



**Extensão de Programa Gráfico para Análise de Treliças e Pórticos
Espaciais via MATLAB e GUI**

Murilo Felix Duailibe Barros Rego Filho

Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de
Janeiro, PUC-Rio

Ênfase: Estruturas

Área: Mecânica Computacional

Orientador: Luiz Fernando Campos Ramos Martha

Rio de Janeiro, Junho 2017

Agradecimentos

Aos meus pais pelo apoio imensurável que deram durante esta longa jornada acadêmica, pelo carinho, e, principalmente, por nunca desistirem de mim.

Aos meus irmãos pelo carinho e companhia.

Ao meu orientador Luiz Fernando Martha, pela confiança que teve em mim e pelo apoio que me deu durante a consecução deste trabalho.

Ao meu amigo Rafael Lopez Rangel, que partilhou comigo os conhecimentos indispensáveis para a feitura deste trabalho.

Ao professor Alessandro Gaio Chimenton, pela ajuda que me forneceu com os tópicos de Álgebra Linear.

Lista de Figuras

Figura 1.1 Níveis de abstração adotados na análise estrutural.....	1
Figura 1.2 Modelo estrutural de pórtico espacial.....	2
Figura 2.1 Tipos diferentes de modelos de análise existentes no LESM 3D.....	5
Figura 2.2 Convenção dos eixos locais adotados no LESM 3D.....	7
Figura 2.3 Orientação positiva dos esforços internos em modelos de treliça plana.....	10
Figura 2.4 Enumeração dos graus de liberdade de um elemento de treliça plana.....	11
Figura 2.5 Orientação positiva dos esforços internos em modelos de pórtico plano.....	12
Figura 2.6 Enumeração dos graus de liberdade de um elemento de pórtico plano.....	13
Figura 2.7 Orientação positiva dos esforços internos em modelos de grelha.....	14
Figura 2.8 Enumeração dos graus de liberdade de um elemento de grelha.....	15
Figura 2.9 Orientação positiva dos esforços internos em modelos de treliça espacial.....	16
Figura 2.10 Enumeração dos graus de liberdade de um elemento de treliça espacial.....	18
Figura 2.11 Orientação positiva dos esforços internos em modelos de pórtico espacial.....	19
Figura 2.12 Enumeração dos graus de liberdade de um elemento de pórtico espacial.....	20
Figura 3.1 Campo de deslocamentos no interior de uma barra para deslocamentos planos.....	26
Figura 3.2 Influência da deformação cisalhante de acordo com comprimento dos elementos...27	
Figura 3.3 Configurações deformadas elementares impostas por deslocabilidades atuantes no nó inicial.....	30
Figura 3.4 Configurações deformadas elementares impostas por deslocabilidades atuantes no nó final.....	31
Figura 3.5 Matriz de rigidez local para elementos de pórtico espacial.....	31
Figura 4.1 Diagrama de atividades simplificado.....	38
Figura 4.2 Fluxograma dos estágios operacionais do LESM.....	39
Figura 5.1 Diagrama de classes do LESM 3D.....	45

Figura 5.2 Tela principal de interface gráfica do LESM 3D.....	57
Figura 5.3 GUI auxiliar para gerenciamento dos materiais.....	58
Figura 5.4 GUI auxiliar para gerenciamento das propriedades das seções transversais.....	59
Figura 5.5 GUI auxiliar para gerenciamento dos pontos nodais.....	59
Figura 5.6 GUI auxiliar para gerenciamento dos elementos de barra.....	60
Figura 5.7 GUI auxiliar para gerenciamento das cargas nodais.....	60
Figura 5.8 GUI auxiliar para gerenciamento dos carregamentos nos elementos.....	61
Figura 5.9 GUI auxiliar para gerenciamento das restrições de apoio e deslocamentos prescritos.....	61
Figura 5.10 Diagrama de sequência simplificado.....	66
Figura 5.11 Diagrama de sequência de cálculo de forças internas.....	72

Lista de Tabelas

Tabela 5.1 Propriedades e métodos da classe Drv.....	46
Tabela 5.2 Propriedades e métodos da classe Anm.....	47
Tabela 5.3 Propriedades e métodos da classe Elem.....	50
Tabela 5.4 Propriedades e métodos da classe Lelem.....	51
Tabela 5.5 Propriedades e métodos da classe Node.....	52
Tabela 5.6 Propriedades e métodos da classe Material.....	52
Tabela 5.7 Propriedades e métodos da classe Section.....	53
Tabela 5.8 Propriedades e métodos da classe Draw.....	55
Tabela 5.9 Propriedades e métodos da classe Print.....	56
Tabela 5.10 Propriedades de cada modelo de análise.....	63

Índice

1	Introdução.....	1
1.1	Organização do Trabalho.....	5
2	Modelos de Elementos Lineares.....	5
2.1	Visão Geral.....	5
2.2	Considerações Gerais.....	6
2.3	Treliça Plana.....	9
2.4	Pórtico Plano.....	12
2.5	Grelha.....	14
2.6	Treliça Espacial.....	16
2.7	Pórticos Espaciais.....	18
3	Formulações Matemáticas para Elementos Lineares.....	20
3.1	Visão Geral.....	22
3.2	Matriz de Transformação por Rotação Genérica.....	22
3.3	Teoria de Vigas.....	25
3.4	Matrizes de Rigidez.....	28
3.5	Reações de Engastamento Perfeito.....	32
3.6	Funções de Forma.....	34
3.7	Deslocamentos Internos em Elementos Carregados.....	36
4	Implementação Computacional.....	37
4.1	Visão Geral.....	37
4.2	Estágios de Implementação de um Programa para Análise de Estrutura.....	37
4.3	O Uso do MATLAB no Desenvolvimento do Programa.....	39
4.4	Paradigma de Programação Orientada a Objeto.....	40
4.5	Paradigma de Programação Dirigida a Eventos.....	43
5	O Desenvolvimento do LESM 3D.....	44
5.1	Visão Geral.....	44
5.2	Classes do LESM.....	44
5.3	Interface Gráfica do Usuário.....	56
5.4	Pré-Processamento de Dados.....	62

5.5 Processamento de Dados.....	64
5.6 Pós-Processamento de Dados.....	73
6 Conclusão.....	80
Bibliografia.....	81
Apêndice A Coeficientes de Rigidez Locais de Elementos Lineares.....	82
Apêndice B Reações de Engastamento Perfeitos para Cargas Distribuídas Linearmente.....	85
Apêndice C Reações de Engastamento Perfeito para Variação de Temperatura.....	90
Apêndice D Funções de Forma.....	94
Apêndice E Deslocamentos Internos para Cargas Linearmente Distribuídas.....	98
Apêndice F Deslocamentos Internos para Variação de Temperatura.....	101

1. Introdução

Uma estrutura pode ser entendida como um conjunto capaz de receber ações externas, absorvê-las internamente e transmiti-las até os seus apoios, onde encontram um sistema de forças externas equilibrante.

Para a elaboração de projetos estruturais eficazes é necessário que o engenheiro entenda todo o processo descrito acima e, para isto, é fundamental o conhecimento dos conceitos da área de Análise de Estruturas para a correta consecução do projeto estrutural.

A análise estrutural corresponde a fase do projeto estrutural onde é feita a idealização do comportamento da estrutura, utilizando todas as teorias físicas e matemáticas resultantes da formalização da engenharia estrutural como ciência.

A análise estrutural moderna trabalha com quatro níveis de abstração com relação a estrutura que está sendo analisada (Martha, 2010). Estes níveis podem ser visualizados na figura 1.1.

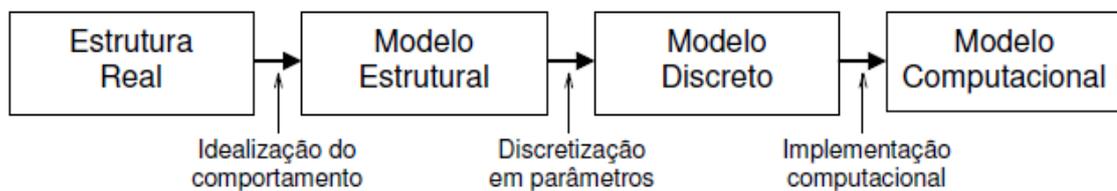


Figura 1.1 – Níveis de abstração adotados na análise estrutural

O primeiro nível representa a estrutura no mundo real.

O segundo nível apresenta o modelo analítico, modelo estrutural, utilizado para representar matematicamente a estrutura.

O terceiro nível apresenta o modelo discreto, concebido dentro das metodologias de cálculo dos métodos de análise, que pode ser resolvido manualmente ou implementado computacionalmente.

O quarto nível é o resultado da implementação computacional.

Para a implementação computacional existem no mercado inúmeros programas de análise estrutural, esses em sua maioria são caros, complexos, além de exigir do engenheiro grande familiarização com a sua interface para possibilitar seu uso pleno, mesmo assim, nem sempre os resultados são os desejados, pois tais programas, regra geral, não possuem código aberto.

Para fugir da mercantilização dos programas de análise estrutural, pouco a pouco, estão surgindo, principalmente em ambientes acadêmicos, programas de código aberto, gratuitos, criados por professores ou alunos, que agem movidos pelo desejo de criar programas acessíveis, didáticos e aperfeiçoáveis pelo uso recorrente dos usuários, que podem adequá-los as suas necessidades.

Dentro dessa visão, no ano de 2016, o então aluno de graduação do curso de Engenharia Civil da PUC-RJ, Rafael Lopez Rangel e o Professor Luiz Fernando Campos Ramos Martha, seu orientador, criaram o programa LESM (Linear Elements Structure Model), que vem a ser um programa gráfico de análises de estruturas, de código aberto, com viés educacional.

Em 2017, o autor do presente trabalho, orientado também pelo Professor Luiz Fernando Campos Ramos Martha, tomou para si a tarefa de implementar no programa LESM os modelos de análise tridimensionais, ou seja, aperfeiçoar o programa antes somente 2D para o 3D.

Temos, portanto, que o presente trabalho objetiva demonstrar como se deu a extensão do LESM para o LESM 3D e, também, detalhar o programa modificado.

Este trabalho é, por assim dizer, uma continuação direta do trabalho Development of a Graphic Program for Structural Analysis of Linear Element de Rafael Lopez Rangel, que em suas considerações finais anteviu a possibilidade da implementação dos modelos de análise tridimensional.

Assim, os objetivos do trabalho que resultou no LESM, foram incorporados ao presente trabalho.

As características do LESM descritas nesse trabalho podem se encontrar datadas devido ao desenvolvimento constante do programa.

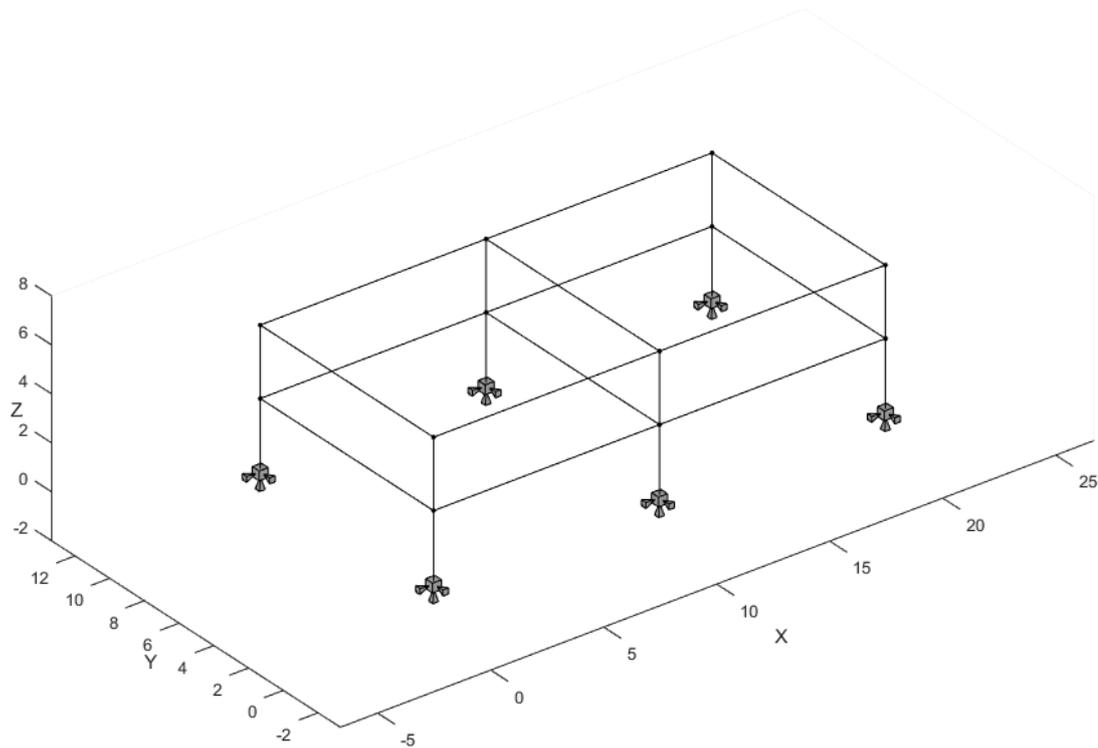


Figura 1.2 – Modelo estrutural de pórtico espacial

1.1 Organização do Trabalho

O presente trabalho é dividido em seis capítulos e seis apêndices.

O capítulo 1 trata da introdução, onde se descreve os motivos que levaram a execução do trabalho.

O capítulo 2 (Modelos de Estruturas Reticuladas) é composto de detalhes das simplificações adotadas pelo LESM, conceitos do método da rigidez direta e detalhes dos modelos de análise implementados pelo programa.

O capítulo 3 (Formulações Matemáticas de Elementos Lineares) é composto por um detalhamento do processo de obtenção de uma Submatriz de rotação que foi utilizada em todo o programa, teorias de vigas implementadas, além dos conceitos de matrizes de rigidez, funções de forma, reações de engastamento perfeito e deslocamentos nodais prescritos.

O capítulo 4 descreve os estados de implementação, a importância do MATLAB na consecução do trabalho, o paradigma de programação orientada a objetos e o paradigma de programação dirigida a eventos.

O capítulo 5 é composto de explicações acerca do funcionamento do programa. Nele é descrito as características de cada uma das classes implementadas no LESM, bem como a interface gráfica do programa e um detalhamento dos diferentes estágios de implementação do programa.

O capítulo 6 apresenta as conclusões obtidas com a consecução deste trabalho e propostas para futuras versões do LESM.

2 Modelos de Estruturas Reticuladas

2.1 Visão Geral

Estruturas reticuladas ou retilíneas são modelos estruturais compostos de elementos unidimensionais, ou seja, são estruturas em que o comprimento prevalece sobre as outras duas dimensões. Estes elementos unidimensionais são chamados de barra e podem ser representados através de seus eixos. A mecânica dos sólidos idealiza o comportamento das barras através de um conjunto de hipóteses sobre seu comportamento cinemático e mecânico, tendo se convencionado que as seções transversais de uma barra que se deformam permanecem planas.

Estruturas reticuladas podem ser classificadas de acordo com o modelo geométrico, o tipo de conectividade entre os elementos e a direção das cargas aplicadas. Os tipos mais comuns de estruturas reticuladas são treliças planas, pórticos planos, grelhas, treliças espaciais e pórticos espaciais. Podemos chamar estes diferentes tipos de modelos estruturas de modelos de análises porque cada um deles recebe um tratamento distinto no processo de análise estrutural. Nas seções seguintes descreveremos as características dos modelos de análises existentes no LESM 3D (treliças planas, pórticos planos, grelhas, treliças espaciais e pórticos espaciais).



Figura 2.1 – Tipos diferentes de modelos de análises existentes no LESM 3D (treliça plana, treliça espacial, pórtico plano, pórtico espacial e grelha)

2.2 Considerações Gerais

Em modelos de estruturas reticuladas, os elementos (ou barras) são definidos por um nó inicial e um nó final. Tais barras podem possuir eixo reto ou curvo e seção transversal ou variável. Os nós podem permitir a rotação relativa de elementos a eles conectados (nós articulados ou rotulados) ou podem impedir a rotação relativa aos elementos a eles conectados (nós rígidos). Em elementos conectados por nós rígidos o ângulo entre as barras na configuração deformada e indeformada é mantido, exigindo a ação de um momento fletor para preservar o ângulo original. Já em elementos conectados por nós articulados a ocorrência de rotação relativa faz que o ângulo na configuração deformada seja diferente do ângulo original da configuração indeformada, fazendo com que o momento fletor seja nulo.

No LESM 3D todos os modelos são compostos por elementos de eixo reto com seção transversal constante e possuem conexões rígidas ou rotuladas em suas extremidades. Os modelos planos são dispostos apenas no plano XY, já os modelos espaciais estão dispostos no espaço XYZ.

No referido programa há dois tipos de sistemas de coordenadas para referencia-los, o sistema de coordenada global, cuja representação se dá nos eixos globais, e o sistema de coordenada local, cuja representação se dá nos eixos locais. Os

eixos globais especificam um ponto ou direção relativos a uma referência fixa, no caso os eixos X, Y e Z globais. Os eixos locais de um elemento, são definidos para todos os tipos de elemento da seguinte maneira:

- O eixo local x é determinado por um segmento de reta com extremidade inicial no nó inicial e extremidade final no nó final.
- O eixo local y é determinado de maneira que seja perpendicular aos eixos locais x e z. Sendo, assim, necessário calcular o produto vetorial do vetor local z com o vetor local x para obter y.
- O eixo local z de um elemento pode ser fornecido pelo usuário ou calculado pelo programa de modo que seja perpendicular ao eixo local x do elemento semelhante à do eixo global z.

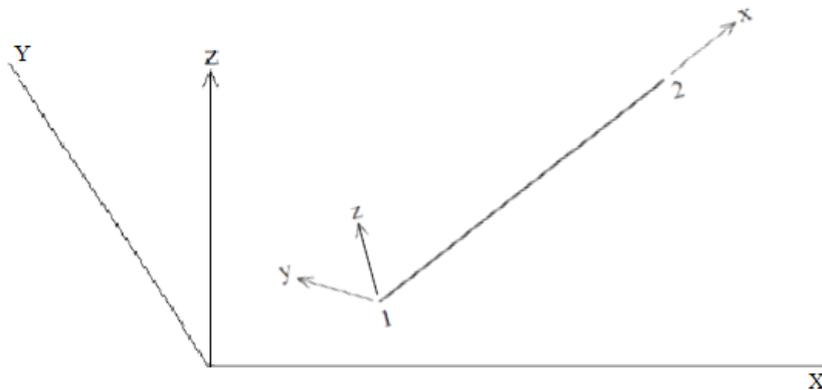


Figura 2.2 – Convenção dos eixos locais adotados no LESM 3D

Os graus de liberdade, ou seja, as possíveis direções de deslocamentos ou rotações nodais, na extremidade de uma barra isolada de um elemento são a combinação dos graus de liberdade dos nós iniciais e finais. Tais direções são identificadas por números que podem ser expressos no sistema de eixos globais ou locais. A ordem de

enumeração dos graus de liberdade para um elemento no sistema local será apresentada na próxima seção para cada tipo de modelo de análise.

Em modelos discretizados por deslocamentos nodais (componente de movimento na direção do grau de liberdade), como no caso do método da rigidez direta, a configuração deformada de um elemento é representada pelos deslocamentos dos nós iniciais e finais. Esses deslocamentos são montados em vetores:

- $\{d\}$: vetor de deslocamentos nodais de um elemento no sistema global.
- $\{d'\}$: vetor de deslocamentos nodais de um elemento no sistema local.

Os esforços internos de um elemento são representados por forças generalizadas, que são deslocamentos e momentos atuantes nas direções do grau de liberdade um elemento para equilibra-lo na configuração deformada. Essas forças são montadas em vetores:

- $\{f\}$: vetor de forças generalizadas de um elemento no sistema global.
- $\{f'\}$: vetor de forças generalizadas de um elemento no sistema local.

É frequentemente necessário a conversão desses vetores de forças e deslocamentos do sistema local para o global e vice-versa, assim uma matriz de transformação por rotação é extremamente necessária no LESM 3D.

Assim, o LESM 3D, utiliza uma matriz auxiliar $[T]$ para compor a matriz de transformação por rotação $[R]$. Essa matriz $[T]$ pode ser facilmente calculada através dos vetores locais unitários do elemento descritos nas coordenadas globais, tal procedimento será descrito no item 3.2 deste trabalho.

Cada modelo de análise existente no LESM 3D possui uma matriz de transformação por rotação particular, composta das submatrizes $[T_{3 \times 3}]$ e $[T_{2 \times 2}]$ da matriz $[T]$, que serão apresentadas nas próximas seções.

As equações 2.1 e 2.2 demonstram como se dá a conversão do vetor nodal de deslocamentos de um sistema de coordenadas para o outro. A conversão dos vetores de forças generalizadas pode ser realizada de forma análoga.

$$\{d'\} = [R] \{d\} \quad (2.1)$$

$$\{d\} = [R]^T \{d'\} \quad (2.2)$$

Quanto ao carregamento existem quatro tipos de carregamento existentes no LESM 3D são eles:

- Carregamento nodal concentrado nas direções dos eixos globais X, Y e Z.
- Carregamento uniformemente distribuído no elemento, abrangendo toda a sua extensão, nas direções dos eixos locais x, y e z ou globais X, Y e Z.
- Carregamento linearmente distribuído no elemento, abrangendo toda a sua extensão, nas direções dos eixos locais x, y e z ou globais X, Y e Z.
- Efeitos de variação transversal em y e z de temperatura.

Em adição a isso, deslocamentos nodais prescritos podem ser especificados apenas no sistema global.

2.3 Treliça Plana

A treliça plana é o mais simples dos modelos estruturais reticulados. Modelos deste tipo apresentam todos os seus elementos interconectados por nós perfeitamente articulados e carregamentos pertencentes a um mesmo plano.

Para as treliças planas tem-se:

- Deslocamentos: D_x e D_y
- Forças: F_x e F_y
- Esforços Internos: N

Os sinais do esforço interno N tem seus sentidos positivos representados na convenção de sinais da figura abaixo:

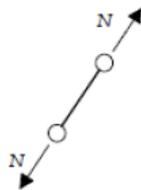


Figura 2.3 – Orientação positiva dos esforços internos em modelos de treliça plana

Há uma regra que deve ser seguida ao conceber este tipo de modelo estrutural, chamada de lei de formação básica das treliças planas. Esta lei estabelece que uma treliça isostática simples continuará sendo isostática ao se acrescentar um novo nó (duas equações), se houver a interligação deste nó a dois nós indeslocáveis (nós de apoio ou nós nas extremidades do elemento já existentes) entre si por meio de duas novas barras (duas incógnitas). Simplificando, a soma do número de reações advindas dos apoios com o número de barras deve ser igual ou maior que duas vezes o número de nós para que a estrutura seja estável.

As seguintes simplificações foram realizadas para a implementação de treliças planas no LESM 3D:

- Os elementos do modelo se encontram no plano XY, podendo ter carregamentos concentrados apenas nas direções globais e carregamentos distribuídos nas direções locais e globais.
- A conexão entre os elementos é sempre rotulada, eliminando as continuidades do momento fletor.
- Os tipos considerados de apoio são: restrição horizontal em X e restrição vertical em Y.
- Modelos deste tipo aceitam deslocamentos prescritos de translação e rotação na direção dos graus de liberdade fixos.
- Modelos deste tipo aceitam cargas concentradas nodais de forças nas direções X e Y, e carregamentos distribuídos nos elementos de forças nas direções X, Y, x e y. Assume-se que apenas o gradiente de temperatura em y influenciará este tipo de modelo.
- A resolução do problema para modelos deste tipo exige como parâmetro geométricos apenas a área da seção transversal.
- Cada nó de um modelo deste tipo possui dois graus de liberdade: translação em X e translação em Y. A enumeração destes graus de liberdade pode ser vista na figura 2.4 abaixo.

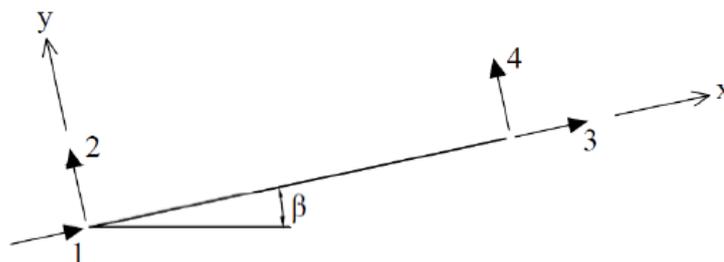


Figura 2.4 – Enumeração dos graus de liberdade de um elemento de treliça plana

- O modelo apresenta a seguinte matriz de transformação por rotação:

$$\text{rot} = \begin{bmatrix} [T_{2 \times 2}] & 0 \\ 0 & [T_{2 \times 2}] \end{bmatrix} \quad (2.3)$$

2.4 Pórtico Plano

Um pórtico plano é um tipo de modelo estrutural reticulado cujos elementos e carregamentos pertencem a um mesmo plano.

Para os pórticos planos tem-se:

- Deslocamentos: D_x , D_y e Θ_z
- Forças: F_x , F_y e M_z
- Esforços Internos: N , Q_y e M_z

Os sinais dos esforços internos N , Q e M tem seus sentidos positivos representados na convenção de sinais da figura abaixo:

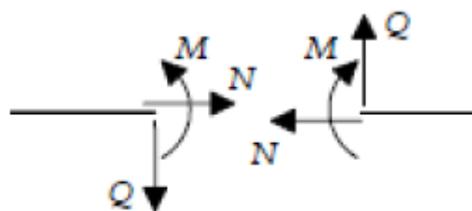


Figura 2.5 – Orientação positiva dos esforços internos em modelos de pórtico plano

As seguintes simplificações foram realizadas para a implementação de pórticos planos no LESM 3D:

- Os elementos do modelo se encontram no plano XY, podendo ter carregamentos concentrados apenas nas direções globais e carregamentos distribuídos nas direções locais e globais.
- A conexão entre os elementos pode ser rígida ou rotulada. Uma conexão rotulada elimina as continuidades do momento fletor. Uma conexão rígida transmite o momento fletor entre as barras e, conseqüentemente, mantém os ângulos da configuração indeformada na configuração deformada.
- Os tipos considerados de apoio são: restrição horizontal em X, restrição vertical em Y e restrição de rotação em Z.
- Modelos deste tipo aceitam deslocamentos prescritos de translação e rotação na direção dos graus de liberdade fixos.
- Modelos deste tipo aceitam cargas concentradas nodais de forças nas direções X e Y e momentos em Z, e carregamentos distribuídos nos elementos de forças nas direções X, Y, x e y e momentos nas direções Z. Assume-se que apenas o gradiente de temperatura em y influenciará este tipo de modelo.
- A resolução do problema para modelos deste tipo exige os seguintes parâmetros geométricos: área da seção transversal, área da seção transversal efetiva ao cisalhamento em y e momento de inércia em z.
- Cada nó de um modelo deste tipo possui três graus de liberdade: translação em X, translação em Y e rotação em Z. A enumeração destes graus de liberdade pode ser vista na figura 2.6 abaixo.

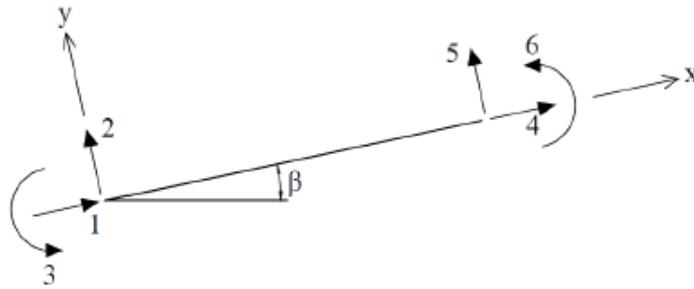


Figura 2.6 – Enumeração dos graus de liberdade de um elemento de pórtico plano

- O modelo apresenta a seguinte matriz de transformação por rotação:

$$\text{rot} = \begin{bmatrix} [T_{3 \times 3}] & 0 \\ 0 & [T_{3 \times 3}] \end{bmatrix} \quad (2.4)$$

2.5 Grelhas

Uma grelha é um tipo de modelo estrutural reticulado cujos elementos pertencem a um plano e os carregamentos atuam perpendicularmente ao plano da estrutura.

Para as grelhas tem-se:

- Deslocamentos: D_z , Θ_x e Θ_y
- Forças: F_z , M_x e M_y
- Esforços Internos: Q_z , T e M_y

Os sinais dos esforços internos N , Q e M têm seus sentidos positivos representados na convenção de sinais da figura abaixo:

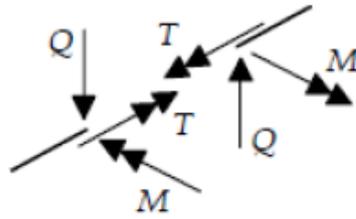


Figura 2.7 – Orientação positiva dos esforços internos em modelos de grelha

As seguintes simplificações foram realizadas para a implementação de grelhas no LESM 3D:

- Os elementos do modelo se encontram no plano XY , enquanto os carregamentos atuam na direção Z .
- A conexão entre os elementos pode ser rígida ou rotulada. Assumimos que uma conexão rotulada elimina a continuidade do momento fletor e, também, do momento torsor.
- Os tipos considerados de apoio são: restrição vertical de translação em Z e restrições de rotação em X e Y . Assumimos que a restrição de rotação em apenas uma direção não pode ser realizada.
- Modelos deste tipo aceitam deslocamentos prescritos de translação e rotação na direção dos graus de liberdade fixos. Assume-se que apenas o gradiente de temperatura em z influenciará este tipo de modelo.
- Modelos deste tipo aceitam cargas concentradas de forças em Z e momentos em X e Y nos nós, e carregamentos distribuídos em Z nos elementos.
- A resolução do problema para modelos deste tipo exige os seguintes parâmetros geométricos: área da seção transversal, área da seção transversal efetiva ao cisalhamento em z , módulo de elasticidade ao cisalhamento e momento de inércia em y .

- Cada nó de um modelo deste tipo possui três graus de liberdade: translação em Z, rotação em X e rotação em Y. A enumeração destes graus de liberdade pode ser vista na figura 2.8 abaixo.



Figura 2.8 – Enumeração dos graus de liberdade de um elemento de grelha

- O modelo apresenta a seguinte matriz de transformação por rotação:

$$\text{rot} = \begin{bmatrix} [T_{3 \times 3}] & 0 \\ 0 & [T_{3 \times 3}] \end{bmatrix} \quad (2.5)$$

2.6 Treliças Espaciais

Uma treliça espacial é um tipo de modelo estrutural reticulado que possui todos os seus elementos interconectados por nós perfeitamente articulados e carregamentos pertencentes ao espaço. Em treliças espaciais ideais todos os elementos estão submetidos somente a esforços axiais N.

Para as treliças espaciais tem-se:

- Deslocamentos: D_x , D_y e D_z
- Forças: F_x , F_y e F_z
- Esforços Internos: N

Os sinais dos esforços internos N tem seus sentidos positivos representados na convenção de sinais da figura abaixo:

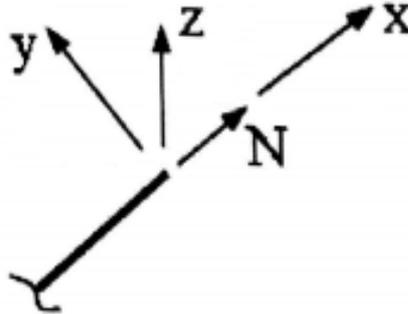


Figura 2.9 – Orientação positiva dos esforços internos em modelos de treliça espacial

Há uma regra que deve ser seguida ao conceber este tipo de modelo estrutural, chamada de lei de formação básica das treliças espaciais. Esta lei estabelece que uma treliça espacial isostática simples continuará sendo isostática ao se acrescentar um novo nó (três equações), se houver a interligação deste nó a três nós indeslocáveis (nós de apoio ou nós nas extremidades do elemento já existentes) entre si através de três novas barras (três incógnitas). Simplificando, a soma do número de reações advindas dos apoios com o número de barras deve ser igual ou maior que três vezes o número de nós para que a estrutura seja estável.

As seguintes simplificações foram realizadas para a implementação de treliças espaciais no LESM 3D:

- Os elementos do modelo se encontram no espaço XYZ, podendo ter carregamentos concentrados apenas nas direções globais e carregamentos distribuídos nas direções locais e globais.
- A conexão entre os elementos é sempre rotulada, eliminando as continuidades dos momentos fletores e, também, do momento torsor.

- Os tipos considerados de apoio são: restrição de translação em X, restrição de translação em Y e restrição de translação em Z.
- Modelos deste tipo aceitam deslocamentos prescritos de translação e rotação na direção dos graus de liberdade fixos.
- Modelos deste tipo aceitam cargas concentradas de forças nas direções globais, e carregamentos de forças nas direções globais e locais. Assume-se que os gradientes de temperatura em y e z influenciarão este tipo de modelo.
- A resolução do problema para modelos deste tipo exige como parâmetro geométricos apenas a área da seção transversal.
- Cada nó de um modelo deste tipo possui três graus de liberdade: translação em X, translação em Y e translação em Z. A enumeração destes graus de liberdade pode ser vista na figura 2.10 abaixo.

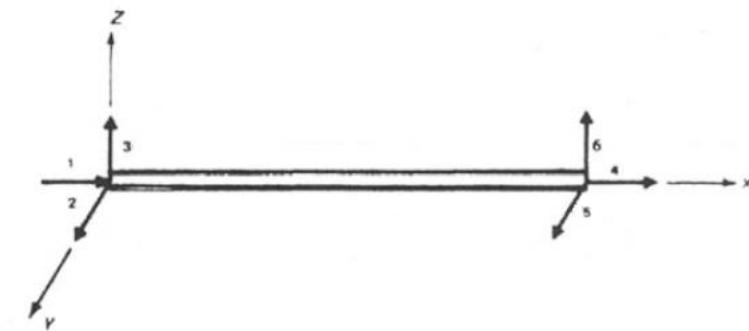


Figura 2.10 – Enumeração dos graus de liberdade de um elemento de treliça espacial

- O modelo apresenta a seguinte matriz de transformação por rotação:

$$\text{rot} = \begin{bmatrix} [T_{3 \times 3}] & 0 \\ 0 & [T_{3 \times 3}] \end{bmatrix} \quad (2.6)$$

2.7 Pórticos Espaciais

O pórtico espacial representa o tipo mais comum de estruturas reticulares, assim, todos os modelos estruturais descritos anteriormente são casos particulares de pórtico espacial.

Para os pórticos espaciais tem-se:

- Deslocamentos: D_x , D_y , D_z , Θ_x , Θ_y e Θ_z
- Forças: F_x , F_y , F_z , M_x , M_y e M_z
- Esforços Internos: N , Q_y , Q_z , T , M_y e M_z

Os sinais dos esforços internos N , Q_y , Q_z , T , M_y e M_z têm seus sentidos positivos representados na convenção de sinais da figura abaixo:

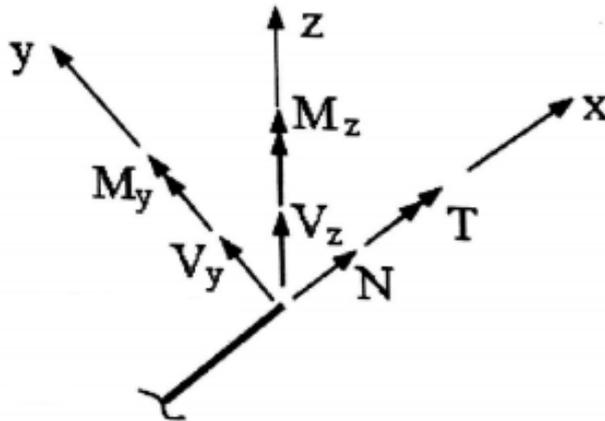


Figura 2.11 – Orientação positiva dos esforços internos em modelos de pórtico espacial

As seguintes simplificações foram realizadas para a implementação de pórticos espaciais no LESM 3D:

- Os elementos do modelo se encontram no espaço XYZ, podendo ter carregamentos concentrados apenas nas direções globais e carregamentos distribuídos nas direções locais e globais.
- A conexão entre os elementos pode ser rígida ou rotulada. Assumimos que uma conexão rotulada elimina as continuidades dos momentos fletores e, também, do momento torsor.
- Os tipos considerados de apoio são: restrição de translação em X, restrição de translação em Y, restrição de translação em Z e restrições de rotação em X, Y e Z. Assumimos que a restrição de rotação em apenas uma direção não pode ser realizada.
- Modelos deste tipo aceitam deslocamentos prescritos de translação e rotação na direção dos graus de liberdade fixos.
- Modelos deste tipo aceitam cargas concentradas de forças em X, Y e Z e momentos em X, Y e Z nos nós, e carregamentos distribuídos de força nos elementos nas direções globais e locais. Assume-se que os gradientes de temperatura em y e z influenciarão este tipo de modelo.
- A resolução do problema para modelos deste tipo exige os seguintes parâmetros geométricos: área da seção transversal, áreas da seção transversal efetivas ao cisalhamento em y e z, módulo de elasticidade ao cisalhamento e momentos de inércia em y e z.
- Cada nó de um modelo deste tipo possui seis graus de liberdade: translação em X, translação em Y, translação em Z, rotação em X, rotação em Y e rotação em Z. A enumeração destes graus de liberdade pode ser vista na figura 2.12 abaixo.

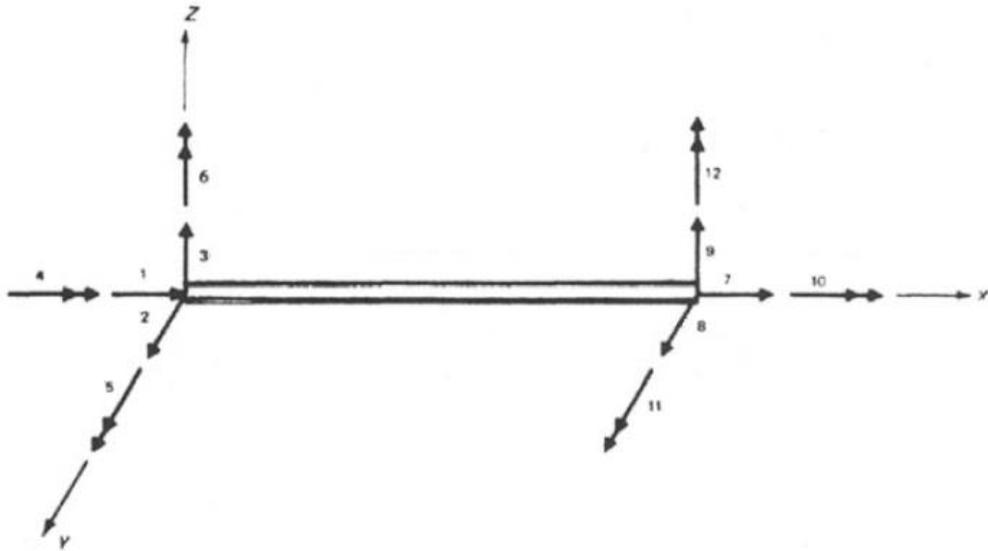


Figura 2.12 – Enumeração dos graus de liberdade de um elemento de pórtico espacial

- O modelo apresenta a seguinte matriz de transformação por rotação:

$$\text{rot} = \begin{bmatrix} [T_{3 \times 3}] & 0 & 0 & 0 \\ 0 & [T_{3 \times 3}] & 0 & 0 \\ 0 & 0 & [T_{3 \times 3}] & 0 \\ 0 & 0 & 0 & [T_{3 \times 3}] \end{bmatrix} \quad (2.7)$$

3 Formulações Matemáticas para Elementos Lineares

3.1 Visão Geral

No capítulo anterior foram apresentados os conceitos básicos de alguns modelos de estruturas lineares e muitas das considerações feitas ao implementar estes modelos de análise no LESM 3D. Já este capítulo é dedicado a explicar os tópicos abaixo:

- Matrizes de Transformação por Rotação
- Teorias de Vigas Utilizadas para Descrever o Comportamento à Flexão dos Elementos Lineares: Teoria de Vigas de Euler-Bernoulli (ou Navier) e Teoria de Vigas de Timoshenko
- Matrizes de Rigidez
- Reações de Engastamento Perfeito
- Funções de Forma
- Deslocamentos Internos em Elementos Carregados

As deduções completas das fórmulas apresentadas neste capítulo podem ser encontradas no livro intitulado “Análise Matricial de Estruturas: Aplicada a Modelos Lineares” – Luiz Fernando Martha, 2017.

3.2 Matriz de Transformação por Rotação Genérica

Para implementar os modelos tridimensionais foi preciso mudar a maneira que o programa LESM efetua as transformações necessárias durante os processos de pré-processamento, processamento e pós-processamento, buscando padronizar tais procedimentos para diminuir o esforço computacional.

Esta padronização foi obtida através de uma matriz de transformação, denominada [T]. Esta matriz é usada por todos os modelos de análise implementados no programa LESM e em diferentes etapas do processo de análise, sendo implementada como uma propriedade do elemento de barra. Descreveremos abaixo o procedimento para obtê-la:

- 1) Calcular o vetor representativo da direção do elemento de barra:

$$\vec{v}_x = [dx_x, dx_y, dx_z] = [(x_2 - x_1), (y_2 - y_1), (z_2 - z_1)] \quad (3.1)$$

- 2) Calcular o produto vetorial de \vec{v}_z (fornecido pelo usuário ou pré-definido pelo programa) com \vec{v}_x :

$$\vec{v}_z = [dz_x, dz_y, dz_z] \quad (3.2)$$

$$\vec{v}_y = \vec{v}_z \times \vec{v}_x = [dy_x, dy_y, dy_z] \quad (3.3)$$

- 3) Assim, obteremos os seguintes vetores na base canônica:

$$\vec{v}_x = dx_x * \hat{x} + dx_y * \hat{y} + dx_z * \hat{z} \quad (3.4)$$

$$\vec{v}_y = dy_x * \hat{x} + dy_y * \hat{y} + dy_z * \hat{z} \quad (3.5)$$

$$\vec{v}_z = dz_x * \hat{x} + dz_y * \hat{y} + dz_z * \hat{z} \quad (3.6)$$

4) Calcular o módulo de cada um destes vetores para normalizá-los futuramente:

$$L_x = \sqrt{dx_x^2 + dx_y^2 + dx_z^2} \quad (3.7)$$

$$L_y = \sqrt{dy_x^2 + dy_y^2 + dy_z^2} \quad (3.8)$$

$$L_z = \sqrt{dz_x^2 + dz_y^2 + dz_z^2} \quad (3.9)$$

5) Normalizar os vetores, obtendo, assim, os vetores unitários que representam os eixos locais da barra:

$$\hat{v}_x = \frac{\vec{v}_x}{L_x} = [v_{x_x} \quad v_{x_y} \quad v_{x_z}] \quad (3.10)$$

$$\hat{v}_y = \frac{\vec{v}_y}{L_y} = [v_{y_x} \quad v_{y_y} \quad v_{y_z}] \quad (3.11)$$

$$\hat{v}_z = \frac{\vec{v}_z}{L_z} = [v_{z_x} \quad v_{z_y} \quad v_{z_z}] \quad (3.12)$$

6) Montar a matriz [T] abaixo:

$$[T] = \begin{bmatrix} v_{x_x} & v_{x_y} & v_{x_z} \\ v_{y_x} & v_{y_y} & v_{y_z} \\ v_{z_x} & v_{z_y} & v_{z_z} \end{bmatrix} \quad (3.13)$$

A matriz [T] é uma matriz de transformação por rotação, levando as coordenadas do sistema local para o global. Sua inversa é equivalente à sua transposta e pode ser usada para fazer a transformação do sistema global para o local. Esta matriz

é utilizada para montar a matriz de rotação [R] e, também, na classe draw e na classe lelem (que serão posteriormente explicadas).

As submatrizes de [T] abaixo são usadas para compor a matriz de transformação por rotação [R] da treliça e dos demais tipos de modelo, respectivamente:

$$[T_{2 \times 2}] = \begin{bmatrix} v_{x_x} & v_{x_y} \\ v_{y_x} & v_{y_y} \end{bmatrix} \quad (3.14)$$

$$[T_{3 \times 3}] = \begin{bmatrix} v_{x_x} & v_{x_y} & v_{x_z} \\ v_{y_x} & v_{y_y} & v_{y_z} \\ v_{z_x} & v_{z_y} & v_{z_z} \end{bmatrix} \quad (3.15)$$

Vale mencionar que a matriz T é equivalente a matriz de transformação por rotação para casos bidimensionais (onde $z_1 = z_2 = 0$), utilizadas no pórtico plano e grelhas, apresentada abaixo:

$$\text{rot} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (3.16)$$

3.3 Teoria de Vigas

Ao realizar análises de primeira ordem, assumindo a hipótese de pequenos deslocamentos, a formulação matemática para elementos sujeitos a efeitos por flexão pode ser baseada em duas teorias distintas: teoria de vigas de Euler-Bernoulli (também chamada de Navier) e teoria de vigas de Timoshenko.

Ambas as teorias adotam uma série de premissas como (vide figura 3.1):

- Hipótese da manutenção das seções transversais planas
- Ângulo de rotação da seção transversal possui valores muito pequenos
- Despreza-se as deformações e tensões totais na direção transversal y da barra
- Considera-se um estado plano de tensões em cada lâmina vertical da barra

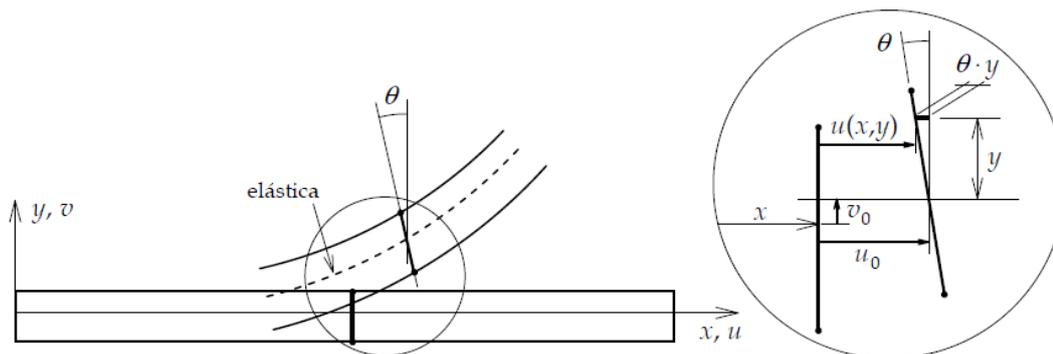


Figura 3.3 – Campo de deslocamentos no interior de uma barra para deslocamentos planos (Martha, 2017)

Porém, estas duas teorias se diferem quanto a consideração da deformação por cisalhamento de uma barra.

Na teoria de vigas de Euler-Bernoulli assume-se que não há deformação por cisalhamento quando o elemento sofre flexão, de maneira que sua seção transversal permanece plana e normal ao eixo longitudinal da barra.

Já na teoria de vigas de Timoshenko a deformação por cisalhamento é considerada de forma aproximada, de modo que elementos submetidos à flexão apresentam seções transversais planas, embora esta não seja normal ao eixo longitudinal da barra.

A influência da deformação cisalhante é menor quanto maior for o elemento. Por conta disto os resultados de análise obtidos pelas duas teorias de vigas são praticamente os mesmos quando o comprimento é muito maior que as dimensões da seção transversal. Para representar isto, um fator utilizado como medida da deformação cisalhante relevante de um modelo é usado nas formulações da teoria de vigas de Timoshenko. Este fator é obtido através da equação 3.17 abaixo, onde E é o módulo de elasticidade, G é módulo de torção a cisalhamento, I é momento de inércia da seção transversal ao longo do eixo fletor, A_s é a área da seção transversal obtida pelo produto da área completa com fator de forma de cisalhamento e L é o comprimento do elemento. Este fator é inversamente proporcional ao quadrado do comprimento do tamanho.

Assim sendo, para elementos delgados o valor é aproximadamente zero, ou seja, a teoria de viga de Euler-Bernoulli fornece resultados próximos aos fornecidos pela teoria de Timoshenko quando o comprimento do eixo da barra é muito maior que as dimensões correspondentes à seção transversal da barra.

$$\Omega = \frac{EI}{GA_z L^2} \quad (3.17)$$

Assumindo uma seção transversal retangular com o fator de forma de cisalhamento de 5/6 e um coeficiente de Poisson de 0.3, a equação anterior recai na equação 3.18, e a variação do fator Ω para relações diferentes de comprimento de elemento e da altura de sua transversal, h, conforme mostrado na figura 3.2.

$$\Omega = 0.26 \left(\frac{L}{h} \right)^{-2} \quad (3.18)$$

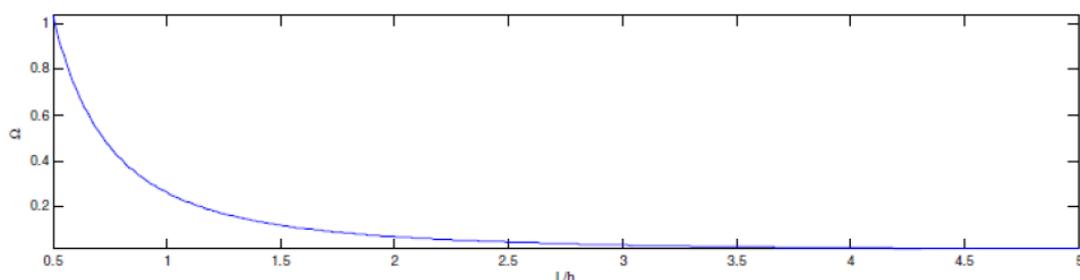


Figura 3.2 – Influência da deformação cisalhante de acordo com o comprimento dos elementos

Pode-se observar que o comportamento ao cisalhamento de um elemento é significativo quando o comprimento é menor que duas vezes a altura da seção transversal, de maneira que os resultados dessas duas teorias começam a se divergir nesse ponto.

Para uma razão L/h baixa a teoria de vigas de Timoshenko é mais precisa. Além disso, o esforço computacional realizado para calcular os resultados através desta teoria é praticamente o mesmo da teoria de vigas de Navier e os resultados obtidos se aproximam mais da realidade, o que faz com que muitos programas estruturais adotem a viga de Timoshenko como o tipo de elemento padrão.

No LESM 3D ambas as teorias podem ser selecionadas para uso no processo de análise de pórticos planos, grelhas e pórticos espaciais, que possuem momento fletor. Em modelos de treliça estas duas teorias podem ser usadas sem distinção, já que é admitido que estes comportamentos não sofrem flexão e as vigas de Navier e Timoshenko são equivalentes no comportamento axial.

3.4 Matrizes de Rigidez

A matriz de rigidez é imprescindível para a análise pelo método da rigidez direta, já que é a responsável pela discretização do comportamento contínuo da estrutura através de uma superposição de soluções cinematicamente determinadas.

O sistema de equações de equilíbrio de um elemento pode ser expresso de forma matricial pela equação 3.19 abaixo, onde $\{f\}$ é um vetor de forças generalizadas, $\{d\}$ é um vetor de deslocamentos nodais e $[k]$ é a matriz de rigidez de um elemento.

$$\{f\} = [k] \{d\} \quad (3.19)$$

O sistema acima é formulado primeiramente no sistema local. Posteriormente deve-se rotacionar as componentes desse sistema para o sistema global. A rotação do vetor de forças generalizadas $\{f\}$ e do vetor de deslocamentos nodais $\{d\}$ é dada pelas equações 2.1 e 2.2. Já a rotação da matriz de rigidez no sistema local para o sistema global é realizada de acordo com a equação 3.20 abaixo, onde $[k_{el}]$ é a matriz de rigidez do elemento no sistema local, $[R]$ é a matriz de transformação por rotação correspondente ao elemento e $[k_{eg}]$ é matriz de rigidez do elemento no sistema global.

$$[k_{eg}] = [R]^T [k_{el}] [R] \quad (3.20)$$

As matrizes de rigidez dos elementos lineares são matrizes quadradas simétricas com dimensão equivalente ao número de graus de liberdade do elemento: quatro para elementos de treliça plana; seis para elementos de pórtico plano, grelha e treliça espacial e doze para elementos de pórtico espacial.

A matriz de rigidez é composta pelos coeficientes de rigidez. Estes podem ser entendidos como força ou momentos que devem atuar em uma extremidade de uma barra isolada na direção de certa deslocabilidades para equilibrá-la, quando uma deslocabilidade unitária é imposta, isoladamente, em uma das extremidades.

Os valores dos coeficientes de rigidez no sistema local podem ser encontrados no apêndice A (Coeficientes de Rigidez Locais de Elementos Lineares), na forma de matrizes desacopladas para cada efeito: coeficientes de rigidez axiais, [kea], coeficientes de rigidez torsionais, [ket], e coeficientes de rigidez por flexão, [kef]. Os coeficientes de rigidez devem seguir as convenções positivas dos graus de liberdade correspondentes ao tipo de modelo, apresentadas no capítulo 2.

$$[k_{el}] = \begin{bmatrix} k_{ea}(1,1) & 0 & k_{ea}(1,2) & 0 \\ 0 & 0 & 0 & 0 \\ k_{ea}(2,1) & 0 & k_{ea}(2,2) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.20)$$

$$[k_{el}] = \begin{bmatrix} k_{ea}(1,1) & 0 & 0 & k_{ea}(1,2) & 0 & 0 \\ 0 & k_{ef}(1,1) & k_{ef}(1,2) & 0 & k_{ef}(1,3) & k_{ef}(1,4) \\ 0 & k_{ef}(2,1) & k_{ef}(2,2) & 0 & k_{ef}(2,3) & k_{ef}(2,4) \\ k_{ea}(2,1) & 0 & 0 & k_{ea}(2,2) & 0 & 0 \\ 0 & k_{ef}(3,1) & k_{ef}(3,2) & 0 & k_{ef}(3,3) & k_{ef}(3,4) \\ 0 & k_{ef}(4,1) & k_{ef}(4,2) & 0 & k_{ef}(4,3) & k_{ef}(4,4) \end{bmatrix} \quad (3.21)$$

$$[k_{el}] = \begin{bmatrix} k_{et}(1,1) & 0 & 0 & k_{et}(1,2) & 0 & 0 \\ 0 & k_{ef}(2,2) & k_{ef}(2,1) & 0 & k_{ef}(2,4) & k_{ef}(2,3) \\ 0 & k_{ef}(1,2) & k_{ef}(1,1) & 0 & k_{ef}(1,4) & k_{ef}(1,3) \\ k_{et}(2,1) & 0 & 0 & k_{et}(2,2) & 0 & 0 \\ 0 & k_{ef}(4,1) & k_{ef}(4,1) & 0 & k_{ef}(4,4) & k_{ef}(4,3) \\ 0 & k_{ef}(3,2) & k_{ef}(3,1) & 0 & k_{ef}(2,4) & k_{ef}(3,3) \end{bmatrix} \quad (3.22)$$

$$[k_{el}] = \begin{bmatrix} kea(1,1) & 0 & 0 & kea(1,2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ kea(2,1) & 0 & 0 & kea(2,2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.23)$$

A composição da matriz de rigidez no sistema local do pórtico espacial é realizada através de uma interpolação dos efeitos obtidos nos casos de pórticos planos e grelhas. As figuras 3.3 e 3.4 abaixo apresentam os coeficientes de rigidez locais quando uma deslocabilidade i é imposta ao elemento de barra.

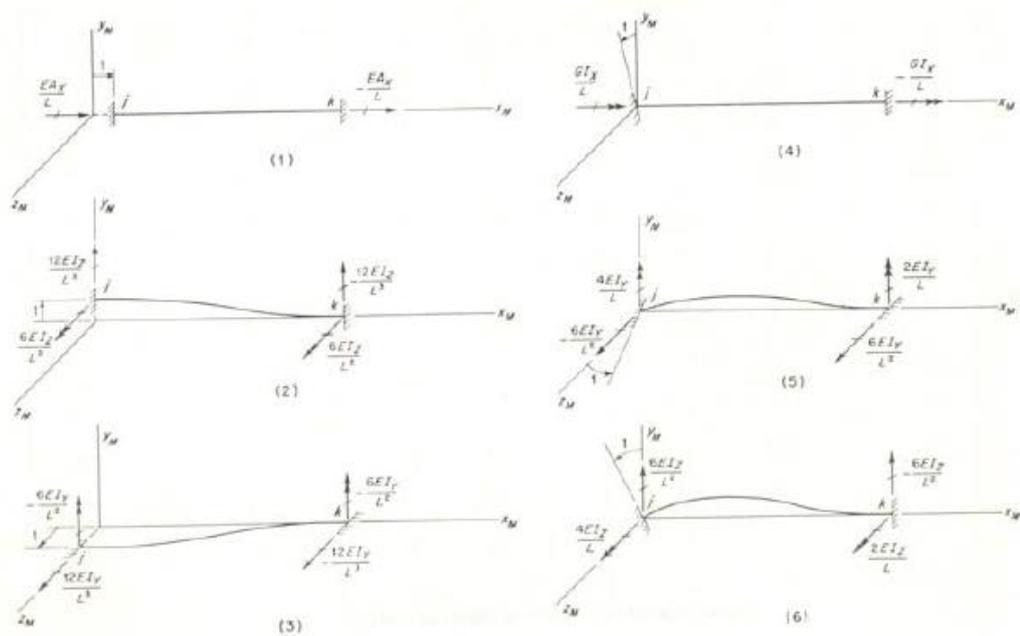


Figura 3.3 – Configurações deformadas elementares impostas por deslocabilidades atuantes no nó inicial

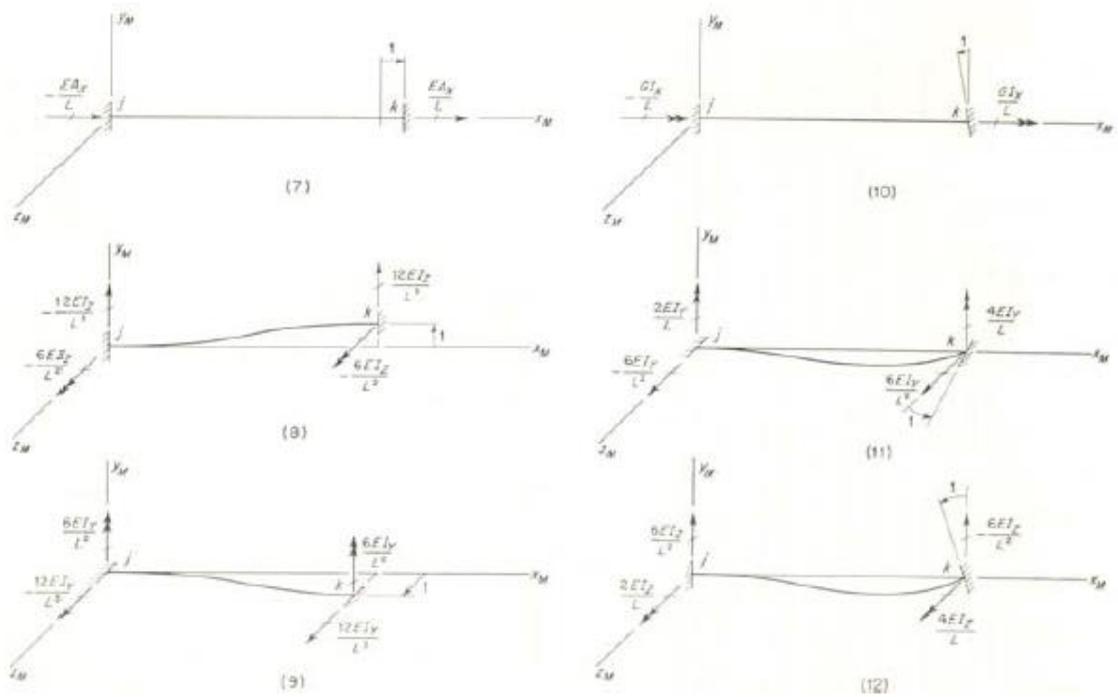


Figura 3.4 – Configurações deformadas elementares impostas por deslocabilidades atuantes no nó final

Através dos coeficientes de rigidez locais obtidos nas figuras 3.3 e 3.4, a matriz de rigidez, de elementos de pórtico espacial, no sistema local para o pórtico espacial foi montada abaixo (vide figura 3.5).

$$[ke] = \begin{bmatrix}
 kea(1,1) & 0 & 0 & 0 & 0 & 0 & kea(1,2) & 0 & 0 & 0 & 0 & 0 \\
 0 & kef_{XY}(1,1) & 0 & 0 & 0 & kef_{XY}(1,2) & 0 & kef_{XY}(1,3) & 0 & 0 & 0 & kef_{XY}(1,4) \\
 0 & 0 & kef_{YZ}(1,1) & 0 & kef_{YZ}(1,2) & 0 & 0 & 0 & kef_{YZ}(1,3) & 0 & kef_{YZ}(1,4) & 0 \\
 0 & 0 & 0 & ket(1,1) & 0 & 0 & 0 & 0 & 0 & ket(1,2) & 0 & 0 \\
 0 & 0 & kef_{YZ}(2,1) & 0 & kef_{YZ}(2,2) & 0 & 0 & 0 & kef_{YZ}(2,3) & 0 & kef_{YZ}(2,4) & 0 \\
 0 & kef_{XY}(2,1) & 0 & 0 & 0 & kef_{XY}(2,2) & 0 & kef_{XY}(2,3) & 0 & 0 & 0 & kef_{XY}(2,4) \\
 kea(2,1) & 0 & 0 & 0 & 0 & 0 & kea(2,2) & 0 & 0 & 0 & 0 & 0 \\
 0 & kef_{XY}(3,1) & 0 & 0 & 0 & kef_{XY}(3,2) & 0 & kef_{XY}(3,3) & 0 & 0 & 0 & kef_{XY}(3,4) \\
 0 & 0 & kef_{YZ}(3,1) & 0 & kef_{YZ}(3,2) & 0 & 0 & 0 & kef_{YZ}(3,3) & 0 & kef_{YZ}(3,4) & 0 \\
 0 & 0 & 0 & ket(2,1) & 0 & 0 & 0 & 0 & 0 & ket(2,2) & 0 & 0 \\
 0 & 0 & kef_{YZ}(4,1) & 0 & kef_{YZ}(4,2) & 0 & 0 & 0 & kef_{YZ}(4,3) & 0 & kef_{YZ}(4,4) & 0 \\
 0 & kef_{XY}(4,1) & 0 & 0 & 0 & kef_{XY}(4,2) & 0 & kef_{XY}(4,3) & 0 & 0 & 0 & kef_{XY}(4,4)
 \end{bmatrix}$$

Figura 3.5 – Matriz de rigidez local para elementos de pórtico espacial

3.5 Reações de Engastamento Perfeito

As reações de engastamento perfeito de uma barra podem ser entendidas como reações de força ou momento que atuam na direção de uma deslocabilidade local de uma barra, sujeita a uma solicitação externa genérica, com as extremidades fixas para equilibrá-la. Tais reações correspondem às cargas equivalentes nodais, provocadas pelos carregamentos internos ou variações de temperatura no elemento na direção oposta.

As formulações das reações de engastamento implementadas no programa LESM 3D consideram um carregamento distribuído linearmente com as respectivas componentes axiais e transversais. Também consideram uma variação de temperatura uniforme ao longo das faces, superior e inferior, dos elementos.

Os valores das reações de engastamento para cada uma das situações implementadas no programa estão presentes no apêndice B (Reações de Engastamento para Carregamentos Distribuídos Linearmente) e C (Reações de Engastamento para Variação de Temperatura).

Assim como os valores dos coeficientes de rigidez, as reações de engastamento perfeito são fornecidas na forma de vetores desacoplados para cada efeito individual: reações de engastamento perfeito axiais, [fea], e transversais, [fef], de barras isoladas. Estas reações obedecem a convenção dos graus de liberdade adotada pelo tipo de modelo correspondente.

As equações abaixo fornecem os vetores com os valores das reações de engastamento para cada tipo de modelo (treliça plana, pórtico plano, grelha, treliça espacial e pórtico espacial, respectivamente).

$$[fel] = \begin{bmatrix} fea(1) \\ fef_{xy}(1) \\ fea(2) \\ fef_{xy}(2) \end{bmatrix} \quad (3.24)$$

$$[\text{fel}] = \begin{bmatrix} \text{fea}(1) \\ \text{fef}_{xy}(1) \\ \text{fef}_{xy}(2) \\ \text{fea}(2) \\ \text{fef}_{xy}(3) \\ \text{fef}_{xy}(4) \end{bmatrix} \quad (3.25)$$

$$[\text{fel}] = \begin{bmatrix} \text{fea}(1) \\ \text{fef}_{xz}(2) \\ \text{fef}_{xz}(1) \\ \text{fea}(2) \\ \text{fef}_{xz}(4) \\ \text{fef}_{xz}(3) \end{bmatrix} \quad (3.26)$$

$$[\text{fel}] = \begin{bmatrix} \text{fea}(1) \\ \text{fef}_{xy}(1) \\ \text{fef}_{xz}(1) \\ \text{fea}(2) \\ \text{fef}_{xy}(3) \\ \text{fef}_{xz}(3) \end{bmatrix} \quad (3.27)$$

$$[\text{fel}] = \begin{bmatrix} \text{fea}(1) \\ \text{fef}_{xy}(1) \\ \text{fef}_{xz}(1) \\ 0 \\ \text{fef}_{xz}(2) \\ \text{fef}_{xy}(2) \\ \text{fea}(2) \\ \text{fef}_{xy}(3) \\ \text{fef}_{xz}(3) \\ 0 \\ \text{fef}_{xz}(4) \\ \text{fef}_{xy}(4) \end{bmatrix} \quad (3.28)$$

Por vezes é necessário que estes vetores sejam convertidos para o sistema global. Isto é realizado usando matriz de transformação por rotação do elemento conforme mostrado na equação abaixo, onde [fel] é o vetor das reações de engastamento no sistema local e [feg] é o vetor das reações de engastamento no sistema global.

$$[feg] = [R]^T [fel] \quad (3.29)$$

3.6 Funções de Forma

As funções de forma descrevem a configuração deformada de um elemento sujeito a um deslocamento nodal simples e unitário conferindo deslocamentos transversais ao longo do eixo longitudinal. Essas funções não são usadas no método da rigidez direta, mas elas são necessárias para representar graficamente a configuração deformada de pórticos planos, grelhas e pórticos espaciais. Modelos de treliça não fazem uso das funções de forma porque permanecem retas, logo a configuração deformada pode ser obtida somente pelas coordenadas dos deslocamentos nodais.

- $N_i(x)$: Função de Forma do Elemento: Função que descreve a configuração deformada de um elemento isolada quando um deslocamento nodal simples e unitário d_i é imposto na direção do grau de liberdade i .

Com as funções de forma, a configuração deformada de um elemento isolado pode ser descrita pela interpolação dos deslocamentos nodais inicial e final. As equações 3.30 e 3.31 fornecem o deslocamento axial, $u(x)$, e transversal, $v(x)$, de um elemento em termos de suas funções de forma.

$$u(x) = Na_1(x) * d1 + Na_2(x) * d7 \quad (3.30)$$

$$v_y(x) = Nf_{xy_1}(x) * d2 + Nf_{xy_2}(x) * d5 + Nf_{xy_3}(x) * d8 + Nf_{xy_4}(x) * d11 \quad (3.31)$$

$$v_z(x) = Nf_{xz_1}(x) * d3 + Nf_{xz_2}(x) * d6 + Nf_{xz_3}(x) * d9 + Nf_{xz_4}(x) * d12 \quad (3.32)$$

Essas equações podem ser expressas na forma matricial, conforme demonstrado nas equações 3.33, 3.34 e 3.35, onde $[N]$ é a matriz de funções de forma. Vale mencionar que a matriz da equação 3.34 corresponde a matriz utilizada em pórticos planos e grelhas, enquanto a matriz da equação 3.35 corresponde a matriz utilizada em pórticos espaciais.

$$\begin{bmatrix} u(x) \\ v(x) \end{bmatrix} = \begin{bmatrix} Na_1(x) & 0 & 0 & Na_4(x) & 0 & 0 \\ 0 & N_{f_1}(x) & N_{f_2}(x) & 0 & N_{f_3}(x) & N_{f_4}(x) \end{bmatrix} * \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} \quad (3.33)$$

$$\begin{bmatrix} u(x) \\ v_y(x) \\ v_z(x) \end{bmatrix} = \begin{bmatrix} Na(1) & 0 & 0 & 0 & 0 & 0 & Na(2) & 0 & 0 & 0 & 0 & 0 \\ 0 & N_{f_{xy}}(1) & 0 & 0 & N_{f_{xy}}(2) & 0 & 0 & N_{f_{xy}}(3) & 0 & 0 & N_{f_{xz}}(4) & 0 \\ 0 & 0 & N_{f_{xz}}(1) & 0 & 0 & N_{f_{xz}}(2) & 0 & 0 & N_{f_{xz}}(3) & 0 & 0 & N_{f_{xz}}(4) \end{bmatrix} * \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \end{bmatrix} \quad (3.34)$$

$$\{u\} = [N] \{d\} \quad (3.35)$$

As expressões de funções de forma estão presentes no apêndice D (Funções de Forma) para cada extremidade contínua de elemento.

3.7 Deslocamentos Internos em Elementos Carregados

Em modelos de pórticos planos, grelhas e pórticos espaciais a configuração deformada de um elemento é influenciada por deslocamentos internos provocados por carregamentos e variação de temperatura, além dos deslocamentos nas extremidades.

As expressões que descrevem as componentes de deslocamentos axiais e transversais para estes dois efeitos estão presentes no apêndice E (Deslocamentos Internos para Cargas Linearmente Distribuídas) e apêndice F (Deslocamentos Internos para Variação de Temperatura). Para determinar o deslocamento total em qualquer seção transversal, tais expressões devem ser adicionadas aos deslocamentos internos em termos das funções de forma e deslocamentos nodais.

4 Implementação Computacional

4.1 Visão Geral

Os métodos de Análise de Estruturas podem ser divididos em dois diferentes grupos, o grupo dos métodos clássicos e o grupo dos métodos matriciais, dentre os dois o matricial é o mais adequado para implementação computacionais, já que o uso de matrizes em linguagens de programação de alto nível permite a automatização de processos.

Este capítulo apresentará alguns conceitos de Ciência da Computação que foram usados no LESM e que podem ser usadas para o desenvolvimento de qualquer outro programa de análise estrutural.

4.2 Estágios de Implementação de um Programa para Análise de Estruturas

Um programa de análise estrutural possui três estágios típicos: pré-processamento, processamento e pós-processamento.

O estágio de pré-processamento consiste na construção do modelo a partir dos dados de entrada captados, sendo nesta etapa que os objetos que representam os materiais, seções, nó, elementos e cargas são criados ou construídos. Esta captação de dados pode ser realizada através de arquivos de texto ou numa interface gráfica que permita o modelamento gradual da estrutura.

O estágio de processamento de dados consiste nos cálculos realizados através de procedimentos do método da rigidez direta. Além da solução numérica do problema estrutura, há também o armazenamento de resultados. A figura 4.1 apresenta o diagrama de atividades simplificado correspondente a este estágio. Este procedimento será melhor detalhado no item 5.5 (Processamento de Dados).

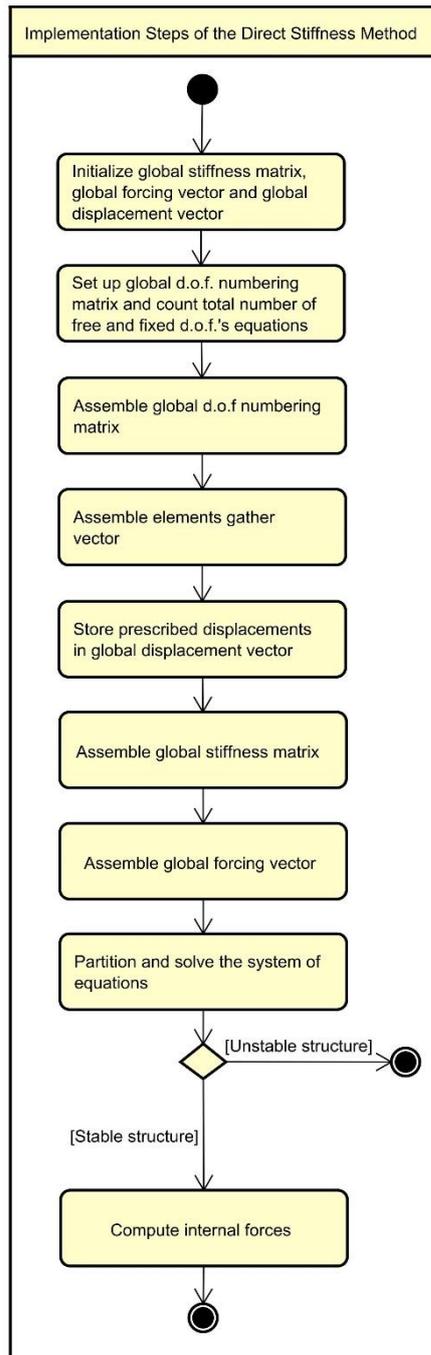


Figura 4.1 – Diagrama de atividades simplificado (procedimentos existentes no estágio de processamento)

O estágio de pós-processamento consiste na exibição dos dados de saída através de resultados textuais e gráficos. Os resultados gráficos a serem exibidos são a configuração deformada e o diagrama de esforços internos, assim, nesta etapa, é necessário a realização dos cálculos dos esforços internos da estrutura (vide figura 4.2).

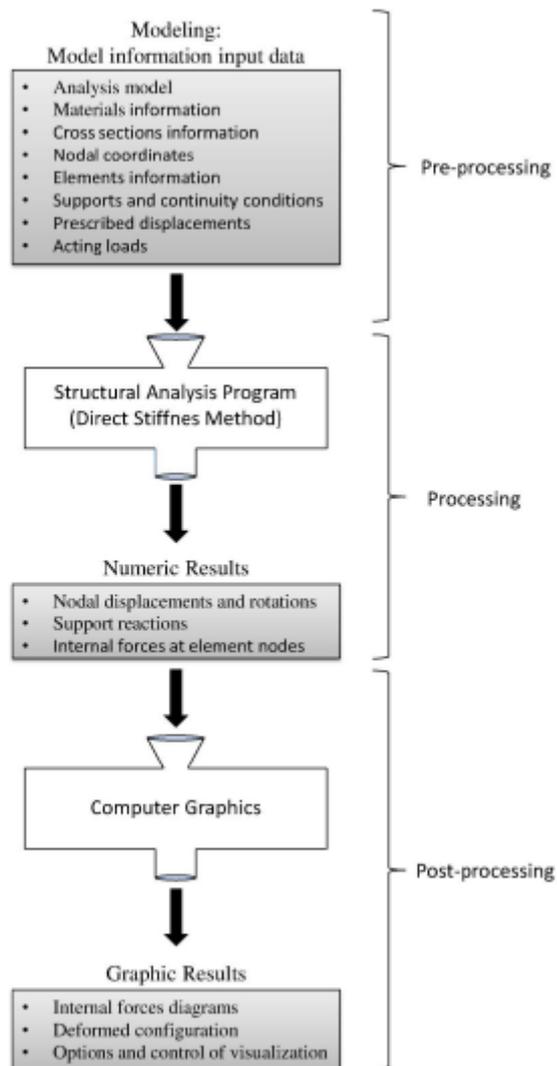


Figura 4.2 – Fluxograma dos estágios operacionais do LESM

4.3 O Uso do MATLAB no Desenvolvimento do Programa

O MATLAB (abreviação para MATrix LABoratory)) é, essencialmente, um programa otimizado para a execução de cálculos científicos e de engenharia. Um de seus grandes diferenciais é a linguagem MATLAB e o acesso a uma ampla biblioteca de funções predefinidas para que a programação técnica se torne mais fácil e eficiente. Tais diferenciais possibilitam resolver problemas técnicos com muito mais facilidade do que as linguagens tradicionais como C e Fortran.

A principal vantagem do MATLAB perante as demais linguagens é a fácil utilização de matrizes que o programa oferece, já que todo número no programa é uma matriz. Um número escalar no MATLAB é uma matriz 1x1. Apenas isso já seria o suficiente para optar-se por este programa, já que o método da rigidez direta é um método iminentemente matricial. Além disso, é possível citar ainda algumas outras vantagens para a escolha desse programa na confecção do LESM como:

- Linguagem de alto nível: A linguagem MATLAB é uma linguagem de nível altíssimo, ou seja, sua linguagem se aproxima da linguagem humana e se distancia da linguagem da máquina. Tal característica facilita o viés educacional que este programa propõe;
- Interface gráfica do usuário: O MATLAB possui ferramentas que possibilitam construir interativamente uma interface gráfica de usuário para seus programas;
- Funções pré-definidas: O MATLAB possui uma grande biblioteca de funções pré-definidas, tais funções representam soluções testadas e empacotadas para diversas tarefas técnicas;
- Desenhos Independentes de Dispositivos: O MATLAB possui muitos comandos para desenhos e imagens, estes podem ser exibidos em qualquer dispositivo de saída gráfica suportado pelo computador do usuário;
- O Compilador MATLAB: A existência de um compilador MATLAB separado permite a compilação de um código MATLAB como um programa efetivamente executável, assim pode-se converter em um programa executável facilmente distribuível.

Podemos citar como principais desvantagens no uso do MATLAB o fato de sua linguagem própria ser mais lenta do que as linguagens compiladas e o alto custo do programa.

4.4 Paradigma de Programação Orientada a Objetos

Diferentemente do paradigma de computação estruturada, onde o código é estruturado como uma sequência de procedimentos a serem realizados, o paradigma da

programação orientada a objetos é baseado, obviamente, nos objetos, instâncias diretas da classe.

A classe é responsável pela criação da funcionalidade de um programa. Cada classe é composta por todas as informações pertinentes ao estado de um objeto (propriedades) e, também, por suas respectivas funções pertinentes ao comportamento de um objeto (métodos). Quando um atributo precisa de certa configuração padrão, ela deve ser especificada ainda no começo da definição da classe, ou dentro dos blocos das propriedades ou métodos. Toda classe pode ser entendida como um protótipo para um determinado objeto.

Vale mencionar que há dois métodos que devem sempre ser implementados em uma definição de classe, o método construtor e o método destrutor. O primeiro deve ter o mesmo nome da classe e é necessário para criar objetos de uma classe através da prescrição de valores de suas propriedades. O segundo, método destrutor, é usado para limpar os objetos, deletando os seus valores.

O objeto, instância direta da classe, é responsável pelo controle do funcionamento do programa. Cada objeto tem seu estado e comportamento controlados pelos métodos definidos pela classe correspondente.

A classe consiste numa ideia, enquanto o objeto representa uma entidade existente no mundo real. Resumidamente, a classe apresenta um conceito abstrato, possui atributos e ações, enquanto o objeto apresenta um conceito concreto, possui valores e executa ações.

Uma das principais vantagens do paradigma de programação orientada a objetos é a sua enorme flexibilidade, possibilitando programas com códigos mais simples e maior facilidade na manutenção e futuras expansões do programa. Estas particularidades permitiram a extensão do LESM para o LESM 3D.

Outras características também foram fundamentais para escolha do paradigma de Programação Orientada ao Objeto (OOP), na implementação do LESM 3D, como:

- Abstração: É o princípio no qual certas classes não podem ser instanciadas, gerando objetos. Estas classes, classes abstratas, consistem em um modelo para

as classes derivadas dela, subclasses. Assim, todas as definições da classe abstrata devem ser implementadas na classe derivada;

- Herança: É o princípio que permite às classes herdar o estado e o comportamento de outra classe. Este princípio é altamente importante devido à redução e organização que confere ao código, facilitando, inclusive, implementações futuras. Exemplo disso se dá na implementação dos modelos de análise para treliça espacial e pórtico espacial que foram criadas a partir da classe base *Anm*. Vale mencionar que configurações da superclasse afetam as classes existentes;
- Polimorfismo: é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que tem a mesma identificação (assinatura), mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. Esse mecanismo é fundamental na programação orientada a objetos, permitindo definir funcionalidades que operem genericamente com objetos, abstraindo-se de seus detalhes particulares quando esses não forem necessários;
- Encapsulação: é o princípio de projeto pelo qual cada componente de um programa deve agregar toda a informação relevante para sua manipulação como uma unidade (uma capsula). Aliado ao conceito de ocultamento de informação é um poderoso mecanismo de programação orientada a objeto.

O acesso às propriedades e aos métodos da classe podem ser controlados via propriedades de acessibilidade, permitindo três configurações diferentes: público (acessível a todos), privado (acesso restrito) e protegido (acesso só é permitido para certas exceções). A acessibilidade de todas as propriedades e métodos no LESM é pública.

Há também uma diferenciação nos tipos de métodos, método abstrato e método estático, implementados no LESM. O primeiro é usado para declarar métodos da classe abstrata, enquanto o segundo é usado para especificar métodos que não dependem do objeto.

Vale mencionar ainda que, diferentemente das linguagens de programação usuais, o MATLAB apresenta dois tipos de classe diferentes, as classes do tipo Value

(padrão), quando objetos deste tipo são atribuídos ou passados para uma função criando uma cópia dos dados, e classes do tipo Handle, quando objetos deste tipo são atribuídos ou passados para uma função criando uma referência.

No LESM, por sua vez, todas as classes são handle e, portanto, são derivadas de uma classe abstrata embutida do MATLAB chamada de `matlab.mixinSetGet`. Classes derivadas desta classe embutida herdam os métodos `set` e `get`, responsáveis por receber ou retornar valores de uma propriedade específica. Tais métodos são chamados quando uma propriedade é referenciada usando o operador ponto e elas devem ser implementadas para cada propriedade da classe.

4.5 Paradigma de Programação Dirigida a Eventos

O paradigma de programação orientada a evento segue um fluxo de controle onde a execução dos algoritmos é condicionada a algo ocorrido anteriormente (evento). Esses eventos podem ocorrer através de clique de mouse, uso de checkbox, etc. Tais eventos determinam o fluxo do programa ao chamar funções específicas, chamadas funções callback, com a implementação do código sendo responsável por um evento específico.

Este tipo de paradigma é amplamente utilizado no desenvolvimento de interfaces gráficas, embora possa ser usado em ambientes como banco de dados.

A interface do LESM foi criada usando o Graphical User Interface Development (GUIDE), uma ferramenta que permite criar objetos de controle de interface com usuário, `UIControl`, em janelas gráficas, para implementar caixas de diálogo no estilo “Visual Basic”. O MATLAB gera automaticamente um arquivo estruturado com as respectivas funções de callback relacionadas às ações que podem ser executadas em cada um dos componentes adicionados à interface.

5 O Desenvolvimento do LESM 3D

5.1 Visão Geral

Nos capítulos anteriores apresentou-se alguns conceitos fundamentais das áreas de Análise de Estruturas e Ciência da Computação necessários para entender a implementação de um programa de análise estrutural, incluído comentários de como esses conceitos são considerados e implementados no LESM 3D. Este capítulo combina estas duas áreas e descreve o desenvolvimento dos estágios operacionais e a interface do programa. Também apresenta uma descrição completa de todas as classes criadas com o paradigma OOP.

5.2 Classes do LESM 3D

O programa é dividido em 9 classes, sendo 7 delas relacionadas ao estágio de processamento, para criar um modelo estrutural e calcular os resultados da análise. As outras duas são associadas ao estágio de pós-processamento, para desenhar ou imprimir os resultados na tela do computador.

Como mencionado no capítulo 4 (implementação computacional), a acessibilidade aos atributos de todas as propriedades e métodos no LESM 3D é pública, isso permite que as propriedades de todos os objetos sejam acessadas e modificadas por métodos de qualquer objeto ou funções de qualquer arquivo, como consequência disso os objetos podem interagir facilmente entre si e o código torna-se mais limpo e simples de entender.

A chamada de uma propriedade ou método pode ser feita com o operador ponto, que implicitamente chama os métodos set ou get, herdados da classe própria do MATLAB responsáveis pela classe Handle. Fazendo as chamadas do programa como

classes handle, é possível referenciar objetos ao invés de criar cópias para serem passadas para métodos e funções.

O fluxograma da figura 5.1 ilustra a interação entre todas as classes implementadas através de um diagrama de classes simplificado do LESM 3D. Cada caixa representa um objeto de uma classe sem a distinção se este é uma superclasse ou subclasse. As classes Draw e Print pertencem ao estágio de pós-processamento do programa, enquanto as demais classes estão relacionadas ao estágio de processamento do programa.

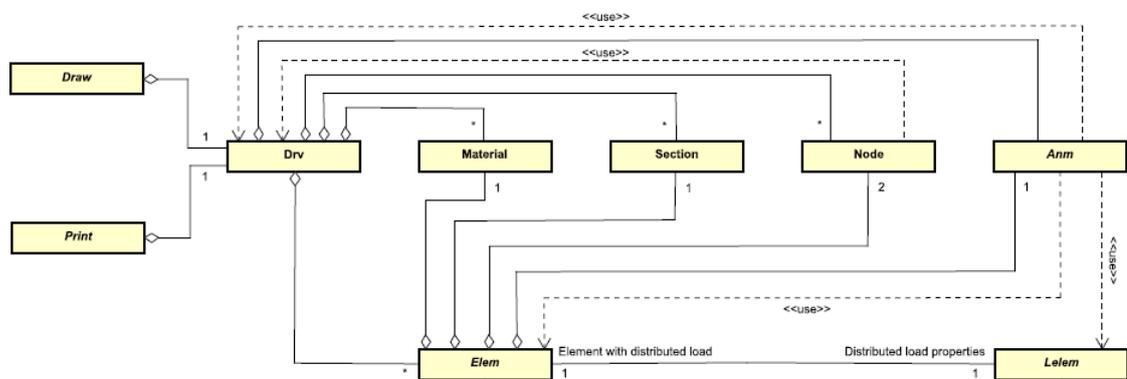


Figura 5.1 - Diagrama de classes do LESM 3D

Nas seções subsequentes serão descritas as características de cada classe, através de tabelas contendo o nome destas, seguido de uma breve explicação de suas propriedades e métodos. Optou-se por não incluir os métodos set, get e clean, já que estes possuem as mesmas funcionalidades em todas as classes e foram previamente explicados.

5.2.1 Classe Drv (Driver)

Esta classe comanda o processo de análise do modelo estrutural no LESM 3D. Suas propriedades guardam informações gerais acerca do modelo estrutural e seus métodos correspondem a procedimentos do método da rigidez direta. Um único objeto

é criado durante o estágio de processamento, já que todos os dados do modelo podem ser acessados usando as propriedades de um objeto dessa classe.

Classe Drv	
Propriedades	Descrição
anm	Ponteiro para objeto da classe Anm
materials	Vetor de ponteiros para objetos da classe Material
sections	Vetor de ponteiros para objetos da classe Section
nodes	Vetor de ponteiros para objetos da classe Node
elems	Vetor de ponteiros para objetos da classe Elem
K	Matriz de rigidez global
F	Vetor de forças globais
D	Vetor de deslocamentos globais
ID	Matriz com graus de liberdade enumerados
nmat	Número dos materiais
nsec	Número das seções transversais
np	Número dos pontos nodais
nel	Número dos elementos
neq	Número das equações
neqfree	Número de equações de graus de liberdade livres
neqfixed	Número de equações de graus de liberdade fixos
Métodos Estáticos	Descrição
openFile	Abre um arquivo de entrada
Métodos Públicos	Descrição
fictRotConstraint	Inserir ou remove uma limitação de rotação fictícia nos nós
dimKFD	Dimensiona a matriz global de rigidez, o vetor de forças e o vetor de deslocamentos
assembleDOFNum	Monta a matriz com graus de liberdade enumerados
assembleGle	Monta o vetor de espalhamento de um elemento
gblMtx	Monta a matriz de rigidez global
assembleElemMtx	Inserir a matriz de rigidez de um elemento à matriz de rigidez global
elemLoads	Adiciona os carregamentos nodais equivalentes de um elemento ao sistema global de forças
assembleENL	Inserir o vetor de cargas nodais equivalentes ao vetor de forças globais
solveEqnSystem	Particiona e resolve o sistema de equações
elemIntForce	Calcula as forças internas de cada elemento
process	Processa o modelo atual de acordo com o método da rigidez direta

Tabela 5.1 - Propriedades e métodos da classe Drv

5.2.2 Classe Anm (Modelo de Análise)

Esta é uma superclasse abstrata que declara métodos abstratos dependentes do modelo de análise. A implementação de tais métodos abstratos é realizada nas subclasses, de acordo com o comportamento específico do modelo de análise correspondente. O comportamento dos cinco modelos de análises foi descrito no capítulo 2 (Modelos de Elementos Lineares). As subclasses derivadas dessa superclasse são: *Anm_Truss2D*, *Anm_Frame2D*, *Anm_Grillage*, *Anm_Truss3D* e *Anm_Frame3D*.

Como o valor das propriedades de um objeto de uma subclasse *Anm* é sempre o mesmo, o uso de cada subclasse é baseado nos métodos. Por essa razão, somente um objeto é criado no estágio de processamento e esse é usado para chamar os respectivos métodos.

Classe Anm	
Propriedades	Descrição
analysis_type	Tipo de modelo de análise
ndof	Número de graus de liberdade por nó
Métodos Abstratos	Descrição
glbToLocElemRotMtx	Monta a matriz de transformação por rotação
setupDOFNum	Configura a matriz com os graus de liberdade enumerados
setupPrescDispl	Armazena os deslocamentos prescritos no vetor de deslocamentos globais
elemLocStiffMtx	Monta a matriz de rigidez local de um elemento
nodalLoads	Adiciona componentes de cargas nodais ao vetor de forças globais
elemLocUniformLoadFEF	Monta o vetor de reações de engastamento de um elemento para carregamentos uniformes
elemLocLinearLoadFEF	Monta o vetor de reações de engastamento de um elemento para carregamentos lineares
elemLocThermalLoadFEF	Monta o vetor de reações de engastamento de um elemento para variação de temperatura
initIntForce	Inicializa o vetor de forças internas de um elemento com valores nulos
assembleIntForce	Monta o vetor de forças internas de um elemento

Tabela 5.2 - Propriedades e métodos da classe Anm

5.2.3 Classe Elem (Elemento)

Esta é uma superclasse abstrata que define genericamente um objeto de elemento linear, implementando os atributos e os métodos correspondentes ao comportamento dos elementos lineares de acordo com o tipo de elemento.

As propriedades desta superclasse representam o elemento através de um comportamento tridimensional nos eixos locais. Um objeto da subclasse *Anm* correspondente é responsável pela transformação deste comportamento tridimensional em um comportamento específico de acordo com o modelo.

Conforme explicado na seção 3.2 (Teorias de Vigas), elementos lineares podem ser divididos em dois tipos de acordo com a teoria considerada para o comportamento à flexão. Estes tipos compõem as duas subclasses abaixo:

- *Elem_Navier*: Esta subclasse adota o comportamento à flexão de Navier (Euler-Bernoulli) para elementos lineares. Assume-se que não há deformação por cisalhamento e, portanto, a seção transversal permanece plana e normal ao eixo longitudinal da barra.
- *Elem_Timoshenko*: Esta subclasse adota o comportamento à flexão de Timoshenko para elementos lineares. A deformação por cisalhamento é tratada de maneira aproximada e, portanto, um momento fletor faz com que a seção não seja normal ao eixo longitudinal da barra, embora esta se mantenha plana.

Cada método abstrato correspondente ao comportamento à flexão possui duas implementações, que se diferem de acordo com o plano que os elementos estão inseridos, aos distintos graus de liberdade de cada um dos modelos de análise e, também, das propriedades das seções transversais.

Vale mencionar que esta superclasse não possui propriedades ou métodos relacionados com carregamentos nos elementos. Assim, algumas de suas propriedades são ponteiros para objetos da classe *Lelem*, tal classe contém todas as informações necessárias dos carregamentos atuantes nos elementos.

Classe Elem	
Propriedades	Descrição
nen	Número de nós um elemento (sempre igual a 2)
type	Tipo de elemento
anm	Ponteiro para um objeto da classe Anm
load	Ponteiro para um objeto da classe Lelem
material	Ponteiro para um objeto da classe Material
section	Ponteiro para um objeto da classe Section
nodei	Ponteiro para objeto da classe nó correspondente ao nó inicial
nodef	Ponteiro para objeto da classe nó correspondente ao nó final
hingei	Flag para rótula no nó inicial
hingef	Flag para rótula no nó final
length	Tamanho do elemento
cosine_X	Cosseno diretor com o eixo global X
cosine_Y	Cosseno diretor com o eixo global X
cosine_Z	Cosseno diretor com o eixo global X
vz_X	Componente X do vetor local z
vz_Y	Componente Y do vetor local z
vz_Z	Componente Z do vetor local z
gle	Vetor de espalhamento
rot	Matriz de rotação do sistema global para o sistema local
tMtx	Submatriz de transformação de base por rotação do sistema local para o global
kel	Matriz de rigidez no sistema local
fel_distribLoad	Reações de engastamento no sistema local para carregamento distribuído
fel_thermalLoad	Reações de engastamento no sistema local para variações térmicas
axial_force	Vetor de forças axiais das extremidades dos elementos
shear_force_Y	Vetor de forças cortantes das extremidades do elemento relacionado ao eixo local y
shear_force_Z	Vetor de forças cortantes das extremidades do elemento relacionado ao eixo local z
bending_moment_Y	Vetor de forças cortantes das extremidades do elemento relacionado ao eixo local y
bending_moment_Z	Vetor de forças cortantes das extremidades do elemento relacionado ao eixo local z
torsion_moment	Vetor de momentos torsores das extremidades do elemento
Métodos Abstratos	Descrição
flexuralStiffCoeff_XY	Gera uma matriz de coeficientes de rigidez à flexão relacionadas ao plano XY
flexuralStiffCoeff_XZ	Gera uma matriz de coeficientes de rigidez à flexão relacionadas ao plano XZ
flexuralDisplShapeFunctionVector_XY	Gera uma matriz de funções de forma relacionadas ao plano XY
flexuralDisplShapeFunctionVector_XZ	Gera uma matriz de funções de forma relacionadas ao plano XY
Métodos Públicos	Descrição

gblStiffMtx	Gera a matriz de rigidez no sistema global
axialStiffCoeff	Gera a matriz de rigidez dos coeficientes de rigidez axiais
torsionStiffCoeff	Gera a matriz de rigidez dos coeficientes de rigidez à torção
gblStiffCoeff	Gera um vetor de forças internas provocadas pelos deslocamentos nas extremidades
axialDisplShapeFunctionVector	Gera um vetor das funções de forma axiais

Tabela 5.3 - Propriedades e métodos da classe *Elem*

5.2.4 Classe *Lelem* (Carregamento no Elemento)

A classe *Lelem* é uma superclasse abstrata que especifica as componentes de carregamento de um objeto de elemento linear, implementado métodos relacionados ao comportamento genérico dos elementos lineares e declarando métodos que dependem do tipo de elemento. O comportamento genérico dos elementos lineares e os tipos de elementos foram brevemente explicados na seção anterior. Essa classe considera carregamentos que atuam nas três dimensões.

Diferentemente da classe *Elem*, onde os métodos são funções que dependem das propriedades físicas e geométricas dos elementos, os métodos desta classe são aqueles que fazem uso apenas das informações de carregamento, como o cálculo das reações de engastamento de barras isoladas (FEF), cargas equivalentes nodais (ENL) e deslocamentos internos. O acesso às informações sobre os elementos onde estes carregamentos se encontram, é conferido por meio de propriedades cujas informações são ponteiros para a classe *Elem*.

Assim como na classe *Elem*, as subclasses do *Lelem* são responsáveis pela implementação de métodos abstratos que lidam com o comportamento à flexão. Essas subclasses são: *Lelem_Navier* e *Lelem_Timoshenko*. Cada método abstrato é implementado considerando o momento fletor no plano XY (treliça plana, pórtico plano, treliça espacial e pórtico espacial) e o momento fletor no plano local XZ (grelhas, treliça espacial e pórtico espacial), porque componentes de carregamento distintos e as propriedades das seções transversais são usadas em cada um dos casos.

Classe Lelem	
Propriedades	Descrição
elem	Ponteiro para um objeto da classe Elem
uniformDir	Flag para direção do carregamento distribuído (sistema global ou local)
uniformGbl	Vetor de carregamentos uniformemente distribuídos no sistema global
uniformLcl	Vetor de carregamentos uniformemente distribuídos no sistema local
linearDir	Flag para direção do carregamento linear (sistema global ou local)
linearGbl	Vetor de carregamentos linearmente distribuídos no sistema global
linearLcl	Vetor de carregamentos linearmente distribuídos no sistema local
tempVar_X	Variação de temperatura no centro de gravidade
tempVar_Y	Gradiente de temperatura em relação ao eixo local y
tempVar_Z	Gradiente de temperatura em relação ao eixo local z
Métodos Abstratos	Descrição
flexuralDistribLoadFEF_XY	Gera o vetor FEF de flexão para um carregamento distribuído no plano local xy
flexuralDistribLoadFEF_XZ	Gera o vetor FEF de flexão para um carregamento distribuído no plano local xz
flexuralThermalLoadFEF_XY	Gera o vetor FEF de flexão para uma variação de temperatura no plano local xy
flexuralThermalLoadFEF_XZ	Gera o vetor FEF de flexão para uma variação de temperatura no plano local xz
locDisplDistribLoad_XY	Gera o vetor de deslocamentos internos para um carregamento distribuído no plano local xy
locDisplDistribLoad_XZ	Gera o vetor de deslocamentos internos para um carregamento distribuído no plano local xz
locDisplThermalLoad_XY	Gera o vetor de deslocamentos internos para uma variação de temperatura no plano local xy
locDisplThermalLoad_XZ	Gera o vetor de deslocamentos internos para uma variação de temperatura no plano local xz
Métodos Públicos	Descrição
axialDistribLoadFEF	Gera o vetor FEF axial para um carregamento distribuído
axialThermalLoadFEF	Gera o vetor FEF axial para uma variação de temperatura
gblDistribLoadENL	Gera o vetor de carregamento nodal equivalente para um carregamento distribuído
gblThermalLoadENL	Gera o vetor de carregamento nodal equivalente para uma variação de temperatura

Tabela 5.4 - Propriedades e métodos da classe *Lelem*

5.2.5 Classe Node (Nó)

Esta classe especifica as propriedades dos pontos nodais localizados em um espaço tridimensional. Um objeto desta classe possui três coordenadas e seis graus de

liberdade, logo os vetores das condições essenciais de contorno, deslocamentos prescritos e carregamentos nodais possuem seis posições, cada uma associada com um grau de liberdade de mesmo número. Embora os valores estejam configurados para todas as posições, apenas os valores dos graus de liberdade associados ao respectivo modelo de análise são usados.

Classe Node	
Propriedades	Descrição
id	Número de identificação do nó
coord_X	Coordenada no eixo global X
coord_Y	Coordenada no eixo global Y
coord_Z	Coordenada no eixo global Z
ebc	Vetor de flags das condições de contorno essenciais
prescDispl	Vetor dos valores dos deslocamentos prescritos
nodalLoad	Vetor das componentes de deslocamentos nodais
Métodos Públicos	Descrição
axialDistribLoadFEF	Conta o número de elementos conectados a um nó

Tabela 5.5 - Propriedades e métodos da classe *Node*

5.2.6 A classe **Material**

A classe *Material* especifica as propriedades de um material homogêneo e isotrópico. A classe não possui métodos próprios.

Classe Material	
Propriedades	Descrição
id	Número de identificação do material
elasticity	Módulo de elasticidade
poisson	Coefficiente de Poisson
shear	Módulo de elasticidade ao cisalhamento
thermExp	Coefficiente de expansão térmica

Tabela 5.6 - Propriedades e métodos da classe *Material*

5.2.7 A classe Section

Esta classe define uma seção genérica tridimensional especificando todas as suas propriedades, mesmo que algumas delas não sejam usadas pelo modelo de análise correspondente. Essa classe também não tem nenhum método implementado além daqueles requeridos em outros métodos.

Classe Section	
Propriedades	Descrição
id	Número de identificação da seção
area_X	Área relacionada ao eixo local x (área cheia)
area_Y	Área relacionada ao eixo local y (área cisalhante efetiva)
area_Z	Área relacionada ao eixo local z (área cisalhante efetiva)
inertia_X	Momento de inércia em relação ao eixo local x (inércia à torção)
inertia_Y	Momento de inércia em relação ao eixo local y (inércia à flexão)
inertia_Z	Momento de inércia em relação ao eixo local z (inércia à flexão)
height_Y	Altura relacionada ao eixo local y
height_Z	Altura relacionada ao eixo local z

Tabela 5.7 - Propriedades e métodos da classe *Section*

5.2.8 A classe Draw

Esta classe é uma superclasse abstrata que define um objeto para desenhar as entidades gráficas implementando métodos estáticos para plotar figuras geométricas e declarando métodos abstratos cuja implementação depende do modelo de análise. As classes derivadas que podem implementar tais métodos abstratos são: *Draw_Truss2D*, *Draw_Frame2D*, *Draw_Grillage*, *Draw_Truss3D* e *Draw_Frame3D*.

Somente um objeto desta classe é criado para o estágio de pós-processamento e seus métodos são usados para demonstrar o modelo estrutural e os resultados de análise na tela, assim como alguns parâmetros de visualização. Por essa razão esse objeto precisa de alguma propriedade que seja um ponteiro para a classe *Drv* que contenha todas as informações acerca do modelo e seus resultados.

Os métodos estáticos foram implementados para generalizar o código plotando figuras geométricas que são usadas comumente para desenhar o modelo e os resultados. Ao usar tais métodos, ao invés de escrever o código para plotar os mesmos formatos múltiplas vezes, basta chamar a função correspondente para que ela plote a figura correspondente, simplificando, assim, a implementação de métodos abstratos.

Classe Draw	
Propriedades	Descrição
drv	Ponteiro para um objeto da classe Drv
size	Parâmetro de tamanho do desenho
az	Aximute da tela
elev	Elevação da tela
Métodos Estáticos	Descrição
coordTransform2D	Transforma as componentes de coordenadas locais x e y para as globais X e Y
coordTransform3D	Transforma as componentes de coordenadas locais x,y e z para as globais X,Y e Z
circle	Plota um círculo com posição, tamanho e cor definidos
sphere	Plota uma esfera com posição, tamanho e cor definidos
square	Plota um quadrado com posição, tamanho e cor definidos
cube	Plota um cubo com posição, tamanho e cor definidos
triangle	Plota um triangulo com posição, tamanho, orientação e cor definidos
pyramid	Plota uma pirâmide com posição, tamanho, orientação e cor definidos
arrow2D	Plota uma seta plana com posição, tamanho, orientação e cor definidos
arrow3D	Plota uma seta espacial com posição, tamanho, orientação e cor definidos
spear3D	Plota uma seta espacial com posição, tamanho, orientação e cor definidos na direção dos eixos locais
moment2D	Plota um símbolo de momento plano com posição, tamanho, orientação e cor definidos
moment3D	Plota um símbolo de momento espacial com posição, tamanho, orientação e cor definidos
Métodos Públicos	Descrição
setLimits	Estabelece os limites dos eixos
model	Desenha o modelo estrutural (somente com nós, suporte e elementos)
elements	Desenha elementos com rótulas ou contínuos
nodes	Desenha pontos nodais com condições de suporte
elemLoadsScaleFactor	Calcula o fator de escala para os carregamentos distribuídos
elemLoads	Desenha carregamentos distribuídos (uniforme e linear)
nodalLoads	Desenha cargas nodais e momentos aplicados
nodalPrescDispl	Desenha indicações de deslocamentos nodais prescritos
thermalLoads	Desenha representação de variação térmica
nodeID	Plota identificações com os números dos nós
elementID	Plota identificações com os números dos elementos
elementOrientation	Desenha as orientações dos elementos

deformScaleFactor	Calcula o fator de escala para a configuração deformada da estrutura
deformConfig	Desenha a configuração deformada da estrutura
axialScaleFactor	Calcula o fator de escala para os diagramas de força axial
axialForce	Desenha o diagrama de força axial
shearScaleFactor	Calcula o fator de escala para os diagramas de força cortante
shearForce_XY	Desenha o diagrama de força cortante resultante (plano xy local)
shearForce_XZ	Desenha o diagrama de força cortante resultante (plano xz local)
bendingMoment ScaleFactor	Calcula o fator de escala para os diagramas de momento fletor
bendingMoment	Desenha o diagrama de momento fletor resultante (plano xy local)
bendingMoment_XZ	Desenha o diagrama de momento fletor resultante (plano xz local))
torsionMoment	Desenha o diagrama de momento torsor resultante
reactions	Desenha as reações de apoio com seus respectivos valores

Tabela 5.8 - Propriedades e métodos da classe *Draw*

5.2.9 A Classe Print

Esta superclasse abstrata é similar à classe *Draw* em termos do uso e aplicabilidade, porém ao invés de definir um objeto para desenhar as entidades ela define um objeto para imprimir a informação sobre o modelo e os resultados em um arquivo de saída textual durante o estágio de pós-processamento. Essa classe implementa métodos públicos e declara métodos abstratos para imprimir informação textual que depende do modelo de análise. As subclasses derivadas desta superclasse são: *Print_Truss2D*, *Print_Frame3D*, *Print_Grillage*, *Print_Truss3D* e *Print_Frame3D*.

Um simples objeto dessa classe é criado para permitir o uso de métodos. Uma das propriedades desse objeto é um ponteiro para um objeto da classe *Drv* que contém todas as informações necessárias sobre o modelo e seus resultados.

Classe Print	
Propriedades	Descrição
Drv	Ponteiro para um objeto da classe <i>Drv</i>
Txt	Identificador para um arquivo de saída
Métodos Abstratos	Descrição
results	Imprime os resultados da análise
analysisLabel	Imprime os tipos de modelo de análise

section	Imprime informações sobre as seções transversais
nodalSupport	Imprime informações sobre as condições de suporte
nodalPrescDispl	Imprime informações sobre os deslocamentos prescritos
nodalLoads	Imprime informações sobre os carregamentos nodais
elements	Imprime informações sobre os elementos
unifElementLoads	Imprime informações sobre os carregamentos uniformes
linearElementLoads	Imprime informações sobre os carregamentos lineares
temperatureVariation	Imprime informações sobre as variações de temperatura
nodalDisplRot	Imprime os resultados dos deslocamentos e rotações nodais
reactions	Imprime os resultados das reações de apoio
intForces	Imprime os resultados das forças internas nos nós do elemento
Métodos Públicos	Descrição
header	Imprime o cabeçalho da análise dos resultados
modelDescrip	Imprime a descrição global do modelo
material	Imprime as informações sobre as propriedades do material
nodalCoords	Imprime informações sobre as coordenadas nodais

Tabela 5.9 - Propriedades e métodos da classe *Print*

5.3 Interface Gráfica do Usuário

Algumas mudanças foram feitas para adaptar a interface gráfica do programa para os modelos tridimensionais. Assim, esse item resumirá o que foi descrito no trabalho de conclusão de curso de Rafael Lopez Rangel (2016), intitulado “Development of a Graphic Program for Structural Analysis of Linear Element Models”, com as alterações feitas na interface gráfica para a consecução dos modelos tridimensionais.

A interface de usuário do LESM é composta de uma janela principal, onde é possível criar, visualizar e analisar um modelo e uma série de funções auxiliares (janelas auxiliares) responsáveis por captar os dados de entrada necessários para adicionar ou remover componentes do modelo. Cada uma dessas janelas é uma figura independente, criada usando a ferramenta GUIDE do MATLAB.

A janela principal da interface gráfica do LESM é composta por cinco painéis, uma barra de ferramentas e um botão para rodar o processo de análise (vide figura 5.2).

Descreveremos como se dá o comportamento do programa, através destas partes, nos parágrafos subsequentes.

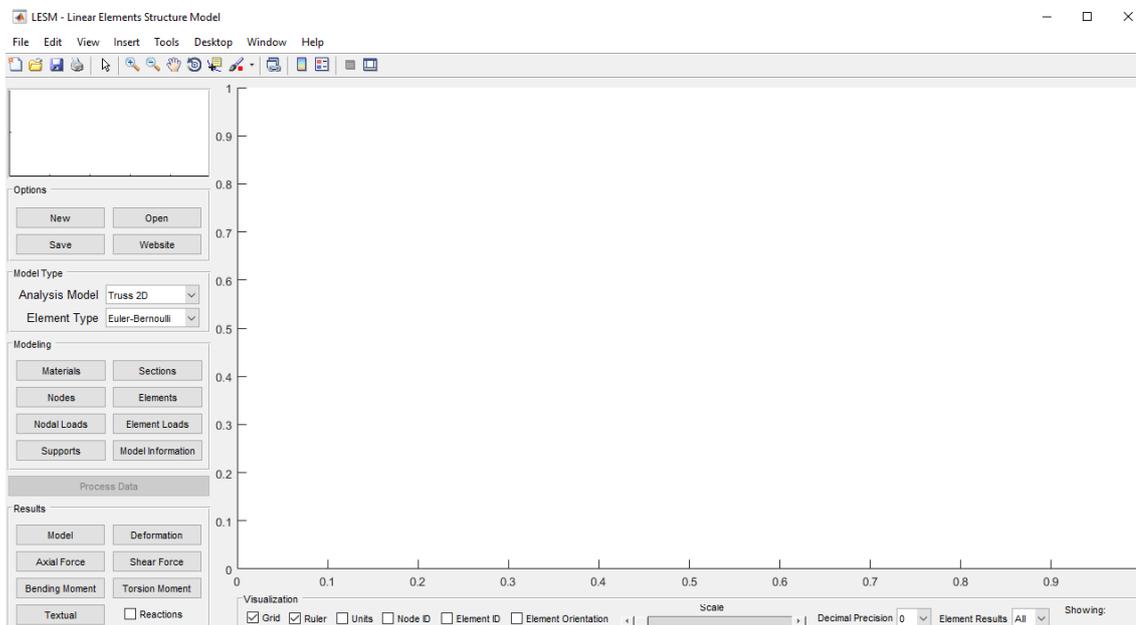


Figura 5.2 – Tela principal de interface gráfica do usuário do LESM 3D

A tela é um componente da interface que define a área, situada em um sistema de eixos com três dimensões, onde o programa desenhará certas figuras geométricas para representar o modelo estrutural e os resultados obtidos no pós-processamento.

Há três barras de ferramentas pré-definidas que podem ser adicionadas a janela principal do programa:

- Figure Toolbar: Adiciona opções básicas de visualização (disponível no início do programa).
- Camera Toolbar: Adiciona opções avançadas de visualização.
- Plot Edit Toolbar: Adiciona ferramentas que possibilitam criar textos, linhas formatos a janela.

O painel “Options” permite ao usuário criar um novo modelo, abrir um modelo pré-existente, salvar o modelo atual e abrir o website do programa que contém o manual do programa.

O painel “Model Type” é usado para escolher o tipo de modelo de análise (treliça plana, pórtico plano, grelha, treliça espacial e pórtico espacial) e, também, o tipo de elemento (Navier ou Timoshenko) a serem utilizados.

O tipo de análise e o tipo de elemento devem ser definidos logo no início da criação de um modelo estrutural e não podem ser alterados durante a elaboração do modelo ou na fase de pós-processamento. Vale mencionar que cada modelo de análise possui configurações de visualização próprias.

O painel “Modelling” é usado para criar, alterar e deletar componentes (materiais, seções, nós, elementos, cargas nodais, carregamentos distribuídos e restrições) do modelo estrutural, sendo acessível permanentemente ao usuário. O clique em um botão deste painel abre uma GUI auxiliar para gerenciar a criação de objetos do componente selecionado permitindo ao usuário preencher apenas os campos de dados usados pelo modelo selecionado (as unidades dos dados de entrada são imutáveis). Neste painel foi criado apenas uma caixa de dialogo modal presente para os modelos tridimensionais, que permite ao usuário criar um vetor local v_z diferente do vetor pré-definido pelo programa.

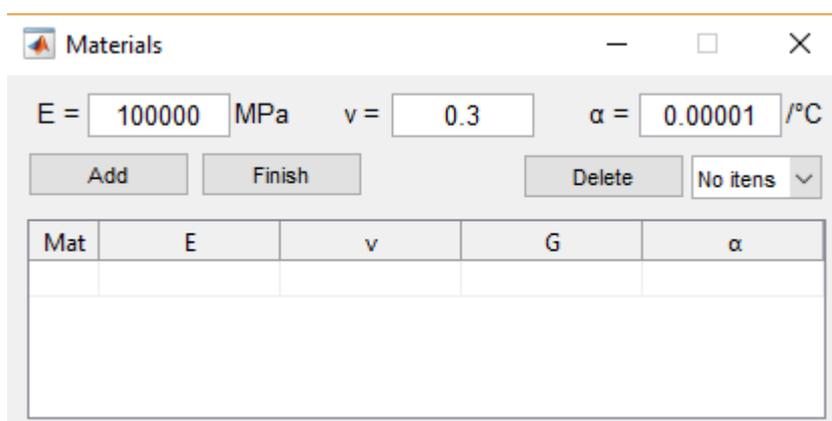


Figura 5.3 – GUI auxiliar para gerenciamento dos materiais

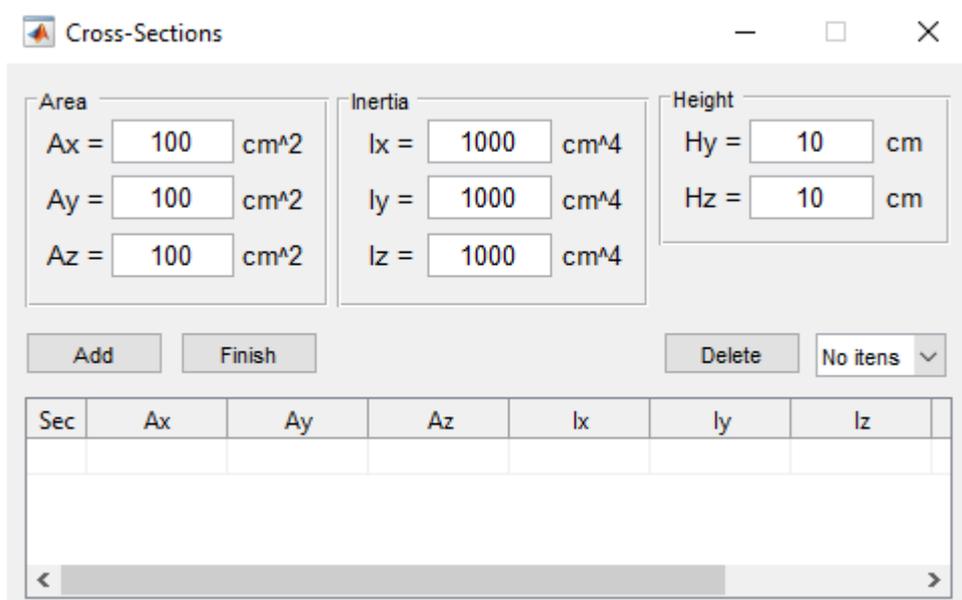


Figura 5.4 – GUI auxiliar para gerenciamento das propriedades das seções transversais

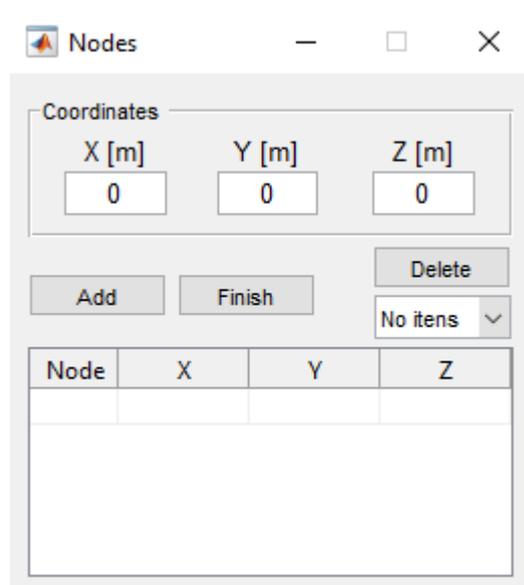


Figura 5.5 – GUI auxiliar para gerenciamento dos pontos nodais

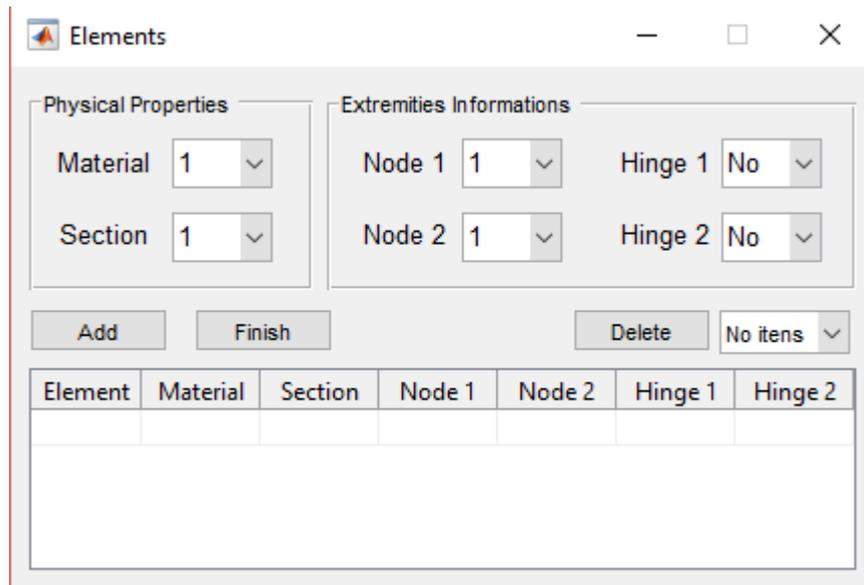


Figura 5.6 – GUI auxiliar para gerenciamento dos elementos de barra

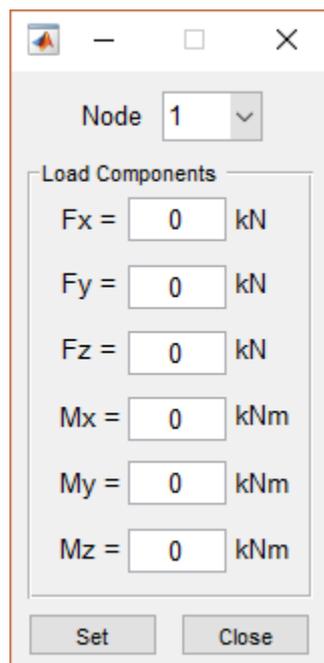


Figura 5.7 – GUI auxiliar para gerenciamento das cargas nodais

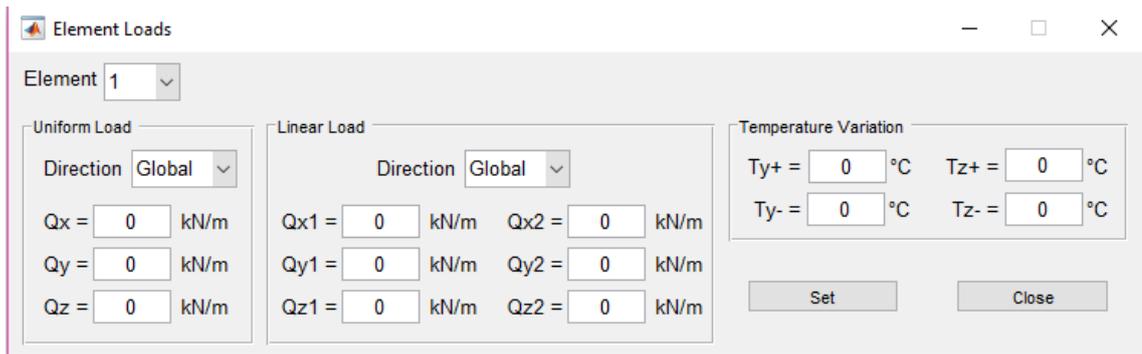


Figura 5.8 – GUI auxiliar para gerenciamento dos carregamentos nos elementos

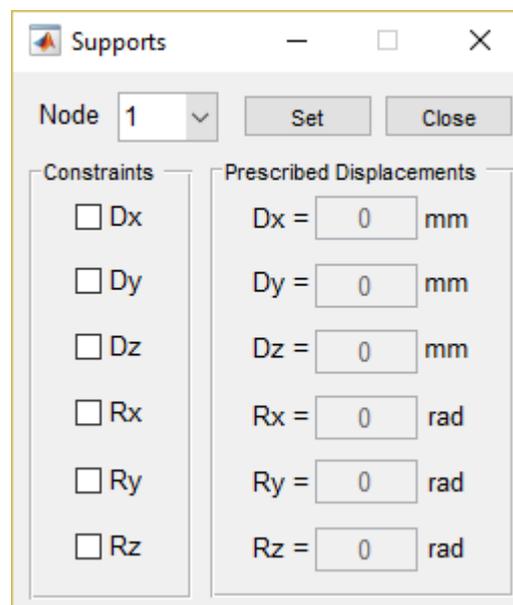


Figura 5.9 – GUI auxiliar para gerenciamento das restrições de apoio e deslocamentos prescritos

O botão para o processamento dos dados é responsável por rodar o código que calcula os resultados de análise através do método da rigidez direta. Esses resultados são então colocados como propriedades dos objetos usados no estágio de pós-processamento para desenhar resultados gráficos ou criar arquivos de texto com os resultados numéricos. Este botão é habilitado somente quando é identificado que um modelo está sendo construído (criação de um ponto nodal). Quando este botão é clicado e os dados são processados com sucesso, os botões no painel “Results” são habilitados e este botão é desabilitado até que sejam feitas novas mudanças no modelo. Toda vez que um modelo é carregado ou modificado, seus dados devem ser processados novamente antes de checar os resultados.

Resultados gráficos e textuais podem ser checados através do clique de botões no painel “Results”. As funções auxiliares responsáveis pelos diagramas de esforços internos e reações chamadas pelos botões e caixa de seleção, respectivamente, serão discutidas na seção 5.6.

5.4 Pré-Processamento de Dados

Este estágio consiste no fornecimento de dados de entrada com todas as informações necessárias correspondentes ao modelo da estrutura real desejada. Essas informações podem ser fornecidas através de um processo de modelamento ou através do carregamento de um arquivo já existente que contém a descrição do modelo em um formato que o LESM 3D possa ler e interpretar.

Ao carregar um arquivo executa-se uma função auxiliar chamada `readFile`, que lerá as informações do modelo e criará os objetos que armazenarão as informações nas propriedades correspondentes. O modelo é plotado na tela, mas não há encaminhamento algum para a etapa de processamento de dados. Esta função pode ler os tipos de arquivos abaixo:

- Arquivos `lsm`: Arquivo padrão do programa. Quando um modelo é salvo, seus dados são gravados em um arquivo deste tipo.
- Arquivos `Neutros`: Arquivo cujo formato fora desenvolvido com o objeto de conter todas as informações necessárias para programas de análises de elementos finitos. Esses formatos são bem similares com o arquivo `lsm`. Mais informações sobre o arquivos neutros podem ser encontradas no endereço: <http://www.tecgraf.puc-rio.br/neutralfile>.

Vale mencionar que estes dois tipos de arquivos guardam dados correspondentes a um modelo tridimensional, sendo necessário que a função `readFile` adquira informações sobre o modelo de análise para que a subclasse `Anm` correspondente adequa as informações ao modelo desejado.

O processo de modelamento permite ao usuário construir um modelo novo ou modificar um modelo pré-existente. Deve-se inserir através das GUIs auxiliares os dados de entrada correspondentes às informações sobre os materiais, seções transversais, nós, elementos, condições de suporte e carregamentos.

Para evitar erros durante o processo de modelamento optou-se por controlar a experiência do usuário através de duas medidas:

- Desabilitando determinados campos: Há o bloqueio de campos correspondentes aos dados de entrada que não são usados no modelo de análise selecionado.
- Desenvolvimento de um método chamado fictRotConstraint: Este método insere ou remove uma restrição de rotação fictícia em nós com mais de um elemento rotulado conectado nesta extremidade para, assim, impedir que o modelo torne-se instável.

A figura 5.X indica as propriedades que poderão ser definidas nessa etapa para a análise do modelo, preparando para a etapa seguinte:

Propriedades	Treliça 2D	Pórtico 2D	Grelha	Treliça 3D	Pórtico 3D
Área da Seção Transversal	Ax	Ax Ay	Ax Az	Ax	Ax Ay Az
Inercia da Seção Transversal	-	Iz	Ix Iy	-	Ix Iy Iz
Altura da Seção Transversal	Hy	Hy	Hx	Hy Hx	Hy Hx
Restrições	Dx Dy	Dx Dy Rz	Dz Rx Ry	Dx Dy Dz	Dx Dy Dz Rx Ry Rz
Deslocamentos Prescritos	Dx Dy	Dx Dy Rz	Dz Rx Ry	Dx Dy Dz	Dx Dy Dz Rx Ry Rz
Cargas Nodais	Fx Fy	Fx Fy Mz	Fz Mx My	Fx Fy Fz	Fx Fy Fz Mx My Mz
Carregamento Uniforme	Qx Qy	Qx Qy	Qx Qy	Qx Qy Qz	Qx Qy Qz
Carregamento Linear	Qx1 Qy1 Qx2 Qy2	Qx1 Qy1 Qx2 Qy2	Qz1 Qz2	Qx1 Qy1 Qz1 Qx2 Qy2 Qz2	Qx1 Qy1 Qz1 Qx2 Qy2 Qz2
Variação de Temperatura	Ty+ Ty-	Ty+ Ty-	Tz+ Tz-	Ty+ Tz+ Ty- Tz-	Ty+ Tz+ Ty- Tz-

Tabela 5.10 – Propriedades de cada modelo de análise

5.5 Processamento de dados

Esta etapa é a responsável pelo cálculo dos resultados da análise estrutural do modelo desejado. O botão “Process Data” é habilitado assim que um modelo estrutural é criado e um clique neste botão criará um objeto da classe Drv, classe responsável por “comandar” o processo de análise do programa.

Este estágio engloba um grupo de funções implementadas cujos dados de entrada foram fornecidos no estágio de pré-processamento. Os resultados obtidos na análise são guardados nas propriedades desses objetos que precisam ser acessados pelos objetos das classes de pós-processamento.

O algoritmo usado para calcular os resultados da análise é regido pelo método da rigidez direta. Como mencionando anteriormente, este é um método matricial que usa as relações de rigidez dos elementos para calcular os deslocamentos desconhecidos dos nós, reações de apoio e os esforços internos nas extremidades dos elementos.

O problema gira em torno da resolução de um sistema de equações de equilíbrio correspondente ao modelo estrutural analisado. Esse sistema tem a forma dada pela equação 5.1, onde $\{F\}$ é o vetor global de forças, $[K]$ é a matriz de rigidez global e $\{D\}$ é o vetor global de deslocamentos nodais. As variáveis desse sistema são os valores de deslocamentos nodais na direção dos graus de liberdade livres e os valores das reações de apoio correspondentes às forças nas direções dos graus de liberdade fixos. Vale mencionar que o número de equações no sistema é equivalente ao número total de graus de liberdade do modelo.

$$\{F\} = [K]\{D\} \quad (5.1)$$

Após a resolução do sistema dado pela equação 5.1, é necessário que o programa forneça os resultados dos esforços internos nas direções das coordenadas generalizadas no sistema local de cada barra, situada nas extremidades dessas barras.

Assim, os esforços internos nas extremidades de barras descarregadas são obtidos através da resolução, consecutiva, de dois sistemas, dados pelas equações 5.2 e

5.3, onde $\{f\}$ é o vetor de forças no sistema global, $[k]$ é a matriz de rigidez no sistema global, $\{d\}$ é o vetor de deslocamentos nodais no sistema global, $\{f^*\}$ é o vetor de forças no sistema local e $[R]$ é matriz de transformação por rotação do sistema global para o local.

Adicionalmente para elementos de barra carregados deve-se sobrepor os resultados encontrados na equação 5.3 (solução global), aos resultados das reações de engastamento perfeito no sistema local da barra (solução local).

$$\{f\} = [k]\{d\} \quad (5.2)$$

$$\{f^*\} = [R]\{f\} \quad (5.3)$$

A explicação desse método será dividida em três partes a serem explicadas nas seções subsequentes: Montagem do Sistema de Equações, Resolução do Sistema de Equações e Cálculo das forças Internas nas Extremidades.

5.5.1 Montagem do Sistema de Equações

Na primeira parte do método os termos do sistema de equações de equilíbrio da equação 5.1 devem ser montados de maneira que sua resolução possa ser particionada. Sem esse particionamento a matriz composta pelos coeficientes de rigidez do problema seria singular e, portanto, o sistema não admitiria solução. O diagrama de sequência apresentado na figura 5.20 indica a ordem dos métodos chamados para montar a matriz de rigidez global, o vetor global de deslocamentos e vetor global de forças. As caixas desse diagrama possuem o nome e o método das classes, não especificando as subclasses.

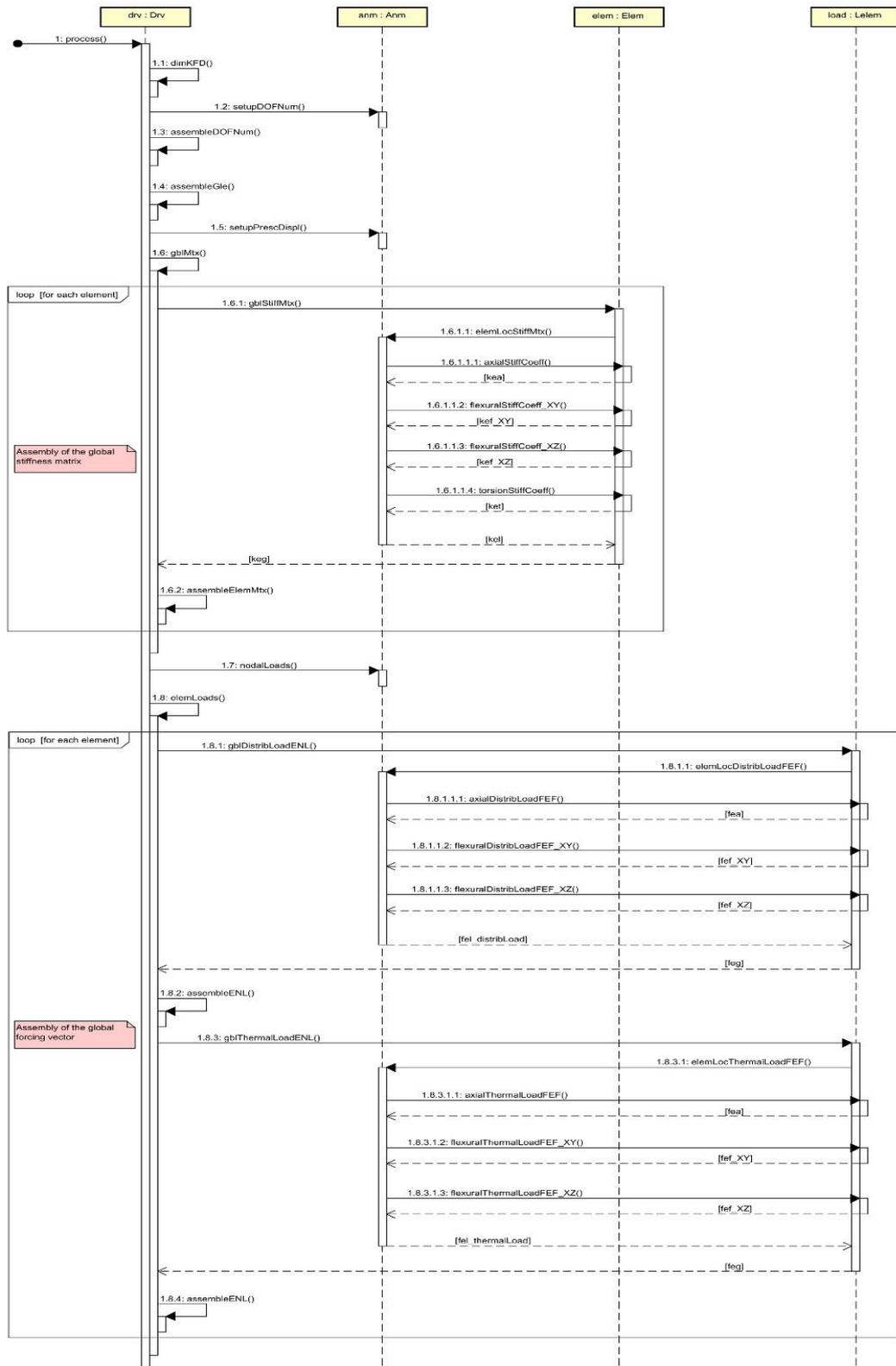


Figura 5.10 – Diagrama de sequência simplificado

5.5.1.1 Inicialização das Variáveis Auxiliares

O primeiro passo do processo de análise consiste na inicialização e dimensionamento da matriz de rigidez global, do vetor global de deslocamentos e do vetor global de forças. Este procedimento é realizado através do método “dimKfD”. Este método cria uma matriz quadrada e dois vetores colunas com tamanhos correspondentes ao número de equações do sistema, todos preenchidos com zeros.

Para o correto particionamento do sistema, as equações relacionadas aos graus de liberdade livres devem ser enumeradas primeiro antes das equações dos graus de liberdade fixos. Este procedimento é realizado pelo método “setupDOFNum”. Posteriormente devem ser criadas duas novas variáveis auxiliares para guardar o número de graus de liberdade de cada nó e elemento (matriz ID e o vetor de espalhamento). A matriz ID tem o número das colunas igual ao número de nós do modelo e o número de linhas igual ao número de graus de liberdade por nó. O termo $ID_{(ij)}$ fornece o número de graus de liberdade “i” do “j”. Esta matriz é montada pelo método “assembleDOFNum”.

O vetor de espalhamento é um vetor coluna com o tamanho correspondente ao número de graus de liberdade de um elemento, ou seja, é igual a soma dos graus de liberdade dos nós inicial e final. Esse vetor é posto como uma propriedade dos objetos da classe *Elem* e fornece o número de equações de graus de liberdade de cada. Este vetor de espalhamento de elementos é montado no método “assembleGle”.

5.5.1.2 Montagem do Vetor de Deslocamentos Globais

Este vetor armazena os valores dos deslocamentos nodais no sistema global. Sua ordem de montagem segue a ordem de enumeração das equações, fazendo com que os deslocamentos dos graus de liberdade fixos fiquem situados no fim do vetor.

A primeira parte deste vetor corresponde às variáveis desconhecidas do sistema de equação, enquanto a segunda parte guarda os valores conhecidos dos deslocamentos. Os deslocamentos na direção dos graus de liberdade são nulos caso não

haja deslocamentos prescritos. O método “setupPrescDispl” guarda os valores dos deslocamentos prescritos no vetor de deslocamentos globais.

5.5.1.3 Montagem da Matriz de Rigidez Global

Esta matriz é a responsável por correlacionar o vetor de deslocamentos globais e o vetor de forças globais. É montada através de uma soma direta das matrizes de rigidez de cada elemento, implementada pelo método “gblMtx”.

Na seção 3.4 (Matrizes de Rigidez), foi apresentado as matrizes de rigidez dos elementos lineares no sistema local usando os termos de coeficientes de rigidez, dados no apêndice A (Coeficientes de Rigidez do Elemento Local). Os coeficientes de rigidez dessas matrizes são gerados nos métodos axialStiffCoeff, flexuralStiffCoeff_XY, flexuralStiffCoeff_XZ e torsionStiffCoeff. A montagem da matriz de rigidez no sistema local de cada elemento é feita pelo método “elemLocStiffMtx”, através da alocação dos termos de acordo com o tipo de modelo da matriz de coeficientes de rigidez nas posições certas da matriz de rigidez desejada.

A matriz de rigidez no sistema local deve ser transformada na matriz de rigidez global, conforme preconiza a equação 5.1.

Para isso devemos primeiramente rotacionar a matriz de rigidez no sistema local para o sistema global através do método “gblStiffMtx”, usando a matriz de transformação por rotação do elemento, como mostrado na equação 3.4.

Posteriormente, devemos aplicar o método “gblMtx” às matrizes de rigidez no sistema global de cada elemento para obtermos a matriz de rigidez global. Vale mencionar que o método “gblMtx” usa um outro método chamado de “assembleElemMtx”, para inserir os termos das matrizes de rigidez no sistema global de cada elemento na posição correta da matriz de rigidez global.

A adição das contribuições das matrizes de rigidez de cada elemento, pelo método “assembleElemMtx”, só é possível graças ao vetor de espalhamento, responsável por associar cada coordenada generalizada local com a coordenada global correspondente.

5.5.1.4 Montagem do Vetor de Forças Globais

Este vetor armazena os valores das forças que atuam nos nós na direção dos graus de liberdade no sistema global. A ordem de montagem desse vetor segue a ordem enumerada das equações. Assim, as forças desconhecidas associadas aos graus de liberdade fixos (reações de apoio) se localizam no fim do vetor, enquanto os valores de forças conhecidas se localizam no início.

O valor dessas forças pode ser obtido através da combinação das contribuições vindas das cargas nodais aplicadas e das cargas equivalentes nodais de elementos carregados. Inicialmente, assume-se que as forças desconhecidas deste vetor armazenam cargas aplicadas diretamente aos graus de liberdade fixos.

O método “nodalLoads” é responsável por adicionar a contribuição das cargas nodais aplicadas a qualquer termo do vetor de forças globais. Lembrando que a matriz ID é usada para associar os graus de liberdade dos nós à equação de número correspondente no vetor de forças globais.

O método “elemLoads” é responsável por adicionar a contribuição das cargas nodais equivalentes a cada elemento carregado, incluindo carregamentos distribuídos e cargas termiais, ao vetor de forças globais. As cargas nodais equivalentes correspondem às reações de engastamento perfeito em elementos carregados na direção oposta.

Os vetores de reações de engastamento perfeito de elementos lineares são dados no sistema local, usando termos dos vetores das reações de engastamento perfeito desacoplados para cada efeito individual. Esses vetores desacoplados são dados nos apêndice A (reações de engastamento perfeito para elementos com carregamento linearmente distribuídos) e B (Reações de Engastamento Perfeito para Variação de Temperatura) e gerados nos métodos axialDistribLoadFEF, axialThermalLoadFef, flexuralDistribLoadFEF_XY, flexuralDistribLoadFEF_XZ, flexuralThermalLoad_XY e flexuralThermalLoad_XZ. A montagem do vetor de reações de engastamento perfeito para cada elemento é feito nos métodos elemLocUniformLoadFEF, elemLocLinearLoadFEF e elemLocThermalLoadFEF através da alocação de termos dos vetores desacoplados nas posições corretas do vetor de reações de engastamento perfeito.

Para calcular o vetor de cargas equivalentes nodais de um elemento, deve-se rotacionar o vetor de reações de engastamento perfeito para o sistema global e multiplica-lo por -1 para inverter as direções das forças. Este procedimento é realizado nos métodos `gbldistribLoadENL` e `gbIThermalLoadENL`. O método “`elemLoads`” obtém esses vetores rotacionados e chama o método “`assembleENL`” para inseri-los nas posições corretas do vetor de forças globais. Os vetores de espalhamento são usados para associar os graus de liberdade dos elementos a equação de número correspondente no vetor de forças globais.

5.5.2 Resolução do Sistema de Equações

Após a montagem do sistema de equações de equilíbrio da equação 5.1, este deve ser resolvido para a obtenção das incógnitas de nosso problema (deslocamentos nodais desconhecidos e reações de apoio). Esta resolução só é possível com o devido particionamento dos componentes da equação 5.1. Lembrando que a ordem de enumeração dos graus de liberdade é que faz ser possível a divisão do sistema de equações nas equações 5.4 e 5.5.

O particionamento da matriz de rigidez global, do vetor de deslocamentos globais, e do vetor de forças globais é descrito abaixo. O subscrito “l” se refere a livre e o “f” se refere a restringido.

- $\{D_l\}$: vetor de deslocamentos nodais nas direções dos graus de liberdade livres (desconhecidos)
- $\{F_l\}$: vetor de cargas nodais nas direções dos graus de liberdade livres (conhecidos)
- $\{D_f\}$: vetor de deslocamentos nodais nas direções dos graus de liberdade fixos (conhecidos)
- $\{F_f\}$: Vetor de cargas nodais nas direções dos graus de liberdade fixos (desconhecidos)

$$[K_{ll}]\{D_l\} + [K_{lf}]\{D_f\} = \{F_l\} \quad (5.4)$$

$$[K_{nl}] \{D_l\} + [K_{ff}] \{D_f\} = \{F_f\} \quad (5.5)$$

A manipulação da equação 5.4 resulta na equação 5.6 abaixo:

$$[K_{nl}] \{D_l\} = \{F_l\} - [K_{ff}] \{D_f\} \quad (5.6)$$

É fácil perceber que o lado direito da equação 5.6 é composto apenas de valores conhecidos, assim, a solução desse sistema nos fornecerá as deslocabilidades globais livres do problema $\{D_l\}$. A resolução da equação 5.6 permite que encontremos as reações de apoio a partir da equação 5.5 da seguinte maneira:

$$[F_f] = [K_{nl}] \{D_l\} + [K_{ff}] \{D_f\} \quad (5.7)$$

O particionamento do sistema de equações e a resolução deste é feita no método “solveEqnSystem” da classe Drv. Este método é responsável por auferir a estabilidade do sistema, conferindo se a submatriz K_{ll} é singular ou não, antes do início da resolução do sistema. O modelo é instável caso esta matriz seja singular e o processamento de dados é interrompido com a exibição de uma mensagem de erro.

5.5.3 Cálculo dos Esforços Internos nas Extremidades

Os esforços internos de um elemento são inicialmente calculados em suas extremidades, levando em consideração os efeitos da análise global e local do modelo estrutural. É necessário calcular os esforços internos ao longo do comprimento dos elementos, de acordo com os valores das extremidades para a geração dos resultados gráficos. Lembrando que a geração dos diagramas de esforços internos é realizada no estágio de pós-processamento.

A análise global fornece apenas os esforços internos resultantes dos elementos dos efeitos de deslocamentos nodais. O valor desses deslocamentos é obtido quando o sistema de equação de equilíbrio representado pelas equações 5.6 e 5.7 é resolvido.

A análise local dos elementos fornece os efeitos dos esforços internos ao longo da barra e devem ser sobrepostos aos efeitos de deslocamentos nodais.

Todo o processo de cálculo de esforços internos é realizado no método “elemIntForce” como indicado na figura abaixo. Esta figura corresponde a uma parte do diagrama de sequência do programa. Esse diagrama não especifica o plano onde a flexão ocorre para métodos relacionados ao comportamento de flexão.

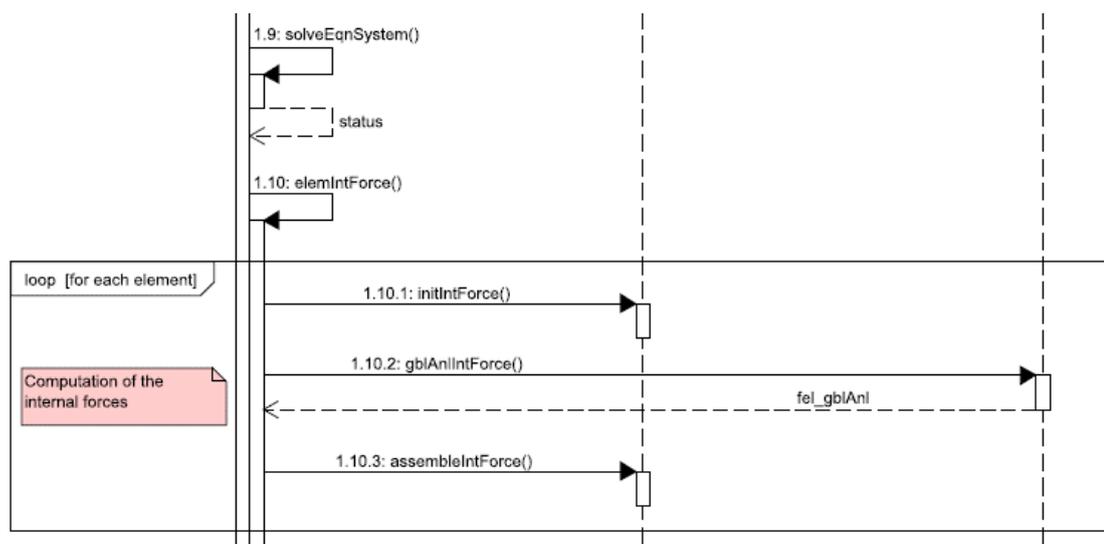


Figura 5.11 – Diagrama de sequência de cálculo das forças internas

Inicialmente, os vetores de esforços internos de cada elemento são inicializados com valores nulos no método “initIntForce”, devendo-se adicionar a contribuição da análise global a esses vetores vazios. O método “gblAnlIntForce” utiliza o vetor de espalhamento do elemento para obter o vetor de deslocamentos globais correspondente $\{d\}$, converte-o para o sistema local usando a matriz de transformação por rotação $\{R\}$ e multiplica o vetor obtido $\{d'\}$ pela matriz de rigidez local do elemento, $\{k'\}$, que deve ser montada mais de uma vez (vide equações 5.7 e 5.8). O resultado desse produto é o vetor de forças generalizadas no sistema local, correspondente às forças aplicadas nas extremidades do elemento para manter o elemento na configuração deformada. Esse vetor é então adicionado aos vetores de forças internas no método “assembleIntForce”.

$$\{d'\} = [R]\{d\} \quad (5.7)$$

$$\{f'\} = [k'] \{d'\} \quad (5.8)$$

O método “elemIntForce” é o responsável por adicionar as contribuições da análise local ao vetor de forças internas. Este método adiciona as contribuições de cada tipo de carregamento no elemento através da obtenção de um vetor composto pelas reações de engastamento perfeito, ao vetor de esforços internos, através do método “assembleIntForce”. Os vetores das reações de engastamento perfeito devem ser montados mais uma vez, como fora realizado na montagem do vetor de forças globais (5.5.1.4).

A convenção de sinais dos esforços internos calculados se difere da convenção apresentada no capítulo 2 (Modelos de Estruturas Reticuladas). Esta convenção adota que os esforços internos são positivos quando atuam nas direções positivas do grau de liberdade correspondente. Esta convenção de sinais é mantida quando se imprime os resultados textuais, porém é necessário a devida conversão para a convenção usual (apresentada no capítulo 2) no estágio de pós-processamento para a exibição dos resultados gráficos.

Visto que um dos principais objetivos do programa é ser utilizado no ensino do método da rigidez direta, optou-se por um código mais limpo e de fácil entendimento em razão da eficiência. Assim, propositadamente, a matriz de rigidez de todos os elementos e o vetor de reações de engastamento perfeito nas extremidades de todos os elementos carregados são calculados e montados duas vezes ao longo do processamento de dados.

5.6 Pós-Processamento de Dados

Uma vez obtidos os resultados da etapa de processamento esses tornam-se avaliáveis para os usuários graficamente ou por meio de impressão na etapa de pós-processamento. Esta seção é dedicada a explicação das considerações e o desenvolvimento do código para expor tais resultados de análise presentes no painel “Results” da interface gráfica do programa.

As componentes de gráfico deste painel são habilitadas somente quando os dados do modelo, computados na fase de processamento, são processados adequadamente. Habilitando, assim, os botões que fornecem os diagramas de esforços internos existentes no modelo de análise correspondente. Quando um botão ou o checkbox presente neste painel é clicado, um método da classe de pós-processamento é chamada através de uma função de callback.

A interpretação dos diagramas de reações internas depende da convenção adotada para determinar a parte inferior e a superior de um elemento. A convenção adotada pelo LESM adota as particularidades abaixo:

- A parte inferior de colunas (coordenadas do elemento $x_i = x_f$) se encontra na porção à direita dos elementos de barra, tanto para os planos XY como para os planos XZ.
- A parte inferior de vigas (coordenadas do elemento $x_i \neq x_f$) se encontra na porção inferior dos elementos de barra, tanto para os planos XY como para os planos XZ.

A convenção acima é que determinará o módulo dos valores nos diagramas de esforços internos no LESM e será adotada por todos os modelos de análise presentes no programa.

Vale mencionar que cada função de plotagem de diagrama e, também, da configuração deformada apresentam uma função de fator de escala correspondente. Tal função calcula um fator de escala que multiplicará as coordenadas utilizadas no desenho do diagrama para, assim, desenhar um diagrama adequado às características do elemento. Há ainda um slider, situado no painel “Visualization”, que também poderá alterar a escala do diagrama.

As únicas adaptações necessárias para readequar a interface para os modelos de análise tridimensionais foram a implementação de janelas adicionais presentes nos modelos de pórticos tridimensionais. Tais janelas são abertas ao clicar nos botões “Shear Force” e “Bending Moment” e apresentam as caixas de seleção “All”, “Qy” (ou “My”) e “Qz” (ou “Mz”), correspondentes aos resultados adicionais que podem ser obtidos nesse tipo de modelo.

5.6.1 Configuração Deformada

A representação gráfica da configuração deformada de um modelo estrutural exige a discretização do elemento de barra em um determinado número de nós e a interpolação dos deslocamentos e rotações desses nós através das funções de forma e do conhecimento dos graus de liberdade do modelo para, assim, obter os deslocamentos e rotações em qualquer ponto ao longo do eixo da barra.

As funções de forma podem ser entendidas como funções de interpolação de deslocamentos e rotações, definindo as elásticas elementares de uma barra isolada, podendo ser usadas para calcular os deslocamentos de cada nó criado dentro do elemento como resultado dos deslocamentos das extremidades.

Isso é feito através da montagem da matriz de funções de forma com os respectivos valores das expressões das funções de forma, dados no apêndice D (funções de forma). Assim, as equações apresentadas no item 3.6, podem ser usadas para obtermos um vetor com componentes de deslocamentos axiais e transversais de um nó interno.

As expressões de deslocamentos internos em elementos carregados são usadas para calcular os deslocamentos de cada nó dentro de um elemento como resultado na análise local. As expressões que descrevem as componentes de deslocamentos axiais e transversais são dadas no apêndice E (deslocamentos internos para carregamentos linearmente distribuídos) e apêndice F (deslocamentos internos para variação de temperatura). Para determinar os deslocamentos totais de um nó interno as componentes de deslocamento resultam dos deslocamentos nodais (análise global) e deslocamentos internos (análise local) são combinadas.

Adiciona-se as componentes de deslocamentos totais de cada nó interno às coordenadas originais para representar graficamente a configuração deformada do elemento ao conectar os pontos obtidos.

A composição das matrizes de funções de forma de treliças espaciais e pórticos espaciais, implementadas nesta atualização, diferem-se dos seus correspondentes planos, já que tais modelos possuem elementos um número maior de deslocabilidades. Assim, a matriz de função de forma da treliça espacial tornou-se uma matriz 2×6 (a rotação foi considerada desnecessária na implementação) e a matriz de

pórtico espacial tornou-se uma matriz 2x12 (a rotação foi considerada desnecessária na implementação). Vale mencionar que a matriz representativa dos pórticos espaciais fora montada como uma superposição dos modelos de pórtico plano e grelha.

5.6.2 Diagrama de Forças Normais

A função que fornece o valor da força axial em qualquer posição de elementos carregados, $N(x)$, pode ser obtida a partir da equação diferencial 5.9.

$$\frac{dN(x)}{dx} = -p(x) \quad (5.9)$$

A função que descreve a distribuição de carregamentos axiais no elemento, $p(x)$, é computada combinando carregamentos uniformes e lineares. Assumindo um carregamento genérico distribuído para essa combinação, na forma da equação 5.10, as forças axiais internas são expressas na equação 5.10.

$$p(x) = Ax + B \quad (5.10)$$

$$N(x) = -\frac{Ax^2}{2} - Bx + C \quad (5.11)$$

A constante C é calculada usando as condições de contorno para as forças axiais nas extremidades do elemento. O diagrama de forças axiais é disposto com os valores positivos na parte de cima do elemento.

5.6.3 Diagrama de Força Cortante

As funções que fornecem os valores das forças cortantes em quaisquer posições de elementos carregados, $V_y(x)$ e $V_z(x)$, podem ser obtidas das equações diferenciais 5.12 e 5.13.

$$\frac{dV_y(x)}{dx} = -q_y(x) \quad (5.12)$$

$$\frac{dV_z(x)}{dx} = -q_z(x) \quad (5.13)$$

As funções que descrevem os carregamentos transversais distribuídos de um elemento, $q_y(x)$ e $q_z(x)$, e as força cortantes são:

$$q_y = Ax + B \quad (5.14)$$

$$q_z = Dx + E \quad (5.15)$$

$$V_y(x) = -\frac{Ax^2}{2} - Bx + C \quad (5.16)$$

$$V_z(x) = -\frac{Dx^2}{2} - Ex + F \quad (5.17)$$

O diagrama de cortante é desenhado de modo semelhante ao diagrama de força axial para elementos descarregados, já em elementos carregados transversalmente a força cortante é calculada para cada nó interno do elemento.

As constantes C e E são calculadas usando as condições de contorno de força cortante nas extremidades do elemento. O diagrama de força cortante é representado com os valores positivos plotados na parte superior do elemento de barra.

Foi necessário implementar trechos no código correspondentes às equações 5.13, 5.15 e 5.17 devido à adição do modelo de pórtico tridimensional. Tais equações não são utilizadas em modelos de pórticos planos e grelhas.

5.6.4 Diagrama de Momento Fletor

As funções que fornecem os valores dos momentos fletores em quaisquer posições de elementos carregados, $M_z(x)$ e $M_y(x)$, podem ser obtidas das equações diferenciais 5.19 e 5.21, deduzidas a partir das equações 5.18 e 5.20.

$$\frac{d^2M_z(x)}{dx^2} = -q_z(x) \quad (5.18)$$

$$M_y(x) = -\frac{Ax^3}{6} - \frac{Bx^2}{2} + Cx + D \quad (5.19)$$

$$\frac{d^2M_y(x)}{dx^2} = -q_y(x) \quad (5.20)$$

$$M_z(x) = -\frac{Ex^3}{6} - \frac{Fx^2}{2} + Gx + H \quad (5.21)$$

As constante C e E são calculadas conforme explicado na seção 5.6.3 e a constante D usando as condições de contorno de momento fletor nas extremidades nas equações 5.14 e 5.15. Os diagramas são desenhados com os valores positivos situados no lado tensionado do elemento.

O diagrama de momento fletor nos pórticos planos, grelhas e pórticos espaciais é desenhado através da discretização dos elementos carregados em vários pontos nodais internos.

Foi necessário implementar trechos no código correspondentes às equações 5.20 e 5.21, devido à adição do modelo de pórtico tridimensional. Tais equações não são utilizadas em modelos de pórticos planos e grelhas.

5.6.5 Diagrama de Momento Torsor

Presente em modelos de grelha e de pórtico espacial, o valor do momento torsor será constante devido a configuração de carregamentos suportada pelo LESM. Tais valores são calculados para as extremidades do modelo e possuem sinais definidos pelas convenções de sinais presentes no capítulo 2. Assim, optou-se por apenas indicar os valores de momento torsor, sendo estes imprimidos próximos ao ponto médio de cada elemento.

6. Conclusão

O resultado final obtido com a extensão do LESM para modelos de análises tridimensionais (LESM 3D) foi alcançado de forma satisfatória. As implementações feitas no programa original tornaram o seu código ainda mais limpo e fácil de entender. Assim, o principal objetivo pretendido na consecução deste trabalho, estender um programa de análises de estruturas de cunho educacional para o recebimento de modelos de análise tridimensionais, foi atendido.

Durante a consecução do trabalho comprovou-se que implementar o método da rigidez direta computacionalmente torna o seu entendimento muito mais simples, ainda que em modelos tridimensionais, o que reforça o viés educacional buscado originariamente pelos criadores do LESM.

Além disso, o uso eficaz da programação orientada a objetos permitiu o cumprimento do segundo objetivo, um código que permita facilmente futuras inserções.

As mudanças implementadas exigiram a criação de diversas subclasses e a correspondente alteração de classes já existentes para se adequar às novas necessidades do programa. Tais mudanças não exigiram alterações na alma do programa, a classe `Drv`, responsável por comandar o estágio de processamento de dados. Isso demonstra que a arquitetura do LESM fora feita de maneira que, realmente, possibilite implementações adicionais em seu código, o que comprova de forma plena o alcance de seu objetivo inicial.

Bibliografia

1. Rangel, R. L., "Development of a Graphic Program for Structural Analysis of Linear Element Models", PUC-Rio, 2016.
2. Martha, L.F., "Análise de Estruturas: Conceitos e Métodos Básicos", Elsevier, 2010.
3. Martha, L.F., "Análise Matricial de Estruturas: Aplicada a Modelos Lineares", Elsevier, 2017.
4. Santana, M. V. B., "Desenvolvimento de Sistema Computacional via MATLAB/GUI (Graphical User Interface) para Análise Geometricamente Não Linear de Estruturas", UFOP, 2015.
5. Soriano, H. L., "Análise de Estruturas: Formulação Matricial e Implementação Computacional", Ciência Moderna, 2005.
6. Paz, M., "Structural Dynamics: Theory and Computation", Kluwer Academic Publishers, 1980.
7. Register, A. H., "A Guide to MATLAB Object-Oriented Programming", Chapman & Hall/CRC, 2007.
8. Attaway, S., "Matlab: a Practical Introduction to Programming and Problem Solving", 4th edition, Butterworth-Heinemann/Elsevier, 2016.
9. Chapman, S. J., "MATLAB Programming for Engineers", 5th edition, Chegg, 2016.
10. Almeida, M. C. F., "Estruturas Isostáticas", Oficina de Textos, 2011.
11. Oliveira, J. L. P., "Aplicação da Integração Direta em Modelo de Elementos Finitos para Análise Não-Linear em Rotodinâmica", UFRJ, 2007.

Apêndice A Coeficientes de Rigidez Locais para Elementos Lineares

A.1 Coeficientes de Rigidez Axiais:

$$[k_{ea}] = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \quad (\text{A.1})$$

A.2 Coeficientes de Rigidez Torsionais:

Elemento com extremidades livres:

$$[k_{et}] = \begin{bmatrix} \frac{GJ}{L} & -\frac{GJ}{L} \\ -\frac{GJ}{L} & \frac{GJ}{L} \end{bmatrix} \quad (\text{A.2})$$

Elemento com extremidades rotuladas:

$$[k_{er}] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{A.3})$$

A.3 Coeficientes de Rigidez Flexão:

Parâmetros usados na teoria de viga de Timoshenko (para teoria de viga de Euler-Bernoulli, $\Omega = 0$):

$$\Omega = \frac{EI}{GA_s L^2} \quad (\text{A.4})$$

$$\lambda = 1 + 3\Omega \quad (\text{A.5})$$

$$\mu = 1 + 12\Omega \quad (\text{A.6})$$

$$\lambda = 1 + 3\Omega \quad (\text{A.7})$$

Elemento com extremidades livres:

$$[\text{kef}]_{xy} = \begin{bmatrix} \frac{12EI}{\mu L^3} & \frac{6EI}{\mu L^2} & -\frac{12EI}{\mu L^3} & \frac{6EI}{\mu L^2} \\ \frac{6EI}{\mu L^2} & \frac{\lambda 4EI}{\mu L} & -\frac{6EI}{\mu L^2} & \frac{\gamma 2EI}{\mu L} \\ -\frac{12EI}{\mu L^3} & -\frac{6EI}{\mu L^2} & \frac{12EI}{\mu L^3} & -\frac{6EI}{\mu L^2} \\ \frac{6EI}{\mu L^2} & \frac{\gamma 2EI}{\mu L} & -\frac{6EI}{\mu L^2} & \frac{\lambda 4EI}{\mu L} \end{bmatrix} \quad (\text{A.7})$$

$$[\text{kef}]_{xz} = \begin{bmatrix} \frac{12EI}{\mu L^3} & -\frac{6EI}{\mu L^2} & -\frac{12EI}{\mu L^3} & -\frac{6EI}{\mu L^2} \\ -\frac{6EI}{\mu L^2} & \frac{\lambda 4EI}{\mu L} & \frac{6EI}{\mu L^2} & \frac{\gamma 2EI}{\mu L} \\ -\frac{12EI}{\mu L^3} & \frac{6EI}{\mu L^2} & \frac{12EI}{\mu L^3} & \frac{6EI}{\mu L^2} \\ -\frac{6EI}{\mu L^2} & \frac{\gamma 2EI}{\mu L} & \frac{6EI}{\mu L^2} & \frac{\lambda 4EI}{\mu L} \end{bmatrix} \quad (\text{A.8})$$

Elemento com extremidade inicial rotulada:

$$[\text{kef}]_{xy} = \begin{bmatrix} \frac{3EI}{\lambda L^3} & 0 & -\frac{3EI}{\lambda L^3} & \frac{3EI}{\lambda L^2} \\ 0 & 0 & 0 & 0 \\ -\frac{3EI}{\lambda L^3} & 0 & \frac{3EI}{\lambda L^3} & -\frac{3EI}{\lambda L^2} \\ \frac{3EI}{\lambda L^2} & 0 & -\frac{3EI}{\lambda L^2} & \frac{3EI}{\lambda L} \end{bmatrix} \quad (\text{A.9})$$

$$[\text{kef}]_{xz} = \begin{bmatrix} \frac{3EI}{\lambda L^3} & 0 & -\frac{3EI}{\lambda L^3} & -\frac{3EI}{\lambda L^2} \\ 0 & 0 & 0 & 0 \\ -\frac{3EI}{\lambda L^3} & 0 & \frac{3EI}{\lambda L^3} & \frac{3EI}{\lambda L^2} \\ -\frac{3EI}{\lambda L^2} & 0 & \frac{3EI}{\lambda L^2} & \frac{3EI}{\lambda L} \end{bmatrix} \quad (\text{A.10})$$

Elemento com extremidade final rotulada:

$$[\text{kef}]_{xy} = \begin{bmatrix} \frac{3EI}{\lambda L^3} & \frac{3EI}{\lambda L^2} & -\frac{3EI}{\lambda L^3} & 0 \\ \frac{3EI}{\lambda L^2} & \frac{3EI}{\lambda L} & -\frac{3EI}{\lambda L^2} & 0 \\ -\frac{3EI}{\lambda L^3} & -\frac{3EI}{\lambda L^2} & \frac{3EI}{\lambda L^3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.11})$$

$$[\text{kef}]_{xz} = \begin{bmatrix} \frac{3EI}{\lambda L^3} & -\frac{3EI}{\lambda L^2} & -\frac{3EI}{\lambda L^3} & 0 \\ -\frac{3EI}{\lambda L^2} & \frac{3EI}{\lambda L} & \frac{3EI}{\lambda L^2} & 0 \\ -\frac{3EI}{\lambda L^3} & \frac{3EI}{\lambda L^2} & \frac{3EI}{\lambda L^3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.12})$$

Elemento com extremidades rotuladas:

$$[\text{kef}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.13})$$

Apêndice B Reações de Engastamento Perfeito para Cargas Distribuídas Linearmente

B.1 Reações de Engastamento para Carga Distribuída Axialmente

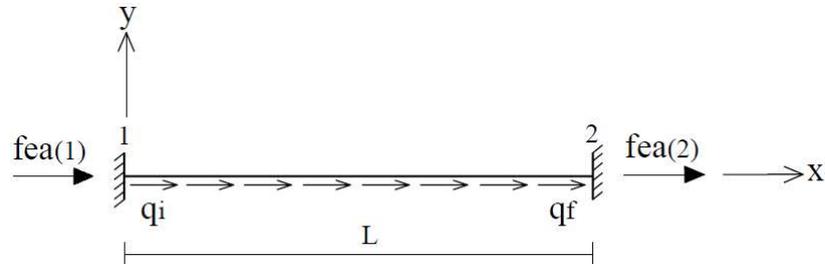


Figura B.1 – Reações de Engastamento Perfeito para Cargas Distribuídas Axialmente no Plano XY

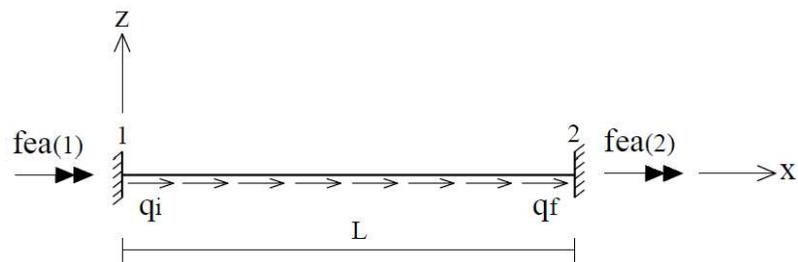


Figura B.2 – Reações de Engastamento Perfeito para Cargas Distribuídas Axialmente no Plano XZ

$$[\text{fea}]_{xy} = \begin{bmatrix} -\left(\frac{q_i L}{3} + \frac{q_f L}{6}\right) \\ -\left(\frac{q_i L}{6} + \frac{q_f L}{3}\right) \end{bmatrix} \quad (\text{B.1})$$

$$[\text{fea}]_{xz} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{B.2})$$

B.2 Reações de Engastamento Perfeito para Cargas Transversalmente Distribuídas

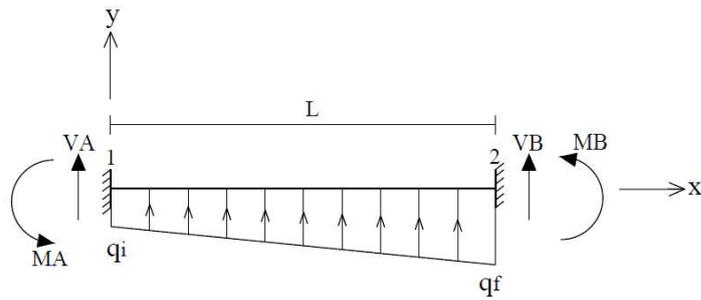


Figura B.3 – Reações de Engastamento Perfeito para Cargas Distribuídas Axialmente no Plano XY

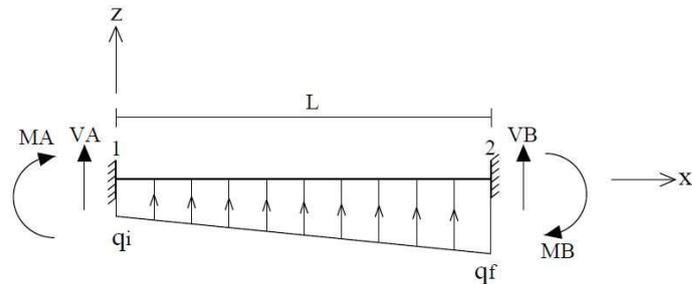


Figura B.4 – Reações de Engastamento Perfeito para Cargas Distribuídas Axialmente no Plano XZ

$$q_0 = q_i \quad (\text{B.3})$$

$$q_1 = q_f - q_i \quad (\text{B.4})$$

Parâmetros usados na teoria de viga de Timoshenko (para teoria de viga de Euler-Bernoulli, $\Omega = 0$):

$$\Omega = \frac{EI}{GA_s L^2} \quad (\text{B.5})$$

$$\gamma = 1 - 6\Omega \quad (\text{B.6})$$

$$\lambda = 1 + 3\Omega \quad (\text{B.7})$$

$$\mu = 1 + 12\Omega \quad (\text{B.8})$$

$$r = 1 + \frac{40\Omega}{3} \quad (\text{B.9})$$

$$s = 1 + 15\Omega \quad (\text{B.10})$$

$$t = 1 + \frac{80\Omega}{7} \quad (\text{B.11})$$

$$u = 1 + 10\Omega \quad (\text{B.12})$$

Valores das reações de engastamento perfeito para elementos com extremidades contínuas:

$$V_A = -\left(\frac{q_0 L}{2} + \frac{3q_1 L r}{20\mu}\right) \quad (\text{B.13})$$

$$M_{A_{xy}} = -\left(\frac{q_0 L^2}{12} + \frac{q_1 L^2}{30\mu}\right) \quad (\text{B.14})$$

$$M_{A_{xz}} = \left(\frac{q_0 L^2}{12} + \frac{q_1 L^2}{30\mu}\right) \quad (\text{B.15})$$

$$V_B = -\left(\frac{q_0 L}{2} + \frac{7q_1 L t}{20\mu}\right) \quad (\text{B.16})$$

$$M_{B_{xy}} = \left(\frac{q_0 L^2}{12} + \frac{q_1 L^2 u}{20\mu}\right) \quad (\text{B.17})$$

$$M_{B_{xz}} = -\left(\frac{q_0 L^2}{12} + \frac{q_1 L^2 u}{20\mu}\right) \quad (\text{B.17})$$

Coefficientes de transmissão de momentos: (B.18)

$$t_{AB} = t_{BA} = \frac{\gamma}{2\lambda} \quad (\text{B.19})$$

Elemento com extremidades contínuas:

$$[fef] = \begin{bmatrix} VA \\ MA \\ VB \\ MB \end{bmatrix} \quad (B.20)$$

Elemento com nó inicial rotulado:

$$[fef]_{xy} = \begin{bmatrix} VA - \frac{MA(1 + tAB)}{L} \\ 0 \\ VB + \frac{MA(1 + tAB)}{L} \\ MB - MA \times tAB \end{bmatrix} \quad (B.21)$$

$$[fef]_{xz} = \begin{bmatrix} VA + \frac{MA(1 + tAB)}{L} \\ 0 \\ VB - \frac{MA(1 + tAB)}{L} \\ MB - MA \times tAB \end{bmatrix} \quad (B.22)$$

Elemento com nó final rotulado:

$$[fef]_{xy} = \begin{bmatrix} VA - \frac{MB(1 + tBA)}{L} \\ MA - MB \times tBA \\ VB + \frac{MB(1 + tBA)}{L} \\ 0 \end{bmatrix} \quad (B.23)$$

$$[fef]_{xz} = \begin{bmatrix} VA + \frac{MB(1 + tBA)}{L} \\ MA - MB \times tBA \\ VB - \frac{MB(1 + tBA)}{L} \\ 0 \end{bmatrix} \quad (B.24)$$

Elemento com extremidades rotuladas:

$$[\text{fef}] = \begin{bmatrix} -\left(\frac{q_0L}{2} + \frac{q_1L}{6}\right) \\ 0 \\ -\left(\frac{q_0L}{2} + \frac{q_1L}{3}\right) \\ 0 \end{bmatrix} \quad (\text{B.25})$$

Apêndice C Reações de Engastamento Perfeito para Variação de Temperatura

C.1 Reações de Engastamento Perfeito Axiais para Variação de Temperatura

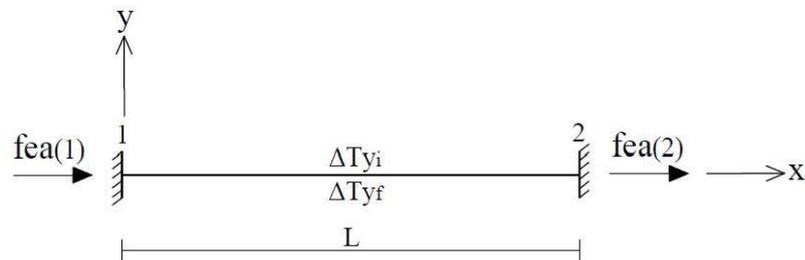


Figura C.1 – Reações de engastamento perfeito axiais para variação de temperatura no plano XY

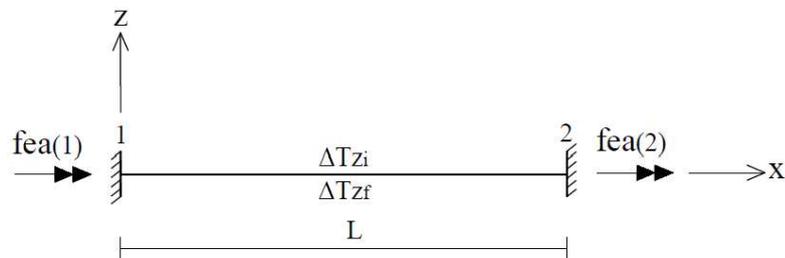


Figura C.2 – Reações de engastamento perfeito axiais para variação de temperatura no plano XZ

$$\Delta T_{x_{xy}} = \frac{\Delta T_{y_i} + \Delta T_{y_f}}{2} \quad (C.1)$$

$$\Delta T_{x_{xz}} = \frac{\Delta T_{z_i} + \Delta T_{z_f}}{2} \quad (C.2)$$

$$[fea] = \begin{bmatrix} EA \times \alpha \times \Delta T_x \\ -EA \times \alpha \times \Delta T_x \end{bmatrix} \quad (C.3)$$

C.2 Reações de Engastamento Perfeito para Gradiente de Temperatura

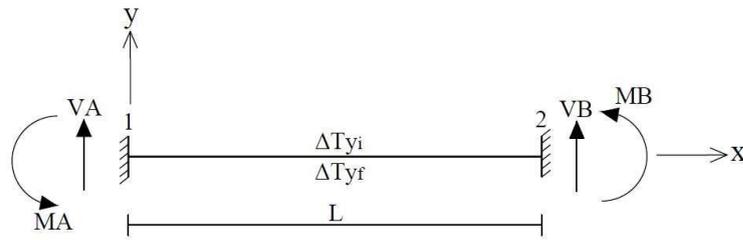


Figura C.3 - Reações de engastamento perfeito para gradiente de temperatura no plano XY

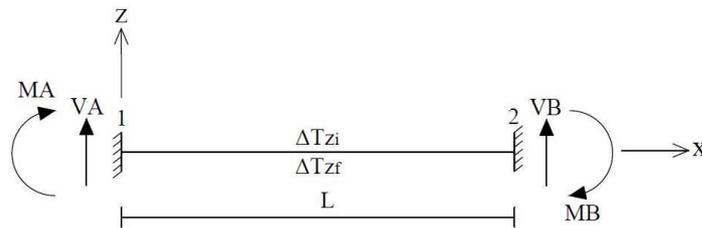


Figura C.4 - Reações de engastamento perfeito para gradiente de temperatura no plano XZ

$$\Delta T_y = \Delta T_{y_f} - \Delta T_{y_i} \quad (C.4)$$

$$\Delta T_z = \Delta T_{z_f} - \Delta T_{z_i} \quad (C.5)$$

Parâmetros usados na teoria de viga de Timoshenko (para teoria de viga de Euler-Bernoulli, $\Omega = 0$):

$$\Omega = \frac{EI}{GA_s L^2} \quad (C.6)$$

$$\gamma = 1 - 6\Omega \quad (C.7)$$

$$\lambda = 1 + 3\Omega \quad (C.8)$$

Reações de engastamento perfeito para um elemento com extremidades contínuas:

$$MA_{xy} = \frac{EI \times \alpha \times \Delta Ty}{h_y} \quad (C.9)$$

$$MA_{xz} = -\frac{EI \times \alpha \times \Delta Tz}{h_z} \quad (C.10)$$

$$MB_{xy} = -\frac{EI \times \alpha \times \Delta Ty}{h_y} \quad (C.11)$$

$$MB_{xz} = \frac{EI \times \alpha \times \Delta Tz}{h_z} \quad (C.12)$$

Coefficiente de transmissão de momentos:

$$t_{AB} = t_{BA} = \frac{\gamma}{2\lambda} \quad (C.13)$$

Elemento com extremidades contínuas:

$$[f_{ef}] = \begin{bmatrix} 0 \\ MA \\ 0 \\ MB \end{bmatrix} \quad (B.14)$$

Elemento com nó inicial rotulado:

$$[f_{ef}]_{xy} = \begin{bmatrix} \frac{MA(1 + t_{AB})}{L} \\ 0 \\ -\frac{MA(1 + t_{AB})}{L} \\ MB - MA \times t_{AB} \end{bmatrix} \quad (C.15)$$

$$[f_{ef}]_{xz} = \begin{bmatrix} -\frac{MA(1+t_{AB})}{L} \\ 0 \\ \frac{MA(1+t_{AB})}{L} \\ MB - MA \times t_{AB} \end{bmatrix} \quad (C.16)$$

Elemento com nó final rotulado:

$$[f_{ef}]_{xy} = \begin{bmatrix} \frac{MB(1+t_{BA})}{L} \\ MA - MB \times t_{BA} \\ -\frac{MB(1+t_{BA})}{L} \\ 0 \end{bmatrix} \quad (C.17)$$

$$[f_{ef}]_{xz} = \begin{bmatrix} -\frac{MB(1+t_{BA})}{L} \\ MA - MB \times t_{BA} \\ \frac{MB(1+t_{BA})}{L} \\ 0 \end{bmatrix} \quad (C.18)$$

Elemento com extremidades rotuladas:

$$[f_{ef}] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (C.19)$$

Apêndice D Funções de Forma

Parâmetros usados na teoria de viga de Timoshenko (para teoria de viga de Euler-Bernoulli, $\Omega = 0$):

$$\Omega = \frac{EI}{GA_s L^2} \quad (D.1)$$

$$\gamma = 1 - 6\Omega \quad (D.2)$$

$$\lambda = 1 + 3\Omega \quad (D.3)$$

$$\mu = 1 + 12\Omega \quad (D.4)$$

Funções de forma para comportamento axial:

$$N_{a_1} = 1 - \frac{x}{L} \quad (D.5)$$

$$N_{a_2} = \frac{x}{L} \quad (D.6)$$

Funções de forma para comportamento a flexão em xy ($I = I_z$ e $A_s = A_y$):

Elemento com extremidades livres:

$$N_{f1_{xy}} = 1 - \frac{12\Omega x}{L\mu} - \frac{3x^2}{L^2\mu} + \frac{2x^3}{L^3\mu} \quad (D.7)$$

$$N_{f2_{xy}} = x - \frac{6\Omega x}{\mu} - \frac{2\lambda x^2}{L\mu} + \frac{x^3}{L^2\mu} \quad (D.8)$$

$$N_{f3_{xy}} = \frac{12\Omega x}{L\mu} + \frac{3x^2}{L^2\mu} - \frac{2x^3}{L^3\mu} \quad (D.9)$$

$$N_{f4_{xy}} = \frac{-6\Omega x}{\mu} - \frac{\gamma x^2}{L\mu} + \frac{x^3}{L\mu} \quad (D.10)$$

Elemento com extremidade inicial rotulada:

$$N_{f1_{xy}} = 1 - \frac{3x}{2L\lambda} - \frac{3\Omega x}{L\lambda} + \frac{x^3}{2L^3\lambda} \quad (D.11)$$

$$N_{f2_{xy}} = 0 \quad (D.12)$$

$$N_{f3_{xy}} = \frac{3x}{2L\lambda} + \frac{3\Omega x}{L\lambda} - \frac{x^3}{2L^3\lambda} \quad (D.13)$$

$$N_{f4_{xy}} = -\frac{\gamma x}{2\lambda} - \frac{3\Omega x}{\lambda} + \frac{x^3}{2L^2\lambda} \quad (D.14)$$

Elemento com extremidade final rotulada:

$$N_{f1_{xy}} = 1 - \frac{3\Omega x}{L\lambda} - \frac{3x^2}{2L^2\lambda} + \frac{x^3}{2L^3\lambda} \quad (D.15)$$

$$N_{f2_{xy}} = x - \frac{3\Omega x}{\lambda} - \frac{3x^2}{2L\lambda} + \frac{x^3}{2L^2\lambda} \quad (D.16)$$

$$N_{f3_{xy}} = \frac{3\Omega x}{L\lambda} + \frac{3x^2}{2L^2\lambda} - \frac{x^3}{2L^3\lambda} \quad (D.17)$$

$$N_{f4_{xy}} = 0 \quad (D.18)$$

Elemento com extremidades rotuladas:

$$N_{f1_{xy}} = 1 - \frac{x}{L} \quad (D.19)$$

$$N_{f2_{xy}} = 0 \quad (D.20)$$

$$N_{f3_{xy}} = \frac{x}{L} \quad (D.20)$$

$$N_{f4_{xy}} = 0 \quad (D.22)$$

Funções de forma para comportamento a flexão em xz (I = Iz e As = Az):

Elemento com extremidades livres

$$N_{f1_{xz}} = 1 - \frac{12\Omega x}{L\mu} - \frac{3x^2}{L^2\mu} + \frac{2x^3}{L^3\mu} \quad (D.23)$$

$$N_{f2_{xz}} = -x + \frac{6\Omega x}{\mu} + \frac{2\lambda x^2}{L\mu} - \frac{x^3}{L^2\mu} \quad (D.24)$$

$$N_{f3_{xz}} = \frac{12\Omega x}{L\mu} + \frac{3x^2}{L^2\mu} - \frac{2x^3}{L^3\mu} \quad (D.25)$$

$$N_{f4_{xz}} = \frac{6\Omega x}{\mu} + \frac{\gamma x^2}{L\mu} - \frac{x^3}{L\mu} \quad (D.26)$$

Elemento com extremidade inicial rotulada:

$$N_{f1_{xz}} = 1 - \frac{3x}{2L\lambda} - \frac{3\Omega x}{L\lambda} + \frac{x^3}{2L^3\lambda} \quad (D.27)$$

$$N_{f2_{xz}} = 0 \quad (D.28)$$

$$N_{f3_{xz}} = \frac{3x}{2L\lambda} + \frac{3\Omega x}{L\lambda} - \frac{x^3}{2L^3\lambda} \quad (D.29)$$

$$N_{f4_{xz}} = \frac{\gamma x}{2\lambda} + \frac{3\Omega x}{\lambda} - \frac{x^3}{2L^2\lambda} \quad (D.30)$$

Elemento com extremidade final rotulada

$$N_{f1_{xz}} = 1 - \frac{3\Omega x}{L\lambda} - \frac{3x^2}{2L^2\lambda} + \frac{x^3}{2L^3\lambda} \quad (\text{D.31})$$

$$N_{f2_{xz}} = -x + \frac{3\Omega x}{\lambda} + \frac{3x^2}{2L\lambda} - \frac{x^3}{2L^2\lambda} \quad (\text{D.32})$$

$$N_{f3_{xz}} = \frac{3\Omega x}{L\lambda} + \frac{3x^2}{2L^2\lambda} - \frac{x^3}{2L^3\lambda} \quad (\text{D.33})$$

$$N_{f4_{xz}} = 0 \quad (\text{D.34})$$

Elemento com extremidades rotuladas:

$$N_{f1_{xz}} = 1 - \frac{x}{L} \quad (\text{D.35})$$

$$N_{f2_{xz}} = 0 \quad (\text{D.36})$$

$$N_{f3_{xz}} = \frac{x}{L} \quad (\text{D.37})$$

$$N_{f4_{xz}} = 0 \quad (\text{D.38})$$

Apêndice E Deslocamentos Internos para Cargas Linearmente Distribuídas

E.1 Deslocamento Axial

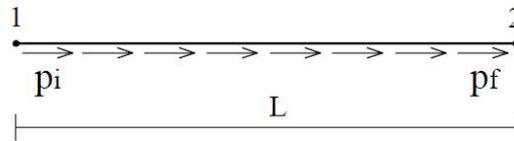


Figura E.1 – Elementos sujeitos a carregamento distribuídos axialmente

$$p_0 = p_i \quad (\text{E.1})$$

$$p_1 = p_f - p_i \quad (\text{E.2})$$

$$u(x) = p_0 \left(-\frac{x^2}{2EA} + \frac{Lx}{2EA} \right) + p_1 \left(-\frac{x^3}{6EAL} + \frac{Lx}{6EA} \right) \quad (\text{E.3})$$

E.2 Deslocamentos Transversais

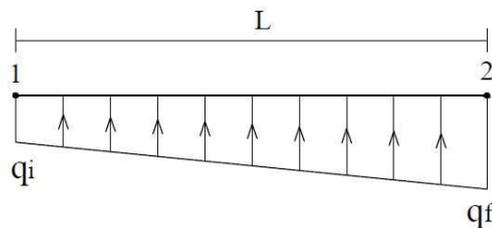


Figura E.2 – Elementos sujeitos a carregamento distribuídos transversalmente

$$q_0 = q_i \quad (\text{E.4})$$

$$q_1 = q_f - q_i \quad (\text{E.5})$$

$$v(x) = v_{q_0}(x) + v_{q_1}(x) \quad (\text{E.6})$$

Parâmetros usados na teoria de viga de Timoshenko (para teoria de viga de Euler-Bernoulli, $\Omega = 0$):

$$\Omega = \frac{EI}{GA_s L^2} \quad (\text{E.7})$$

$$\xi = 1 + 48\Omega \quad (\text{E.8})$$

$$\zeta = 1 + \frac{40\Omega}{3} \quad (\text{E.9})$$

$$\eta = 1 + 15\Omega \quad (\text{E.10})$$

$$\lambda = 1 + 3\Omega \quad (\text{E.11})$$

$$\mu = 1 + 12\Omega \quad (\text{E.12})$$

$$\nu = 1 + 24\Omega \quad (\text{E.13})$$

$$\vartheta = 1 + 8\Omega \quad (\text{E.14})$$

$$\varsigma = 1 + 5\Omega \quad (\text{E.15})$$

$$\psi = 1 + \frac{24\Omega}{5} \quad (\text{E.16})$$

$$\omega = 1 + \frac{20\Omega}{9} \quad (\text{E.17})$$

Elemento com extremidades contínuas:

$$v_{q0}(x) = q_0 \left(\frac{x^4}{24 EI} - \frac{L v x^3}{12 EI \mu} + \frac{L^2 x^2 \xi}{24 EI \mu} - \frac{L^2 \Omega x^2}{2 EI} + \frac{L^3 \Omega v x}{2 EI \mu} \right) \quad (\text{E.18})$$

$$v_{q1}(x) = q_1 \left(\frac{x^5}{120 EI L} - \frac{L x^3 \zeta}{40 EI \mu} - \frac{L \Omega x^3}{6 EI} + \frac{L^2 \eta x^2}{60 EI \mu} + \frac{3 L^3 \Omega x \zeta}{20 EI \mu} \right) \quad (\text{E.19})$$

Elemento com nó inicial rotulado:

$$v_{q0}(x) = q_0 \left(\frac{x^4}{24 EI} - \frac{L \vartheta x^3}{16 EI \lambda} - \frac{L^2 \Omega x^2}{2 EI} + \frac{L^3 x \xi}{48 EI \lambda} + \frac{3 L^3 \Omega \vartheta x}{8 EI \lambda} \right) \quad (\text{E.20})$$

$$v_{q1}(x) = q_1 \left(\frac{x^5}{120 EI L} - \frac{L \zeta x^3}{60 EI \lambda} - \frac{L \Omega x^3}{6 EI} + \frac{L^3 \eta x}{120 EI \lambda} + \frac{L^3 \Omega \zeta x}{10 EI \lambda} \right) \quad (\text{E.21})$$

Elemento com nó final rotulado:

$$v_{q0}(x) = q_0 \left(\frac{x^4}{24 EI} - \frac{5 L \psi x^3}{48 EI \lambda} + \frac{L^2 \mu x^2}{16 EI \lambda} - \frac{L^2 \Omega x^2}{2 EI} + \frac{5 L^3 \Omega \psi x}{8 EI \lambda} \right) \quad (\text{E.22})$$

$$v_{q1}(x) = q_1 \left(\frac{x^5}{120 EI L} - \frac{3 L \omega x^3}{80 EI \lambda} - \frac{L \Omega x^3}{6 EI} + \frac{7 L^2 x^2}{240 EI \lambda} + \frac{9 L^3 \Omega \omega x}{40 EI \lambda} \right) \quad (\text{E.23})$$

Elemento com extremidades rotuladas:

$$v_{q0}(x) = q_0 \left(\frac{x^4}{24 EI} - \frac{L x^3}{12 EI} - \frac{L^2 \Omega x^2}{2 EI} + \frac{L^3 \mu x}{24 EI} + \frac{L^3 \Omega x}{2 EI} \right) \quad (\text{E.24})$$

$$v_{q1}(x) = q_1 \left(\frac{x^5}{120 EI L} - \frac{L x^3}{36 EI} - \frac{L \Omega x^3}{6 EI} + \frac{7 L^3 x}{360 EI} + \frac{L^3 \Omega x}{6 EI} \right) \quad (\text{E.25})$$

Apêndice F Deslocamentos Internos para Variação de Temperatura

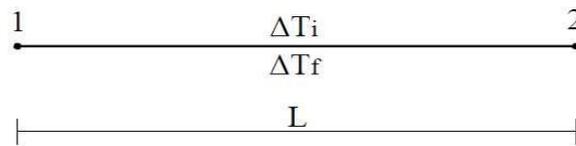


Figura F.1 – Elemento com variação de temperatura

$$\Delta T = \Delta T_f - \Delta T_i \quad (\text{F.1})$$

F.1 Deslocamentos Axiais

$$u(x) = 0 \quad (\text{F.2})$$

F.2 Deslocamentos Transversais

Parâmetros usados na teoria de viga de Timoshenko (para teoria de viga de Euler-Bernoulli, $\Omega = 0$):

$$\Omega = \frac{EI}{GA_s L^2} \quad (\text{F.3})$$

$$\gamma = 1 - 6\Omega \quad (\text{F.4})$$

$$\lambda = 1 + 3\Omega \quad (\text{F.5})$$

$$\mu = 1 + 12\Omega \quad (\text{F.6})$$

Elemento com extremidades contínuas:

$$v(x) = 0 \quad (\text{F.7})$$

Elemento com nó inicial rotulado:

$$v(x) = \Delta T \left(-\frac{x^3}{4L\lambda} + \frac{x^2}{2} - \frac{L\mu x}{4\lambda} + \frac{3L\Omega x}{2\lambda} \right) \quad (\text{F.8})$$

Elemento com nó final rotulado:

$$v(x) = \Delta T \left(\frac{x^3}{4L\lambda} - \frac{\gamma x^2}{4\lambda} - \frac{3L\Omega x}{2\lambda} \right) \quad (\text{F.9})$$

Elemento com extremidades rotuladas:

$$v(x) = \Delta T \left(\frac{x^2}{2} - \frac{Lx}{2} \right) \quad (\text{F.10})$$