Simulação de forças em treliças utilizando Python

Paulo Kim, Raphael Lahiry, Rodrigo Coelho e Rodrigo Mattar
Insper – Instituto de Ensino e Pesquisa

ass

INTRODUÇÃO

As treliças são estruturas baseadas no desenho de um triângulo, constituídas de material que seja resistente o bastante para ser usada em diferentes aplicações estruturais, sendo muito comum em pontes. As treliças surgiram como uma estrutura mais econômica do que as vigas para superarem vãos maiores ou suportar cargas maiores **[3]**. Uma das construções mais famosas que é constituída quase que puramente por treliças é a Torre Eiffel.

Para a modelagem de estruturas treliçadas podemos utilizar o método dos elementos finitos. Já a solução de um sistema de treliça pode ser obtida a partir de diversos métodos analíticos, como por exemplo o método dos nós ou o método das seções. Todos esses cálculos podem ser realizados à mão, mas em projetos maiores que envolvem uma quantidade grande de membros, chegar à solução pode ser muito demorado **[2]**. Com avanço da tecnologia da computação, projetos de engenharia estrutural se tornaram totalmente atrelados a ferramentas computacionais **[1]**. Assim é possível desenvolver e utilizar métodos numéricos em softwares, como o método Jacobi ou Gauss-Seidel, que proporcionam uma grande economia de tempo e agilidade para o engenheiro conseguir solucionar e validar sistemas estruturais.

Com isso em mente, a ideia desse projeto é construir um software capaz de simular a atuação de forças em um sistema de treliças em 2D a partir de qualquer input do usuário.

ESTRUTURA DO SOFTWARE

Para a resolução desse problema, construímos um código em Python. Para isso seguimos as seguintes etapas:

- Importação dos dados via arquivo Excel (.xls)
- Destrinchamento dos dados em variáveis
- Plot inicial para visualização e validação do input de dados
- Cálculos da matriz K para cada elemento
- Cálculos da matriz K global
- Aplicação do solver utilizando o método de elementos finitos de Gauss-Seidel
- Cálculos das reações de apoio e deslocamentos
- Cálculos das deformações
- Cálculos das tensões internas
- Cálculos das forças internas
- Plot final com visualização do sistema após a deformação
- Save do output do modelo em um arquivo (.txt)

Com a estrutura feita, partimos para o teste do software. Para isso, utilizamos como input o seguinte sistema:

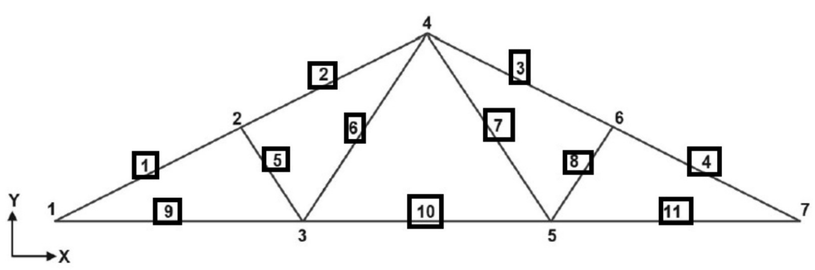

Figura 1: Exemplo de validação do software

Outros dados que não estão mostrados na figura são a presença de um suporte pinado no nó 1, travando seu deslocamento no eixo X e Y, e um suporte com rolamento no nó 7, travado seu deslocamento no eixo Y. Com o exemplo de validação em mão, colocamos ele em nosso modelo e obtivemos o seguinte gráfico:

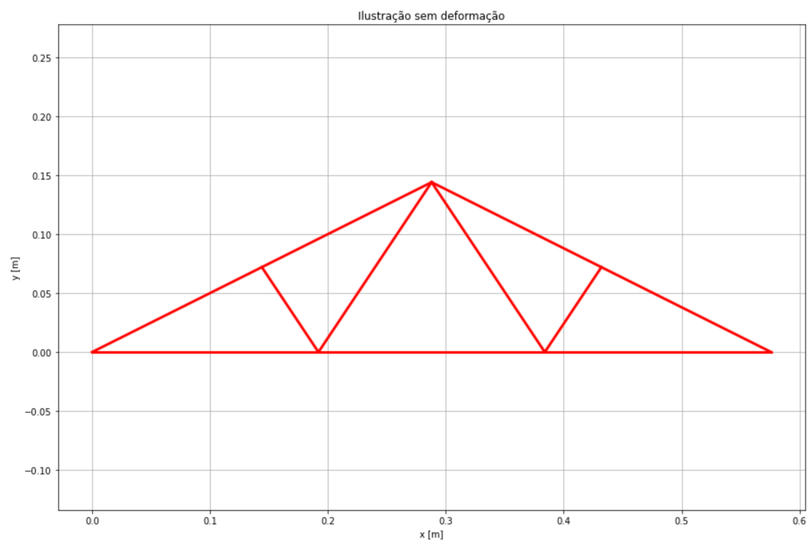

Figura 1: Ilustração do exemplo de entrada antes da aplicação das forças

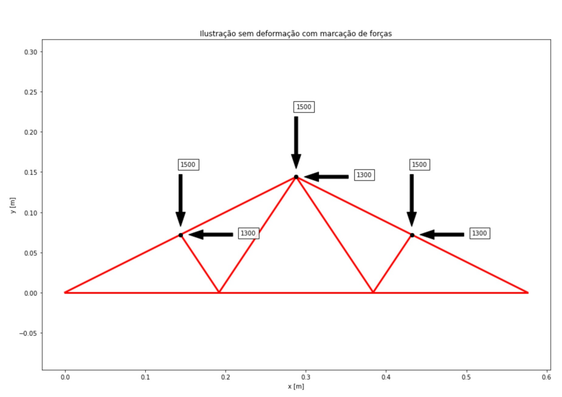

Figura 2: Ilustração das forças (N) do exemplo de entrada

Com os dados corretamente colocados no nosso modelo, foi possível então partir para sua resolução, visando analisar o resultado dessas forças em cima do nosso modelo de exemplo

RESULTADOS E DISCUSSÃO

Com o resultado do nosso modelo refinado em mãos, conseguimos obter um resultado visual muito bom, como pode ser observado abaixo. Nele podemos a posição do nós após a atuação das forças. Dadas as forças da Figura 2, é totalmente condizente com a realidade o resultado obtido, em que obtivemos um deslocamento expressivo no eixo Y para todos os nós com exceção dos nós 1 e 7, visto que eles possuem uma restrição para se movimentar nesse eixo. Além disso, um resultado que não é tão visível mas está presente nos dados finais e no gráfico é a movimentação dos de todos os nós para a esquerda, com exceção do nó 1, que possui uma restrição para o eixo X.

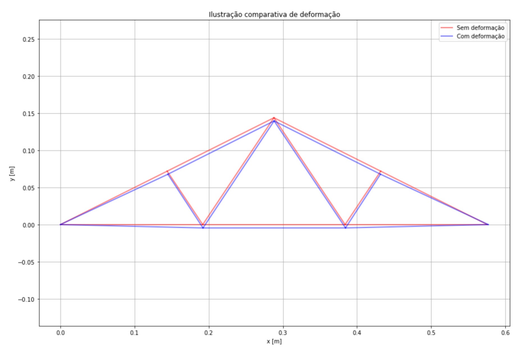

Figura 3: Comparação do sistema com e sem deformação

```
Reacoes de apoio [N]  Deformacoes []          Tensoes internas [Pa]
[[3900.00000118]      [[-0.00617581]          [[-1.23516136e+09]
 [2899.99999859]       [-0.00433904]           [-8.67807334e+08]
 [1599.99999843]]      [-0.00364692]           [-7.29384078e+08]
                       [-0.00340734]           [-6.81468335e+08]
Deslocamentos [m]      [-0.00072969]           [-1.45938980e+08]
[[ 0.         ]         [ 0.00072969]           [ 1.45938980e+08]
 [ 0.         ]         [ 0.0018457 ]           [ 3.69139773e+08]
 [ 0.00097028]         [-0.0018457 ]           [-3.69139773e+08]
 [-0.00416384]         [ 0.00180952]           [ 3.61904761e+08]
 [ 0.00034743]         [ 0.001     ]           [ 1.99999999e+08]
 [-0.00450318]         [ 0.00304762]]          [ 6.09523809e+08]]
 [ 0.00025083]
 [-0.00428701]        Forcas internas [N]
 [ 0.00053943]        [[-6484.5971316 ]
 [-0.00447852]         [-4555.9885012 ]
 [-0.00031315]         [-3829.26640827]
 [-0.00410209]         [-3577.7087605 ]
 [ 0.00112457]         [ -766.17964482]
 [ 0.         ]]       [  766.17964438]
                       [ 1937.98380975]
                       [-1937.98381058]
                       [ 3899.999996  ]
                       [ 1049.99999722]
                       [ 3199.99999687]]
```

Figura 4: Resultados numéricos do nosso software

Para validação do nosso resultado, utilizamos outro software já reconhecido e validado pelo mercado que é o LISA FEA. Ao inputarmos os mesmos dados nesse software, obtivemos a seguinte figura:

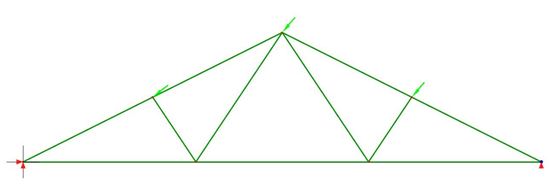

Figura 5: Modelo simulado no LISA FEA

Deslocamento em X [m]			
Nó	SOFTWARE	LISA	DIFF (%)
1	0,00000000	0,000000000	0,00000000
2	0,00097028	0,000970275	-0,00049119
3	0,00034743	0,000347429	-0,00041118
4	0,00025083	0,000250833	0,00135105
5	0,00053943	0,000539429	-0,00026483
6	-0,00031315	-0,000313150	0,00001285
7	0,00112457	0,001124571	0,00012703

Deslocamento em Y [m]			
Nó	SOFTWARE	LISA	DIFF (%)
1	0,00000000	0,0000000	0,00000000
2	-0,00416384	-0,004163841	0,00002196
3	-0,00450318	-0,004503184	0,00008334
4	-0,00428701	-0,004287010	0,00000992
5	-0,00447852	-0,004478519	-0,00002230
6	-0,00410209	-0,004102086	-0,00009892
7	0,00000000	0,0000000	0,00000000

Figura 6: Tabelas comparativas entre software e LISA

Com os resultados comparativos em mãos, podemos observar que o software desenvolvido chegou em resultados muito próximos para os deslocamentos, possuindo uma diferença máxima em módulo de 0,00135105%.

CONCLUSÃO

Com a construção do nosso software, conseguimos validar sua performance utilizando o método de Gauss, já que a diferença entre os vetores obtidos em ambas abordagens são praticamente iguais. Em relação a sua melhoria na performance, sabemos que um input de maior dimensionalidade irá aumentar a exigência de memória e tempo em seu cálculo, devido a sua complexidade $O(n^2)$, o que nos leva a poder reestruturar a lógica do nosso algoritmo em questão de eficiência. Quanto ao código em si, algumas otimizações em loops e cálculos não aperfeiçoados seriam efetivas e poderiam trazer benefícios em tempo de processamento.

Portanto, podemos concluir que foi construído um software com uma alta fidelidade à realidade e a outros softwares já vigentes no mercado utilizando os conteúdos ensinados em aula.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] SILVA, A. C. L. D; MIRANDA, Walzenira Parente. O papel do engenheiro no uso de softwares para cálculo estrutural.

[2] SOUZA, Diego Aparecido. Implementação de software computacional para análise estática de treliças planas via Método de Elementos Finitos.

[3] ARCELORMITTAL. Você sabe o que é uma treliça? . Disponível em: https://blog.arcelormittal.com.br/trelica/. Acesso em: 5 jun. 2022.