

UNIVERSIDADE FEDERAL DO CEARÁ  
CENTRO DE TECNOLOGIA  
ENGENHARIA CIVIL  
DEPARTAMENTO DE ENGENHARIA ESTRUTURAL E CONSTRUÇÃO CIVIL

ELIAS SARAIVA BARROSO

**ANÁLISE COMPUTACIONAL DE GRELHAS**

FORTALEZA  
2013

ELIAS SARAIVA BARROSO

**ANÁLISE COMPUTACIONAL DE GRELHAS**

Monografia submetida à Coordenação do Curso de Engenharia Civil da Universidade Federal do Ceará, como requisito parcial para obtenção do grau de Engenheiro Civil.

Orientador: Prof. D. Sc. Evandro Parente Junior

FORTALEZA  
2013

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca de Ciências e Tecnologia

---

B285a

Barroso, Elias Saraiva.

Análise computacional de grelhas / Elias Saraiva Barroso. – 2013.  
83 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Estrutural e Construção Civil, Curso de Engenharia Civil, Fortaleza, 2013.

Orientação: Prof. Dr. Evandro Parente Júnior.

1. Grelhas (Engenharia de estruturas). 2. Engenharia de estruturas – Programas de computador. I.  
Título.

---

CDD 620

ELIAS SARAIVA BARROSO

ANÁLISE COMPUTACIONAL DE GRELHAS

Monografia apresentada ao Curso de Engenharia Civil da Universidade Federal do Ceará, como requisito parcial para a obtenção do Título de Engenheiro Civil.

Aprovada em 08 / 02 / 2013.

BANCA EXAMINADORA

---

Prof. D. Sc. Evandro Parente Junior (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Prof. D. Sc. Antônio Macário Cartaxo de Melo  
Universidade Federal do Ceará (UFC)

---

Prof. Prof. D. Sc. Tereza Denyse Pereira de Araújo  
Universidade Federal do Ceará (UFC)

*A Deus.*

*Aos meus pais,*

*Eliseu Barroso e Maria José.*

## **AGRADECIMENTOS**

Ao Prof. Evandro pela excelente orientação neste trabalho e na bolsa de iniciação científica.

Ao CNPQ, pelo apoio financeiro prestado em minha iniciação científica.

Aos meus irmãos Eliseu Neto e Lígia, que tanto me ajudaram durante minha vida.

Aos colegas de laboratório Iuri Barcelos, Luiz Antônio, Edson Dantas e Pedro Sanderson, por toda ajuda que me deram ao longo deste trabalho.

A todos os meus professores da graduação, que contribuirão para formação que tenho hoje.

Ao professor Rodrigo Amaral de Codes, que muito me ajudou como coordenador no período do segundo ano de faculdade.

Aos professores Silvrano Adonias, Evandro Parente e Verônica Castelo Branco, que me ensinaram da forma mais rigorosa, como devo estudar e trabalhar.

A todos os meus amigos.

## **Resumo**

Na engenharia as análises computacionais são importantes ferramentas de projeto, pois viabilizam soluções de problemas que necessitam de grande quantidade de operações matemáticas. Este trabalho teve como objetivo desenvolver um programa computacional para análise estrutural de grelhas, onde foi implementada uma interface gráfica para modelagem e visualização tridimensional do problema físico. Também foi adicionado funcionalidades ao analisador estrutural de Elementos Finitos FAST, para processamento de problemas de análise linear contendo elementos de grelha, este analisador foi desenvolvido na Universidade Federal do Ceará. O programa desenvolvido foi verificado utilizando exemplos de grelhas presentes na literatura. Foi realizado um estudo sobre uma das importantes aplicações das estruturas de grelha, o processo de análise de lajes pelo Método de Grelha Equivalente, mostrando sua importância na atual prática de engenharia estrutural brasileira. Também foram realizados exemplos de analogia de grelha utilizando o programa desenvolvido, de forma a avaliar o desempenho do programa para utilização deste método.

Palavras chaves: Análise de grelhas, Implementação computacional, Elementos Finitos.

## **ABSTRACT**

In engineering, computational analysis are important design tools, enabling solutions for problems that require large amount of mathematical operations. This study aimed to develop a computer program to analyze structural grids, which has been implemented in a graphical interface for modeling and visualization of three-dimensional physical problem. Were also added functionalities to Finite Element Structural Analyzer FAST, for processing linear analysis problems, containing elements of grid, this analyzer was developed at the Federal University of Ceará. The software was created using examples of grids found in the literature. A study was conducted on one of the important applications of grid structures, the analysis process of slabs by Method Equivalent Grid, demonstrating its importance in the current practice of structural engineering in Brazil. Samples of analogy grid were also performed by using the program developed in order to evaluate the performance of the program for this method.

Keywords: Analysis of grids, computational implementation, Finite Elements.

## LISTA DE FIGURAS

Figura 1 – Exemplo de uma grelha.....	16
Figura 2 – Exemplo do processo de espalhamento.....	18
Figura 3 – Elemento de grelha no eixo global.....	19
Figura 4 – Elemento de grelha no eixo local.....	19
Figura 5 – Coeficientes de rigidez de um elemento de grelha.....	20
Figura 6 – Esforços de engastamento perfeito.....	21
Figura 7 – Ângulo entre os eixos globais e locais de um elemento.....	21
Figura 8 – Elemento de grelha no sistema global.....	22
Figura 9 – Convenções de sinais dos esforços locais dos elementos.....	23
Figura 10 – Funções de forma de cada deslocabilidade do elemento de grelha.....	24
Figura 11 – Estrutura do programa FAST.....	25
Figura 12 – Exemplo de problema escrito em um arquivo de entrada.....	26
Figura 13 – Estrutura de classes dos elementos.....	27
Figura 14 – Estrutura de classes dos carregamentos.....	28
Figura 15 – Exemplo 1 de validação do programa FAST.....	29
Figura 16 – Processo de espalhamento do exemplo 1.....	31
Figura 17 – Resultados visualizados pelo programa GridCalc.....	33
Figura 18 – Exemplo 2 de validação do programa FAST.....	34
Figura 19 – Processo de espalhamento do exemplo 2.....	36
Figura 20 – Resultados visualizados pelo programa GridCalc.....	38
Figura 21 – Procedimento para realização de uma análise estrutural computacional.	39
Figura 22 – Editor de código Qt.....	40
Figura 23 – Editor de Interface do Qt, o Designer.Fonte: Autor.....	41
Figura 24 – Formato da interface do programa GridCalc para Windows.....	42
Figura 25 – Formato da interface do programa GridCalc para Linux.....	43
Figura 26 – Estrutura de classes do programa GridCalc.....	44
Figura 27 – Tela do pré-processador do programa GridCalc.....	45
Figura 28 – Representação do nó através de sua região de tolerância.....	46
Figura 29 – Correção da intercessão. ....	48
Figura 30 – Malha quadrada criada pelo programa GridCalc.....	50

Figura 31 – Tela do pós-processador do programa GridCalc.....	51
Figura 32 – Procedimento de desenho gráfico do OpenGL.....	52
Figura 33 – Exemplo de traçado de deformada, para dois números de divisões.....	53
Figura 34 – Exemplo de uma malha utilizada para análise de uma laje.....	55
Figura 35 – Exemplo de utilização da analogia de grelha.....	56
Figura 36 – Seção transversal T de um elemento de grelha para laje nervurada.....	57
Figura 37 – Carregamentos distribuídos sobre os elementos de grelha.....	59
Figura 38 – Laje do exemplo 1.....	61
Figura 39 – Malha 1x1 utilizada no exemplo 1.....	62
Figura 40 – Malha tridimensional 1m x 1m visualizada no programa GridCalc.....	63
Figura 41 – Deformada tridimensional visualizada no programa GridCalc.....	63
Figura 42 – Deformada da viga central da grelha do exemplo 1 de malha 1 m x 1m..	63
Figura 43 – Momento fletor da viga central da grelha do exemplo 1 de malha 1m x 1m.....	64
Figura 44 – Deslocamentos obtidos no exemplo 1, por elementos finitos, malha 1m x 1m. ....	64
Figura 45 – Momentos obtidos no exemplo 1, por elementos finitos, malha 1m x 1m. ....	65
Figura 46 – Malha 0,5x0,5 utilizada no exemplo 1.....	65
Figura 47 – Malha tridimensional 0,5x0,5 visualizada no programa GridCalc.....	66
Figura 48 – Deformada da viga central da grelha do exemplo 1 de malha 0,50x0,50.	66
Figura 49 – Momento fletor da viga central da grelha do exemplo 1 de malha 0,50m x 0,50m.....	67
Figura 50 – Deslocamentos obtidos no exemplo 1, por elementos finitos, malha 0m50m x 0,50m.....	67
Figura 51 – Momentos obtidos no exemplo 1, por elementos finitos, malha 0,5m x 0,5m.....	68
Figura 52 – Malha 0.25m x 0.25m utilizada no exemplo 1.....	68
Figura 53 – Malha tridimensional visualizada no programa GridCalc.....	69
Figura 54 – Deformada da viga central da grelha do exemplo 1 de malha 0,25x0,25.	69
Figura 55 – Momento fletor da viga central da grelha do exemplo 1 de malha	

0,25x0,25.....	70
Figura 56 – Deslocamentos obtidos no exemplo 1, por elementos finitos, malha 0,25m x 0,25m.....	70
Figura 57 – Momentos obtidos no exemplo 1, por elementos finitos, malha 0,25m x 0,25m.....	71
Figura 58 – Laje do exemplo 2.....	73
Figura 59 – Deformada da viga central da grelha do exemplo 2 de malha 1x1.....	74
Figura 60 – Momento fletor da viga central da grelha do exemplo 2 de malha 1x1....	74
Figura 61 – Deslocamentos obtidos no exemplo 2, por elementos finitos, malha 1m x 1m.....	75
Figura 62 – Momentos obtidos no exemplo 2, por elementos finitos, malha 1m x 1m. ....	75
Figura 63 – Deformada da viga central da grelha do exemplo 2 de malha 0,50x0,50.	76
Figura 64 – Momento fletor da viga central da grelha do exemplo 2 de malha 0,50x0,50.....	76
Figura 65 – Deslocamentos obtidos no exemplo 2, por elementos finitos, malha 0,5m x 0,5m.....	77
Figura 66 – Momentos obtidos no exemplo 2, por elementos finitos, malha 0,5m x 0,5m.....	77
Figura 67 – Deformada da viga central da grelha do exemplo 2 de malha 0,25x0,25.	78
Figura 68 – Momento fletor da viga central da grelha do exemplo 2 de malha 0,25x0,25.....	78
Figura 69 – Deslocamentos obtidos no exemplo 2, por elementos finitos, malha 0,25m x 0,25m.....	79
Figura 70 – Momentos obtidos no exemplo 2, por elementos finitos, malha 0,25m x 0,25m.....	79

## LISTA DE TABELAS

Tabela 1 – Funções utilizadas pela classe cGrid.....	27
Tabela 2 – Principais funções da interface gráfica utilizadas pelo pré-processador.....	45
Tabela 3 – Principais funções da estrutura de dados utilizadas pelo pré-processador..	45
Tabela 4 – Principais funções do OpenGL utilizadas no programa.....	52
Tabela 5 – Principais funções GLWidget utilizadas no programa.....	52
Tabela 6 – Resumo das propriedades dos elementos utilizadas no exemplo 1 e 2, para analogia de grelha.....	71
Tabela 7 – Resultados obtidos no exemplo 1.....	71
Tabela 8 – Resultados obtidos no exemplo 2.....	80

## SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Objetivos.....	14
1.1.1	Objetivo geral.....	14
1.1.2	Objetivos específicos.....	14
1.2	Metodologia do trabalho.....	15
1.3	Estrutura do texto.....	15
2	ANÁLISE ESTRUTURAL DE GRELHAS.....	16
2.1	Método da Rigidez Direta.....	16
2.2	Formulação do elemento de grelha.....	18
2.3	Implementação computacional do método de rigidez direta.....	24
2.4	Implementação computacional do elemento de grelha.....	26
2.5	Exemplos para verificação.....	28
2.5.1	Exemplo 1.....	28
2.5.2	Exemplo 2.....	33
3	MODELAGEM E VISUALIZAÇÃO DE GRELHAS.....	39
3.1	Análise estrutural computacional.....	39
3.2	Implementação do pré-processador.....	44
3.3	Implementação do pós-processador.....	50
4	ANÁLISE DE LAJES PELO MÉTODO DA ANALOGIA DE GRELHA.....	55
4.1	Analogia de grelha.....	56
4.2	Rigidez a flexão.....	57
4.3	Rigidez à torção.....	58
4.4	Carregamento.....	59
4.5	Exemplos.....	60
4.5.1	Exemplo 1.....	60
4.5.2	Exemplo 2.....	72
5	CONCLUSÃO.....	81
	REFERÊNCIAS BIBLIOGRÁFICAS.....	82
	ANEXOS.....	84

## 1. INTRODUÇÃO

A análise e dimensionamento de estruturas é uma das etapas de maior importância no projeto de uma edificação, por tratar diretamente com a segurança de pessoas. As análises computacionais estão sendo cada vez mais utilizadas, viabilizando o uso de procedimentos numéricos, alguns com resultados aproximados.

As lajes são estruturas das edificações responsáveis por suportar diretamente o peso das pessoas e objetos, por isso devem ser analisadas de forma segura. Atualmente a análise de lajes pode ser realizada pelo método de analogia da grelha equivalente, um método computacional aproximado.

A crescente utilização destas análises computacionais no universo profissional delega às escolas de engenharia o dever de preparar seus alunos para tais competências. Desta maneira, é de extrema importância que existam softwares de licença livre sobre análise estrutural, um exemplo disto é o programa Ftool, que é referência em análise estrutural de vigas e pórticos planos, amplamente utilizado em várias escolas de engenharia do país.

Por isto é de grande motivação que neste trabalho seja concebido um programa computacional que realize análise estrutural de grelhas, similar ao Ftool, possibilitando também um melhor aprendizado do método de analogia de grelha.

### 1.1. Objetivos

#### 1.1.1 *Objetivo geral*

Desenvolver um sistema computacional para análise estrutural de grelhas planas que integre a modelagem, análise e visualização.

#### 1.1.2 *Objetivos específicos*

- Implementar um analisador utilizando elementos finitos de grelha.
- Implementar uma interface gráfica de pré-processamento e pós-processamento.
- Aplicar a exemplos de grelha presentes na literatura, para verificação do programa.

- Utilizar o programa criado para análise de lajes pelo método da analogia de grelha.

## **1.2. Metodologia do trabalho**

A metodologia utilizada no trabalho consistiu inicialmente em realizar revisão sobre a linguagem de programação c/c++, as ferramentas Qt e OpenGL, análise estrutural de grelhas e o método de analogia de grelha. O programa foi desenvolvido à medida que as revisões bibliográficas forem sendo realizadas, onde foi continuamente verificado as funcionalidades de cada componente. Por fim foram realizados exemplos de grelhas e aplicação do método de analogia de grelha, para avaliação do desempenho do programa.

## **1.3. Estrutura do texto**

O presente trabalho é composto por 5 Capítulos.

O Capítulo 2 trata da análise estrutural de grelhas, onde é realizada uma revisão bibliográfica dos conceitos do método da rigidez direta, dando ênfase no elemento de grelha. Também é abordado a implementação computacional utilizada no analisador desenvolvido e realizados exemplos de verificação.

O Capítulo 3 trata dos algoritmos utilizados para implementação da interface gráfica existente no programa desenvolvido, sendo também realizado um estudo sobre as ferramentas computacionais utilizadas, o Qt e o OpenGL.

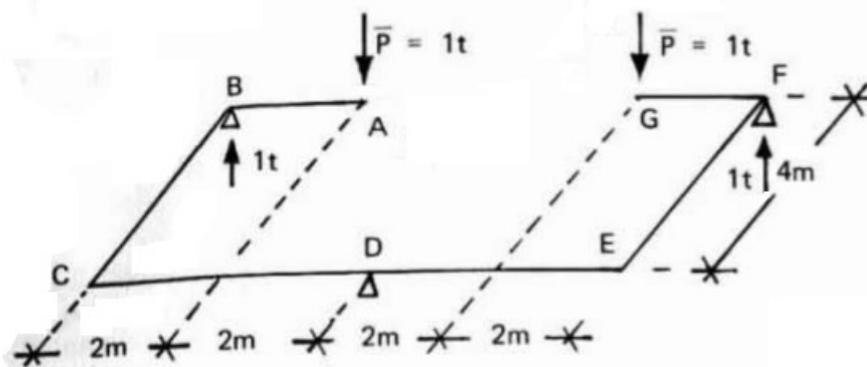
O Capítulo 4 realiza uma revisão bibliográfica do método de analogia de grelha, mostrando as principais considerações que devem ser adotadas para utilização do método. Também são apresentados exemplos para estudo da influência das malhas sobre os resultados obtidos, os quais são comparados com soluções analíticas da Teoria das Placas.

O Capítulo 5 trata das conclusões sobre o desempenho do programa desenvolvido e comentários sobre os exemplos realizados.

## 2. ANALISE ESTRUTURAL DE GRELHAS

As grelhas são estruturas reticuladas planas submetidas à cargas transversais ao seu plano, como ilustrado na Figura 1, de forma que aparecerão esforços cortante, momento fletor e momento torsor na estrutura.

Figura 1 – Exemplo de uma grelha.



Fonte: Süsskind (1980).

Neste capítulo será discutido o Método da Rigidez Direta no item 2.1, que é um método utilizado para análise estrutural. O item 2.2 trata da formulação utilizada no método da rigidez direta para o elemento de grelha. Os itens 2.3 e 2.4 tratam da implementação computacional do método da rigidez direta e do elemento de grelha. O item 2.5 mostra dois exemplos para validação da implementação computacional realizada.

### 2.1. Método da Rigidez Direta

O Método da Rigidez Direta é uma metodologia de análise estrutural que utiliza uma formulação adaptada do método dos deslocamentos, para implementação computacional. Este método é utilizado em elementos de barras, considerando as seguintes hipóteses:

- A validade da lei de Hooke.
- As deformações são consideradas irrelevantes quando comparadas com as dimensões da barra, o que permite considerar que o equilíbrio da estrutura ocorre na posição indeformada.

O objetivo do método é determinar os esforços e deslocamentos da estrutura, onde

devem ser atendidos, segundo MARTHA (2010):

- A condição de compatibilidade;
- As leis constitutivas dos materiais;
- As condições de equilíbrio.

A estrutura é dividida em casos básicos, onde são satisfeitas as condições de compatibilidade, mas não as de equilíbrio. Considerando as hipóteses adotadas, o equilíbrio da estrutura é atingido pela soma dos efeitos em cada caso básico. De maneira simplificada, a ideia é somar as cargas nodais geradas por ações externas (forças, momentos, recalques e variações de temperatura) e força elástica gerada pela deformação da estrutura, obtendo então a estrutura no seu estado final de deformações e esforços, onde deve ser satisfeito o equilíbrio da estrutura.

A Equação (1) representa o equilíbrio global da estrutura, onde  $\{\beta\}$  é o vetor de termos de carga gerado pelas ações externas,  $[K]$  é a matriz de rigidez global da estrutura,  $\{d\}$  é o vetor das deslocabilidades consideradas na estrutura e  $\{f\}$  é o valor das cargas nodais atuantes em cada nó onde fora considerado as deslocabilidades. Observe que o tamanho destes vetores e da matriz de rigidez é determinado pelo número de graus de liberdade do problema.

$$\{\beta\} + [K] \{d\} = \{f\} \quad (1)$$

Na prática, a equação (1) é utilizada para calcular os deslocamentos dos graus de liberdade. Uma vez conhecidos os deslocamentos, é possível determinar a força elástica exercida por eles em cada grau de liberdade, com isso pode-se calcular os esforços na estrutura.

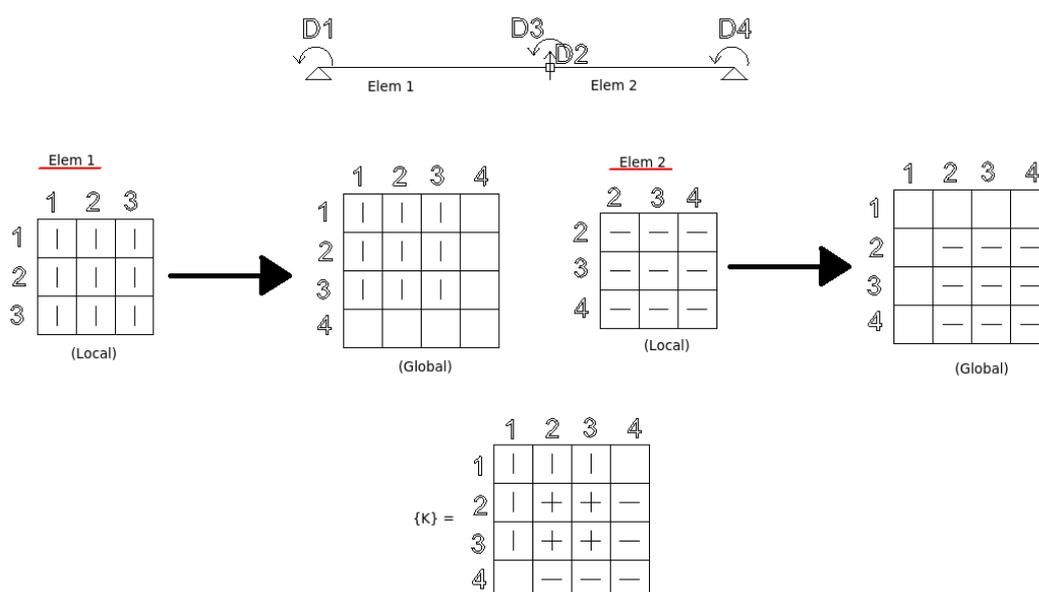
O método da rigidez propõe, sobre tudo, operacionalizar o procedimento de obtenção dos vetores e matriz de rigidez necessários para aplicação da Equação (1). Para isso a seguinte metodologia é adotada no método:

- a) O problema é resolvido por elemento, onde é adotado um eixo local para cada um, sendo calculadas as contribuições locais de cada elemento para os vetores  $\{\beta\}$ ,  $\{f\}$  e a matriz de rigidez  $[K]$ .
- b) As contribuições dos elementos são transformadas para coordenadas globais,

com a utilização de uma matriz de transformação para cada elemento.

- c) Os vetores  $\{\beta\}$ ,  $\{f\}$  e a matriz de rigidez  $[K]$ , são gerados por um procedimento chamado de espalhamento, que consiste em distribuir, para cada elemento do problema, sua contribuição para cada um desses vetores e matriz globais. A Figura 2 mostra um exemplo do processo de espalhamento, em uma estrutura que possui dois elementos de viga e 4 graus de liberdade.

Figura 2 – Exemplo do processo de espalhamento.



Fonte: Autor.

- d) Os deslocamentos globais da estrutura são calculados utilizando a Equação (1).  
e) Os esforços internos em cada elemento são calculados, utilizando a expressão:

$$\{g_i\} = [K_i] \{u_i\} + \{f_i\} \quad (2)$$

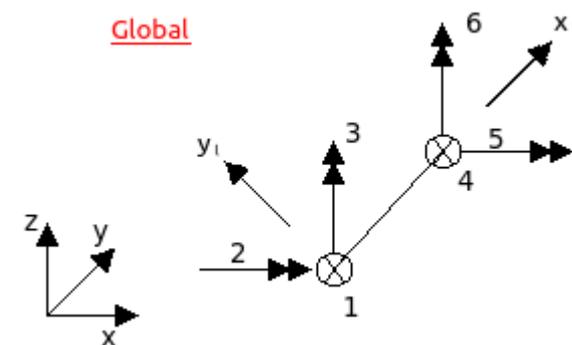
Onde  $\{g_i\}$  é o vetor de esforços internos,  $K_i$  é a matriz de rigidez local do elemento,  $u_i$  é o vetor dos deslocamentos locais do elemento e  $f_i$  é o vetor dos esforços de engastamento perfeito do elemento.

## 2.2. Formulação do elemento de grelha

O elemento de grelha possui seis graus de liberdade, duas translações na direção Z e quatro rotações, duas em torno do eixo X e duas em torno do eixo Y, onde é considerado a seguinte ordem;  $W_1, R_{x1}, R_{y1}, W_2, R_{x2}$  e  $R_{y2}$ . A Figura 3 ilustra um elemento de grelha com seus

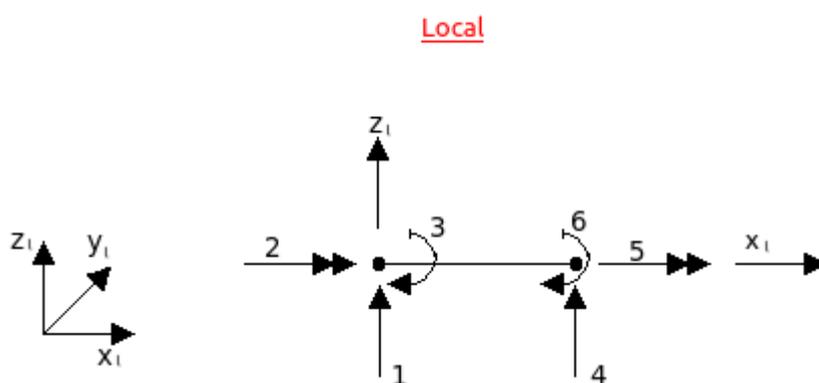
graus de liberdade considerando o eixo global e a Figura 4 mostra o mesmo elemento de grelha com seus graus de liberdade considerando o eixo local.

Figura 3 – Elemento de grelha no eixo global.



Fonte: Autor.

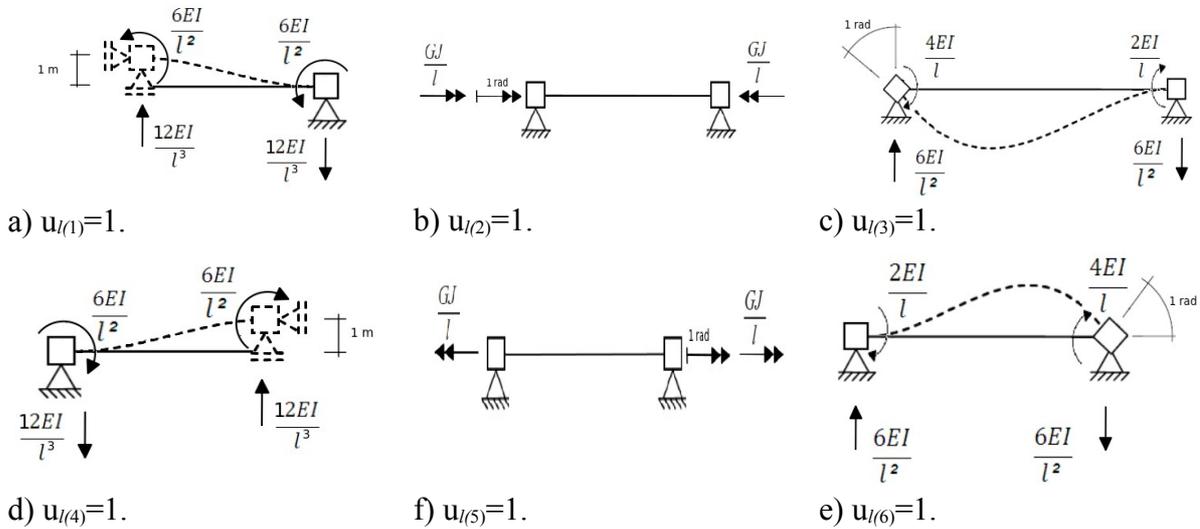
Figura 4 – Elemento de grelha no eixo local.



Fonte: Autor.

A matriz de rigidez é obtida considerando os coeficientes de rigidez gerados por um deslocamento unitário em cada grau de liberdade no sistema local, conforme é ilustrado na Figura 5. A equação (3) mostra a matriz de rigidez do elemento de grelha.

Figura 5 – Coeficientes de rigidez de um elemento de grelha.

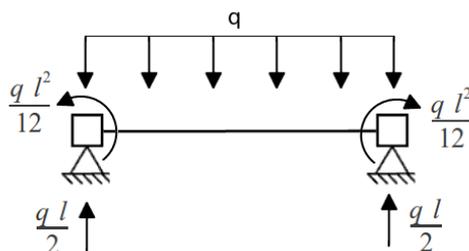


Fonte: Autor.

$$[K] = \begin{bmatrix} \frac{12EI}{l^3} & 0 & \frac{-6EI}{l^2} & \frac{-12EI}{l^3} & 0 & \frac{-6EI}{l^2} \\ 0 & \frac{GJ}{l} & 0 & 0 & \frac{-GJ}{l} & 0 \\ \frac{-6EI}{l^2} & 0 & \frac{4EI}{l} & \frac{6EI}{l^2} & 0 & \frac{2EI}{l} \\ \frac{-12EI}{l^3} & 0 & \frac{6EI}{l^2} & \frac{12EI}{l^3} & 0 & \frac{6EI}{l^2} \\ 0 & \frac{-GJ}{l} & 0 & 0 & \frac{GJ}{l} & 0 \\ \frac{-6EI}{l^2} & 0 & \frac{2EI}{l} & \frac{6EI}{l^2} & 0 & \frac{4EI}{l} \end{bmatrix} \quad (3)$$

Os coeficientes de engastamento perfeito de um carregamento distribuído uniformemente sobre o elemento de grelha, está ilustrado na Figura 6.

Figura 6 – Esforços de engastamento perfeito.



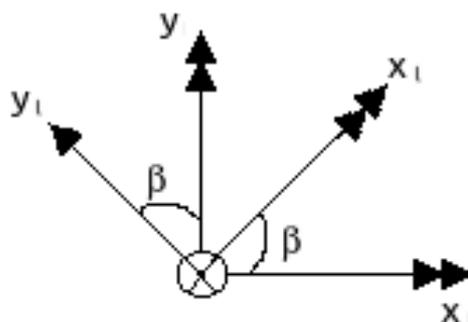
Fonte: Autor.

O vetor gerado pelos coeficientes está descrito na equação (4).

$$\{f_i\} = \begin{Bmatrix} \frac{q \cdot l}{2} \\ 0 \\ -\frac{q \cdot l^2}{12} \\ \frac{q \cdot l}{2} \\ 0 \\ \frac{q \cdot l^2}{12} \end{Bmatrix} \quad (4)$$

O vetor  $\{f_i\}$  e matriz  $[K_1]$  são obtidos no sistema local do elemento. Para obter as expressões globais, é necessário a utilização da matriz de transformação do elemento de grelha. Por ser uma estrutura plana, a grelha apresenta eixo  $z$  local coincidente ao eixo  $z$  global, de forma que as ações e deslocamentos dos elementos sofram apenas transformação por uma rotação de um dado ângulo  $\beta$ , conforme está ilustrado na Figura 7.

Figura 7 – Ângulo entre os eixos globais e locais de um elemento.



Fonte: Autor.

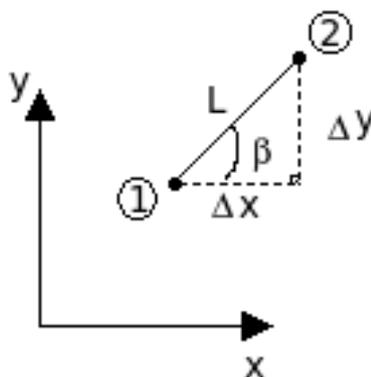
A relação entre as coordenadas pode ser obtida pela Equação (5), estas equações

são utilizadas para relacionar os deslocamentos locais e globais, definindo a matriz de transformação do elemento, que é utilizada para realizar mudança de rotação de vetores de um sistema de eixos para outro.

$$\begin{aligned} z_l &= z \\ x_l &= x \cos \beta + y \sin \beta \\ y_l &= y \cos \beta - x \sin \beta \end{aligned} \quad (5)$$

A Figura 8 mostra um elemento de grelha no sistema global, que possui um determinado ângulo global  $\beta$ , o cosseno e o seno do ângulo podem ser obtidos em função das coordenadas nodais do elemento, de acordo com a Equação (6).

Figura 8 – Elemento de grelha no sistema global.



Fonte: Autor.

$$\begin{aligned} L &= \sqrt{\Delta X^2 + \Delta Y^2} \\ \cos \beta &= \frac{\Delta X}{L} \\ \sin \beta &= \frac{\Delta Y}{L} \end{aligned} \quad (6)$$

A matriz de transformação do elemento de grelha (MCGUIRE, 2000) é obtida pela expressão:

$$[T]=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\beta) & \sin(\beta) & 0 & 0 & 0 \\ 0 & -\sin(\beta) & \cos(\beta) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(\beta) & \sin(\beta) \\ 0 & 0 & 0 & 0 & -\sin(\beta) & \cos(\beta) \end{bmatrix} \quad (7)$$

A matriz de rigidez global do elemento e o vetor do coeficiente de engastamento do carregamento distribuído podem ser obtidos, respectivamente, pela equação (8) e equação (9).

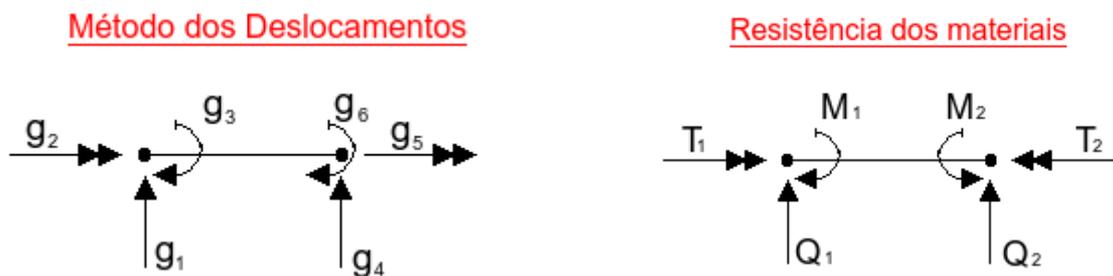
$$[K_g]=[T^t][K_l][T] \quad (8)$$

$$[F_g]=[T^t][F_l] \quad (9)$$

Onde  $[K_g]$  é a matriz de rigidez global do elemento,  $[T^t]$  é a transposta da matriz de transformação,  $[K_l]$  é a matriz de rigidez local do elemento,  $[T]$  é a matriz de transformação do elemento,  $\{F_g\}$  é o vetor dos coeficientes de engastamento do carregamento uniformemente distribuído global e  $\{F_l\}$  é o vetor dos coeficientes de engastamento do carregamento uniformemente distribuído local.

Os esforços podem ser escritos na convenção de sinais do Método dos Deslocamentos ou na convenção de sinais da Resistência dos Materiais, conforme ilustrado na Figura 9. A Equação (10) é utilizada para obter os esforços na convenção da Resistência dos Materiais.

Figura 9 – Convenções de sinais dos esforços locais dos elementos.



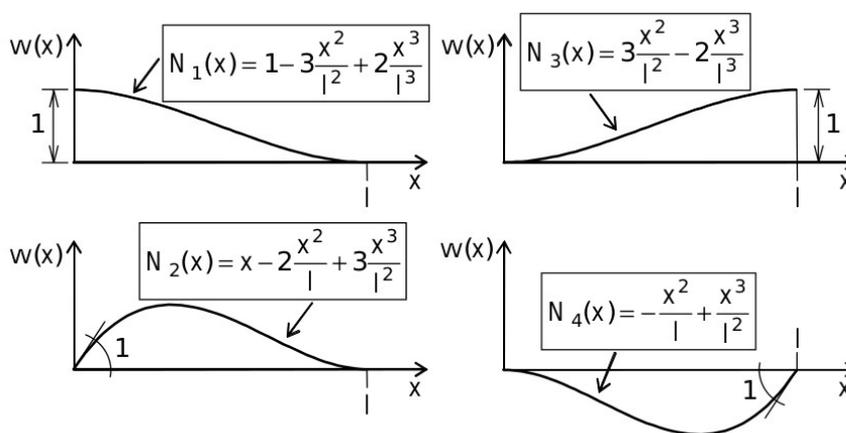
Fonte: Autor.

$$\begin{aligned}
 Q_1 &= g_1 \\
 T_1 &= -g_2 \\
 M_1 &= g_3 \\
 Q_2 &= -g_4 \\
 T_2 &= g_5 \\
 M_2 &= -g_6
 \end{aligned}
 \tag{10}$$

A deformada do elemento de grelha no sistema local pode ser obtida com a soma das funções de forma de cada deslocabilidade do elemento (MARTHA, 2010). A Figura 10 mostra as funções de forma para cada deslocabilidade unitária, sendo que a função de forma final pode ser obtida pela soma dos produtos das funções de forma pelas suas respectivas deslocabilidades, conforme está representado na equação (16).

$$w(x) = N_1(x) \cdot w_1 + N_2(x) \cdot r_{y1} + N_3(x) \cdot w_2 + N_4(x) \cdot r_{y2} \tag{11}$$

Figura 10 – Funções de forma de cada deslocabilidade do elemento de grelha.



Fonte: Autor.

### 2.3. Implementação computacional do método de rigidez direta

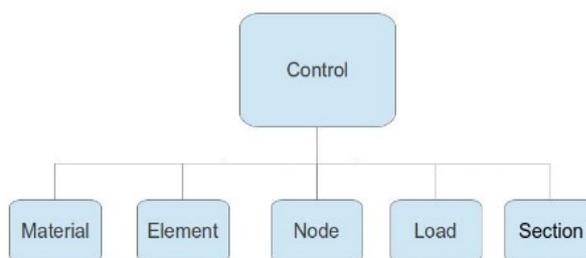
O método da rigidez direta foi implementado no programa de análise estrutural pelo Método dos Elementos Finitos, FAST. Este programa foi desenvolvido no laboratório de mecânica computacional da Universidade Federal do Ceará, sob a coordenação do professor Evandro Parente Junior. Foi escrito em linguagem de programação C/C++ e utiliza o conceito

de programação orientada a objeto, que facilita a expansão do programa.

A Figura 11 mostra a estrutura existente no programa, que possui uma classe central *cControl*, que utiliza funções e recursos das demais classes para solucionar o problema de análise estrutural, montar as matrizes globais do problema e desenvolver tarefas de leitura e escrita de arquivos. Esta classe central realiza procedimentos descritos no item 2.1, como montagem da matriz global e solução da Equação (4).

Os vetores e matrizes locais são implementados nas classes *cElement* e *cLoad*, onde também são implementadas funções de leitura e escrita de cada elemento, de modo que no final da execução do programa a classe *cControl* possa imprimir os resultados dos esforços em cada elemento.

Figura 11 – Estrutura do programa FAST.



Fonte: Autor.

O programa realiza a leitura de um arquivo de entrada contendo as informações do problema, este arquivo de texto possui uma série de rótulos e argumentos, que são interpretados pelo algoritmo de leitura do programa e armazenados. A Figura 12 mostra um exemplo de arquivo de entrada para um problema de uma grelha.

Figura 12 – Exemplo de problema escrito em um arquivo de entrada.

```
%HEADER
Output file generated by FAST.
Input file generated by GridCalc.
```

```
%NODE.COORD
```

```
3
1 0 0 0
2 0 -4 0
3 6 -4 0
```

```
%NODE.SUPPORT
```

```
2
1 0 0 1 1 1 0
3 0 0 1 1 1 0
```

$EI=5.10^5 \text{ kN.m}^2$   
 $GJ=4.10^5 \text{ kN.m}^2$

```
%MATERIAL.ISOTROPIC
```

```
1
1 500000 0
```

```
%SECTION.PROPERTY
```

```
1
1 0 0 0 1.6 1 0 0 0 0
```

```
%ELEMENT.GRID
```

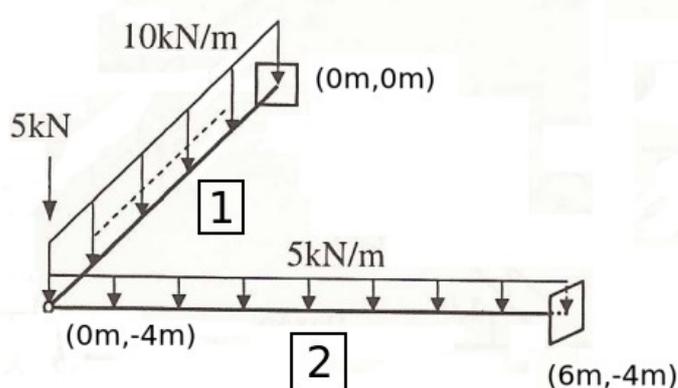
```
2
1 1 1 2 1
2 1 1 3 2
```

```
%LOAD.CASE.NODAL.FORCE
```

```
1
2 0 0 -5 0 0 0
```

```
%LOAD.CASE.GRID.UNIFORM
```

```
2
1 -10
2 -5
```

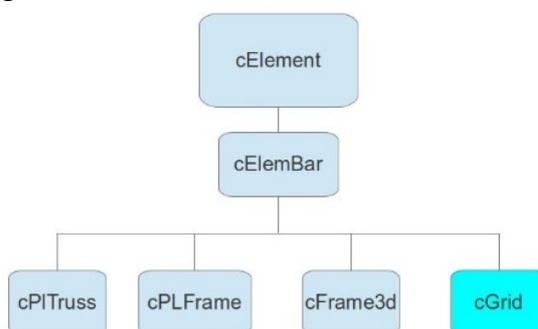


Fonte: Autor.

## 2.4. Implementação computacional do elemento de grelha

A utilização de elementos de grelha foi introduzida no analisador FAST pela criação de duas classes novas, a *cGrid* e a *cGridUnifLoad*. A Figura 13 mostra a estrutura de classes utilizadas pelo programa FAST para realizar procedimentos com elementos, onde a classe *cGrid* foi criada herdando funções da classe base *cElemBar*.

Figura 13 – Estrutura de classes dos elementos.



Fonte: Autor.

A Tabela 1 mostra as principais funções pertencentes a classe cGrid. A maior parte delas são funções virtuais, elas são herdadas da classe base cElemBar e são reimplementadas para cada elemento criado, isto é outro aspecto da programação orientada ao objeto.

Tabela 1 – Funções utilizadas pela classe cGrid.

Nome da função	Descrição
<i>GetElastMat()</i>	Calcula a matriz de rigidez local do elemento.
<i>GetTrMat()</i>	Calcula a matriz de transformação do elemento.
<i>StiffMat()</i>	Calcula a matriz global do elemento.
<i>EqvForce()</i>	Carrega o vetor dos esforços de engastamento perfeito do elemento.
<i>NodalStress()</i>	Calcula os esforços internos nodais para impressão.
<i>IntForce()</i>	Calcula os esforços internos do elemento.

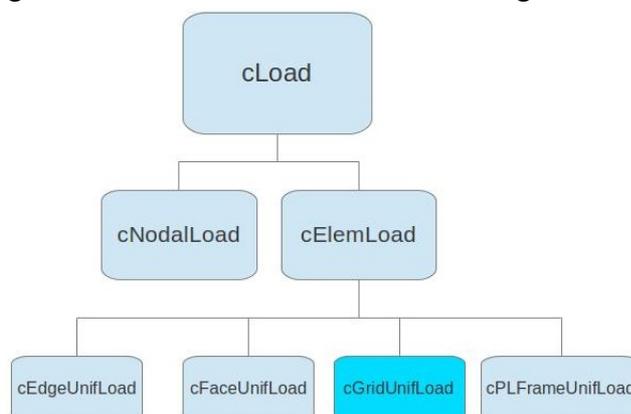
Fonte: Autor.

A Figura 14 mostra a estrutura de classes utilizadas pelo programa FAST para realizar procedimentos com os carregamentos, a classe cGridUnifLoad foi criada herdando funções da classe base cElemLoad. É importante observar que só foi desenvolvido um tipo de carregamento sobre o elemento, o distribuído uniformemente, que é o mais utilizado para análise de lajes pelo método da analogia de grelha.

A classe *cGridUnifLoad* possui duas funções importantes, a função *Read()* responsável por leitura dos carregamentos no arquivo de entrada descrito na Figura 12 e a função *ExtForce()* responsável pela montagem dos vetores dos coeficientes de engastamento

perfeito local e global, utilizando a Equação (4) e Equação (7).

Figura 14 – Estrutura de classes dos carregamentos.



Fonte: Autor.

A implementação de outros tipos de carregamento pode ser feita pelo mesmo procedimento utilizado para carga uniformemente distribuída, a única diferença ocorrerá no valor dos esforços de engastamento perfeito gerados e no número de parâmetros de entrada que são necessários no arquivo de entrada. No caso da carga uniformemente distribuída é necessário o número de identificação do elemento e o valor da carga (ver Figura 12). Para uma carga trapezoidal seria necessário o valor da carga nas duas extremidades.

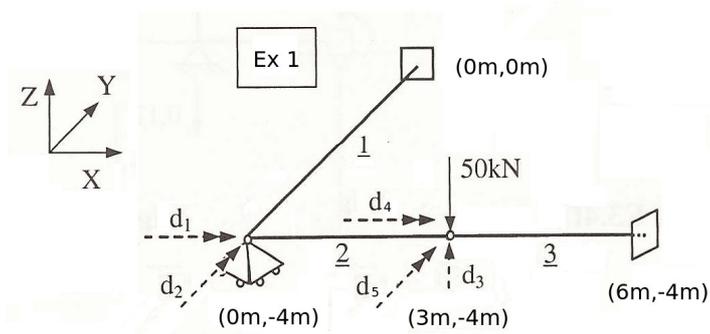
## 2.5. Exemplos para verificação

A verificação do analisador FAST foi realizada com o uso de dois exemplos retirados da literatura (SORIANO,2006), onde foram solucionados utilizando o Método dos Deslocamentos. Nos itens 2.5.1 e 2.5.2, os exemplos são solucionados utilizando o Método da Rigidez Direta.

### 2.5.1 Exemplo 1

A Figura 15 ilustra o primeiro exemplo utilizado para validação do analisador estrutural FAST. O problema é composto de três elementos de grelha, onde cada barra possui propriedades  $EI=5 \times 10^5 \text{ kNxm}^2$  e  $GJ=4 \times 10^5 \text{ kNxm}^2$ .

Figura 15 – Exemplo 1 de validação do programa FAST.



Fonte: Autor.

a) Para cada elemento, são calculados a matriz de rigidez local e global, matriz de transformação e vetor de cargas local e global:

$$(Elemento 1) \quad [K_l] = \begin{vmatrix} 93750 & 0 & -187500 & -93750 & 0 & -187500 \\ 0 & 100000 & 0 & 0 & -100000 & 0 \\ -187500 & 0 & 500000 & 750000 & 0 & 15625 \\ -93750 & 0 & 187500 & 93750 & 0 & 187500 \\ 0 & -100000 & 0 & 0 & 100000 & 0 \\ -187500 & 0 & 15625 & 187500 & 0 & 500000 \end{vmatrix}$$

$$[T] = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} \quad [T^t] = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{vmatrix}$$

$$[K_g] = [T^t][K_l][T] = \begin{vmatrix} 93750 & -187500 & 0 & -93750 & -187500 & 0 \\ -187500 & 500000 & 0 & 750000 & 250000 & 0 \\ 0 & 0 & 100000 & 0 & 0 & -100000 \\ -93750 & 187500 & 0 & 93750 & 187500 & 0 \\ -187500 & 15625 & 0 & 187500 & 500000 & 0 \\ 0 & 0 & -100000 & 0 & 0 & 100000 \end{vmatrix}$$

$$(Elemento 2) \quad [K_l] = \begin{vmatrix} \frac{2000000}{9} & 0 & \frac{-1000000}{3} & \frac{-2000000}{9} & 0 & \frac{-1000000}{3} \\ 0 & \frac{400000}{3} & 0 & 0 & \frac{-400000}{3} & 0 \\ \frac{-1000000}{3} & 0 & \frac{2000000}{3} & \frac{1000000}{3} & 0 & \frac{1000000}{3} \\ \frac{-2000000}{9} & 0 & \frac{1000000}{3} & \frac{2000000}{9} & 0 & \frac{1000000}{3} \\ 0 & \frac{-400000}{3} & 0 & 0 & \frac{400000}{3} & 0 \\ \frac{-1000000}{3} & 0 & \frac{1000000}{3} & \frac{1000000}{3} & 0 & \frac{2000000}{3} \end{vmatrix}$$

$$[T]=[T^t]=\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$[K_g]=[T^t][K_l][T]=[K_l]$$

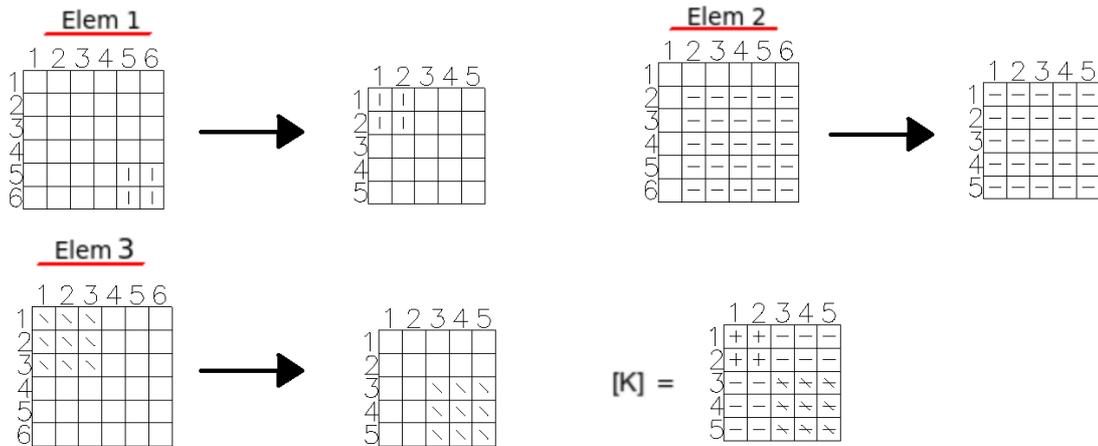
$$(Elemento3) \quad [K_l]=\begin{vmatrix} \frac{2000000}{9} & 0 & \frac{-1000000}{3} & \frac{-2000000}{9} & 0 & \frac{-1000000}{3} \\ 0 & \frac{400000}{3} & 0 & 0 & \frac{-400000}{3} & 0 \\ \frac{-1000000}{3} & 0 & \frac{2000000}{3} & \frac{1000000}{3} & 0 & \frac{1000000}{3} \\ \frac{-2000000}{9} & 0 & \frac{1000000}{3} & \frac{2000000}{9} & 0 & \frac{1000000}{3} \\ 0 & \frac{-400000}{3} & 0 & 0 & \frac{400000}{3} & 0 \\ \frac{-1000000}{3} & 0 & \frac{1000000}{3} & \frac{1000000}{3} & 0 & \frac{2000000}{3} \end{vmatrix}$$

$$[T]=[T^t]=\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$[K_g]=[T^t][K_l][T]=[K_l]$$

- b) A Matriz de rigidez é obtida pelo processo de espalhamento das matrizes globais dos elementos, conforme está ilustrado na Figura 16. Observe que neste exemplo não existem carregamentos sobre os elementos, apenas sobre o terceiro nó, o que faz com que nenhum elemento contribua para o vetor  $\{f\}$ .

Figura 16 – Processo de espalhamento do exemplo 1.



Fonte: Autor.

$$[K_g] = \begin{vmatrix} 633333 & 0 & 0 & -133333 & 0 \\ 0 & 766667 & 333333 & 0 & 333333 \\ 0 & 333333 & 444444 & 0 & 0 \\ -133333 & 0 & 0 & 2666667 & 0 \\ 0 & 333333 & 0 & 0 & 1333333 \end{vmatrix}$$

$$\{f\} = \begin{vmatrix} 0 \\ 0 \\ -50 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

$$[K]\{u\} = \{f\} \rightarrow \{u\} = \begin{vmatrix} 0 \\ 8.65 \times 10^{-5} \\ -1.77 \times 10^{-4} \\ 0 \\ -2.16 \times 10^{-5} \end{vmatrix}$$

c) Uma vez calculado os deslocamentos dos graus de liberdade, são calculados os esforços para cada elemento. Observe que estes esforços obtidos não estão considerando a convenção de sinais da resistência dos materiais, a qual é considerada após a obtenção dos resultados.

$$(Elemento 1) \quad \{u_g\} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 8.65 \times 10^{-5} \end{vmatrix}$$

$$\{u_i\} = [T] \{u_g\} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 8.65 \times 10^{-5} \\ 0 \end{vmatrix}$$

$$\{g_i\} = [K_i] \{u_i\} = \begin{vmatrix} 0 \\ -8.6538 \text{ kN.m} \\ 0 \\ 0 \\ 8.6538 \text{ kN.m} \\ 0 \end{vmatrix} \rightarrow \text{Para R.M } \{R_i\} = \begin{vmatrix} 0 \\ -8.6538 \text{ kN.m} \\ 0 \\ 0 \\ -8.6538 \text{ kN.m} \\ 0 \end{vmatrix}$$

$$(\text{Elemento 2}) \quad \{u_i\} = \{u_g\} = \begin{vmatrix} 0 \\ 0 \\ 8.65 \times 10^{-5} \\ -1.77 \times 10^{-4} \\ 0 \\ -2.16 \times 10^{-5} \end{vmatrix}$$

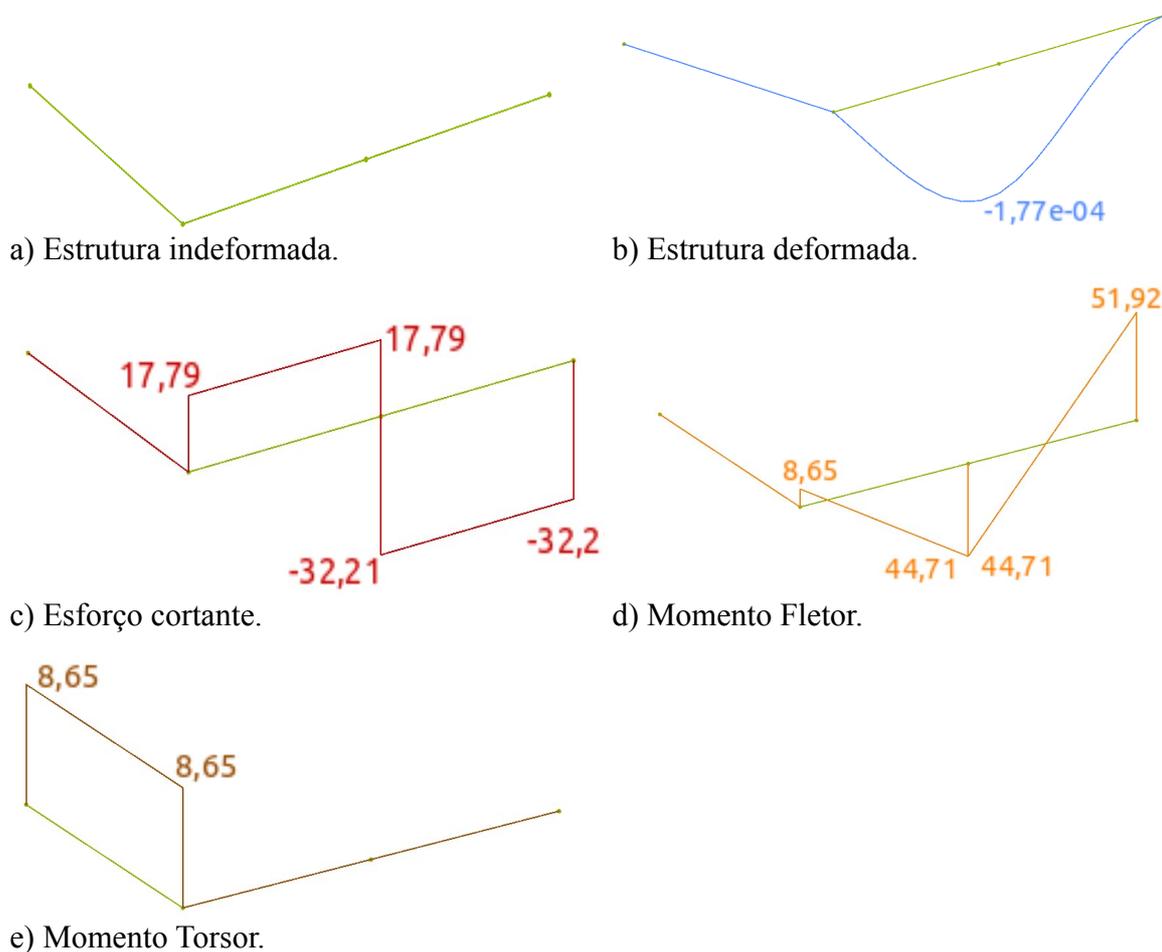
$$\{g_i\} = [K_i] \{u_i\} = \begin{vmatrix} 17.79 \text{ kN} \\ 0 \\ -8.65 \text{ kN.m} \\ -17.79 \text{ kN} \\ 0 \\ -44.71 \text{ kN.m} \end{vmatrix} \rightarrow \text{Para R.M } \{R_i\} = \begin{vmatrix} 17.79 \text{ kN} \\ 0 \\ -8.65 \text{ kN.m} \\ 17.79 \text{ kN} \\ 0 \\ 44.71 \text{ kN.m} \end{vmatrix}$$

$$(\text{Elemento 3}) \quad \{u_i\} = \{u_g\} = \begin{vmatrix} -1.77 \times 10^{-4} \\ 0 \\ -2.16 \times 10^{-5} \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

$$\{g_i\} = [K_i] \{u_i\} = \begin{vmatrix} -32.21 \text{ kN} \\ 0 \\ 44.71 \text{ kN.m} \\ 32.21 \text{ kN} \\ 0 \\ 51.92 \text{ kN.m} \end{vmatrix} \rightarrow \text{Para R.M } \{R_i\} = \begin{vmatrix} -32.21 \text{ kN} \\ 0 \\ 44.71 \text{ kN.m} \\ -32.21 \text{ kN} \\ 0 \\ -51.92 \text{ kN.m} \end{vmatrix}$$

d) Os gráficos dos esforços podem ser traçados em função dos valores das extremidades de cada elemento. A Figura 17 mostra os resultados obtidos pelo programa *GridCalc*.

Figura 17 – Resultados visualizados pelo programa *GridCalc*.



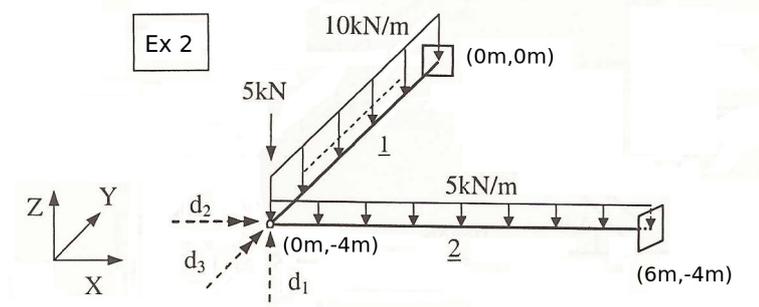
Fonte: Autor.

Os resultados obtidos foram iguais aos obtidos por Soriano (2006), Exemplo 3.5, página 156.

### 2.5.2 Exemplo 2

A Figura 18 ilustra o segundo exemplo utilizado para validação do analisador estrutural FAST. O problema é composto por dois elementos de grelha, onde cada um possui propriedades  $EI=5 \times 10^5 \text{ kNxm}^2$  e  $GJ=4 \times 10^5 \text{ kNxm}^2$ .

Figura 18 – Exemplo 2 de validação do programa FAST.



Fonte: Autor.

a) Para cada elemento, são calculados a matriz de rigidez local e global, matriz de transformação e vetor de cargas local e global:

$$(\text{Elemento 1}) \quad [K_l] = \begin{bmatrix} 93750 & 0 & -187500 & -93750 & 0 & -187500 \\ 0 & 100000 & 0 & 0 & -100000 & 0 \\ -187500 & 0 & 500000 & 750000 & 0 & 15625 \\ -93750 & 0 & 187500 & 93750 & 0 & 187500 \\ 0 & -100000 & 0 & 0 & 100000 & 0 \\ -187500 & 0 & 15625 & 187500 & 0 & 500000 \end{bmatrix}$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad [T^t] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$[K_g] = [T^t][K_l][T] = \begin{bmatrix} 93750 & -187500 & 0 & -93750 & -187500 & 0 \\ -187500 & 500000 & 0 & 750000 & 250000 & 0 \\ 0 & 0 & 100000 & 0 & 0 & -100000 \\ -93750 & 187500 & 0 & 93750 & 187500 & 0 \\ -187500 & 15625 & 0 & 187500 & 500000 & 0 \\ 0 & 0 & -100000 & 0 & 0 & 100000 \end{bmatrix}$$

$$\{f_l\} = \begin{bmatrix} 20 \\ 0 \\ -13.333333 \\ 20 \\ 0 \\ 13.333333 \end{bmatrix} ; \quad \{f_g\} = [T^t]\{f_l\} ; \quad \{f_g\} = \begin{bmatrix} 20 \\ -13.333333 \\ 0 \\ 20 \\ 13.333333 \\ 0 \end{bmatrix}$$

$$(Elemento2) \quad [K_i] = \begin{vmatrix} \frac{250000}{9} & 0 & \frac{-250000}{3} & \frac{-250000}{9} & 0 & \frac{-250000}{3} \\ 0 & \frac{200000}{3} & 0 & 0 & \frac{-200000}{3} & 0 \\ \frac{-250000}{3} & 0 & \frac{1000000}{3} & \frac{250000}{3} & 0 & \frac{500000}{3} \\ -250000 & 0 & 250000 & 250000 & 0 & 250000 \\ \frac{9}{9} & 0 & \frac{3}{3} & \frac{9}{9} & 0 & \frac{3}{3} \\ 0 & \frac{-200000}{3} & 0 & 0 & \frac{200000}{3} & 0 \\ \frac{-250000}{3} & 0 & \frac{500000}{3} & \frac{250000}{3} & 0 & \frac{1000000}{3} \end{vmatrix}$$

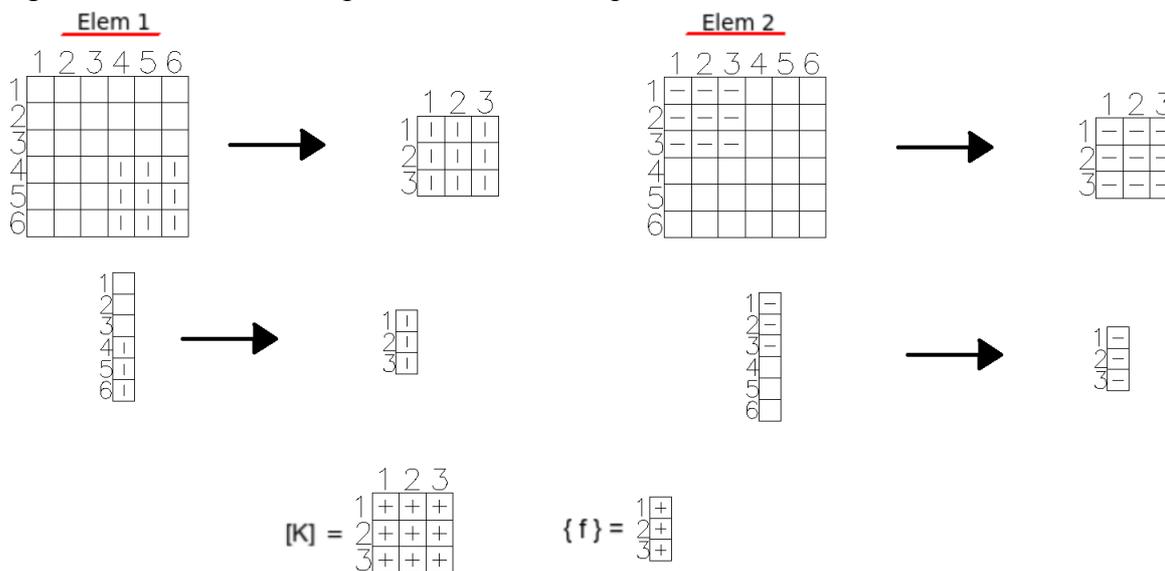
$$[T] = [T^t] = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$[K_g] = [T^t][K_i][T] = [K_i]$$

$$\{f_i\} = \begin{vmatrix} 15 \\ 0 \\ -15 \\ 15 \\ 0 \\ 15 \end{vmatrix} ; \quad \{f_g\} = [T^t]\{f_i\} = \{f_i\}$$

b) A Matriz de rigidez  $[K]$  e o vetor de cargas  $\{f\}$  são obtidos pelo processo de espalhamento das matrizes globais dos elementos, conforme está ilustrado na Figura 19.

Figura 19 – Processo de espalhamento do exemplo 2.



Fonte: Autor.

$$[K_g] = \begin{vmatrix} 121528 & 187500 & -833333 \\ 187500 & 566667 & 0 \\ -833333 & 0 & 433333 \end{vmatrix}$$

$$\{f\} = \begin{vmatrix} -5 \\ 0 \\ 0 \end{vmatrix} - \begin{vmatrix} 20 \\ 13.33333 \\ 0 \end{vmatrix} - \begin{vmatrix} 15 \\ 0 \\ -15 \end{vmatrix} = \begin{vmatrix} -40 \\ -13.33333 \\ 15 \end{vmatrix}$$

$$[K]\{u\} = \{f\} \rightarrow \{u\} = \begin{vmatrix} -7.52 \times 10^{-4} \\ 2.25 \times 10^{-4} \\ -1.10 \times 10^{-4} \end{vmatrix}$$

c) Uma vez calculado os deslocamentos dos graus de liberdade, são calculados os esforços para cada elemento. Neste exemplo, todos os elementos possuem carregamento distribuído, e os coeficientes de engastamento perfeito são somados para obtenção dos esforços nas extremidades das barras.

$$(Elemento 1) \quad \{u_g\} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ -7.52 \times 10^{-4} \\ 2.25 \times 10^{-4} \\ -1.10 \times 10^{-4} \end{vmatrix}$$

$$\{u_i\} = [T] \{u_g\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -7.52 \times 10^{-4} \\ 1.10 \times 10^{-4} \\ 2.25 \times 10^{-4} \end{Bmatrix}$$

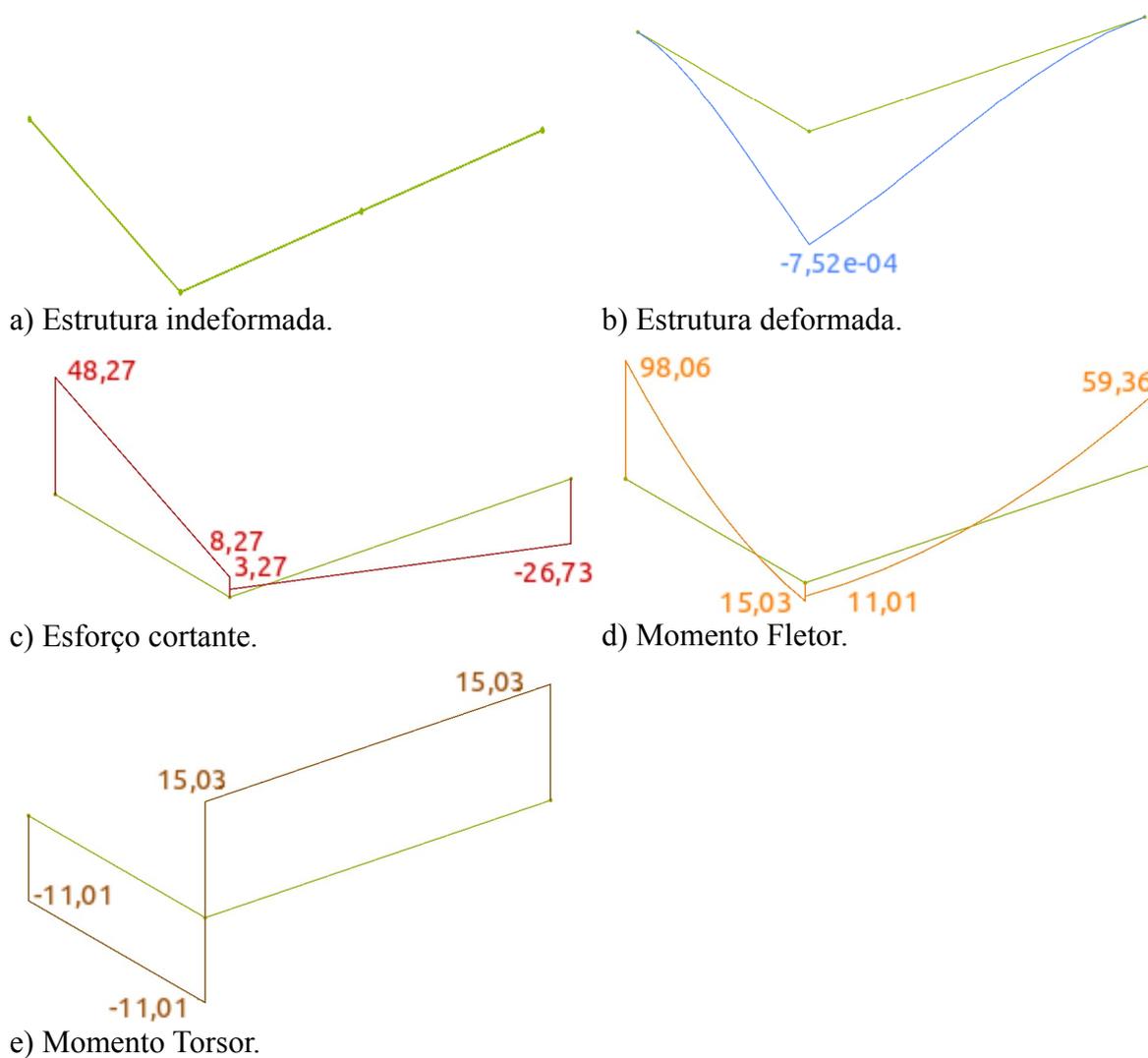
$$\{g_i\} = [K_i] \{u_i\} + \{f_i\} = \begin{Bmatrix} 28.27 \\ -11.01 \\ -84.73 \\ -28.27 \\ 11.01 \\ -28.36 \end{Bmatrix} + \begin{Bmatrix} 20 \\ 0 \\ -13.3333 \\ 20 \\ 0 \\ 13.3333 \end{Bmatrix} = \begin{Bmatrix} 48.27 \\ -11.01 \\ -98.06 \\ -8.27 \\ 11.01 \\ -15.03 \end{Bmatrix} \rightarrow \text{Para R.M } \{R_i\} = \begin{Bmatrix} 48.27 \text{ kN} \\ 11.01 \text{ kN.m} \\ -98.06 \text{ kN.m} \\ 8.27 \text{ kN} \\ 11.01 \text{ kN.m} \\ 15.03 \text{ kN.m} \end{Bmatrix}$$

$$(\text{Elemento 2}) \quad \{u_i\} = \{u_g\} = \begin{Bmatrix} -7.52 \times 10^{-4} \\ 2.25 \times 10^{-4} \\ -1.10 \times 10^{-4} \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

$$\{g_i\} = [K_i] \{u_i\} + \{f_i\} = \begin{Bmatrix} -11.73 \\ 15.03 \\ 26.01 \\ 11.73 \\ -15.03 \\ 44.36 \end{Bmatrix} + \begin{Bmatrix} 15 \\ 0 \\ -15 \\ 15 \\ 0 \\ 15 \end{Bmatrix} = \begin{Bmatrix} 3.27 \\ 15.03 \\ 11.01 \\ 26.73 \\ -15.03 \\ 59.36 \end{Bmatrix} \rightarrow \text{Para R.M } \{R_i\} = \begin{Bmatrix} 3.27 \text{ kN} \\ -15.03 \text{ kN.m} \\ 11.01 \text{ kN.m} \\ -26.73 \text{ kN} \\ -15.03 \text{ kN.m} \\ -59.36 \text{ kN.m} \end{Bmatrix}$$

d) Os gráficos dos esforços podem ser traçados em função dos valores das extremidades de cada elemento. A Figura 20 mostra os resultados obtidos pelo programa GridCalc.

Figura 20 – Resultados visualizados pelo programa GridCalc.



Fonte: Autor.

Os resultados obtidos foram iguais aos obtidos por Soriano (2006), Exemplo 3.6, página 158.

### 3. MODELAGEM E VISUALIZAÇÃO DE GRELHAS

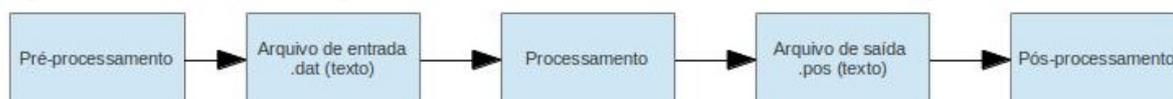
A modelagem e visualização de modelos computacionais de engenharia é uma etapa de extrema importância, pois possibilita conceber de forma clara e correta os problemas físicos analisados, além de facilitar a análise e interpretação dos resultados obtidos nas análises.

Neste capítulo é realizada uma revisão dos conceitos utilizados para implementação computacional do pré-processador e pós-processador concebidos neste trabalho. O item 3.1 aborda a estrutura do programa, apresentando as classes e as bibliotecas utilizadas para implementação. O item 3.2 trata especificamente das implementações utilizadas no pré-processador, enquanto que o item 3.3 trata das implementações utilizadas no pós-processador.

#### 3.1. Análise estrutural computacional

A Figura 21 mostra o procedimento comumente utilizado para análise estrutural computacional, onde inicialmente é gerado um arquivo de entrada contendo geometria (elementos e nós) e atributos (seção, material, apoios e cargas) do problema, etapa conhecida como pré-processamento. O arquivo de entrada é lido e interpretado por um analisador, onde através de métodos de análise estrutural, gera um arquivo de saída com os resultados das análises realizadas, etapa conhecida como processamento. Finalmente o arquivo de saída é lido para poder ser visualizado de maneira adequada, dependendo do problema em questão.

Figura 21 – Procedimento para realização de uma análise estrutural computacional.



Fonte: Autor.

No presente trabalho, foi desenvolvido um programa para realizar o pré-processamento e pós-processamento de problemas de análise estrutural de grelhas, o programa GridCalc. O analisador utilizado para o processamento foi o programa FAST, ao qual foram feitas adições para processamento de problemas contendo elementos de grelha. O programa GridCalc é composto por duas partes, a Interface gráfica, onde são implementadas

funções de interação com o usuário, e a estrutura de dados, onde são armazenados os dados criados pelo usuário.

A implementação da interface gráfica foi realizada utilizando o Qt 4.8, que é um ambiente integrado para desenvolvimento de software (IDE), ilustrado na Figura 22. O Qt possui muitas vantagens, primeiro destaca-se a facilidade que se tem em desenvolver interface gráficas, o programa possui excelente documentação com vários exemplos. Além disso um dos componentes do Qt, o Designer (ver Figura 23), proporciona uma maneira extremamente fácil de desenhar a interface, de maneira que em poucas horas de uso um desenvolvedor com pouca experiência em linguagem de programação c/c++ consegue compilar e executar um programa complexo de interface gráfica.

Figura 22 – Editor de código Qt.

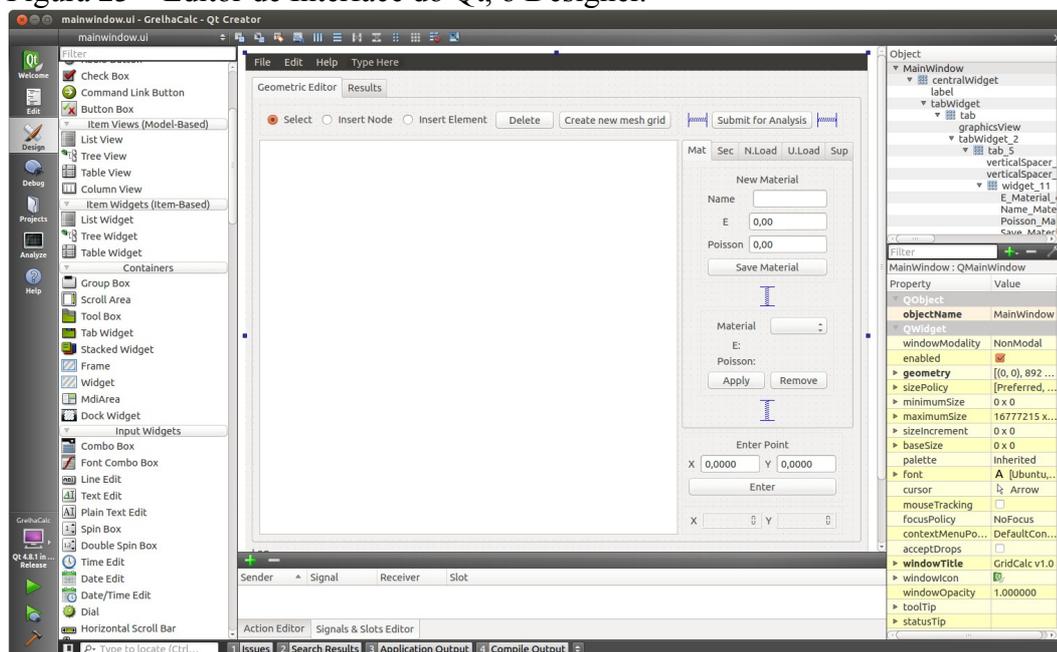
```

1  #include "core.h"
2  #include "element.h"
3  #include "node.h"
4  #include <stdlib.h>
5  #include <stdio.h>
6  #include <cmath>
7  #include <iostream>
8  #include <fstream>
9  |
10 |
11 | Core::Core()
12 | {
13 |     node_selected=false;
14 |     element_selected=false;
15 |     tolerance=0.05;
16 | }
17 | //Create a new node in node list and return this pointer.
18 | Node *Core::Create_Node (double x, double y)
19 | {
20 |
21 |     std::list<Node*>::iterator it_node;
22 |     bool invalid_node=false;
23 |     Node *node;
24 |     for (it_node=Node::Node_List.begin(); it_node!=Node::Node_List.end(); it_node++)
25 |     {
26 |         if (((*it_node)->X()-tolerance)<=x && ((*it_node)->X()+tolerance)>=x && ((*it_node)->Y()-tolerance)<=y && ((*it_node)->Y()+tolerance)>=y)
27 |         {
28 |             invalid_node=true;
29 |             node=(*it_node);
30 |         }
31 |     }
32 |     if (invalid_node==false)
33 |     {
34 |         node=new Node(x,y);
35 |         Node::Node_List.push_back(node);
36 |     }
37 |     return (node);
38 | }
39 |
40 |
41 | Element *Core::Create_Element(Node *a0, Node *a1)
42 | {
43 |     std::list<Element*>::iterator it_element;
44 |     Element *new_element;
45 |     bool invalid_element=false;
46 |     for (it_element=Element::Element_List.begin(); it_element!=Element::Element_List.end(); it_element++)
47 |     {
48 |         if (((*it_element)->node1()->X())==a0->X() && (*it_element)->node1()->Y()==a0->Y() && ((*it_element)->node2()->X())==a1->X() && (*it_element)->node2()->Y()==a1->Y())
49 |         {
50 |             invalid_element=true;
51 |         }
52 |         if (invalid_element==false)
53 |         {
54 |             new_element=new Element(a0,a1);
55 |         }
56 |     }
57 |     return (new_element);
58 | }
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100|

```

Fonte: Autor.

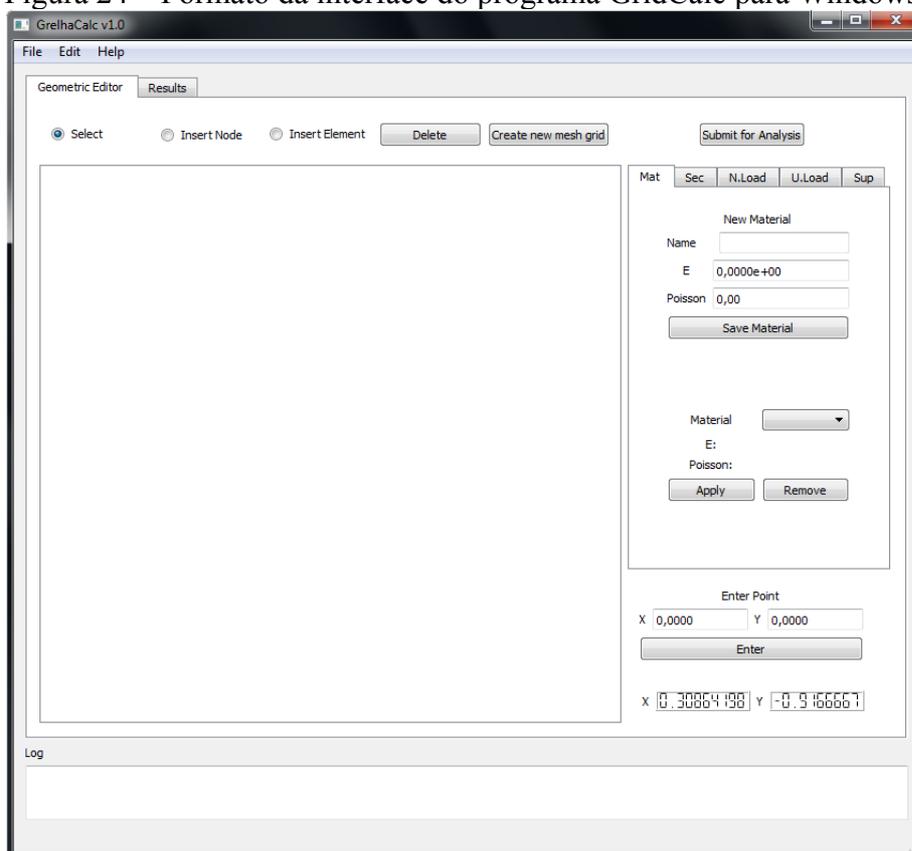
Figura 23 – Editor de Interface do Qt, o Designer.



Fonte: Autor.

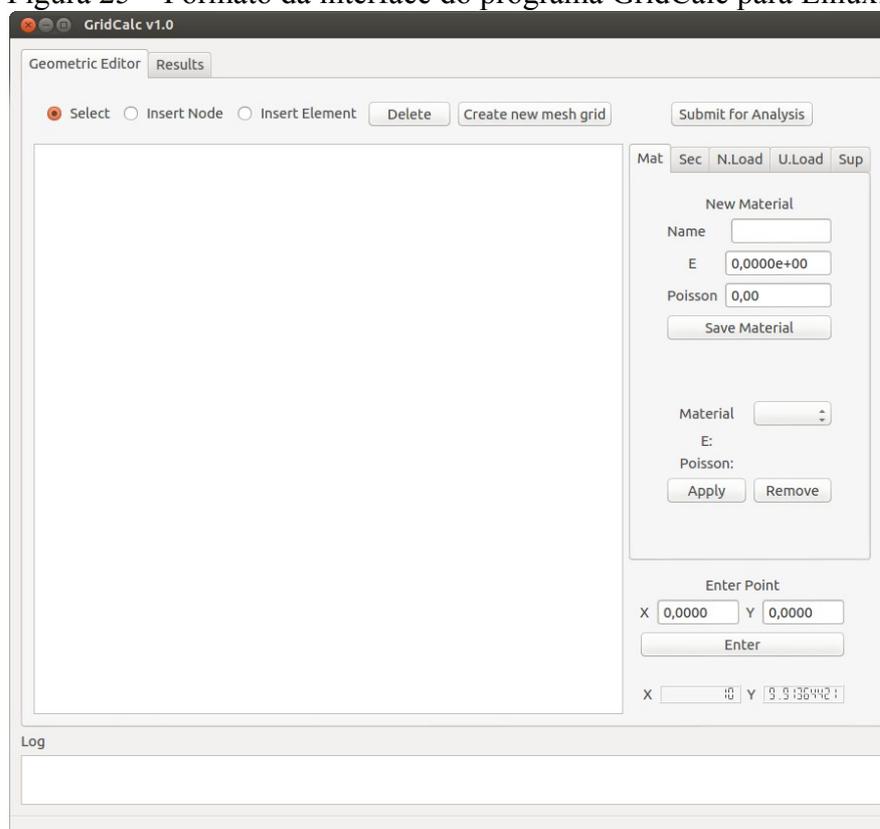
Outra vantagem do Qt é que ele é um sistema multiplataforma, podendo compilar programas em vários sistemas operacionais, entre eles destacam-se o Microsoft Windows, o Mac OS e o GNU/Linux e estes são sistemas operacionais que o programa desenvolvido neste trabalho busca atuar. Cada sistema operacional possui um visual nativo dos objetos gráficos do Qt, como ilustram a Figura 24 e Figura 25, mostrando o formato da interface do programa desenvolvido para Windows e GNU/Linux.

Figura 24 – Formato da interface do programa GridCalc para Windows.



Fonte: Autor.

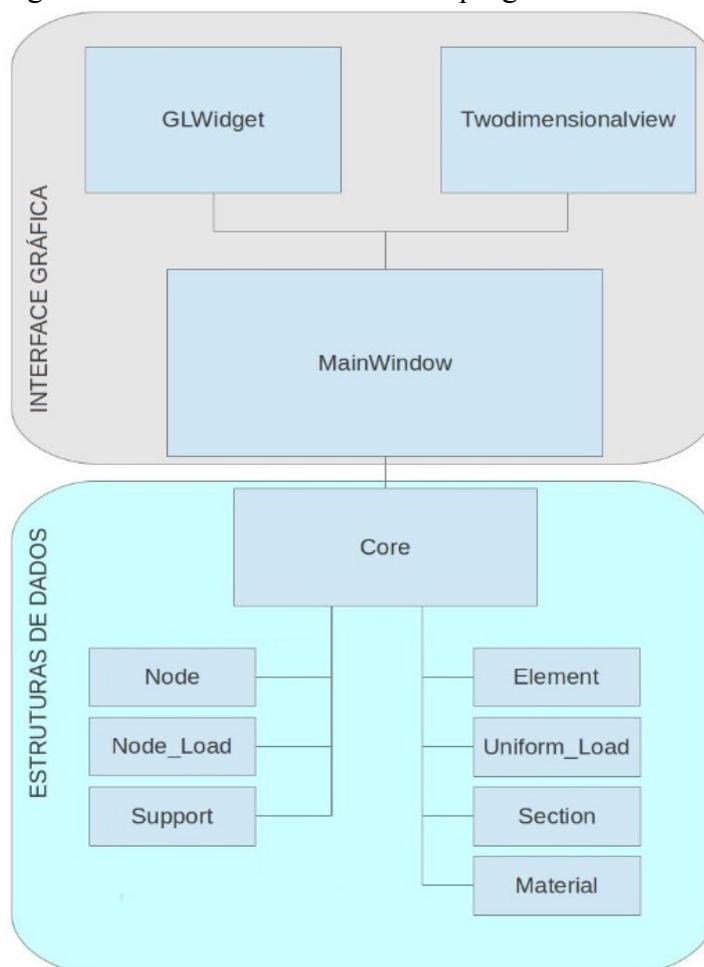
Figura 25 – Formato da interface do programa GridCalc para Linux.



Fonte: Autor.

Apesar do Qt ser uma excelente ferramenta de trabalho, uma boa implementação não deve utilizar dentro da estrutura de dados, as bibliotecas pertencentes a IDE utilizada na interface gráfica, isto garante uma fácil adaptação do programa a uma nova interface gráfica, se for necessário no futuro. A Figura 26 mostra a estrutura de classes utilizada no programa GridCalc, observando-se que nenhuma função específica do Qt é utilizada nas classes pertencentes a estrutura de dados.

Figura 26 – Estrutura de classes do programa GridCalc.



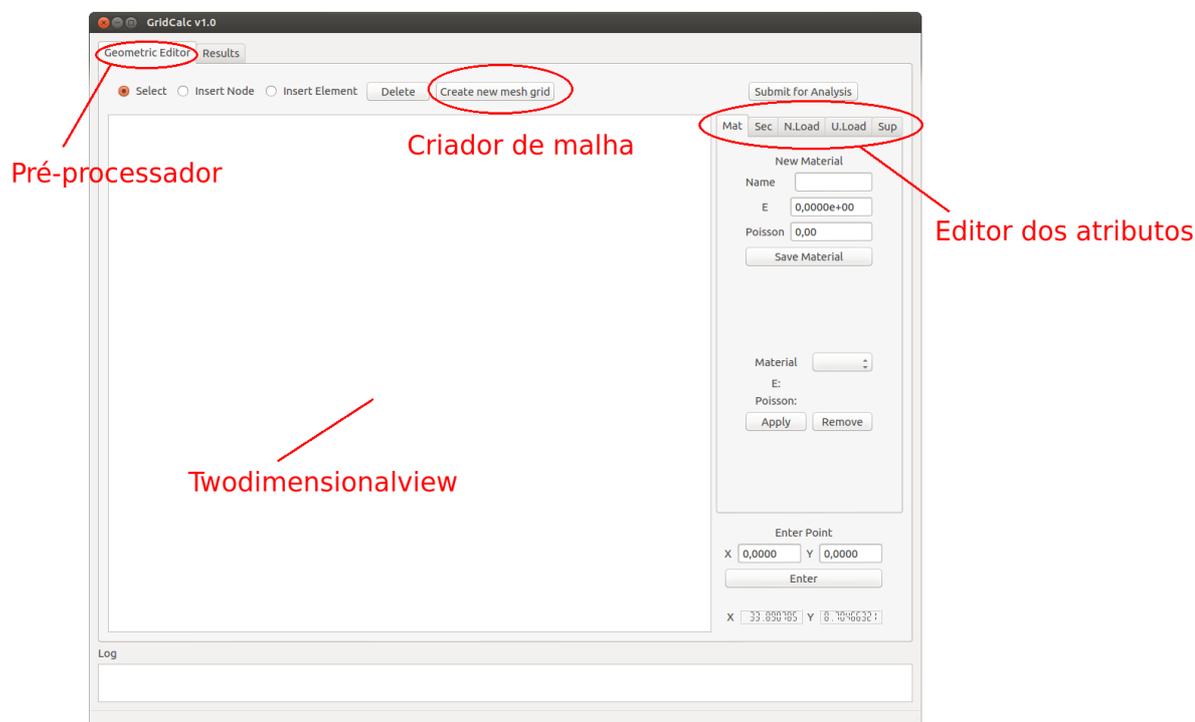
Fonte: Autor.

A classe Twodimensionalview foi utilizada para implementação da visualização contida no pré-processador, que herda as funções de uma classe base do Qt, a QGraphicsview. A classe GLWidget foi utilizada na implementação da visualização do pós-processador, utilizando a interface de programação de aplicativos OpenGL.

### 3.2. Implementação do pré-processador

A Figura 27 mostra a tela do programa desenvolvido GridCalc. O programa possui duas abas centrais que separam o pré-processador e pós-processador. O pré-processador está contido no editor de geometria, sendo composto por um setor referente a criação e edição da geometria do problema (elementos e nós), criação e edição dos atributos (propriedades dos materiais, propriedades das seções, cargas e apoios) e um quadro de visualização em planta do problema (implementado pela classe Twodimensionalview).

Figura 27 – Tela do pré-processador do programa GridCalc.



Fonte: Autor.

A Tabela 2 mostra as principais funções utilizadas na implementação do pré-processador, sendo as funções *DrawRect()*, *DrawLine()* e *DrawText()* específicas da classe de visualização *Twodimensionalview*. As demais funções fazem parte da classe central da estrutura de dados *Core* (ver Figura 26).

Tabela 2 – Principais funções da interface gráfica utilizadas pelo pré-processador.

Função	Entrada	Descrição
<i>DrawRect ()</i>	Ponto de referência, comprimento, altura e cor do retângulo	Desenha no Twodimensionalview um retângulo de comprimento, altura, cor e coordenadas do vértice superior fornecidos.
<i>DrawLine ()</i>	Ponto inicial, ponto final e cor do elemento.	Desenha no Twodimensionalview uma linha de ponto inicial, final e cor fornecidos.
<i>DrawText ()</i>	Ponto de referência, uma frase, um tipo de cor.	Desenha no Twodimensionalview uma frase de cor e coordenada de posição fornecidos

Fonte: Autor

Tabela 3 – Principais funções da estrutura de dados utilizadas pelo pré-processador.

Função	Entrada	Descrição
--------	---------	-----------

CorrectLines ()	Não possui	Modifica a estrutura de dados de maneira a adicionar nós nos cruzamentos de elementos sobrepostos.
CreateMesh ()	Ponto de referência, comprimento na direção x e y da malha, divisão da malha na direção x e y.	Cria uma malha retangular de elementos utilizando o ponto inferior esquerdo como referência.
CreateNode ()	Ponto de referência.	Adiciona na estrutura de dados um nó situado num ponto (x,y,0) fornecido, verificando se este ponto não está contido na área de tolerância dos demais pontos.
LineIntersection()	Dois elementos.	Retorna o ponto de interseção dos dois elementos, ou 0 se não houver intercessão.
ImportProp ()	Elemento de referência.	Importa os atributos de um elemento para outro.
CreateElement ()	Ponto inicial e final do elemento.	Adiciona na estrutura de dados dois nós de coordenadas fornecidas e um elemento.
setNode()	Dois nós de referência	Altera os nós atuais do elemento, para os nós fornecidos.

Fonte: Autor.

O Quadro 1 mostra o algoritmo utilizado para criação dos nós da estrutura. Apesar dos nós da estrutura serem representados por pontos flutuantes precisos, a tela do computador é composta por um número inteiro de pixels, cada pixel representa um conjunto de pontos. Para programas de computador que utilizam entrada de dados com o uso da coordenada do mouse sobre janelas de visualização, não é possível mensurar precisamente qual ponto está sendo mapeado. Por exemplo, uma simples tentativa de ligar dois elementos, para um mesmo nó, poderia ser impossível utilizando a coordenada do mouse, se não fosse considerada uma região de tolerância. Por este motivo os nós da estrutura são representados por uma região quadrada, onde o lado do quadrado é a tolerância utilizada no problema, de maneira que não é permitido a criação de um novo nó que esteja dentro da área de tolerância de outro, conforme está ilustrado na Figura 28.

Figura 28 – Representação do nó através de sua região de tolerância.



Fonte: Autor.

Quadro 1 – Algoritmo utilizado para criação dos nós do problema.

```

Obter pela interface gráfica a coordenada do novo nó:  $x_p$  e  $y_p$ .
para  $i=1$  até  $n$  faça: ( $n$  é o numero total de nós)
    se  $x_p \in \{no(i).x \mp Tol/2\}$  &  $y_p \in \{no(i).y \mp Tol/2\}$  então
        fim (nó já existe!)
    senão
        Cria um novo nó na estrutura de dados: new Node ( $P_x, P_y$ )
    fim_se
fim_para

```

Fonte: Autor.

O Quadro 2 mostra o algoritmo utilizado para desenho na tela dos nós e elementos do problema. Uma vez que o usuário selecione nós ou elementos da estrutura, o programa aplica diferentes cores para visualização do estado de seleção.

Quadro 2 – Algoritmo utilizado para desenhar os nós e elementos do problema.

```

para  $i=1$  até  $n$  faça: ( $n$  é o numero total de nós)
    Obter o ponto do vértice superior esquerdo:

     $P^i \leftarrow [(no(i).x, no(i).y) - (Tol/2, -Tol/2)]$ 

    se  $no(i).sel$  então
         $cor \leftarrow$  vermelho
    senão
         $cor \leftarrow$  preto
    fim_se
    DrawRect( $P^i$ , Tol, Tol, cor) (desenha o nó)
fim_para

para  $i=1$  até  $m$  faça: ( $m$  é o numero total de elementos)
    Carregar pontos dos nós do elemento:  $elem(i).no1$  e  $elem(i).no2$ .
    se  $elem(i).sel$  então
         $cor \leftarrow$  vermelho
    senão
         $cor \leftarrow$  preto
    fim_se
    DrawLine( $elem(i).no1$ ,  $elem(i).no2$ , cor) (desenha o elemento)
fim_para

```

Fonte: Autor.

O Quadro 3 mostra o algoritmo utilizado para criação dos elementos. A condição para criação do elemento é de que não exista nenhum outro elemento na estrutura de dados, que possua os mesmos nós independente da ordem (n1, n2) ou (n2,n1).

Quadro 3 – Algoritmo utilizado para criar elementos.

```

Entrada pela interface:  $N_1$  e  $N_2$  (Nós iniciais e finais do elemento)

para  $i=1$  até  $n$  faça: ( $n$  é o numero total de elementos)
  se ( $\text{elem}(i).\text{no1}=N_1$  &  $\text{elem}(i).\text{no2}=N_2$ ) ou ( $\text{elem}(i).\text{no1}=N_2$  &  $\text{elem}(i).\text{no2}=N_1$ ) então
    retorna (elemento já existe!)
  senão
    CriaElem ( $N_1, N_2$ ) (criação de um novo elemento)
  fim_se
fim_para

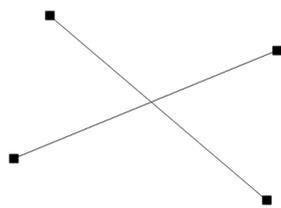
```

Fonte: Autor.

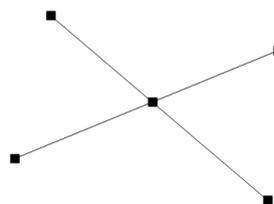
A presença de elementos sobrepostos, introduzidos pelo usuário, é um dos problema presente na implementação do pré-processador (ver Figura 29). Uma vez que seja adicionado, por exemplo, um elemento composto pelos nós (1,1) e (3,3), e outro elemento composto pelos nós (1,3) e (3,1), existe um ponto interseção entre estes elementos, que para aplicação do método da rigidez direta deverá existir um nó neste ponto, transformando os dois elementos em quatro elementos. O Quadro 4 mostra o algoritmo utilizado para realizar a correção de intercessões entre os elementos da estrutura de dados.

Figura 29 – Correção da intercessão.

Intercessão inserida pelo usuário



Correção com criação de 1 nó



Fonte: Autor.

Quadro 4 – Algoritmo utilizado para correção de intercessão dos elementos.

```

para i=1 até n faça: (n é o numero total de elementos)
  para j=1 até n faça:
    se LineIntersection(elem(i),elem(j)) então
      Calcula o ponto de interseção e cria um nó:
        CreateNode(xint,yint) (Nint)
      Armazena os nós finais dos elementos: Ni2 e Nj2
      Alterar o segundo nó de i e j como sendo Nint
        elem(i).setNode(Ni1,Nint)
        elem(j).setNode(Nj1,Nint)
      Cria os novos elementos:
        CreateElement(Nint, Ni2) (elemnovo (i))
        CreateElement(Nint, Nj2) (elemnovo (j))
      Atualiza os atributos dos elementos criados:
        elem(i).ImportProp(elemnovo (i))
        elem(j).ImportProp(elemnovo (j))
    fim_se
  fim_para
fim_para

```

Fonte: Autor.

Um dos objetivos do programa é tornar fácil a utilização do Método da Analogia de Grelha, para análise estrutural de pavimentos. Para isto foi desenvolvida a função *CreateMesh()*, que cria automaticamente uma malha de elementos de grelha. O Quadro 5 mostra o algoritmo utilizado para criação de uma malha retangular, que são as mais utilizadas em análise de pavimentos. Uma vez implementado o algoritmo presente no Quadro 4, é simples a implementação da malha, pois não há necessidade prévia de considerar os nós de intercessões, pois o algoritmo do Quadro 4 resolve este problema. A Figura 30 mostra um exemplo de malha gerada pelo programa desenvolvido.

Quadro 5 – Algoritmo utilizado para geração da malha quadrada.

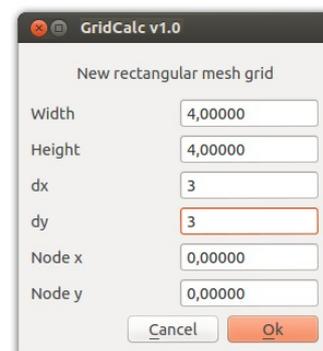
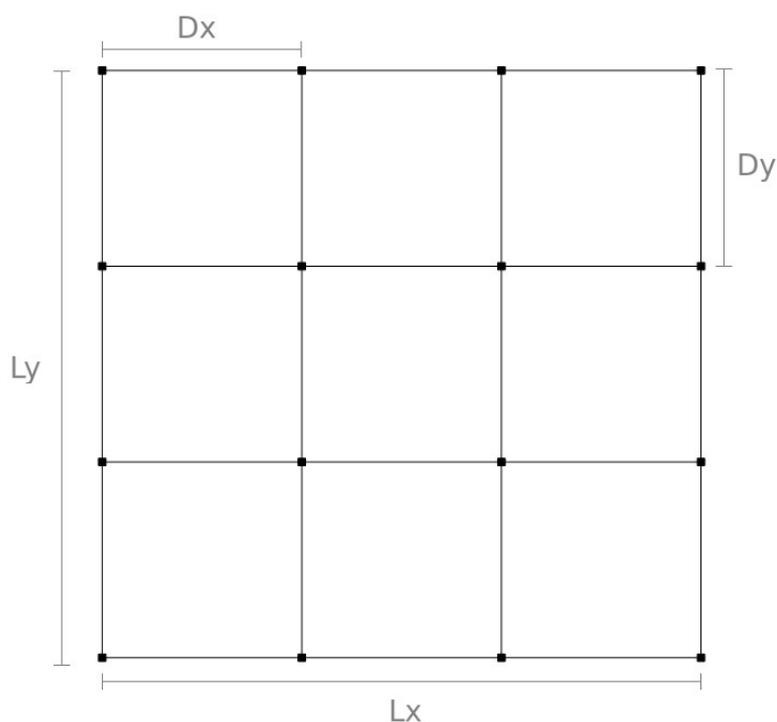
```

Entrada pela interface:
  Dimensões totais da malha, nas direções x e y:  $L_x$  e  $L_y$ 
  Número de Divisões da malha, nas direções x e y:  $d_x$  e  $d_y$ 
  Nos inferior esquerdo e direito da malha:  $N_i$  e  $N_f$ 
Calcular o comprimento dos elementos na direção x e y:
   $D_x \leftarrow L_x/d_x.$ 
   $D_y \leftarrow L_y/d_y.$ 
para  $m=0$  até  $d_y$  faça:
  CreateElement( $N_i + (0,m.D_y)$ ,  $N_j + (0,m.D_y)$ )
fim_para
para  $n=0$  até  $d_x$  faça:
  CreateElement( $N_i + (n.D_x,0)$ ,  $N_j + (n.D_x,0)$ )
fim_para
Executar o Quadro 4 de correção de interseções.

```

Fonte: Autor.

Figura 30 – Malha quadrada criada pelo programa GridCalc.



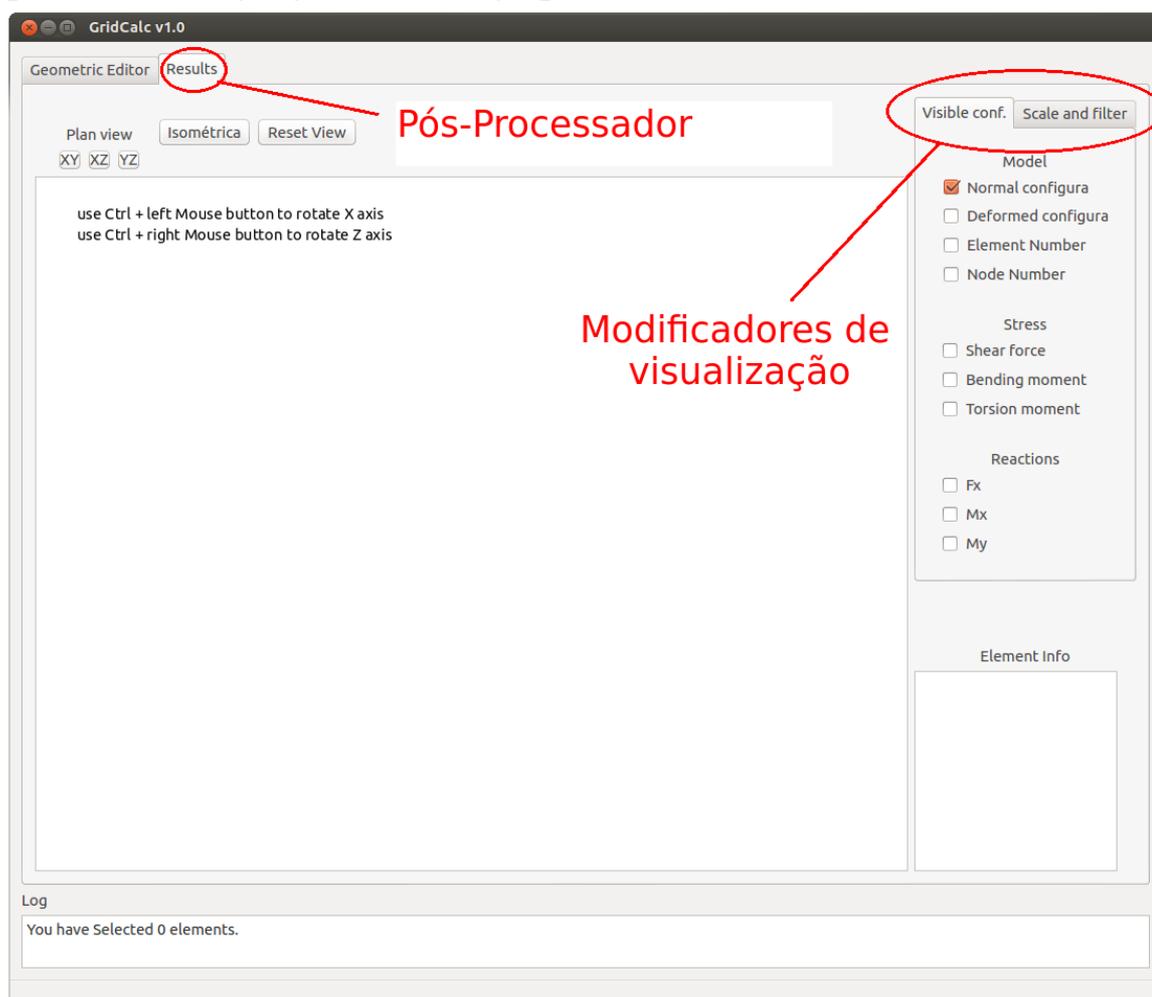
Fonte:

Autor.

### 3.3. Implementação do pós-processador

A Figura 31 mostra a tela do programa desenvolvido GridCalc, o pós-processador está contido no menu resultados, e é composto por um setor referente aos modificadores de visualização do problema e um quadro de visualização tridimensional dos resultados do problema, implementado pela classe GLWidget.

Figura 31 – Tela do pós-processador do programa GridCalc.



Fonte: Autor.

Os modificadores de visualização determinam qual parte da estrutura de dados será desenhada, regula as escalas do desenho e aplica filtros para condicionar o desenho de parte da estrutura.

O GLWidget utiliza as funções de desenho do OpenGL para a visualização tridimensional. A Figura 32 mostra o procedimento simplificado utilizado pelo OpenGL para desenho gráfico. Quando um programa faz chamadas de funções do OpenGL, este armazena os comandos, texturas, geometria e outros dados em um buffer de comandos. O buffer de comandos é esvaziado, dando início à segunda etapa onde parte dos dados sofreram transformação e efeitos de iluminação. Após isto é iniciada a etapa de Rasterização, quando será de fato realizada a montagem da imagem, ao final deste processo a imagem é exibida no monitor (Frame Buffer).

Figura 32 – Procedimento de desenho gráfico do OpenGL.



Fonte: Manssour (2013)

A Tabela 4 mostra as principais funções do OpenGL utilizadas no programa desenvolvido.

Tabela 4 – Principais funções do OpenGL utilizadas no programa.

Função	Entrada	Descrição
glBegin(GL_LINES)	Ponto inicial, ponto final e cor do elemento.	Desenha no GLWidget uma reta tridimensional com pontos e cor fornecidos.
glBegin(GL_POINTS)	Coordenada do ponto e cor.	Desenha no GLWidget um ponto de coordenada e cor fornecidos
glTranslate()	Vetor de Translação	Aplica uma translação de valor fornecido. através da matriz de projeção do modelo.
glRotate()	Ângulo de rotação e eixo de rotação	Aplica uma rotação no eixo fornecido. através da matriz de projeção do modelo.
glScale()	Fator de escala.	Aplica um fator de escala, através da matriz de projeção do modelo.

Fonte: Autor

A Tabela 5 mostra as principais funções utilizadas no pós-processador desenvolvido.

Tabela 5 – Principais funções GLWidget utilizadas no programa.

Função	Entrada	Descrição
DrawNode()	Ponto inicial, ponto final e cor do elemento.	Desenha no GLWidget uma reta tridimensional com pontos e cor fornecidos.
DrawNodeNumber()	Coordenada do ponto e cor.	Desenha no GLWidget um ponto de coordenada e cor fornecidos
DrawElementDeformed()	Elemento da lista de dados.	Desenha a deformada do elemento.
DrawElementShear()	Elemento da lista de dados.	Desenha o gráfico de esforço cortante do elemento.
DrawElementBending()	Elemento da lista de dados.	Desenha o gráfico de momento fletor do elemento.
DrawElementTorsion()	Elemento da lista de dados.	Desenha o gráfico do momento torsor do elemento.

Fonte: Autor

O desenho da configuração deformada da estrutura é realizado, utilizando a função de forma descrita na Equação (11), pela função criada no programa *DeformedZ()*, para determinar as coordenadas da curva deformada. Estas coordenadas são desenhadas na origem do eixo local de cada elemento, uma vez que o OpenGL possui funções para transladar e rotacionar parte do desenho inserido pelo código.

O Quadro 6 mostra o algoritmo utilizado para desenhar a configuração deformada de cada elemento, observe que a curva é composta por trechos de retas, que ao serem desenhados no GLWidget aparentam ser uma curva. Os algoritmos para desenho dos esforços, *DrawElementShear()*, *DrawElementBending()* e *DrawElementTorsion()* são análogos ao algoritmo para desenho da deformada.

Quadro 6 – Algoritmo utilizado para desenhar a deformada do elemento.

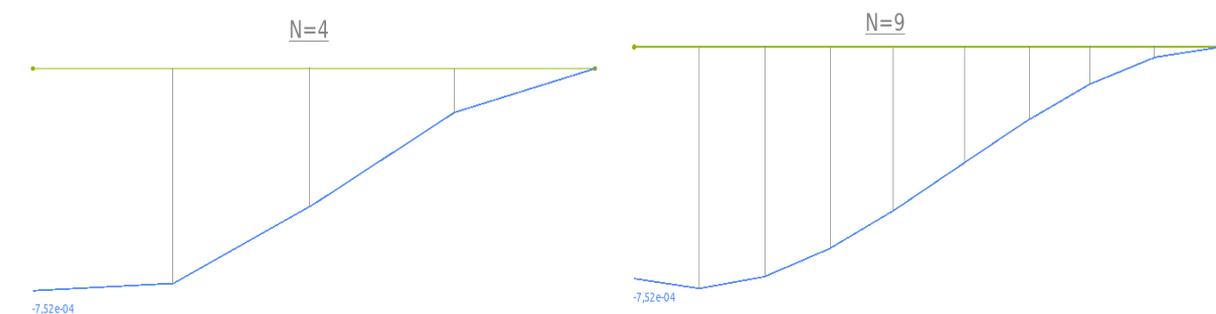
```

para i=1 até n faça: (n é o numero total de elementos)
  para m=1 até DeformedDivision faça: (m é um trecho do elemento)
    Calcular o comprimento de cada trecho:  $L_t \leftarrow (\text{elem}(i).\text{comprimento} / \text{DeformedDivision})$ 
    Calcular o ponto inicial do trecho:  $N_i \leftarrow \text{elem}(i).\text{DeformedZ}(m L_t)$ 
    Calcular o ponto final do trecho:  $N_f \leftarrow \text{elem}(i).\text{DeformedZ}((m+1) L_t)$ 
    Desenhar o trecho: DrawLine( $N_i, N_f, \text{azul}$ )
  fim_para
  Calcular o ângulo global do elemento  $\beta$ 
  Transladar a deformada: glTranslate( $P_{i1}$ )
  Rotaciona a deformada: glRotate( $\beta$ )
fim_para

```

Fonte: Autor.

Figura 33 – Exemplo de traçado de deformada, para dois números de divisões.



Fonte:

Autor.

Os desenhos dos esforços nas barras também são realizados no seus eixos locais para depois serem levados para o eixo global de cada elemento. O Quadro 7 mostra o algoritmo utilizado para desenho do pós-processador, onde são desenhados as geometrias e atributos relacionados aos nós, geometria e atributos relacionados aos elementos, e depois é realizada uma modificação global do modelo, que consiste em aumento da escala do desenho pela função *ScaleGlobal()*, rotação da câmera pela função *RotateGlobal()* e translação da câmera pela função *TranslateGlobal()*. Os comandos *ElementFilter()* e *NodeFilter()* são

utilizadas para condicionar o desenho do nó ou elemento, e os itens associados a eles, como por exemplo os gráficos dos esforços são associados aos elementos, as reações de apoio são associados aos nós.

Quadro 7 – Algoritmo utilizado para desenho do pós-processador.

```

para i=1 até m faça: (m é o numero total de nós)
  se NodeFilter(node(i)) for verdadeiro então
    DrawNode(node(i))
    DrawNodeNumber(node(i))
    DrawDeformedNode(node(i))
    DrawNodeFy(node(i))
    DrawNodeMx(node(i))
    DrawNodeMy(node(i))
  fim_se
fim_para

para i=1 até n faça: (n é o numero total de elementos)
  se ElementFilter(elem(i)) for verdadeiro então
    DrawElement(elem(i))
    DrawElementNumber(elem(i))
    DrawDeformedElement(elem(i))
    DrawElementShear(elem(i))
    DrawElementBending(elem(i))
    DrawElementTorsion(elem(i))
    Calcular o ângulo global do elemento  $\beta$ 
    Transladar a deformada: glTranslate ( $P_{i1}$ )
    Rotaciona a deformada: glRotate( $\beta$ )
  fim_se
fim_para

ScaleGlobal()
RotateGlobal()
TranslateGlobal()

```

Fonte: Autor.

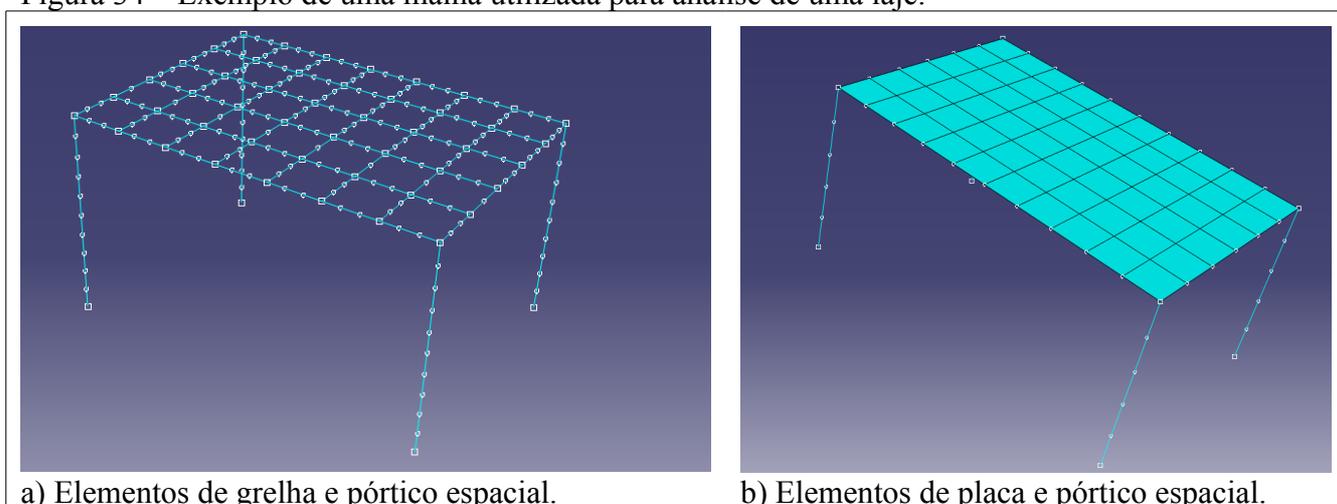
#### 4. ANÁLISE DE LAJES PELO MÉTODO DA ANALOGIA DE GRELHA

No projeto estrutural, podem ser utilizados diferentes métodos para a análise de pavimentos de edificações, dentre eles tem-se:

- Utilização de Tabelas de Marcus ou Czerny;
- Utilização de modelo de elementos finitos de placa;
- Utilização de Analogia de Grelha;

A análise de lajes através da analogia de grelha é um método consagrado entre os projetistas de estruturas, sendo atualmente o método mais utilizado pelos softwares de cálculo estrutural nacionais. O modelo de grelha possui implementação computacional simplificada quando comparado com o modelo utilizando Elementos Finitos de placas. A Figura 34 mostra um exemplo de uma laje modelada por elementos de grelha e elementos de placa.

Figura 34 – Exemplo de uma malha utilizada para análise de uma laje.



Fonte: Autor.

Antigamente, a maioria dos calculistas utilizavam as tabelas de Czerny ou Marcus para analisar as lajes, principalmente pela falta de computadores com configurações suficientes para executar programas de análise estrutural, entretanto, com o avanço da tecnologia, aos poucos estes profissionais foram utilizando cada vez mais os computadores para elaboração de projetos estruturais.

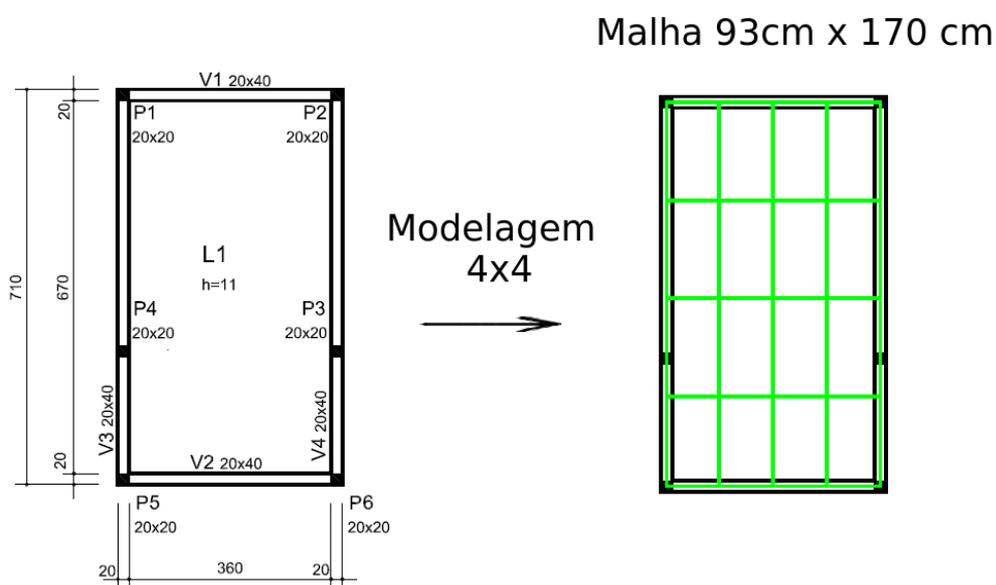
A análise de pavimentos utilizando as tabelas não permitem que sejam analisadas vigas e lajes simultaneamente, porém os métodos computacionais podem realizar a análise do pavimento de forma integrada, obtendo os resultados de viga e laje ao mesmo tempo. Além

disso pode-se considerar condições de contorno mais próximas da realidade, uma vez que a flecha das vigas influencia na flecha obtida nas lajes. Por estes motivos é que os Métodos de Analogia de Grelha e Elementos Finitos estão sendo utilizados, uma vez que é fácil e rápido a obtenção de resultados mais confiáveis e menos conservadores que os obtidos pelas tabelas de Marcus e Czerny.

#### 4.1. Analogia de grelha

O método de analogia de grelha consiste em modelar um determinado pavimento de uma edificação como sendo uma malha de elementos de grelha. A Figura 35 mostra um exemplo de utilização do método da analogia de grelha para uma laje maciça, onde elemento vertical possui comprimento de 170 cm e cada elemento horizontal possui comprimento de 93 cm. Observe que os elementos da extremidade representam as vigas da estrutura, e os nós onde possuem pilares devem ter deslocamento na direção z igual a zero.

Figura 35 – Exemplo de utilização da analogia de grelha.



Fonte: Autor.

No caso de lajes maciças, a quantidade e posição dos elementos de grelhas é indeterminado, sendo função apenas da discretização utilizada no modelo. Para lajes nervuradas, os elementos devem ser localizados nos eixos das nervuras, para que cada elemento de grelha represente um elemento de seção T, formado pela nervura e mesa.

## 4.2. Rigidez a flexão

A rigidez à flexão das seções dos elementos são calculados pelo produto do módulo de elasticidade transversal e o momento de inércia da seção transversal dos elementos. Segundo a NBR 6118 (ABNT, 2007), o módulo de elasticidade secante do concreto pode ser calculado pela expressão:

$$E = 0,85 \times 5600 \times \sqrt{f_{ck}} \quad (12)$$

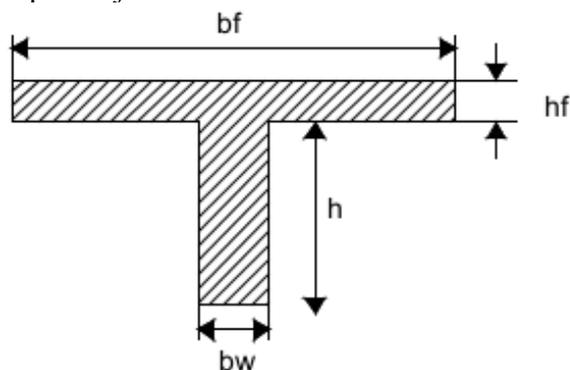
Onde  $E$  é o módulo de elasticidade secante do concreto e  $f_{ck}$  é a resistência a compressão do concreto. Os valores de  $E$  e  $f_{ck}$  são dados em MPa.

No caso de lajes maciças o momento de inércia pode ser calculado pela Equação (13), considerando uma seção transversal retangular:

$$I = \frac{bh^3}{12} \quad (13)$$

Onde  $I$  é o momento de inércia da seção transversal do elemento;  $b$  é a largura da faixa considerada na malha e  $h$  é a altura da laje.

Figura 36 – Seção transversal T de um elemento de grelha para laje nervurada.



Fonte: Autor.

No caso de lajes nervuradas, a seção é composta pela nervura e mesa, formando uma seção transversal T. A Figura 36 ilustra a seção de um elemento de grelha para laje nervuradas, segundo STRAMANDINOLI (2003), o momento de inércia pode ser calculado pela Equação:

$$I = \frac{b_f h_f^3}{12} + b_f h_f \left( y_{cg} - \frac{h_f}{2} \right)^2 + \frac{b_w h^3}{12} + b_w h \left( h_f + \frac{h}{2} - y_{cg} \right)^2 \quad (14)$$

Onde  $b_f$  é a largura da seção T;  $b_w$  é a largura da nervura;  $h$  é a diferença de altura da nervura com a mesa;  $h_f$  é a altura da mesa;  $y_{cg}$  é a altura do centroide da seção T medida a partir da borda superior e pode ser calculado pela expressão:

$$y_{cg} = \frac{b_w h \left( h_f + \frac{h}{2} \right) + b_f h_f^2}{b_w h + b_f h_f} \quad (15)$$

Kennedy e Bali (1978) consideram que a área da mesa deve ser majorada por um fator  $\frac{1}{1-\nu^2}$  para considerar o efeito de Poisson. Com esta consideração, o momento de inércia da seção T é calculado pela equação (16) e o  $y_{cg}$  é calculado pela equação (17).

$$I = \frac{b_f h_f^3}{12(2-\nu^2)} + \frac{b_f h_f \left( y_{cg} - \frac{h_f}{2} \right)^2}{(1-\nu^2)} + \frac{b_w h^3}{12} + b_w h \left( h_f + \frac{h}{2} - y_{cg} \right)^2 \quad (16)$$

$$y_{cg} = \frac{b_w h \left( h_f + \frac{h}{2} \right) + \frac{b_f h_f^2}{2(1-\nu^2)}}{b_w h + \frac{b_f h_f}{(1-\nu^2)}} \quad (17)$$

### 4.3. Rigidez à torção

A rigidez à torção dos elementos de grelha são calculados pelo produto do módulo de elasticidade transversal pelo momento polar de inércia da seção transversal do elemento. O módulo de elasticidade longitudinal ( $G$ ) de um material isotrópico é dado por:

$$G = \frac{E}{2(1+\nu)} \quad (18)$$

De acordo com a NBR 6118 (2007), o valor do coeficiente de Poisson para estruturas de concreto é igual a 0,2.

Para lajes maciças, segundo Montoya (1973) e Humbly (1976), o momento de inércia à torção ( $J$ ) pode ser calculado utilizando a expressão:

$$J=2I \quad (19)$$

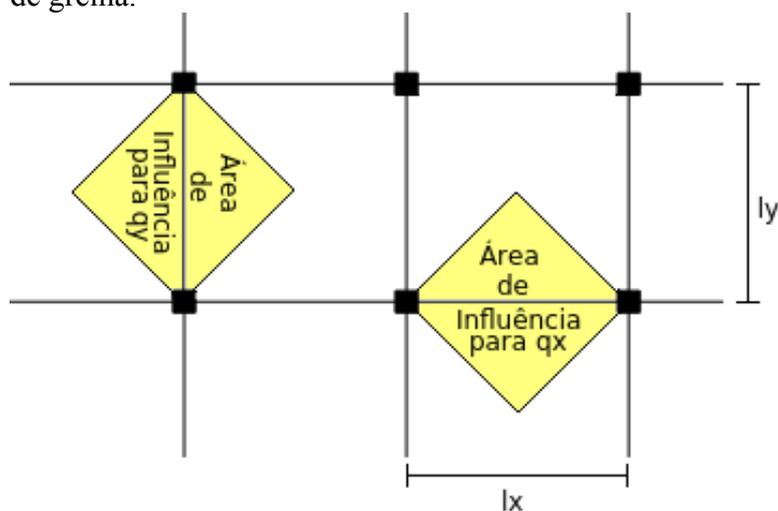
O motivo desta consideração está na diferente concepção da torção, entre o elemento de placa delgada e o elemento de grelha.

No caso de lajes nervuradas, Stramandinoli (2003) recomenda desprezar a rigidez à torção dos elementos, considerando  $J$  igual à zero, para análises elásticas. Esta recomendação está relacionada com a dificuldade construtiva de utilizar estribos e barras longitudinais adicionais nas nervuras das lajes.

#### 4.4. Carregamento

O carregamento é obtido considerando a área de influência de cada elemento, uma vez conhecida a malha da grelha e o carregamento distribuído no pavimento, como está ilustrado na Figura 37.

Figura 37 – Carregamentos distribuídos sobre os elementos de grelha.



Fonte: Autor.

Pôde-se calcular o carregamento distribuído uniformemente em cada elemento na direção  $x$  pela Equação (20) e em cada elemento na direção  $y$  pela Equação (21).

$$q_x = \frac{q_{metro^2} \cdot A_i}{l_x} \quad (20)$$

$$q_y = \frac{q_{metro^2} \cdot A_i}{l_y} \quad (21)$$

Onde  $q_x$  é a carga distribuída sobre o elemento da direção x,  $q_y$  é a carga distribuída sobre o elemento da direção y,  $q_{metro^2}$  é a carga uniformemente distribuída na laje,  $A_i$  é a área de influência do elemento,  $l_x$  é o espaçamento da malha na direção x e  $l_y$  é o espaçamento da malha na direção y.

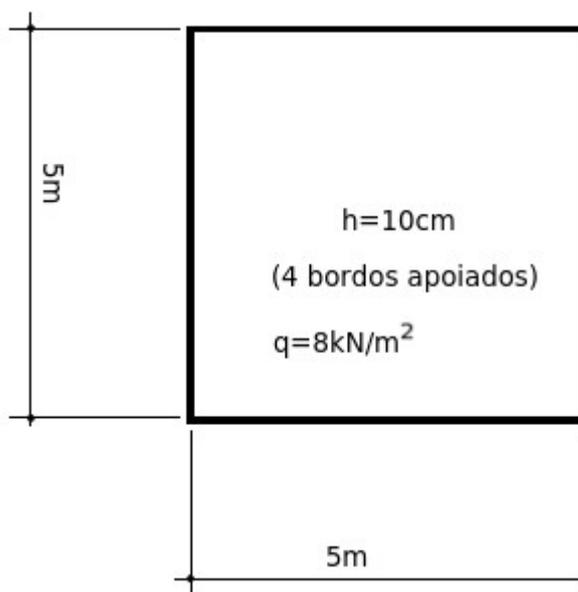
#### 4.5. Exemplos

Os exemplos a seguir têm o objetivo de mostrar, utilizando o programa desenvolvido, uma aplicação do método de analogia de grelha para lajes maciças. Os resultados são comparados com os da Teoria de Placas, obtidos com a utilização das tabelas de Czerny, presentes no Anexo A. Para melhor comparação entre os métodos, o deslocamento em z das vigas também foi considerado nulo em suas extremidades, da mesma forma que é considerado pelas tabelas de Czerny.

##### 4.5.1 Exemplo 1

A Figura 38 mostra a geometria de uma laje analisada no Exemplo 1. A laje utiliza o material de concreto armado de  $f_{ck}$  igual a 30 MPa. O coeficiente de Poisson do concreto vale 0,2.

Figura 38 – Laje do exemplo 1.



Fonte: Autor.

O módulo de elasticidade pode ser calculado de acordo com a Equação (12).

$$E = 0,85 \times 5600 \times \sqrt{30} = 26071593,74 \text{ kPa}$$

Utilizando a tabela de Czerny, pode-se calcular os momentos no centro da placa e a flecha máxima:

$$M_x = M_y = \frac{q l_x^2}{\alpha_x} = \frac{8 \times 25}{22,7} = 8,8 \text{ kNm/m}$$

$$w_{max} = \frac{q l_x^4}{E h^3 \alpha_2} = \frac{8 \times 5^4}{26071593,74 \times 0,1^3 \times 21,4} = 8,96 \times 10^{-3} \text{ m}$$

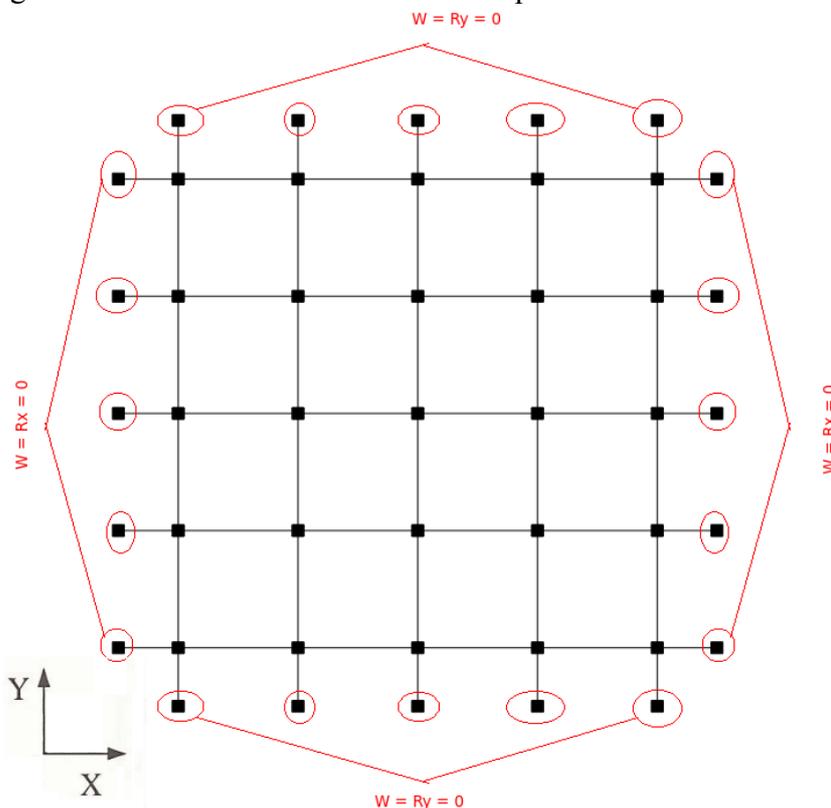
A laje do exemplo foi analisada por analogia de grelha e foram utilizadas três malhas para estudo da influência da discretização do modelo. A rigidez à flexão foi calculada segundo a Equação (13) e a rigidez à torção foi calculada segundo a recomendação de Humbly (1976) pela Equação (19).

a) Malha 1m x 1m

A Figura 39 mostra a primeira malha utilizada para análise da laje por analogia de grelha. Os nós de extremidade são modelados para impedir a translação no eixo Z e rotação que gere torção nos elementos, por isso, a condição de apoio referente a rotação dos nós de extremidade, que recebem elementos do eixo X, é impedir rotação em torno do eixo X e para

os nós que recebem elementos do eixo Y é impedir rotação em torno do eixo Y, conforme ilustrado na Figura 39.

Figura 39 – Malha 1x1 utilizada no exemplo 1.



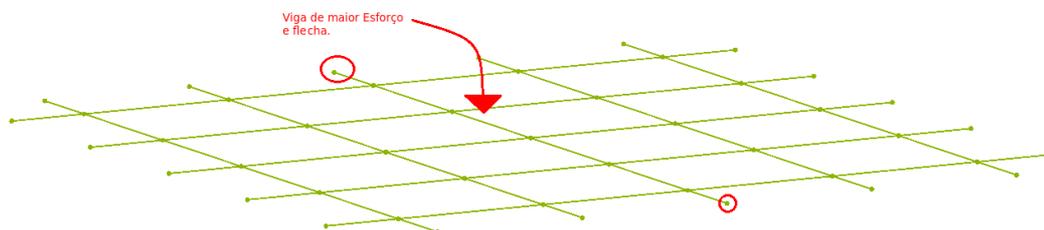
Fonte: Autor.

A carga distribuída nos elementos foi calculada considerando a área de influência de cada elemento. Pode-se observar que os elementos laterais e os de canto possuem a mesma carga distribuída que os demais:

$$q = \frac{1 \times 1 \times 8}{2 \times 1} = 4 \text{ kN/m}$$

A Figura 40 mostra a malha tridimensional visualizada no programa *GridCalc*, a viga marcada possui o maior momento fletor e maior flecha da estrutura.

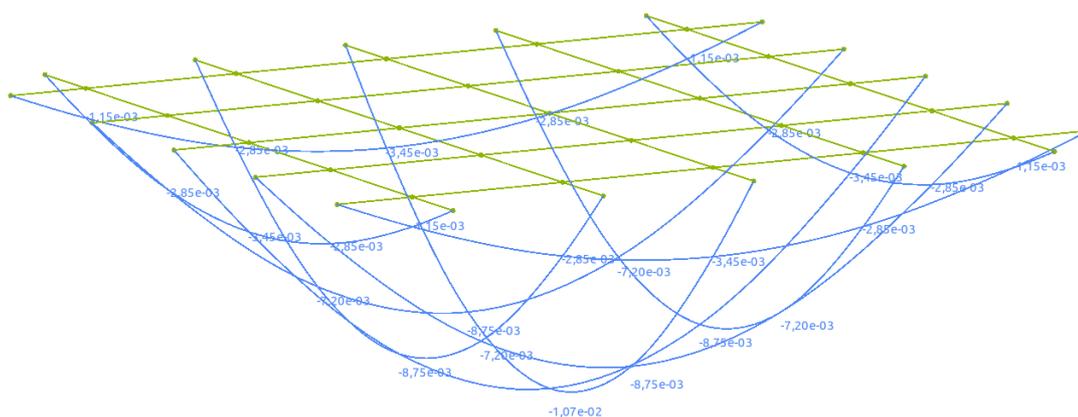
Figura 40 – Malha tridimensional 1m x 1m visualizada no programa GridCalc.



Fonte: Autor.

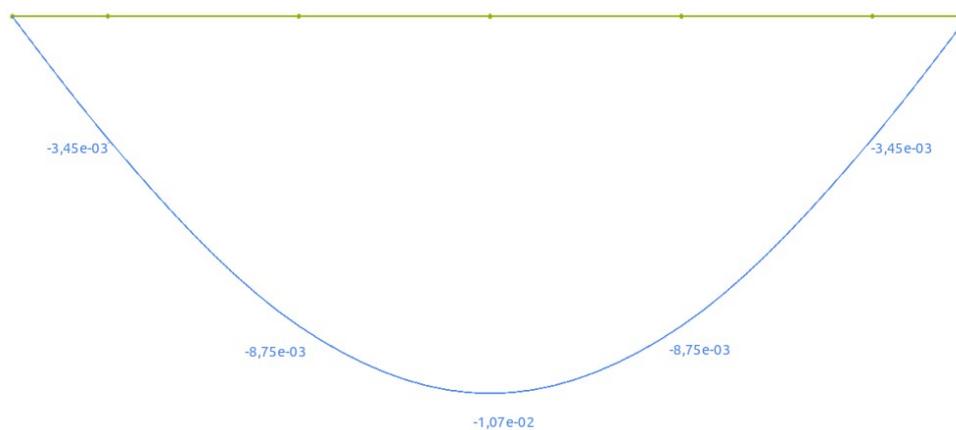
A Figura 41 mostra a deformada tridimensional visualizada pelo pós-processador, Figura 42 mostra a deformada da viga central, onde é obtido a flecha máxima.

Figura 41 – Deformada tridimensional visualizada no programa GridCalc.



Fonte: Autor.

Figura 42 – Deformada da viga central da grelha do exemplo 1 de malha 1 m x 1m.

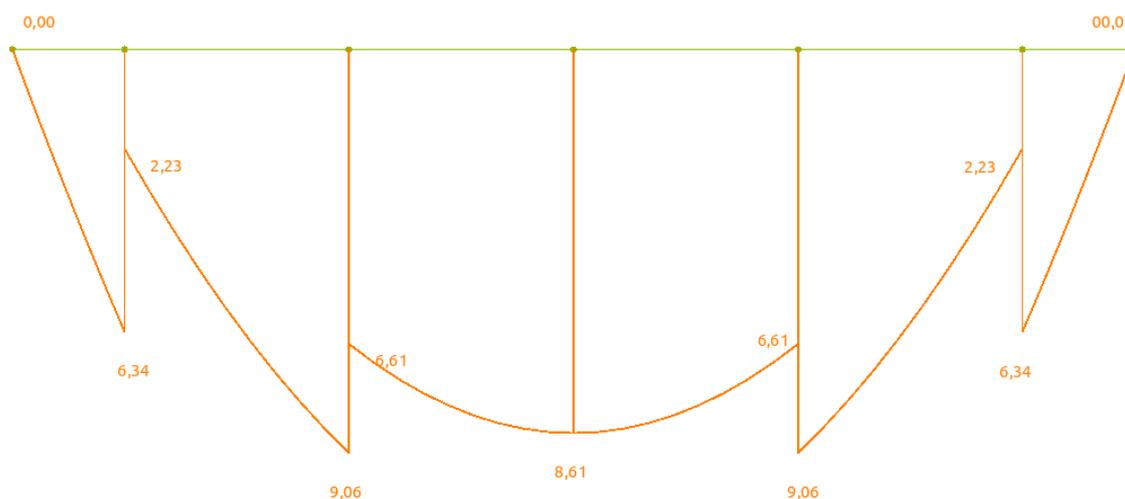


Fonte: Autor.

A Figura 43 mostra o momento fletor da viga central, onde foi obtido um

momento fletor de 9,06 kNm/m.

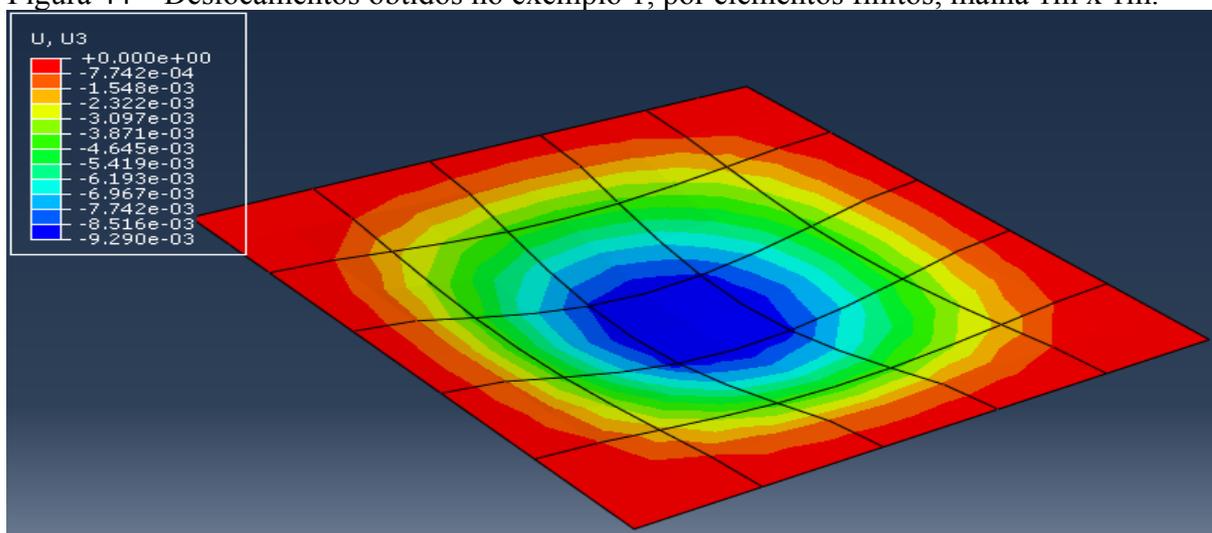
Figura 43 – Momento fletor da viga central da grelha do exemplo 1 de malha 1m x 1m.



Fonte: Autor.

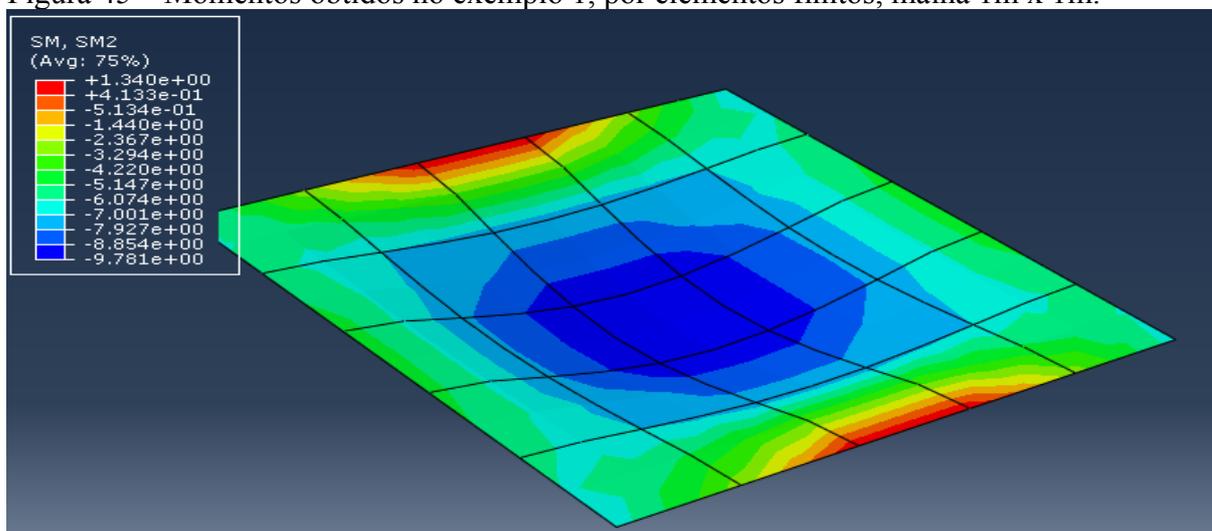
A Figura 44 e Figura 45 mostram os resultados obtidos por elementos finitos de placa, da malha utilizada.

Figura 44 – Deslocamentos obtidos no exemplo 1, por elementos finitos, malha 1m x 1m.



Fonte: Autor.

Figura 45 – Momentos obtidos no exemplo 1, por elementos finitos, malha 1m x 1m.

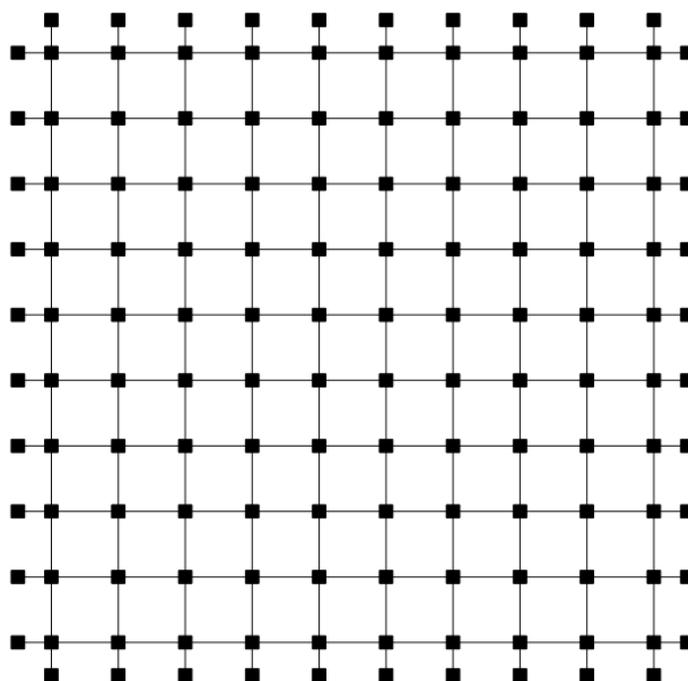


fonte:Autor.

b) Malha 0.5m x 0.5m

A Figura 46 mostra a segunda malha utilizada no exemplo 1, ela possui espaçamento de 0,5 metros.

Figura 46 – Malha 0,5x0,5 utilizada no exemplo 1.



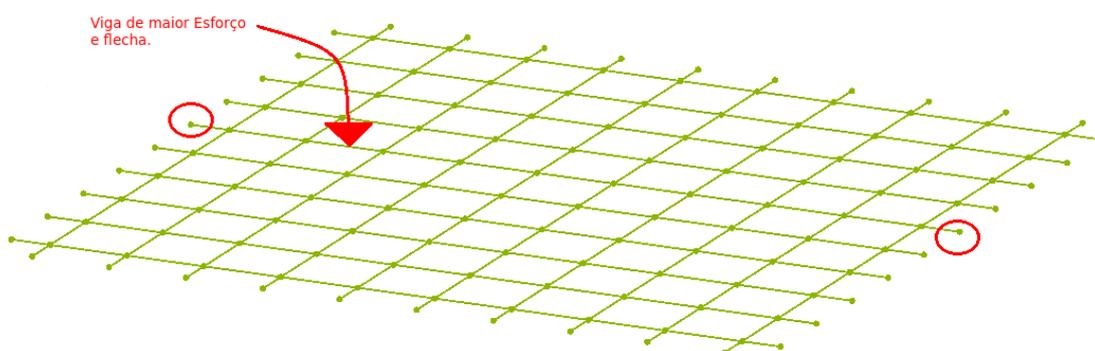
Fonte: Autor.

A carga distribuída nos elementos foi calculada considerando a área de influência de  $0,125\text{m}^2$ :

$$q_x = q_y = \frac{0.5 \times 0.5 \times 8}{2 \times 0.5} = 2 \text{ kN/m}$$

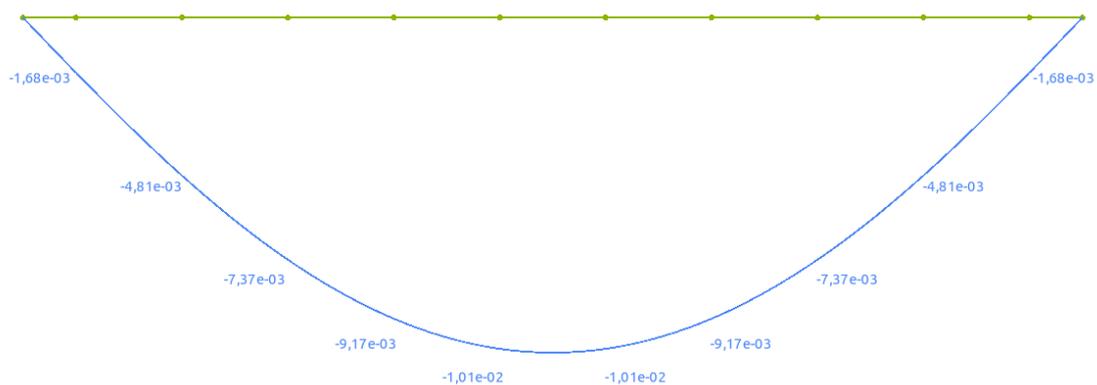
A Figura 47 mostra a nova malha 0.5x0.5 visualizada pelo pós-processador, a viga central que está indicada, possui o momento e flecha máxima do problema. A figura Figura 48 mostra a deformada da viga indicada.

Figura 47 – Malha tridimensional 0,5x0,5 visualizada no programa GridCalc.



Fonte: Autor.

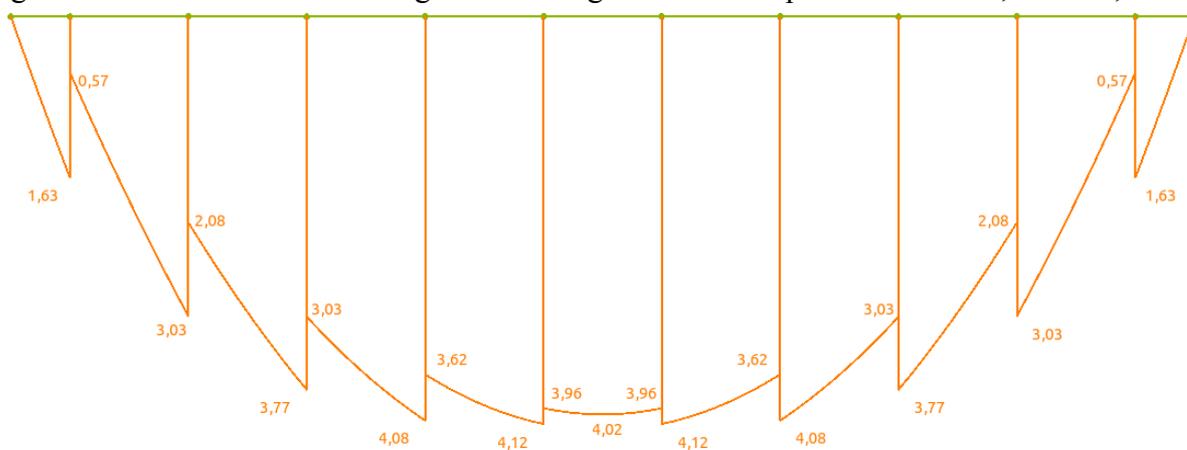
Figura 48 – Deformada da viga central da grelha do exemplo 1 de malha 0,50x0,50.



Fonte: Autor.

A Figura 49 mostra o diagrama de momentos fletores da viga central da malha, onde foi obtido um momento fletor de 8,24 kNm/m.

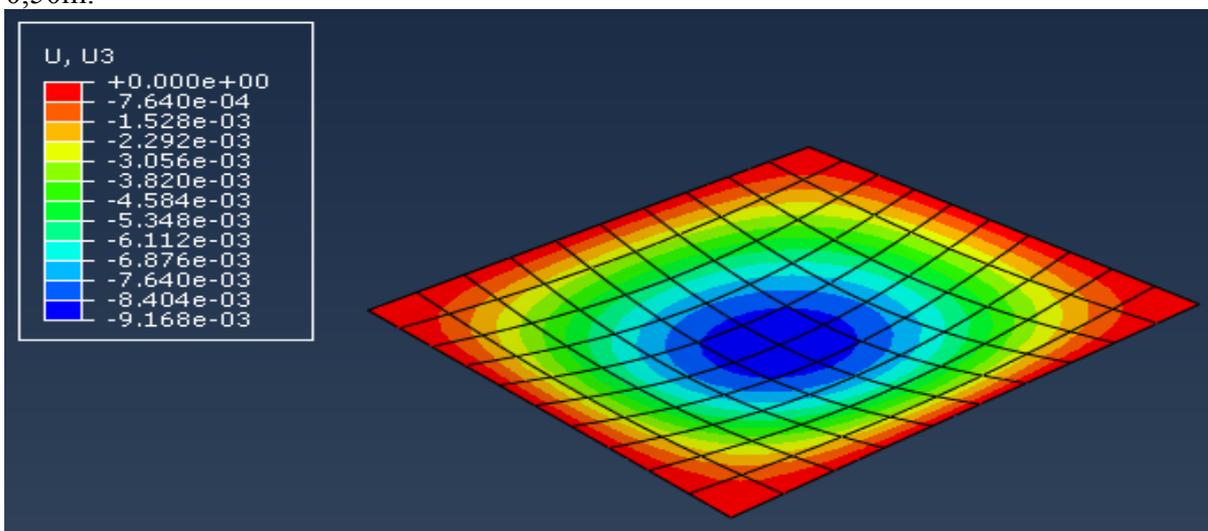
Figura 49 – Momento fletor da viga central da grelha do exemplo 1 de malha 0,50m x 0,50m.



Fonte: Autor.

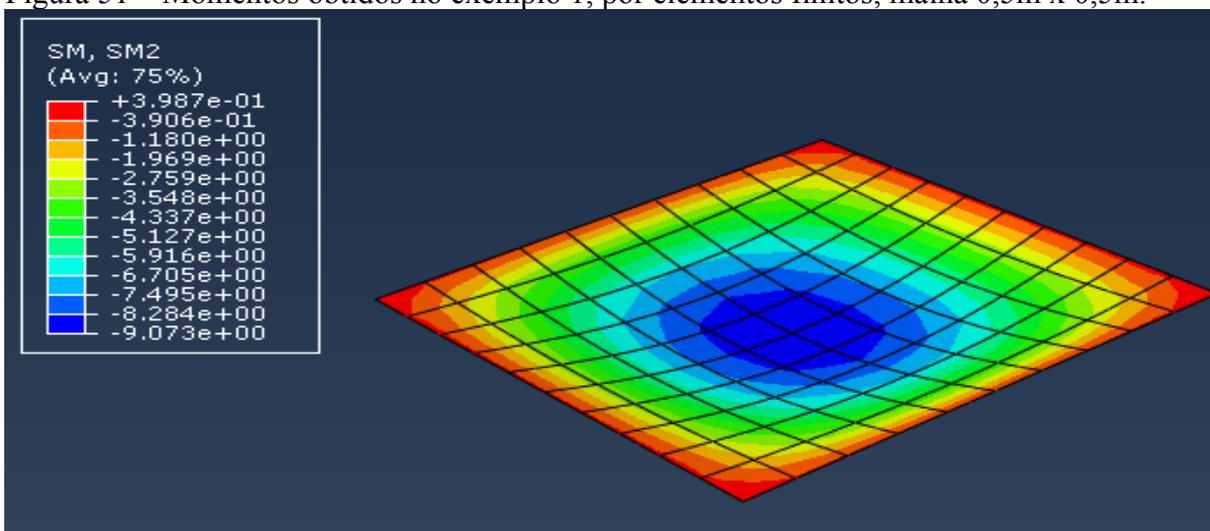
A Figura 50 e Figura 51 mostram os resultados obtidos por elementos finitos de placa, da malha utilizada.

Figura 50 – Deslocamentos obtidos no exemplo 1, por elementos finitos, malha 0m50m x 0,50m.



Fonte: Autor.

Figura 51 – Momentos obtidos no exemplo 1, por elementos finitos, malha 0,5m x 0,5m.

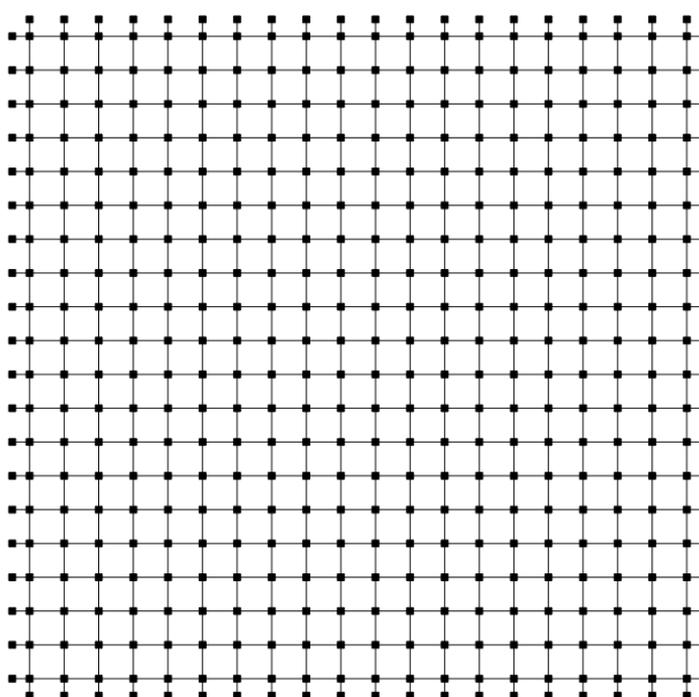


fonte:Autor.

c) Malha 0,25m x 0,25m.

A Figura 52 mostra a terceira malha utilizada no exemplo 1, ela possui espaçamento de 0,25 m.

Figura 52 – Malha 0.25m x 0.25m utilizada no exemplo 1.



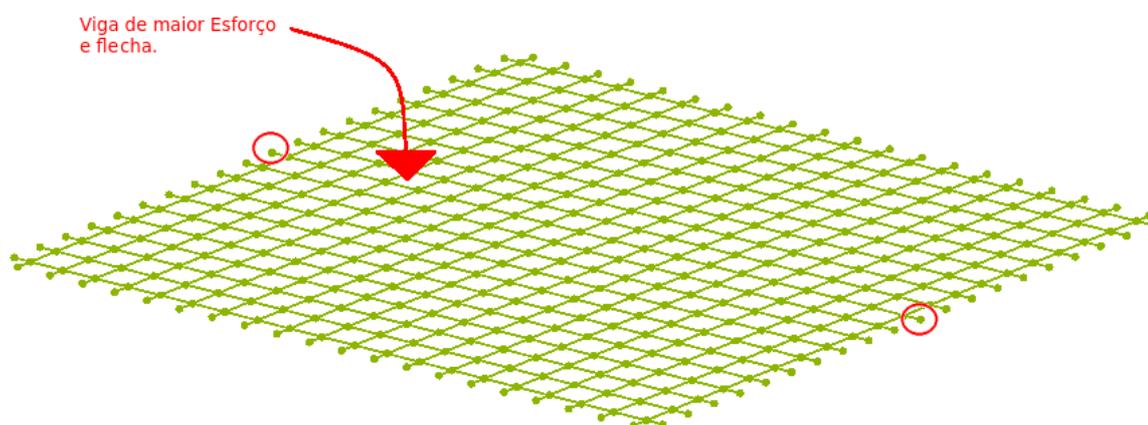
Fonte: Autor.

A carga distribuída nos elementos foi calculada considerando a área de influência de  $0,0625\text{m}^2$ :

$$q_x = q_y = \frac{0.25 \times 0.25 \times 8}{2 \times 0.25} = 1 \text{ kN/m}$$

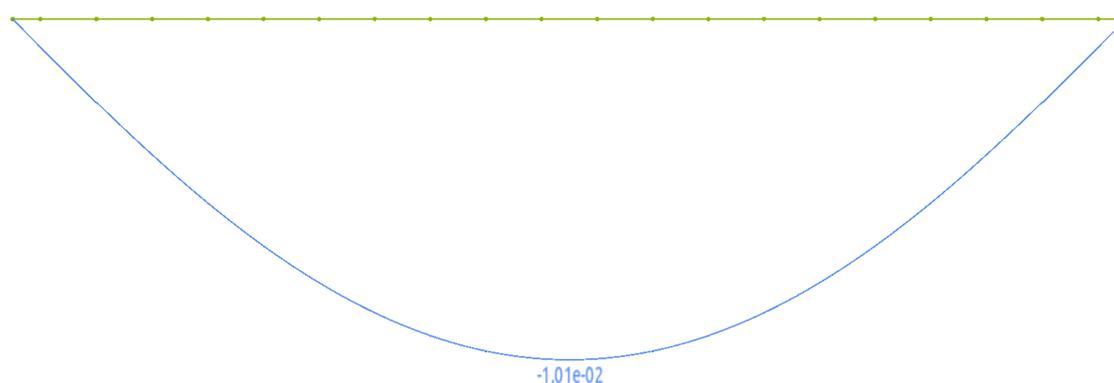
A Figura 53 mostra a nova malha 0.25x0.25 visualizada pelo pós-processador, a décima primeira viga que está indicada, possui o momento e flecha máxima do problema. A figura Figura 54 mostra a deformada da viga indicada.

Figura 53 – Malha tridimensional visualizada no programa GridCalc.



Fonte: Autor.

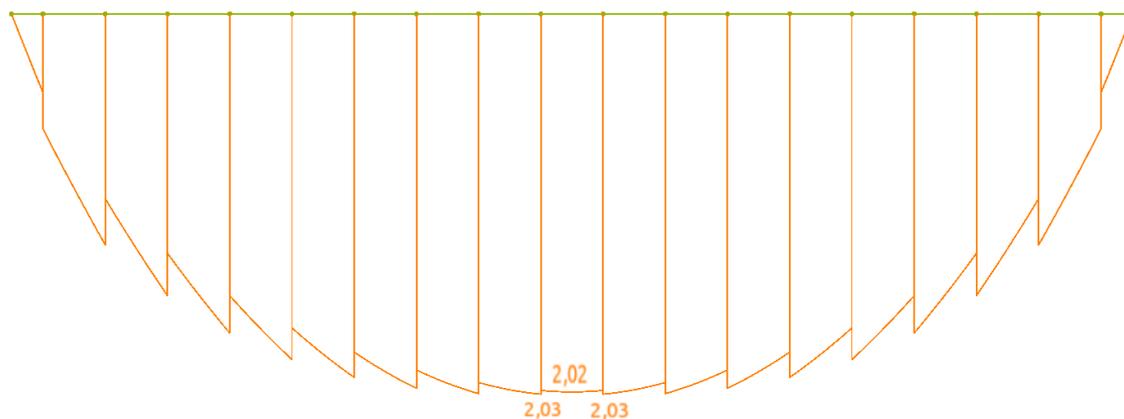
Figura 54 – Deformada da viga central da grelha do exemplo 1 de malha 0,25x0,25.



Fonte: Autor.

A Figura 49 mostra o diagrama de momentos fletores da viga central da malha, onde foi obtido um momento fletor de 8,12 kNm/m.

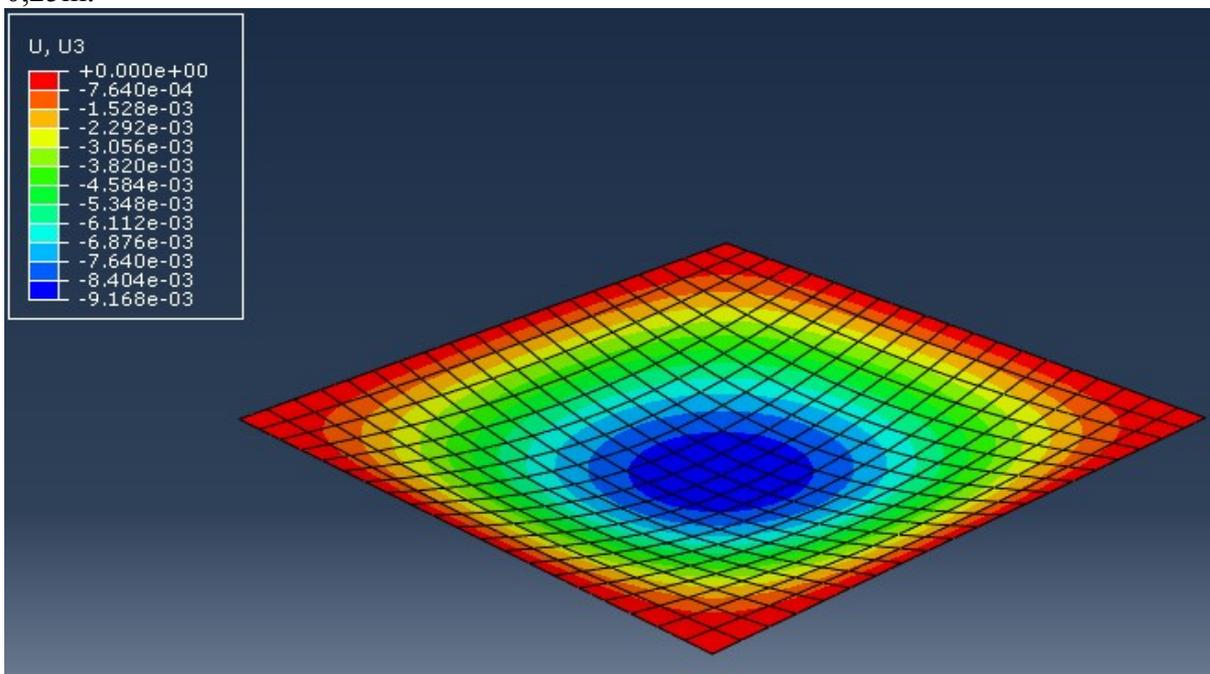
Figura 55 – Momento fletor da viga central da grelha do exemplo 1 de malha 0,25x0,25.



Fonte: Autor.

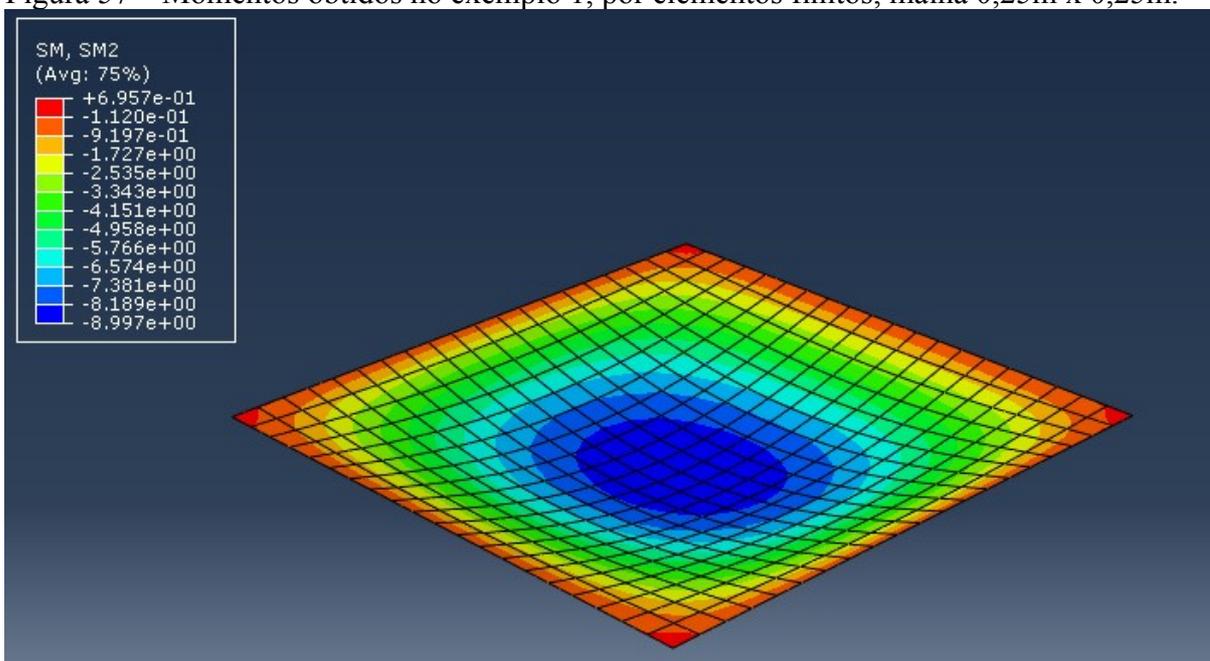
A Figura 56 e Figura 57 mostram os resultados obtidos por elementos finitos de placa, da malha utilizada.

Figura 56 – Deslocamentos obtidos no exemplo 1, por elementos finitos, malha 0,25m x 0,25m.



Fonte: Autor.

Figura 57 – Momentos obtidos no exemplo 1, por elementos finitos, malha 0,25m x 0,25m.



Fonte: Autor.

A Tabela 6 mostra um resumo das propriedades utilizadas nos elementos em cada análise realizada.

Tabela 6 – Resumo das propriedades dos elementos utilizadas no exemplo 1 e 2, para analogia de grelha.

Malha utilizada	I (m <sup>4</sup> )	J (m <sup>4</sup> )	Q (kN/m <sup>2</sup> )
1m x 1m	8,334x10 <sup>-5</sup>	1,1667x10 <sup>-4</sup>	4
0,5m x 0,5m	4,1667x10 <sup>-5</sup>	8,334x10 <sup>-5</sup>	2
0,25m x 0,25m	2,0834x10 <sup>-5</sup>	4,1667x10 <sup>-5</sup>	1

Fonte: Autor.

A Tabela 7 mostra os resultados obtidos nas análises realizadas no exemplo 1, comprando os momentos e flechas obtidas de cada malha com os resultados analíticos da Teoria das Placas.

Tabela 7 – Resultados obtidos no exemplo 1.

	Momento fletor máximo (kNm/m)	Diferença do momento (%)	Flecha máxima (mm)	Diferença da flecha (%)
Malha 1x1	9,06	+2,95	10,70	+19,42
Malha 0,5x0,5	8,24	-6,36	10,10	+12,72
Malha 0,25x0,25	8,12	-7,73	10,10	+12,72
Elementos Finitos de Placa (1x1)	9,78	+11,14	9,29	+3,68
Elementos Finitos de Placa (0,5x0,5)	9,07	+3,07	9,17	+2,34
Elementos Finitos de Placa (0,25x0,25)	9,00	+2,27	9,17	+2,34
Teoria das Placas	8,80		8,96	

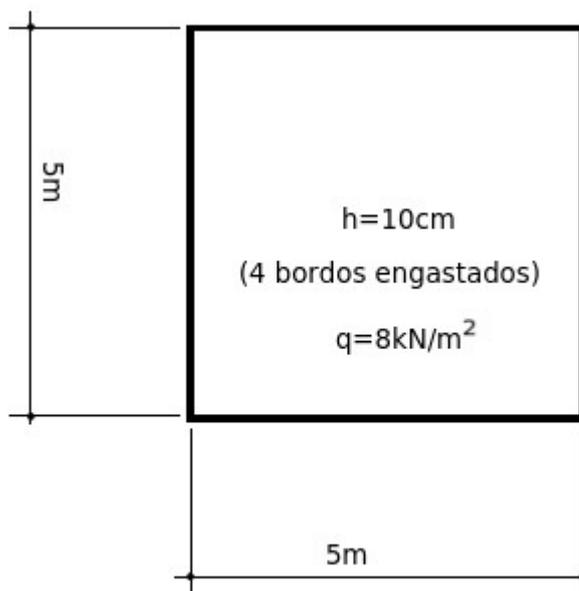
Fonte: Autor.

Pode-se observar que os valores de momento fletor obtidos pela analogia de grelha não convergiram para a solução analítica, além de tender a valores menores que os obtidos pela Teoria das Placas, a medida que é utilizada uma malha mais densa. A flecha obtida nas análises por analogia de grelha são maiores que os resultados analíticos da Teoria das Placas. Os momentos máximos não ocorreram no centro da laje, devido as discontinuidades geradas pela rigidez à torção dos elementos. Com a maior discretização da malha, observa-se uma menor influência destas discontinuidades. O modelo por elementos finitos de placa apresentou convergência para os valores da solução analítica.

#### 4.5.2 Exemplo 2

O Exemplo 2 possui a mesma geometria e carregamento do exemplo 1, mudando a condição de contorno da laje para engastada nos quatro bordos, conforme ilustrado na Figura 58.

Figura 58 – Laje do exemplo 2.



Fonte: Autor.

Utilizando a tabela de Czerny, pode-se calcular os momentos positivos no centro da placa, os momentos negativos máximos e a flecha máxima:

$$M_x = M_y = \frac{ql_x^2}{\alpha_x} = \frac{8 \times 25}{47,3} = 4,23 \text{ kNm/m}$$

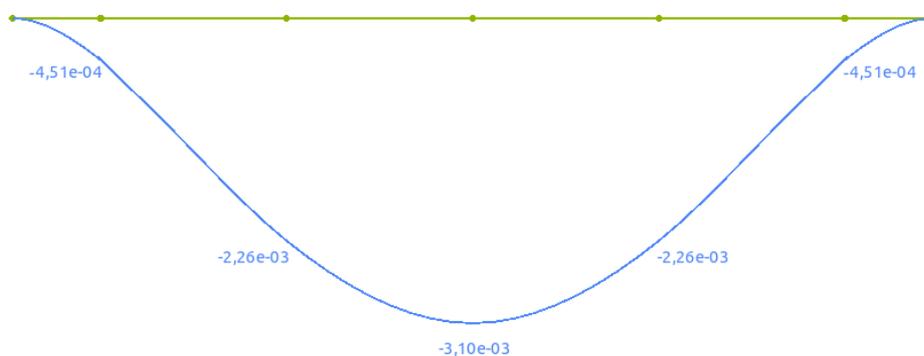
$$M'_x = M'_y = \frac{-8 \times 5^2}{19,4} = -10,31 \text{ kNm/m}$$

$$w_{max} = \frac{ql_x^4}{Eh^3\alpha_2} = \frac{8 \times 5^4}{26071593,74 \times 0,1^3 \times 68,50} = 2,8 \times 10^{-3} \text{ m}$$

a) Malha 1m x 1m

A malha do exemplo 2 com espaçamento de 1 metro pode ser visualizada na Figura 39. A Figura 59 mostra a deformada da viga central da grelha (ver Figura 40), onde ocorrem os momentos e flecha máximas da laje. O carregamento dos elementos é igual ao calculado no exemplo 1, na malha 1m x 1m.

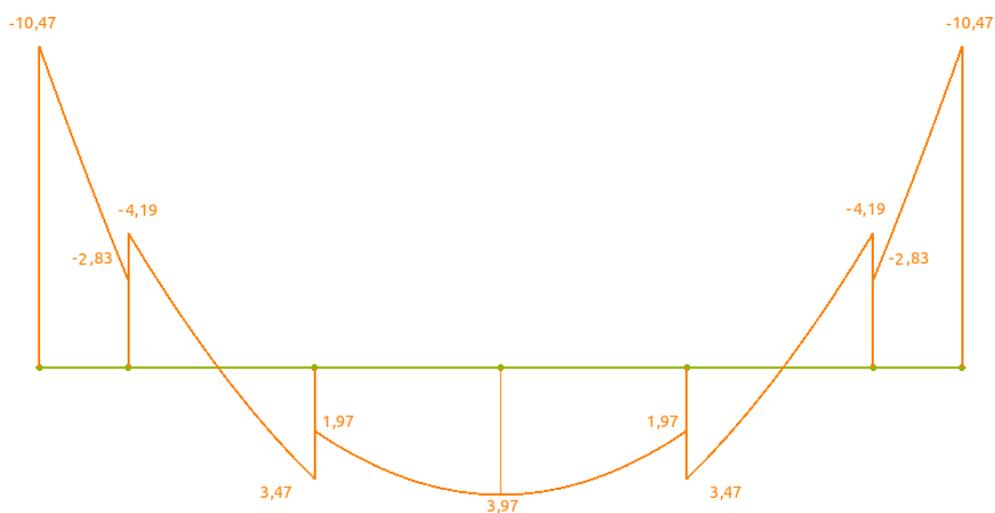
Figura 59 – Deformada da viga central da grelha do exemplo 2 de malha 1x1.



Fonte: Autor.

A Figura 60 mostra os momentos fletores obtidos na viga central da grelha, onde foram obtidos para momento fletor negativo máximo  $-10,47$  kNm/m e positivo  $3,97$  kNm/m.

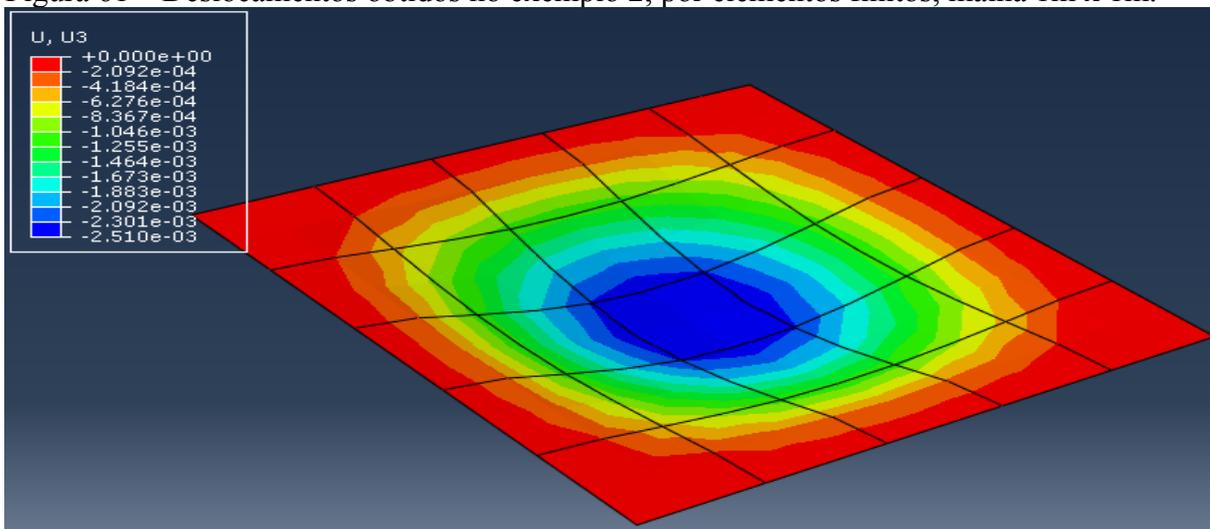
Figura 60 – Momento fletor da viga central da grelha do exemplo 2 de malha 1x1.



Fonte: Autor

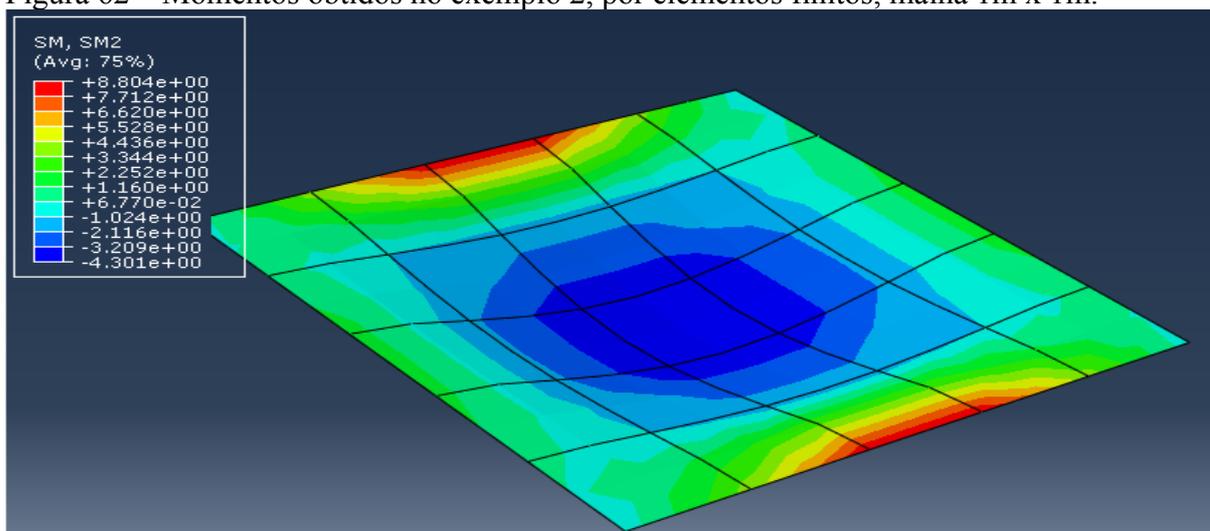
A Figura 61 e Figura 62 mostram os resultados obtidos por elementos finitos de placa, da malha utilizada.

Figura 61 – Deslocamentos obtidos no exemplo 2, por elementos finitos, malha 1m x 1m.



Fonte: Autor.

Figura 62 – Momentos obtidos no exemplo 2, por elementos finitos, malha 1m x 1m.

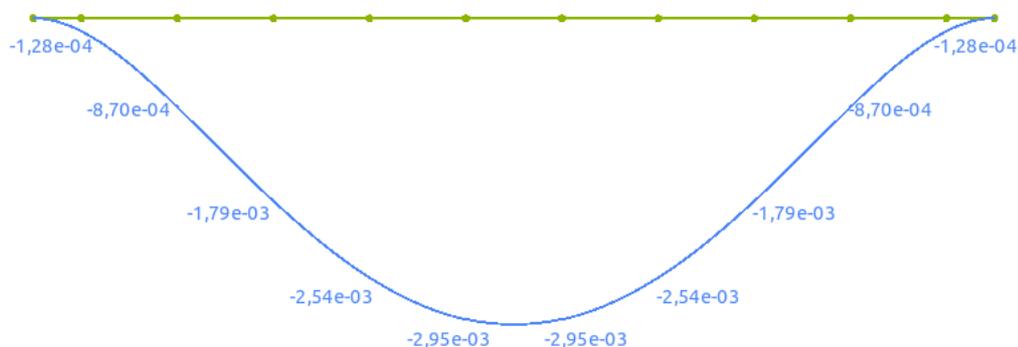


Fonte: Autor.

b) Malha 0,50m x 0,50m

A malha do exemplo 2 com espaçamento de 0,5 metros pode ser visualizada na Figura 46. A Figura 63 mostra a deformada da viga central da grelha (ver Figura 47), onde ocorrem os momentos e flecha máxima da laje. O carregamento dos elementos é igual ao calculado no exemplo 1, na malha 0,5m x 0,5m.

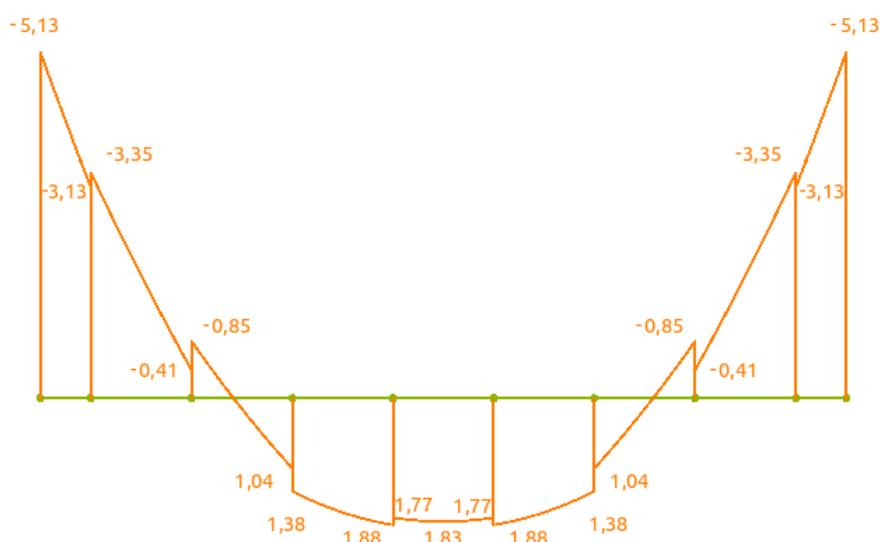
Figura 63 – Deformada da viga central da grelha do exemplo 2 de malha 0,50x0,50.



Fonte: Autor.

A Figura 64 mostra os momentos fletores obtidos na viga central da grelha (ilustrada na Figura 40), onde foram obtidos para momento fletor negativo máximo -10,26 kNm/m e positivo 3,76 kNm/m.

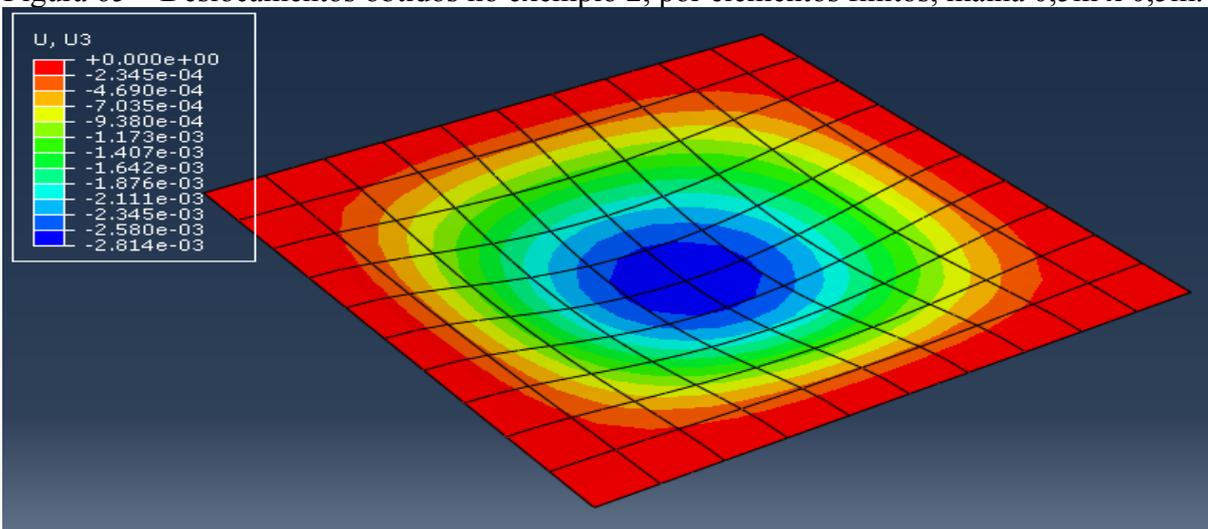
Figura 64 – Momento fletor da viga central da grelha do exemplo 2 de malha 0,50x0,50.



Fonte: Autor

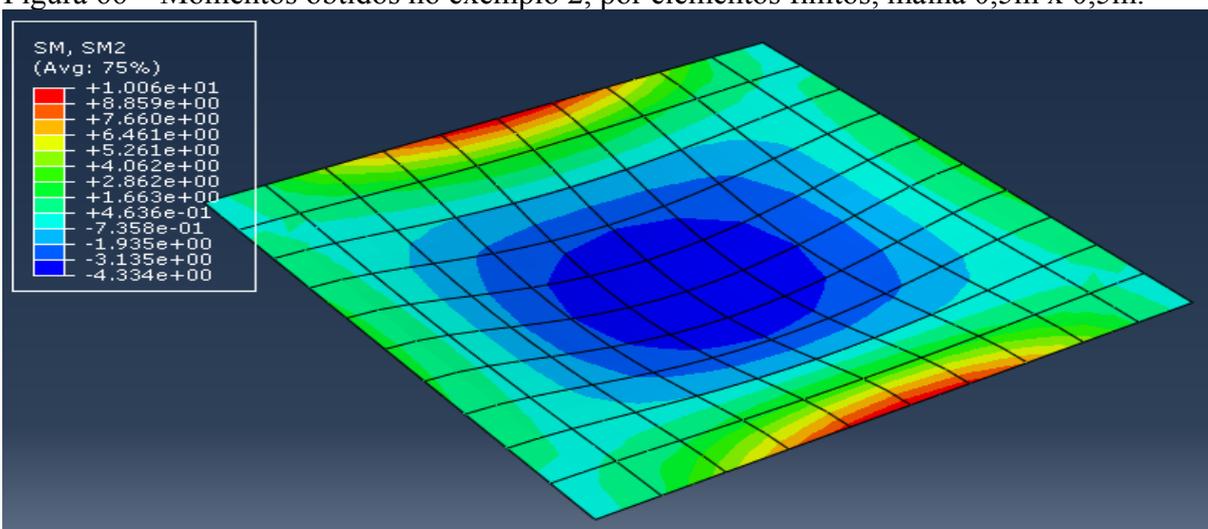
A Figura 65 e Figura 66 mostram os resultados obtidos por elementos finitos de placa, da malha utilizada.

Figura 65 – Deslocamentos obtidos no exemplo 2, por elementos finitos, malha 0,5m x 0,5m.



Fonte: Autor.

Figura 66 – Momentos obtidos no exemplo 2, por elementos finitos, malha 0,5m x 0,5m.



Fonte: Autor.

### c) Malha 0,25m x 0,25m

A malha do exemplo 2 com espaçamento de 0,25 m e pode ser visualizada na Figura 52. A Figura 67 mostra a deformada da décima primeira viga da grelha (ver Figura 53), onde ocorrem os momentos e flecha máxima da laje. O carregamento dos elementos é igual ao calculado no exemplo 1, na malha 0,25 m x 0,25 m.

Figura 67 – Deformada da viga central da grelha do exemplo 2 de malha 0,25x0,25.

Fonte: Autor.

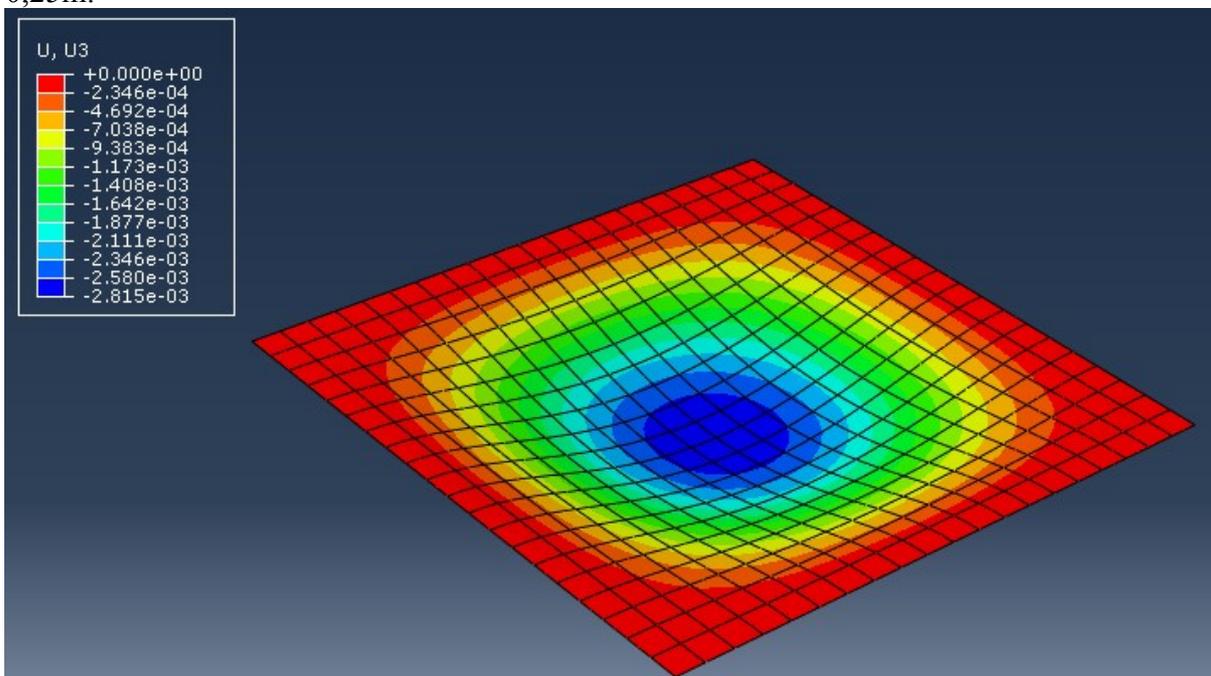
A Figura 68 mostra os momentos fletores obtidos na viga central da grelha, onde foram obtidos para o momento fletor negativo máximo de -10,40 kNm/m e momento positivo máximo 3,72 kNm/m.

Figura 68 – Momento fletor da viga central da grelha do exemplo 2 de malha 0,25x0,25.

Fonte: Autor

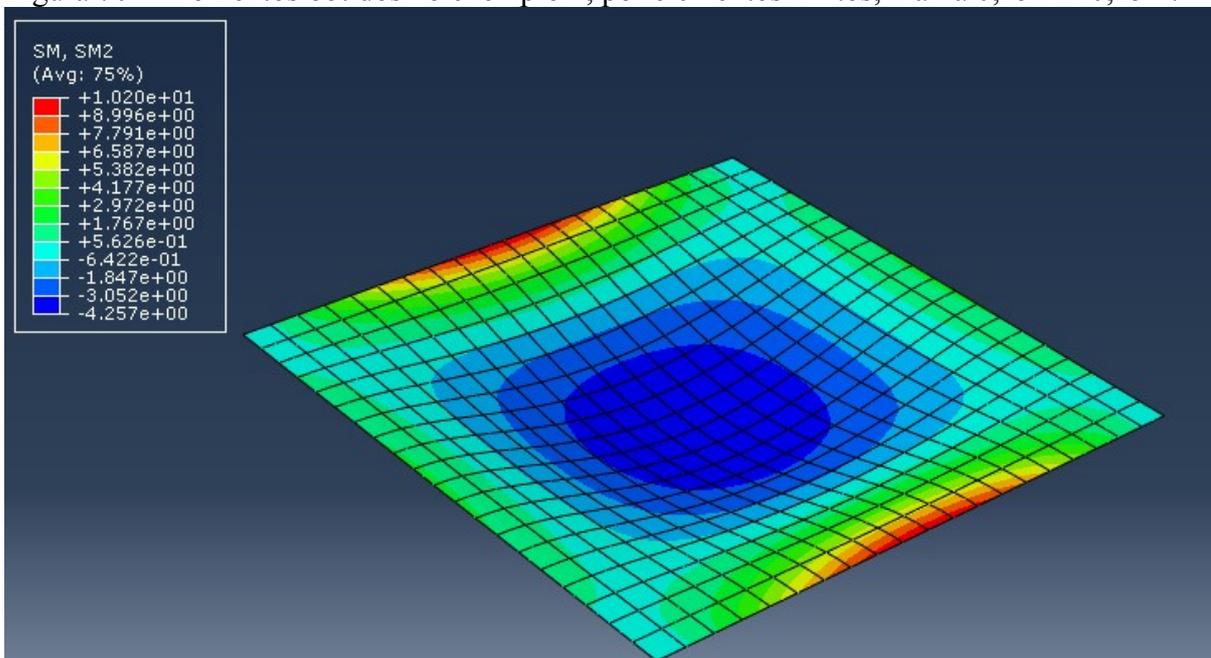
A Figura 69 e Figura 70 mostram os resultados obtidos por elementos finitos de placa, da malha utilizada.

Figura 69 – Deslocamentos obtidos no exemplo 2, por elementos finitos, malha 0,25m x 0,25m.



Fonte: Autor.

Figura 70 – Momentos obtidos no exemplo 2, por elementos finitos, malha 0,25m x 0,25m.



Fonte: Autor.

As propriedades utilizadas nos elementos de cada análise, referente ao exemplo 2, estão presentes na Tabela 6. A Tabela 8 mostra os resultados obtidos nas análises realizadas no exemplo 2, comprando os momentos e flechas obtidas de cada malha com os resultados analíticos da Teoria das Placas.

Tabela 8 – Resultados obtidos no exemplo 2.

	Momento positivo máximo (kNm/m)	Diferença do momento positivo (%)	Momento negativo (kNm/m)	Diferença do momento negativo (%)	Flecha máxima (mm)	Diferença da flecha (%)
Malha 1x1	3,97	-6,00	-10,47	-1,55	3,10	+10,71
Malha 0,5x0,5	3,76	-11,0	-10,26	+0,48	2,95	+5,36
Malha 0,25x0,25	3,72	-12,0	-10,40	-0,87	3,01	+7,5
Elementos Finitos de Placa (1x1)	4,30	+1,65	-8,80	+14,65	2,51	-10,36
Elementos Finitos de Placa (0,5x0,5)	4,33	+2,36	-10,01	+2,91	2,81	0,36
Elementos Finitos de Placa (0,25x0,25)	4,26	+0,70	-10,02	+2,81	2,82	+0,74
Teoria das Placas	4,23		-10,31		2,80	

Fonte: Autor.

Da mesma forma que foi observado no exemplo 1, o momento fletor positivo obtido por analogia de grelha apresentaram valores menores que os obtidos pela Teoria das Placas. Os momentos negativos e flechas apresentaram valores maiores analíticos, sendo observado em ambos uma não convergência dos resultados, entre as malhas de 0,5m x 0,5m e 0,25m x 0,25m. Também foi observada a presença de descontinuidades nos gráficos de momento fletor, devido a rigidez a torção dos elementos. Os resultados de elementos finitos de placa convergiram para solução analítica.

## 5. CONCLUSÃO

O presente trabalho teve com objetivo o desenvolvimento de uma ferramenta computacional capaz de gerar, através de uma interface gráfica bem elaborada, problemas de engenharia estrutural com elementos de grelha. Também foram realizadas contribuições para um analisador estrutural acadêmico desenvolvido na Universidade Federal do Ceará, onde foi adicionado funcionalidades específicas para análise de grelhas.

Os exemplos de verificação mostraram que foi realizada uma correta aplicação das formulações e algoritmos desenvolvidos para o analisador, pre-processador e pós-processador. Os exemplos apresentaram resultados iguais aos encontrados na literatura, em termos de valores dos esforços e deslocamentos, e também da visualização tridimensional dos resultados.

Os resultados obtidos nos exemplos de analogia de grelha mostram que para lajes maciças, o método pode fornecer resultados contra segurança de momentos positivos e negativos, enquanto que a flecha permaneceu acima dos valores obtidos pela solução analítica. Este mesmo comportamento foi observado por STRAMANDINOLI (2003) em exemplos semelhantes ao realizados no presente trabalho. O modelo de elementos finitos de placa apresentou convergência nos dois exemplos.

A utilização do método de analogia de grelha pela maior parte dos projetistas de estruturas torna evidente a necessidade de estudo deste método nas universidades. Por isso é de fundamental importância que ferramentas semelhantes ao programa desenvolvido neste trabalho sejam implementadas, auxiliando o aprendizado de métodos computacionais de engenharia. O programa desenvolvido apresenta funcionalidades úteis para a aplicação deste método de análise de lajes, porém são necessárias mais funções para modelagem de pavimentos de edificações, considerando por exemplo malhas de geometria complexa.

## REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6118/2007 – Projeto de estrutura de concreto – Procedimento. Rio de Janeiro, 2007.

HAMBLY, E.C. **Bridge deck behavior**. London, Chapman and Hall, 1976.

Mansour, I. H. PUCRS. Introdução à OpenGL. Disponível em: <<http://www.inf.pucrs.br/~manssour/OpenGL/Introducao.html>>. Acesso em: 20 janeiro 2013.

MARTHA, L. F. **Ftool - Two-Dimensional Frame Analysis Tool**. 2002. Patente: Programa de Computador. Número do registro: INPI 98002876, título: "Ftool - Two-Dimensional Frame Analysis Tool" .

MARTHA, Luiz Fernando. **Análise de estruturas: Conceitos e Métodos Básicos**. ed. Rio de Janeiro: Elsevier, 2010. 524 p.

McGuire, W., Gallagher, R. H. and Ziemian, 2000, **Matrix Structural Analysis**, 2nd edition, John Wiley, New York.

MONTOYA, J.; MESENGUER, A.G.; CABRE, F. M. Hormigon Armado, Editorial Gustavo Gili, S.A. Barcelona, 1973.

SORIANO, H. L.; LIMA, S. S. . **Análise de Estruturas – Método das Forças e Método dos Deslocamentos**. Segunda Edição Atualizada. ed.Rio de Janeiro: Ciência Moderna, 2006. v. 1. 322p .

STRAMANDINOLI, J. S. B. **CONTRIBUIÇÕES À ANÁLISE DE LAJES NERVURADAS POR ANALOGIA DE GRELHA**. 2003. 179 f. Dissertação (Mestrado em Engenharia Civil) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

Süssekind, J. C., **Curso de Análise de Estruturas**. ed.São Paulo: Globo, 1977. v. 2. 293p.

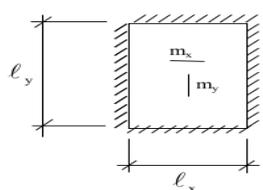
## ANEXOS

## ANEXO A – TABELAS DE CZERNY

**Tabelas de Czerny**

TABELA 1 - TIPO 1  
Laje com as 4 bordas livremente apoiadas  
(carga uniforme)

$l_y/l_x$	$\alpha_x$	$\alpha_y$	$\beta_x$	$\beta_y$	$\alpha_2$
1,00	22,7	22,7			21,4
1,05	20,8	22,5			19,4
1,10	19,3	22,3			17,8
1,15	18,1	22,3			16,5
1,20	16,9	22,3			15,4
1,25	15,9	22,4			14,3
1,30	15,2	22,7			13,6
1,35	14,4	22,9			12,9
1,40	13,8	23,1			12,3
1,45	13,2	23,3			11,7
1,50	12,7	23,5			11,2
1,55	12,3	23,5			10,8
1,60	11,9	23,5			10,4
1,65	11,5	23,5			10,1
1,70	11,2	23,5			9,8
1,75	10,8	23,5			9,5
1,80	10,7	23,5			9,3
1,85	10,4	23,5			9,1
1,90	10,2	23,5			8,9
1,95	10,1	23,5			8,7
2,00	9,9	23,5			8,6
>2	8,0	23,5			6,7



$$m_x = \frac{p\ell_x^2}{\alpha_x}$$

$$m_y = \frac{p\ell_y^2}{\alpha_y}$$

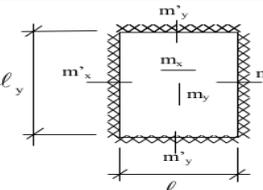
$$w_{\max} = \frac{p\ell_x^4}{Eh^3\alpha_2}$$

$$\nu = 0,2$$

**Beton-Kalender (1976)**

TABELA 2 - TIPO 6  
Laje com as 4 bordas engastadas  
(carga uniforme)

$l_y/l_x$	$\alpha_x$	$\alpha_y$	$\beta_x$	$\beta_y$	$\alpha_2$
1,00	47,3	47,3	19,4	19,4	68,5
1,05	43,1	47,3	18,2	18,8	62,4
1,10	40,0	47,8	17,1	18,4	57,6
1,15	37,3	48,3	16,3	18,1	53,4
1,20	35,2	49,3	15,5	17,9	50,3
1,25	33,4	50,5	14,9	17,7	47,6
1,30	31,8	51,7	14,5	17,6	45,3
1,35	30,7	53,3	14,0	17,5	43,4
1,40	29,6	54,8	13,7	17,5	42,0
1,45	28,6	56,4	13,4	17,5	40,5
1,50	27,8	57,3	13,2	17,5	39,5
1,55	27,2	57,6	13,0	17,5	38,4
1,60	26,6	57,8	12,8	17,5	37,6
1,65	26,1	57,9	12,7	17,5	36,9
1,70	25,5	57,8	12,5	17,5	36,3
1,75	25,1	57,7	12,4	17,5	35,8
1,80	24,8	57,6	12,3	17,5	35,4
1,85	24,5	57,5	12,2	17,5	35,1
1,90	24,2	57,4	12,1	17,5	34,7
1,95	24,0	57,2	12,0	17,5	34,5
2,00	24,0	57,1	12,0	17,5	34,3
>2	24,0	57,0	12,0	17,5	32,0



$$m_x = \frac{p\ell_x^2}{\alpha_x}$$

$$m_y = \frac{p\ell_y^2}{\alpha_y}$$

$$m'_x = -\frac{p\ell_x^2}{\beta_x}$$

$$m'_y = -\frac{p\ell_y^2}{\beta_y}$$

$$w_{\max} = \frac{p\ell_x^4}{Eh^3\alpha_2}$$

$$\nu = 0,2$$

**Beton-Kalender (1976)**