

DESENVOLVIMENTO DE ALGORITMO EM PYTHON PARA ANÁLISE DE ESTRUTURAS DE GRELHA

Isabel Franco Oliveira Gomes¹
Arthur Álx de Araújo Albuquerque²

RESUMO

A engenharia estrutural desempenha um papel fundamental na concepção e análise de estruturas que atendam a rigorosos padrões de segurança e eficiência. Nesse contexto, a aplicação de métodos numéricos, como o Método da Rigidez Direta, torna-se essencial para a idealização e análise estrutural. O presente estudo tem como propósito o desenvolvimento de um código em linguagem de programação Python fundamentado no Método da Rigidez Direta, capaz de determinar com precisão deformações, reações de apoio e esforços internos de estruturas de grelhas no regime elástico linear. O algoritmo elaborado ao término do estudo demonstra eficácia, produzindo resultados satisfatórios que se equiparam aos dados encontrados na literatura ou gerados por cálculos manuais de análise estrutural de sistemas estruturais de grelha. A fim de validar o algoritmo, são apresentados alguns exemplos de grelhas sujeitas a diversos tipos de carregamentos, condições de contorno e ligações, com o objetivo de comparar os resultados obtidos através do código desenvolvido com aqueles provenientes de cálculos manuais e de outros autores da mesma área temática.

Palavras-chave: Grelha. Método da Rigidez Direta. Algoritmo. Análise estrutural.

DEVELOPMENT OF AN ALGORITHM IN PYTHON FOR ANALYZING GRID STRUCTURES

ABSTRACT

Structural engineering plays a fundamental role in the design and analysis of structures that meet rigorous safety and efficiency standards. In this context, the application of numerical methods, such as the Direct Stiffness Method, becomes essential for structural idealization and analysis. In this context, the purpose of the present study is to develop a code in Python programming language based on the Direct Stiffness Method, capable of accurately determining deformations, support reactions and internal efforts of grid structures in the linear elastic regime. The algorithm developed at the end of the study demonstrates effectiveness, producing satisfactory results that are comparable to data found in the literature or generated by manual calculations of structural analysis of structural grid systems. In order to validate the algorithm, some examples of grids subjected to different types of loads, boundary conditions and connections are presented, with the aim of comparing the results obtained through the developed code with those coming from manual calculations and from other authors of the same field. thematic area.

Keywords: Grid. Direct Stiffness Method. Algorithm. Structural analysis.

Recebido em 19 de setembro de 2024. Aprovado em 22 de outubro de 2024

¹ UNIALFA, Centro Universitário Alves Faria, E-mail: isabelfrancooliveira15@gamil.com

² UFG, Universidade Federal de Goiás, E-mail: arthuralax@ufg.br

INTRODUÇÃO

Uma estrutura pode ser definida como a componente de uma construção que suporta as forças externas e mantém o equilíbrio da edificação. Ela é constituída por diversos elementos estruturais que, em conjunto, formam um sistema estrutural (CARVALHO e FIGUEIREDO FILHO, 2014).

Na engenharia existem diversos tipos de sistemas estruturais, dentre os quais temos as estruturas reticuladas (LEET, UANG e GILBERT, 2010). Estas são compostas por elementos lineares chamados de barras, cujas dimensões transversais são significativamente menores que suas dimensões longitudinais, unidos por nós, que podem ser rígidos ou articulados. A classificação das estruturas reticuladas pode ser feita considerando o seu arranjo espacial, suas solicitações, os esforços internos e pelos estados de tensões característicos de cada modelo estrutural (GERE e WEAVER, 1990).

Os modelos estruturais reticulados são divididos em treliças, vigas, grelhas e pórticos, que podem ser planos, quando todas as barras e carregamentos estão contidos em um mesmo plano ou espaciais quando a disposição das barras e/ou carregamentos é tridimensional, onde cada tipo de estrutura possui esforços internos característicos (KASSIMALI, 1999).

O sistema estrutural de grelhas pode ser definido como um conjunto de barras conectadas por ligações rígidas e/ou flexíveis denominados de nós ou pontos nodais em que as cargas atuam perpendicularmente ao plano da estrutura (KASSIMALI, 1999) **Erro! Fonte de referência não encontrada.** Nos elementos do sistema estrutural de grelha podem existir esforços internos cisalhantes, de momento fletor e momento torçor, podendo gerar estado de tensão de flexão simples com torção.

As estruturas de grelhas podem ser utilizadas para análise estrutural de placas e lajes, com intuito de dimensionar elementos como por exemplo estruturas de coberturas, pisos, tabuleiros de pontes, viadutos e passarelas.

Em busca de conceber estruturas seguras e eficientes, foram desenvolvidos e aprimorados diversos métodos de análise e cálculos, dos quais, o Método da Rigidez Direta (MRD), que é fundamentado na generalização do método dos deslocamentos e também chamado de Análise Matricial de Estruturas, destaca-se na implementação computacional de estruturas reticuladas, uma vez que é baseado na manipulação de vetores e matrizes (MARTHA, 2018).

O MRD é amplamente utilizado na análise estrutural devido à sua eficiência computacional e à sua capacidade de lidar com estruturas complexas. No entanto, ele requer uma compreensão sólida das propriedades de rigidez dos materiais e das técnicas de formulação de matrizes. Além disso, o MRD é formulado em torno do equilíbrio dos nós de um modelo e as incógnitas do problema são os deslocamentos (MARTHA, 2022). Ao realizar a discretização de um sistema estrutural de grelha, tem-se que cada elemento possui dois nós, cada qual com três graus de liberdade, sendo eles: o deslocamento vertical, e as rotações da seção transversal devido à torção e à flexão.

Neste contexto, o presente trabalho tem como objetivo mostrar uma estratégia de implementação do Método da Rigidez Direta em Python para a análise de sistemas estruturais reticulados utilizando elementos de grelha. Em seguida, com a finalização da implementação realiza-se com o código a análise do comportamento estrutural de três sistemas estruturais de grelhas submetidas a diferentes tipos de carregamento, condições de contorno e ligações internas dentro do regime linear elástico.

MÉTODO DA RIGIDEZ DIRETA (MRD) PARA ESTRUTURA DE GRELHA

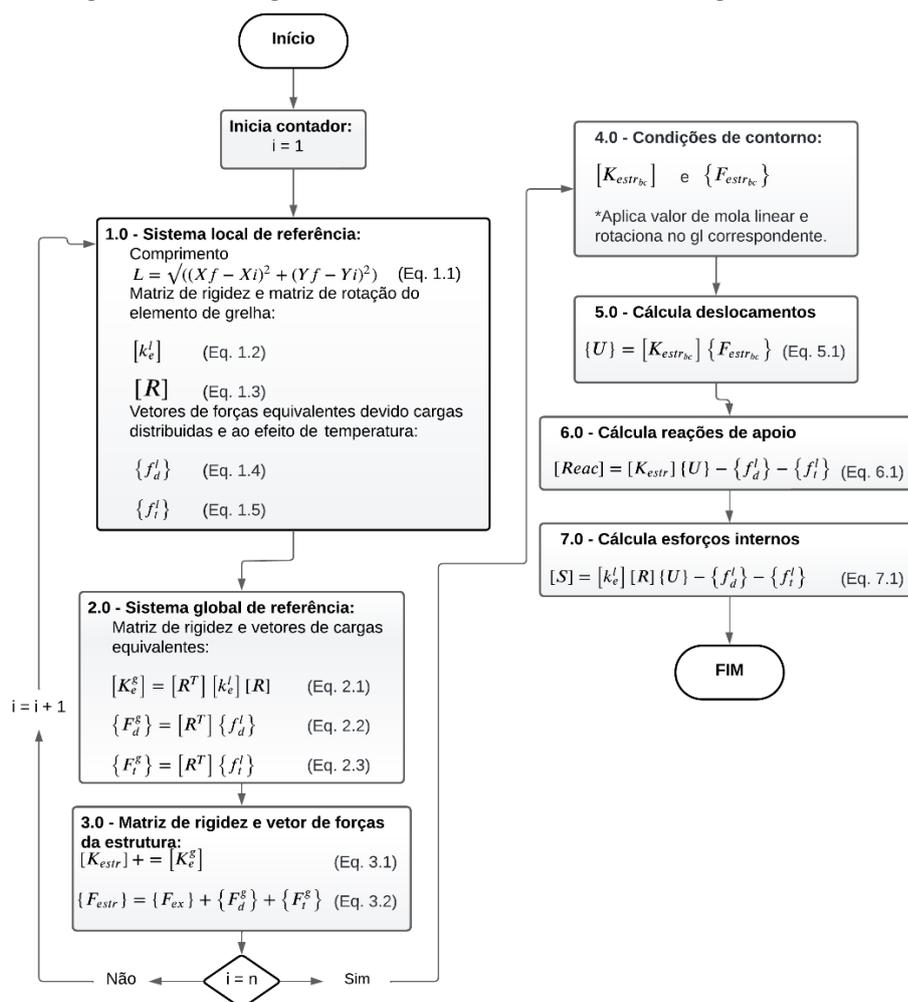
O MRD, também conhecido como Análise Matricial de Estruturas, é uma técnica amplamente explorada na análise estrutural para determinar as respostas de uma estrutura a diferentes solicitações. Ele é baseado no princípio de que as respostas da estrutura podem ser calculadas diretamente a partir das propriedades de rigidez dos elementos individuais que a compõem.

Para realizar o cálculo utilizando o MRD, primeiramente é necessário discretizar toda a estrutura, ou seja, numerar os pontos nodais, elementos, graus de liberdade e definir a incidência de cada elemento (indicar nó inicial e final). Além disso, é indispensável compatibilizar as unidades de medidas dos dados necessários para o cálculo.

No caso de estruturas de grelha, os dados utilizados são: E – Módulo de elasticidade do material, G – Módulo de elasticidade transversal do material, I – Inércia da seção transversal, J – Inércia à torção, coeficiente de dilatação térmica, coeficiente de rigidez à rotação e linear referentes a mola translacionais e rotacionais, comprimentos dos elementos e diversos tipos de cargas, tais como pontuais e distribuídas.

A Figura 1 mostra o fluxo de cálculo utilizado na Análise Matricial de Estruturas e também a ideia geral usada na implementação computacional em Python deste trabalho.

Figura 1 - Fluxograma de cálculo do Método da Rigidez Direta



Fonte: os autores, 2024.

DESENVOLVIMENTO DO ALGORITMO

Organização da implementação do código desenvolvido

O algoritmo foi implementado utilizando a linguagem de programação Python no ambiente de desenvolvimento Visual Studio Code, seguindo uma estruturação em três fases distintas: pré-processamento, processamento e pós-processamento. Durante o pré-processamento, o código realiza a leitura dos dados de entrada referentes à estrutura a ser analisada. Os dados de entrada são inicialmente armazenados em um arquivo de formato *.txt* e organizados conforme o modelo apresentado na Figura 2.

Figura 2: Modelo do arquivo de entrada escrito em formato *.txt*.

Fonte: os autores, 2024.

```
Universidade Federal de Goiás
Escola de Engenharia Civil e Ambiental
Programa de GRELHA
Autora: Isabel Franco Oliveira Gomes
Curso: Engenharia Civil

=====
nnodes nelems ndirconst ndirload nmat nsec dim
=====

===Materials=====
material young_modulus alpha G
=====

===Transversal cross section=====
sections      inertia      height      area      J
=====

===Coordinates=====
nnode  Coordx      Coordy      Coordz
=====

===Elements=====
element node-1 node-2 material sections tipo_carreg qzi qzf mxi mxj deltaT_inf deltaT_sup free_direction-1 free_direction-2
===== (node-1) ===== (node-2) =====

===Boundary Conditions=====
node direction support_type value
===== (1-rigid / 2-spring) =====

===Loading=====
node direction value
=====
```

Neste modelo, inicia-se inserindo o número de nós ou pontos nodais (nnodes), número de elementos (nelems), de direções restritas (ndirconst), de direções carregadas com cargas pontuais (ndirloads), de materiais (nmat) e seções transversais (nsec) diferentes que possa ter na estrutura e a dimensão (dim) em que a estrutura está sendo analisada, que no caso dos sistemas estruturais de grelha é tridimensional e, portanto 3.

Em seguida, colocamos os valores de módulos de elasticidade (E – young_modulus), coeficiente de dilatação térmica (alpha) caso tenha efeito de temperatura, módulo de elasticidade transversal (G) para cada material (material) diferente que tiver na estrutura. Da mesma forma, é necessário informar os valores de inércia (I - inertia), inércia à torção (J), altura (height) e área (area) de cada seção transversal (sections). E também, as coordenadas de cada nó da estrutura (Coordinates).

Seguindo a discretização feita para estrutura, preenche-se a incidência (node1 e node2), material, seção transversal, carregamentos e tipos de ligações internas (Free_direction) referente a cada elemento da estrutura. Onde, os carregamentos distribuídos podem ser forças (qxi e qxf) ou momento torçor (mxi e mxj). A diferença de temperatura é aplicada informando as variações de temperatura nas faces superior (deltaT_sup) e inferior (delta_inf) da barra.

Deve-se informar também, as condições de contorno (Boundary Conditions) existentes na estrutura em análise, apresentando quais graus de liberdade estão restritos nos pontos nodais (node; direction) que haja apoios rígidos ou semirrígidos (Support_type: 1- rigid / 2 spring). No

caso de molas, pode-se adicionar o valor (value) da constante elástica e indicar se é uma mola linear ou rotacional pelo grau de liberdade ao qual ela está associada. Por último, as cargas pontuais (Loading) são definidas colocando o valor associado ao grau de liberdade em que ela está inserida, conforme apresentado no arquivo de entrada da Figura 2.

Continuando a implementação, no estágio de processamento, o algoritmo se baseia na aplicação de métodos de análise matricial apresentado anteriormente com o auxílio do fluxograma da **Erro! Fonte de referência não encontrada.**, conforme descrito no item de implementação em Python do código desenvolvido.

Por fim, no pós-processamento, são geradas saídas que incluem informações sobre os resultados da análise, como as reações de apoio, as forças internas e os deslocamentos, possibilitando uma avaliação detalhada do comportamento da estrutura.

Implementação em python do código desenvolvido

O algoritmo em Python inicia importando a biblioteca *numpy* para lidar com arrays (vetores e matrizes) e usa a função *open()* para abrir o arquivo de entrada chamado 'Grelha_Ex.3.txt' em modo de leitura ('r') como mostrado na figura 3.

Figura 3: Importando a biblioteca numpy em Python

```
import numpy as np

file = open('Grelha_Ex.3.txt', 'r', encoding='utf')
```

Fonte: os autores, 2024.

Para ler cada linha do arquivo de entrada sem valores a ser armazenados, foi utilizado várias chamadas do comando "*file.readline()*". Já para as linhas que contêm informações sobre a malha (número de nós, elementos, etc.) e propriedades dos materiais (módulo de elasticidade, etc.) foram lidas e armazenadas em arrays *numpy* conforme o exemplo da figura 4. Em que, primeiramente define o tamanho do vetor e o tipo de variável que ele pode receber e em seguida armazena dentro do vetor. É importante ressaltar que a função "*.split()*" foi usada para ler apenas os valores desconsiderando os espaços dentro do arquivo de entrada.

Figura 4: Código definindo e armazenando valores nos vetores

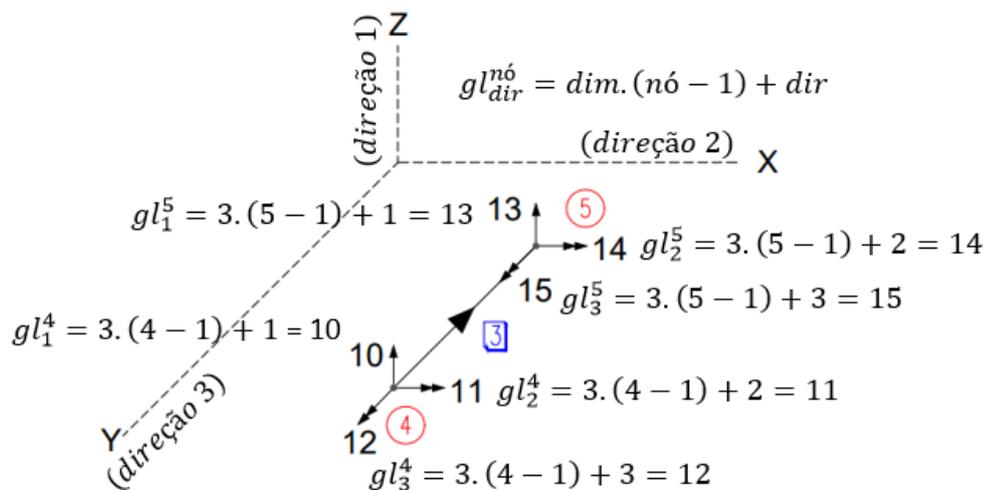
```
young = np.zeros((nmat), dtype = float)
alpha = np.zeros((nmat), dtype = float)
G = np.zeros((nmat), dtype = float)
for imat in range(0,nmat):
    line = file.readline().split()
    young[int(line[0])-1] = float(line[1])
    alpha[int(line[0])-1] = float(line[2])
    G[int(line[0])-1] = float(line[3])
```

Fonte: os autores, 2024.

Dessa mesma forma, foi feita a leitura das seções transversais, coordenadas dos nós, informações sobre os elementos, como nós conectados, materiais e carregamentos distribuídos, variação de temperatura, cargas externas pontuais aplicadas à estrutura e condições de contorno.

No entanto, no MRD, para gerar os vetores de força e deslocamento da estrutura, assim como para montar a matriz de rigidez global da estrutura é necessário identificar os graus de liberdade correspondente a cada ponto nodal. A Figura 5 mostra como pode-se realizar tal correspondência. Neste trabalho, para o elemento de grelha foi adotado as direções para cada eixo conforme mostrado na Figura 5, e em cada nó temos três direções, que vai alterar os valores dos graus de liberdade conforme cada ponto nodal.

Figura 5: Correspondência de graus de liberdade para estruturas de grelha.



Fonte : os autores, 2024.

Assim, no caso dos vetores que necessitam inserir o valor de acordo com os graus de liberdade globais associados a ele, foi realizada a implementação conforme apresentada na Figura 6, onde temos os vetores de condições de contorno (cc) e dos valores de apoios elásticos (mola). No Python, assim como em várias outras linguagens de programação inicia a contagem em zero, sendo, portanto, necessário subtrair um na expressão apresentada anteriormente na Figura 5, a fim de obter o valor dos graus de liberdade correspondente corretos dentro do programa.

Figura 6: Inserir valores em vetores de acordo com graus de liberdade.

```
cc = np.zeros((ngl), dtype = int) #Vetor de condições de contorno
mola = np.zeros((ngl), dtype = float)
for idirconst in range(0,ndirconst):
    line = file.readline().split()
    inode = int(line[0])
    idir = int(line[1])
    cc[(3*(inode-1)+idir)-1] = int(line[2])
    mola[(3*(inode-1)+idir)-1] = float(line[3])
```

Fonte: os autores, 2024.

Quando inserimos algum tipo de articulação interna no MRD, aumentamos um grau de liberdade na estrutura, que no caso de rótula, será um grau de liberdade de rotação e no caso de engaste móvel, aumenta um grau de liberdade de deslocamentos translacional. Assim, para possibilitar inserir diferentes tipos de articulações internas tais como engaste móvel e rótulas, foi desenvolvido a matriz de incidência conforme o exemplo apresentado na

Figura 7. Sendo necessário escrever esse mesmo código feito para o nó inicial para o nó final do elemento.

Figura 7: Código da matriz de conectividade do nó inicial do elemento de grelha.

```
#Cria a matriz de conectividade
conec = np.zeros((nelems,6), dtype = int)
ngl = 3*nnodes
for ielems in range(0,nelems):

    #Nó inicial
    if freedirci[ielems] == 0: # Articulação perfeita
        conec[ielems,0] = 3*(element[ielems,0]-1)+0
        conec[ielems,1] = 3*(element[ielems,0]-1)+1
        conec[ielems,2] = 3*(element[ielems,0]-1)+2

    elif freedirci[ielems] == 1: # Engaste móvel
        ngl += 1
        conec[ielems,0] = ngl - 1
        conec[ielems,1] = 3*(element[ielems,0]-1)+1
        conec[ielems,2] = 3*(element[ielems,0]-1)+2

    elif freedirci[ielems] == 2: # Rótula
        ngl += 1
        conec[ielems,0] = 3*(element[ielems,0]-1)+0
        conec[ielems,1] = ngl - 1
        conec[ielems,2] = 3*(element[ielems,0]-1)+2

    elif freedirci[ielems] == 3: # Rótula
        ngl += 1
        conec[ielems,0] = 3*(element[ielems,0]-1)+0
        conec[ielems,1] = 3*(element[ielems,0]-1)+1
        conec[ielems,2] = ngl - 1
```

Fonte: os autores, 2024.

Na parte de processamento, ou seja, da resolução da estrutura, começamos com a importação da função `sqrt` do módulo `math`. A função `sqrt` é utilizada para calcular a raiz quadrada de um número. Ao importá-la dessa forma, podemos utilizá-la diretamente em nosso código sem precisar qualificar o nome do módulo toda vez que a chamarmos. A seguir foi definido os tamanhos e tipos de *Strings* de cada vetor e matriz que serão utilizados como mostrado na Figura 8. Ainda na mesma figura, pode-se observar que é aberto o arquivo de saída denominado “*Arq_Saída_Grid24*” no modo escrita “*w*”.

Figura 8: Importando função, abrindo arquivo de saída e definindo matriz e vetores em Python.

```
from math import sqrt
file=open("Arq_Saída_Grid24.txt","w")
kestr = np.zeros((ngl,ngl), dtype = float)
fequi = np.zeros((ngl), dtype = float)
ftemp = np.zeros((ngl), dtype = float)
for ielems in range(0,nelems):
```

Fonte: os autores, 2024.

Na sequência, criou-se um laço de repetição por onde para cada elemento será calculado as informações apresentadas no item 1.0 e 2.0 do fluxograma da Figura 1. Desse modo, os comprimentos dos elementos são calculados a partir das coordenadas dos nós conforme a equação (Eq. 1.1) e os resultados são escritos dentro do arquivo de saída de formato *.txt*, tal como apresentam as Figura 9 e Figura 10.

Figura 9: Código para calcular o comprimento dos elementos de grelha.

```
#Comprimento do elemento (Length)
length = 0.0
for idim in range(0,dim):
    length += (coord[element[ielems,1]-1,idim]-coord[element[ielems,0]-1,idim])**2
length = sqrt(length)
```

Fonte: os autores, 2024.

Figura 10: Exemplo de como escrever dentro do arquivo de saída em *.txt*

```
file.write("Elemento: "f"{ielems+1}" "\n")
file.write("Comprimento: "f"{length:.4e}" "\n\n")
```

Fonte: os autores, 2024.

Em seguida, foram definidas e calculadas as matrizes de rigidez locais e as matrizes de rotação para cada elemento da estrutura, conforme as equações (Eq. 1.2) e (Eq. 1.3) apresentado no fluxograma do MRD. Em que, a implementação em linguagem Python, das relações de rigidez formuladas em função do material, seção transversal e comprimento do elemento, foi realizada tal como mostra as Figura 11 e Figura 12. Os cossenos e senos da matriz de rotação são calculados com base nas coordenadas nodais.

Figura 11: Código da matriz de rigidez local do elemento de grelha.

```
kelem_local = np.zeros((2*dim,2*dim))
kelem_local[0,0] = 12.0E+00*young[materials[ielems]-1]*inertia[sections[ielems]-1]/length**3
kelem_local[0,2] = 6.0E+00*young[materials[ielems]-1]*inertia[sections[ielems]-1]/length**2
kelem_local[0,3] = -kelem_local[0,0]
kelem_local[0,5] = kelem_local[0,2]

kelem_local[1,1] = (G[materials[ielems]-1]*J[sections[ielems]-1])/length
kelem_local[1,4] = -kelem_local[1,1]

kelem_local[2,0] = kelem_local[0,2]
kelem_local[2,2] = 4.0E+00 * young[materials[ielems]-1]*inertia[sections[ielems]-1]/length
kelem_local[2,3] = -kelem_local[0,2]
kelem_local[2,5] = 2.0E+00 * young[materials[ielems]-1]*inertia[sections[ielems]-1]/length

kelem_local[3,0] = -kelem_local[0,0]
kelem_local[3,2] = -kelem_local[0,2]
kelem_local[3,3] = kelem_local[0,0]
kelem_local[3,5] = -kelem_local[0,2]

kelem_local[4,1] = -kelem_local[1,1]
kelem_local[4,4] = kelem_local[1,1]

kelem_local[5,0] = kelem_local[0,2]
kelem_local[5,2] = kelem_local[2,5]
kelem_local[5,3] = -kelem_local[0,2]
kelem_local[5,5] = kelem_local[2,2]
```

Fonte: os autores, 2024.

Figura 12: Código da matriz de rotação do elemento de grelha.

```
#Matriz de Rotação
rot = np.zeros((2*dim,2*dim))
cos_teta = (coord[element[ielems,1]-1,0]-coord[element[ielems,0]-1,0])/length
sin_teta = (coord[element[ielems,1]-1,1]-coord[element[ielems,0]-1,1])/length

rot[0,0] = 1; rot[3,3] = 1
rot[1,1] = cos_teta; rot[1,2] = sin_teta
rot[2,1] = -sin_teta; rot[2,2] = cos_teta
rot[4,4] = cos_teta; rot[4,5] = sin_teta
rot[5,4] = -sin_teta; rot[5,5] = cos_teta
```

Fonte: os autores, 2024.

No próximo passo, realiza-se a transformação das matrizes de rigidez locais do elemento para o sistema global usando as matrizes de rotação e a equação (Eq. 2.1) como apresentado na Figura 13.

Figura 13: Código para obter matriz de rigidez global do elemento de grelha.

```
#Matriz de Rigidez no Sistema Global (do elemento)
kelem_global = np.zeros((2*dim,2*dim))
kelem_global = np.dot(np.transpose(rot),np.dot(kelem_local,rot))
```

Fonte: os autores, 2024.

Os vetores devido a cargas distribuídas e temperatura para cada elemento são definidos e calculados da mesma forma que as matrizes de rigidez e de rotação, indicando o tamanho e tipo de variável. E também são transformados para o sistema global de coordenadas, conforme as equações (Eq. 2.2) e (Eq. 2.3) apresentadas no fluxograma da Figura 1.

Em seguida, as matrizes de rigidez globais (Eq. 3.1) e vetores de carga da estrutura (Eq. 3.2) são montados a partir das contribuições de cada elemento e as condições de contorno, apresentada no item 4 do fluxograma, são aplicadas de acordo com a técnica de 0 e 1, que consiste em zerar as linhas e colunas das matrizes de rigidez globais nas direções dos graus de liberdade restritos e ao mesmo tempo subtraindo todas as componentes do vetor de força pela multiplicação entre os termos dos coeficientes da matriz de rigidez pelo deslocamento prescrito considerado.

Todo o procedimento anteriormente descrito é mostrado com a implementação apresentada nas Figura 14 Figura 15. Também se observa na Figura 15, que em casos de apoio semirrígidos (representados por molas) os valores das constantes das molas são inseridos na matriz de rigidez no respectivo grau de liberdade.

Figura 14: Montando a matriz de rigidez da estrutura e os vetores de cargas equivalente ao carregamento distribuído e ao efeito de temperatura.

```
#Matriz de rigidez da estrutura no sistema global e Montagem do vetor de forças distribuidas equivalentes
# e devido à temperatura da estrutura global
for inode in range(0,2):
    for idir in range(0,3):
        fequi[conec[ielems,3*inode+idir]] += fequi_global[3*inode+idir]
        ftemp[conec[ielems,3*inode+idir]] += ftemp_global[3*inode+idir]
    for jnode in range(0,2):
        for jdir in range(0,3):
            kestr[conec[ielems,3*inode+idir],conec[ielems,3*jnode+jdir]] += kelem_global[3*inode+idir,3*jnode+jdir]
```

Fonte: os autores, 2024.

Figura 15: Aplicando as condições de contorno e molas na estrutura de grelha.

```
#Aplicação das condições de contorno
for idof in range (0,ngl):
    if cc[idof] == 1:
        festrcc[idof] = 0.0
        for jdof in range (0,ngl):
            kestrcc[idof,jdof] = 0.0
            kestrcc[jdof,idof] = 0.0
        kestrcc[idof,idof] = 1.0
    elif cc[idof] == 2:
        kestrcc[idof,idof] += mola[idof]
```

Fonte: os autores, 2024.

Após a aplicação das condições de contorno, o sistema de equações lineares é resolvido usando a relação aplicada no MRD, dada pelas equações (Eq. 5.1) e (Eq. 6.1), para obter os deslocamentos nodais da estrutura, e as reações de apoio, que são calculadas a partir dos deslocamentos e forças aplicadas tal como apresentado na Figura 16.

Figura 16: Cálculo dos deslocamentos nodais e reações de apoio da estrutura.

```
#Calcula os deslocamentos
displ = np.linalg.solve(kestrcc,festrcc)

#Calcula as reações de apoio
reac = np.dot(kestr,displ)-fequi-ftemp
```

Fonte: os autores, 2024.

Por fim, o cálculo dos esforços internos (forças cortantes, momentos torção e fletor) em cada elemento são calculados com base nos deslocamentos nodais usando a relação da equação (Eq. 7.1) apresentada no fluxograma da Figura 1. Para isso, é necessário obter um vetor de deslocamento no sistema global de cada elemento e em seguida rotacioná-lo para o sistema local conforme pode ser observado nas Figura 17 e Figura 18 respectivamente.

Figura 17: Determina o vetor de deslocamento nodais de cada elemento.

```
displ_elem = np.zeros((2*dim), dtype = float)
for inode in range(0,2):
    for idir in range(0,3):
        idof = conec[ielems,3*inode+idir]
        idof_elem = 3*inode+idir
        displ_elem[idof_elem] = displ[idof]
```

Fonte: os autores, 2024.

Figura 18: Cálculo dos esforços internos nodais de cada elemento de grelha.

```
displ_elem = np.dot(rot,displ_elem)
internal_force = np.dot(kelem_local,displ_elem)-fequi_local-ftemp_local
```

Fonte: os autores, 2024.

RESULTADOS

O algoritmo desenvolvido e apresentado neste trabalho visa determinar com precisão deslocamentos, reações de apoio e esforços internos de estruturas reticuladas de grelhas com diferentes tipos de ligações, condições de contorno e carregamentos.

A fim de validar o código computacional desenvolvido os resultados obtidos a partir da simulação numérica foram confrontados com cálculos manuais e da literatura. Nesse trabalho,

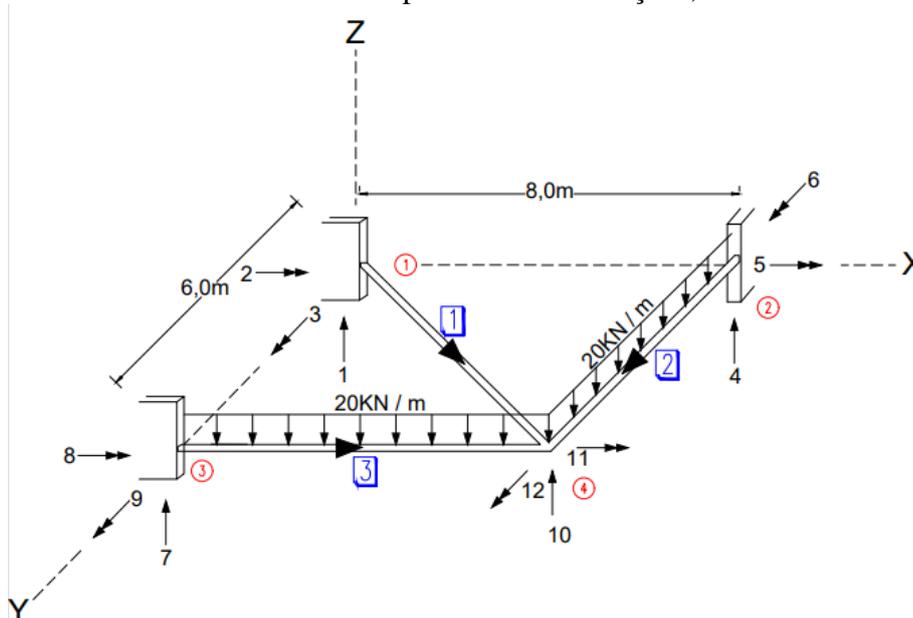
são fornecidos três exemplos de grelha submetida a carregamentos diversos, que são apresentados a seguir.

Exemplo 1

Para validação do código considere a grelha submetida ao carregamento distribuído apresentada na Figura 19 retirada da bibliografia (KASSIMALI , 1999). Para realizar a simulação esta estrutura foi discretizada em três elementos e quatro pontos nodais gerando doze graus de liberdade, dos quais apenas três são livres.

Todos os elementos possuem as mesmas propriedades físicas e geométricas: Módulo de elasticidade do material $E= 200 \text{ GPa}$; Módulo de elasticidade transversal do material $G= 76 \text{ GPa}$; Inércia da seção transversal $I = 3,47 \cdot 10^{-4} \text{ m}^4$; Inércia à torção $J = 1,15 \cdot 10^{-4} \text{ m}^4$.

Figura 19: Estrutura analisada no exemplo 1 com solicitações, dimensões e discretização.



Fonte: Os autores, 2024.

Os valores de deslocamentos, reações de apoio e esforços internos nodais de cada elemento obtidos pelo código desenvolvido, bem como as comparações com os resultados pela referência (KASSIMALI , 1999) são mostrados nas Tabela 1

Tabela 2, Tabela 3 e Tabela 4.

Tabela 1: Deslocamentos nodais.

Deslocamentos da estrutura de Grelha					
Algoritmo desenvolvido			Kassimali (1999)		
GL	Valor	Und.	GL	Valor	Und.
10	-0,0055951	m	1	-0,0055951	m
11	0,0011330	rad	2	0,001133	rad
12	-0,0054856	rad	3	-0,0054856	rad

Fonte: os autores, 2024.

Tabela 2: Reações de Apoio.

Reações de Apoio da estrutura de Grelha					
Algoritmo desenvolvido			Kassimali(1999)		
GL	Valor	Und.	GL	Valor	Und.
1	0,014686	KN	4	0,014686	KN
2	-50,662	KN.m	5	-50,662	KN.m
3	59,140	KN.m	6	59,14	KN.m
4	144,670	KN	7	144,67	KN
5	-445,06	KN.m	8	-445,06	KN.m
6	7,9907	KN.m	9	7,9907	KN.m
7	135,32	KN	10	135,32	KN
8	-12,378	KN.m	11	-12,378	KN.m
9	375,52	KN.m	12	375,52	KN.m

Fonte: os autores, 2024.

Tabela 3: Esforços internos dos elementos obtidos com o código computacional do presente trabalho.

Esforços Internos da estrutura de Grelha – Código desenvolvido								
Elemento 1			Elemento 2			Elemento 3		
GL	Valor	Und.	GL	Valor	Und.	GL	Valor	Und.
1	0,014686	KN	4	144,670	KN	7	135,32	KN
2	-5,0455	KN.m	5	7,9907	KN.m	8	-12,378	KN.m
3	77,709	KN.m	6	445,060	KN.m	9	375,520	KN.m
10	-0,014686	KN	10	-24,668	KN	10	24,683	KN
11	5,0455	KN.m	11	-7,9907	KN.m	11	12,378	KN.m
12	-77,562	KN.m	12	62,952	KN.m	12	67,013	KN.m

Fonte: os autores, 2024.

Tabela 4: Esforços internos segundo Kassimali (1999).

Esforços Internos da estrutura de Grelha – Kassimali (1999)								
Elemento 1			Elemento 2			Elemento 3		
GL	Valor	Und.	GL	Valor	Und.	GL	Valor	Und.
4	0,014686	KN	7	144,67	KN	10	135,32	KN
5	-5,0455	KN.m	8	7,9907	KN.m	11	-12,378	KN.m
6	77,709	KN.m	9	445,060	KN.m	12	375,520	KN.m
1	-0,014686	KN	1	-24,668	KN	1	24,683	KN
2	5,0455	KN.m	2	-7,9907	KN.m	2	12,378	KN.m
3	-77,562	KN.m	3	62,952	KN.m	3	67,013	KN.m

Fonte: os autores, 2024.

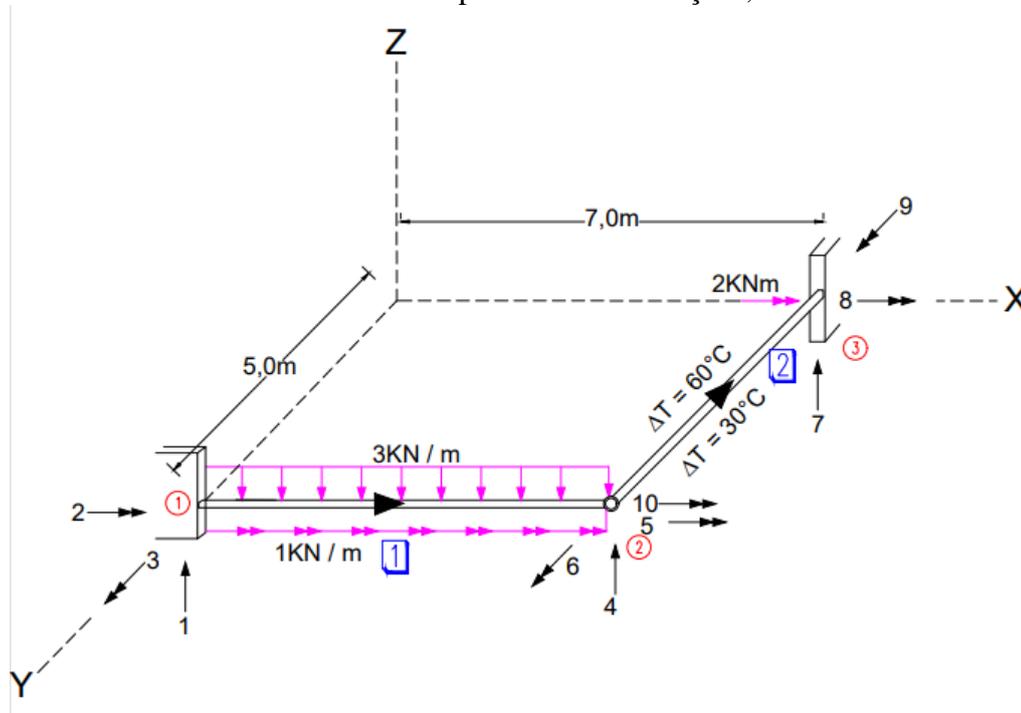
Como pode-se observar nas tabelas, os valores encontrados via código desenvolvido neste trabalho e referência adotada estão consistentes, não apresentando variações entre os valores. Essa consistência nos resultados comprova a eficácia e a validade da análise realizada pelo código nesta aplicação em estruturas de grelha submetida a carregamentos distribuídos.

Exemplo 2

A estrutura de grelha apresentada na Figura 20 possui dois elementos, três nós, dez graus de liberdade, momento fletor aplicado no ponto nodal três, carregamento distribuído linear e de momento torçor, variação de temperatura e uma rótula no nó dois. Todos os elementos da grelha possuem as mesmas propriedades físicas e geométricas com relação a seção transversal, sendo elas: Módulo de elasticidade do material $E= 20\text{GPa}$; Módulo de

elasticidade transversal do material $G = 8,3333 \cdot 10^6$; Inércia da seção transversal $I = 6,4 \cdot 10^{-4} m^4$; Inércia à torção $J = 6,976 \cdot 10^{-4} m^4$; seção transversal de 40x12cm; e coeficiente de dilatação térmica $\alpha = 8 \cdot 10^{-6} / ^\circ C$.

Figura 20 : Estrutura analisada no exemplo 2 com solicitações, dimensões e discretização.



Fonte: os autores, 2024.

Neste exemplo, é possível notar que os graus de liberdade 4, 5, 6 e 10 são livres, enquanto que os demais graus de liberdade são restritos. Essa configuração implica que teremos quatro valores para os deslocamentos nodais e seis valores para as reações de apoio. Salienta-se que devido a ligação interna rotulada, usando o Método da Rigidez Direta, surge o grau de liberdade à rotação 10 no elemento ao qual ela está sendo considerada, que nesse caso, é o elemento 2.

Dessa forma, para verificar e assegurar que os resultados obtidos para deslocamentos, reações de apoio e esforços internos sejam condizentes com o esperado, foram realizadas comparações com cálculos analíticos manuais aplicando os conceitos e formulários da Análise Matricial de estruturas, nos quais foram iguais, e portanto as Tabela 5, Tabela 6 e

Tabela 7 apresentam os resultados obtidos para reações, deslocamentos nodais, e esforços internos dos elementos respectivamente.

Tabela 5: Reações de Apoio para o exemplo 2

Reações de Apoio da estrutura de Grelha		
Grau de liberdade	Valor	Unidade
1	16,345	KN
2	-7,000	KN.m
3	37,723	KN.m
7	4,6553	KN
8	-2,3276	KN.m
9	3,1898	KN.m

Fonte: o autor, 2024.

Tabela 6: Deslocamentos nodais para o exemplo 2

Deslocamentos da estrutura de Grelha		
Grau de liberdade	Valor	Unidade
4	-0,022654	m
5	0,0042145	rad
6	0,0027435	rad
10	0,0075462	rad

Fonte: os autores, 2024.

Tabela 7: Esforços internos nodais dos elementos do exemplo 2.

Esforços Internos da estrutura de Grelha					
Elemento 1			Elemento 2		
Grau de liberdade	Valor	Unidade	Grau de liberdade	Valor	Unidade
1	16,345	KN	4	-4,6553	KN
2	-7,000	KN.m	10	3,1898	KN.m
3	37,723	KN.m	6	0,0000	KN.m
4	4,6553	KN	7	4,6553	KN
5	0,0000	KN.m	8	-3,1898	KN.m
6	3,1898	KN.m	9	23,2760	KN.m

Fonte: os autores, 2024.

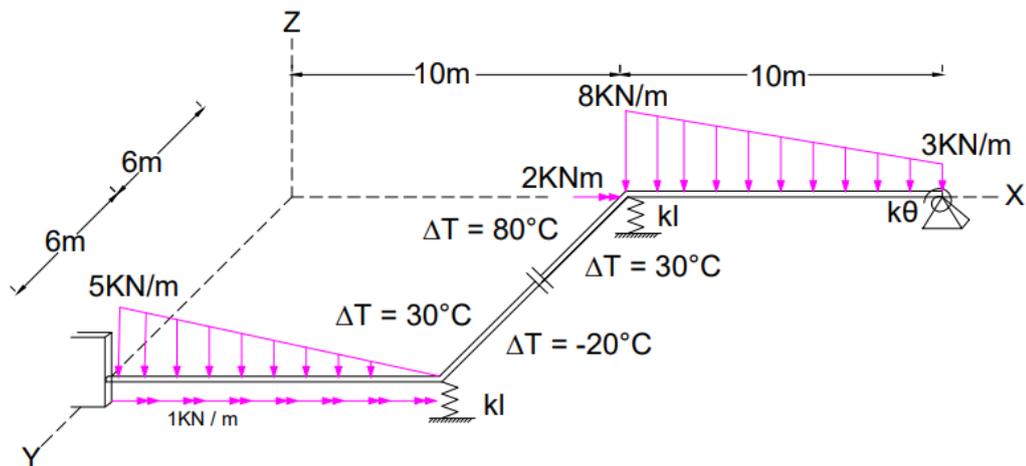
Podemos perceber que os valores são condizentes com o esperado, surgindo valores de reações e deslocamentos nos graus de liberdade previstos. Além disso, os valores apresentados nas tabelas foram conferidos e validados por cálculos manuais de análise estrutural de sistemas de grelha via Método da Rigidez direta. Logo, pode-se afirmar que o código apresenta resultados validos para esse tipo de estrutura analisada no exemplo 2.

Exemplo 3

A Figura 21 mostra uma grelha composta por cinco nós, quatro elementos com as incidências mostradas, dezesseis graus de liberdade, considerando o desacoplamento surgido por causa da presença do engaste móvel. A figura mostra também que a estrutura se encontra submetida aos carregamentos distribuídos linear de modo triangular e trapezoidal, momento fletor, molas lineares nos pontos nodais dois e quatro, variação de temperatura, mola rotacional no nó cinco, engaste móvel no nó três, e carregamento distribuído de momento torçor no elemento um.

Neste exemplo, as propriedades físicas e geométricas dos elementos um e quatro são as mesmas que foram utilizadas no exemplo 2, enquanto que para os elementos dois e três foram aplicados valores referentes ao material e secção transversal iguais a: Módulo de elasticidade do material $E = 250\text{GPa}$; Módulo de elasticidade transversal do material $G = 96,154 \cdot 10^6 \text{ KN/m}^2$; Inércia da secção transversal $I = 6,4 \cdot 10^{-4} \text{ m}^4$; Inércia à torção $J = 6,976 \cdot 10^{-4} \text{ m}^4$; seção transversal de $40 \times 12 \text{ cm}$; e coeficiente de dilatação térmica $\alpha = 11 \cdot 10^{-6} / ^\circ\text{C}$; Coeficiente de rigidez linear $Kl = 1700 \text{ KN/m}$; Coeficiente de rigidez à rotação $K\theta = 9200 \text{ KNm/rad}$.

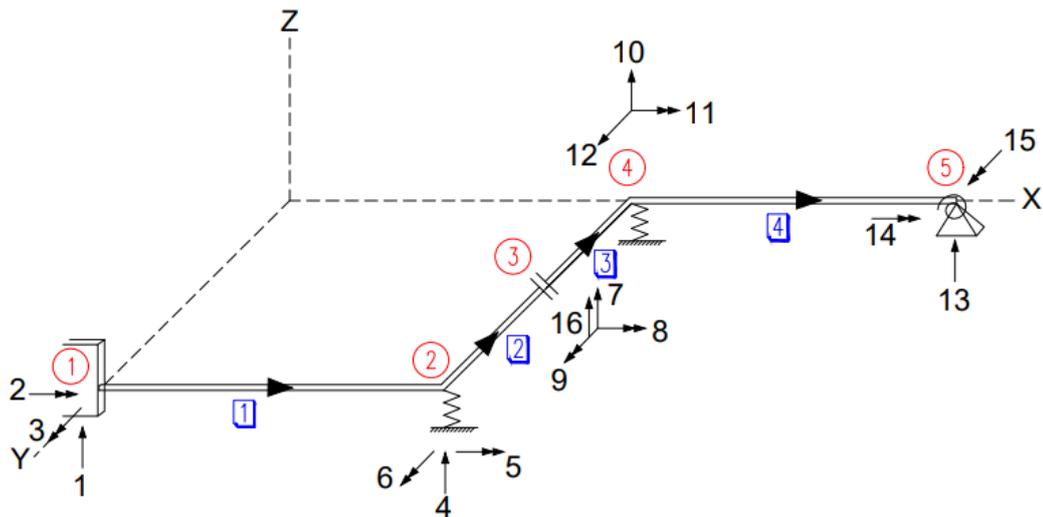
Figura 21: Estrutura analisada no exemplo 3 com solicitações, dimensões e condições de contorno.



Fonte: os autores, 2024.

A Figura 22, apresenta a discretização do sistema estrutural da grelha analisada neste exemplo, onde, tem-se numerado os pontos nodais, elementos, graus de liberdade correspondente a cada nó, bem como a incidência adotada para cada elemento.

Figura 22: Discretização da estrutura analisada no exemplo 3.



Fonte: os autores, 2024.

As

Tabela 8, Tabela 9 e Tabela 10 apresentam os resultados obtidos com o código computacional desenvolvido. De modo semelhante ao exemplo 2, os valores apresentados nas tabelas foram conferidos e validados por meio de cálculos manuais de análise estrutural de sistemas de grelha utilizando o Método da Rigidez Direta.

Tabela 8: Reações de Apoio para o exemplo 3

Reações de Apoio		
Algoritmo desenvolvido		
GL	Valor	Und.
1	19,979	KN
2	-15,529	KN.m
3	37,362	KN.m
4	5,0214	KN
10	29,024	KN
13	25,976	KN
14	3,5293	KN.m
15	-30,674	KN.m

Fonte: os autores, 2024.

Tabela 9: Deslocamentos nodais para o exemplo 3

Deslocamentos da estrutura		
Algoritmo desenvolvido		
GL	Valor	Und.
4	-0,0029538	m
5	-0,012047	rad
6	0,00000449	rad
7	0,045202	m
8	0,0040046	rad
9	-0,000375	rad
10	-0,017073	m
11	-0,004038	rad
12	-0,00075449	rad
15	0,0033341	rad
16	-0,016972	m

Fonte: os autores, 2024.

Tabela 10: Esforços internos nodais dos elementos do exemplo 3

Esforços Internos da estrutura de Grelha – Código desenvolvido											
Elemento 1			Elemento 2			Elemento 3			Elemento 4		
GL	Valor	Und.	GL	Valor	Und.	GL	Valor	Und.	GL	Valor	Und.
1	19,979	KN	4	0,0000	KN	16	0,0000	KN	10	29,024	KN
2	-15,529	KN.m	5	-4,2425	KN.m	8	-4,2425	KN.m	11	-3,5293	KN.m
3	37,362	KN.m	6	-5,5293	KN.m	9	-5,5293	KN.m	12	4,2425	KN.m
4	5,0214	KN	7	0,0000	KN	10	0,0000	KN	13	25,976	KN
5	5,5293	KN.m	8	4,2425	KN.m	11	4,2425	KN.m	14	3,5293	KN.m
6	-4,2425	KN.m	9	5,5293	KN.m	12	5,5293	KN.m	15	-30,674	KN.m

Fonte: os autores, 2024.

Neste exemplo, podemos perceber analisando a

Tabela 8, que o maior valor de reação de apoio ocorreu no grau de liberdade 3, seguido dos graus de liberdade 15 e 10, os quais podem ser devido aos tipos de carregamentos e apoios, onde há maior rigidez e solicitação na estrutura.

Por outro lado, analisando a Tabela 9, nota-se que o maior valor de deslocamento na estrutura ocorreu no grau de liberdade 7, justamente onde existe uma liberdade translacional entre as barras devido ao engaste móvel.

Na Tabela 10, o maior valor de esforço interno geral ocorreu no grau de liberdade 3, que corresponde ao momento fletor no nó inicial do elemento 1. Enquanto que os maiores valores de esforços cortante surgiram no elemento 4.

Outro ponto relevante a ser observado, é que os esforços internos apresentados na Tabela 10, são iguais para os elementos 2 e 3, em que o esforço cortante é nulo em ambos elementos. Isso pode ser devido a concepção do sistema estrutural, bem como dos carregamentos, condições de contornos e ligações internas. Além disso, observa-se que, além desses elementos possuírem as mesmas características relacionadas ao material e a secção transversal, a variação de temperatura é a mesma para os dois elementos.

Por fim, após estas análises, é notório que o algoritmo desenvolvido e apresentado neste trabalho gera resultados consistentes para estruturas de grelhas mais robustas conforme apresentado neste exemplo.

CONCLUSÃO

O desenvolvimento do algoritmo em Python, fundamentado no Método da Rigidez Direta, demonstrou ser uma ferramenta eficaz para a análise de deslocamentos, reações de apoio e esforços internos em estruturas de grelha no regime elástico linear. A implementação seguiu uma metodologia robusta, dividida em pré-processamento, processamento e pós-processamento, garantindo a leitura adequada dos dados, o cálculo eficiente das propriedades estruturais e a geração detalhada dos resultados.

Os resultados obtidos foram comparados com cálculos manuais e dados da literatura, mostrando a consistência e validade da implementação realizada. Nos exemplos testados, os deslocamentos nodais, reações de apoio e esforços internos calculados com o código desenvolvido coincidiram com os valores obtidos por métodos analíticos e outras referências bibliográficas, comprovando a precisão e a confiabilidade da implementação.

Destaca-se que após a implementação o código computacional desenvolvido permite a análise de sistemas estruturais de grelhas submetidas a diferentes tipos de carregamentos, condições de contorno e ligações internas, incluindo apoios rígidos, semirrígidos e articulações internas diversas. Utilizando Python e bibliotecas como *numpy*, o código realiza cálculos complexos de forma rápida e precisa.

REFERÊNCIAS

- CARVALHO, C.; FIGUEIREDO FILHO, J. R. D. **Cálculo e detalhamento de estruturas usuais de concreto armado: segundo a NBR 6118:2014**. 4. ed. São Carlos: EdUFSCar, 2014. 415 p.
- GERE, J.; WEAVER, J. **Matrix Analysis of Framed Structures**. 3. ed. New York: Van Nostrand Reinhold, 1990.
- KASSIMALI, J. **Matrix analysis of structure**. 2. ed. Stamford, USA: Cengage learning, 1999.
- LEET, M.; UANG, C. M.; GILBERT, A. M. **Fundamentos da Análise Estrutural**. 3. ed. Porto Alegre: AMGH, 2010.
- MARTHA, L. **Análise Matricial de Estruturas com orientação a objetos**. 1. ed. Rio de Janeiro: Gen Itc, 2018.
- MARTHA, L. **Análise de Estruturas: Conceitos e Métodos Básicos**. Rio de Janeiro : Grupo GEN, 2022.