Projeto Mecânica dos Sólidos

João Pedro Pieroni de Castro; Lucas Astur; André Toyama; Felipe Frid Buniac

Engenharia da Computação, Instituto de Ensino e Pesquisa INSPER

Objetivos

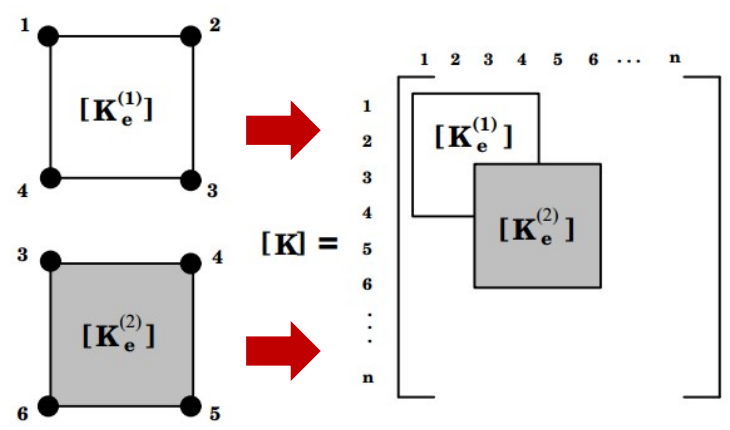
Com base nos conceitos vistos em aula, desenvolver um programa em Matlab que leia dados de entrada da treliça, fornecido pelo usuário e analise tensão, deformação e deslocamento.

Introdução

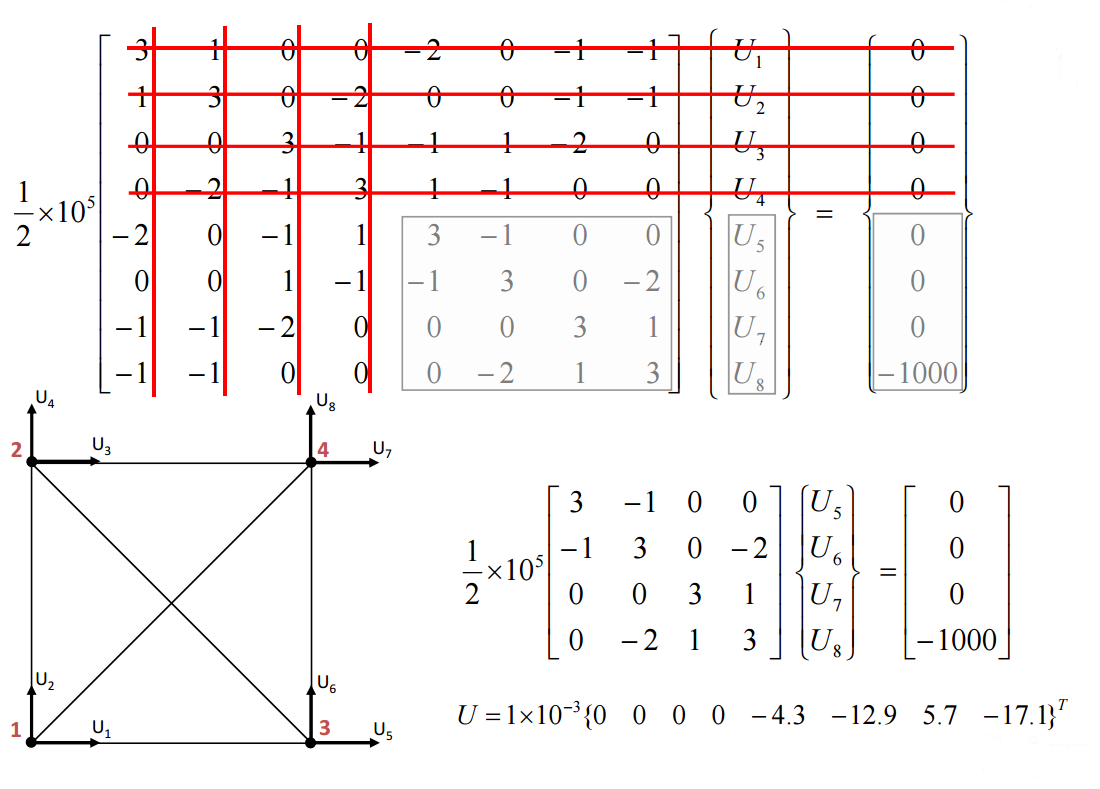
Em aula vimos que podemos analisar tração/compressão de uma treliça plana usando o método dos elementos finitos que resulta em um sistema de equações algébricas que pode ser transformado em uma matriz e então calculando a inversa da matriz global para achar o deslocamento nodal. Posteriormente foi observado que existem métodos mais eficientes de se calcular computacionalmente o deslocamento, através de métodos como o Jacobi ou o Gauss-Seidel, sendo esta última a mais eficiente. Com base nestas observações, criamos um programa de análise e cálculo de treliças planas que recebe dados acerca da estrutura da treliça e devolve um arquivo contendo as forças de reação, deslocamentos nodais, deformação e tensão dos elementos da treliça.

Metodologia

Primeiramente, pelo método de elementos finitos foi calculada a matriz de flexibilidade de cada nó da treliça e a matriz de rigidez de cada elemento. A partir da matriz de flexibilidade, todas as matrizes de rigidez de cada elemento de barra foram superpostas em uma apenas uma matriz, a matriz global. Após achar a matriz global, foram aplicadas condições de contorno de acordo com os nós que estão fixados por apoios, de acordo com as figuras a seguir:

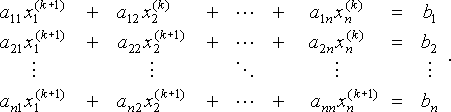


*Fig. 1 – Superposição das matrizes de cada elemento em uma matriz flobal.*

**

*Fig. 2 – Aplicação de cond. de contorno em uma matriz global genérica (com travas nos graus de liberdade 1,2,3 e 4).*

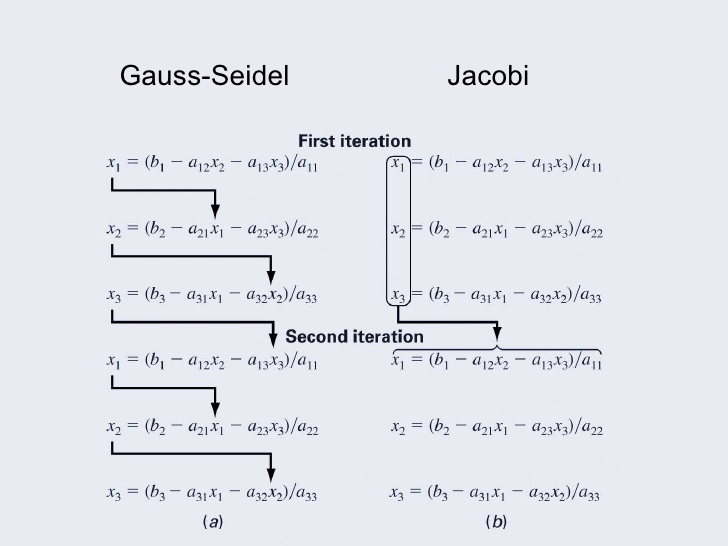
Como a própria figura 2 evidencia, os deslocamentos nodais (matriz U) podem ser encontrados multiplicando o inverso da matriz global com o vetor de forças. No entanto, essa solução não se demonstra muito eficiente em computação. Com isso, foi aplicado o método de Gauss-Seidel para descobrir os deslocamentos nodais, método no qual “chuta-se” um valor de deslocamento nodal usando um método sistemático para obter a raiz da equação [1] como visto na figura 3.



*Fig. 3 – Método de Gauss-Seidel*

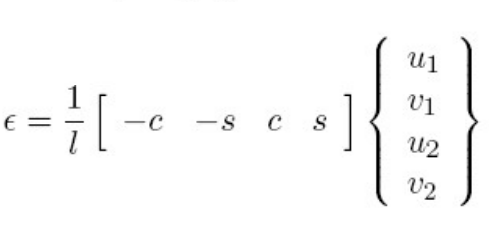
Cada valor de “x” calculado no método é imediatamente usado na próxima equação para determinar o próximo “x”.

Outro método interessante, mas não tão eficiente quanto, é o método de jacobi. A diferença entre o Gauss-Seidel e o Jacobi é que no Jacobi o valor de x é descoberto em todas as equações da iteração e o resultado desses “x” é usado na próxima iteração como visto na figura 4, logo abaixo.

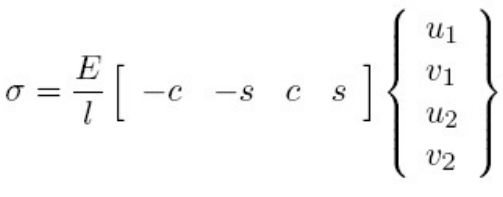


*Fig. 4.*

Uma vez descoberto os deslocamentos nodais, podemos utiliza-los para calcular a deformação e a tensão de cada barra, dadas pelas fórmulas:

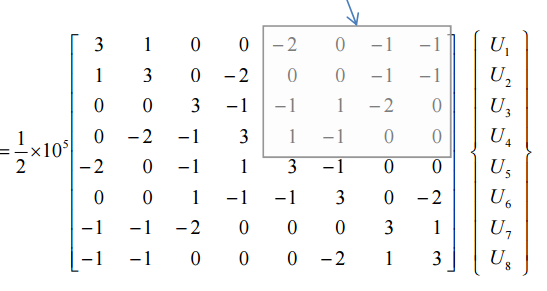


*Eq. 1 – Deformação específica*



*Eq. 2 – Tensão*

Para finalizar o cálculo, podemos calcular as forças de reação dos graus de liberdade travados ao aplicar condições de contorno na matriz global, de modo a ignorar colunas cujos graus de liberdade estejam travados e linhas, cujos graus de liberdade estejam livre, assim como na figura 5:



*Fig. 5 – pegando matriz utilizada no cálculo das reações após aplicação de condições de contorno (matriz genérica).*

Uma vez obtida o Kg pro calculo da reação, basta multiplica-lo pela matriz de deslocamentos nodais para obtermos as forças de reação em cada grau de liberdade.

Implementação do código:

O código enviado em anexo possui todas os trechos devidamente comentados para seu completo entendimento pode ser encontrado neste link:  
[*https://github.com/AToyama/ProgramTuss-TermoMecSol/tree/master/ProjetoFinal*](https://l.facebook.com/l.php?u=https%3A%2F%2Fgithub.com%2FAToyama%2FProgramTuss-TermoMecSol%2Ftree%2Fmaster%2FProjetoFinal&h=ATOmv-_J6xfgt5Ajbrfo1F0JpAyrvhKGoS5eHET9mtrxRJq9gwyN4Soim5O6zwD0ZZAfesg0rlJXzVW46rYRLmeGZfNS-HYkft6oMz8vkNWQqYtykUkAvcw01Su5uR9EwOq4)  
Nele contém os seguintes arquivos:  
Projeto\_final.m --> Código do projeto.  
GaussSidel.m --> implementação do Gauss-Seidel utilizado no projeto.  
In.fem --> Arquivo de entrada para a treliça designada ao nosso grupo no documento do projeto.

Resultados:

De acordo com os arquivos de saída gerados pela implementação do Truss Solver em MatLab, podemos comprovar que o programa funciona e está gerando os resultados esperados para uma treliça qualquer.  
Com os cálculos conseguimos então determinar deformação e a tensão de cada barra e sabemos que chegamos a uma solução quando pelo método numérico, a diferença do vetor de deslocamento entre iterações, ou seja, o erro é menor que a tolerância máxima determinada pelo usuário de acordo com sua aplicação.  
Então podemos determinar quanto de esforço cada barra pode aguentar através da tensão admissível, deformando-se sem romper.

Referencias Bibliográficas

## [1]Chapra, Steven. Numerical Methods for Engineers 6º. Ed, pág. 300