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

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

# Resumo

*O trabalho neste artigo foi desenvolvido pelo orientando Luan Henrique Sirtoli, e pelo orientador Msc. Jean Carlo Ferreira de Oliveira, e contém a documentação da base teórica, implementação e resultados do desenvolvimento de um software para dimensionamento e calculo de vigas bidimensais baseado no método dos elementos finitos. A parte matemática será documentada neste documento, e o projeto do software será documentado na página oficial do projeto no GitHub. Este artigo é parte manuscrita do trabalho de conclusão de curso apresentado ao curso de Engenharia Mecânica,* *da Faculdade do Centro Leste, como requisito parcial para obtenção do título de Engenheiro Mecânico.*

**Palavras-chaves:** *Método dos Elementos Finitos, trabalho de graduação, Resistência dos Materiais, Software, Dimensionamento.*

**1. INTRODUÇÃO**

Durante sua formação acadêmica e na vida profissional, muitas vezes o Engenheiro Mecânico e Civil encontra-se com problemas estruturais de difícil resolução. Durante a fase acadêmica ou exercendo sua profissão, é necessário resolver problemas mais complexos, como estruturas de Vigas e Pórticos, que necessitam de uma parte matemática mais robusta, e nem sempre o engenheiro tem rapidamente à mão um software que auxilie na resolução, ou que possa validar os cálculos já previamente feitos. Esse é um problema que pode ser ocasional na vida do engenheiro se o mesmo trabalhar na área de projetos ou se o acadêmico estiver cursando as disciplinas de Resistência dos Materiais I e II.

Portanto, esse trabalho se baseia em fornecer um software didático, prático, open-source para dimensionamento e cálculos, que possa suprir a necessidade de ambos os usuários.

**2. METODOLOGIA**

Para o Projeto proposto, será projetado um software voltado à engenharia, tanto acadêmica quanto profissional. Como o dimensionamento de estruturas é uma parte refinada e delicada da engenharia, necessitamos de uma sólida base em Resistencia dos Materiais para dar ao nosso projeto a robustez e seriedade necessária para cálculos algébricos 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]. Portanto, para darmos essa garantia de cálculos, utilizaremos então, o Método dos Elementos Finitos (Finite Element Analysis).

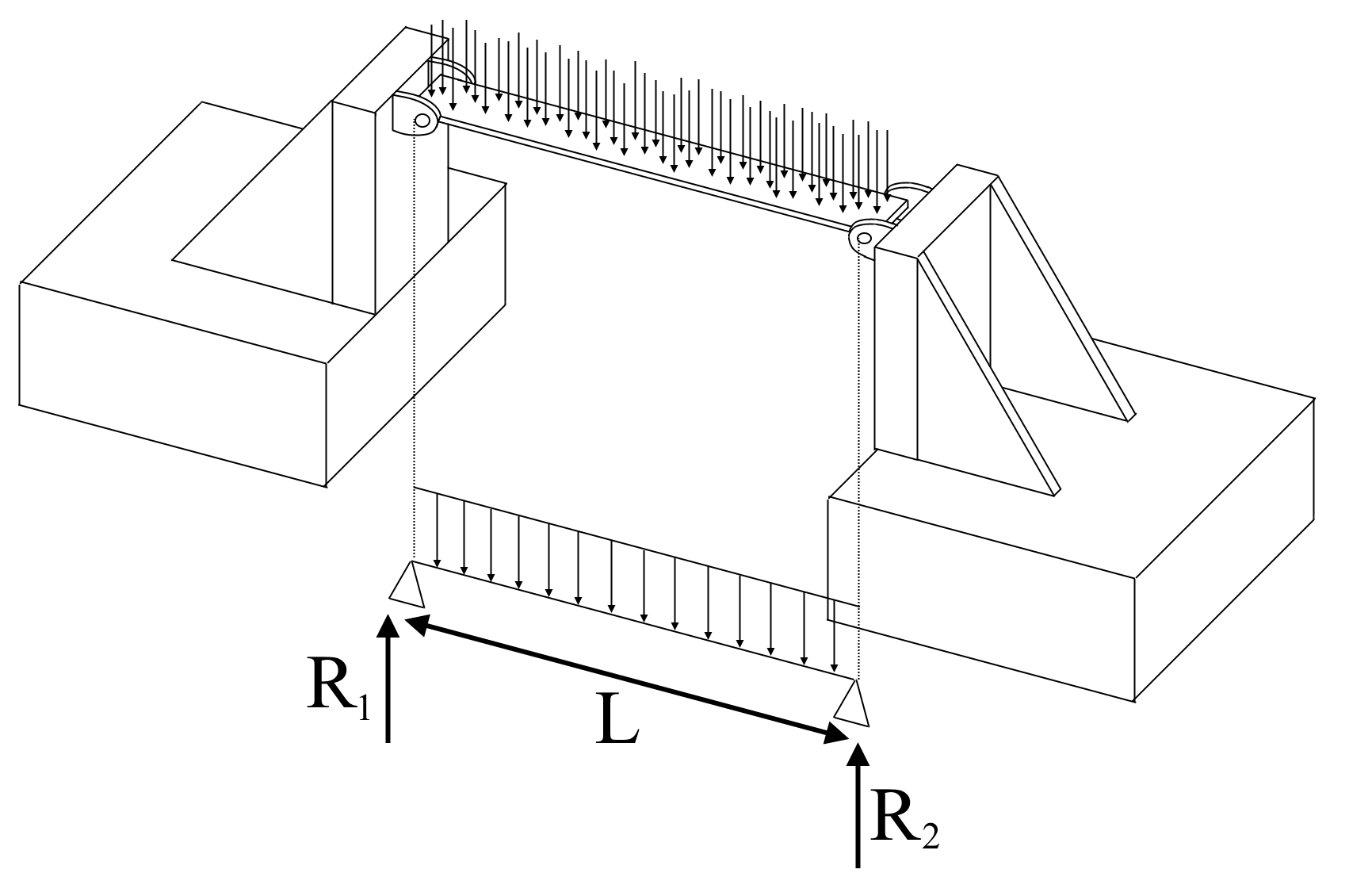
Para esse projeto, também se faz necessária a utilização de programação voltada à aplicativos Web, para que as simulações se tornem mais simples de serem executadas pelo usuário, tornando assim, o software mais prático, podendo ser executado em qualquer tipo de dispositivo móvel. Com esse tipo de linguagem, pode-se montar uma página modular leve, que se adaptará ao dispositivo onde será executado, montando uma prévia do problema proposto e podendo fazer interações com outros softwares.

**2.1. O Método dos Elementos Finitos**

O Método dos Elementos Finitos se baseia na simplificação de problemas complexos que muitas vezes seriam impossíveis de serem calculados por métodos simples de Resistência dos Materiais, transformando vigas, pórticos ou treliças em elementos menores e mais simples, possibilitando assim os cálculos. Esse método faz grande uso de soluções algébricas para os cálculos, sendo os mesmos de aplicação matricial.

**2.1.1. Modelagem da Estrutura**

Para a resolução de um problema real de estrutura, é necessário fazer uma análise apurada do problema a ser resolvido, transformando o problema real em um problema simplificado que seja capaz de ser modelado dentro dos padrões da análise de estruturas. “[...] 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 abaixo, 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]

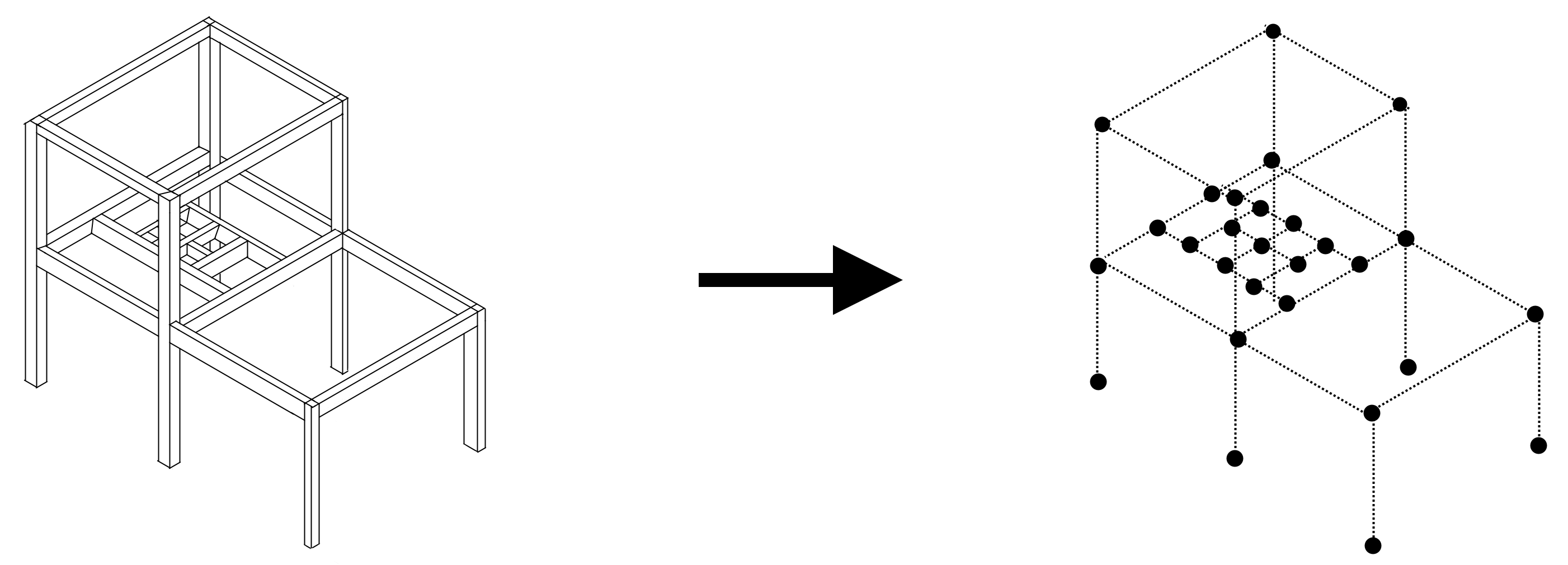
Um modelo idealizado como o acima, pode ser resolvido com certa facilidade, pois o mesmo pode ser resolvido a partir das equações de Equilíbrio. [2]

“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, ∑F representa a soma de todas as forças que agem sobre o corpo, e ∑M0 é a soma dos momentos de todas as forças em torno de qualquer ponto O dentro ou fora do corpo.” [2]

Como trabalharemos com nossas idealizações bidimensionais, a aplicação dessas equações também será feita em duas dimensões.[2]

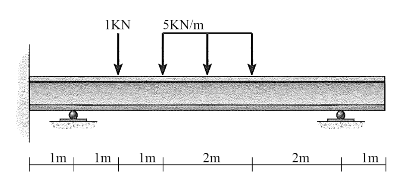
Entretanto, ao serem modeladas estruturas mais complexas, as equações de equilíbrio não bastarão mais para a resolução do problema, como indica o Grau de Hiperestaticidade. Esse grau indica o número de reações excedentes ao número de equações possuídas, que, para o caso de uma estrutura bidimensional, são três. Caso o grau de hiperestaticidade seja maior que zero, possuiremos então, uma **viga hiperestática**. Se a idealização do problema se mostrar hiperestática, teremos que partir então, para uma nova abordagem, o **Método dos Elementos Finitos**. [1]

Para início da resolução do problema pelo método dos elementos finitos, é necessário fazermos uma abordagem diferente na modelagem do sistema a ser avaliado. Para a modelagem de um problema resolvido pelo método, necessitamos transformar nossa estrutura em uma **estrutura reticulada**. [1]

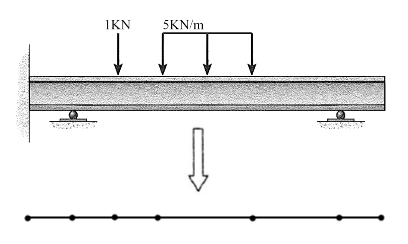


**Figura 2.** Modelagem de um sistema reticulado [próprio]

Para a modelagem de um sistema reticulado de uma viga, é necessário identificar cada nó estrutural para sabermos os elementos que estarão tendo solicitações mecânicas. [1]

 **Figura 3.** Exemplo de viga a ser dimensionado [próprio]

Para identificarmos corretamente os elementos correspondentes à uma estrutura hiperestática, iremos transformar cada força, momento, apoios de primeiro e segundo grau, engaste, ponta livre, e início e fim de carregamento como nós. A partir do exemplo da figura 3, teremos então, a seguinte estrutura idealizada. [1]

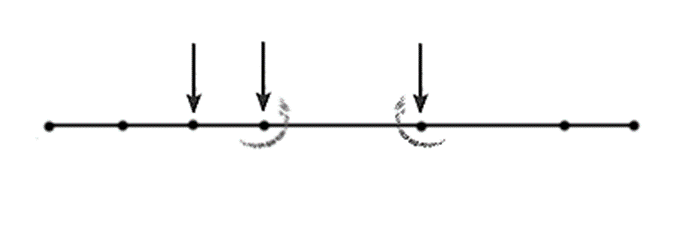


**Figura 4.** Modelagem de um sistema reticulado de viga [próprio]

Após reduzirmos nossa estrutura à uma estrutura reticulada, é necessário aplicar nos nós, as forças que foram aplicadas na estrutura. Para cargas e momentos aplicados, iremos aplica-los diretamente, já para a carga distribuída, é necessário transforma-la em forças e momentos equivalentes em cada ponto. [1]

Para tal, é necessário considerar os apoios como duas reações, e calcular com base nas equações de equilíbrio. Assim, encontrando as forças e momentos equivalentes, e podendo dar continuidade na análise da estrutura reticulada. [1]

Portanto, a estrutura reticulada final será de tal forma, que somente existirão os esforços aplicados sobre os nós. [1]



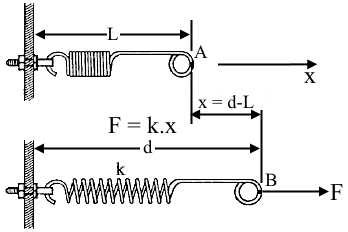
**Figura 5.** Estrutura reticulada de viga com esforços aplicados [próprio]

**2.1.2. Teoria da Rigidez do Elemento**

As cargas aplicadas sobre os nós irão gerar esforços nos elementos da viga, ou seja, causar 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, iremos analisar o comportamento de cada estrutura como se fossem 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]

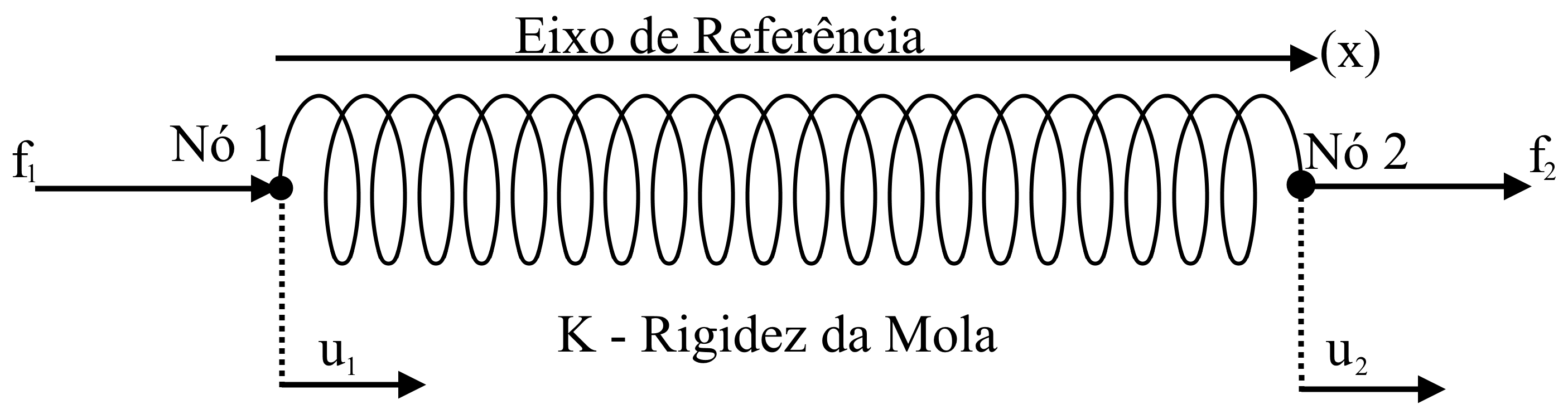
Assim, uma exemplificação para essa relação é exemplificada na figura abaixo:



**Figura 6.** Mola deformada sujeita à esforço [próprio]

Portanto, a partir da teoria de molas, obtemos a seguinte equação:

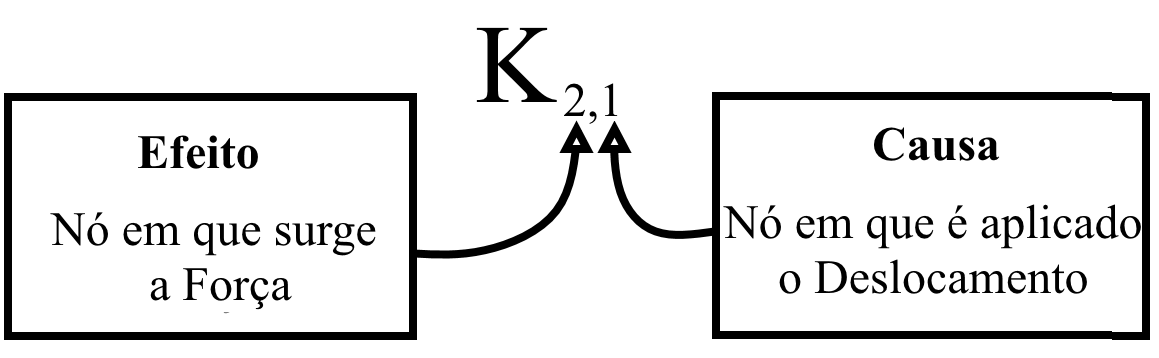
Com a equação acima em mãos, e aplicando o diagrama de corpo livre na mola, obteremos uma matriz de rigidez de 2 colunas e 2 linhas, pois há dois nós e um único grau de liberdade em cada, ou seja, somente possível movimento para o eixo x. Por fim, obtemos a configuração abaixo: [1]



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

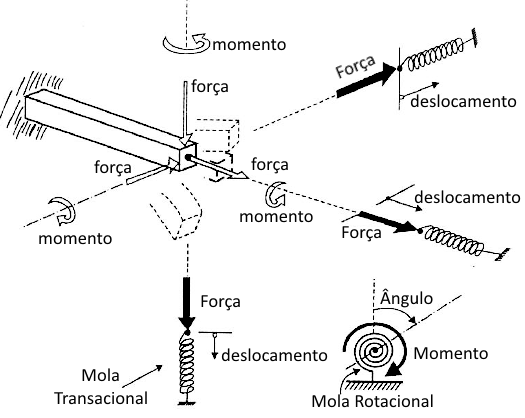
Com essas referências aplicadas na equação de rigidez da mola, obtemos a seguinte equação matricial: [1]

“[...] 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]. Sabendo a definição física dos termos contidos na matriz, podemos generalizar o significado para qualquer elemento finito. [1]

****

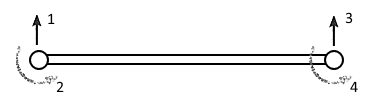
**Figura 8.** Efeito e causa no coeficiente de rigidez [próprio]

Como uma viga se comporta da mesma maneira que um elemento mola, podemos utilizar essa equação para definir as deformações de um elemento de viga. [1]



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

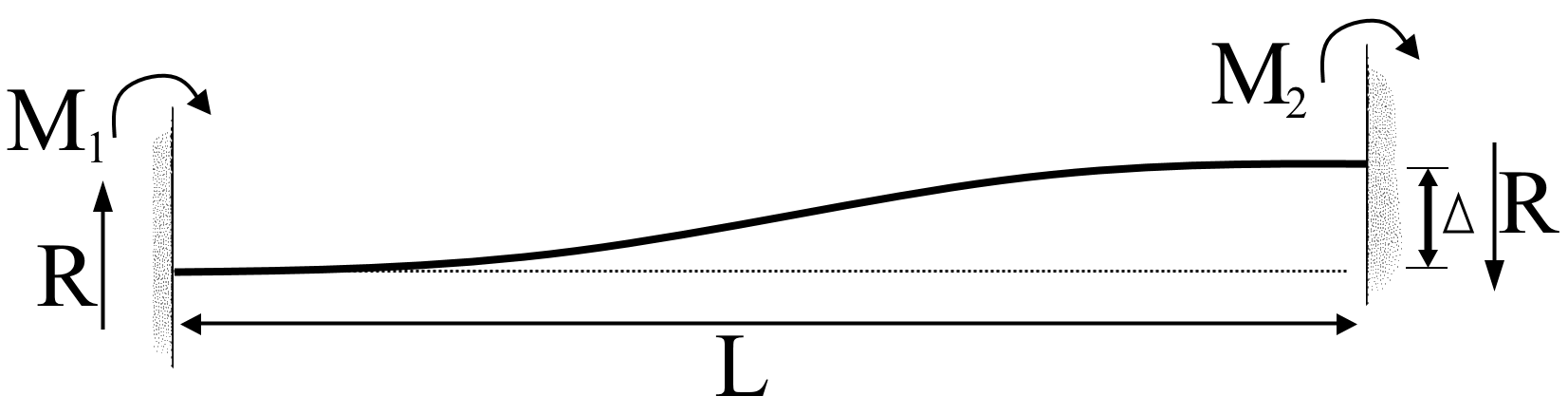
Como o trabalho proposto se baseia em uma viga exposta à deslocamento em Y e em inclinações angulares, temos então 2 graus de liberdade em cada nó, ou seja, uma matriz de rigidez local quadrada de 4 colunas e 4 linhas. [1]



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

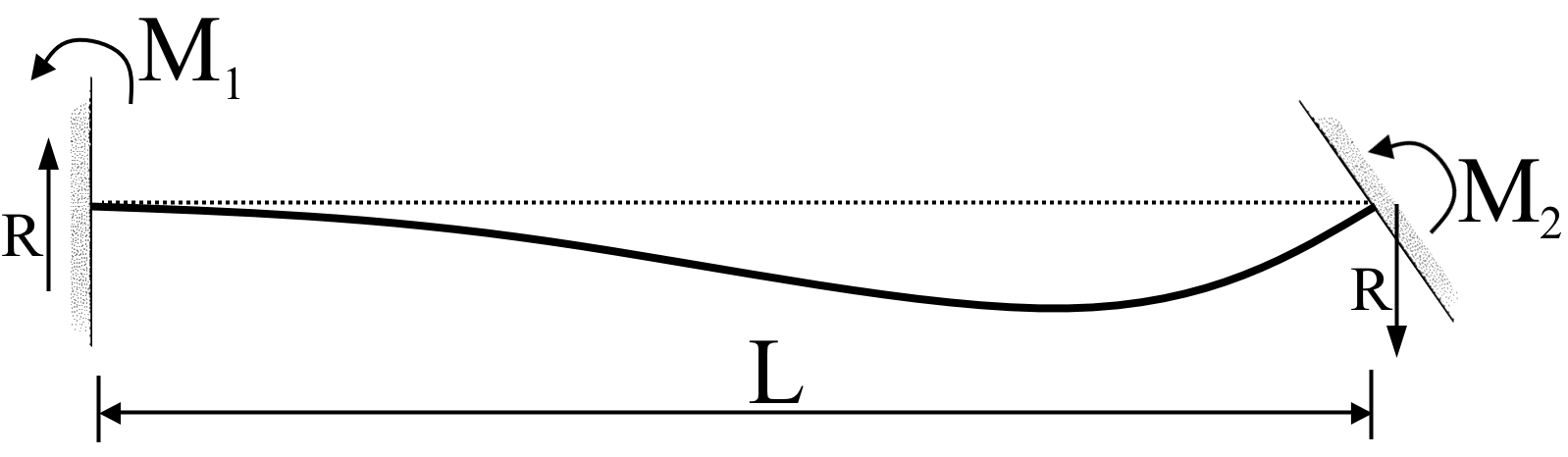
Nossa equação final, considerando as forças e momentos nodais, e deslocamentos e inclinações será: [1]

Para definirmos então, 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 abaixo, temos, para um deslocamento unitário: [1]



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

Para inclinação unitária: [1]



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

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

Ao trabalhar com treliças ou pórticos, é necessária a utilização de uma matriz de transformação, pois os elementos dos mesmos podem não estar alinhados. Porém, como nosso intuito é de generalizar um elemento de viga, os elementos da mesma sempre estarão alinhados, não necessitando assim, de uma matriz de transformação. [1]

Após a montagem das matrizes de rigidez local para todos os elementos da estrutura reticulada inicial, devemos montar a matriz global, fazendo a somatória de todos os valores de “k” coincidentes com os mesmos índices. [1]

Como estamos trabalhando com uma viga, os índices iniciais do próximo elemento, sempre serão os índices finais do elemento anterior. Isso causará uma “diagonalização” na soma dos elementos na matriz global. A matriz de rigidez global terá o comportamento abaixo durante sua montagem. [1]

Após montada a mesma, será necessária também, a montagem da matriz global reduzida, pois é necessário a inclusão dos apoios aplicados no problema proposto. O meio de dizer ao MEF que existem condições de contorno, é através da matriz global reduzida. Pois ao serem adicionados apoios de primeiro e segundo grau, não existe deslocamento no respectivo nó, e se houver engaste, não haverá deslocamento nem inclinação. Para tal, é necessário zerar a linha correspondente ao grau bloqueado, e adicionar um valor unitário no “k” de iguais índices. [3]

Ao ser feita a multiplicação matricial abaixo, iremos ter as seguintes equações: [3]

Portanto:

Portanto, ao inserir as condições de contorno, dizemos ao MEF quais pontos da viga não irão se deslocar ou inclinar. Assim, é feita multiplicação matricial utilizando a matriz de rigidez global reduzida e o vetor de forças e momentos aplicados, é encontrado os deslocamentos e inclinações nodais. Então, para encontrarmos o vetor com forças e reações nodais, multiplicamos o vetor de deslocamentos e inclinações encontrado, pela matriz de rigidez global original. [3]

Possuindo então, todos os deslocamentos, inclinações e esforços nos nós, podemos dar início à montagem dos gráficos. Para montagem dos gráficos, necessitaremos da função de cada gráfico. Como possuímos os valores de deslocamento e de inclinação, faremos ambas as funções. Para se desenvolver a função de deslocamento, iremos utilizar uma função polinomial que possua 4 coeficientes, pois o elemento possui quatro graus de liberdade. A função para esse caso é a **função cubica**.[1][3] Portanto:

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

Matricialmente:

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

Resolvendo a equação matricial, encontramos as quatro constantes, e podemos aplicar nas quatro funções abaixo: [1]

Função de Deslocamento:

Função de Inclinação:

Função de Momento Fletor:

Função de Força Cortante:

Essas funções irão ser repetidas para cada segmento da viga, gerando assim os gráficos necessários. [1]

**2.2. Programação Necessária**

Como as premissas do software são, ser didático, prático, open-source e em aplicações web, se faz necessário utilizar linguagens de programação que são de boa performance, e que sejam utilizáveis em várias plataformas como computadores e smartphones. Então, para atender as premissas, as linguagens abaixo foram escolhidas.

**HTML**: É a linguagem padrão de páginas Web. É uma linguagem mundialmente difundida, e de fácil entendimento por diversos dispositivos como computadores, notebooks, smartphones e tablets. A mesma também é de fácil construção e entendimento, pode ser construída por um editor de texto simples. Portanto, devido à premissa de o software ser de aplicações Web, o HTML se faz necessário, sendo utilizada a versão HTML5. [4]

**CSS**: Linguagem complementar ao HTML. Criada para dividir em um outro documento a parte de estilos separada da estrutura da página HTML, fornecendo assim para a página mais dinamismo, um código mais limpo e 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, iremos necessitar de bibliotecas especificas para o software, pois parte dos componentes de cada é um tanto simples, e necessita uma incrementação para atender os requisitos do que será calculado e do que iremos exibir na página. Será utilizada 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. Será necessária para o software, 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, teremos ferramentas suficientes para o desenvolvimento de um software que atenderá as premissas necessitadas e que poderá atender os objetivos finais do projeto.

**3. ANÁLISE DO PROBLEMA**

No mercado atualmente, não existem programas gratuitos que façam cálculos estruturais complexos como o de elementos finitos, e mesmo os que fazem cálculos simples, não são didáticos o suficiente. Geralmente são softwares simples, que calculam uma estrutura de viga genérica baseada no método das secções, e não fazem cálculos de vigas hiperestáticas.

Portanto, para que seja possível o desenvolvimento de um software completo, há a necessidade da modificação do código fonte do mesmo, porém, para a modificação desse código, existem alguns empecilhos, como:

a) Na maioria dos casos é menos trabalhoso começar uma estrutura de códigos nova do zero.

b) Problemas de direitos autorais, licenças ou registros de patentes.

c) Código mal estruturado, onde a modificação geralmente ocasiona em problemas de compilação.

d) Não possuem estrutura modular, e assim, não possibilitam futuras melhorias.

e) Falta de documentação, que demanda tempo para compreendimento total do código.

f) Problemas de maleabilidade, para serem integrados em outros sistemas, como uma plataforma de Ensino à Distância.

Além dos problemas citados, grande parte dos softwares não tem mobilidade de plataformas, sendo assim, de difícil execução por dispositivos de maior facilidade de acesso, como tablets e smartphones.

**4. IMPLANTAÇÃO**

**4.1. Primeiros Passos**

A partir dos critérios, da metodologia do Método dos Elementos finitos e premissas que definimos anteriormente, e iremos então construir o software. É definida a página inicial construída em HTML com componentes do BootStrap, que irá conter quatro painéis, sendo dois painéis para manipulações do problema, um para a pré-visualização do problema proposto, e um contendo os resultados obtidos após a simulação. [4][10]

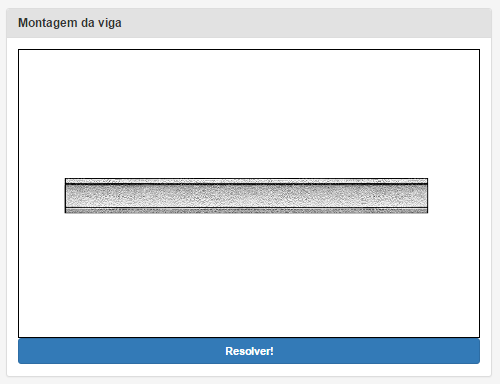
Para a manipulação do problema, com utilização dos componentes do BootStrap, foram criados dentro dos painéis botões que representam os apoios e os esforços, que ao serem acionados irão adicionar aos seus respectivos painéis os campos para preenchimento dos dados, onde será necessário a inserção da posição para os apoios, e 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, que são o momento de inércia em metros à quarta potência e comprimento total em metros, e inserção das propriedades do material, que é o módulo de elasticidade em Giga Pascais. Os campos acima foram criados de modo a evitar erros de inserção, bloqueando a entrada de letras, com exceção de notação cientifica. [1][2][4][10]

Para a pré-visualização do problema, foi desenvolvido um painel contendo um Canvas, que é um quadro onde podem ser inseridos diversos objetos. O mesmo foi desenvolvido com base no FabricJS, de modo que ao inicializar o software, o mesmo 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 pré-visualização do problema irá ser dado início aos cálculos. Utilizando funções no JavaScript , irão ser preenchidas as arrays necessárias para resolução do problema, que irão conter as posições dos apoios, posições e intensidades dos esforços, posições de cada nó. Com os dados obtidos, o software irá utilizar o Método dos Elementos Finitos desenvolvido no capítulo 2 deste artigo em conjunto com a biblioteca MathJS e dar inicio aos cálculos, encontrando as matrizes de deslocamentos e inclinações, e a matriz de forças e reações. Irá então dar inicio ao cálculo da matriz de coeficientes, e então, a parte de cálculoes estará finalizada para ser exibida no ultimo painel. [6][7]

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

Portanto, após a cálculo dos dados, o software irá utilizar a biblioteca MathJax exibir a resolução do problema proposto em um texto explicativo com as funções aplicadas para resolução do problema, com matrizes representando as matrizes do MEF, exibindo os deslocamentos e inclinações nos pontos descritos, os esforços aplicados e reações em cada nó, e por fim, utilizando os componentes da biblioteca HighCharts, irão ser 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]

**4.2. Definição das Funções**

Para darmos ao software funcionalidade, as funções em JavaScript necessitam ser desenvolvidas, pois as mesmas serão a parte lógica do programa, que darão dinamismo aos botões para adição de componentes ao problema, que farão o preenchimento das arrays necessárias, que farão os cálculos matriciais e que exibirão os resultados ao usuário.

Portanto, as funções abaixo serão definidas, e serão melhor especificadas na página oficial do projeto no GitHub.[15][16]

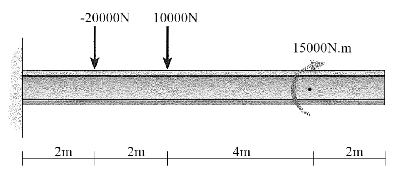
**5. RESULTADOS**

Após a implantação do nosso software, é necessário verificar a eficiência dos cálculos do mesmo. Portanto, para tal, irá ser 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]

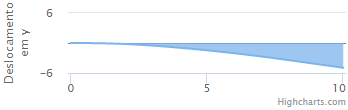
**5.1. Problema Proposto 1**

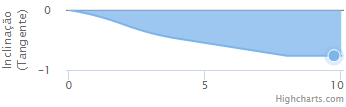
Para o primeiro problema, é proposto à simulação de uma viga isostática simples de comprimento igual à 10m com modulo de elasticidade de E=210Gpa e momento de inércia de I=, engastada em x=0m, com duas cargas aplicadas, sendo uma de -20000N aplicada em x=2m, uma de 10000N em x=4m um momento de 15000N.m aplicada em x=8m.

**5.1.1. Simulação da Estrutura**

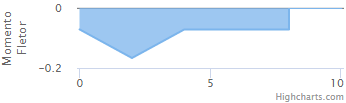
**Figura 15.** Viga à ser simulada no software. [próprio]

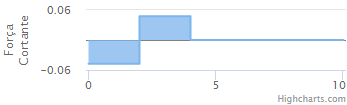
Após a simulação da estrutura proposta, obtemos os seguintes resultados no software:





**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 dos Resultados Analiticos com Dados do Software do Problema 1**

Ao desenvolver o problema analiticamente, obtemos os resultados demonstrados na primeira coluna dos quadros abaixo, comparados com os dados do software na segunda coluna:

**Quadro 1.** Resultados das deflexões do primeiro problema [próprio].

|  |  |  |
| --- | --- | --- |
| **Deslocamento Nó 1** | 0m | -1.24176e-13m |
| **Inclinação Nó 1** | 0 | 4.87836e-14 |
| **Deslocamento Nó 2** | 0.20635m | 0.206349m |
| **Inclinação Nó 2** | 0.2381 | 0.238095 |
| **Deslocamento Nó 3** | 0.95238m | 0.952381m |
| **Inclinação Nó 3** | 0.47619 | 0.476190 |
| **Deslocamento Nó 4** | 3.42857m | 3.428571m |
| **Inclinação Nó 4** | 0.7619 | 0.761905 |
| **Deslocamento Nó 5** | 4.95238m | 4.952381m |
| **Inclinação Nó 5** | 0.7619 | 0.761905 |

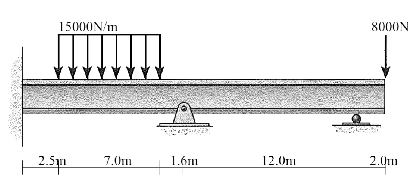
**Quadro 2.** Resultados dos esforços do primeiro problema [próprio].

|  |  |  |
| --- | --- | --- |
| **Força Nó 1** | 10000.0N | 10000.00N |
| **Momento Nó 1** | -15000.0Nm | -14999.99Nm |
| **Força Nó 2** | -20000.0N | -20000.00N |
| **Momento Nó 2** | 0.0Nm | 0Nm |
| **Força Nó 3** | 10000.0N | 10000.00N |
| **Momento Nó 3** | 0.0Nm | -1.4552e-11Nm |
| **Força Nó 4** | 0.0N | -1.7462e-10N |
| **Momento Nó 4** | 15000.0Nm | 14999.99Nm |
| **Força Nó 5** | 0.00N | -5.8207e-11N |
| **Momento Nó 5** | 0.00N.m | 1.7463e-10Nm |

**5.2. Problema Proposto 2**

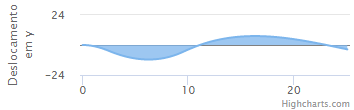
Para o segundo problema, é proposto à simulação de uma viga hiperestática de comprimento igual à 25,1m com modulo de elasticidade de E=210Gpa e momento de inércia de I=, engastada em x=0m, com dois apoios, um de primeiro grau em x=23.1m e um de segundo grau em x=11,1m, com uma carga distribuída de -15000N/m aplicada de x=2,5m até x=9,5m e uma carga pontual de 8000N em x=25,1m.

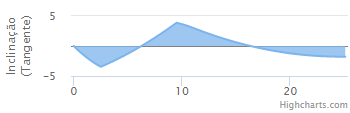
**5.2.1. Simulação da Estrutura**



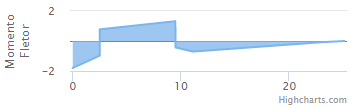
**Figura 18.** Segunda viga à ser simulada no software. [próprio]

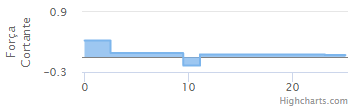
Após a simulação da estrutura proposta, obtemos os seguintes resultados no software:





**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 dos Resultados Analiticos com Dados do Software do Problema 2**

Ao desenvolver o problema analiticamente, obtemos os resultados demonstrados na primeira coluna dos quadros abaixo, comparados com os dados do software na segunda coluna:

**Quadro 3.** Resultados das deflexões do segundo problema [próprio].

|  |  |  |
| --- | --- | --- |
| **Deslocamento Nó 1** | 0m | -1.4018e-13m |
| **Inclinação Nó 1** | 0 | -1.0134e-13 |
| **Deslocamento Nó 2** | 4.75418m | 4.7542m |
| **Inclinação Nó 2** | 3.46171 | 3.4617 |
| **Deslocamento Nó 3** | 5.53441m | 5.5344m |
| **Inclinação Nó 3** | -3.87554 | -3.8755 |
| **Deslocamento Nó 4** | 0m | 3.1598e-14m |
| **Inclinação Nó 4** | -2.96907 | -2.9691 |
| **Deslocamento Nó 5** | 0m | 0m |
| **Inclinação Nó 5** | 1.7131 | 1.7131 |
| **Deslocamento Nó 6** | 3.5278m | 3.5278m |
| **Inclinação Nó 6** | 1.7893 | 1.7893 |

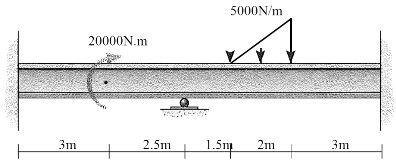
**Quadro 4.** Resultados dos esforços do segundo problema [próprio].

|  |  |  |
| --- | --- | --- |
| **Força Nó 1** | -68874.0N | -68873.84N |
| **Momento Nó 1** | -376880.0Nm | -376875.56Nm |
| **Força Nó 2** | 52500.0N | 52500.00N |
| **Momento Nó 2** | 367500.0Nm | 367499.99Nm |
| **Força Nó 3** | 52500.0N | 52500.00N |
| **Momento Nó 3** | -367500.0Nm | -367500.00Nm |
| **Força Nó 4** | -47116.0N | -47115.83N |
| **Momento Nó 4** | 0.0Nm | 2.6921e-10Nm |
| **Força Nó 5** | 2989.7N | 2989.67N |
| **Momento Nó 5** | 0.0Nm | 5.8208e-11Nm |
| **Força Nó 6** | 8000.0N | 8000.00N |
| **Momento Nó 6** | 0.0Nm | -1.1642e-10Nm |

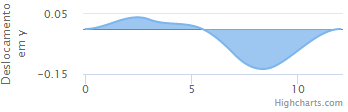
**5.3. Problema Proposto 3**

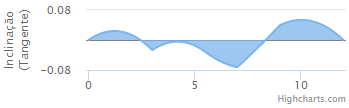
Para o terceiro problema, é proposto à simulação de uma viga hiperestática de comprimento igual à 12m com modulo de elasticidade de E=210Gpa e momento de inércia de I=, bi engastada em x=0m e x=12m, com um apoio de primeiro grau em x=5.5m, com uma carga distribuída triangular de inicio 0N/m e fim de 10000N/m aplicada de x=7m até x=9m e um momento de 20000N em x=3m.

**5.3.1. Simulação da Estrutura**

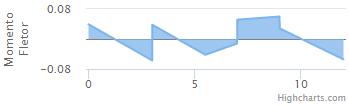


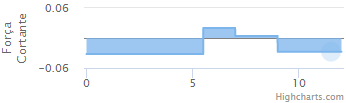
**Figura 21.** Terceira viga à ser simulada no software. [próprio]





**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 dos Resultados Analiticos com Dados do Software do Problema 3**

Ao desenvolver o problema analiticamente, obtemos os resultados demonstrados na primeira coluna dos quadros abaixo, comparados com os dados do software na segunda coluna:

**Quadro 5.** Resultados das deflexões do segundo problema [próprio].

|  |  |  |
| --- | --- | --- |
| **Deslocamento Nó 1** | 0m | 1.5591e-16m |
| **Inclinação Nó 1** | 0 | -8.9650e-16 |
| **Deslocamento Nó 2** | -0.03182m | -0.0318m |
| **Inclinação Nó 2** | 0.02693 | 0.0269 |
| **Deslocamento Nó 3** | 0m | -2.3387m |
| **Inclinação Nó 3** | 0.03196 | 0.0320 |
| **Deslocamento Nó 4** | 0.08402m | 0.0840m |
| **Inclinação Nó 4** | 0.07253 | 0.0725 |
| **Deslocamento Nó 5** | 0.12038m | 0.1204m |
| **Inclinação Nó 5** | -0.039 | -0.0389 |
| **Deslocamento Nó 6** | 0m | 0m |
| **Inclinação Nó 6** | 0 | 0 |

**Quadro 6.** Resultados dos esforços do segundo problema [próprio].

|  |  |  |
| --- | --- | --- |
| **Força Nó 1** | 6740.50N | 6740.69N |
| **Momento Nó 1** | 8225.40N.m | 8225.70Nm |
| **Força Nó 2** | 0.00N | 9.0949e-13N |
| **Momento Nó 2** | 20000.00N.m | 20000Nm |
| **Força Nó 3** | -10964.00N | -10964.32N |
| **Momento Nó 3** | 0.00N.m | 0Nm |
| **Força Nó 4** | 3333.00N | 3333.33N |
| **Momento Nó 4** | 13333.00N.m | 13333.33Nm |
| **Força Nó 5** | 6666.00N | 6666.66N |
| **Momento Nó 5** | -6666.00N.m | -6666.66Nm |
| **Força Nó 6** | -5775.90N | -5776.37N |
| **Momento Nó 6** | 11394.00N.m | 11394.49Nm |

**5.4. Análise dos Dados**

Ao desenvolver o problema analiticamente e com o software, nota-se que os valores obtidos para deflexões e esforços de ambos os casos estão muito próximos, onde se nota que em alguns casos os valores chegam a ser iguais. Assim, o software demonstra grande eficiência em simular os problemas propostos.

**6. CONCLUSÃO**

Ao terem sido estudados casos genéricos, e abrangido todas as possibilidades de carregamentos e apoios com o software dando cálculos com grande precisão, nota-se que o software foi eficiente em atingir o objetivo de ser um software confiável.

Porém, durante o desenvolvimento do software, para possibilitar o mesmo foi adotado para o cálculo dos gráficos a utilização de uma função quadrática com base nos coeficientes da Equação 23. Essa equação quadrática resulta em dados aproximados quando é utilizada para modelagem de problemas com carregamentos. Essa aproximação é feita para possibilitar o desenvolvimento do software, e resulta assim, em uma limitação do mesmo.

O projeto também foi desenvolvido sem pré-integração com nenhum software, portanto, mesmo sendo desenvolvido modularmente, para fazer a interação com outro software é necessário pequenas modificações.

Portanto, ao fim deste trabalho, avaliando os dados fornecidos, mesmo considerando as limitações do software, podemos dizer que o trabalho atingiu o objetivo proposto.

Com esse trabalho, podemos também desenvolver trabalhos futuros como a simulação exata dos carregamentos retangulares e triangulares, a expansão do software para abranger problemas tridimensionais, e também fazer a integração do mesmo com o sistema ESO da Faculdade UCL.

**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.