Projeto de Software para análise de treliças planas

Software de Solução Numérica para Elementos Finitos de Uma Treliça 2D

Insper, Engenharia da Computação, Quinto Semestre, 2022

Giancarlo Vanoni, Matheus Oliveira, Nívea Abreu

https://github.com/gianvr/APS4-MecSol.git

Introdução

O desenvolvimento de softwares computacionais que realizem análises estáticas e dinâmicas de treliças planas tem grande importância nos dias atuais, já que permite a simulação de estruturas e possíveis situações sem nenhum custo material, além de proporcionar maior segurança e integridade a obra e todos seus envolvidos. Segundo Rovere & Schneider (2003)[1], o avanço computacional das últimas décadas possibilitou o desenvolvimento de programas computacionais para análise estrutural de estruturas contínuas e reticuladas, o que foi de grande valia para simulações mais abrangentes e próximas da realidade. Segundo supracita Silva & Borges[2], para que um software seja considerado educacional, ele deve ser preferencialmente livre, permitir o desenvolvimento continuado, possuir fonte disponível e ser bem documentado (ORMONDE & SOUZA, 2012)[3]. Dessa maneira a construção de um Software que torne a análise de treliças planas que seja acessível e prática para Engenheiros e público em geral, se torna essencial nos dias atuais, seja para garantia de segurança, ou para a maior aproximação de projetos físico e computacionais para outros escopos.

Dessa forma, primou-se pela utilização da linguagem de programação Python, para leitura de um arquivo de entrada Excel com todos os dados correlatos a estrutura (nós, elementos, restrições, apoios e forças de carregamento). Após a leitura desta entrada, realiza-se diferentes cálculos via aproximação de elementos finitos, especificamente neste caso o método de Gauss-Seidel e/ou Jacobi, para obtenção aproximada de valores de Reações, deslocamentos e esforços internos atuantes na estrutura.

Estruturação do Software

O software foi estruturado em uma classe chamada Solver.

A classe é composta por 4 métodos públicos: import_file, solve_problem, generate_outfile, plot; que tem funções respectivas de leitura dos dados de entrada, realizar a solução numérica da estrutura, geração do arquivo de saída com resultados e plotagem gráfica da estrutura. O método público solve_problem faz utilização interna de mais de 6 métodos privados que realizam cálculos intermediários, como matriz de rigidez de cada elemento, matriz global de superposição, deslocamento nodais, esforços atuantes e reações.

Egram utilizadas duas soluções numéricas para o problema Jacobi o Gauss Soidol

Validação do Software

Para a etapa de validação, foram confrontados resultados obtidos pelo Software construído pela equipe e pelo Software Lisa Fea, que é uma referência na solução de problemas numéricos de estruturas estáticas.

Para validar a leitura dos parâmetros internos, plotou-se a estrutura inicial com suas restrições aplicadas e o vetor de carregamento nos respectivos nós, conforme se averigua na figura 2.

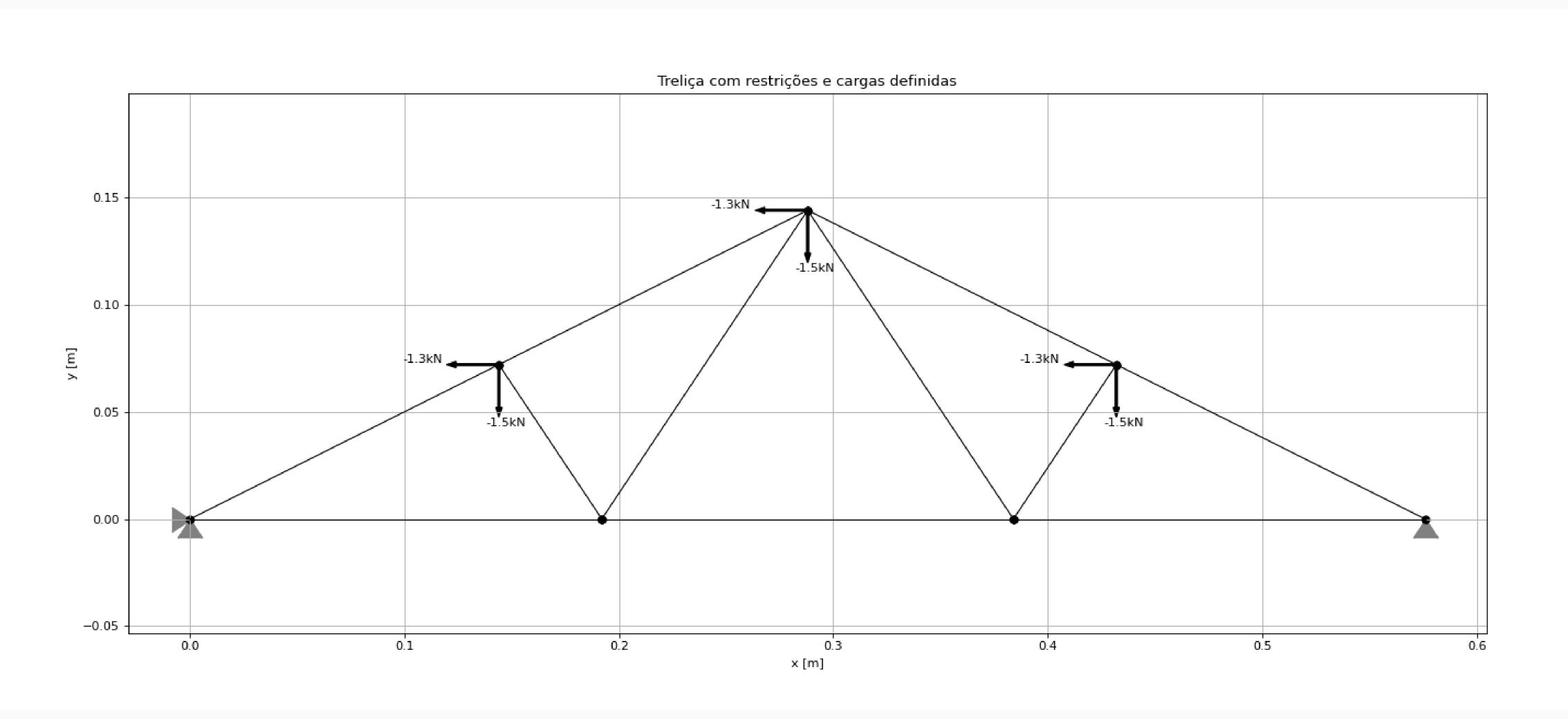
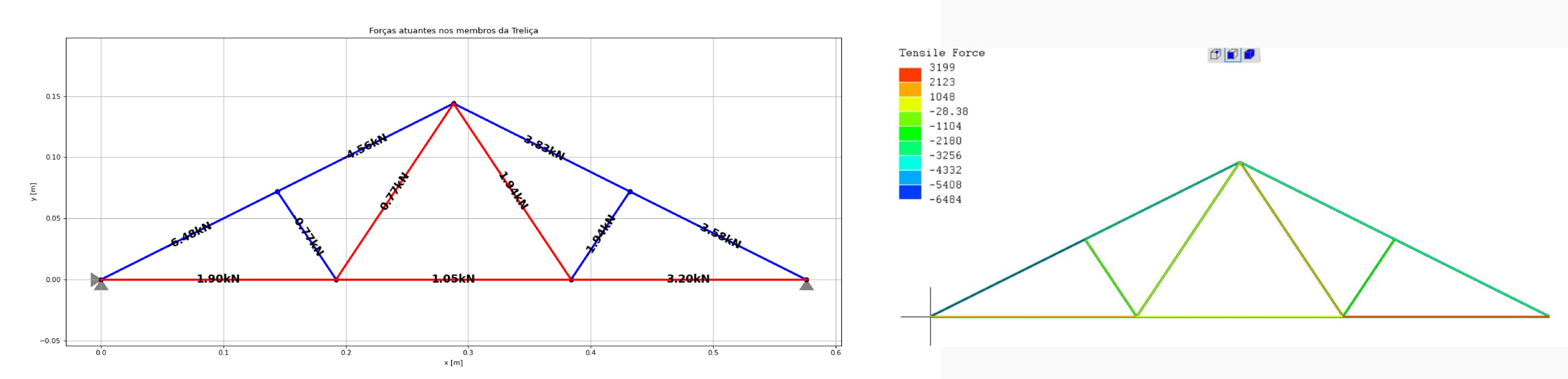


Figura 2 - Estrutura inicial com restrições e vetor de carregamento aplicados

Posteriormente, calcula-se os valores de saída do Software conforme organograma do Software na sessão anterior. Para conferência de sua razoabilidade com valores de simulação, comparou-se os resultados análiticos e gráficos à saída do Software Lisa Fea. Realizou-se a plotagem dos esforços atuantes na estrutura do Software, e comparou-se com o simulado pelo Lisa, conforme mostram as figuras 3 e 4.



Figuras 3,4 - Esforços atuantes simulados via Software e esforços obtidos via Lisa.

Constata-se uma boa semelhança dos resultados gráficos, visto que os sinais de tensão e compressão se mostram acertados e com valores bem próximos aos esperados pelo Lisa. Concomitantemente, simulou-se graficamente a a deformação da estrutura, obtendo-se resultados coerentes entre os dois softwares e sendo novamente uma boa validação da eficiência de análise do Software desenvolvido,

Conclusão

Doravante validação e estruturação do Software construídos apresentados anteriormente, constata-se que o Software cumpre satisfatoriamente as métricas de solução para estruturas planas bidimensionais. Conforme averiguado, os valores obtidos pelo software tiveram diferença efetiva inferior a 0.02% em relação aos valores gerados por Softwares consolidados como Lisa Fea, ilustrando a eficácia dos passos adotados para a construção do programa computacional, tal qual a utilização adequada dos métodos de resolução numérica de Gauss-Seidel e Jacobi. Conclui-se assim que, para as condições enunciadas pelo problema proposto, o software cumpre adequadamente todos os requisitos com alta taxa de confiabilidade de resultados, alem de gerar plotagens que tornem ainda mais intuitivos os resultados gerados, permitindo melhor validação e análise posterior.

Para possíveis melhorias a serem adotadas na estrutura do Software, encontram-se dentre as principais:

- Criação de uma interface gráfica para seleção de pontos nodais, elementos e dados do material, facilitando a entrada de dados;
- Saída de dados ainda mais complexos, permitindo simulações animadas da estrutura e mudanças mais responsivas;
- Resolução de problemas em 3 dimensões;
- Resolução e análise de estruturas com possibilidade de movimentação em eixo de rotação ou translação, abordando uma maior gama de problemas.

Destaca-se ainda que o Software encontra-se em seus estados iniciais de construção, o que pode ocasionar em erros de soluções de estruturas com maior grau de complexidade, além de dificultar relativamente a entrada de dados pelos usuários.

Bibliografia e Referências

[1] Rovere, H. L. L., & Schneider, A. F., 2003. ANEST — Programa Educacional para Análise de Estruturas Reticuladas. XIV Simpósio Brasileiro de Informática na Educação.

[2] BORGES, Rafael de Assis; SILVA, Sebastião Simão da. ESTUDO DE TRELIÇAS PLANAS E ESPACIAIS UTILIZANDO A LINGUAGEM DE PROGRAMAÇÃO PYTHON E O SOFTWARE VTK. **XXXVII Iberian Latin American Congress On Computational Methods in Engineering**, Cajazeiras, Paraíba, Brasil, 2016.