

DESENVOLVIMENTO DE SOFTWARE PARA ANÁLISE DE TRELIÇAS PLANAS

Pedro Luiz
Engenharia de computação
INSPER
São paulo, Brasil
pedrolcb@al.insper.edu.br

Gabriel Zezze
Engenharia de computação
INSPER
São paulo, Brasil
gabrielzfz@al.insper.edu.br

Gabriel Duarte
Engenharia de computação
INSPER
São paulo, Brasil
gabrielmd@al.insper.edu.br

Abstrato—Este documento relata à construção de um software para cálculo de treliças e sua possível aplicação no contexto de engenharia. (*Abstract*)

Keywords—engenharia, software, trelicas (*key words*)

I. INTRODUÇÃO

Este documento irá apresentar o processo de desenvolvimento de um software para calcular as forças nos nós de treliças o qual irá auxiliar na construção de uma ponte para uma competição entre grupos da sala de aula para testar qual ponte suporta a maior carga.

II. CONCEITOS MECÂNICOS

A. TRELIÇAS

As treliças são as estruturas constituídas de barras e ligações onde serão aplicadas forças resultando em reações nos nós.

B. Nós

Ligação entre elementos da estrutura.

C. Apoios

Elementos que ligam a estrutura com o solo e restringem o movimento.

D. Tensões

São as reações das barras contra a deformação.

E. Momento Fletor

É a soma de todos os momentos gerando a reação no corpo que tende a fletir em relação ao eixo longitudinal.

F. Deformação

Alteração no formato ou tamanho do corpo sujeito a uma força externa.

G. Deslocamento

Alteração na posição dos nós.

III. DESENVOLVIMENTO DO SOFTWARE

A. Escolhendo uma linguagem

Primeiramente, para iniciar o desenvolvimento foi necessário decidir a linguagem de programação que seria usada. A linguagem escolhida deve possuir capacidades

matemáticas além de facilidade para lidar com matrizes, por fim a decisão foi usar o python que satisfaz todas as necessidades descritas.

B. Arquitetura de implementação

A arquitetura escolhida é uma arquitetura simples, na qual no arquivo main.py faz a leitura da entrada, extração e formatação de todos os dados necessários. O arquivo solver.py possui duas implementações de resoluções, método de gauss e método de jacobi, por padrão o método utilizado é o método de gauss no entanto é possível alterar o software para usar o método de jacobi, existe também o arquivo matrizes.py o qual possui funções para auxiliar no gerenciamento das matrizes de restrições e conexões. Por fim, o software também utiliza o arquivo funcoesTermosol.py que possui funções para auxiliar na saída do software e leitura de dados.

C. Escolhendo uma linguagem

O software é executado no seguinte fluxo : primeiramente os dados são lidos do excel e formatados no main.py utilizando as funções auxiliares do arquivo funcoesTermosol.py. Após isso em posse dos dados formatados as matrizes de restrições e conexões são construídas e, por fim, estas matrizes são inseridas na função solver_gauss, que retorna um vetor de deslocamento o qual permite o cálculo das deformações, forças internas e tensões internas. Logo após é utilizada a função gera_saida do arquivo funcoesTermosol.py para gerar a saída formatada de um jeito legível.

IV. ANÁLISE DE RESULTADOS E TESTES

A. O Teste

Para validar os resultados do software foi utilizado uma treliça mais simples com resultados previamente calculados por um artigo e imputar essa treliça no programa para conferir a saída do programa com os resultados previamente calculados e verificados.

A treliça usada é representada na Imagem 1 e os resultados a serem comparados estão representados nas Tabelas 1 e 2

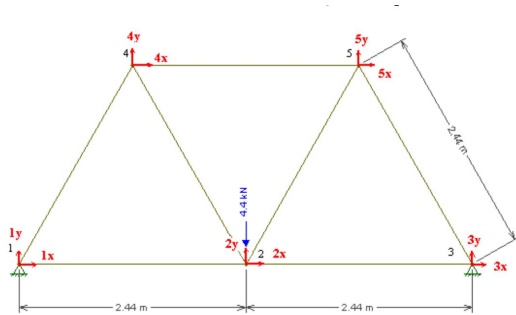


Imagem 1

Item	x[m]	y[m]
1	0	0
2	0	$-7 \cdot 10^{-5}$
3	0	0
4	$1,2 \cdot 10^{-5}$	$-3,5 \cdot 10^{-5}$
5	$-1,2 \cdot 10^{-5}$	$-3,5 \cdot 10^{-5}$

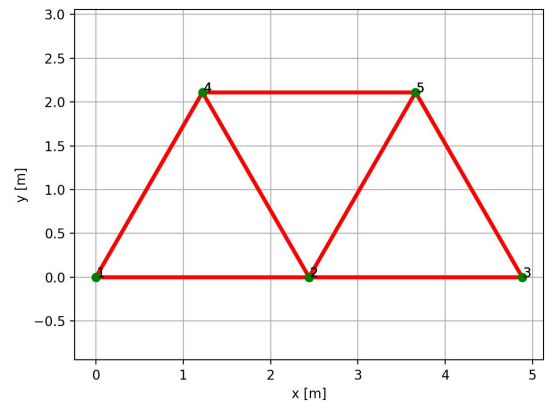
Tabela 1-Vetor Deslocamento validado.

Tabela 2 - Tensões internas de cada elemento validadas

Elemento	Tensão interna [mPa]
1	0
2	-2
3	2
4	-2
5	2
6	0
7	-2

B. Os Resultados

Inserindo essa treliça no software desenvolvido foram obtidos os resultados presentes nas Tabelas 3 e 4:



Item	x[m]	y[m]
1	0	0
2	0	$-6,96 \cdot 10^{-5}$
3	0	0
4	$1,2 \cdot 10^{-5}$	$-3,48 \cdot 10^{-5}$
5	$-1,2 \cdot 10^{-5}$	$-3,48 \cdot 10^{-5}$

Tabela 3 - Vetor deslocamento obtido no software desenvolvido

Elemento	Tensão Interna [Pa]
1	$1,128 \cdot 10^{-4}$
2	$-1,976 \cdot 10^6$
3	$1,976 \cdot 10^6$
4	$-1,976 \cdot 10^6$
5	$1,976 \cdot 10^6$
6	$1,128 \cdot 10^{-4}$
7	$-1,976 \cdot 10^6$

Tabela 4 - Tensões internas de cada elemento obtidas pelo software desenvolvido

Como pode ser observado os resultados apresentados pelo artigo estão muito semelhantes aos resultados apresentados pelo software. Assim, é razoável afirmar que o programa desenvolvido é válido.

V. MÉTODOS UTILIZADOS

O software foi construído utilizando o método de Gauss-Seidel que é um método numérico para a solução de sistemas de equações lineares como mostra a fórmula abaixo.

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)} \right), i = 1, 2, \dots, n$$

VI. CONSTRUÇÃO DA PONTE

A. A proposta:

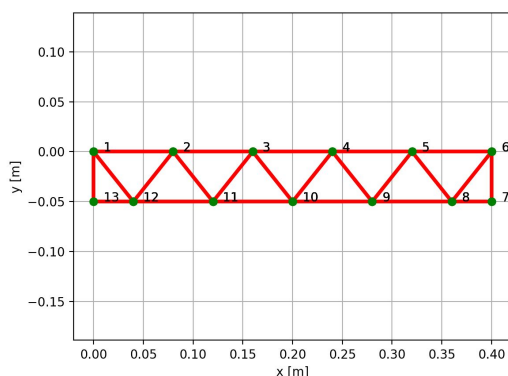
Foi proposta à modelagem de uma ponte que transpassasse um vão livre de 400mm. Cada elemento dessa ponte pode ter, no máximo, 110mm de comprimento e a massa total da estrutura não pode ser maior que 300g. O material utilizado possui densidade de 848 [kg/m³], módulo de elasticidade longitudinal de 4 [Gpa] e resistência à tração/compressão de 18/18 [Mpa].

B. Os requisitos:

Nenhum membro pode ultrapassar a tensão de ruptura em tração ou compressão e ter deformação maior que 5%. Além disso, nenhum nó pode se deslocar mais do que 20 mm em X ou em Y e a estrutura deve suportar uma carga mínima de 150 N. Para fazer essas verificações no software foi feita uma varredura nas listas de deformações e tensões finais para ver se havia alguma irregularidade.

C. A construção:

A partir de conceitos estudados e algumas pesquisas foi desenvolvida uma primeira versão da ponte representada pela figura abaixo.



Sendo que os nós 1, 6, 7 e 13 são apoios do tipo engaste, ou seja, eles têm o deslocamento em x e y restritos. Após a construção foi feita uma análise que verificou que a estrutura suporta a carga mínima necessária.

D. Otimizando:

Analisando os resultados obtidos após a primeira iteração, foi possível verificar os pontos mais frágeis da estrutura, sendo eles os elementos formados pelos nós 7-8 e 12-13. Assim, aumentou-se a área transversal deles para suportarem uma tensão maior. Além disso, notou-se que os elementos formados pelos nós 1-13 e 6-7 possuíam tensão interna 0, fazendo com que eles fossem removidos para melhorar a relação carga/massa do sistema.

VII. LIMITAÇÕES

A. SOFTWARE

A limitação do software é em relação ao jeito que o plano é simulado, no software o plano é simulado em duas dimensões ou seja, qualquer força ou peso no eixo Z é desconsiderada.

B. MÉTODOS

Os métodos de jacobi e Gauss-Seidel apesar de muito poderosos possuem a limitação de poderem ser usados em apenas sistemas lineares de dimensão 3.

VIII. APLICAÇÕES

O software desenvolvido possui muitas aplicações possíveis no contexto de engenharia já que este permite à simulação de estruturas de duas dimensões é também à obtenção de valores essenciais durante a projeção de qualquer estrutura como Tensão interna, vetor deslocamento entre outros. Um exemplo de aplicação para um software semelhante é demonstrado na aula de Design de pontes através do solidworks (software semelhante de simulação de forças e estruturas) na qual é usada o software para projetar e modelar pontes com precisão é informacoes valiosas(referência 1)

Outro exemplo de aplicação para um software como o desenvolvido é presente na referência 2 no qual é comparado resultados reais medidos em pontes para trens de alta velocidade com resultados obtidos através de um software é assim é observado uma grande correlação entre os dois resultados demonstrando a precisão do software.

IX. Referências

- [1][Bridge Design Project](#)
- [2][High-speed train-track-bridge dynamic interactions – Part II: experimental validation and engineering application](#)