

SOFTWARE ACADÊMICO PARA RESOLUÇÃO DE VIGA GENÉRICA PELO MÉTODO DOS ELEMENTOS FINITOS

Luan Henrique Sirtoli – luanhs@ucl.br
Msc. Jean Carlo Ferreira de Oliveira – jean@ucl.br

Faculdade do Centro Leste, Rod. ES – 010, km 6, S/N, Manguinhos, Serra - ES

Resumo

O presente artigo tem o objetivo de estudar o Método dos Elementos Finitos (MEF) com propósito de desenvolver um software para cálculo de vigas unidimensionais. Este software deverá ser didático, prático e open-source. O artigo tem início fundamentando a correta modelagem da estrutura reticulada, e é estudado o elemento mola para dar base ao MEF. Assim, são desenvolvidas as equações contendo as matrizes de rigidez local e global para encontrar as deflexões e esforços na viga. A função de deslocamento é definida a partir de uma equação cúbica com quatro coeficientes, e suas derivadas definem as funções de inclinação, momento fletor e esforço cortante. Assim, o MEF é implementado em uma página utilizando as linguagens de programação HTML5, CSS3 e JavaScript, e com o auxílio das bibliotecas Bootstrap, JQuery, FabricJS, MathJS, MathJax e HighCharts. Após a implementação, são propostos três problemas a serem solucionados de forma analítica e pelo software, comparando e analisando os dados calculados. Por fim, é discorrido se o software atingiu a finalidade proposta, avaliando os trabalhos futuros possíveis.

Palavras-chaves: Método dos Elementos Finitos, Resistência dos Materiais, Viga.

1. INTRODUÇÃO

Durante sua formação acadêmica, muitas vezes o aluno de engenharia se depara com problemas analíticos de difícil solução, os quais necessitam de uma parte matemática mais robusta. Nem sempre o aluno tem prontamente à sua disposição um software que auxilie na resolução, ou que possa validar os cálculos previamente feitos. Esse é um problema que pode ser ocasional ao calcular as deflexões, inclinações, momentos fletores e esforços cortantes dos problemas de Vigas e Pórticos, se o mesmo estiver cursando as disciplinas de Resistência dos Materiais I e II.

Portanto, esse trabalho se baseia em fornecer um software didático, prático e open-source, que possa verificar a solução analítica de vigas unidimensionais que foram realizados pelos alunos de engenharia.

2. METODOLOGIA

Como o cálculo de estruturas é uma parte refinada e delicada da engenharia, é necessária uma sólida base em Resistência dos Materiais para dar ao nosso projeto a robustez e seriedade necessária para cálculos analíticos complexos. “[...] na área de Cálculo Estrutural, o engenheiro deve garantir que a estrutura objeto de análise não estará sujeita a falhas sob as diversas condições de operação.” [1].

Para esse projeto, será utilizada programação Web, a fim de que as simulações possam ser executadas em qualquer tipo de dispositivo por meio de um simples navegador, tornando assim, o software mais prático e simples.

Com as linguagens voltadas à construção de páginas Web que serão estudadas na Seção 2.2, pode-se construir uma página modular leve, que se adaptará ao dispositivo onde será executado. Assim, o software poderá montar uma prévia do problema proposto rapidamente e será capaz de interagir com outras plataformas, por exemplo, a plataforma ESO (Ensino Superior Online).

2.1. O Método dos Elementos Finitos (MEF)

O MEF se baseia na simplificação de problemas complexos que muitas vezes seriam impossíveis de calcular por métodos simples de Resistência dos Materiais. Desse modo, vigas, pórticos ou treliças são transformados em elementos menores e mais simples ligados por nós, possibilitando assim os cálculos. [1]

2.1.1. Modelagem da Estrutura

Para a resolução de um problema real de estrutura, é necessário fazer uma análise apurada da questão a ser resolvida, simplificando o problema real de modo que seja capaz de ser modelado dentro dos padrões da análise de estruturas.[1] “[...] ao se iniciar um processo de cálculo da estrutura de objeto de análise, o engenheiro deve formular um **Esquema de Cálculo** para essa estrutura ou, em outras palavras, um **Modelo de Cálculo**, em que a **Estrutura é Idealizada de sorte que se possa analisa-la.**” [1]. Na Figura 1, temos um exemplo de uma idealização com uma viga bi apoiada, pois ambos os vínculos permitem rotação nas extremidades.

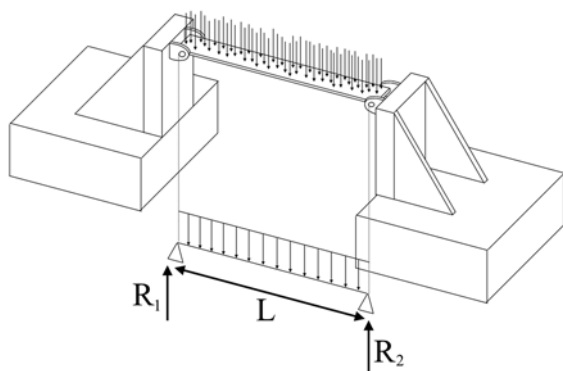


Figura 1. Idealização de um problema real de Engenharia [próprio]

Um modelo idealizado como o da Figura 1 pode ser resolvido com certa facilidade, pois o mesmo pode ser solucionado a partir das equações de Equilíbrio. [2]

“O equilíbrio de um corpo exige um equilíbrio de forças, para impedir a translação ou um movimento acelerado do corpo ao longo de uma trajetória reta ou curva, e um equilíbrio de momentos, para impedir que o corpo gire. Essas condições podem ser expressas matematicamente pelas duas equações vetoriais. Nessas fórmulas, $\sum F$ representa a soma de todas as forças que agem sobre o corpo, e $\sum M_0$ é a soma dos momentos de todas as forças em torno de qualquer ponto O dentro ou fora do corpo.” [2]

$$\sum F = 0; (1)$$

$$\sum M = 0; (2)$$

Como as idealizações para o projeto são unidimensionais, essas equações serão aplicadas para o eixo Y. [2]

$$\sum F_y = 0; (3)$$

$$\sum M_o = 0; (4)$$

Entretanto, modelando estruturas mais complexas, as equações de equilíbrio não bastarão mais para a resolução do problema, como indica o Grau de Hiperestaticidade. Esse grau indica o número de reações excedentes ao número de equações possuídas, que, para o caso de uma estrutura unidimensional são duas equações. Caso o grau de hiperestaticidade seja maior que zero, a viga será **hiperestática**. Se a idealização do problema se mostrar hiperestática, uma nova metodologia será necessária para resolver o problema. Para o caso deste trabalho, o **Método dos Elementos Finitos** será adotado. [1]

Para início da resolução do problema pelo MEF, é necessário modelar o sistema de modo diferente.

Assim, a estrutura será modelada como uma **estrutura reticulada**, como exemplificada na Figura 2. [1]

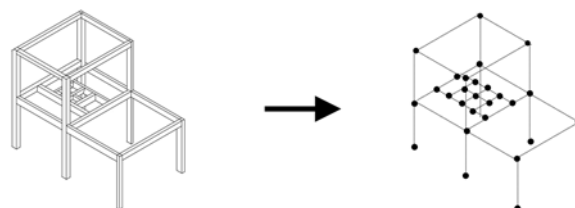


Figura 2. Modelagem de um sistema reticulado [próprio]

Para a modelagem do sistema reticulado de uma viga, é necessário identificar cada nó estrutural dos elementos que sofre solicitações mecânicas. [1]

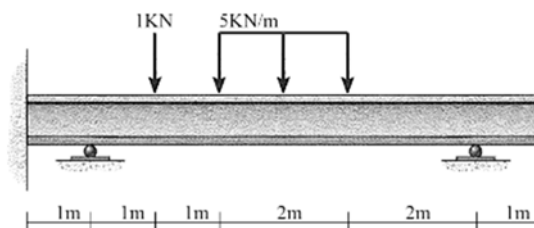


Figura 3. Exemplo de viga a ser calculada [próprio]

Para identificar corretamente os elementos correspondentes à uma estrutura hiperestática, cada força, momento, apoios de primeiro e segundo grau, engaste, ponta livre, e início e fim de carregamento será transformado em nós. Assim, será obtida a estrutura idealizada na Figura 4. [1]

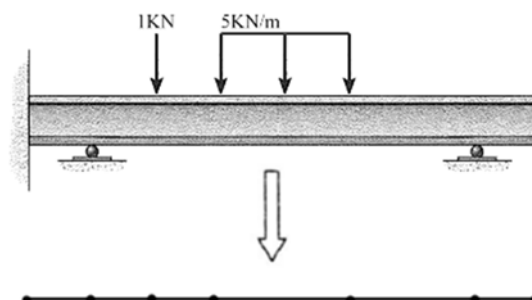


Figura 4. Modelagem de um sistema reticulado de viga [próprio]

Após reduzir a estrutura à uma estrutura reticulada, é necessário aplicar aos nós os esforços que foram aplicados na estrutura. As cargas e momentos aplicados irão ser aplicados diretamente, e a carga distribuída será transformada em forças e momentos equivalentes em cada ponto. [1]

Portanto, a estrutura reticulada final será modelada de tal forma, que somente haverá esforços aplicados sobre os nós, como exemplificada na Figura 5: [1]

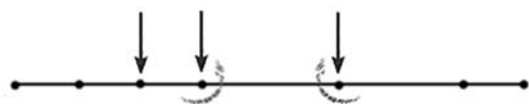


Figura 5. Estrutura reticulada de viga com esforços aplicados [próprio]

2.1.2. Rigidez do Elemento

As cargas aplicadas sobre os nós gerarão esforços nos elementos da viga causando deformações na estrutura reticulada. Devido as propriedades do material os elementos da viga tenderão a se deformar, porém, irão resistir a essa deformação. Esse comportamento se assemelha ao comportamento de um elemento mola, assim, o comportamento de cada estrutura será idealizada como se fosse uma mola. [1] “A Análise Matricial de Estruturas, e, em consequência o Método dos elementos Finitos têm como ponto de partida a **Relação entre Forças Nodais e Deslocamentos Nodais** para cada **Elemento Individual**. Essa ideia fundamental está relacionada ao **Conceito de Rigidez**. Todos nós temos a ideia de rigidez desde as primeiras aplicações com os elementos elásticos (ou Molas) da Física Básica. A constante elástica da mola, que é a medida quantitativa da **Rigidez da mola**, é expressa por intermédio da relação entre a força aplicada e o deslocamento medido na extremidade da mola”. [1] Assim, essa relação é demonstrada na Figura 6:

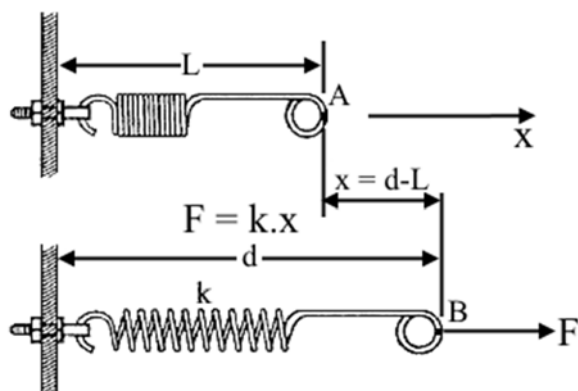


Figura 6. Mola deformada sujeita à esforço [próprio]

A partir do comportamento da mola, a seguinte equação é definida:

$$F = K.X; (5)$$

Ao aplicar o diagrama de corpo livre no elemento mola, obtém-se a configuração demonstrada na Figura 7: [1]

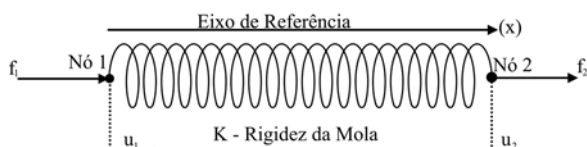


Figura 7. Mola aplicada no Diagrama de Corpo Livre [próprio]

Assim, analisando o diagrama de corpo livre da Figura 7, percebe-se que os dois nós possuem um único grau de liberdade cada, gerando uma matriz de rigidez de 2 colunas e 2 linhas.

Com essas referências aplicadas na Equação 5, a seguinte equação matricial é obtida: [1]

$$\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}; (6)$$

“[...] os termos da matriz de rigidez do elemento finito representam relações de causa e efeito. A causa é um deslocamento Unitário imposto no nó, e os efeitos são as forças que surgem nos nós do elemento devido a esse deslocamento” [1]. Ao saber a definição física dos termos contidos na matriz, pode-se generalizar o significado para qualquer elemento finito. [1]

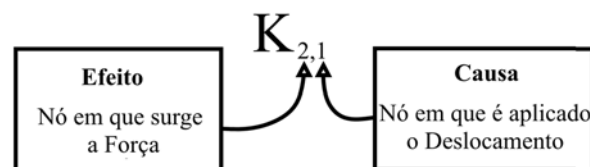


Figura 8. Efeito e causa no coeficiente de rigidez [próprio]

Como uma viga se comporta da mesma maneira que um elemento mola, é possível utilizar a Equação 6 para definir as deformações de um elemento de viga. [1]

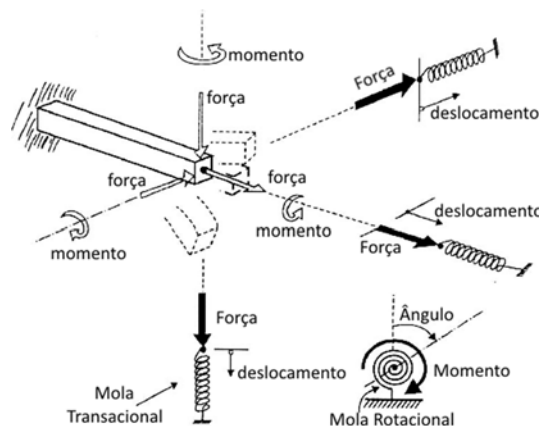


Figura 9. Barra sujeita à esforços, comparada à mola [1]

Como o trabalho proposto se baseia em uma viga exposta à deslocamento em Y e em inclinações angulares, cada nó terá 2 graus de liberdade, como demonstrada na Figura 10. Assim, essa viga possuirá uma matriz de rigidez local de 4 colunas e 4 linhas. [1]



Figura 10. Graus de liberdade representados em um elemento de viga [Próprio]

A equação final considerando as forças e momentos nodais, e deslocamentos e inclinações será: [1]

$$\begin{Bmatrix} f_1 \\ M_1 \\ f_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ \theta_1 \\ u_2 \\ \theta_2 \end{Bmatrix}; \quad (7)$$

Para definir os termos da matriz de rigidez local, é necessário aplicar valores unitários para o deslocamento e a inclinação, a fim de encontrar a força que surge nesse nó. Portanto, como demonstrado nas Figuras 11 e 12, uma deflexão unitária será imposta: [1]

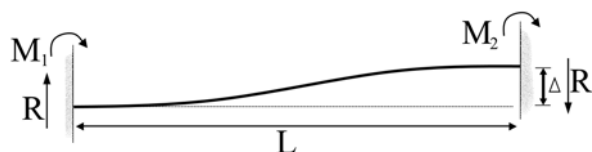


Figura 11. Elemento de viga com deslocamento unitário [Próprio]

$$M_1 = M_2 = \frac{6E.I}{L^2} \cdot x; \quad (8) \quad R = \frac{12E.I}{L^3} \cdot x; \quad (9)$$

Para inclinação unitária: [1]

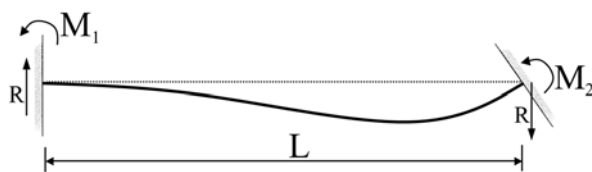


Figura 12. Elemento de viga com inclinação unitária [Próprio]

$$M_1 = \frac{2E.I}{L} \cdot \theta; \quad (10)$$

$$M_2 = \frac{4E.I}{L} \cdot \theta; \quad (11)$$

$$R = \frac{6E.I}{L^2} \cdot \theta; \quad (12)$$

Então, aplicando os valores encontrados na Equação 8: [1]

$$\begin{Bmatrix} f_1 \\ M_1 \\ f_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} \frac{12E.I}{L^3} & \frac{6E.I}{L^2} & -\frac{12E.I}{L^3} & \frac{6E.I}{L^2} \\ \frac{6E.I}{L^2} & \frac{4E.I}{L} & -\frac{6E.I}{L^2} & \frac{2E.I}{L} \\ -\frac{12E.I}{L^3} & -\frac{6E.I}{L^2} & \frac{12E.I}{L^3} & -\frac{6E.I}{L^2} \\ \frac{6E.I}{L^2} & \frac{2E.I}{L} & -\frac{6E.I}{L^2} & \frac{4E.I}{L} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ \theta_1 \\ u_2 \\ \theta_2 \end{Bmatrix}; \quad (13)$$

Ao trabalhar com treliças ou pórticos é necessária a utilização de uma matriz de transformação, pois as estruturas citadas serão bidimensionais e tridimensionais, gerando assim angulações entre os elementos. Porém, como a estrutura a ser generalizada é uma viga, os elementos da mesma sempre estarão alinhados devido sua unidimensionalidade, não necessitando de uma matriz de transformação. [1]

Após a montagem das matrizes de rigidez local para todos os elementos da estrutura reticulada inicial, é gerada a matriz global, fazendo a somatória de todos os valores de “k” coincidentes com os mesmos índices. [1] Como a estrutura trabalhada é unidimensional, os índices iniciais do próximo elemento sempre serão os índices finais do elemento anterior. Isso causará uma “diagonalização” na soma dos coeficientes da matriz global. A matriz de rigidez global terá o comportamento abaixo durante sua montagem. [1]

$$K_{GI} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & 0 & 0 \\ k_{21} & k_{22} & k_{23} & k_{24} & 0 & 0 \\ k_{31} & k_{32} & k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} \\ k_{41} & k_{42} & k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} \\ 0 & 0 & k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix}; \quad (14)$$

Após montada a matriz global, será necessária a inclusão das condições de contorno que representam os apoios aplicados no problema proposto. A forma de incluir as condições de contorno no MEF é através da matriz global reduzida. [1]

Para tal, é necessário zerar a linha correspondente ao grau bloqueado, e adicionar um valor unitário no “k” de iguais índices. [3]

$$K_{GR} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ k_{31} & k_{32} & k_{33}+k_{11} & k_{34}+k_{12} & k_{13} & k_{14} \\ k_{41} & k_{42} & k_{43}+k_{21} & k_{44}+k_{22} & k_{23} & k_{24} \\ 0 & 0 & k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix}; \quad (15)$$

Assim, a Equação 5 se torna:

$$\{F\} = [K]_{GR} \cdot \{X\}; \quad (16)$$

Ao adicionar apoios de primeiro e segundo grau, não haverá deslocamento no respectivo nó, e se houver engaste, não haverá deslocamento nem inclinação. Essa afirmação é comprovada na multiplicação matricial demonstrada nas Equações 17 e 18.

$$f_1 = 0 = 1 \cdot u_1 + 0 \cdot \theta_1 + 0 \cdot u_2 + 0 \cdot \theta_2; \quad (17)$$

$$M_1 = 0 = 0 \cdot u_1 + 1 \cdot \theta_1 + 0 \cdot u_2 + 0 \cdot \theta_2; \quad (18)$$

As Equações 17 e 18 resultam, respectivamente, em:

$$u_1 = 0; \quad (19)$$

$$\theta_1 = 0; \quad (20)$$

Portanto, ao inserir as condições de contorno na matriz de rigidez global, é dito ao MEF quais pontos da viga não irão se deslocar ou inclinar.

Assim, feita a multiplicação matricial da Equação 21, encontram-se os deslocamentos e inclinações nodais.

$$\{X\} = [K]_{GR}^{-1} \cdot \{F\}; \quad (21)$$

Então, para encontrarmos o vetor de forças contendo as reações, é necessário utilizar a Equação 22. [3]

$$\{F\} = [K]_{GL} \cdot \{X\}; \quad (22)$$

Ao possuir todos os deslocamentos, inclinações e esforços nos nós resultantes da Equação 21, é dado início à montagem dos gráficos. Para montagem dos gráficos são necessárias as funções de cada gráfico. Para se desenvolver a função de deslocamento, será utilizada uma função polinomial que possua 4 coeficientes, pois o elemento possui quatro graus de liberdade. A função para esse caso é a **função cúbica**. [1][3]. Portanto, a equação de deslocamento é:

$$v(x) = C_1 + C_2 \cdot x + C_3 \cdot x^2 + C_4 \cdot x^3; \quad (23)$$

Como a função de Inclinação é a derivada primeira da função de deslocamento: [1][3]

$$d(v(x)) = v'(x) = 1 + C_2 + 2 \cdot C_3 \cdot x + 3 \cdot C_4 \cdot x^2 \quad (24)$$

Matricialmente:

$$\begin{Bmatrix} v(x) \\ v'(x) \end{Bmatrix} = \begin{bmatrix} 1 & x & x^2 & x^3 \\ 0 & 1 & 2 \cdot x & 3 \cdot x^2 \end{bmatrix} \cdot \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}; \quad (25)$$

Aplicando a Equação 25 para $x=0$ e $x=L$, que são o primeiro e segundo nó: [1][3]

$$\begin{Bmatrix} v_1 \\ v'_1 \\ v_2 \\ v'_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & L & L^2 & L^3 \\ 0 & 1 & 2 \cdot L & 3 \cdot L^2 \end{bmatrix} \cdot \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}; \quad (26)$$

Ao resolver a Equação matricial 26, encontram-se as quatro constantes, e podendo assim, aplicar nas Equações 27, 28, 29 e 30: [1]

Função de Deslocamento:

$$v(x) = C_1 + C_2 \cdot x + C_3 \cdot x^2 + C_4 \cdot x^3; \quad (27)$$

Função de Inclinação:

$$v^{(x)} = C_2 + 2 \cdot C_3 \cdot x + 3 \cdot C_4 \cdot x^2; \quad (28)$$

Função de Momento Fletor:

$$M(x) = E \cdot I \cdot v''(x) = 2 \cdot C_3 + 6 \cdot C_4 \cdot x; \quad (29)$$

Função de Força Cortante:

$$V(x) = E \cdot I \cdot v'''(x) = 6 \cdot C_4; \quad (30)$$

Tais funções serão repetidas para cada segmento da viga, gerando assim os gráficos necessários. [1]

2.2. Programação Necessária

A partir das premissas definidas na Introdução, são escolhidas linguagens de programação de boa performance. Essas linguagens devem ser de fácil e rápida execução em vários dispositivos, como computadores e smartphones. Então, as linguagens abaixo foram escolhidas.

HTML: É a linguagem padrão de páginas Web. É uma linguagem mundialmente difundida, e de fácil execução por diversos dispositivos como computadores, notebooks, smartphones e tablets. A mesma também é de fácil construção e entendimento, e pode ser construída por um editor de texto simples. Portanto, como o software será um aplicativo Web, o HTML foi escolhido. A versão utilizada será o HTML5. [4]

CSS: Linguagem complementar ao HTML. Criada para desenvolver em um outro documento a parte de estilos separada da estrutura da página HTML. Assim, a página será mais dinâmica, terá um código mais limpo e fornecerá novas possibilidades de complementos para a página. Portanto, como o software precisa ser dinâmico e interativo com o usuário, é necessária a utilização do CSS, sendo utilizada a versão CSS3. [5]

JavaScript: É a linguagem de programação do HTML e Web. O JavaScript modifica o conteúdo de páginas HTML, executa funções, modifica estilos de CSS, pode esconder ou mostrar objetos, atribuir funções à objetos, entre outros. O JavaScript é uma linguagem de scripts dinâmica e eficiente, que pode fazer cálculos matemáticos com facilidade, e com resposta rápida. Portanto, é necessário o uso do JavaScript para os cálculos matriciais, e toda a parte matemática inerente ao software. [6][7]

Mesmo com as linguagens acima definidas, serão necessárias bibliotecas adicionais para o software. Parte dos componentes de cada linguagem é um tanto simples, e necessita ser incrementada para fazer os cálculos necessários e exibir os resultados na página. Serão utilizadas então, as bibliotecas abaixo.

Bootstrap: É uma biblioteca de componentes open-source responsiva desenvolvida pelo Twitter, baseada em HTML, CSS e JS. Fornece diversos componentes para construção de páginas adaptáveis para dispositivos móveis e computadores. Construída em SASS [8] e JQuery [9], a mesma fornece componentes pré-definidos como botões, painéis, barras de navegação, estruturas para montagem de componentes, que facilitam a construção da página. [10]

JQuery: É uma biblioteca compacta baseada em JavaScript, que compacta as funções JS, e reduz grande parte do código. Faz manipulações HTML, controle de eventos, e todas as funções JavaScript com maior simplicidade e versatilidade, com menor quantidade de código. [9]

FabricJS: É uma biblioteca para HTML baseada em JavaScript, que provém objetos interativos para aplicação no componente Canvas do HTML5. Reduz grande parte da programação necessária para se implementar objetos, e facilita a atribuição de atributos aos mesmos. Como o projeto necessita de um pré-processamento e mostrar o problema a ser desenvolvido ao usuário, a função Canvas irá ser utilizada, e para facilitar a adição e manipulação dos objetos no Canvas, o FabricJS será necessário. [11]

MathJS: Biblioteca mais importante a ser utilizada. É uma biblioteca complementar ao JavaScript para cálculos matemáticos de livre distribuição, flexível e eficiente. Suporta cálculos com grandes números, números complexos, cálculos matriciais, métodos de cálculos pré-programados, e é compatível com a biblioteca matemática já existente no JavaScript. Para o projeto, serão feitos muitos cálculos complexos que necessitam de resultados rápidos, portanto a MathJS se mostra de grande importância para o projeto. [12]

MathJax: É uma biblioteca baseada em JavaScript e CSS para exibição de tipografias matemáticas de alta qualidade. Essencial para a saída de dados do software, exibindo matrizes, equações e funções matemáticas. Essa biblioteca será utilizada devido a necessidade de exibição dos valores calculados após a resolução do problema proposto. [13]

HighCharts: Biblioteca extensa de gráficos e tabelas com base em JavaScript. Será utilizada para exibir os gráficos finais do problema. [14]

Então, com as linguagens e bibliotecas citadas acima, tem-se as ferramentas necessárias ao desenvolvimento de um software que atenderá as premissas e objetivos definidos na Introdução.

3. IMPLEMENTAÇÃO

A partir dos critérios definidos na Introdução e da

metodologia delineada, será construído o software. A página inicial é construída em HTML com componentes do Bootstrap, e conterá quatro painéis. Dois painéis terão a função de manipular o problema, exemplificados na Figura 13. Um dos painéis conterá a pré-visualização do problema proposto exemplificado na Figura 14. E o último painel conterá os resultados obtidos após a simulação, exemplificados nas Figuras 16 e 17. [4][10]

Para a manipulação do problema, com utilização dos componentes do Bootstrap foram criados botões dentro dos painéis, que representam os apoios e os esforços. Ao serem acionados, irão acrescentar aos seus respectivos painéis os campos para preenchimento dos dados, sendo necessário a inserção da posição para os apoios, e a posição e intensidade para os esforços. [10]



Figura 13. Painéis de apoios e esforços. [Próprio]

Para definição da viga a ser simulada, também foram designados campos para a inserção dos dados referentes à geometria da viga e das propriedades do material. Esses campos irão solicitar o momento de inércia em metros à quarta potência, o comprimento total em metros e o módulo de elasticidade em Giga Pascal. Os campos acima foram criados de modo a evitar erros de inserção, bloqueando a entrada de letras, com exceção de notação científica. [1][2][4][10]

Para a pré-visualização do problema foi desenvolvido um painel contendo um Canvas com base no FabricJS onde podem ser inseridos diversos objetos. Ao inicializar o software o painel carrega automaticamente a imagem de uma viga, que auxiliará na montagem do problema proposto. [11]

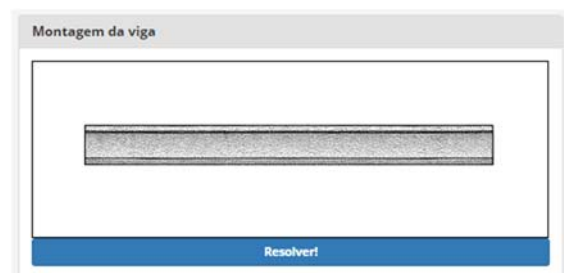


Figura 14. Painel com viga padrão a ser inserida. [próprio]

Após a inserção dos dados do problema, o painel da Figura 14 irá exibir uma prévia do problema a ser calculado. Após o início da resolução do problema, os vetores (arrays) necessários serão preenchidos por funções do JavaScript. Esses vetores conterão as posições dos apoios, posições e intensidades dos

esforços e as posições de cada nó. Com os dados obtidos, o software irá utilizar o MEF desenvolvido no Capítulo 2 do presente artigo em conjunto com a biblioteca MathJS e dar início aos cálculos. Assim, encontrará as matrizes de deslocamentos e inclinações, e a matriz de forças e reações. Irá calcular a matriz de coeficientes e exibir os valores calculados no último painel. [6][7]

O último painel foi desenvolvido de modo a somente exibir valores após a inserção dos dados e validação dos mesmos. Assim, o tempo de processamento é reduzido deixando a página mais limpa e evitando erros de cálculos desnecessários. [4]

Portanto, após o cálculo dos dados, o software utilizará as bibliotecas MathJax e HighCharts para exibir a resolução do problema proposto. Um texto explicativo contendo as funções aplicadas para resolução do problema e as matrizes do MEF será exibido. Essas matrizes conterão os deslocamentos e inclinações nos pontos descritos, os esforços aplicados e reações em cada nó. Por fim serão exibidos os gráficos de deslocamentos, inclinações, momento fletor e força cortante. [13]

A lógica do software foi desenvolvida de modo a evitar erros de projeto, assim, ao ser inserida uma viga que possui grau de hiperestaticidade abaixo de 2, ou seja, número de apoios insuficiente, a mesma irá retornar uma mensagem de erro. [6][7]

Para dar ao software funcionalidade, funções em JavaScript foram desenvolvidas. As mesmas são a parte lógica do programa e dão dinamismo aos botões, assim, adicionando componentes ao problema. Essas funções farão o preenchimento das arrays necessárias, farão a modelagem da estrutura reticulada e todos os cálculos matriciais que serão exibidos ao usuário.

4. DOCUMENTAÇÃO DO SOFTWARE

Ao finalizar a implementação do software, é necessário documentar o projeto. Assim, é necessário descrever todas as funções JavaScript, as variáveis utilizadas e disponibilizar os códigos fonte HTML, CSS e JS. Como o software desenvolvido é open-source, toda essa documentação será feita no GitHub [15]. O GitHub é a maior plataforma de hospedagem de códigos-fonte do mundo atualmente, hospedando códigos de grandes empresas renomadas da tecnologia. Assim, o GitHub será o repositório aberto do projeto, e toda a documentação do software será feita na página oficial do projeto no GitHub. [15][16]

5. RESULTADOS

Após a finalizar a construção do software, é necessário verificar a eficiência dos cálculos do mesmo. Portanto, para tal, será verificado analiticamente o Método dos Elementos Finitos aplicado à problemas isostáticos e hiperestáticos, e comparar com os cálculos feitos pelo software. Os problemas propostos a seguir foram definidos: [1][2]

5.1. Problema Proposto 1

Para o primeiro problema, é proposta a simulação de uma viga isostática simples de comprimento igual à 10m com módulo de elasticidade de $E=210\text{GPa}$ e momento de inércia de $I=100^{-6}\text{m}^4$, engastada em $x=0\text{m}$, com duas cargas aplicadas, sendo uma de -20KN aplicada em $x=2\text{m}$, uma de 10KN em $x=4\text{m}$ um momento de 15KN.m aplicada em $x=8\text{m}$.

5.1.1. Simulação da Estrutura

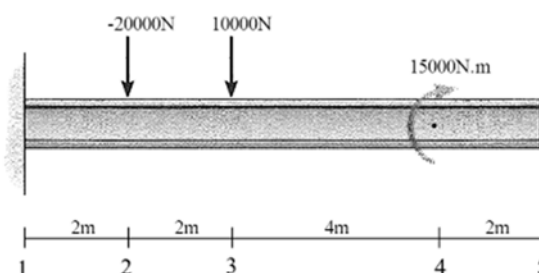


Figura 15. Viga a ser simulada no software. [próprio]

Após a simulação da estrutura proposta na Figura 15, obtemos os seguintes resultados no software:

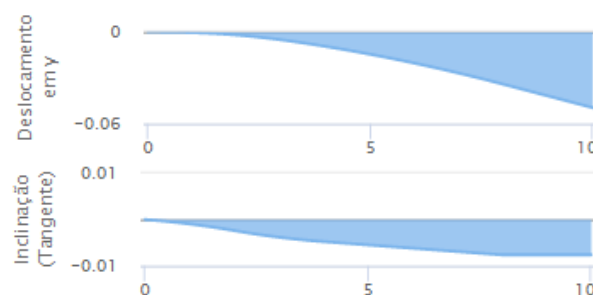


Figura 16. Gráficos de deslocamento e de inclinações desenvolvidos pelo software. [próprio]

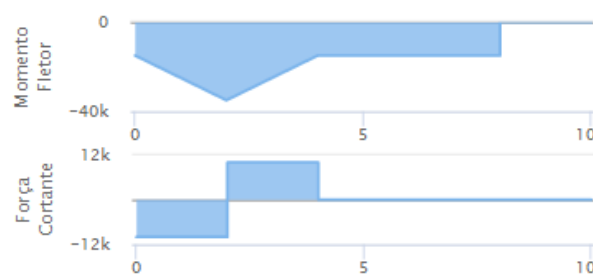


Figura 17. Gráficos de momento fletor e esforço cortante desenvolvidos pelo software. [próprio]

5.1.2. Comparação e Análise dos Resultados do Problema 1

Ao desenvolver o problema analiticamente, os dados dos Quadros 1 e 2 são obtidos. Assim, comparando com as informações obtidas no software, encontra-se o erro residual.

Quadro 1. Deflexões do primeiro problema [próprio].

GL	ANALÍT.	SOFTWARE	ERRO
U1 (m)	0.00	8.38e-16	∞
$\Theta 1$ (rad)	0.00	-3.02e-16	∞
U2 (m)	2.06e-3	2.063e-3	0.1456%
$\Theta 2$ (rad)	2.4e-3	2.381e-3	-0.7917%
U3 (m)	9.52e-3	9.524e-3	0.0420%
$\Theta 3$ (rad)	4.76e-3	4.762e-3	0.0420%
U4 (m)	3.429e-2	3.4286e-2	-0.0117%
$\Theta 4$ (rad)	7.62e-3	7.619e-3	-0.0131%
U5 (m)	4.9524e-2	4.9524e-2	0.0000%
$\Theta 5$ (rad)	7.62e-3	7.619e-3	-0.0131%

Quadro 2. Esforços do primeiro problema [próprio].

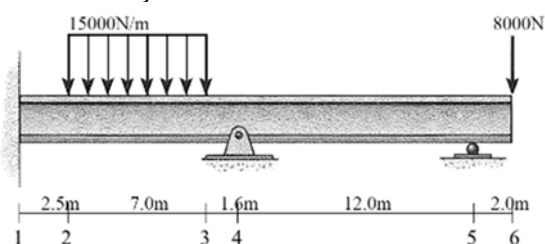
GL	ANALÍT.	SOFTWARE	ERRO
F1 (N)	1.000e4	0,999e4	0.0000%
M1(Nm)	-1.50e4	-1.50e4	0.0000%
F2 (N)	-2.00e4	-2.00e4	0.0000%
M2(Nm)	0.00	0.00	∞
F3 (N)	1.00e4	1.00e4	0.0000%
M3(Nm)	0.00	-5.82e-11	$-\infty$
F4 (N)	0.00	0.00	∞
M4(Nm)	1.50e4	1.50e4	0.0000%
F5 (N)	0.00	0.00	∞
M5(Nm)	0.00	0.00	∞

Analisando os resultados obtidos no Problema 1, nota-se que a margem de erro é muito pequena. Esses erros são provenientes do erro de cálculo da linguagem. Os erros obtidos com valores infinitos são devido o valor analítico ser igual à zero. Como qualquer divisão por zero tende ao infinito, esse erro é justificado. Esse erro não representa um problema na análise, pois o valor real é muito menor que os outros valores reais. Assim, como a variação de valores é muito pequena, o resultado do software reflete os cálculos reais do MEF.

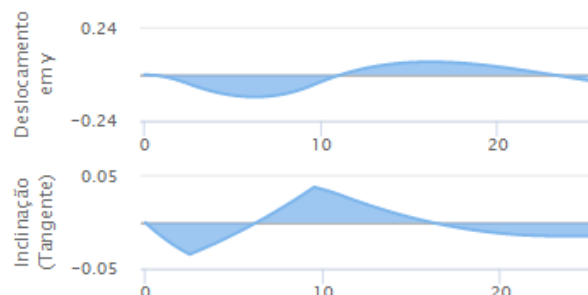
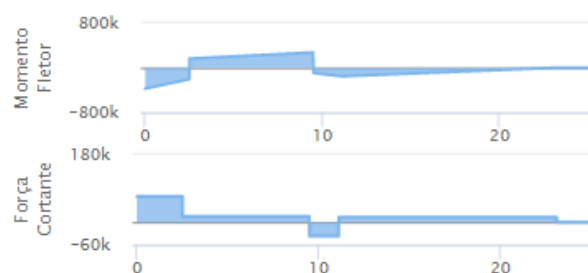
5.2. Problema Proposto 2

Para o segundo problema, é proposta a simulação de uma viga hiperestática de comprimento igual à 25,1m com módulo de elasticidade de $E=210\text{GPa}$ e momento de inércia de $I=100^{-6}\text{ m}^4$, engastada em $x=0\text{m}$, com dois apoios, um de primeiro grau em $x=23.1\text{m}$ e um de segundo grau em $x=11,1\text{m}$, com uma carga distribuída de -15KN/m aplicada de $x=2,5\text{m}$ até $x=9,5\text{m}$ e uma carga pontual de 8KN em $x=25,1\text{m}$.

5.2.1. Simulação da Estrutura

**Figura 18.** Segunda viga simulada no software. [próprio]

Após a simulação da estrutura proposta, obtemos os seguintes resultados no software:

**Figura 19.** Gráficos de deslocamento e de inclinações desenvolvidos pelo software. [próprio]**Figura 20.** Gráficos de momento fletor e esforço cortante desenvolvidos pelo software. [próprio]

5.2.2. Comparação e Análise dos Resultados do Problema 2

Ao desenvolver o problema analiticamente, os dados dos Quadros 3 e 4 são obtidos. Assim, comparando com as informações obtidas no software, encontra-se o erro residual.

Quadro 3. Deflexões do segundo problema [próprio].

GL	ANALÍT.	SOFTWARE	ERRO
U1 (m)	0.00	-9.31e-15	$-\infty$
$\Theta 1$ (rad)	0.000	8.575e-16	∞
U2 (m)	4.754e-2	4.726e-2	-0.5890%
$\Theta 2$ (rad)	3.462e-2	3.443e-2	-0.5488%
U3 (m)	5.534e-2	5.461e-2	-1.3191%
$\Theta 3$ (rad)	-3.876e-2	-3.845e-2	-0.7998%
U4 (m)	0.00	9.98e-17	∞
$\Theta 4$ (rad)	-2.969e-2	-2.907e-2	-2.0882%
U5 (m)	0.000	0.000	∞
$\Theta 5$ (rad)	1.713e-2	1.653e-2	-3.5026%
U6 (m)	3.528e-2	3.506e-2	-0.6236%
$\Theta 6$ (rad)	1.789e-2	1.753e-2	-2.0123%

Quadro 4. Esforços do segundo problema [próprio].

GL	ANALÍT.	SOFTWARE	ERRO
F1 (N)	-68874.0	-68873.84	-0.0002%
M1(Nm)	-376880.0	-376875.56	-0.0012%
F2 (N)	52500.00	52500.00	0.0000%
M2(Nm)	367500.0	367499.99	0.0000%
F3 (N)	52500.00	52500.00	0.0000%
M3(Nm)	-367500.0	-367500.00	0.0000%

F4 (N)	-47116.00	-47115.83	-0.0004%
M4(Nm)	0.00	2.69e-10	∞
F5 (N)	2989.70	2989.67	-0.0010%
M5(Nm)	0.00	5.82e-11	∞
F6 (N)	8000.00	8000.00	0.0000%
M6(Nm)	0.00	-1.16e-10	$-\infty$

Analisando os resultados obtidos no Problema 2, nota-se que a margem de erro é pequena. Esses erros são provenientes do erro de cálculo da linguagem. Assim como no primeiro problema, existem erros de valores iguais a infinito. Da mesma forma, esses valores serão irrelevantes ao problema. Também, no intervalo dos nós 2 e 3 a função analítica esperada é uma função polinomial de quarto grau, devido ao carregamento distribuído. Nesse mesmo intervalo, o software utiliza uma função cúbica, representada na Equação 23. Assim, mesmo com uma pequena variação de valores, o resultado do software reflete a realidade do cálculo do MEF.

5.3. Problema Proposto 3

Para o terceiro problema, é proposta a simulação de uma viga hiperestática de comprimento igual à 12m com módulo de elasticidade de $E=210\text{GPa}$ e momento de inércia de $I=100^{-6}\text{m}^4$, bi engastada em $x=0\text{m}$ e $x=12\text{m}$, com um apoio de primeiro grau em $x=5.5\text{m}$, com uma carga distribuída triangular de início 0N/m e fim de 10000N/m aplicada de $x=7\text{m}$ até $x=9\text{m}$ e um momento de 20000N em $x=3\text{m}$.

5.3.1. Simulação da Estrutura

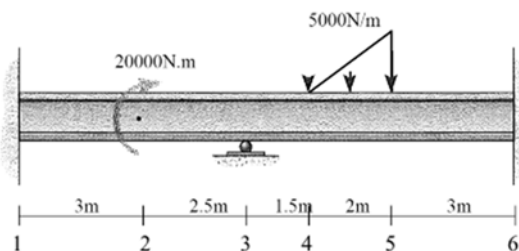


Figura 21. Terceira viga a ser simulada no software. [próprio]

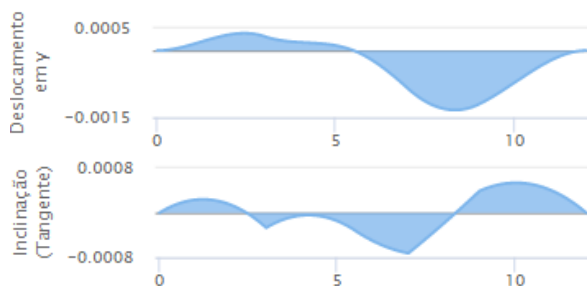


Figura 22. Gráficos de deslocamento e de inclinações desenvolvidos pelo software. [próprio]

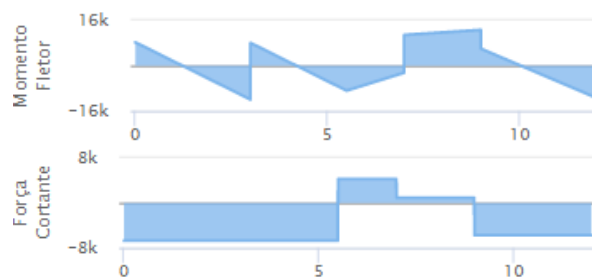


Figura 23. Gráficos de momento fletor e esforço cortante desenvolvidos pelo software. [próprio]

5.3.2. Comparação e Análise dos Resultados do Problema 3

Ao desenvolver o problema analiticamente, os dados dos Quadros 5 e 6 são obtidos. Assim, comparando com as informações obtidas no software, encontra-se o erro residual.

Quadro 5. Deflexões do segundo problema [próprio].

GL	ANALÍT.	SOFTWARE	ERRO
U1 (m)	0.00	-1.56e-16	$-\infty$
Θ1 (rad)	0.00	-8.97e-16	$-\infty$
U2 (m)	-3.182e-2	-3.18e-2	-0.0629%
Θ2 (rad)	2.693e-2	2.69e-2	-0.1114%
U3 (m)	0.00	-1.51e-18	$-\infty$
Θ3 (rad)	3.196e-2	3.20e-2	0.1252%
U4 (m)	8.402e-2	8.40e-2	-0.0238%
Θ4 (rad)	7.253e-2	7.25e-2	-0.0414%
U5 (m)	0.12038	0.1204	0.0166%
Θ5 (rad)	-3.9e-2	-3.89e-2	-0.2564%
U6 (m)	0.00	0.00	∞
Θ6 (rad)	0.00	0.00	∞

Quadro 6. Esforços do segundo problema [próprio].

GL	ANALÍT.	SOFTWARE	ERRO
F1 (N)	6740.50	6740.69	0.0028%
M1 (Nm)	8225.40	8225.70	0.0036%
F2 (N)	0.00	9.09e-13	∞
M2 (Nm)	20000.00	20000.00	0.0000%
F3 (N)	-10964.00	-10964.32	0.0029%
M3 (Nm)	0.00	0.00	∞
F4 (N)	3333.00	3333.33	0.0099%
M4 (Nm)	13333.00	13333.33	0.0025%
F5 (N)	6666.00	6666.66	0.0099%
M5 (Nm)	-6666.00	-6666.66	0.0099%
F6 (N)	-5775.90	-5776.37	0.0081%
M6 (Nm)	11394.00	11394.49	0.0043%

Por fim, analisando os resultados obtidos no Problema 3, nota-se também que há uma pequena margem de erro. Esses erros são provenientes do erro de cálculo da linguagem. Da mesma forma que nos dois primeiros problemas, os valores iguais a infinito são irrelevantes. Também, no intervalo dos nós 2 e 3 a função analítica esperada é uma função polinomial de quinto grau, devido ao carregamento distribuído triangular. Nesse

mesmo intervalo, o software utiliza uma função cúbica, representada na Equação 23. Assim, mesmo com uma pequena variação de valores, o resultado do software reflete a realidade do cálculo do MEF.

6. CONCLUSÃO

De acordo com os dados obtidos pelos três problemas estudados, verifica-se que em todos os casos as margens de erro foram pequenas em comparação aos valores reais. Considerando também, que os casos estudados abrangeram todas as possibilidades de carregamentos e apoios, verifica-se que o software conseguiu analisar todos os casos possíveis de problemas de vigas unidimensionais estudados por alunos de engenharia.

Porém, durante a execução deste estudo, para encontrar os coeficientes da Equação 23 foi utilizada uma equação cúbica, que resulta no cálculo de valores aproximados quando utilizados carregamentos na viga a ser solucionada. Outros elementos de viga como o elemento parabólico são elementos mais recomendados para modelagens que suportarão cargas distribuídas. Esses modelos podem gerar uma equação de grau cinco, que é a equação que gera valores exatos no intervalo de aplicação dos carregamentos uniformes e triangulares. Assim, a mudança da modelagem pode ser fonte para uma pesquisa futura.

O software também não possui nativamente as API's (Application Programming Interface) de integração necessárias para implantar software em outras plataformas. As mesmas são o conjunto de funções que possibilitam o software ter suas funções utilizadas por outras plataformas, como a ESO. Assim, a falta dessas API's representa outra limitação do software, e a inserção das mesmas no software pode ser fonte para uma futura pesquisa.

Avaliando também a metodologia do MEF, percebe-se que ao ser adicionado um carregamento sobre um apoio, os esforços equivalentes estarão aplicados diretamente sobre o apoio. Como um apoio de qualquer gênero nunca admite deslocamento vertical, esses esforços equivalentes não serão considerados nos cálculos. Assim, a realidade de um carregamento não será representada. Portanto, caso seja necessário solucionar uma viga com carregamentos distribuídos ao longo de toda a viga, o modelo reticulado aplicado pelo software não é recomendado.

Portanto, ao fim deste trabalho, avaliando os dados fornecidos e considerando as limitações do software, pode se dizer que o trabalho atingiu o objetivo proposto. Assim, foi desenvolvido um software open-source, visto que estará disponível via internet por meio do GitHub. O software desenvolvido também é didático, explicando como os cálculos foram obtidos. O software também é prático, por ser de fácil e rápida execução em qualquer dispositivo, e também representa a didaticamente o MEF, tendo em vista os

problemas que foram solucionados por meio do software.

REFERÊNCIAS

- [1] ALVES FILHO, AVELINO. **Elementos finitos: A Base da Tecnologia CAE**, São Paulo: Editora Érica, 2000.
- [2] BEER, F. P.; JHONSTON, E. R. **Resistência dos Materiais**, São Paulo, Makron Books, 1996.
- [3] OLIVEIRA, J. C. F. **Dimensionamento de Estruturas Metálicas Espaciais com o Desenvolvimento de um Software de Elementos Finitos Orientado a Objeto**, 2002. 124 f. Dissertação de Mestrado – Departamento de Engenharia Mecânica, UFES – Universidade Federal do Espírito Santo, Vitória – Espírito Santo, 2002.
- [4] TUTORIAL, HTML. Disponível em: https://www.w3schools.com/html/html_intro.asp. Acesso em, v. 11, 2017.
- [5] TUTORIAL, CSS. Disponível em: https://www.w3schools.com/css/css_intro.asp. Acesso em, v. 11, 2017.
- [6] TUTORIAL, JavaScript. Disponível em: https://www.w3schools.com/js/js_intro.asp. Acesso em, v. 11, 2017.
- [7] OFFICIAL WEBSITE, JavaScript. Disponível em: <https://www.javascript.com/>. Acesso em, v. 11, 2017.
- [8] OFFICIAL WEBSITE, SASS. Disponível em: <http://sass-lang.com/>. Acesso em, v. 11, 2017.
- [9] OFFICIAL WEBSITE, JQuery. Disponível em: <https://jquery.com/>. Acesso em, v. 11, 2017.
- [10] OFFICIAL WEBSITE, Bootstrap. Disponível em: <https://getbootstrap.com/>. Acesso em, v. 11, 2017.
- [11] OFFICIAL WEBSITE, FabricJS. Disponível em: <http://fabricjs.com/>. Acesso em, v. 11, 2017.
- [12] OFFICIAL WEBSITE, MathJS. Disponível em: <http://mathjs.org/>. Acesso em, v. 11, 2017.
- [13] OFFICIAL WEBSITE, MathJax. Disponível em: <https://www.mathjax.org/>. Acesso em, v. 11, 2017.
- [14] OFFICIAL WEBSITE, HighCharts. Disponível em: <https://www.highcharts.com/>. Acesso em, v. 11, 2017.
- [15] OFFICIAL WEBSITE, GitHub. Disponível em: <https://github.com/>. Acesso em, v. 11, 2017.
- [16] SOFTWARE ACADÊMICO PARA RESOLUÇÃO DE VIGA GENÉRICA PELO MÉTODO DOS ELEMENTOS FINITOS, GitHub. Disponível em: <https://github.com/UCL-MecSol/VigaElemFinitos/>. Acesso em, v. 11, 2017.