4 Numerical Results and Validation

Numerical examples available in the literature were tested, and the results demonstrated the good performance of the implemented tool for the tested examples. This section will cover some examples.

4.1 Cantilever grate subjected to a vertical load presented by Soriano [8]

The problem consists of a cantilevered grid structure with adopted lengths a = 3 m and b = 2 m, as shown in Figure 3. The geometric and material properties are unitary. A vertical force P = 28kN is applied to one of the nodes. Tables 1 to 2 present the current and reference solutions for the internal forces, respectively. Tables 3 and

4 also show the solutions for the support reactions, one being the reference and the current solution. As can be seen, the numerical solutions converge with Soriano's solution [8], being equal to. Figures 4 and 5 show the source code for the calculation and the steps from node definition to property application and, finally, the results output.

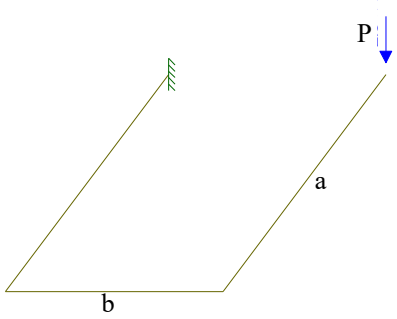


Figure 3. Cantilevered grille (Soriano [8]).

Table 1. Present solution for internal forces (initial and final)

Element	Shear(kN)	Torsion(kNm)	Moment(kNm)
1	28,0 – 28,0	56,0 – 56,0	0,0 – 84,0
2	28,0 – 28,0	84,0 – 84,0	-56,0 – 0,0
3	28,0 – 28,0	0,0 – 0,0	-84,0 – 0,0

Table 2. Reference solution for internal forces (initial and final)

Element	Shear(kN)	Torsion(kNm)	Moment(kNm)
1	28 – 28	56 – 56	0 – 84
2	28 – 28	84 – 84	-56 – 0
3	28 – 28	0 – 0	-84 – 0

Table 3. Reference support reactions

Node	V(kN)	T(kNm)	M(kNm)
1	28	-56	0,0
2	0,0	0,0	0,0
3	0,0	0,0	0,0
4	0,0	0,0	0,0

Table 4. Present solution of support reactions

Node	V(kN)	T(kNm)	M(kNm)
1	28,0	-56,0	0,0
2	0,0	0,0	0,0
3	0,0	0,0	0,0
4	0,0	0,0	0,0

```
from modulos.no import no  as n
from modulos.apoio import apoio as ap
from modulos.forca_aplicada import forca_aplicada as qa
from modulos.elemento import elemento as el
from modulos.propriedade_elemento import propriedade_elemento as pel
from modulos.geometria import prop_secao as ps
from modulos.carga_distribuida import carregamento_dist as cd

b = 1
h = 1
A = b * h
I = 1
E = 1 #MPa
J = 1
G = 1

n.no(0,0) #A
n.no(3,0) #B
n.no(3,2) #C
n.no(0,2)

ap.terc_gen(1)
```

Figure 4. Source code presentation example 1 - Part I.

```

#Conectividade em sentido positivo

el.elemento(1,2) #A B
el.elemento(2,3) #B C
el.elemento(3,4) #C D

for i in range(1,el.n_elem + 1):
    pel.generico(E,G,i)

for i in range(1,el.n_elem + 1):
    ps.generico(A,I,J,i)

qa.forca_aplicada_no(4,-28,0,0)

from modulos.matrizrigidez import resultado

resultado.executar()

for i in range(1,n.n_no + 1):
    n.exibir_reacoes_no(i)

for i in range(1,n.n_no + 1):
    n.exibir_deslocamento_no(i)
```

Figure 5. Source code presentation example 1 - Part II

4.2 Isostatic grid of two orthogonal bars subjected to a distributed load presented by Soriano [8]

The next example consists of an isostatic grid structure, as shown in Figure 6, with constraints at three nodes. The geometric and material properties are unitary. A distributed load q = 20kN/m is applied across the elements. Tables 5 to 8 present the reference and current solutions of the structure provided by Soriano [8]. Figures 7 and 8 show the source code for the calculation and the steps from node definition to property application and, finally, result generation, for the second example. The solutions are all the same, as expected from literature.

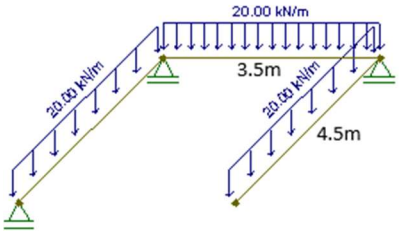


Figure 6. three-supported grid (Soriano [8]).

Table 5. Reference internal forces (initial and final)

Element	Shear(kN)	Torsion(kNm)	Moment(kNm)
1	90 – 0,0	0,0 – 0,0	0,0 – 202,5
2	35 – -35	-202,5 – -202,5	0,0 – 0,0
3	90 – 0,0	0,0 – 0,0	-202,5 – 0,0

Table 6. Present solution for internal forces (initial and final)

Element	Shear(kN)	Torsion(kNm)	Moment(kNm)
1	90,0 – 0,0	0,0 – 0,0	0,0 – 202,50
2	35,0 – -35,0	-202,50 – -202,50	0,0 – 0,0
3	90 – 0,0	0,0 – 0,0	-202,50 – 0,0

Table 7. Reference support reactions

Node	V(kN)	T(kNm)	M(kNm)
1	90	0,0	0,0
2	35	0,0	0,0
3	125	0,0	0,0
4	0,0	0,0	0,0

Table 8. Present solution of support reactions

Node	V(kN)	T(kNm)	M(kNm)
1	90,0	0,0	0,0
2	35,0	0,0	0,0
3	125,0	0,0	0,0
4	0,0	0,0	0,0

```
from modulos.no import no_ as n
from modulos.apoio import apoio as ap
from modulos.forca_aplicada import forca_aplicada as qa
from modulos.elemento import elemento as el
from modulos.propriedade_elemento import propriedade_elemento as pel
from modulos.geometria import prop_secao as ps
from modulos.carga_distribuida import carregamento_dist as cd

b = 1
h = 1
A = b * h
I = 1
E = 1 #MPa
J = 1
G = 1

n.no(0,0) #A
n.no(0,4.5) #B
n.no(3.5,4.5) #C
n.no(3.5,0)

ap.prim_gen(1)
ap.prim_gen(2)
ap.prim_gen(3)
```

Figure 7. Source code presentation example 2- Part I.

```
#Conectividade em sentido positivo

el.elemento(1,2) #A B
el.elemento(2,3) #B C
el.elemento(3,4) #C D

for i in range(1,el.n_elem + 1):
    pel.generico(E,G,i)

for i in range(1,el.n_elem + 1):
    ps.generico(A,I,J,i)

cd.carga_dist(1,-20,-20)
cd.carga_dist(2,-20,-20)
cd.carga_dist(3,-20,-20)

from modulos.matrizrigidez import resultado

resultado.executar()

for i in range(1,n.n_no + 1):
    n.exibir_reacoes_no(i)

for i in range(1,n.n_no + 1):
    n.exibir_deslocamento_no(i)
```

Figure 8. Source code presentation example 2-Part II.

4.3 Cantilever grid subjected to distributed loads presented by Soriano [8]

The last problem presented is a cantilevered grid structure, as shown in Figure 9, with a fixation constraint at the first node, and with values of a, b, and c equal to 6 m, 4 m, and 3 m, respectively. The geometric and material properties are unitary. Distributed loads q = 20kN/m are applied along two elements. Tables 9 to 12 show the reference solutions and the current solution for the internal forces and support reactions of the problem presented by Soriano [8]. Figures 10 and 11 show the source code for the calculation and the steps from defining the nodes to applying the properties and, finally, the output of the results, for the last example. It can also be observed that the results are the same as those presented by Soriano.

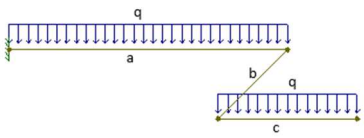


Figure 9. Cantilever grid structure (Soriano [8])

Table 9. Reference internal forces (initial and final)

Element	Shear(kN)	Torsion(kNm)	Moment(kNm)
1	100 – 60	-240 – -240	-810 – -90
2	-60 – -60	90 – 90	-240 – 0
3	60– 0,0	0,0 – 0,0	-90 – 0,0

Table 10. Present solution for internal forces (initial and final)

Element	Shear(kN)	Torsion(kNm)	Moment(kNm)
1	100,0 – 60,0	-240,0 – -240,0	-810,0 – -90,0
2	-60,0 – -60,0	90,0 – 90,0	-240,0 – 0
3	60,0– 0,0	0,0 – 0,0	-90 – 0,0

Table 11. Reference support reactions

Node	V(kN)	T(kNm)	M(kNm)
1	180	240	810
2	0,0	0,0	0,0
3	125	0,0	0,0
4	0,0	0,0	0,0

Table 12. Present solution of reactions

Node	V(kN)	T(kNm)	M(kNm)
1	180,0	240,0	810,0
2	0,0	0,0	0,0
3	0,0	0,0	0,0
4	0,0	0,0	0,0

```python
from modulos.no import no_ as n
from modulos.apoio import apoio as ap
from modulos.forca_aplicada import forca_aplicada as qa
from modulos.elemento import elemento as el
from modulos.propriedade_elemento import propriedade_elemento as pel
from modulos.geometria import prop_secao as ps
from modulos.carga_distribuida import carregamento_dist as cd

b = 1
h = 1
A = b * h
I = 1
E = 1 #MPa
J = 1
G = 1

#Adotando a = 6, b = 4, c = 3 e p = 20kN/m

n.no(0,0) #A
n.no(6,0) #B
n.no(6,-4) #C
n.no(9,-4) #D

ap.terc_gen(1)

```

Figure 10. Source code presentation example 3-Part I.

```python
#Conectividade em sentido positivo

el.elemento(1,2) #A B
el.elemento(3,2) #B C
el.elemento(3,4) #C D

for i in range(1,el.n_elem + 1):
    pel.generico(E,G,i)

for i in range(1,el.n_elem + 1):
    ps.generico(A,I,J,i)

cd.carga_dist(1, -20, -20)
cd.carga_dist(3, -20, -20)

from modulos.matrizrigidez import resultado

resultado.executar()

for i in range(1,n.n_no + 1):
    n.exibir_reacoes_no(i)

for i in range(1,n.n_no + 1):
    n.exibir_deslocamento_no(i)
```

Figure 11. Source code presentation example 3-Part II.

5 Conclusions

The development of a Python program for analyzing grid structures using the Finite Element Method (FEM) has proven to be an efficient and flexible approach to structural analysis. By implementing the stiffness method in a modular and automated manner, the program allows for the calculation of nodal displacements, internal forces, and support reactions with high reliability. It is important to note that the program is still under development for the implementation of the graphical interface. In conclusion, the use of Python, combined with its numerical and visualization libraries, has proven to be a suitable choice for developing open-source tools for both teaching and research. This approach allows for code customization and expansion, encouraging innovation and the dissemination of knowledge in computational methods applied to engineering.

Acknowledgements. The authors would like to thank the University of Amazonas State for the financial support granted through the “ICT Ações Afirmativas” Program: 2025-2026 Edition/SISPROJ Project Number 57952.

Authorship statement. The authors hereby confirm that they are the sole liable people responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors or has the permission of the owners to be included here.

References

[1] T.J.R Hughes. *The Finite Element Method – linear static and dynamic finite element analysis*. Prentice-Hall International, Inc. Englewood Cliffs, New Jersey, 1987.
[2] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method: Basic Formulation and Linear Problems,* 4th Edition, vol. 1. McGraw-Hill, 1989.
[3] J.T. Oden and J.N. Reddy, *An introduction to the mathematical theory of finite elements*. John Wiley & Sons, New York, 1976.
[4] J.N Reddy. *Energy and variational methods in applied mechanics*. John Wiley & Sons, New York, 1984.
[5] L. F Martha. *Análise de Estruturas: Conceitos e Métodos Básicos*. Rio de Janeiro: Elsevier, 2010.
[6] L. E. Vaz. *Método dos elementos finitos em análise de estruturas.*Rio de Janeiro: Elsevier, 2011.
[7] R. C. Hibbeler. *Análise das Estruturas.* 8º ed. Pearson Education do Brasil, São Paulo, 2013.
[8] H. L. Soriano. *Estática das Estruturas.*4° ed revista - 2014.Rio de Janeiro: Editora Ciência Moderna Ltda., 2014.