



PROGRAMA PARA ANÁLISE NÃO-LINEAR FÍSICA DE TRELIÇAS PLANAS

Marques, Guilherme José Bandeira Jovino

Viniski, Bianca Mendes Carneiro

Nogueira, Breno Ribeiro

Ribeiro, Paulo Marcelo Vieira

guilhermejbjm@gmail.com

biancaviniski@hotmail.com

breno_nogueira@hotmail.com

paulo.vribeiro@ufpe.br

Universidade Federal de Pernambuco – UFPE

Centro de Tecnologia e Geociências

Departamento de Engenharia Civil

Rua Acadêmico Hélio Ramos, s/n, 50740-530, Pernambuco, Recife, Brasil

Abstract. *Análises estruturais usuais são realizadas com a hipótese de linearidade física do material componente dos elementos estruturais. Uma visão mais abrangente do problema surge com a consideração de relações constitutivas definidas por funções arbitrárias. O presente trabalho tem como finalidade a apresentação de um código desenvolvido em MATLAB, com interface gráfica amigável, para a análise estática de treliças planas considerando a não-linearidade física do material em questão. O pré-processamento de treliças para níveis progressivos de complexidade foi realizado com o auxílio do software GiD e posteriormente exportado ao programa principal, que resolve um sistema de equações não-lineares por meio de métodos aproximados, tais como: método de Newton-Raphson, método de Newton-Raphson modificado e método incremental. O programa apresenta de forma clara e didática os efeitos das não-linearidades em estruturas. Os resultados foram posteriormente comparados aos obtidos com códigos comerciais em elementos finitos e confirmam a validade das rotinas propostas.*

Keywords: *treliças planas, elementos finitos, não-linearidade física*

1 INTRODUÇÃO

O alto nível de detalhamento nas análises de modelos estruturais vem crescendo e exigindo dos profissionais ferramentas adequadas. A análise não linear de estruturas em geral é pouco utilizada nos softwares comerciais e com aplicação geralmente complexa, limitando a utilização a usuários mais qualificados.

A grande maioria dos dimensionamentos utilizados tem como base teórica a Lei de Hooke, que exige a linearidade física dos materiais a fim de simplificar os cálculos, deixando uma margem de segurança satisfatória. Porém, a análise linear, apesar de simplificar os cálculos matemáticos, não demonstram fisicamente o comportamento físico real dos materiais. Como exemplo, podemos tomar as deformações nodais em estruturas treliçadas. A análise linear garante que a rigidez das peças não muda, porém, tal rigidez depende da deformação da mesma, sendo variável ao longo do processo da própria deformação.

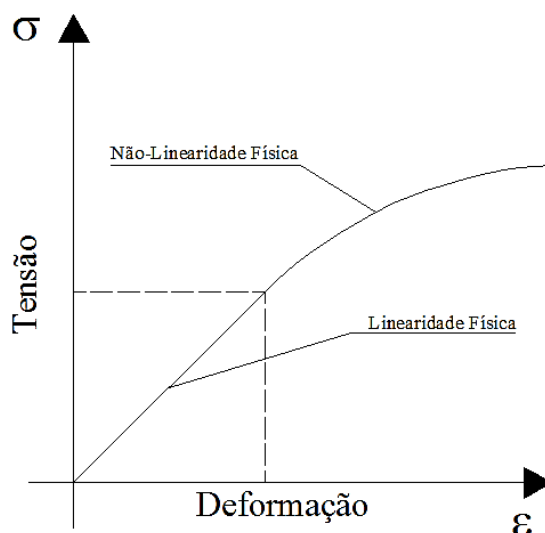


Figura 1. Gráfico Tensão x Deformação

Segundo o gráfico representado na Fig. 1, tomando por base um material elástico, vemos que o mesmo tem dois comportamentos ao longo de sua deformação. Inicialmente a peça, submetida a um esforço, deforma seguindo um regime elástico linear. Posteriormente, esse regime passa a ser elástico não linear, ou seja, os deslocamentos passam a seguir uma função não linear dependente de seu módulo de elasticidade que por sua vez é função das próprias deformações.

Este projeto tem como objetivo o desenvolvimento de um software capaz de analisar estruturas planas treliçadas, levando em consideração a não linearidade física do material. Também terá como proposta trazer para o usuário uma entrada gráfica simples e fácil de se utilizar, com introdução de dados feita através do software GiD (2015). A modelagem do programa será feita pelo software MATLAB (2009), com algoritmos baseados nos métodos de Newton-Raphson para resolver os sistemas não lineares de qualquer problema prescrito.

2 ANÁLISE TRELIÇAS PLANAS PELO MÉTODO DA RIGIDEZ

2.1 ANÁLISE LINEAR

Antes de entrarmos no raciocínio da não linearidade física dos materiais, faremos uma revisão na análise matricial linear de treliças planas. Segundo Vaz (2011), a análise matricial de estruturas planas reticuladas sistematizou as operações matemáticas da análise de estruturas fazendo o uso de álgebra matricial que opera vetores e matrizes.

2.1.1 COORDENADAS E CONECTIVIDADES

Inicialmente é feito uma compatibilidade das barras que irão compor a estrutura treliçada e as coordenadas cartesianas dos nós existentes. A figura 2 exemplifica uma treliça plana contendo seus elementos nodais, elementos lineares e suas respectivas conectividades. Tomaremos como exemplo a figura a seguir,

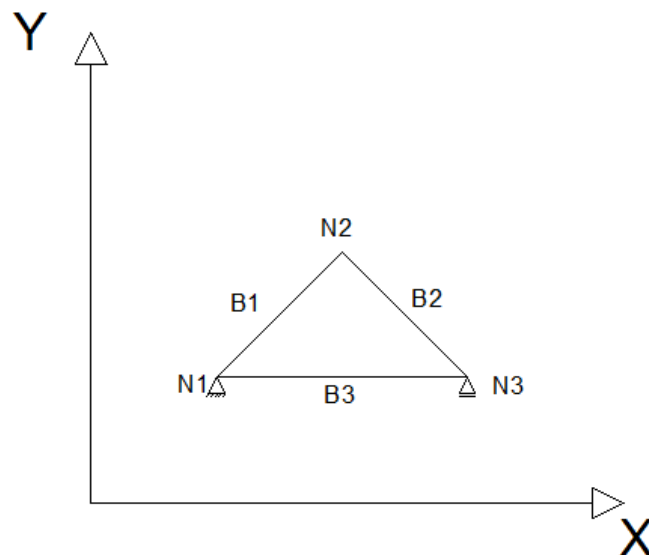


Figura 2. Esquema da treliça composta por suas conectividades e nós

Coordenadas Nodais: N1 = (0,0)

N2 = (1,1)

N3 = (2,0)

Conectividades das barras: B1 = N1-N2

B2 = N2-N3

B3 = N3-N1

2.1.2 MATRIZ DE RIGIDEZ LOCAL DAS BARRAS

Isolando uma das barras que compõe a estrutura da treliça plana:

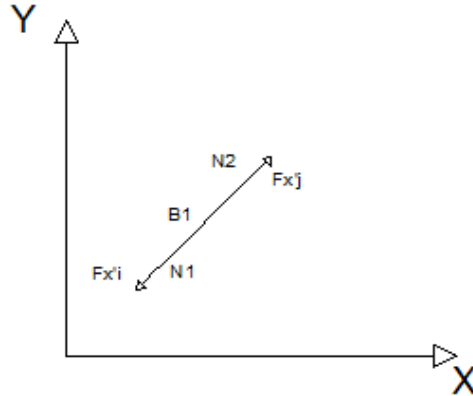


Figura 3. Esquema das barras isoladas com suas respectivas forças

De acordo com Vaz (2011), toda a sistematização se baseia na ideia de sistema local e global de coordenadas. A figura 3 representa barra “B1” isolada das demais, suas conectividades, “N1” e “N2”, bem como os esforços atuantes nos nós inicial e final da barra estudada, tais esforços seguindo a direção do eixo local do elemento. Podemos calcular a matriz de rigidez local de cada barra sendo:

$$\Delta L = \left(D_x^L \right)_j - \left(D_x^L \right)_i \quad (1)$$

$$\Delta L = \frac{P \cdot L}{E \cdot A} \leftarrow P = \frac{\Delta L \cdot E \cdot A}{L} \leftarrow \left(F_x^L \right)_i = \frac{-E \cdot A \cdot \Delta L}{L} \quad (2)$$

$$\Delta L = \frac{P \cdot L}{E \cdot A} \leftarrow P = \frac{\Delta L \cdot E \cdot A}{L} \leftarrow \left(F_x^L \right)_j = \frac{E \cdot A \cdot \Delta L}{L} \quad (3)$$

Substituindo a equação da deformação linear da barra (1) na equação de Hooke representada por (2) e (3), chegamos ao resultados que alinham esforços e deslocamentos nodais de cada barra (4) e (5), seja ele atuantes no nó inicial ou final do elemento.

$$\left(F_x^L \right)_i = \frac{-E \cdot A \cdot \Delta L}{L} \leftarrow \left(F_x^L \right)_i = \frac{-E \cdot A \cdot \left[\left(D_x^L \right)_j - \left(D_x^L \right)_i \right]}{L} \leftarrow \left(F_x^L \right)_i = \frac{-E \cdot A \cdot \left(D_x^L \right)_j}{L} + \frac{E \cdot A \cdot \left(D_x^L \right)_i}{L} \quad (4)$$

$$\left(F_x^L \right)_j = \frac{E \cdot A \cdot \Delta L}{L} \leftarrow \left(F_x^L \right)_j = \frac{E \cdot A \cdot \left[\left(D_x^L \right)_j - \left(D_x^L \right)_i \right]}{L} \leftarrow \left(F_x^L \right)_j = \frac{E \cdot A \cdot \left(D_x^L \right)_j}{L} - \frac{E \cdot A \cdot \left(D_x^L \right)_i}{L} \quad (5)$$

Reescrevendo as equações (4) e (5), chegamos ao sistema matricial de esforços nodais de cada elemento (6):

$$\begin{bmatrix} \left(F_x^L \right)_i \\ \left(F_y^L \right)_i \\ \left(F_x^L \right)_j \\ \left(F_y^L \right)_j \end{bmatrix} = \begin{bmatrix} \frac{E \cdot A}{L} & 0 & \frac{-E \cdot A}{L} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{-E \cdot A}{L} & 0 & \frac{E \cdot A}{L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \left(D_x^L \right)_i \\ \left(D_y^L \right)_i \\ \left(D_x^L \right)_j \\ \left(D_y^L \right)_j \end{bmatrix} \leftarrow F_1^L = S_1^L \cdot D_1^L \quad (6)$$

2.1.3 MATRIZ DE ROTAÇÃO DOS EIXOS

Para simplificar os cálculos e direcionar o raciocínio para um único eixo, foi estabelecido que todos as variáveis do cálculo deveriam seguir os eixos globais. A figura 4 mostra o esquema de rotação dos eixos locais para os eixos globais, conectando seus componentes através da angulação teta.

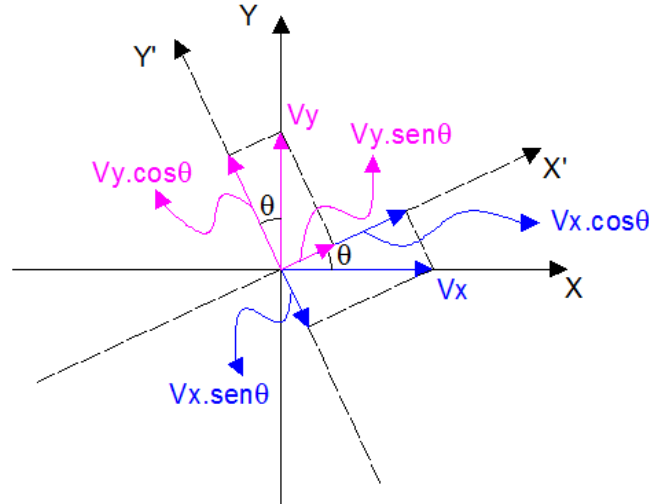


Figura 4. Esquema das barras isoladas com suas respectivas forças

Analisando a figura 4, podemos concluir que os vetores locais são representados por um somatório vetorial do produto de senos e cossenos pelos vetores em coordenadas globais (7) e (8).

$$V_x^L = V_x \cdot \cos \theta + V_y \cdot \sin \theta \quad (7)$$

$$V_y^L = -V_x \cdot \sin \theta + V_y \cdot \cos \theta \quad (8)$$

Analogamente aos vetores acima, podemos utilizar a rotação para os deslocamentos nodais, representados pelas equações (9) e (10):

$$D_x^L = D_x \cdot \cos \theta + D_y \cdot \sin \theta \quad (9)$$

$$D_y^L = -D_x \cdot \sin \theta + D_y \cdot \cos \theta \quad (10)$$

Reescrevendo a transformação de eixos em linguagem matricial chegamos a matriz RT de rotação (11):

$$RT = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \\ 0 & 0 & -\sin \theta & \cos \theta \end{pmatrix} \quad (11)$$

2.1.4 MATRIZ DE RIGIDEZ GLOBAL DAS BARRAS

Seguindo o raciocínio de utilizar todos os elementos em coordenadas globais, podemos reescrever a matriz de rigidez global das barras multiplicando os vetores força e deslocamentos pela matriz de rotação. Substituindo os esforços e deslocamentos locais pelos seus respectivos em eixo global chegamos a equação (12). Multiplicando-se ambos os lados da igualdade pela inversa da matriz de rotação (13) e isolando os vetores força e deslocamento, chegamos a equação matricial de rigidez global de cada barra (14).

$$RT \cdot F = S^L \cdot RT \cdot D \quad (12)$$

$$RT^{-1} \cdot RT \cdot F = RT^{-1} \cdot S^L \cdot RT \cdot D \quad (13)$$

$$F = RT^{-1} \cdot S^L \cdot RT \cdot D \quad (14)$$

Analogamente à matriz de rigidez local das barras, podemos escrever a matriz de rigidez global das barras como representado na equação (15), substituindo-se o produto $RT^{-1} \cdot S^L \cdot D$ pelo índice S.

$$F = S \cdot D \quad (15)$$

2.1.5 MATRIZ DE RIGIDEZ GLOBAL DA ESTRUTURA. CONTRIBUIÇÃO DAS BARRAS.

Para definir a contribuição que cada barra terá na matriz de rigidez da estrutura teremos que seguir duas condições: A condição de equilíbrio e a condição de compatibilidade

A **condição de equilíbrio** em cada nó da estrutura, estabelece que as cargas externas aplicadas nos nós devem ser iguais a soma das solicitações nos extremos das barras que concorrem neste nó.

Já a **condição de compatibilidade** estabelece que as barras que convergem para o mesmo nó têm seus deslocamentos nodais equivalentes.

Com as duas condições satisfeitas, podemos montar a matriz de rigidez da estrutura com as respectivas participações de todas as barras que compõem a treliça.

Na análise linear de treliças planas utilizando o método da rigidez, o sistema final de equações é um sistema linear simples, com $\underline{x}$ incógnitas (onde x é duas vezes o número de nós) e $\underline{x}$ equações. Após atribuir as condições de contorno dos apoios o sistema é calculado de maneira simplificada, não necessitando o uso de cálculos numéricos para se obter os resultados desejados.

2.2 ANÁLISE NÃO-LINEAR

Segundo Macedo (2014), a não linearidade física caracteriza-se por causar relações não lineares entre a tensão e a deformação e pelo fato de a análise estrutural depender do caminho ou histórico de deformação do material. Na análise linear de treliças planas, levamos em consideração, pela lei de Hooke, que o módulo de elasticidade das peças estruturais era uma constante. A análise não-linear não segue essa linha de raciocínio. O módulo de elasticidade é ajustado, segundo uma equação não-linear, à medida que a estrutura vai se deformando. A ideia do cálculo não-linear foi ajustar a equação da tensão (16) à uma equação análoga à lei de Hooke, onde módulo de elasticidade não é apenas em uma constante e sim uma equação que seja função dos deslocamentos nodais que iremos calcular. Para o cálculo da equação não-linear, estamos utilizando polinômios do terceiro grau (17) que são funções do deslocamento percentual (18). Tal polinômio tem a finalidade de se aproximar da curva tensão-deformação, empregada segundo uma série de pontos amostrais. Substituindo-se a equação (18) na equação (17), alinhamos os deslocamentos nodais à equação de tensões (19) e podemos substituir o polinômio na equação (16). Fazendo os ajustes necessários em (20) e (21) e definindo o polinômio do segundo grau da equação (21) como o módulo de elasticidade não-linear, chegamos à equação (22), análoga à análise linear anteriormente estudada.

$$\sigma(\varepsilon) \cdot A = F \quad (16)$$

$$\sigma(\varepsilon) = C1 \cdot \varepsilon^3 + C2 \cdot \varepsilon^2 + C3 \cdot \varepsilon \quad (17)$$

$$\varepsilon = \frac{d}{L} \quad (18)$$

$$\sigma(d) = C1 \cdot \left(\frac{d}{L}\right)^3 + C2 \cdot \left(\frac{d}{L}\right)^2 + C3 \cdot \frac{d}{L} \quad (19)$$

$$\left[C1 \cdot \left(\frac{d}{L}\right)^3 + C2 \cdot \left(\frac{d}{L}\right)^2 + C3 \cdot \frac{d}{L} \right] \cdot A = F \quad (20)$$

$$\left[C1 \cdot \left(\frac{d}{L}\right)^2 + C2 \cdot \left(\frac{d}{L}\right)^1 + C3 \right] \cdot \frac{A \cdot d}{L} = F \quad (21)$$

$$\frac{E(d) \cdot A \cdot d}{L} = F \quad (22)$$

Como visto anteriormente, transformamos o módulo de elasticidade constante em um polinômio que é função da própria deformação a ser calculada e chegamos a uma equação análoga à função linear calculada no início do artigo. Utilizando os mesmos artifícios para o cálculo linear e os métodos numéricos a serem estudados, chegamos as deformações em treliças planas levando em consideração o regime não-linear físico do material.

3 MÉTODOS NUMÉRICOS PARA SOLUÇÃO DE EQUAÇÕES NÃO-LINEARES

Algumas equações não lineares necessitam de métodos aproximados para o cálculo de seus resultados. Existem, na literatura atual, diversos métodos numéricos aproximados que nos dão resultados com baixa margem de erros. Ivančo (2011) apresenta diversos métodos numéricos capazes de aproximar de maneira satisfatória equações não linear. Nesse artigo foram estudados três métodos: o método incremental, o método de Newton-Raphson e o método de Newton-Raphson modificado.

3.1 MÉTODO INCREMENTAL

O método incremental consiste na divisão do carregamento total em pequenos incrementos de cargas (24) e (25). Para cada incremento é calculado o incremento de deformação (26) levando em consideração a direção da reta tangente naquele ponto (23). As deformações são ajustadas a cada interação (27), ajustando assim a reta tangente no novo ponto. A rotina se repete até que o número de interações seja atingido. Seja: $P(x) = f(x)$, calcula-se:

$$K(x) = \frac{d}{dx} f(x) \quad (23)$$

$$\Delta P \quad (24)$$

$$i = \frac{P}{\Delta P} \quad (25)$$

$$\Delta x_{i+1} = \frac{\Delta P}{K(x_i)} \quad (26)$$

$$x_{i+1} = x_i + \Delta x_{i+1} \quad (27)$$

A figura 5 representa o método graficamente.

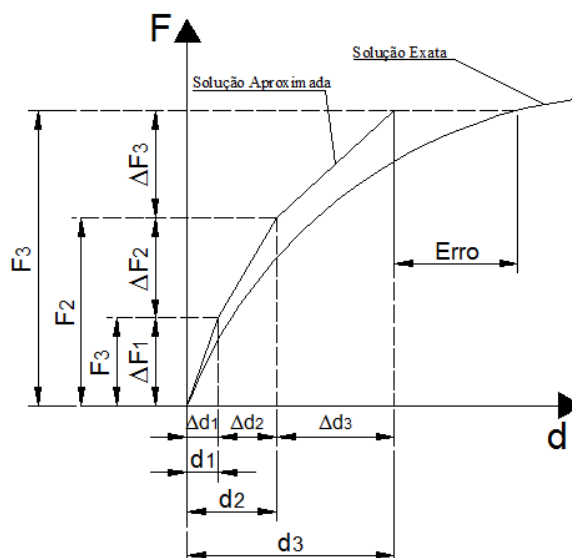


Figura 5. Esquema de interação do método incremental. Fonte: Vladimír Ivančo ,2011.

Segundo o gráfico representado na Fig. 5, o método incremental tem como solução um valor aproximado, porém, sua precisão depende do número de interações previsto. Quanto menor for o incremento de carga usado, menor serão os erros resultantes deste método.

3.2 MÉTODO DE NEWTON-RAPHSON

O método de Newton-Raphson, diferente do método incremental, não depende dos incrementos de cargas para se calcular valores aproximados. Supondo conhecermos um valor de deformação inicial x_0 , calculamos a reta tangente à curva no ponto inicial (28). Geometricamente se dividirmos o cateto adjacente pela tangente do ângulo chegamos ao cateto oposto, a equação (29) chega ao deslocamento causado pela força P se seguido a direção da tangente calculada. Utilizando tal deslocamento na equação original (30), chegamos a um valor menor do que o da força P, diferindo por um resíduo calculado na equação (31). Dividindo o resíduo encontrado pela tangente no novo ponto calculamos o incremento de deslocamento (32) que somado ao deslocamento anterior (33) se aproxima ao deslocamento exato causado pela força P.

$$K(x) = \frac{d}{dx} f(x) \quad (28)$$

$$x_1 = \frac{P}{K(x_1)} \quad (29)$$

$$R(x_1) = F(x_1) \quad (30)$$

$$r(i) = P - R(i) \quad (31)$$

$$\Delta x_i = \frac{r(i)}{K(x_i)} \quad (32)$$

$$x_{i+1} = x_i + \Delta x_i \quad (33)$$

A iteração irá parar quando o incremento de deformação tender a uma certa tolerância imposta. A figura 6 exemplifica o método de Newton-Raphson graficamente.

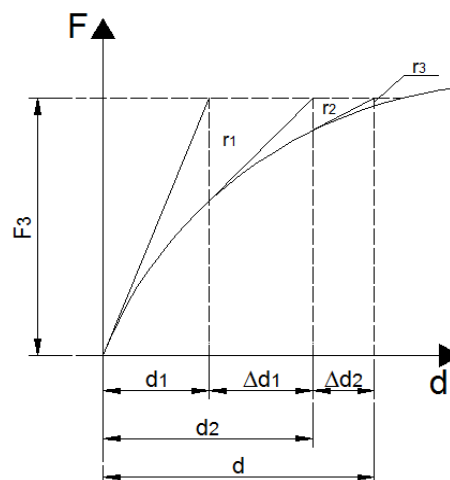


Figura 6. Esquema de iteração do método de Newton-Raphson. Fonte: Ivančo (2011).

3.3 MÉTODO DE NEWTON-RAPHSON MODIFICADO

O método de Newton-Raphson modificado tem apenas uma diferença se comparado ao método de Newton-Raphson convencional: a tangente da curva não é atualizada ao longo das interações (34). Depois de definida qual tangente irá governar o método, essa tangente não é atualizada, sendo mantida até o final do cálculo quando a tolerância for satisfeita. O cálculo da deformação inicial segue o mesmo raciocínio geométrico do método anterior (35). Calculada a força que a deformação inicial causaria (36) e seu resíduo para atingir a força de cálculo P (37), os deslocamentos são atualizados sempre seguindo a direção da tangente adotada no início do método (38) e (39).

$$K = \frac{d}{dx} f(x_0) \quad (34)$$

$$x_1 = \frac{P}{K} \quad (35)$$

$$R(x_1) = F(x_1) \quad (36)$$

$$r(i) = P - R(i) \quad (37)$$

$$\Delta x_i = \frac{r(i)}{K} \quad (38)$$

$$x_{i+1} = x_i + \Delta x_i \quad (39)$$

O critério de parada do método de Newton-Raphson modificado também é o mesmo do método anterior, sendo a tolerância definida inicialmente. A figura 7 esquemática representa graficamente o método de Newton-Raphson modificado.

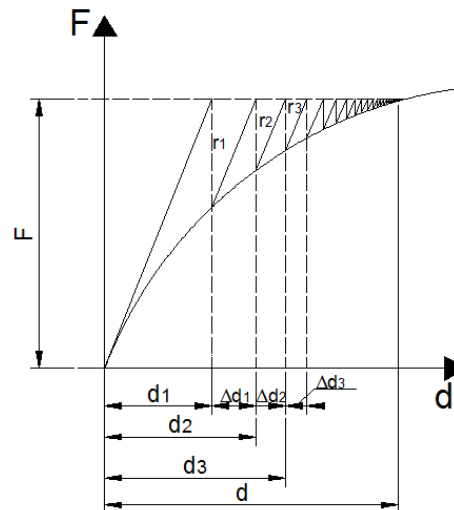


Figura 7. Esquema de iteração do método de Newton-Raphson modificado. Fonte: Ivančo (2011).

O método de Newton-Raphson Modificado, apesar de não atualizar a tangente do sistema, converge ao resultado com mais interações do que o método de Newton-Raphson, devendo ser feito um estudo mais detalhado para saber qual método converge mais rápido ao resultado esperado.

4 MÉTODOS NUMÉRICOS PARA SOLUÇÃO DE SISTEMAS DE EQUAÇÕES NÃO-LINEARES

4.1 MÉTODO DE NEWTON-RAPHSON PARA SISTEMAS DE EQUAÇÕES NÃO LINEARES

Como visto antes, sabemos que o método de Newton-Raphson depende da reta tangente à curva em cada interação. No caso de sistemas não lineares o método é semelhante, porém ao invés de calcularmos a reta tangente à curva, calcularemos o hiperplano tangente ao politopo que o sistema linear determina. Tal hiperplano tangente é chamado de Matriz Jacobiana.

Tomando como exemplo, seja F_n o vetor composto pelas equações não lineares de um sistema de seis incógnitas (41), temos:

$$F_n = \begin{pmatrix} F_1(x_1, x_2, x_3, x_4, x_5, x_6) \\ F_2(x_1, x_2, x_3, x_4, x_5, x_6) \\ F_3(x_1, x_2, x_3, x_4, x_5, x_6) \\ F_4(x_1, x_2, x_3, x_4, x_5, x_6) \\ F_5(x_1, x_2, x_3, x_4, x_5, x_6) \\ F_6(x_1, x_2, x_3, x_4, x_5, x_6) \end{pmatrix} \quad (41)$$

A matriz Jacobiana será composta pelas derivadas parciais de cada equação componente do vetor F_n segundo cada incógnita do sistema (42). Ou seja, para o exemplo a seguir, temos:

Sendo $i = 1..6$ e $j = 1..6$ o número de equações e incógnitas, respectivamente, do sistema não linear, o Jacobiano será:

$$J = \begin{pmatrix} \frac{d}{dx_1} F_1 & \frac{d}{dx_2} F_1 & \frac{d}{dx_3} F_1 & \frac{d}{dx_4} F_1 & \frac{d}{dx_5} F_1 & \frac{d}{dx_6} F_1 \\ \frac{d}{dx_1} F_2 & \frac{d}{dx_2} F_2 & \frac{d}{dx_3} F_2 & \frac{d}{dx_4} F_2 & \frac{d}{dx_5} F_2 & \frac{d}{dx_6} F_2 \\ \frac{d}{dx_1} F_3 & \frac{d}{dx_2} F_3 & \frac{d}{dx_3} F_3 & \frac{d}{dx_4} F_3 & \frac{d}{dx_5} F_3 & \frac{d}{dx_6} F_3 \\ \frac{d}{dx_1} F_4 & \frac{d}{dx_2} F_4 & \frac{d}{dx_3} F_4 & \frac{d}{dx_4} F_4 & \frac{d}{dx_5} F_4 & \frac{d}{dx_6} F_4 \\ \frac{d}{dx_1} F_5 & \frac{d}{dx_2} F_5 & \frac{d}{dx_3} F_5 & \frac{d}{dx_4} F_5 & \frac{d}{dx_5} F_5 & \frac{d}{dx_6} F_5 \\ \frac{d}{dx_1} F_6 & \frac{d}{dx_2} F_6 & \frac{d}{dx_3} F_6 & \frac{d}{dx_4} F_6 & \frac{d}{dx_5} F_6 & \frac{d}{dx_6} F_6 \end{pmatrix} \quad (42)$$

O processo de interação é análogo ao método empregado para as equações escalares, utilizando-se da expansão da série de Taylor, chegamos a equação que relaciona a força com o deslocamento nodal dos elementos (43). Isolando o incremento de deformação podemos igualá-lo com o produto negativo da inversa do jacobiano pelo vetor de equações não lineares de cada elemento (44) e (45). O deslocamento é atualizado à medida que as interações são calculadas (46).

$$F_n + J \cdot \Delta_x = 0 \quad (43)$$

$$\Delta_x = -J^{-1} \cdot F_n \quad (44)$$

$$\Delta_{x_i} = -(J_i)^{-1} \cdot F_{n_i} \quad (45)$$

$$x_{i+1} = x_i + \Delta_{x_i} \quad (46)$$

O critério de parada para o método é idêntico ao método escalar. A interação irá terminar quando os incrementos de deformação atingirem a tolerância imposta inicialmente.

4.2 MÉTODO DE NEWTON-RAPHSON MODIFICADO PARA SISTEMAS DE EQUAÇÕES NÃO LINEARES

Como visto anteriormente, o método de Newton-Raphson modificado difere do método de Newton-Raphson convencional. Da mesma maneira que nas equações não lineares o vetor tangente não atualiza. Para sistemas não lineares a matriz Jacobiana permanecerá a mesma até o término das interações. Logo, utilizando-se da expansão da série de Taylor, chegamos a mesma equação do método anterior (47). Isolando o incremento de deformação podemos calculá-lo multiplicando-se a inversa negativa do jacobiano pelo vetor de equações não lineares de cada elemento (48) e (49). Como já estudado o deslocamento atualiza-se a medida que as interações são calculadas (50).

$$F_n + J \cdot \Delta_x = 0 \quad (47)$$

$$\Delta_x = -J^{-1} \cdot F_n \quad (48)$$

$$\Delta_{x_i} = -J_i^{-1} \cdot F_{n_i} \quad (49)$$

$$x_{i+1} = x_i + \Delta_{x_i} \quad (50)$$

O critério de parada desse método é igual ao método anterior. A interação irá terminar quando os incrementos de deformação atingirem a tolerância imposta inicialmente.

5 ESTRUTURA DO SOFTWARE.

O objetivo desse estudo é desenvolver um programa que visa analisar deslocamentos nodais e esforços nas barras de treliças planas, levando em consideração a não linearidade física do material. Para ajudar no desenvolvimento dos códigos de programação, foi utilizado o software MATLAB (2009). A interação com programas como o Gid (2015) e AutoCAD (2010) são fundamentais para a criação de arquivos que alimentarão a entrada de dados da programação final. A interface gráfica também foi programada para melhor utilização dos usuários.

5.1 INTERFACE GRÁFICA

A figura 8 representa a entrada gráfica do software programado.

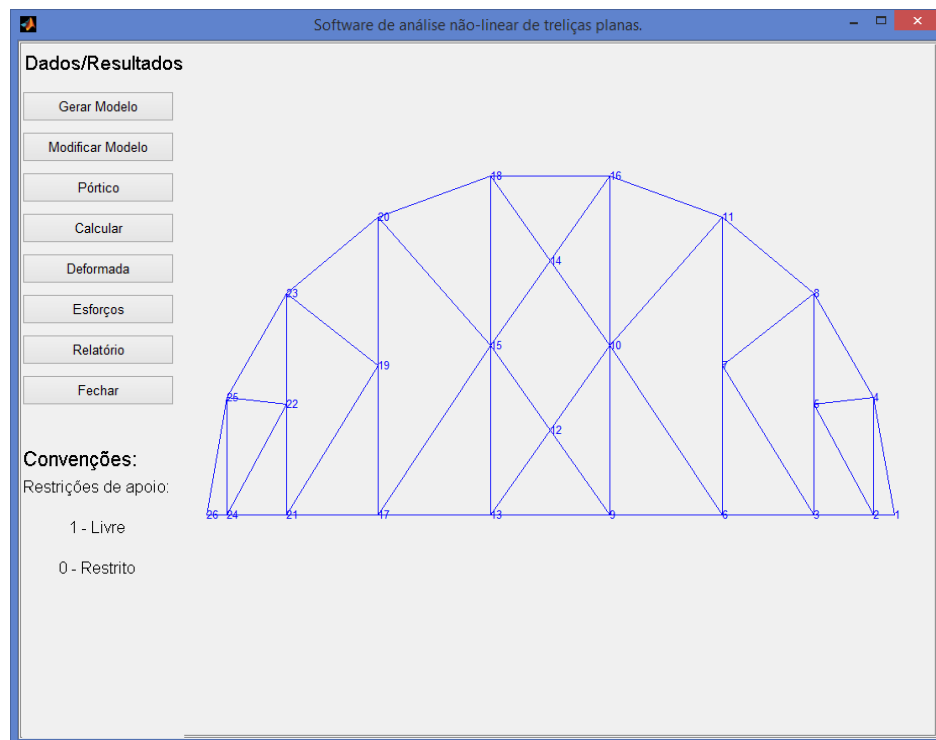


Figura 8. Interface gráfica do código computacional, Fonte: MATLAB (2009).

Com os conceitos de programação do MATLAB e a leitura do livro MATLAB 6 de Hanselman et al. (2004), a interface gráfica foi programada para dar mais facilidade para o usuário que for utilizar o software. A interface gráfica consiste nos comandos básicos do código sendo acionados por “*pushbuttons*”. A interação da escolha dos nós através de pontos de captura para a modificação de suas características (restrições e carregamento) também foi programada. Ao clicar próximo ao nó escolhido o código irá reconhecer o nó que o usuário deseja trabalhar e modificará apenas as características daquele nó. A deformada e os esforços axiais também serão plotados quando desejado, bem como a retirada do relatório com todas as informações do processamento.

5.2 FLUXOGRAMA DE PROGRAMAÇÃO

O fluxograma esquemático de programação é representado pela figura 9.

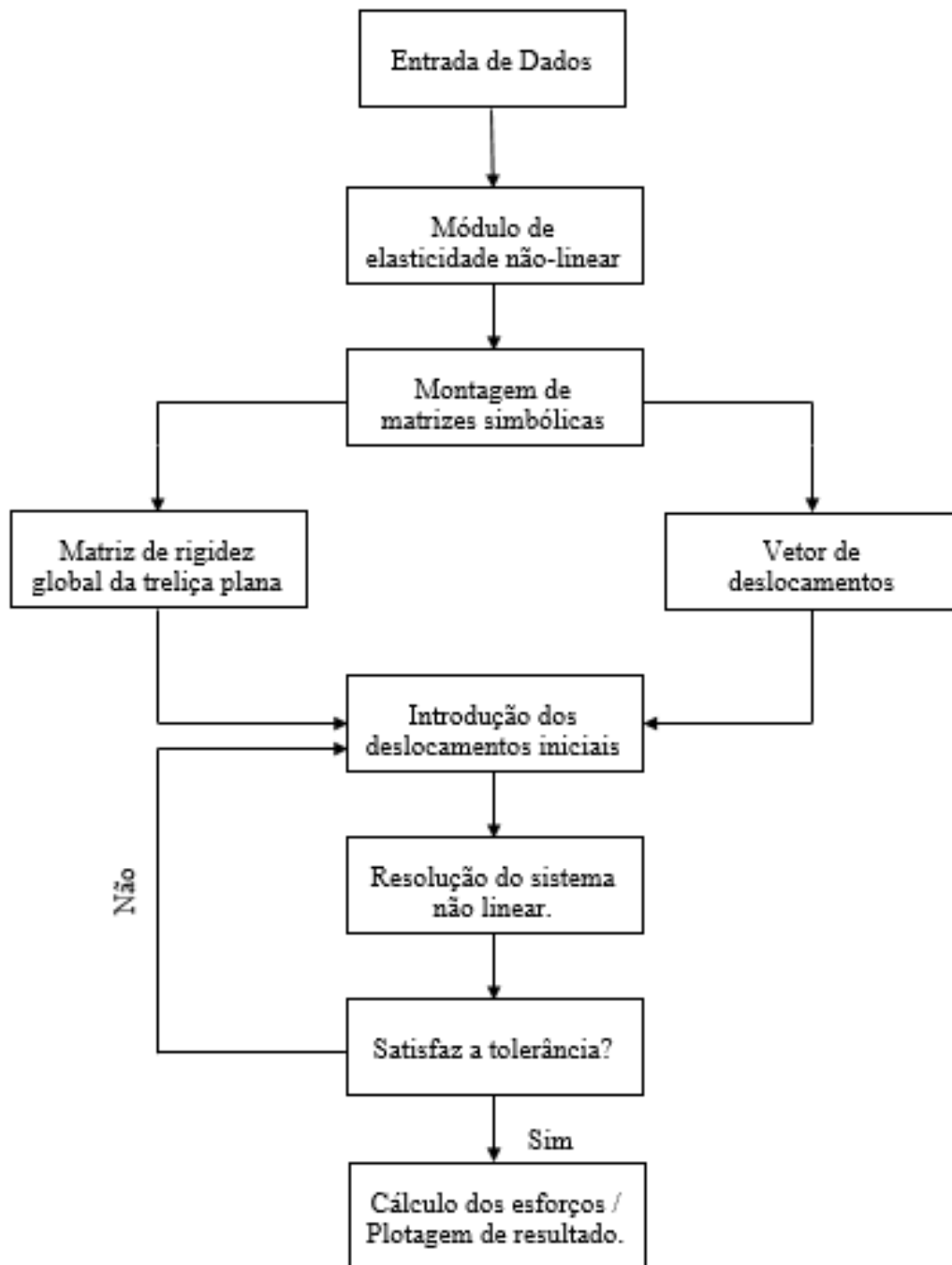


Figura 9. Fluxograma de funcionamento do código de programação.

5.3 CÓDIGO COMPUTACIONAL: PROGRAMAÇÃO MATLAB

A ferramenta utilizada para se programar o código computacional foi o software MATLAB. O MATLAB é um software de programação matemática que se baseia em matrizes e vetores para processar as operações matemáticas. Além da programação matricial o MATLAB nos proporciona a interação entre outros softwares. O caso em estudo utiliza-se dessa ferramenta para fazer a ligação entre saídas de dados do software Gid e a entrada de dados do código computacional programado.

O Gid tem como finalidade o pré-processamento da treliça plana estudada. Depois de modelada o software cria um arquivo de texto contendo todos os nós e suas coordenadas globais, bem como todas as conectividades das barras.

Uma outra ferramenta utilizada foi a interface gráfica que o próprio MATLAB disponibiliza (GUI). A interface gráfica foi programada para que haja a interação do usuário com o programa, facilitando o uso do código.

O código programado subdivide-se em: programa principal e rotinas:

Programa principal:

- *TPANL*: O programa principal denominado TPANL é o responsável por criar a interface gráfica do software e fazer a interação entre as rotinas, fazendo com que cada uma seja acionada ao comando dos botões predefinidos.

Rotinas:

- *GerPor*: A rotina GerPor tem como finalidade importar os arquivos de dados que são retirados do Gid e plotar a treliça para que a mesma seja modificada.
- *CarPor*: A rotina CarPor, depois de plotada a treliça, é responsável por, ao comando de um clique em qualquer nó, alterar as características daquele nó (restrições e carregamentos).
- *ModElast*: O arquivo ModElast processa o módulo de elasticidade não-linear do material, deixando a equação em função dos deslocamentos nodais já nos eixos globais.
- *Solver*: O Solver é a rotina responsável por calcular os deslocamentos nodais utilizando-se do método de Newton-Raphson. A rotina lê o módulo de elasticidade criado e gera as matrizes necessárias para o cálculo (matriz de carregamento, rigidez global da estrutura e deslocamentos). Uma vez criada as matrizes, essa rotina tem como objetivo substituir os deslocamentos simbólicos pelos deslocamentos encontrados em cada interação, até que a tolerância seja satisfeita.
- *CalcNorm*: Depois de encontrados os deslocamentos nodais que satisfaçam a tolerância imposta, a rotina CalcNorm calcula os esforços normais em cada barra da treliça plana.
- *Desl, Indef e Esfor*: As rotinas Desl, Indef e Esfor são as responsáveis por plotar a treliça deformada, o esquema de apoios e carregamento da treliça estudada e os esforços atuantes nas barras. Todos os resultados são plotados em forma gráfica para melhor entendimento do usuário.

6 RESULTADOS

Para a validação dos resultados, utilizou-se um programa comercial, que dentre suas funções, faz análises não lineares, quando atribuído ao módulo de elasticidade uma função predefinida. Com o software ANSYS (2015) foi processada uma treliça plana, como também, a mesma foi calculada de forma analítica. Resultados conclusivos e com um baixo índice de erros foram obtidos, validando o código programado. Analisando o exemplo do livro Métodos dos Elementos Finitos em análise estrutural, Vaz (2011), chegamos a uma nova validação do código com resultados também conclusivos.

6.1 Treliça 1

6.1.1 Resumo

A treliça 1, indicada na figura abaixo, foi processada com uma seção transversal de área $0,01 \text{ m}^2$. A curva Tensão x Deformação foi associada ao polinômio de terceiro grau: $\sigma(\varepsilon) = -1 \cdot 10^{18} \cdot \varepsilon^3 + 10^{11} \cdot \varepsilon$. Para efetuar a validação, foi atribuído valor para a carga P de 50 kN, 100 kN, 120 kN e 160 kN. Para o processamento foi criado um arquivo com pontos amostras que indicassem o polinômio escolhido.

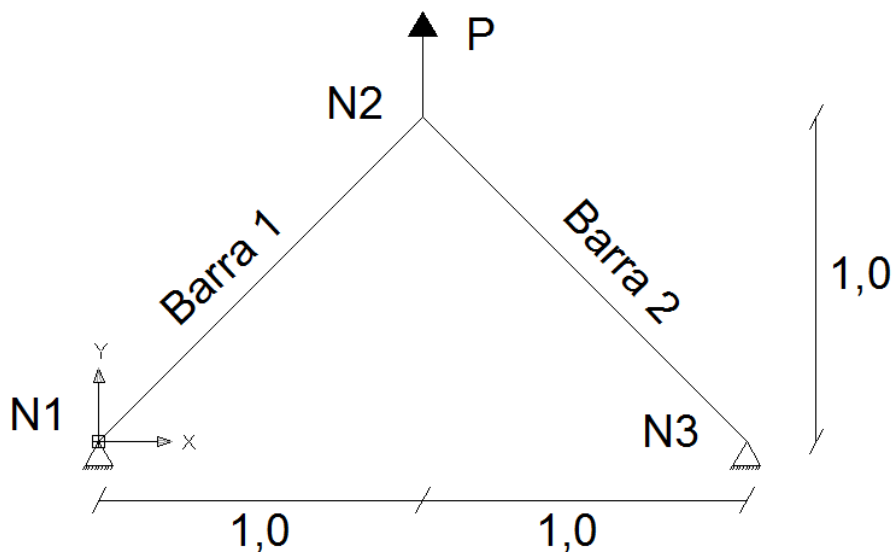


Figura 10. Esquema da treliça 1 processada.

6.1.2 Resultados

Depois de processada a treliça, foram retirados resultados conclusivos para o software. As deformações foram validadas comparando resultados do software Ansys e deslocamentos calculados analiticamente. A validação do comportamento físico do material foi dado através do gráfico Força x Deslocamento, representado pela figura 11, que resultou no regime não linear. A equação analítica resultante foi dada por:

$$F_y := \left[10^{11} \cdot \frac{d_y}{L} - 1 \cdot 10^{18} \cdot \left(\frac{d_y^3}{2L^3} \right) \right] \cdot A$$

Os resultados obtidos das deformações do nó N2 foram representados na tabela 1. Foram analisados os erros entre os resultados do código computacional e os resultados analíticos e retirados do software Ansys (2015).

Deformações (m)					
Def	50 (kN)				
	Ansys	Analítico	Matlab	Erro Ansys	Erro Analítico
dx ₂	0	0	0	0.09763%	-0.04184%
dy ₂	7.170E-05	7.160E-05	7.163E-05		
Def	100 (kN)				
	Ansys	Analítico	Matlab	Erro Ansys	Erro Analítico
dx ₂	0	0	0	0.06671%	0.00000%
dy ₂	1.499E-04	1.498E-04	1.498E-04		
Def	120 (kN)				
	Ansys	Analítico	Matlab	Erro Ansys	Erro Analítico
dx ₂	0	0	0	0.10758%	0.00000%
dy ₂	1.859E-04	1.857E-04	1.857E-04		
Def	160 (kN)				
	Ansys	Analítico	Matlab	Erro Ansys	Erro Analítico
dx ₂	0	0	0	-0.67640%	0.00000%
dy ₂	2.809E-04	2.828E-04	2.828E-04		

Tabela 1. Tabela de deslocamentos nodais.

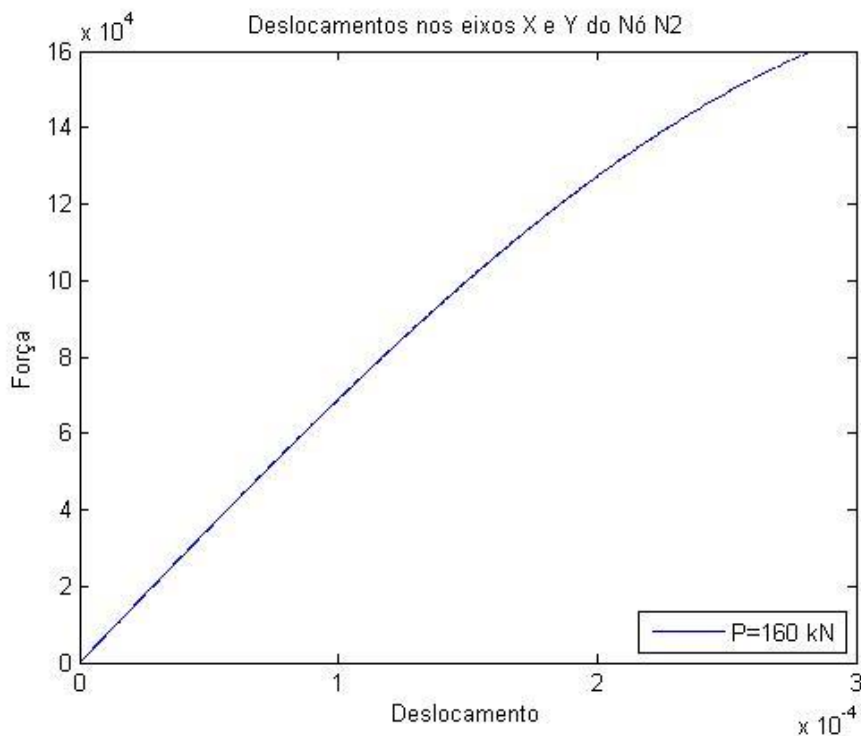


Figura 11. Curva Força x Deslocamento, Fonte: MATLAB(2009).

6.2 Treliça 2

6.2.1 Resumo

A treliça 2, reproduz o exemplo de Vaz (2011), com os valores coerentes com o exemplo anterior. A treliça foi processada com uma seção transversal de área 0.01m^2 . A curva Tensão x Deformação foi associada à equação $\sigma(\varepsilon) = -1 \cdot 10^{18} \cdot \varepsilon^3 + 10^{11} \cdot \varepsilon$. Para efetuar a validação, foi atribuído valor para a carga $P = 140 \text{ kN}$ e para $L=1\text{m}$. Para o processamento foi criado um arquivo com pontos amostras que indicassem a equação escolhida. A figura 12 mostra o esquema da treliça calculada no exemplo.

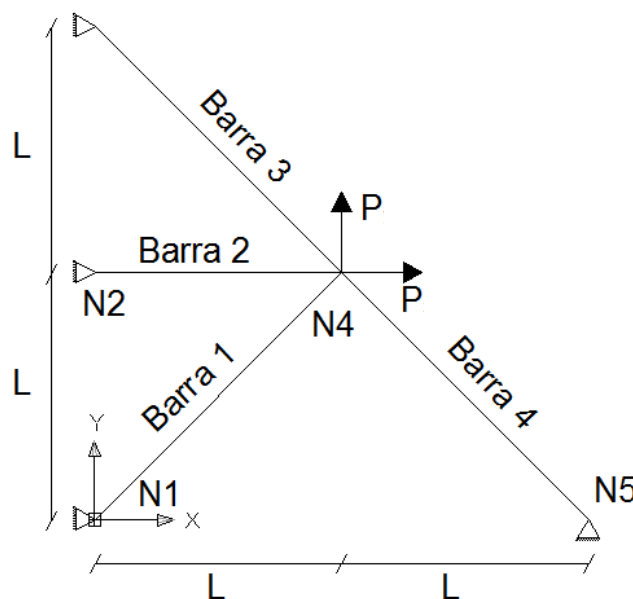


Figura 12. Esquema da treliça 2 processada.

6.2.2 Resultados

Depois de processada a treliça, os mesmos resultados retirados para a treliça 1 foram considerados. A tabela 2 mostra o comparativo entre os resultados dos deslocamentos nodais do nó N4 obtidos no processamento do software Ansys (2015) e os resultados obtidos com o código do Matlab (2009). As curvas Força x Deslocamentos, foram dadas tanto para o processamento no código computacional como para os resultados do software Ansys. A figura (13) representam a curva Força x Deformação gerada pelo código computacional do Matlab (2009) para os eixos X e Y. O gráfico da figura (13) pode ser comparado com as demais figuras (14) e (15) que representam os resultados do processamento do software Ansys (2015).

Deformações			
Def	Px=140 kN e Py=140 kN		
	Ansys	Matlab	Erro
Dx₄	1.3159E-04	1.3179E-04	-0.16%
Dy₄	2.0966E-04	2.0999E-04	-0.16%

Tabela 2. Tabela de deslocamentos nodais.

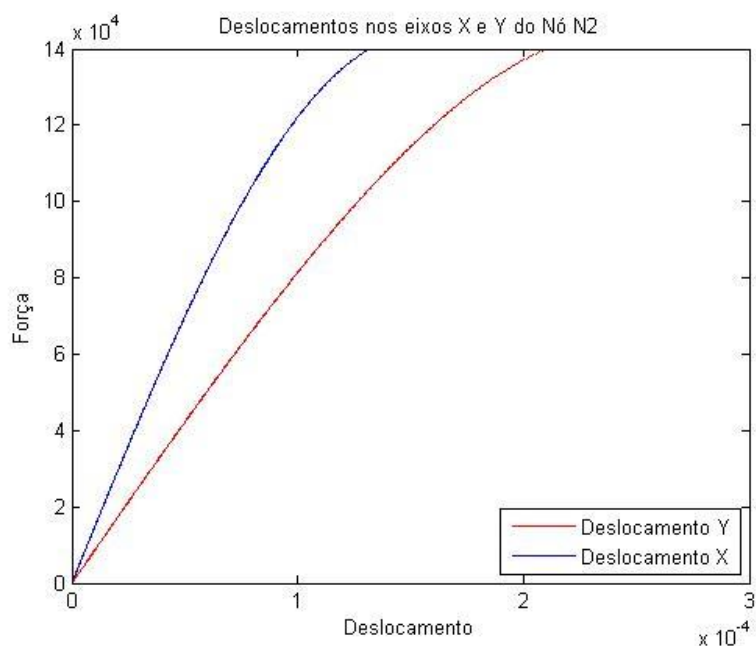


Figura 13. Curva Força x Deslocamento Fonte: MATLAB (2009).

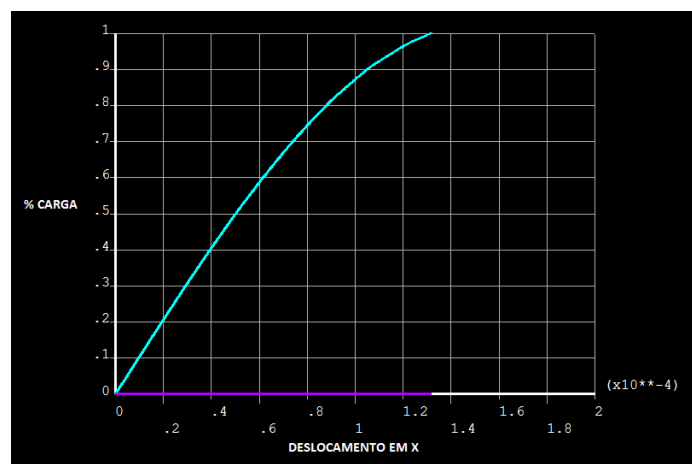


Figura 14. Resultados na direção X (CARGA X DESLOCAMENTO), Fonte: ANSYS (2015)

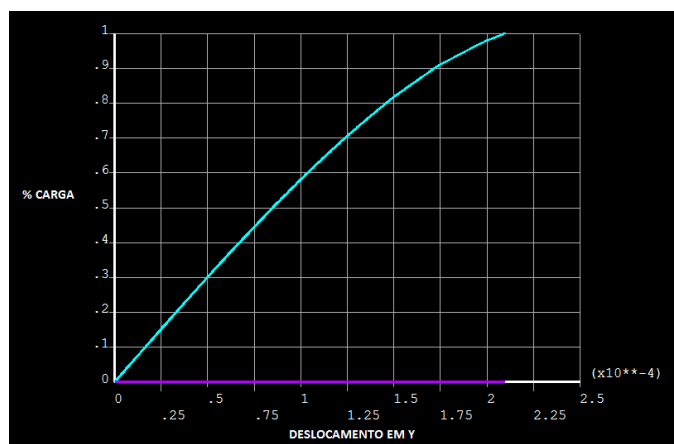


Figura 15. Resultados na direção Y (CARGA X DESLOCAMENTO), Fonte: ANSYS (2015)

7 CONCLUSÃO

As análises realizadas com os métodos de Newton-Raphson, Newton Raphson modificado e o método incremental permitiram uma análise detalhada dos métodos. O método incremental, apesar de ser muito utilizado em projetos acadêmicos, acaba produzindo resultados com uma variação de erro muito grande. Para minorar tais erros é preciso discretizar o carregamento em incrementos de cargas muito pequenos, resultando em processamento mais demorado. O método de Newton-Raphson tem como ponto positivo a rápida convergência para o resultado desejado, porém, a cada interação se calcula a tangente do hiperplano, somando mais uma operação no processamento. O método de Newton-Raphson modificado não necessita do cálculo da tangente ao hiperplano, porém sua convergência a resultados satisfatórios é lenta. A partir dessas conclusões foi que decidimos criar o código baseado no método de Newton-Raphson.

Os resultados, tanto da programação do código como da solução dos problemas propostos, foram satisfatórios. O software desenvolvido processa qualquer treliça plana, estando está de acordo com a estabilidade dos materiais, em tempo hábil. A interface gráfica dá ao usuário a facilidade do uso e o bom entendimento do código. O usuário não irá necessitar de conhecimento aprofundado no assunto para utilizar-se do software. Os resultados retirados estão coerentes e com baixo índice de erros, se comparados aos softwares comerciais, garantindo a análise sem prejudicar o rendimento do projeto. A importância da análise não-linear é ter respostas mais conclusivas do real comportamento dos elementos, visando diminuir custos atribuídos a coeficientes de segurança elevados.

Este artigo é a parte inicial do desenvolvimento de uma dissertação de mestrado que visa a análise não linear de esforços e deformações em estruturas reticuladas.

BIBLIOGRAFIA

Hanselman, Duane, Littlefield, Bruce, 2003. *Matlab 6 Curso Completo*, Prentice Hall.

Vaz, Luiz Eloy, 2011. *Método dos Elementos Finitos em Análise de Estruturas*, Campus.

Ivančo, Vladimír, 2011. *Nonlinear Finite Element Analysis*, Artigo, *Faculty of Mechanical Engineering, Technical University of Kosice, Slovakia*.

Lacerda, Estéfane George Macedo, 2014. *Análise não linear de treliças pelo método dos elementos finitos posicional*. Artigo. Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.

Sistemas de Equações Algébricas e Trancedentas (Não-Lineares), Nota de aula da Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica.

Mathworks, MATLAB, Mathworks Inc., Natick (2009)

CIMNE, Gid, Center for Numerical Methods in Engineering, Barcelona (2015).

ANSYS, ANSYS Inc., Canonsburg (2015)

Autodesk Inc., Autocad, Autodesk Inc, São Rafael, Califórnia (2010).