

Miguel Filipe Magalhães Soares de Carvalho

**Formulação corrotacional para análise de vigas
com elementos finitos**

Lisboa

2010

UNIVERSIDADE NOVA DE LISBOA

Departamento de Engenharia Mecânica e Industrial

Mestrado Integrado em Engenharia Mecânica

Formulação corrotacional para análise de vigas com elementos finitos

Por:

Miguel Filipe Magalhães Soares de Carvalho

Dissertação apresentada na faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para obtenção do grau de Mestre em Engenharia Mecânica.

Orientador: Professor Doutor João Cardoso

Lisboa

2010

Resumo:

A presente dissertação aborda a aplicação do método dos elementos finitos em estruturas constituídas por vigas. É considerado que as vigas que constituem a estrutura sofrem grandes deslocamentos (afastando-se da configuração inicial) mas deformações de pequena amplitude devido à elevada esbelteza das mesmas. O pressuposto descrito é onde se baseia a formulação corrotacional. Nestas situações, uma vez que não existe uma proporcionalidade directa entre forças e deslocamentos, está-se na presença de análises não lineares. Este tipo de análise necessita de um processo iterativo para a sua resolução, uma vez que as equações que regem o comportamento da estrutura dependem da configuração deformada, ou seja, as equações de equilíbrio da estrutura necessitam de ser actualizadas durante o processo. Nesta dissertação utiliza-se o método de Newton-Raphson, o qual, adaptado, passa a um processo incremental-iterativo.

Dos conceitos teóricos passa-se à construção do algoritmo “*PEFNL-2D*” capaz de estudar as situações descritas. Foram criadas quatro versões distintas do referido programa, que são testadas em diversos exemplos. Uma das quatro versões testadas mostrou ser a mais adequada, uma vez que demonstra versatilidade para todos os exemplos efectuados e apresenta uma boa precisão nos resultados.

Palavras-chave: Método dos elementos finitos, análise não linear, formulação corrotacional, método de Newton-Raphson, métodos computacionais.

Abstract:

This thesis deals with finite element application in beam structures. It is considered that the beams forming the structure suffer large displacements (getting far from the original configuration) but small amplitude deformations due to their high slenderness. The corotacional formulation is based on the previous assumption. In these situations, since there are no direct relationship between forces and displacements, a nonlinear analysis is considered. This kind of analysis requires an iterative process for its resolution, since the equations that characterize the structure behavior depend on the deformed configuration, i.e., the structure equilibrium equations need to be updated during the process. In order to solve it, the Newton-Raphson's method is used, which is transformed into an incremental-iterative process.

Based on the theoretical concepts, the "*PEFNL-2D*" was developed; this algorithm is able to study the described situations. Four different versions were created for this program, which are tested in several examples. One of the tested versions was found to be the most appropriate, since it demonstrates versatility for all examples and produces results with good precision.

Keywords: Finite element method, nonlinear analysis, corotacional formulation, Newton-Raphson method, computational methods.

Agradecimentos

Embora a Tese de Mestrado seja um projecto individual, é sem dúvida algo que não depende só de mim mas de todos os que me rodeiam.

Como tal, as minhas palavras de agradecimento e reconhecimento relacionadas com a realização desta dissertação são especialmente dirigidas ao Professor Doutor João Cardoso, não só pela partilha dos seus enormes conhecimentos, mas pela sua enorme dedicação à docência e inesgotável disponibilidade.

Aos meus colegas e amigos André Cunha, Bruno Rodrigues, Gonçalo Pimpão, Gonçalo Peixoto, Luís Ensinas, Pedro Barros, Pedro Carvalho, Pedro Varela e Pedro Santana que de um modo ou de outro me ajudaram e incentivaram a chegar até aqui, fazendo da FCT um sítio melhor.

Aos meus colegas e amigos Daniel Rolo e Nuno Boavida que, por também estarem a realizar a sua dissertação, me acompanharam no dia-a-dia de trabalho, partilhando conhecimentos, opiniões e incentivos.

Ao meu amigo João Faria, companheiro de estudo durante todo o curso até à dissertação, uma palavra de agradecimento por não deixar ninguém cair em desânimo e uma palavra de incentivo para que finalize da melhor forma a sua tese.

À Vera Abecasis, não só pela revisão ortográfica desta dissertação mas acima de tudo pelo seu apoio, compreensão e incentivo.

Como não poderia deixar de ser, agradeço também à minha família que fez de mim parte do que sou hoje. Aos meus Pais, Carlos e Manuela Carvalho, por se preocuparem, me incentivarem e sempre me apoiarem incondicionalmente, bem como por investirem na minha educação. Ao meu irmão, à minha cunhada e à minha madrinha pelo seu interesse e incentivo na conclusão da dissertação.

A todos um sincero obrigado.

Simbologia

$[G_e], G_{JK}$	Matriz geométrica no sistema de coordenadas locais
$[K_e], K_{JK}$	Matriz de rigidez no sistema de coordenadas locais
$[R_n]$	Matriz de transformação do referencial local inicial para o referencial local actual, que corresponde a uma rotação de corpo rígido de ângulo β
$[T_n]$	Matriz de transformação do referencial global para o referencial local actual
$\{X_0\}$	Vector com as coordenadas dos nós do elemento, no referencial global, para a configuração inicial
$\{X_e\}$	Vector com as coordenadas dos nós do elemento, no referencial local para a configuração inicial
$\{X_n\}$	Vector que guarda as coordenadas dos nós do elemento no referencial global, para a iteração n
$\{X_\beta\}$	Vector que guarda as coordenadas dos nós do elemento no referencial global, rodado do ângulo β
$\{u^d\}$	Vector de deslocamentos que produz deformação no elemento, no referencial global
$\{u_e^d\}$	Vector de deslocamentos que produz deformação no elemento, no referencial local
$\{u_n\}$	Deslocamentos dos nós medidos no sistema de coordenadas globais, na iteração n
$\{u^r\}$	Deslocamentos associados à rotação de corpo rígido no referencial global
$\{u_{t,e}^d\}$	Vector com a componente de translação do deslocamento devido à deformação, no referencial local
$\{u_t^d\}$	Vector com a componente de translação do deslocamento devido à deformação, no referencial global
$\{\alpha^d\}$	Vector com as rotações que produzem deformação no elemento
$\{\Delta f_n\}$	Vector das forças residuais na iteração n

$F_J, \{f_e\}$	Vector de forças aplicadas no elemento em coordenadas locais
$[K]$	Matriz de rigidez numa análise linear de estruturas no sistema de coordenadas globais
$N_I, N_{II}, N_{III}, N_{IV}$	Funções de forma
N_{inc}	Número de incrementos
U^A	Energia de deformação associada ao esforço axial
U^F	Energia de deformação associada ao momento flector
$\{f\}$	Vector de forças aplicadas no sistema de coordenadas globais
$\{u\}$	Deslocamentos dos nós medidos no sistema de coordenadas globais
u_I, u_{II}	Deslocamentos segundo o eixo x nos nós do elemento
$u_K, \{u_e\}$	Deslocamentos dos nós medidos no sistema de coordenadas locais
v_I, v_{II}	Deslocamentos segundo o eixo y nos nós do elemento
$\{\alpha\}$	Vector que guarda as rotações totais dos nós, medidas no referencial local
$\{\beta\}$	Vector com a rotação de corpo rígido de cada elemento
ε_{ij}	Tensor das deformações
θ_I, θ_{II}	Rotações em torno do eixo z do elemento
σ_{ij}	Tensor das tensões
$[K(u_n)]$	Matriz de rigidez numa análise não linear de estruturas, iteração n
Δu_n	Incremento dos deslocamentos na iteração n
A	Área de secção
E	Módulo de Young
I	Momento de inércia
$M(x)$	Momento Flector
P, F	Carga Aplicada
U	Energia de deformação

V	Energia potencial das forças aplicadas
e	Excentricidade
l, L	Comprimento
$w(x)$	Equação da linha elástica
δ, u	Deslocamento
π	Energia potencial total

Índice

1	INTRODUÇÃO.....	1
1.1	CONTEXTO.....	1
1.2	OBJECTIVOS.....	2
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO	3
2	ANÁLISE NÃO LINEAR DE ESTRUTURAS	5
2.1	COMPORTAMENTO DE UMA ESTRUTURA	5
2.2	ANÁLISES LINEARES E NÃO LINEARES.....	5
2.3	MÉTODO DOS ELEMENTOS FINITOS EM VIGAS	7
2.3.1	<i>Energia de deformação (U)</i>	8
2.3.1.1	Energia de deformação associada ao esforço axial	9
2.3.1.2	Energia de deformação associada ao momento flector.....	11
2.3.2	<i>Energia potencial das forças aplicadas (V)</i>	17
2.3.2.1	Energia potencial para forças e momentos aplicados nos nós	17
2.3.2.2	Energia potencial para cargas axiais	17
2.3.3	<i>Energia potencial total (π)</i>	21
2.3.4	<i>Transformação de coordenadas</i>	22
2.3.5	<i>Análise não linear</i>	24
2.4	MÉTODO DE NEWTON-RAPHSON	24
2.5	ANÁLISE INCREMENTAL ITERATIVA.....	26
3	FORMULAÇÃO CORROTACIONAL EM VIGAS 2D	29
3.1	FORMULAÇÃO CORROTACIONAL	29
3.1.1	<i>Translações na formulação corrotacional</i>	29
3.1.2	<i>Ângulos de rotação na formulação corrotacional</i>	34
3.1.3	<i>Determinação do ângulo β</i>	35
3.2	ESFORÇOS INTERNOS	37
3.3	ESQUEMA DO ALGORITMO DESENVOLVIDO.....	39
4	EXEMPLOS DE ESTUDO	42
4.1	EXEMPLO VIGA COM MOMENTO	43
4.1.1	<i>Considerações</i>	43
4.1.2	<i>Resultados Obtidos</i>	45
4.2	EXEMPLO VIGA COM FORÇA AXIAL.....	49
4.2.1	<i>Considerações</i>	49
4.2.2	<i>Resultados obtidos</i>	51
4.3	EXEMPLO VIGA COM FORÇA TRANSVERSAL	56
4.3.1	<i>Considerações</i>	56

4.3.2	<i>Resultados obtidos</i>	57
4.4	EXEMPLO PÓRTICO	62
4.4.1	<i>Considerações</i>	62
4.4.2	<i>Resultados obtidos</i>	63
5	CONCLUSÕES	68
	REFERÊNCIAS BIBLIOGRÁFICAS	70
	ANEXOS	71

Índice de figuras

Fig. 2-1-Exemplo de um comportamento não linear (cana de pesca) [Ans07].	6
Fig. 2-2-Comportamento de um pilar comprimido excentricamente. a) Configuração indeformada. b) Configuração deformada. c)Trajectórias de equilíbrio. [Rei01].	7
Fig. 2-3-Elemento Finito com referencial de eixos (x y z).	8
Fig. 2-4-Elemento Finito de uma viga com carregamento axial	9
Fig. 2-5-Funções de forma.	10
Fig. 2-6- Elemento Finito com dois nós	12
Fig. 2-7-Viga com cargas axiais aplicadas	17
Fig. 2-8-Comportamento linear e não linear	18
Fig. 2-9 - Origem da matriz transformação [T]	24
Fig. 2-10-Representação do processo iterativo de Newton-Raphson.	25
Fig. 2-11 - Método incremental iterativo.	27
Fig. 3-1 a) Representação da configuração inicial do elemento de viga b) Representação do elemento de viga na iteração n.	31
Fig. 3-2- Representação do elemento de viga após a rotação β .	33
Fig. 3-3- Problemática do cálculo do ângulo β .	35
Fig. 3-4 - Cálculo do $\alpha_{MÉDIO}$.	36
Fig. 3-5- Deformações e esforços resultantes.	38
Fig. 4-1-Esquema ilustrativo das diferentes configurações do algoritmo “PEFNL-2D”.	42
Fig. 4-2- Exemplo viga com momento a) Esquema do exemplo b) Modelo de elementos finitos.	43
Fig. 4-3-Configuração original e deformada da viga (Ansys).	47
Fig. 4-4 - Deslocamentos para diferentes valores de $M^*=ML/2\pi E$ (adaptado [Urt05]).	48

Fig. 4-5 - Exemplo viga com carga axial a) Comportamento esperado b)Modelo de elementos finitos.....	49
Fig. 4-6 – Aplicação dos diferentes valores de F e comportamento esperado da viga...	50
Fig. 4-7 - Configuração inicial e final da viga para cada um dos incrementos de força efectuados.....	53
Fig. 4-8 - Configuração inicial e deformada da viga (adaptado [Tim61])	54
Fig. 4-9 - Exemplo viga com força transversal a) Esquema do exemplo b)Modelo de elementos finitos.....	56
Fig. 4-10 - Configuração inicial e deformada da viga (para $FL^2/EI=10$).....	59
Fig. 4-11 - Deslocamentos adimensionais em função de diferentes valores de F	60
Fig. 4-12 - Exemplo Pórtico a) Esquema do exemplo b)Modelo de elementos finitos..	62
Fig. 4-13 - Perfil de viga HEB300.....	62
Fig. 4-14 - Configuração inicial e deformada da estrutura.....	65

Índice de tabelas

Tabela 4-1- Propriedades geométricas e materiais da viga.	44
Tabela 4-2-Parâmetros de análise do programa Ansys	45
Tabela 4-3-Resultados para as configurações 1.1 e 1.2 do algoritmo	45
Tabela 4-4- Resultados para as configurações 2.1 e 2.2 do algoritmo	46
Tabela 4-5-Resultados do programa Ansys.....	46
Tabela 4-6-Comparação de resultados Ansys - algoritmo (nó 11).....	47
Tabela 4-7-Comparação de resultados [Urt05] - algoritmo (nó 21).....	48
Tabela 4-8 - N° de iterações e n° de incrementos nas diferentes configurações do programa “PEFNL-2D” (exemplo viga com momento).....	49
Tabela 4-9-Propriedades geométricas e materiais da viga.	50
Tabela 4-10 - Parâmetros de análise do programa Ansys	51
Tabela 4-11 - Resultados para a configuração 1.1 e 1.2 do algoritmo	52
Tabela 4-12 - Resultados para a configuração 2.1 e 2.2 do algoritmo	52
Tabela 4-13 - Resultados da simulação no Ansys	53
Tabela 4-14 - Comparação de resultados Ansys - algoritmo.....	54
Tabela 4-15 - Comparação de resultados [Tim61] - algoritmo	55
Tabela 4-16 - N° de iterações e n° de incrementos nas diferentes configurações do programa “PEFNL-2D” (viga com força axial)	55
Tabela 4-17-Propriedades geométricas e materiais da viga	56
Tabela 4-18-Parâmetros de análise (programa Ansys).....	57
Tabela 4-19- Resultados obtidos para a configuração 1.1 e 1.2 do algoritmo	58
Tabela 4-20 - Resultados obtidos para a configuração 2.1 e 2.2 do algoritmo	58
Tabela 4-21 - Resultados da simulação no programa Ansys.....	59
Tabela 4-22 - Comparação de resultados Ansys - algoritmo.....	60
Tabela 4-23- Comparação de resultados [Urt05] - algoritmo.....	61

Tabela 4-24 - N° de iterações e n° de incrementos nas diferentes configurações do programa “PEFNL-2D” (viga com força transversal).....	61
Tabela 4-25- Características do perfil HEB300 e características materiais da viga.....	63
Tabela 4-26 - Parâmetros da análise efectuada no programa Ansys	63
Tabela 4-27 - Resultados do algoritmo “PEFNL-2D” versão 1.1 e 1.2	64
Tabela 4-28 – Resultados do algoritmo “PEFNL-2D” versão 2.1 e 2.2.....	64
Tabela 4-29 - Resultados da simulação efectuada no Ansys utilizando o elemento de viga BEAM3 e BEAM4.	65
Tabela 4-30 - Comparação de resultados no nó 11 Ansys - algoritmo.....	66
Tabela 4-31 - Comparação de resultados no nó 21 Ansys - algoritmo.....	66
Tabela 4-32 - N° de iterações e n° de incrementos nas diferentes configurações do programa “PEFNL-2D” (Pórtico).....	67

1 INTRODUÇÃO

1.1 Contexto

A noção de estabilidade é muito importante na análise de estruturas e encontra-se associada ao conceito de equilíbrio. A estabilidade de qualquer configuração é avaliada através do comportamento da estrutura. Uma estrutura em equilíbrio diz-se estável ou instável consoante regresse ou não à configuração inicial após a perturbação (acção das forças exteriores) cessar.

O tipo de análise estrutural efectuada condiciona o tipo de problema que se pode resolver. Uma análise linear caracteriza-se pela formulação das equações de equilíbrio na configuração inicial, o que impossibilita o uso deste tipo de análises para problemas de estabilidade. Este fenómeno está intrinsecamente associado a alterações na geometria que produzem um comportamento não linear. O que indica que para o estudar é necessário realizar sempre análises não lineares.

Para processar os cálculos de uma análise não linear, é necessário recorrer a processos iterativos, uma vez que as equações que regem o comportamento da estrutura dependem da configuração deformada, o que faz com que as posições de equilíbrio da estrutura tenham que ser actualizadas durante o processo. É neste contexto que surge a formulação corrotacional que será utilizada no presente estudo.

Interessa portanto compreender de onde surge a formulação corrotacional, esta tem as suas raízes num conceito da mecânica dos meios contínuos: separação dos movimentos de corpo rígido dos movimentos que estão associados à deformação. Os avanços tecnológicos que envolveram esta decomposição de movimentos surgiram na indústria aeronáutica e aeroespacial. O conceito da separação destes dois tipos de movimentos, para uma estrutura completa, foi utilizada por projectistas de estruturas aeroespaciais nas décadas de 50 e 60, tendo como grande objectivo a monitorização do movimento principal das estruturas, neste sentido, procurou-se definir um sistema de eixos cartesianos e ortogonais único, que acompanhasse o movimento do corpo e, em relação ao qual, os deslocamentos, velocidades e acelerações de um ponto material fossem unicamente devidos à deformação [Men06].

A extensão deste conceito, utilizado pela indústria aeronáutica para a análise não linear utilizando o método dos elementos finitos, baseia-se numa modificação simples: em vez de utilizar um sistema de eixos único para a estrutura como um todo, utiliza-se um sistema de eixos por elemento. Esta modificação é essencial para o sucesso da formulação corrotacional, uma vez que ela ajuda a satisfazer uma hipótese inicial básica: que os deslocamentos e rotações devidos à deformação do elemento sejam pequenos em relação ao sistema de eixos corrotacional [Men06].

Alguns trabalhos sobressaem na evolução da formulação corrotacional, um dos quais o de Rankin e Brogan [Ran86] que introduziu o procedimento EICR (*Element Independent Corrotational Formulation*). O importante contributo deste trabalho consiste na criação de “filtros” que possibilitem usar os programas já existentes de análise linear de estruturas, estendendo a sua capacidade à análise não linear. Estes “filtros” actuam no cálculo do vector de forças internas e da matriz de rigidez e são puramente geométricos, podendo ser usados para todos os elementos finitos da mesma família. A solução apresentada pelos referidos autores serve de base à metodologia de análise corrotacional que será desenvolvida nesta dissertação.

1.2 Objectivos

O objectivo da presente dissertação consiste na aplicação do método dos elementos finitos para a análise de estruturas planas, constituídas por vigas, quando ocorrem deslocamentos e rotações finitas e em simultâneo deformações infinitesimais. Estas condições verificam-se, por exemplo, na fase de pós-encurvadura em estruturas constituídas por vigas.

A formulação corrotacional tem sido usada com grande sucesso para resolver este tipo de problemas e será usada também nesta análise. Será elaborado um algoritmo em linguagem *Matlab*¹ e feita a aplicação do mesmo em problemas de demonstração.

A preparação da presente dissertação incluiu três fases distintas:

- (i) Estudo da teoria sobre o método dos elementos finitos e a sua aplicação na análise de estruturas com grandes deslocamentos e rotações. A formulação corrotacional foi analisada com detalhe nesta fase devido à sua importância no contexto geral da tese. Foi também nesta fase que se estudaram os

¹ The MathWorks Corporate

programas a utilizar, o programa de elementos finitos, *Ansys*² e o programa para análise numérica *Matlab*.

- (ii) Estando consolidado o estudo das teorias necessárias, passou-se à implementação das mesmas. Recorreu-se então à elaboração de um programa de computador em *Matlab* que permitiu analisar estruturas planas de vigas nas referidas condições através do método dos elementos finitos.
- (iii) Após a elaboração do referido programa computacional, aplicou-se o mesmo na resolução de problemas, aferindo a qualidade das soluções obtidas através da comparação dos resultados encontrados com os apresentados na literatura e com os calculados no programa *Ansys*.

1.3 Organização da dissertação

Contextualizados os objectivos desta dissertação, interessa especificar a organização da mesma. No capítulo 1, onde se insere este subcapítulo, estão descritos os objectivos deste trabalho e o contexto em que ele se insere para que fique explícita a motivação que levou à sua elaboração.

O capítulo 2 compreende os conceitos teóricos fundamentais usados em análise não linear de estruturas. São brevemente descritos os diferentes comportamentos de uma estrutura, explicadas as diferenças entre uma análise linear e uma análise não linear e é estudada a aplicação do método dos elementos finitos em vigas. É descrito ainda o método iterativo associado à análise não linear a efectuar.

No capítulo 3 é descrita a formulação corrotacional desenvolvida para a elaboração do algoritmo.

Para testar e validar o programa desenvolvido apresentam-se diferentes problemas de aplicação no capítulo 4, os resultados obtidos são comparados com o *Ansys* e/ou exemplos presentes na bibliografia.

No último capítulo, apresentam-se as conclusões desta dissertação, analisando criticamente os resultados obtidos nos problemas aplicados ao programa desenvolvido e verificando as limitações e evoluções que o mesmo poderá apresentar.

²ANSYS, INC. Corporate

2 ANÁLISE NÃO LINEAR DE ESTRUTURAS

2.1 Comportamento de uma estrutura

Para caracterizar o comportamento de uma estrutura submetida a um conjunto de acções, analisa-se a relação existente entre os valores dessas acções e os efeitos por elas provocados na estrutura, como por exemplo, tensões, deformações ou deslocamentos. O objectivo do estudo teórico de estruturas é determinar este comportamento, o que leva a consideração simultânea de vários tipos de equações, tais como [Rei01]:

- (i) Equações de equilíbrio, envolvendo forças aplicadas, esforços e tensões.
- (ii) Relações constitutivas (relações tensões-deformações), envolvendo esforços ou tensões e deformações – descrevem o comportamento do material que constitui a estrutura.
- (iii) Relações cinemáticas (relações deformações-deslocamentos), envolvendo deformações e deslocamentos.
- (iv) Equações de compatibilidade, envolvendo deslocamentos e destinadas a garantir que a estrutura respeita as suas ligações (dos vários elementos entre si e com o exterior).

2.2 Análises lineares e não lineares

Consoante os diferentes problemas em estudo, o comportamento de uma estrutura pode ser “modelado” de várias formas, através da adopção de diferentes hipóteses que incidem sobre as características das equações referidas [Rei01]. A cada modelo de comportamento estrutural corresponde um tipo de análise estrutural diferente.

O tipo de análise estrutural mais simples designa-se por “análise linear de estruturas”, esta baseia-se na hipótese de todas as equações serem lineares, deste modo teremos [Rei01]:

- (i) A linearidade física – relações constitutivas lineares, i.e., materiais elásticos lineares.

- (ii) A linearidade geométrica – equações de equilíbrio escritas na configuração indeformada da estrutura e relações cinemáticas lineares, i.e., a “hipótese dos pequenos deslocamentos”

Este tipo de análise não permite, no entanto, identificar e analisar problemas de instabilidade, o que advém do facto de a origem destes fenómenos ser geometricamente não linear.

Uma “análise não linear de estruturas” é necessária em diferentes casos, que podem ser agrupados nas seguintes categorias principais [Ans07]:

- (i) Não linearidades físicas – relações tensão-extensão não lineares de materiais no domínio plástico e no domínio elástico, estas podem ser provocadas por diferentes factores, como o historial da carga (resposta elasto-plástica), as condições ambientais (temperatura), ou a quantidade de tempo em que a carga é aplicada.
- (ii) Não Linearidades Geométricas – estruturas sujeitas a grandes deformações, sofrem grandes alterações geométricas, o que pode causar a resposta não linear da estrutura (exemplo ilustrativo presente na Fig. 2-1). A não linearidade geométrica é caracterizada por grandes deslocamentos e/ou rotações [Rei01]. Neste caso são necessárias equações de equilíbrio na sua configuração deformada e/ou a consideração de relações cinemáticas não lineares.

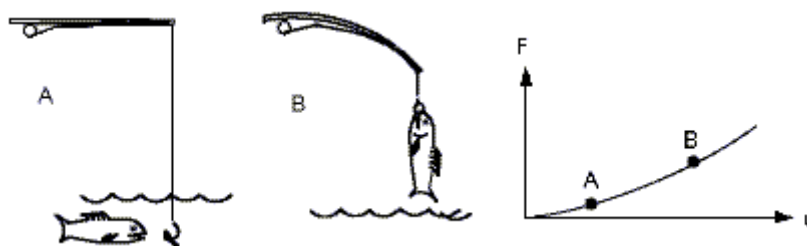


Fig. 2-1-Exemplo de um comportamento não linear (cana de pesca) [Ans07].

Para compreender o comportamento característico de uma viga sujeita aos tipos de análise explicados, considere-se o exemplo [Rei01] ilustrado na Fig. 2-2. A Fig. 2-2 a) ilustra um pilar submetido a uma força vertical de compressão e de valor P , que actua com uma excentricidade em relação ao seu eixo. Na Fig. 2-2 b) é possível observar a configuração deformada do pilar e na Fig. 2-2 c) as trajectórias de equilíbrio (no plano P –

δ , onde δ é o deslocamento horizontal do topo do pilar), obtidas, respectivamente por meio da análise linear e de uma análise geometricamente não linear (equilíbrio de momentos estabelecido na configuração deformada da Fig. 2-2 b)). A não linearidade da segunda trajectória resulta da interacção que existe entre os valores do deslocamento δ e os momentos flectores que actuam no pilar, onde $M(x) = P[e + \delta - w(x)]$ em contraste com o da primeira trajectória onde $M(x) = Pe$. Na Fig. 2-2 c) é possível constatar que, para valores de P elevados, a influência da não linearidade geométrica é extremamente significativa. Os erros associados aos valores de δ fornecidos pela análise linear aumentam com o valor de P e são sempre contra a segurança, ou seja, sempre inferiores aos valores exactos.

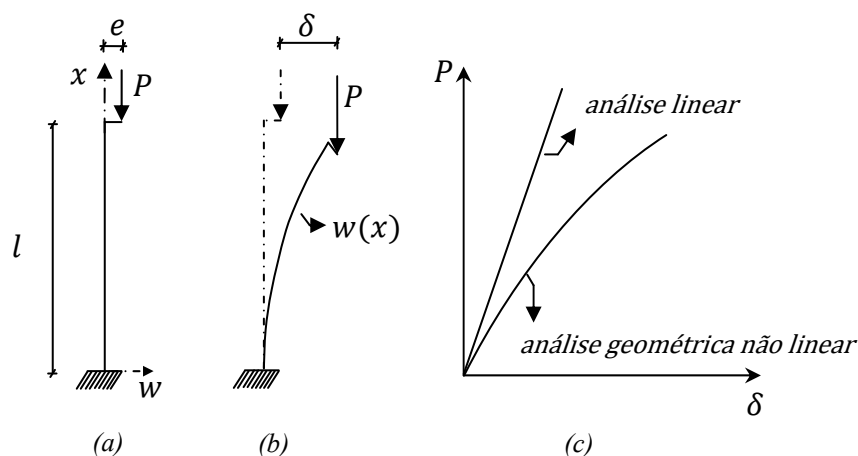


Fig. 2-2-Comportamento de um pilar comprimido excentricamente. a) Configuração indeformada. b) Configuração deformada. c) Trajectórias de equilíbrio. [Rei01].

2.3 Método dos elementos finitos em vigas

O método dos elementos finitos é utilizado em diversos campos de aplicação, para resolver problemas onde a modelação matemática origina um conjunto de equações diferenciais parciais (equações lineares, equações não lineares, entre outros) [Bab89]. Este método pode ser considerado como uma aplicação (ao nível dos elementos) de um método variacional, ou seja, que se baseia em princípios variacionais, tais como o princípio dos trabalhos virtuais, o princípio da energia potencial total estacionária ou o princípio de Hamilton. Cada um destes princípios pode originar diferentes soluções aproximadas de problemas [Par98].

A formulação do método dos elementos finitos em vigas pode por isso obter-se de vários modos. Tal como se observou, um deles consiste em utilizar o princípio da energia potencial total estacionária, o qual se aplica a problemas de equilíbrio estático.

$$\pi = U + V \quad (2.1)$$

Onde π é a energia potencial total, U a energia de deformação e V a energia potencial das forças aplicadas. Este princípio determina que uma posição de equilíbrio é caracterizada pelo facto de corresponder a um ponto de estacionaridade da energia potencial total.

2.3.1 Energia de deformação (U)

A energia de deformação U contida num corpo de volume v é [Men06]:

$$U = \frac{1}{2} \int_v \sigma_{ij} \varepsilon_{ij} dv \quad (2.2)$$

Na equação anterior σ_{ij} e ε_{ij} são respectivamente os tensores das tensões e das deformações em cada ponto do corpo. Numa viga de uma estrutura plana, o cálculo da energia de deformação pode dividir-se em duas componentes, uma associada ao esforço axial e outra associada ao momento flector, ou seja, $U = U^A + U^F$ [Men03]. Isto acontece porque os estados de tensão e extensão associados a cada uma delas podem considerar-se independentes. As parcelas da energia de deformação associadas às tensões tangenciais e às deformações provocadas pelo esforço transversal na viga podem ser desprezadas quando estas se consideram com esbelteza elevada.

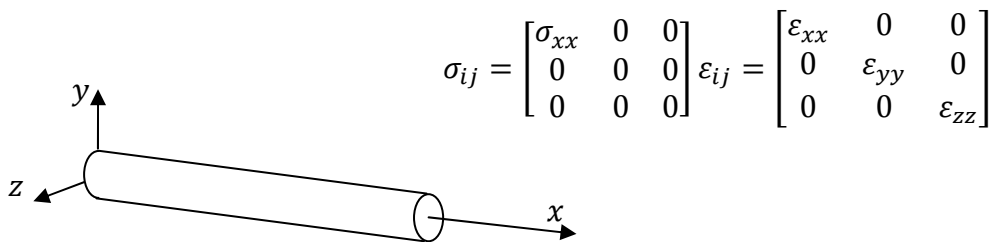


Fig. 2-3-Elemento Finito com referencial de eixos (x y z).

Na Fig. 2-3, pode observar-se o referencial (xyz) utilizado e os tensores das tensões e extensões que caracterizam o estado de tensão e de extensão em cada ponto da

viga. Designam-se os deslocamentos de um ponto segundo x , y por u , v respectivamente e a rotação do eixo da viga em torno de z por θ .

2.3.1.1 Energia de deformação associada ao esforço axial

Numa viga apenas com carregamento axial pode considerar-se o estado de tensão e de extensão uniforme para todos os pontos da mesma secção transversal, sendo $\sigma_{xx} = P/A$ onde P é igual ao esforço axial e A é a área da secção. Considerando comportamento elástico linear, $\sigma_{xx} = E\varepsilon_{xx}$, onde E é o módulo de Young. O único produto diferente de zero em $\sigma_{ij}\varepsilon_{ij}$ é $\sigma_{xx}\varepsilon_{xx}$, logo a equação (2.2) dá origem a:

$$U = \frac{1}{2} \int_0^L \int_A \sigma_{xx} \varepsilon_{xx} dA dx = \frac{1}{2} \int_0^L EA \varepsilon_{xx}^2 dx = \frac{1}{2} \int_0^L EA u_{,x}^2 dx \quad (2.3)$$

Assim, ε_{xx} em qualquer ponto da secção é igual à derivada $u_{,x} = \frac{\partial u}{\partial x}$ do deslocamento segundo x .

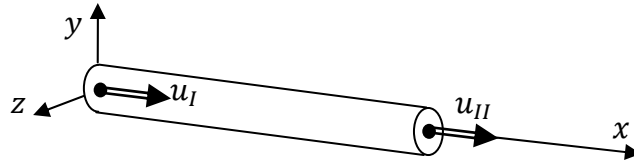


Fig. 2-4-Elemento Finito de uma viga com carregamento axial

Na Fig. 2-4 está representado um elemento finito com dois nós, onde u_I e u_{II} são os deslocamentos segundo x desses nós. Devido ao carregamento axial existirá deslocamento axial u dos pontos situados sobre o eixo da viga. Assumindo que esse deslocamento varia linearmente entre as extremidades tem-se:

$$u = a_0 + a_1 x \quad (2.4)$$

Utilizando duas funções de forma, N_I e N_{II} , o deslocamento u em qualquer ponto da viga será dado por:

$$u = u_I N_I + u_{II} N_{II} = \sum_{J=1}^2 u_J N_J = u_J N_J \quad (2.5)$$

A expressão analítica para estas funções pode ser facilmente obtida a partir do polinómio (2.4) e das condições fronteira da viga da seguinte forma:

$$x = 0 \Rightarrow u = a_0 = u_I$$

$$x = L \Rightarrow u = a_0 + a_1L = u_{II}$$

Tem-se que: $a_0 = u_I$ e $a_1 = \frac{u_{II}-u_I}{L}$. Substituindo no polinómio (2.4) obtém-se:

$$u = \left(1 - \frac{x}{L}\right)u_I + \frac{x}{L}u_{II} \quad (2.6)$$

As duas funções de forma N_I e N_{II} estão representadas na Fig. 2-5.

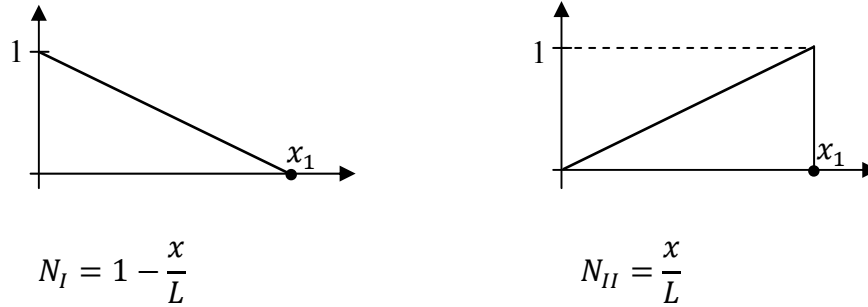


Fig. 2-5-Funções de forma

As derivadas das duas funções representadas são respectivamente:

$$N_{I,1} = -\frac{1}{L}$$

$$N_{II,1} = \frac{1}{L}$$

Utilizando as funções de forma, e sabendo que $u_{,x} = u_J N_{J,x}$, pode escrever-se a energia de deformação (2.3) no elemento:

$$U = \frac{1}{2} \int_0^L EA (u_J N_{J,x})^2 dx = \frac{1}{2} u_J \int_0^L EA N_{J,x} N_{K,x} dx u_K \quad (2.7)$$

Esta expressão fica mais clara se for utilizada uma representação matricial das várias quantidades envolvidas, deste modo, é possível definir a extensão segundo x da seguinte forma:

$$\varepsilon_{xx} = u_{,x} = \sum_{J=1}^2 u_J \frac{\partial N_J}{\partial x} = u_J \frac{\partial N_J}{\partial x} = \{u_I \quad u_{II}\} \begin{Bmatrix} \frac{\partial N_I}{\partial x} \\ \frac{\partial N_{II}}{\partial x} \end{Bmatrix} \quad (2.8)$$

A expressão (2.7) pode ser escrita do seguinte modo na forma matricial:

$$U = \frac{1}{2} \{u_I \quad u_{II}\} \left(\int_0^L \begin{Bmatrix} \frac{\partial N_I}{\partial x} \\ \frac{\partial N_{II}}{\partial x} \end{Bmatrix} \begin{Bmatrix} \frac{\partial N_I}{\partial x} & \frac{\partial N_{II}}{\partial x} \end{Bmatrix} EA dx \right) \begin{Bmatrix} u_I \\ u_{II} \end{Bmatrix} \quad (2.9)$$

Ou, de outra forma:

$$U = \frac{1}{2} u_J K_{JK} u_K = \frac{1}{2} \{u_I \quad u_{II}\} K_{JK} \begin{Bmatrix} u_I \\ u_{II} \end{Bmatrix} \quad (2.10)$$

Onde:

$$K_{JK} = \int_0^L N_{J,x} N_{K,x} EA dx \quad (2.11)$$

A equação (2.11) representa a matriz rigidez que contabiliza a parcela da energia de deformação elástica devida ao carregamento axial. Esta matriz pode ser obtida escrevendo a referida equação sob a forma matricial, para uma viga de secção transversal uniforme:

$$K_{JK} = \int_0^L \begin{Bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{Bmatrix} \begin{Bmatrix} -\frac{1}{L} & \frac{1}{L} \end{Bmatrix} EA dx = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \quad (2.12)$$

2.3.1.2 Energia de deformação associada ao momento flector

A flexão nos planos xy e xz pode ser estudada independentemente. Considera-se apenas o plano xy , que a secção da viga é uniforme e ainda que é válida a lei de Hooke.

A deformação que ocorre numa secção é função apenas do deslocamento transversal, y . Da teoria de vigas sabe-se que $\sigma_{xx} = -\frac{M}{I} y$. Então:

$$\begin{aligned}
 U &= \frac{1}{2} \int_0^L \int_A \sigma_{xx} \varepsilon_{xx} dA dx \Leftrightarrow \\
 &\Leftrightarrow U = \frac{1}{2} \int_0^L \int_A \frac{\sigma_{xx}^2}{E} dA dx \Leftrightarrow \\
 &\Leftrightarrow U = \frac{1}{2} \int_0^L \int_A \frac{M^2}{I^2 E} y^2 dA dx \Leftrightarrow \\
 &\Leftrightarrow U = \frac{1}{2} \int_0^L \frac{M^2}{I^2 E} \left(\int_A y^2 dA \right) dx \Leftrightarrow \\
 &\Leftrightarrow U = \frac{1}{2} \int_0^L \frac{M^2}{IE} dx = \frac{1}{2} \int_0^L EI v_{,xx}^2 dx
 \end{aligned} \tag{2.13}$$

Para esta simplificação tem-se em conta que $I = \int_A y^2 dA$ e que $v_{,xx} = \frac{M}{EI}$.

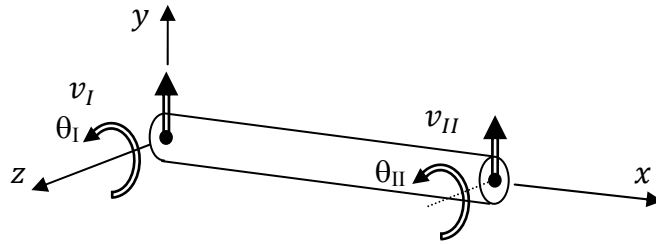


Fig. 2-6- Elemento Finito com dois nós

Considere-se o elemento finito com dois nós, como o apresentado na Fig. 2-6, v_I e v_{II} são os deslocamentos segundo y desses nós e θ_1 e θ_2 as rotações em torno de z do eixo da viga nesses nós. Assumindo que a elástica da viga será representada por um polinómio de terceiro grau, tem-se:

$$v = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \tag{2.14}$$

Pode exprimir-se v utilizando funções de forma, N_j . Contudo, uma vez que o polinómio tem quatro parâmetros, são necessárias quatro funções de forma e os deslocamentos v em quatro nós.

$$v = v_I N_I + \theta_I N_{II} + v_{II} N_{III} + \theta_{II} N_{IV} = u_j N_j \tag{2.15}$$

Na expressão anterior, u_j representa o vector que contém os deslocamentos generalizados nos nós do elemento de viga, ou seja:

$$u_j = \{v_I \quad \theta_I \quad v_{II} \quad \theta_{II}\} \quad (2.16)$$

Como vimos para o caso da viga com carregamento axial é possível obter estas funções de forma explícita. Considerando as condições fronteira:

$$\begin{aligned} x = 0 & \Rightarrow v = a_0 = v_I \\ x = 0 & \Rightarrow v_{,x} = a_1 = \theta_I \\ x = L & \Rightarrow v = a_0 + a_1L + a_2L^2 + a_3L^3 = v_{II} \\ x = L & \Rightarrow v_{,x} = a_1 + 2a_2L + 3a_3L^2 = \theta_{II} \end{aligned}$$

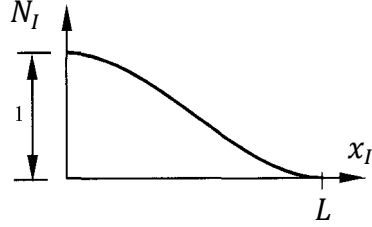
É possível então obter os coeficientes a_0 , a_1 , a_2 e a_3 :

$$\begin{aligned} a_0 &= v_I \\ a_1 &= \theta_I \\ a_2 &= -\frac{3}{L^2}v_I - \frac{2}{L}\theta_I + \frac{3}{L^2}v_{II} - \frac{1}{L}\theta_{II} \\ a_3 &= \frac{2}{L^3}v_I + \frac{1}{L^2}\theta_I - \frac{2}{L^3}v_{II} + \frac{1}{L^2}\theta_{II} \end{aligned}$$

Substituindo os coeficientes no polinómio inicial (2.15) é possível obter v como função de u_j .

$$\begin{aligned} v &= \frac{2}{L^3}v_Ix^3 + \frac{1}{L^2}\theta_Ix^3 - \frac{2}{L^3}v_{II}x^3 + \frac{1}{L^2}\theta_{II}x^3 - \frac{3}{L^2}v_Ix^2 - \frac{2}{L}\theta_Ix^2 \\ &\quad + \frac{3}{L^2}v_{II}x^2 - \frac{1}{L}\theta_{II}x^2 + \theta_Ix + v_I \end{aligned} \quad (2.17)$$

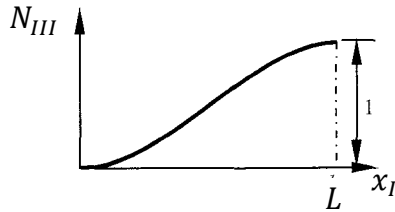
Agrupando os termos referentes a v_I , v_{II} , θ_I e θ_{II} obtêm-se os *polinômios de Hermite*:



$$N_I = 1 - \frac{3}{L^2}x^2 + \frac{2}{L^3}x^3$$



$$N_{II} = x - \frac{2}{L}x^2 + \frac{1}{L^2}x^3$$



$$N_{III} = \frac{3}{L^2}x^2 - \frac{2}{L^3}x^3$$



$$N_{IV} = -\frac{1}{L^2}x^2 + \frac{1}{L^2}x^3$$

(2.18)

A energia de deformação elástica da viga, é expressa em (2.13) como função das segundas derivadas do deslocamento transversal y . Derivando a equação (2.15) obtêm-se:

$$v_{,xx} = v_I N_{I,xx} + \theta_I N_{II,xx} + v_{II} N_{III,xx} + \theta_{II} N_{IV,xx} = u_J N_{J,xx} \quad (2.19)$$

É necessário então obter as derivadas das funções de forma. O cálculo analítico destas derivadas permite obter:

$$N_{I,xx} = -\frac{6}{L^2} + \frac{12}{L^3}x$$

$$N_{II,xx} = -\frac{4}{L} + \frac{6}{L^2}x$$

$$N_{III,xx} = \frac{6}{L^2} - \frac{12}{L^3}x$$

$$N_{IV,xx} = -\frac{2}{L} + \frac{6}{L^2}x$$

Substituindo na equação (2.13) $v_{,xx}$ por $u_J N_{J,xx}$ obtém-se:

$$U = \frac{1}{2} \int_0^L EI (u_J N_{J,xx})^2 dx = \frac{1}{2} u_J \int_0^L EIN_{J,xx} N_{K,xx} dx u_k \quad (2.20)$$

Que também pode ser escrito da seguinte forma:

$$U = \frac{1}{2} u_J K_{JK} u_k$$

Nesta formulação surge K_{JK} que representa a matriz de rigidez que contabiliza a parcela da energia de deformação elástica devida à flexão.

Deste modo:

$$K_{JK} = \int_0^L EIN_{J,xx} N_{K,xx} dx \quad (2.21)$$

Os elementos desta matriz podem ser calculados escrevendo primeiramente a equação (2.19) sob a forma matricial:

$$v_{,xx} = \{v_I \quad \theta_I \quad v_{II} \quad \theta_{II}\} \begin{Bmatrix} -\frac{6}{L^2} + \frac{12}{L^3}x \\ \frac{4}{L} + \frac{6}{L^2}x \\ \frac{6}{L^2} - \frac{12}{L^3}x \\ -\frac{2}{L} + \frac{6}{L^2}x \end{Bmatrix} \quad (2.22)$$

Realizando os produtos e a integração que surgem na expressão da matriz de rigidez (2.21), obtém-se:

$$K_{JK} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (2.23)$$

As duas matrizes de rigidez que se encontram em (2.12) e (2.23), associadas à tracção/compressão e à flexão, podem ser combinadas numa só.

O elemento de viga da Fig. 2-4 contabiliza unicamente a energia de deformação associada à tracção/compressão. O cálculo dessa energia depende apenas do deslocamento segundo o eixo x de cada nó do elemento. Diz-se que cada nó tem um grau de liberdade, uma vez que o elemento tem dois nós, tem um total de dois graus de liberdade. Por este facto o vector de deslocamentos do elemento é um vector 2×1 e a sua matriz de rigidez tem a dimensão 2×2 .

De um modo análogo, diz-se que cada nó do elemento de viga da Fig. 2-6 tem dois graus de liberdade e que esse elemento tem um total de quatro graus de liberdade.

Verifica-se que os graus de liberdade associados à tracção/compressão são independentes dos graus de liberdade associados à flexão. Agrupando num vector de deslocamentos generalizados todos os seis graus de liberdade obtém-se:

$$u_j = \{u_I \quad v_I \quad \theta_I \quad u_{II} \quad v_{II} \quad \theta_{II}\} \quad (2.24)$$

A energia de deformação continua a ser obtida pela expressão:

$$U = \frac{1}{2} u_j K_{JK} u_k \quad (2.25)$$

A matriz de rigidez do elemento de viga com seis graus de liberdade, obtida através da combinação das suas matrizes de rigidez enunciadas anteriormente será:

$$K_{JK} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (2.26)$$

2.3.2 Energia potencial das forças aplicadas (V)

A energia potencial de uma força aplicada sobre um corpo é igual ao produto da força pelo deslocamento do seu ponto de aplicação, afectado do sinal negativo (quando o ponto de aplicação da força se desloca no sentido da força, diminui a energia potencial).

2.3.2.1 Energia potencial para forças e momentos aplicados nos nós

Considerando forças aplicadas sobre os nós de um elemento finito com três graus de liberdade por nó e dois nós, a energia potencial associada a essas forças será:

$$V = -u_j F_j \quad (2.27)$$

Nesta equação, tem-se F_j , que representa o vector contendo as forças generalizadas aplicadas nos nós do elemento, sendo f_i e p_i as forças aplicadas no nó i segundo x e y respectivamente e m_i o momento no nó i em torno de z . Para a situação enunciada temos:

$$u_j = \{u_I \quad v_I \quad \theta_I \quad u_{II} \quad v_{II} \quad \theta_{II}\} \quad F_j = \left\{ \begin{array}{c} f_I \\ p_I \\ m_I \\ f_{II} \\ p_{II} \\ m_{II} \end{array} \right\} \quad (2.28)$$

2.3.2.2 Energia potencial para cargas axiais

Um caso particular ocorre quando o ponto de aplicação de uma força actuando segundo o eixo da viga sofre deslocamento devido à flexão existente na viga. Este facto vai implicar um comportamento não linear da viga.

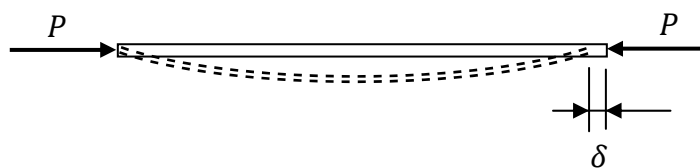


Fig. 2-7-Viga com cargas axiais aplicadas

Este tipo de comportamento, que já teve uma primeira abordagem na secção 2.2, é descrito na Fig. 2-7 e na Fig. 2-8.

Neste caso, o comportamento está associado ao facto dos deslocamentos serem suficientemente grandes para que não possa ser desprezada a modificação da posição do ponto de aplicação das forças axiais, que ocorre durante a deformação.

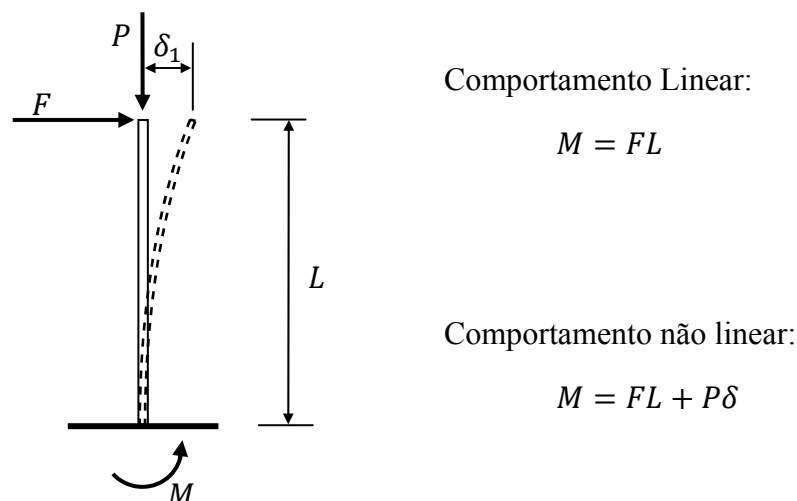


Fig. 2-8-Comportamento linear e não linear

Este comportamento não linear também se diz comportamento geometricamente não linear, ou seja, identifica-se com as não linearidades geométricas referidas na secção 2.2. Esta designação é importante para o diferenciar do comportamento não linear com origem na deformação plástica do material, designado por comportamento materialmente não linear, que se insere nas não linearidades físicas identificadas na secção 2.2. Este último comportamento não é abordado na presente dissertação.

Para pequenos deslocamentos e rotações, é usual considerar que as forças são aplicadas na configuração inicial da viga. Assume-se também que o deslocamento dos pontos de aplicação dessas forças durante a deformação não afecta os esforços internos. Obtém-se assim uma relação linear entre forças e deslocamentos. O mesmo é equivalente a considerar que os integrais realizados anteriormente no cálculo da energia de deformação são sempre realizados na configuração inicial.

Se os deslocamentos e rotações forem grandes, então o efeito da alteração do ponto de aplicação das forças externas deve ser considerado. Para vigas com cargas axiais, tal

pode ser conseguido contabilizando a variação da energia potencial devida à modificação de comprimento da viga por flexão.

O deslocamento δ que se apresenta na Fig. 2-7 pode ser aproximado por:

$$\delta = \frac{1}{2} \int_0^L v_{,x}^2 dx \quad (2.29)$$

A energia potencial associada à força axial P será:

$$V = -P\delta = -\frac{P}{2} \int_0^L v_{,x}^2 dx \quad (2.30)$$

Esta expressão pode ser discretizada recorrendo às funções de forma da viga sujeita à flexão:

$$V = -\frac{P}{2} u_J \int_0^L N_{J,x} N_{K,x} dx u_K \quad (2.31)$$

A energia potencial pode também ser representada da seguinte forma:

$$V = -\frac{P}{2} u_J G_{JK} u_K \quad (2.32)$$

Nesta expressão surge a matriz de rigidez geométrica ou matriz geométrica da viga, G_{JK} . Quando P provoca compressão na viga, os coeficientes de G , multiplicados por P devem ser subtraídos à matriz de rigidez K deduzida anteriormente, provocando uma diminuição da rigidez. No caso de a força P provocar tracção, ocorre um aumento de rigidez. A matriz geométrica é dada por:

$$G_{JK} = \int_0^L N_{J,x} N_{K,x} dx \quad (2.33)$$

$$G_{JK} = \begin{bmatrix} \frac{6}{5L} & \frac{1}{10} & -\frac{6}{5L} & \frac{1}{10} \\ \frac{1}{10} & \frac{2L}{15} & -\frac{1}{10} & \frac{L}{30} \\ -\frac{6}{5L} & -\frac{1}{10} & \frac{6}{5L} & -\frac{1}{10} \\ \frac{1}{10} & -\frac{L}{30} & -\frac{1}{10} & \frac{2L}{15} \end{bmatrix} \quad (2.34)$$

É importante realçar que os deslocamentos e rotações contidos no vector u_j são os indicados em (2.16) e a matriz G terá, por isso, a dimensão 4×4 . Contudo, pode sempre obter-se uma matriz com 6×6 elementos, compatível com (2.24) acrescentando duas linhas e duas colunas, com zeros, para os deslocamentos segundo x , (2.35).

$$G_{JK} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6}{5L} & \frac{1}{10} & 0 & -\frac{6}{5L} & \frac{1}{10} \\ 0 & \frac{1}{10} & \frac{2L}{15} & 0 & -\frac{1}{10} & \frac{L}{30} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{6}{5L} & -\frac{1}{10} & 0 & \frac{6}{5L} & -\frac{1}{10} \\ 0 & \frac{1}{10} & -\frac{L}{30} & 0 & -\frac{1}{10} & \frac{2L}{15} \end{bmatrix} \quad (2.35)$$

A equação (2.14) utilizada para representar a deformada da viga (equação da linha elástica) representa a solução de uma viga sujeita a cargas concentradas que apresente uma variação linear do momento flector, na ausência da força axial P . Como tal, utilizando a matriz de rigidez K_{JK} e a matriz geométrica G_{JK} apenas se poderão obter soluções aproximadas para problemas com cargas axiais.

2.3.3 Energia potencial total (π)

Obtidos todos os termos de U e V , a energia potencial total é agora uma função de várias incógnitas. Essas incógnitas são os deslocamentos e rotações nos nós da viga e estão arrumadas num vector de deslocamentos generalizados. A expressão geral para π é:

$$\pi = \frac{1}{2}u_J K_{JK} u_K - \frac{P}{2}u_J G_{JK} u_K - u_J F_J \quad (2.36)$$

Como enunciado anteriormente, uma posição de equilíbrio é caracterizada por corresponder a um ponto de estacionaridade da energia potencial total. Um ponto de estacionaridade existe quando:

$$\frac{d\pi}{du_J} = 0 \quad (2.37)$$

Derivando a equação (2.36) em ordem aos elementos do vector u_J obtém-se a seguinte equação de equilíbrio:

$$(K_{JK} - P G_{JK})u_K = F_J \quad (2.38)$$

A notação indicial usada até aqui é vantajosa para explicar a formulação por elementos finitos. Contudo é conveniente escrever a equação (2.38) em notação matricial e passar a utilizar esta notação para desenvolver a formulação corrotacional estudada nesta dissertação. A equação (2.38) pode então ser escrita:

$$([K_e] - P[G_e])\{u_e\} = \{f_e\} \quad (2.39)$$

Onde $[K_e] \equiv K_{JK}$, $[G_e] \equiv G_{JK}$, $\{u_e\} = u_K$ e $\{f_e\} = F_J$ como definido na simbologia.

Nesta equação considera-se que a força P é positiva quando tem o sentido indicado na Fig. 2-7.

2.3.4 Transformação de coordenadas

A equação de equilíbrio (2.38) e (2.39) enunciada é válida para um único elemento, contém as matrizes de rigidez, $[K_e]$, e de rigidez geométrica, $[G_e]$, escritas no referencial local do elemento. Neste mesmo referencial estão também definidos o vector $\{u_e\}$ e $\{f_e\}$ que são respectivamente o vector de deslocamentos e o vector de forças.

Para que as matrizes tenham a forma indicada em (2.26) e (2.35), e de acordo com a Fig. 2-4 e Fig. 2-6, o referencial local de cada elemento é definido de modo que o eixo x tenha origem no primeiro nó e esteja alinhado deste para o segundo nó. O eixo y será perpendicular a x e estará contido no plano da estrutura, enquanto o eixo z é perpendicular a esse plano.

A necessidade de juntar ou “assemblar” as matrizes dos vários elementos viga numa única matriz de rigidez global obriga a definir, para além dos vários referenciais locais (um por cada elemento), um único referencial, válido para toda a estrutura. Este referencial é designado por referencial global (X, Y) .

Os vectores de forças e de deslocamentos indicados nas equações (2.28) e (2.39) podem ser transformados para o referencial global usando a matriz de transformação:

$$[T] = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.40)$$

Obtendo-se assim:

$$\{f\} = [T]^T \{f_e\} \Rightarrow \{f_e\} = [T]\{f\}$$

$$\{u\} = [T]^T \{u_e\} \Rightarrow \{u_e\} = [T]\{u\}$$

A equação (2.39) pode reescrever-se, em relação ao referencial global:

$$[T]^T([K_e] - P[G_e])\underbrace{[T]\{u\}}_{\{u_e\}} = \{f\} \quad (2.41)$$

Percebe-se então que a matriz de rigidez adicionada ou subtraída do produto da matriz de rigidez geométrica pela força axial no elemento, deve ser multiplicada à esquerda por $[T]^T$ e à direita por $[T]$ para passar a dizer respeito ao referencial global. A equação (2.39) pode então escrever-se com todos os vectores e matrizes expressas no referencial global:

$$[K(u)]\{u\} = \{f\} \quad (2.42)$$

Onde $[K(u)]$ é a matriz de rigidez em coordenadas globais.

A transformação de coordenadas deve ser feita para cada elemento da estrutura. A “assemblagem” da matriz de rigidez global é feita juntando as matrizes de rigidez de cada elemento referidas ao referencial global.

Em relação à matriz de transformação, esta pode ser calculada de forma eficiente uma vez conhecidas as diferenças entre as coordenadas X dos dois nós do elemento, dx , entre as coordenadas Y dos dois nós, dy , e o comprimento do elemento, l através da equação (2.43). A Fig. 2-9 ilustra o que foi referido anteriormente.

$$[T(u)] = \begin{bmatrix} dx/l & dy/l & 0 & 0 & 0 & 0 \\ -dy/l & dx/l & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & dx/l & dy/l & 0 \\ 0 & 0 & 0 & -dy/l & dx/l & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.43)$$

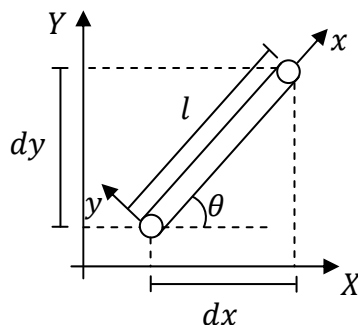


Fig. 2-9 - Origem da matriz transformação $[T]$

2.3.5 Análise não linear

O facto de se considerar que a equação de equilíbrio (2.42) é escrita na configuração indeformada da estrutura resulta na utilização da matriz de transformação, $[T]$, definida com base na posição inicial dos nós do elemento, para transformar o referencial global no referencial local. Se para além disto se desprezar a influência da força axial em cada elemento, P , na rigidez da estrutura, então obtém-se que a matriz de rigidez em coordenadas globais, $[K(u)]$ é constante para todos os valores das forças aplicadas e que o vector de deslocamentos pode ser obtido da equação (2.42) resolvendo um sistema de equações lineares.

Na presente dissertação considera-se que o sistema de equações definido em (2.42) é não linear. Isto é, não só é considerado o efeito da força axial P , como o cálculo da matriz de transformação, $[T]$, é efectuado em relação à posição dos nós do elemento quando este está deformado. Ou seja, $[T]$ e P dependem de u .

Para resolver este tipo de análises recorre-se ao método de Newton-Raphson, que é largamente utilizado na mecânica computacional [Bel00] e analisado na secção seguinte.

2.4 Método de Newton-Raphson

A existência de não linearidades resulta de não se verificar uma proporcionalidade directa entre forças e deslocamentos $F \neq Ku$, [Dia10]. Então para efectuar uma análise não linear, é necessário recorrer a um processo iterativo, uma vez que as equações que regem o comportamento da estrutura dependem da configuração deformada, logo as posições de equilíbrio da estrutura são actualizadas durante o processo. No algoritmo desenvolvido é

utilizado o método de Newton-Raphson [Ans07] para resolver a equação (2.42), método este que também é utilizado pelo programa *Ansys* para análises não lineares.

A Fig. 2-10 ilustra o método de Newton-Raphson. A variável Δf que se encontra na figura denomina-se por resíduo e o objectivo é minimizá-lo:

$$\{\Delta f\} = \{f\} - [K(u)]\{u\} \quad (2.44)$$

Como se pode observar na figura, com as consecutivas iterações o resíduo vai diminuindo, $\Delta f_2 < \Delta f_1$. Na primeira iteração, considera-se $\{u_0\} = 0$ e avalia-se a matriz rigidez $[K]$ na configuração indeformada. Resolvendo o sistema de equações lineares, obtém-se o primeiro incremento nos deslocamentos, $\{\Delta u_1\}$, com o qual se actualizam os deslocamentos. Tem-se portanto:

$$\{u_1\} = \{\Delta u_1\} \quad (2.45)$$

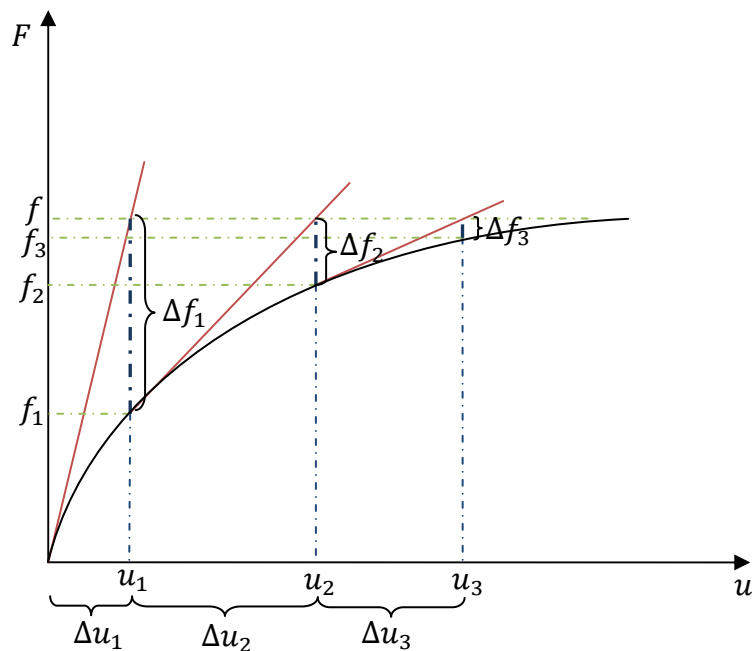


Fig. 2-10-Representação do processo iterativo de Newton-Raphson

Na posse deste dado e calculando a matriz rigidez para a configuração $\{u_1\}$, isto é, considerando as forças axiais nos elementos e as matrizes de transformação de coordenadas correspondentes à configuração $\{u_1\}$ é possível determinar o conjunto de

forças internas $\{f_1\}$ que equilibram a estrutura para essa configuração. Surge então para a primeira iteração:

$$[K(u_1)]\{u_1\} = \{f_1\} \quad (2.46)$$

Uma vez que $\{f_1\}$ é diferente do vector de forças que actuam na estrutura, $\{f\}$, o sistema não se encontra em equilíbrio. O resíduo será então:

$$\{\Delta f_1\} = \{f\} - [K(u_1)]\{u_1\} \quad (2.47)$$

Para a segunda iteração resolve-se o sistema de equações lineares:

$$[K(u_1)]\{\Delta u_2\} = \{\Delta f_1\} \quad (2.48)$$

Com este sistema é possível obter o novo incremento $\{\Delta u_2\}$, através do qual se actualizam os deslocamentos. O processo iterativo apresentado pode ser então brevemente descrito da seguinte forma:

- i) Em cada iteração, utilizando os últimos deslocamentos conhecidos, $\{u_n\}$, calcula-se a matriz rigidez, $[K(u_n)]$ e o vector de forças residuais, $\{\Delta f_n\}$.
- ii) Resolve-se o sistema de equações lineares, calcula-se o incremento nos deslocamentos, $\{\Delta u_{n+1}\}$ e actualizam-se os deslocamentos:

$$\{u_{n+1}\} = \{u_n\} + \{\Delta u_{n+1}\} \quad (2.49)$$

- iii) Repetição sucessiva das etapas anteriores até o resíduo ser suficientemente pequeno.

2.5 *Análise incremental iterativa*

A aplicação do método de Newton-Raphson no algoritmo em estudo sofreu adaptações que o transformam num processo iterativo-incremental. Ou seja, para permitir a

convergência para resultados correctos e conseguir traçar diagramas carga-deslocamento que mostrem o comportamento da estrutura, considera-se que as cargas são progressivamente aumentadas. Com isto procura-se obter a posição de equilíbrio para cada nível de carga, realizando diversas iterações de Newton-Raphson.

No algoritmo desenvolvido optou-se por dividir a carga total aplicada, f_T , por um número inteiro, designado de número de incrementos, N_{inc} . É utilizado o método de Newton-Raphson para um valor da carga aplicada, f :

$$f = \lambda f_T \quad \text{com:} \quad \lambda = \frac{1}{N_{inc}}$$

Após a convergência para uma posição de equilíbrio, o valor de λ é incrementado de $\frac{1}{N_{inc}}$. Isto repete-se até ocorrer convergência para o nível de carregamento total aplicado, f_T .

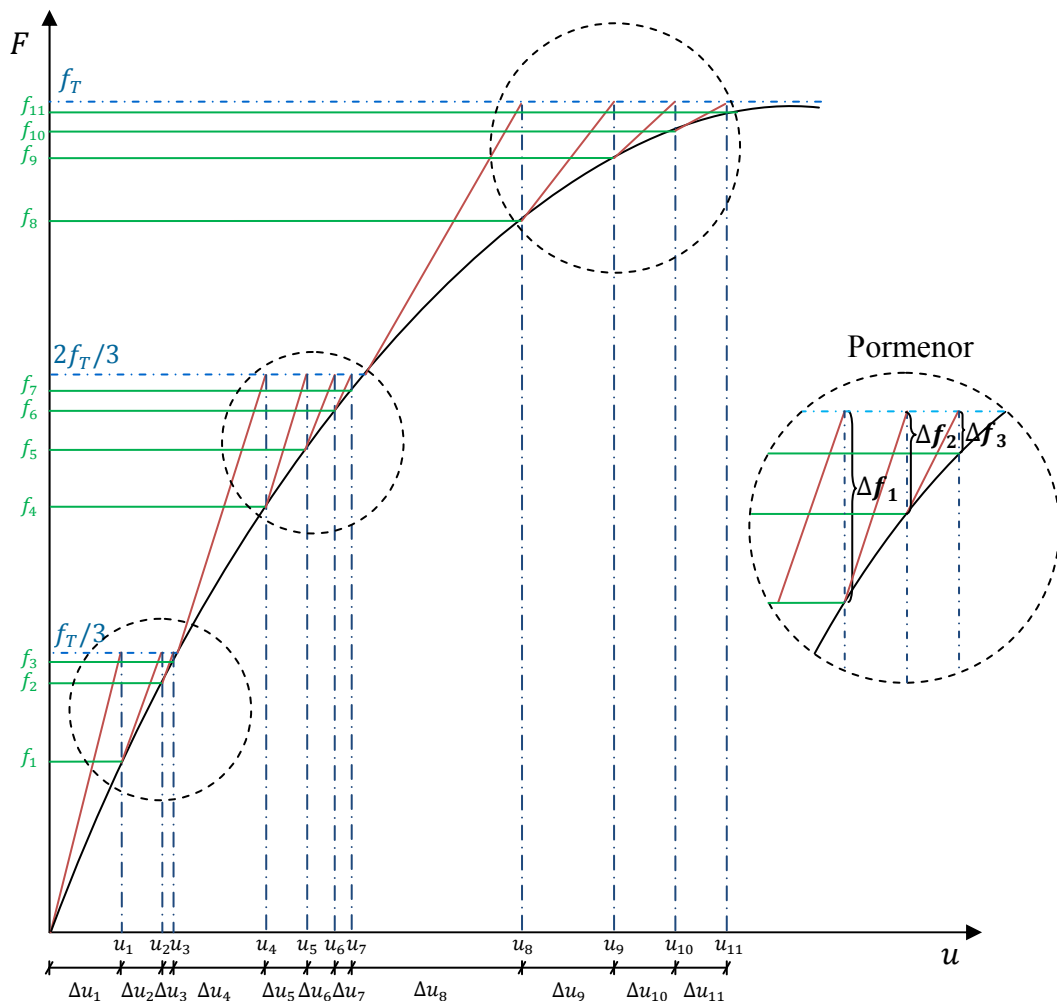


Fig. 2-11 - Método incremental iterativo.

A Fig. 2-11 ilustra o que foi apresentado. Nesta podemos observar que a carga total a aplicar, f_T , está dividida em três partes iguais e que passa por um processo iterativo até atingirmos cada um dos valores da força, $f_T/3$, $2f_T/3$ e f_T . Fazendo um paralelo com a Fig. 2-10, podemos observar que esta se repete três vezes no exemplo ilustrado na Fig. 2-11.

3 FORMULAÇÃO CORROTACIONAL EM VIGAS 2D

3.1 Formulação Corrotacional

A formulação corrotacional é baseada no pressuposto de que as vigas que constituem a estrutura sofrem grandes deslocamentos, afastando-se por isso significativamente da configuração inicial. A sua esbelteza é porém suficientemente elevada para que as deformações produzidas sejam de pequena amplitude, de forma a não ultrapassar o limite elástico do material e permitindo usar relações cinemáticas envolvendo deformações e deslocamentos lineares. Esta metodologia foi apresentada por Rankin e Brogan [Ran86] salientando a vantagem de permitir usar programas desenvolvidos para análise linear de estruturas em análises não lineares, através de transformações de coordenadas apropriadas. [Bat02] A metodologia referida é usada pelo programa *Ansys* para realizar análises de estruturas constituídas por vigas e cascas com grandes rotações.

3.1.1 Translações na formulação corrotacional

Os referidos autores ([Ran86]), explicam que qualquer campo de deslocamentos num corpo deformado pode ser decomposto numa translação de corpo rígido, numa rotação de corpo rígido e numa deformação. Na metodologia de elementos finitos utilizada, se apenas existir translação de corpo rígido, as tensões e deformações não são afectadas. No entanto, se existir rotação de corpo rígido, existem deformações associadas. Posto isto, na equação (3.1) surgem as grandezas envolvidas no deslocamento total para esta formulação. Este deslocamento, $\{u\}$, é igual à soma do deslocamento devido à deformação $\{u^d\}$ com o deslocamento devido à rotação de corpo rígido $\{u^r\}$ [Men03].

$$\{u\} = \{u^d\} + \{u^r\} \quad (3.1)$$

Para efectuar o cálculo dos deslocamentos dos nós associados às deformações, utilizamos as equações presentes em (3.2) [Ans07] ou (3.3) [Ran86]. No presente estudo utiliza-se a equação (3.2), que é utilizada pelo software onde vão serão efectuados os cálculos paralelos para posterior comparação.

O cálculo é efectuado separadamente para os graus de liberdade associados às translações e para os associados às rotações. Para uma correcta compreensão da equação (3.2) é conveniente analisar as diferentes quantidades envolvidas.

$$\{u_t^d\} = [R_n](\{X_0\} + \{u\}) - \{X_0\} \quad (3.2)$$

Nesta equação o vector $\{u_t^d\}$ contém a componente de translação do deslocamento devido à deformação e $[R_n]$ é a matriz de transformação correspondente à rotação de corpo rígido do referencial local inicial para o referencial local rodado do ângulo β . O vector $\{X_0\}$, por sua vez, contém as coordenadas dos nós do elemento no referencial global para a configuração inicial e o vector $\{u\}$ os deslocamentos dos nós no sistema de coordenadas globais. O índice n em $[R_n]$ indica que esta é calculada a cada iteração n .

De um modo análogo, na equação (3.3), surge $\{u_e^d\}$ referente ao vector dos deslocamentos que produzem deformação no elemento no referencial local e $[E]$ que representa a matriz de transformação do referencial global para o referencial local. O vector $\{u\}$ contém os deslocamentos no referencial global, $\{X_0\}$, as coordenadas para o sistema indeformado (configuração inicial) neste mesmo referencial e $\{X_e\}$ as mesmas coordenadas que $\{X_0\}$ mas no referencial local.

$$\{u_{t,e}^d\} = [E]^T(\{X_0\} + \{u\}) - \{X_e\} \quad (3.3)$$

É importante referir a grande diferença em relação a estas equações (presentes na bibliografia enunciada). A equação (3.2) calcula os deslocamentos associados às deformações no referencial global, por outro lado, a equação (3.3) calcula as mesmas quantidades no referencial local, ou seja, no referencial do elemento.

Para obter os deslocamentos no referencial do elemento, utilizando a equação (3.2) tem de ser utilizada a matriz de transformação $[T_n]$, ou seja, $\{u_{t,e}^d\} = [T_n]\{u_t^d\}$. Como tal, se definirmos a equação (3.2) em relação ao referencial local tem-se que:

$$\{u_{t,e}^d\} = [T_n]([R_n](\{X_0\} + \{u\}) - \{X_0\}) \quad (3.4)$$

A matriz $[T_n]$ é definida pela equação (2.40) e o índice n indica que é calculada para a iteração n .

As figuras seguintes explicam de forma explícita o que foi descrito anteriormente, ilustrando as diferentes coordenadas e deslocamentos. Na Fig. 3-1 a) está ilustrada a situação inicial de um elemento de viga com dois nós, onde é possível observar as coordenadas iniciais dos nós, $\{X_0\}$. Este vector, onde estão guardadas as coordenadas dos nós do elemento no referencial global para a configuração inicial representa-se da seguinte forma:

$$\{X_0\} = \begin{Bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \end{Bmatrix} \quad (3.5)$$

Na Fig. 3-1 b) ilustra-se uma configuração correspondente à iteração n do método de Newton-Raphson. Podemos visualizar na figura os deslocamentos totais dos nós, medidos no referencial global, $\{u\}$.

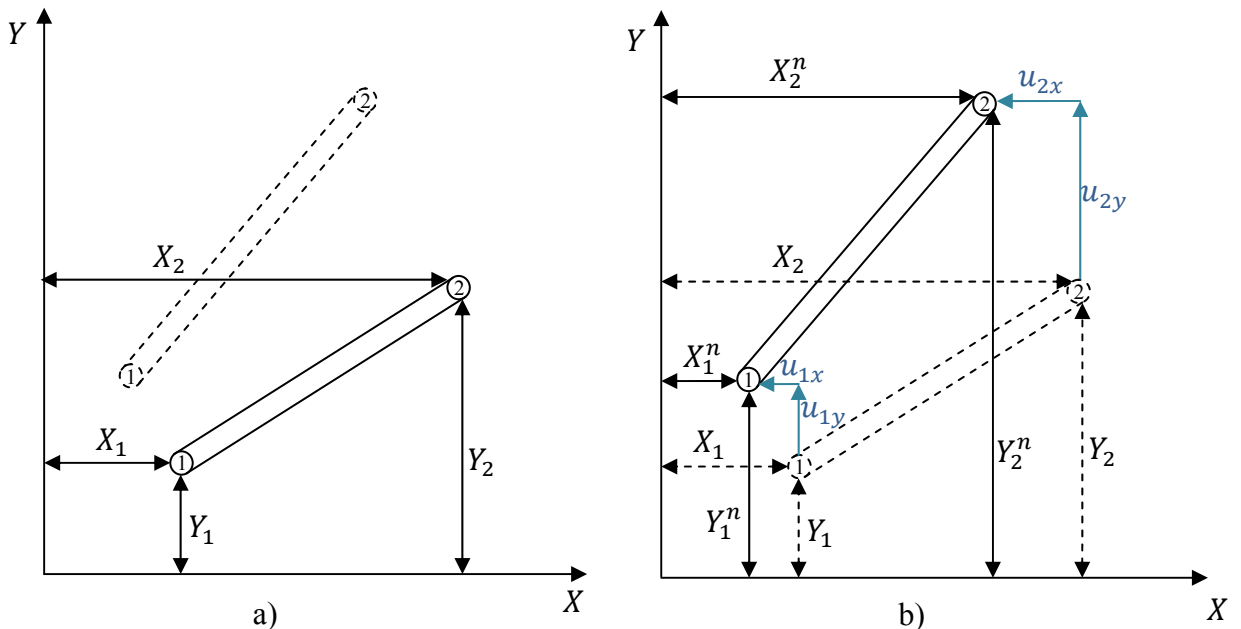


Fig. 3-1 a) Representação da configuração inicial do elemento de viga b) Representação do elemento de viga na iteração n .

Existem ainda na Fig. 3-1 dois vectores que importa referir, o vector $\{u\}$ que guarda os deslocamentos totais dos nós, medidos no referencial global e o vector $\{X_n\}$ que guarda as coordenadas dos nós do elemento no referencial global, para a iteração n .

Os vectores referidos encontram-se em (3.6) e (3.7) respectivamente:

$$\{u\} = \begin{Bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \end{Bmatrix} \quad (3.6)$$

$$\{X_n\} = \begin{Bmatrix} X_1^n \\ Y_1^n \\ X_2^n \\ Y_2^n \end{Bmatrix} \quad (3.7)$$

Sabemos então que as coordenadas ilustradas na Fig. 3-1 a) e b) se relacionam da seguinte forma:

$$\{X_n\} = \{X_0\} + \{u\}. \quad (3.8)$$

Na Fig. 3-2 ilustra-se a metodologia seguida para extrair do vector de deslocamentos a parte associada à rotação de corpo rígido. Pode observar-se que o movimento do elemento da configuração inicial para a configuração na iteração n pode ser decomposto numa translação de corpo rígido (o nó 1 desloca-se para a posição (X_1^n, Y_1^n) embora o elemento se mantenha paralelo à posição inicial e sem deformações), numa rotação de corpo rígido (o elemento roda em torno do nó 1 um ângulo β de forma que os dois nós fiquem sobre o eixo que vai de (X_1^n, Y_1^n) a (X_2^n, Y_2^n)) e finalmente numa deformação (o elemento deforma-se devido ao esforço axial e ao momento flector).

Considerando que a formulação utilizada torna as deformações independentes das translações de corpo rígido, pelo facto das deformações serem proporcionais à diferença entre deslocamentos dos nós e não aos deslocamentos, basta eliminar a componente dos deslocamentos associada à rotação do elemento do ângulo β . Tal é realizado medindo as coordenadas dos nós na iteração n num referencial que rodou também um ângulo β , como indica a Fig. 3-2. Um procedimento semelhante é sugerido por [Fel05].

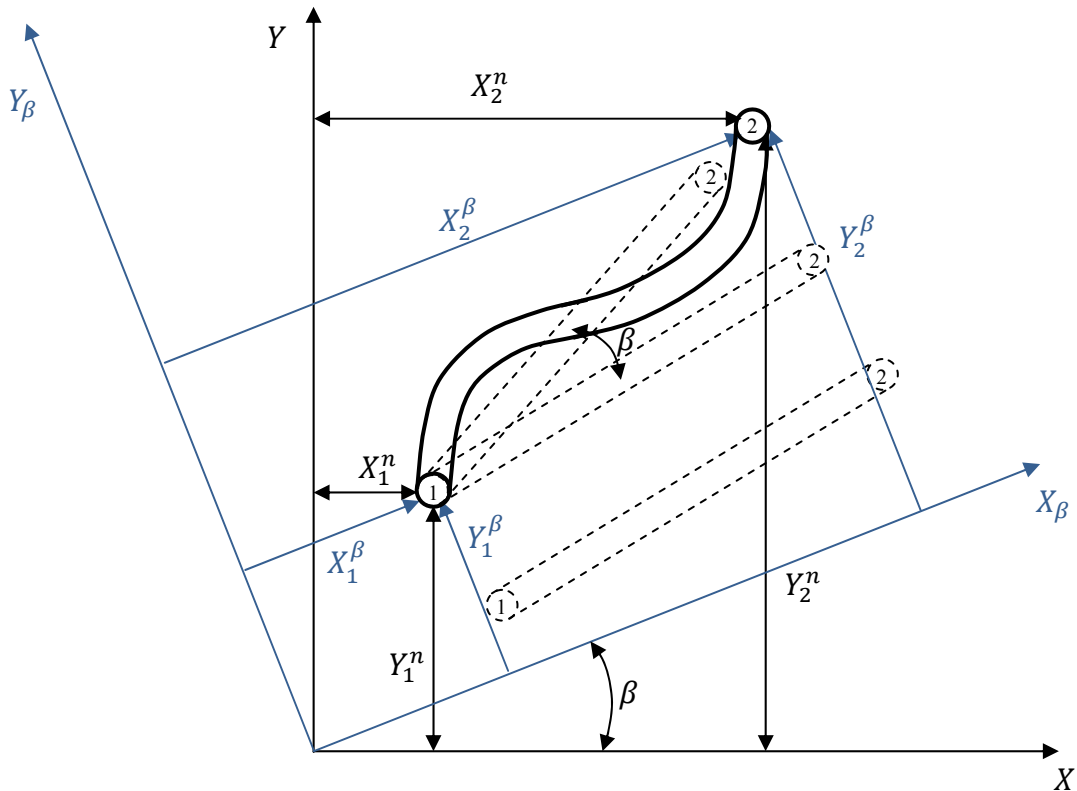


Fig. 3-2- Representação do elemento de viga após a rotação β .

O vector $\{X_\beta\}$ guarda as coordenadas dos nós do elemento no referencial global, rodado do ângulo β . É agora possível relacionar todas as coordenadas ilustradas na Fig. 3-1 a), na Fig. 3-1 b) e na Fig. 3-2:

$$\{X_\beta\} = [R_n]\{X_n\} \Leftrightarrow \{X_\beta\} = [R_n](\{X_0\} + \{u\}) \quad (3.9)$$

Na equação anterior tem-se:

$$[R_n] = \begin{bmatrix} \cos \beta & \sin \beta & 0 & 0 \\ -\sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & \cos \beta & \sin \beta \\ 0 & 0 & -\sin \beta & \cos \beta \end{bmatrix} \quad (3.10)$$

Para os deslocamentos anteriormente referidos, em (3.5) e (3.6) formula-se a equação (3.9) da seguinte forma matricial:

$$\begin{Bmatrix} X_1^\beta \\ Y_1^\beta \\ X_2^\beta \\ Y_2^\beta \end{Bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta & 0 & 0 \\ -\sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & \cos \beta & \sin \beta \\ 0 & 0 & -\sin \beta & \cos \beta \end{bmatrix} \left(\begin{Bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \end{Bmatrix} + \begin{Bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \end{Bmatrix} \right) \quad (3.11)$$

O vector dos deslocamentos que produzem deformação no elemento, $\{u_t^d\}$, é obtido subtraindo o vector $\{X_0\}$ ao vector $\{X_\beta\}$. Ou seja:

$$\{u_t^d\} = \{X_\beta\} - \{X_0\} \quad (3.12)$$

Este vector guarda os deslocamentos que produzem deformação no elemento, embora apenas os associados às translações dos nós e em relação ao referencial global.

3.1.2 Ângulos de rotação na formulação corrotacional

Para os graus de liberdade associados às rotações tem-se o vector $\{\alpha\}$ que guardará as rotações totais dos nós, medidos no referencial local. Subtraindo este vector, à rotação de corpo rígido do elemento, β , obtem-se o vector das rotações que produzem deformação no elemento $\{\alpha^d\}$. Ou seja:

$$\begin{aligned} \{\alpha^d\} &= \{\alpha\} - \{\beta\} \Leftrightarrow \\ \Leftrightarrow \begin{Bmatrix} \alpha_1^d \\ \alpha_2^d \end{Bmatrix} &= \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} - \{\beta\} \end{aligned} \quad (3.13)$$

Tem-se então o vector dos deslocamentos generalizados que produzem deformação no elemento, onde se encontram os deslocamentos associados às translações e os associados às rotações:

$$\{u^d\} = \begin{Bmatrix} u_{1x}^d \\ u_{1y}^d \\ \alpha_1^d \\ u_{2x}^d \\ u_{2y}^d \\ \alpha_2^d \end{Bmatrix} \quad (3.14)$$

Este vector pode ser calculado no referencial local através de:

$$\{u_e^d\} = [T_n]\{u^d\} \quad (3.15)$$

3.1.3 Determinação do ângulo β

O cálculo do ângulo β , que corresponde à rotação de corpo rígido do elemento levanta um problema que pode ser resolvido de dois modos distintos. O seu cálculo pode ser feito a partir das coordenadas actuais dos nós ou das suas rotações.

Utilizando a primeira opção referida, verifica-se que ocorre uma indeterminação pois é impossível distinguir rotações de corpo rígido positivas de negativas e ainda a verdadeira grandeza da rotação, uma vez que esta pode ser um múltiplo de β , na forma $\beta + 2k\pi$, com $k \in \mathbb{Z}$.

Na Fig. 3-3 é possível perceber que o cálculo pelas coordenadas actuais retorna o mesmo valor β para situações distintas, sendo impossível determinar qual a correcta.

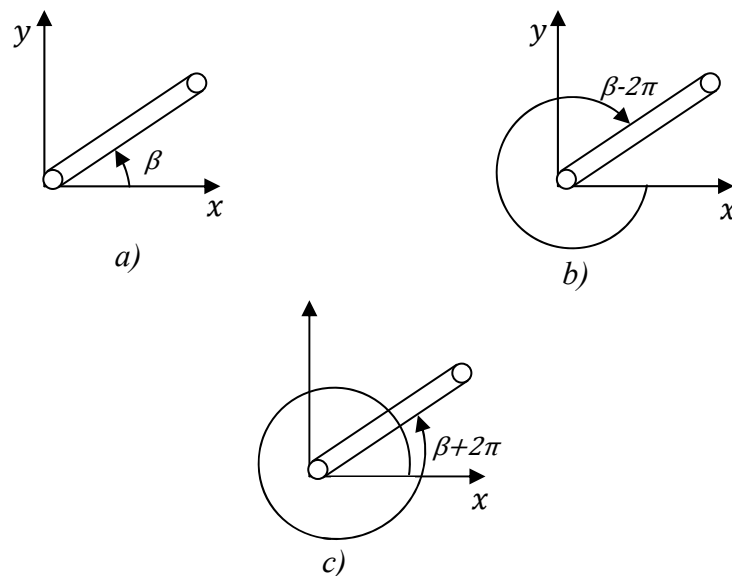


Fig. 3-3- Problemática do cálculo do ângulo β .

Ao desenvolver o algoritmo que realiza a análise de vigas por esta metodologia, optou-se por eliminar esta ambiguidade analisando a rotação média dos nós do elemento na iteração anterior à actual, desta forma:

$$\alpha_{MÉDIO} = \frac{\alpha_1 + \alpha_2}{2} \quad (3.16)$$

Utilizando a rotação média dos nós é possível determinar qual é o valor correcto de β de entre todos os possíveis.

Para isso considerou-se que o ângulo β pertence a uma família de valores $\beta + 2k\pi$, onde $k = -3, -2, -1, 0, +1, +2, +3$. Isto é, arbitrou-se que a rotação β deverá estar compreendida entre -6π e $+6\pi$.

O problema de obter o valor correcto de k pode ser resolvido comparando dois valores, o do ângulo $\beta + 2k\pi$, onde se desconhece o valor de k e o do ângulo de rotação médio dos nós do elemento, $\alpha_{MÉDIO}$, e procurando o valor de k que minimiza d:

$$d = \beta + 2k\pi - \alpha_{MÉDIO} \quad (3.17)$$

Desta forma é possível encontrar a melhor aproximação possível para o valor da rotação e resolvem-se os problemas detectados e ilustrados na Fig. 3-3.

A segunda opção considerada para a construção do algoritmo consiste em utilizar directamente o $\alpha_{MÉDIO}$, fazendo $\beta = \alpha_{MÉDIO}$. Este valor, que é a média dos ângulos α_1 e α_2 apresentados na Fig. 3-4, vai sendo incrementado ao longo das diferentes iterações e para os diferentes elementos.

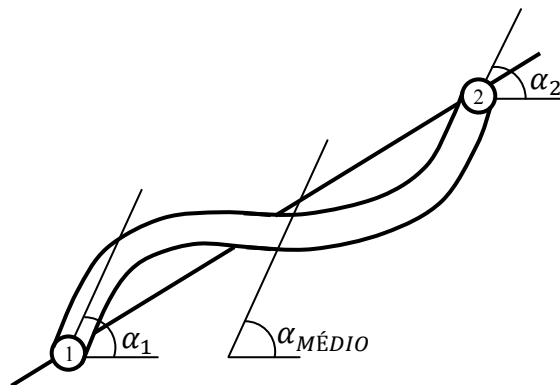


Fig. 3-4 - Cálculo do $\alpha_{MÉDIO}$

3.2 Esforços internos

Em relação aos esforços internos foram analisadas duas maneiras distintas de efectuar o cálculo pretendido. A primeira das quais, executa o cálculo dos mesmos a partir do vector de deslocamentos nodais e da matriz de rigidez do elemento no referencial local, pela equação:

$$[K_e]\{u_e\} = \{f_e\} \quad (3.18)$$

Neste caso, para o cálculo dos esforços internos, que se designam por $\{F_{ea}\}$ no algoritmo em estudo, utilizam-se os deslocamentos nodais associados às deformações (3.15), referidos anteriormente como $\{u_e^d\}$. Estes deslocamentos estão associados ao referencial local. Tem-se então que:

$$\{F_{ea}\} = [k]\{u_e^d\} \quad (3.19)$$

Uma outra metodologia foi utilizada para o cálculo dos esforços internos, directamente a partir do alongamento e das rotações. De facto, ao considerar que o referencial local na iteração n passa nos dois nós do elemento obtém-se sempre que o vector de deslocamentos (3.15) tem u_{1y}^d e u_{2y}^d iguais a zero, e como a origem está sempre no primeiro nó, u_{1x}^d é também igual a zero.

A deformação consiste por isso num alongamento ou encurtamento do elemento, definido por u_{2x}^d e nas deformações devidas à flexão, definidas por α_1^d e α_2^d .

Em [Har73], citado por [Men06] considera-se que os esforços resultantes em cada um dos elementos de viga na configuração actual são: N , V , M_1 e M_2 , sendo N o esforço normal, V o esforço transversal e M_1 e M_2 os momentos flectores nas extremidades inicial e final do elemento. As equações que os definem são:

$$\begin{aligned} N &= EA_0 \varepsilon = \frac{EA_0}{L_0} u_{2x}^d & V &= \frac{M_1 + M_2}{L} = \frac{6EI}{LL_0} (\alpha_1^d + \alpha_2^d) \\ M_1 &= \frac{2EI}{L_0} (2\alpha_1^d + \alpha_2^d) & M_2 &= \frac{2EI}{L_0} (\alpha_1^d + 2\alpha_2^d) \end{aligned} \quad (3.20)$$

Nas equações anteriores é importante definir os diferentes valores envolvidos. E é o módulo da elasticidade longitudinal do material; A_0 a área de secção transversal; I o momento de inércia da secção transversal; L_0 e L os comprimentos dos elementos de viga nas configurações inicial e actual, respectivamente; e ε a deformação nominal, ou seja, $\varepsilon = (L - L_0)/L_0$. Na Fig. 3-5 é possível identificar cada um dos componentes que compõem as equações dos esforços resultantes.

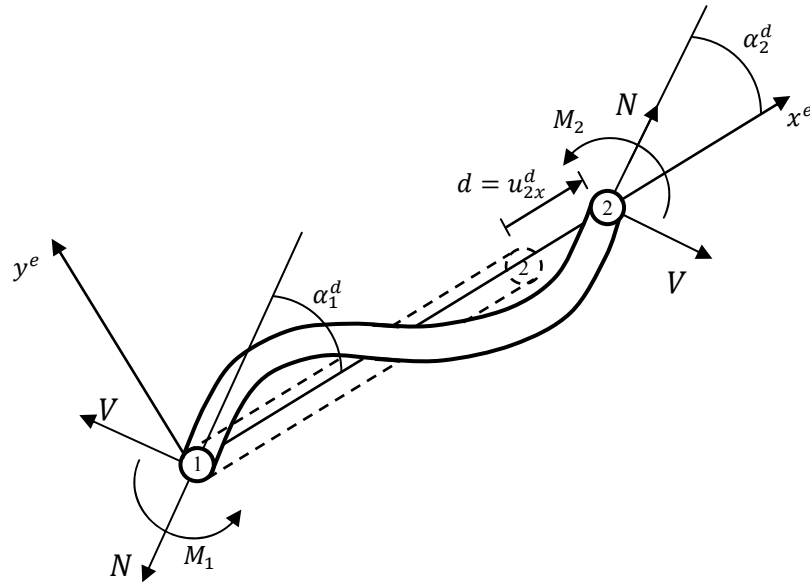


Fig. 3-5- Deformações e esforços resultantes.

Para esta metodologia, o vector de esforços internos (que se designa por fem no algoritmo) fica na seguinte forma:

$$\{F_{em}\} = [-N \quad V \quad M_1 \quad N \quad -V \quad M_2] \quad (3.21)$$

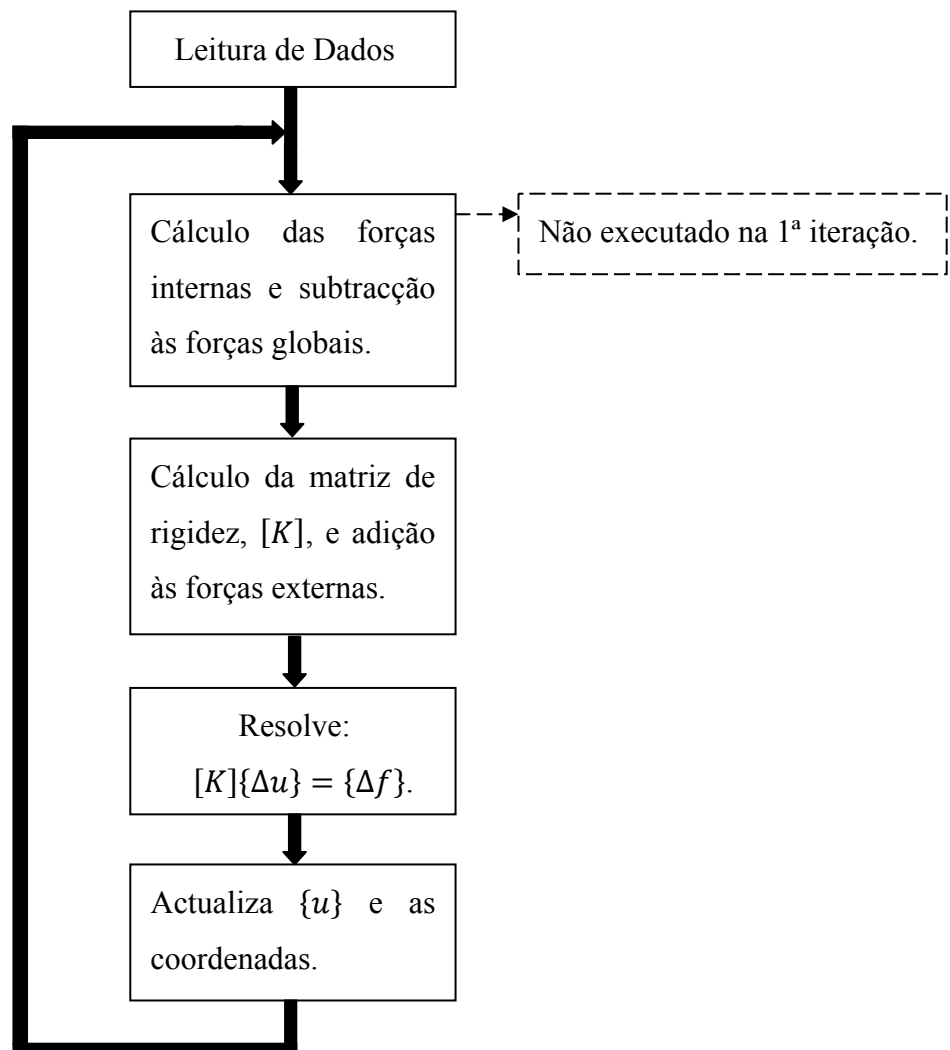
Em qualquer uma destas metodologias, é necessário transformar os esforços encontrados para o referencial global, sendo que a matriz $[T_n]$ faz a transformação do referencial global para o referencial local actual, é necessário multiplicar os esforços encontrados pela sua transposta, ou seja:

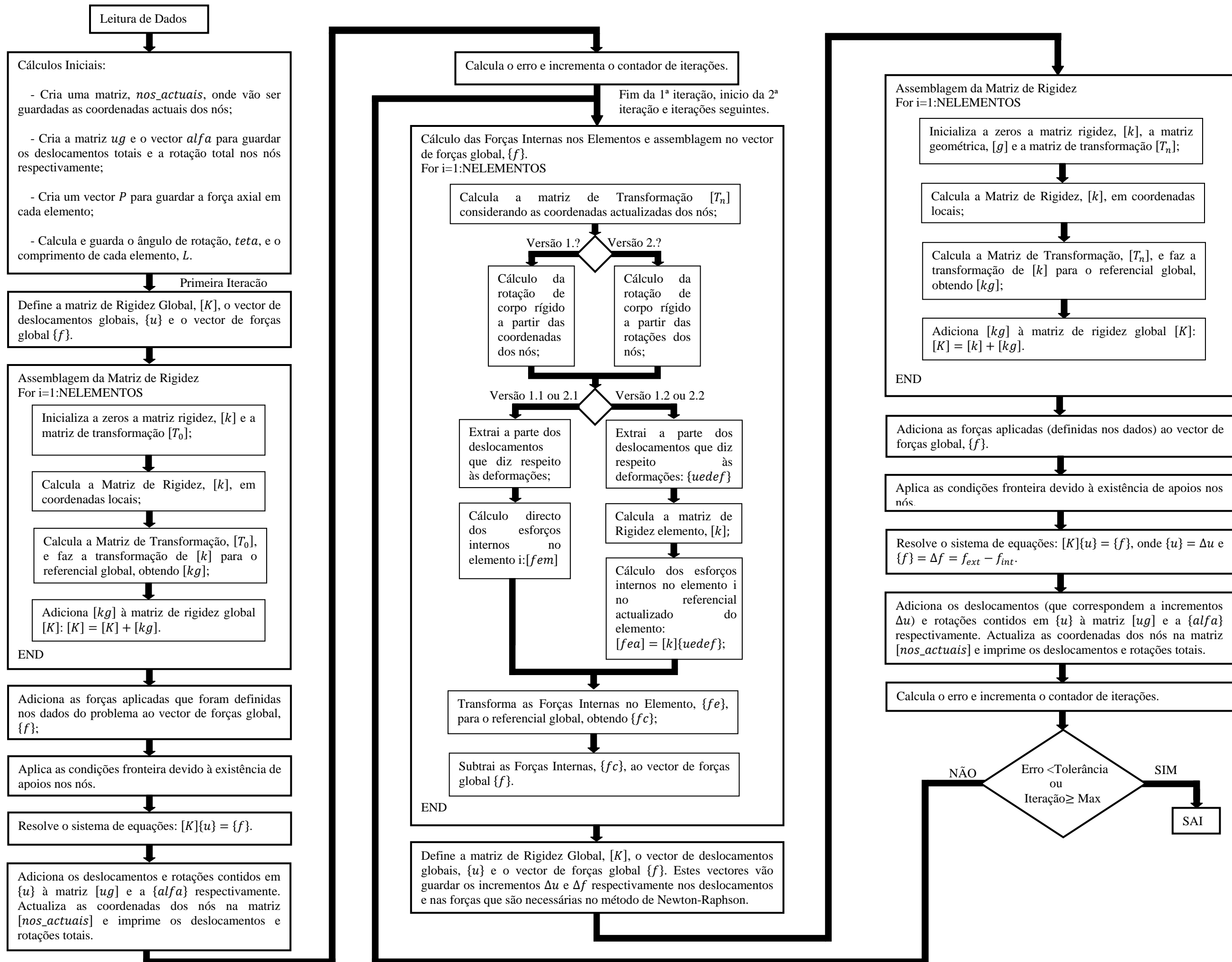
$$[T_n]^T \{F_{em}\} \text{ ou } [T_n]^T \{F_{ea}\} \quad (3.22)$$

Nestas condições já é possível adicionar as forças internas calculadas ao vector de forças globais.

3.3 Esquema do algoritmo desenvolvido

Para auxiliar a compreensão do algoritmo desenvolvido utilizam-se dois diagramas de blocos que ilustram todos os procedimentos que são efectuados no mesmo. O primeiro diagrama apresentado encontra-se muito simplificado para ser possível compreender a ideia geral do algoritmo. De seguida encontra-se um outro diagrama que explica pormenorizadamente o algoritmo “PEFNL-2D”.





4 EXEMPLOS DE ESTUDO

Os exemplos de estudo que se apresentam, são aqueles que se consideram importantes para a análise do correcto funcionamento do algoritmo. Esta análise baseia-se na comparação de resultados de diferentes fontes, sejam elas bibliográficas ou simulações no programa *Ansys*. Em relação ao algoritmo, são testadas todas as diferentes formulações, que foram explicadas anteriormente, estas apresentam-se no esquema da Fig. 4-1.

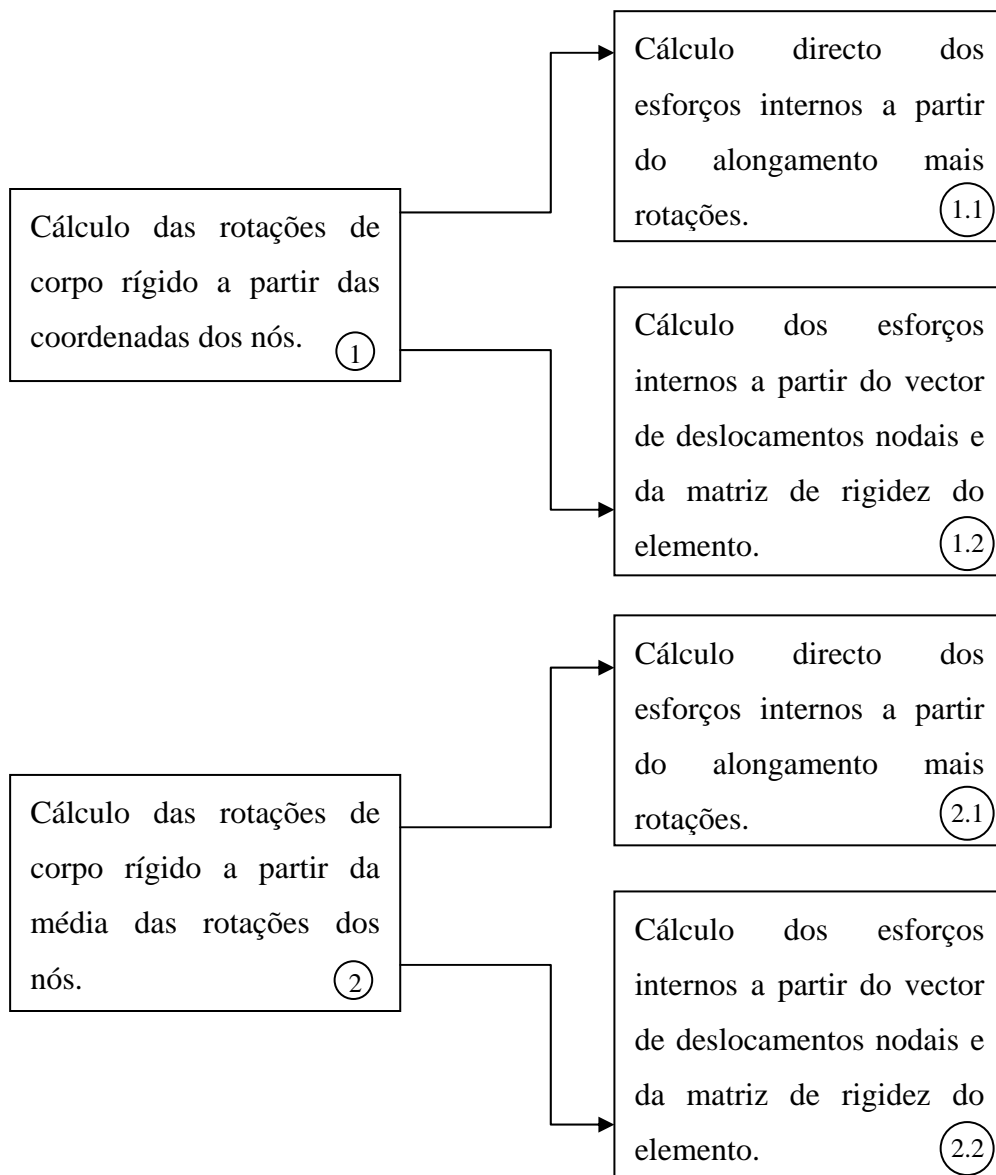


Fig. 4-1-Esquema ilustrativo das diferentes configurações do algoritmo "PEFNL-2D".

Como se pode observar no esquema testam-se quatro configurações diferentes que serão identificadas por 1.1; 1.2; 2.1 e 2.2. O número de incrementos em que a carga total é dividida, bem como o número de iterações que as versões do algoritmo executam até convergir para uma solução são objecto de comparação. Para todos os exemplos e versões começa-se por assumir que $N_{inc} = 1$, número que é aumentado quando não se obtém convergência com um único incremento.

Em anexo apresenta-se o algoritmo desenvolvido e também todos os ficheiros de *input*, os utilizados para o programa Ansys e os utilizados para o algoritmo.

4.1 Exemplo viga com momento

4.1.1 Considerações

Tendo em conta o objectivo de comparação das diferentes soluções encontradas para o algoritmo em causa, considerou-se um exemplo que surge em [Urt05], onde é analisada uma viga encastrada de secção quadrangular (altura h e área A) com um momento M na extremidade e por isso num estado de flexão pura como se representa na Fig. 4-2.

Na mesma figura é possível observar ainda que o modelo de viga considerado tem 21 nós, distribuídos de modo uniforme ao longo da viga e criando 20 elementos.

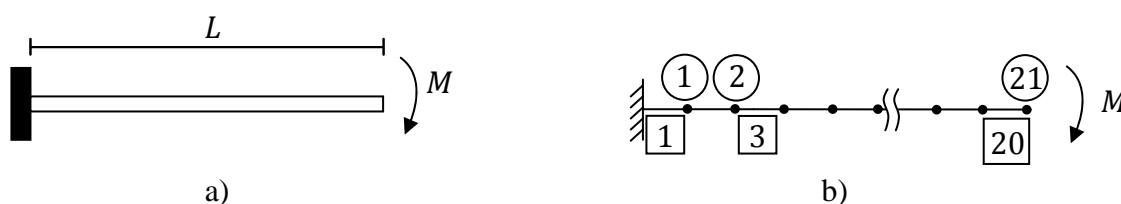


Fig. 4-2- Exemplo viga com momento a) Esquema do exemplo b) Modelo de elementos finitos

No artigo referido, utiliza-se para este exemplo o sistema de unidades imperiais. No entanto, na elaboração da presente dissertação é utilizado o sistema de unidades internacionais. Para poder comparar foi efectuada a conveniente conversão de unidades. Deste modo, na Tabela 4-1 podem observar-se as propriedades geométricas e materiais da viga.

O momento de inércia da secção de viga (quadrangular) em análise é dado pela expressão $I = h^4/12$, logo, $I = 2,16787E - 9 m^4$.

Tabela 4-1- Propriedades geométricas e materiais da viga.

Geométricas		Materiais	
L=100 in	L=2,54 m	E=30E+6 Psi	E=2,06844E+11 Pa
A=0,25 in ²	A=1,6129E-4 m ²		
h=0,5 in	h=0,0127 m		

Em relação ao momento aplicado, [Urt05] baseia-se na teoria de vigas de Euler-Bernoulli. A relação entre o momento M e o raio de curvatura da viga ρ é:

$$\frac{1}{\rho} = \frac{M}{EI} \quad (4.1)$$

Quando a viga flecte até completar um círculo exacto o raio de curvatura é:

$$\rho = \frac{L}{2\pi} \quad (4.2)$$

Substituindo este valor na equação (4.1), conclui-se que a viga consegue a configuração circular se se respeitar um parâmetro adimensional:

$$\frac{ML}{2\pi EI} = 1.0 \quad (4.3)$$

Desta forma, o momento a aplicar na estrutura é:

$$M = \frac{2\pi EI}{L} = 1109.23284 Nm \quad (4.4)$$

Os resultados obtidos pelas diferentes configurações do algoritmo são comparados com os do programa *Ansys*, na Tabela 4-2 estão os parâmetros utilizados para esta análise pelo programa *Ansys*.

Tabela 4-2-Parâmetros de análise do programa Ansys

Análise	Elemento de Viga
ANTYPE=0	BEAM3 (elemento de viga 2D)
NLGEOM=ON	
NEQIT=150	

4.1.2 Resultados Obtidos

Como explicado anteriormente, analisaram-se os resultados das quatro configurações diferentes para o programa “PEFNL-2D”. Apresentam-se também os resultados do programa *Ansys*, de [Urt05] e a respectiva comparação. Na Tabela 4-3 e Tabela 4-4 apresentam-se os resultados das quatro configurações do algoritmo para o problema apresentado.

Tabela 4-3-Resultados para as configurações 1.1 e 1.2 do algoritmo

Nó	Algoritmo desenvolvido versão 1.1			Algoritmo desenvolvido versão 1.2		
	u_{θ} [°]	u_x [m]	u_y [m]	u_{θ} [°]	u_x [m]	u_y [m]
3	-36,00011	-1,54059E-02	-7,75244E-02	-36,00009	-1,54059E-02	-7,75243E-02
5	-72,00014	-1,21947E-01	-2,80485E-01	-72,00012	-1,21947E-01	-2,80485E-01
7	-108,00018	-3,75948E-01	-5,31358E-01	-108,00015	-3,75948E-01	-5,31358E-01
9	-144,00021	-7,77408E-01	-7,34318E-01	-144,00019	-7,77407E-01	-7,34318E-01
11	-180,00024	-1,27000	-8,11841E-01	-180,00022	-1,27000	-8,11841E-01
13	-216,00028	-1,76260	-7,34316E-01	-216,00025	-1,76260	-7,34316E-01
15	-252,00031	-2,16405	-5,31355E-01	-252,00029	-2,16405	-5,31355E-01
17	-288,00034	-2,41805	-2,80482E-01	-288,00032	-2,41805	-2,80482E-01
19	-324,00038	-2,52459	-7,75223E-02	-324,00035	-2,52459	-7,75224E-02
21	-360,00041	-2,54000	-1,01462E-11	-360,00039	-2,54000	-2,80845E-10

Tabela 4-4- Resultados para as configurações 2.1 e 2.2 do algoritmo

Nó	Algoritmo desenvolvido versão 2.1			Algoritmo desenvolvido versão 2.2		
	u_{θ} [°]	u_x [m]	u_y [m]	u_{θ} [°]	u_x [m]	u_y [m]
3	-36,00024	-1,54038E-02	-7,75203E-02	-36,00027	-1,54061E-02	-7,75250E-02
5	-72,00028	-1,21874E-01	-2,80437E-01	-72,0003	-1,21948E-01	-2,80486E-01
7	-108,00031	-3,75505E-01	-5,31327E-01	-108,00033	-3,75949E-01	-5,31359E-01
9	-144,00034	-7,76174E-01	-7,34899E-01	-144,00037	-7,77410E-01	-7,34318E-01
11	-180,00038	-1,26816	-8,14441E-01	-180,0004	-1,27001	-8,11841E-01
13	-216,00041	-1,76199	-7,40541E-01	-216,00044	-1,76260	-7,34315E-01
15	-252,00044	-2,16855	-5,41249E-01	-252,00047	-2,16406	-5,31354E-01
17	-288,00048	-2,43217	-2,90412E-01	-288,0005	-2,41805	-2,80481E-01
19	-324,00051	-2,55009	-7,94253E-02	-324,00054	-2,52459	-7,75217E-02
21	-360,00054	-2,57196	1,63325E-02	-360,00057	-2,54000	-1,92581E-11

Em relação à simulação no programa *Ansys*, encontra-se na Tabela 4-5 os resultados obtidos e na Fig. 4-3 a ilustração da configuração original e deformada da viga.

Tabela 4-5-Resultados do programa *Ansys*

Nó	Ansys		
	u_{θ} [°]	u_x [m]	u_y [m]
3	-36,0001	-0,15406E-01	-0,77524E-01
5	-71,9979	-0,12195	-0,28048
7	-108,003	-0,37595	-0,53136
9	-144,001	-0,77741	-0,73432
11	-180,0042	-1,2700	-0,81184
13	-215,999	-1,7626	-0,73432
15	-251,998	-2,1641	-0,53136
17	-287,997	-2,4181	-0,28048
19	-324,002	-2,5246	-0,77524E-01
21	-360,001	-2,5400	0,26680E-09

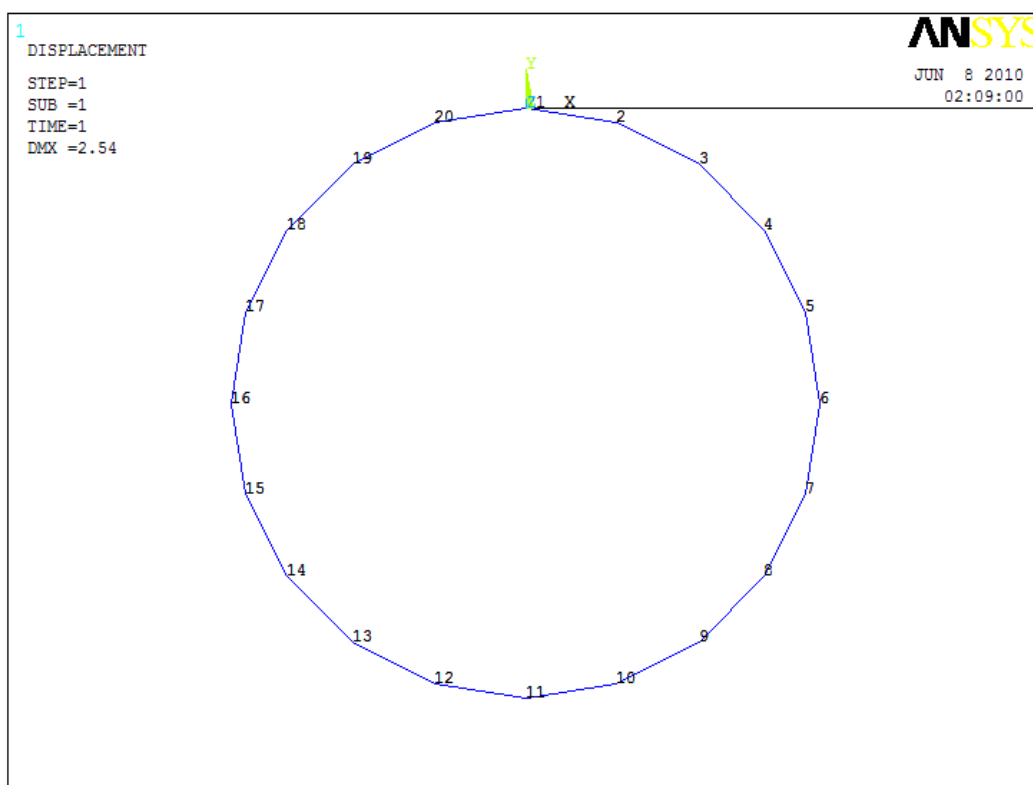


Fig. 4-3-Configuração original e deformada da viga (Ansys)

Para o momento aplicado, comparam-se então os valores encontrados com as diferentes versões do algoritmo “PEFNL-2D” e os calculados pelo programa “Ansys”. A comparação é feita para um dos nós e escolheu-se o nó 11, assinalado na Fig. 4-3, para o efeito. No nó 21, essa comparação não deve ser feita, uma vez que os deslocamentos u_y são muito próximos de zero, o que inviabiliza o cálculo do erro relativo.

Tabela 4-6-Comparação de resultados Ansys - algoritmo (nó 11)

Desl.	Ansys	Algoritmo				Erro relativo [%]			
		V. 1.1	V. 1.2	V. 2.1	V. 2.2	V. 1.1	V. 1.2	V. 2.1	V. 2.2
u_θ [°]	-180,00042	-180,00024	-180,00022	-180,00038	-180,0004	1,00E-04	1,11E-04	2,22E-05	1,11E-05
u_x [m]	-1,27000	-1,27000	-1,27000	-1,26816	-1,27001	0,00	0,00	1,45E-01	7,87E-04
u_y [m]	-0,81184	-8,11841E-01	-8,11841E-01	-8,14441E-01	-8,11841E-01	1,23E-04	1,23E-04	3,20E-01	1,23E-04

Desde já se verifica pelos erros relativos calculados, que as versões 1.1, 1.2 e 2.2 do algoritmo “PEFNL-2D” retornam resultados praticamente iguais aos calculados pelo

programa *Ansys*, a versão 2.1 é a que apresenta um desvio maior, no que diz respeito aos deslocamentos u_x e u_y . Verifica-se então a conformidade de resultados entre o algoritmo desenvolvido e o programa *Ansys*, para as versões 1.1, 1.2 e 2.2.

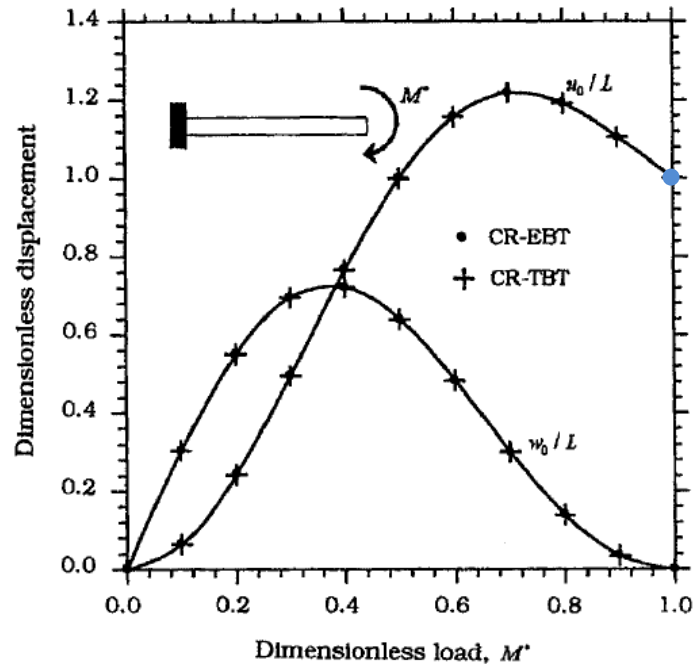


Fig. 4-4 - Deslocamentos para diferentes valores de $M^* = ML/2\pi EI$ (adaptado [Urt05])

É possível ainda comparar os resultados obtidos com [Urt05]. Na Fig. 4-4 é ilustrado um gráfico que relaciona o valor de $M^* = ML/2\pi EI$ com u_0/L e com w_0/L , que são respectivamente, na nomenclatura utilizada na dissertação u_x/L e u_y/L . O ponto azul assinalado na Fig. 4-4 refere-se ao valor do deslocamento nó 21 para $M^* = 1$. Do gráfico retira-se o deslocamento u_x que se apresenta na Tabela 4-7.

Tabela 4-7-Comparação de resultados [Urt05] - algoritmo (nó 21)

Desl. Nó 21	[Urt05] com L=2,54 m	Algoritmo				Erro relativo [%]			
		V. 1.1	V. 1.2	V. 2.1	V. 2.2	V. 1.1	V. 1.2	V. 2.1	V. 2.2
u_x [m]	-2,54000	-2,54000	-2,54000	-2,57196	-2,54000	0	0	1,26	0

Mais uma vez comparando os resultados, verifica-se para todos os casos a concordância de resultados excepto para a versão 2.1 que apresenta um desvio de 1,26%.

É possível ainda comparar (Tabela 4-8) dois factores importantes em relação às diferentes versões do algoritmo, o número de iterações e o número de incrementos em que se dividem as cargas a aplicar. Todas as versões foram testadas com $N_{inc} = 1$, procedendo-se ao aumento dos mesmos apenas se necessário (não convergência para o resultado).

Tabela 4-8 - N° de iterações e n° de incrementos nas diferentes configurações do programa “PEFNL-2D”
(exemplo viga com momento)

	Exemplo viga com momento			
	V1.1	V1.2	V2.1	V2.2
N° de iterações	24	12	90	87
N° de incrementos	2	1	30	3

4.2 Exemplo viga com força axial

4.2.1 Considerações

Para continuar a validação do algoritmo desenvolvido tem-se em conta um exemplo de demonstração incluído no manual de verificação do *Ansys*, denominado de VM136 [Ans07] que consiste numa viga com uma secção transversal de altura h e área A encastrada na base. Sobre a viga está aplicada uma carga vertical F tal como se encontra representado na Fig. 4-5.

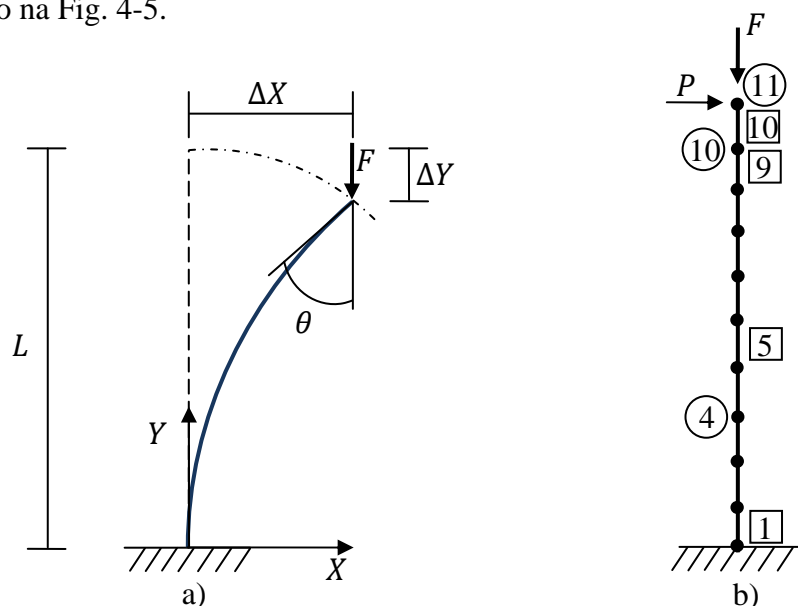


Fig. 4-5 - Exemplo viga com carga axial a) Comportamento esperado b) Modelo de elementos finitos

Em [Tim61] encontra-se também este exemplo, que será mais um ponto de comparação com o algoritmo desenvolvido. Em ambos os casos, ([Tim61] e manual de verificação do *Ansys*) é utilizado o sistema de unidades imperial, pelo que é necessário proceder às devidas conversões para o sistema de unidades internacional adoptado nesta dissertação. Na Tabela 4-9 especificam-se as propriedades geométricas e materiais da viga.

O momento de inércia da secção da viga (quadrangular) em análise é dado pela expressão $I = h^4/12$, ou seja, $I = 2,16787E - 9 \text{ m}^4$.

Tabela 4-9-Propriedades geométricas e materiais da viga.

Geométricas		Materiais	
L=100 in	L=2,54 m	E=30E+6 Psi	E=2,06844E+11 Pa
A=0,25 in ²	A=1,6129E-4 m ²		
h=0,5 in	h=0,0127 m		

A carga crítica da viga, F_{cr} , de acordo com [Tim61] pode ser calculada a partir da equação:

$$F_{cr} = \frac{\pi^2 EI}{4L^2} \Leftrightarrow F_{cr} = 171,49398 \text{ N}$$

O modelo de viga em estudo é decomposto em dez elementos finitos, ligados entre si por onze nós. O estudo efectuado considera o incremento do valor de F numa gama de valores superior à carga crítica calculada anteriormente, para estudar o comportamento de pós-encurvadura da viga. A força F toma assim os valores descritos em seguida, com o comportamento esperado da viga a ser ilustrado na Fig. 4-6:

$$F_1 = F_{cr} \times 1,015 = 174,06639 \text{ N}$$

$$F_2 = F_{cr} \times 1,063 = 182,29810 \text{ N}$$

$$F_3 = F_{cr} \times 1,152 = 197,56107 \text{ N}$$

$$F_4 = F_{cr} \times 1,293 = 221,74172 \text{ N}$$

$$F_5 = F_{cr} \times 1,518 = 260,32786 \text{ N}$$

$$F_6 = F_{cr} \times 1,884 = 323,09466 \text{ N}$$

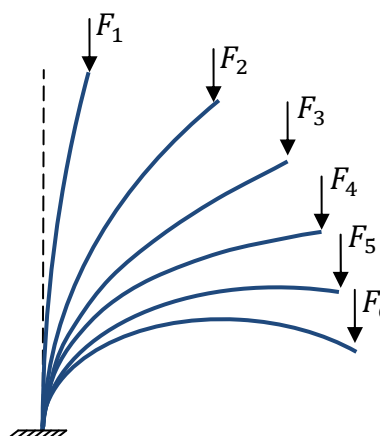


Fig. 4-6 – Aplicação dos diferentes valores de F e comportamento esperado da viga.

Considerou-se ainda uma pequena força aplicada na extremidade da viga (nó 11), que se encontra ilustrada na Fig. 4-5 b) com direcção horizontal, $P = 2,23 N$. Esta é aplicada no início do processo de carregamento com o intuito de provocar uma pequena perturbação de modo a que a viga se comporte de acordo com o desejado e posteriormente é retirada.

Os resultados obtidos pelo algoritmo são comparados com os do programa *Ansys*, na Tabela 4-10 estão os parâmetros utilizados para esta análise.

Tabela 4-10 - Parâmetros de análise do programa *Ansys*

Análise	Elemento de Viga
ANTYPE=0	BEAM3 (elemento de viga 2D)
NLGEOM=ON	
NEQIT=150	

4.2.2 Resultados obtidos

Os resultados encontrados referem-se às quatro versões desenvolvidas do programa “*PEFNL-2D*”. São dados a conhecer os resultados da simulação feita no *Ansys* e procede-se à comparação com os obtidos no algoritmo, posteriormente procede-se da mesma forma com os valores retirados de [Tim61]. Embora se tenham comparado os valores dos deslocamentos para todos os diferentes níveis de força aplicados apenas se apresentam os que correspondem à carga máxima, $F_6 = 323,09466 N$. Deste modo, na Tabela 4-11 e na Tabela 4-12 apresentam-se os resultados obtidos para três das versões do algoritmo pois a versão 2.1 não converge em nenhum dos casos.

Tabela 4-11 - Resultados para a configuração 1.1 e 1.2 do algoritmo

Nó	Algoritmo desenvolvido versão 1.1			Algoritmo desenvolvido versão 1.2		
	u_θ [°]	u_x [m]	u_y [m]	u_θ [°]	u_x [m]	u_y [m]
1	-0,00011	-7,07884E-11	-7,67304E-11	-0,00012	-7,27353E-11	-7,98123E-11
2	-21,16195	4,68122E-02	-4,35348E-03	-21,16197	4,68123E-02	-4,35349E-03
3	-41,40149	1,79133E-01	-4,15442E-02	-41,4015	1,79133E-01	-4,15443E-02
4	-59,9284	3,76068E-01	-1,35134E-01	-59,9284	3,76068E-01	-1,35134E-01
5	-76,19494	6,12013E-01	-2,95084E-01	-76,19492	6,12014E-01	-2,95084E-01
6	-89,90638	8,64263E-01	-5,19321E-01	-89,90633	8,64264E-01	-5,19321E-01
7	-100,97023	1,11703	-7,98369E-01	-100,97014	1,11703	-7,98369E-01
8	-109,42166	1,36189	-1,11987	-109,42156	1,36190	-1,11986
9	-115,35378	1,59640	-1,47145	-115,35365	1,59640	-1,47145
10	-118,86532	1,82210	-1,84198	-118,86518	1,82210	-1,84197
11	-120,02733	2,04286	-2,22159	-120,02718	2,04286	-2,22159

Tabela 4-12 - Resultados para a configuração 2.1 e 2.2 do algoritmo

Nó	Algoritmo desenvolvido versão 2.1			Algoritmo desenvolvido versão 2.2		
	u_θ [°]	u_x [m]	u_y [m]	u_θ [°]	u_x [m]	u_y [m]
1	Não converge para nenhuma solução			-0,00018	-9,70601E-11	-1,11831E-10
2				-21,20747	4,69251E-02	-4,37459E-03
3				-41,45704	1,79448E-01	-4,16881E-02
4				-59,96493	3,76519E-01	-1,35442E-01
5				-76,19315	6,12496E-01	-2,95468E-01
6				-89,85721	8,64735E-01	-5,19595E-01
7				-100,87323	1,11753	-7,98319E-01
8				-109,28262	1,36254	-1,11931
9				-115,18248	1,59732	-1,47025
10				-118,67389	1,82338	-1,84006
11				-119,82908	2,04458	-2,21892

Quanto à simulação no programa *Ansys*, utilizou-se de [Ans07] o ficheiro VM136.dat com as devidas alterações para unidades do sistema internacional. Na Tabela 4-13 encontram-se os resultados obtidos.

Tabela 4-13 - Resultados da simulação no Ansys

Nó	Ansys		
	u_θ [°]	u_x [m]	u_y [m]
1	0	0	0
2	-21,18120	4,68601E-02	-4,36238E-03
3	-41,42580	1,79267E-01	-4,16053E-02
4	-59,94800	3,76267E-01	-1,35272E-01
5	-76,20450	6,12238E-01	-2,95282E-01
6	-89,90400	8,64491E-01	-5,19535E-01
7	-100,95500	1,11726	-7,98544E-01
8	-109,39500	1,36215	-1,11995
9	-115,31900	1,59672	-1,47141
10	-118,82500	1,82249	-1,84179
11	-119,98500	2,04335	-2,22125

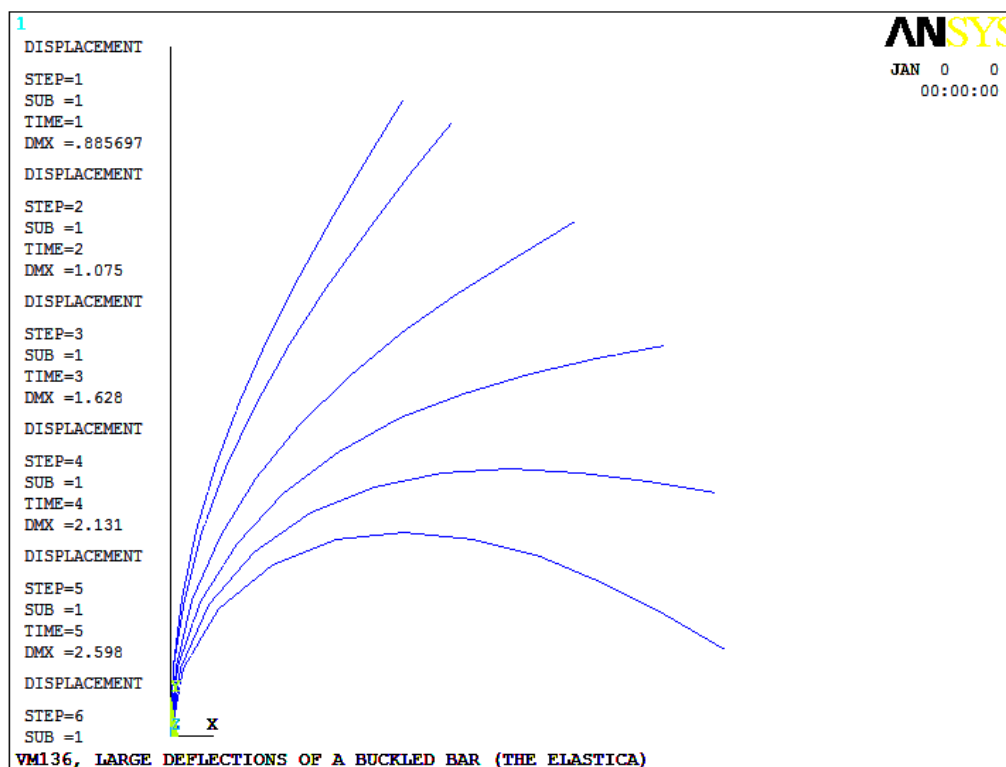


Fig. 4-7 - Configuração inicial e final da viga para cada um dos incrementos de força efectuados.

Na Fig. 4-7 encontra-se ilustrada a configuração inicial e as diversas configurações deformadas da viga. É conveniente lembrar que a carga aplicada vai sendo incrementada até chegar ao valor total F .

A comparação de resultados do programa *Ansys* com o algoritmo é feita para o nó 11 e apresenta-se na Tabela 4-14.

Tabela 4-14 - Comparação de resultados *Ansys* - algoritmo

Desl.	Ansys	Algoritmo			Erro relativo [%]		
		V. 1.1	V. 1.2	V. 2.2	V. 1.1	V. 1.2	V. 2.2
u_θ [°]	-119,98500	-120,02733	-120,02718	-119,82908	3,53E-02	3,52E-02	1,30E-01
u_x [m]	2,04335	2,04286	2,04286	2,04458	2,40E-02	2,40E-02	6,02E-02
u_y [m]	-2,22125	-2,22159	-2,22159	-2,21892	1,53E-02	1,53E-02	1,05E-01

Pelos erros relativos calculados, verifica-se que os resultados obtidos com as versões 1.1, 1.2 e 2.2 do algoritmo apresentam uma boa precisão.

É possível ainda comparar os resultados obtidos com os presentes em [Tim61]. Na Fig. 4-8 está ilustrado o comportamento da viga (presente na referência enunciada) para a situação em estudo.

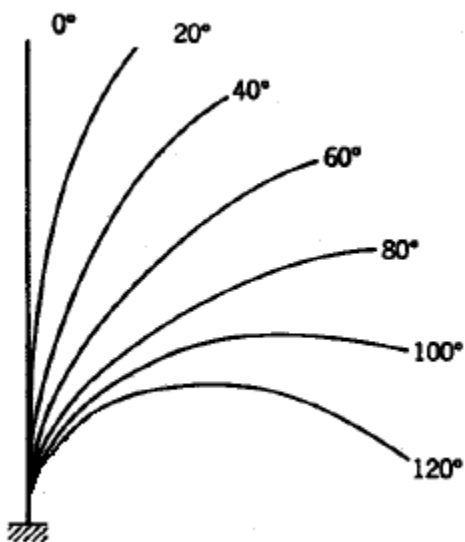


Fig. 4-8 - Configuração inicial e deformada da viga (adaptado [Tim61])

A comparação destes resultados (referentes à situação representada ($\theta = 120^\circ$)) apresenta-se na Tabela 4-15.

Tabela 4-15 - Comparação de resultados [Tim61] - algoritmo

Desl.	[Tim61]	Algoritmo			Erro relativo [%]		
		V. 1.1	V. 1.2	V. 2.2	V. 1.1	V. 1.2	V. 2.2
u_θ [°]	-120,00000	-120,02733	-120,02718	-119,82908	2,28E-02	2,27E-02	1,42E-01
u_x [m]	2,03962	2,04286	2,04286	2,04458	1,59E-01	1,59E-01	2,43E-01
u_y [m]	-2,22758	-2,22159	-2,22159	-2,21892	2,69E-01	2,69E-01	3,89E-01

Pode então verificar-se, mais uma vez, que os valores encontrados pelo algoritmo para os deslocamentos no nó 11 são muito aproximados dos que são apresentados na bibliografia.

Na Tabela 4-16 é feita a comparação entre dois factores importantes em relação às diferentes versões do algoritmo, o número de iterações e o número de incrementos em que se dividem as cargas a aplicar. Todas as versões contam com o mínimo possível de incrementos e apenas se necessário (não convergência para o resultado) se aumentam os mesmos.

Tabela 4-16 - N° de iterações e n° de incrementos nas diferentes configurações do programa "PEFNL-2D" (viga com força axial)

	Exemplo viga com força axial			
	V1.1	V1.2	V2.1	V2.2
N° de iterações	51	53	-	72
N° de incrementos	8	8	-	8

4.3 Exemplo viga com força transversal

4.3.1 Considerações

O exemplo que se segue, encontra-se em [Urt05], neste estuda-se o comportamento de uma viga encastrada com uma secção transversal de altura h e área A . Sobre a extremidade livre da viga está aplicada uma carga vertical F , tal como se encontra representado na Fig. 4-9.

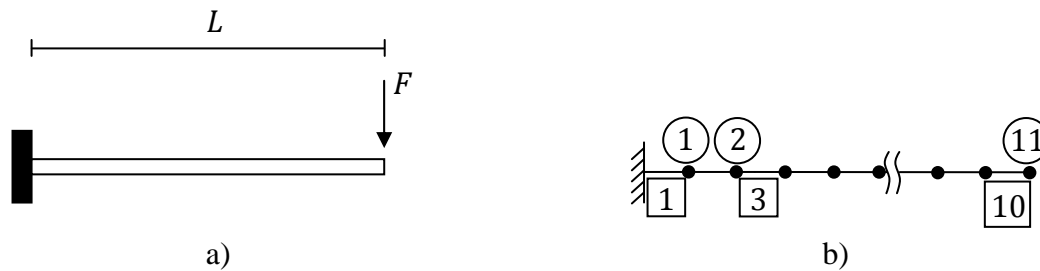


Fig. 4-9 - Exemplo viga com força transversal a) Esquema do exemplo b) Modelo de elementos finitos

Na referência enunciada, os dados encontram-se em unidades imperiais, como tal procede-se à conversão de unidades para o sistema internacional, de modo a que seja possível fazer a comparação de resultados posteriormente.

A carga máxima aplicada na estrutura é [Urt05]:

$$F = \frac{10EI}{L^2} \quad (4.5)$$

Para este exemplo vão ser estudados os resultados para dez valores diferentes da carga adimensional FL^2/EI , começando por $FL^2/EI = 1$ até $FL^2/EI = 10$.

Na Tabela 4-17 especificam-se as propriedades geométricas e materiais da viga.

Tabela 4-17-Propriedades geométricas e materiais da viga

Geométricas		Materiais	
L=100 in	L=2,54 m	E=30E+6 Psi	E=2,06844E+11 Pa
A=1 in ²	A=6,4516E-4 m ²		
h=1 in	h=2,54E-2 m		

O momento de inércia da secção da viga (quadrangular) em análise é dado pela expressão $I = h^4/12$, ou seja, $I = 3,46860E - 8 m^4$.

Desta forma, a força inicial aplicada na viga é:

$$F = \frac{EI}{L^2} = 1112,06383 N$$

Esta força é incrementada até chegar a um valor máximo:

$$F_{MAX} = \frac{10EI}{L^2} = 11120,63830 N$$

Para comparação de resultados, utilizou-se uma vez mais o programa *Ansys*, na Tabela 4-18 encontram-se os parâmetros utilizados para esta análise. Neste caso utilizando o elemento de viga 2D não foi obtida convergência para os valores acima de $\frac{FL^2}{EI} = 3$. Recorreu-se por isso ao elemento de viga 3D (BEAM4), com o qual se obtiveram os resultados da Tabela 4-21.

Tabela 4-18-Parâmetros de análise (programa Ansys)

Análise	Elemento de Viga
ANTYPE=0	BEAM4 (elemento de viga 3D)
NLGEOM=ON	
NEQIT=150	

Relativamente ao elemento de viga utilizado, foi ainda necessário activar a opção “consistent stiffness matrix”. Resta ainda referir que o modelo de elementos finitos possui onze nós, ligados entre si por dez elementos.

4.3.2 Resultados obtidos

Os resultados obtidos para as diferentes configurações do algoritmo “PEFNL-2D” são apresentados na Tabela 4-19 e na Tabela 4-20. De seguida analisam-se os resultados da simulação feita no programa *Ansys* e os apresentados em [Urt05], procedendo à comparação com os encontrados. Estes resultados referem-se aos deslocamentos no nó 11, para diferentes valores de $\frac{FL^2}{EI}$.

Tabela 4-19- Resultados obtidos para a configuração 1.1 e 1.2 do algoritmo

$\frac{FL^2}{EI}$	Algoritmo desenvolvido versão 1.1			Algoritmo desenvolvido versão 1.2		
	u_θ [°]	u_x [m]	u_y [m]	u_θ [°]	u_x [m]	u_y [m]
1	-26,43932	-1,43096E-01	-7,66572E-01	-26,43925	-1,43096E-01	-7,66570E-01
2	-44,81524	-4,07575E-01	-1,25418	-44,81495	-4,07570E-01	-1,25417
3	-56,53782	-6,45756E-01	-1,53362	-56,53731	-6,45745E-01	-1,53361
4	-64,30039	-8,35098E-01	-1,70350	-40,91908	-8,35082E-01	-1,70348
5	-69,70447	-9,84237E-01	-1,81515	-69,70369	-9,84216E-01	-1,81513
6	-73,62715	-1,10359	-1,89359	-73,62629	-1,10356	-1,89357
7	-76,56953	-1,20103	-1,95172	-76,56862	-1,20100	-1,95170
8	-78,83448	-1,28210	-1,99665	-78,83354	-1,28207	-1,99663
9	-80,61463	-1,35071	-2,03257	-80,61362	-1,35067	-2,03256
10	-82,03754	-1,40961E+00	-2,06208E+00	-82,03655	-1,40957	-2,06207

Tabela 4-20 - Resultados obtidos para a configuração 2.1 e 2.2 do algoritmo

$\frac{FL^2}{EI}$	Algoritmo desenvolvido versão 2.1			Algoritmo desenvolvido versão 2.2		
	u_θ [°]	u_x [m]	u_y [m]	u_θ [°]	u_x [m]	u_y [m]
1	Não converge para nenhuma solução			-26,43925	-1,43095E-01	-7,66570E-01
2				-44,81496	-4,07568E-01	-1,25417
3				-56,5374	-6,45743E-01	-1,53362
4				-64,29973	-8,35077E-01	-1,70349
5				-69,7037	-9,84216E-01	-1,81514
6				-73,62629	-1,10355	-1,89358
7				-76,56863	-1,20099	-1,95171
8				-78,83355	-1,28206	-1,99664
9				-80,61362	-1,35066	-2,03257
10				-82,03656	-1,40955	-2,06208

Na Tabela 4-21 encontram-se os resultados para a simulação efectuada no programa Ansys.

Tabela 4-21 - Resultados da simulação no programa Ansys

$\frac{FL^2}{EI}$	Ansys		
	u_θ [°]	u_x [m]	u_y [m]
1	-26,44086	-1,43110E-01	-0,76662
2	-44,81905	-4,07660E-01	-1,25430
3	-56,54406	-6,45940E-01	-1,53380
4	-64,30878	-8,35360E-01	-1,70380
5	-69,71177	-9,84580E-01	-1,81540
6	-73,63654	-1,10400	-1,89390
7	-4396,59160	-1,20160	-1,95210
8	-78,84472	-1,28260	-1,99700
9	-80,62089	-1,35130	-2,03300
10	-441,98537	-1,40880	-2,06210

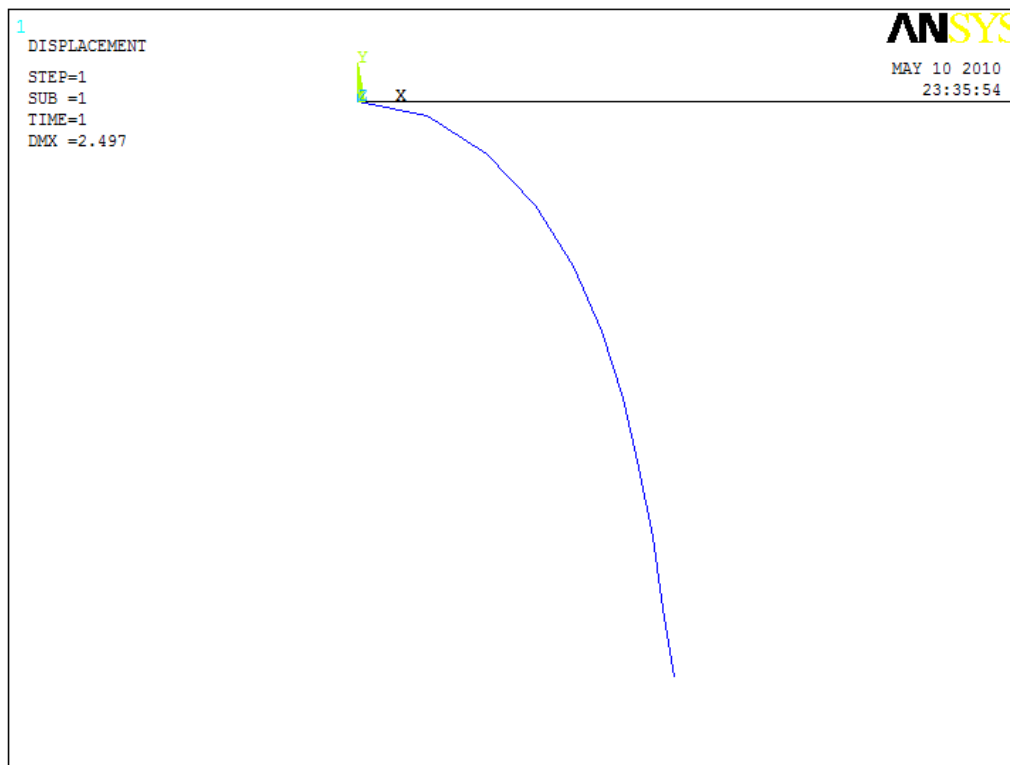


Fig. 4-10 - Configuração inicial e deformada da viga (para $FL^2/EI=10$).

Como é possível constatar na Tabela 4-21, para $\frac{FL^2}{EI} = 7$ e $\frac{FL^2}{EI} = 10$ o programa *Ansys* obtém valores inesperados para a rotação u_θ . Na Fig. 4-10 é possível observar a configuração inicial e a deformada da viga retirada da simulação do programa *Ansys*.

Para efectuar a comparação de resultados entre as diversas versões do algoritmo “PEFNL-2D” e o programa *Ansys* escolheu-se a situação em que $\frac{FL^2}{EI} = 9$, uma vez que, tal como foi referido anteriormente para a situação de carga máxima não se obtiveram resultados satisfatórios para a rotação. Os deslocamentos apresentados referem-se ao nó 11.

Tabela 4-22 - Comparação de resultados *Ansys* - algoritmo

Desl.	Ansys	Algoritmo			Erro relativo [%]		
		V. 1.1	V. 1.2	V. 2.2	V. 1.1	V. 1.2	V. 2.2
u_θ [°]	-80,62089	-80,61463	-80,61362	-80,61362	7,77E-03	9,02E-03	9,02E-03
u_x [m]	-1,35130	-1,35071	-1,35067	-1,35066	4,37E-02	4,66E-02	4,74E-02
u_y [m]	-2,03300	-2,03257	-2,03256	-2,03257	2,12E-02	2,16E-02	2,12E-02

Interessa agora comparar os resultados obtidos com o algoritmo desenvolvido com os que se encontram em [Urt05] (que confirmam os previamente encontrados por Bisshop e Drucker [Bis45]).

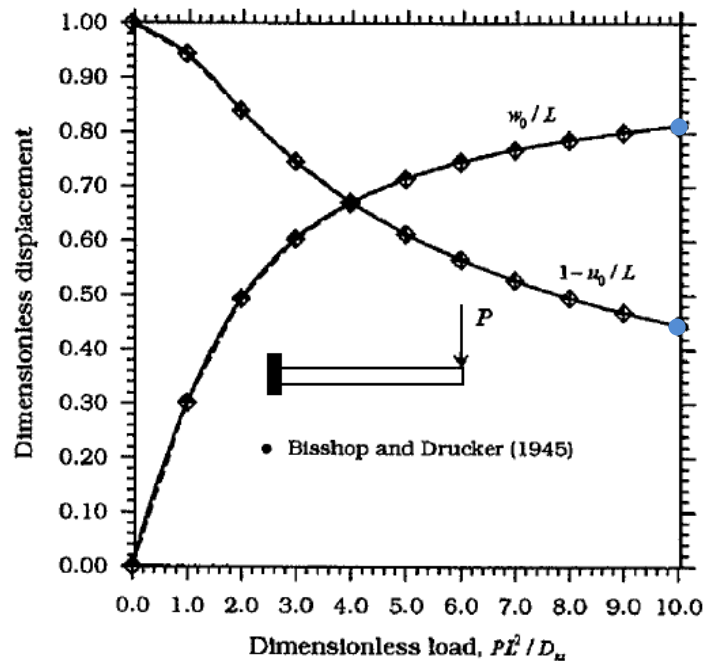


Fig. 4-11 - Deslocamentos adimensionais em função de diferentes valores de F

Na Fig. 4-11 encontram-se os deslocamentos em função de diferentes valores de PL^2/EI . Nesta figura referem-se u_0/L e w_0/L , que representam respectivamente, na nomenclatura utilizada na presente dissertação u_x/L e u_y/L . Os pontos assinalados a azul correspondem aos valores encontrados para os deslocamentos com $FL^2/EI = PL^2/EI = 10$, no nó 11. Estes encontram-se na Tabela 4-23 em unidades do sistema internacional para comparação com os resultados obtidos do algoritmo.

Tabela 4-23- Comparação de resultados [Urt05] - algoritmo

Nó 11	[Urt05] $PL^2/EI = 10$	Algoritmo			Erro relativo [%]		
		V. 1.1	V. 1.2	V. 2.2	V. 1.1	V. 1.2	V. 2.2
u_x [m]	-1,40970	-1,40961	-1,40957	-1,40955	6,38E-03	9,22E-03	1,06E-02
u_y [m]	-2,05740	-2,06208	-2,06207	-2,06208	2,27E-01	2,27E-01	2,27E-01

Como podemos constatar pelos erros relativos calculados, o algoritmo apresenta para as versões 1.1, 1.2 e 2.2 um erro relativo muito baixo tanto na comparação com o *Ansys* como na comparação com [Urt05]. Temos então para este exemplo uma boa precisão nos resultados obtidos.

Na Tabela 4-24 são apresentados os números de iterações e de incrementos para cada uma das versões do algoritmo. Neste exemplo consideraram-se sempre 10 incrementos até chegar à carga total, para que fosse possível a comparação com [Urt05]. Deste modo, apenas o número de iterações é válido para comparação entre as diferentes configurações do programa.

Tabela 4-24 - N° de iterações e n° de incrementos nas diferentes configurações do programa "PEFNL-2D" (viga com força transversal)

	Exemplo viga com força transversal			
	V1.1	V1.2	V2.1	V2.2
N° de iterações	53	53	-	59
N° de incrementos	10	10	-	10

4.4 Exemplo pórtico

4.4.1 Considerações

O último exemplo efectuado tem o interesse de testar o algoritmo desenvolvido numa estrutura e não apenas numa viga como até aqui. Estuda-se então um pórtico constituído por trinta elementos e trinta e um nós. A Fig. 4-12 ilustra a situação em estudo, sobre a estrutura são aplicada duas forças verticais F e uma força horizontal P .

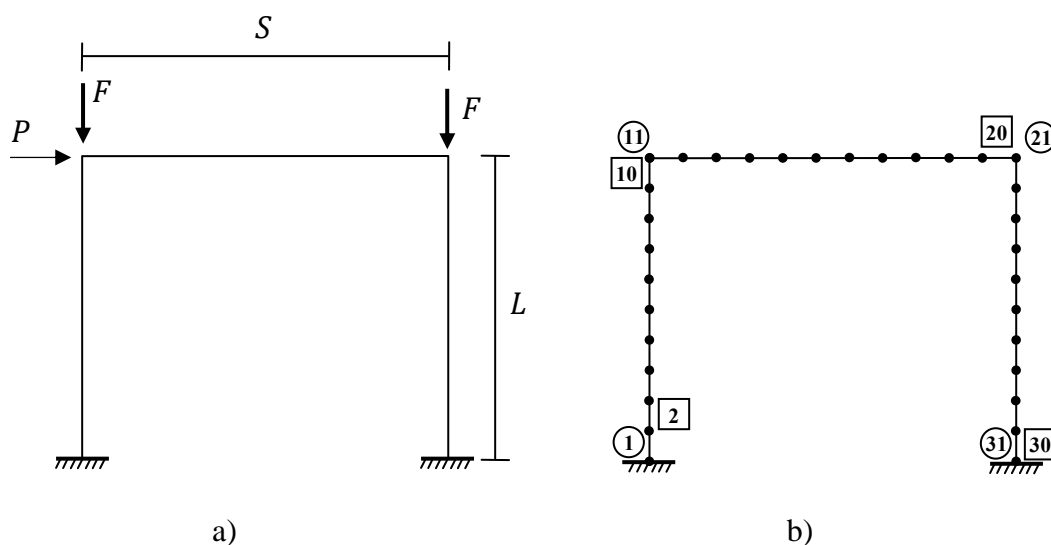


Fig. 4-12 - Exemplo Pórtico a) Esquema do exemplo b) Modelo de elementos finitos

Como podemos observar na figura as forças aplicadas encontram-se nos nós 11 e 21. A força P tem o intuito de provocar uma perturbação de modo a que a viga se comporte como o desejado, os respectivos valores para as cargas aplicadas são: $F = 37500KN$ e $P = 2000KN$, as dimensões L e S são respectivamente 3 e 4 m.

O Perfil de viga utilizado para a construção do pórtico é um HEB300 ilustrado na Fig. 4-13. As suas características apresentam-se na Tabela 4-25.

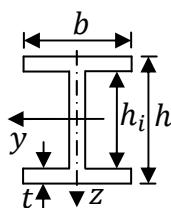


Fig. 4-13 - Perfil de viga HEB300

Tabela 4-25- Características do perfil HEB300 e características materiais da viga

Perfil HEB 300							Material
h [m]	b [m]	h_i [m]	t [m]	A [m ²]	I_y [m ⁴]	I_z [m ⁴]	$E = 210E9 Pa$
0,3	0,3	0,262	0,019	149.1e-4	2.517e-4	0.8563e-4	

A comparação dos resultados obtidos com o algoritmo será efectuada com os do programa *Ansys*, como tal na Tabela 4-26 encontram-se os parâmetros utilizados na mesma.

Tabela 4-26 - Parâmetros da análise efectuada no programa *Ansys*

Análise	Elemento de Viga
ANTYPE=0	BEAM3/BEAM4 (elemento de viga 2D/3D)
NLGEOM=ON	
NEQIT=150	

4.4.2 Resultados obtidos

Os resultados obtidos com as diferentes versões do algoritmo “*PEFNL-2D*” apresentam-se na Tabela 4-27 e na Tabela 4-28. A versão 2.1 do algoritmo não consegue convergir para nenhuma solução, algo que já se tinha verificado para os dois exemplos anteriores.

Em relação à simulação executada no programa *Ansys*, testaram-se os dois elementos de viga, BEAM3 e BEAM4. Obtiveram-se os resultados apresentados na Tabela 4-29. É importante constatar que os diferentes elementos de viga retornam resultados distintos. A configuração inicial e deformada da estrutura, retirada da simulação efectuada, encontra-se na Fig. 4-14.

Tabela 4-27 - Resultados do algoritmo "PEFNL-2D" versão 1.1 e 1.2

Nó	Algoritmo desenvolvido versão 1.1			Algoritmo desenvolvido versão 1.2		
	u_0 [°]	u_x [m]	u_y [m]	u_0 [°]	u_x [m]	u_y [m]
1	-0,00003	2,64761E-09	-7,18578E-08	-0,00001	7,57437E-10	-2,10343E-08
3	-9,48279	5,08411E-02	-8,58220E-03	-10,40899	5,58342E-02	-9,00196E-03
5	-16,03187	1,85163E-01	-2,98972E-02	-17,57045	2,02915E-01	-3,32950E-02
7	-18,35702	3,64699E-01	-6,32128E-02	-20,10269	3,99034E-01	-7,19211E-02
9	-16,01022	5,44125E-01	-9,64949E-02	-17,53077	5,94954E-01	-1,10480E-01
11	-9,44369	6,78134E-01	-1,17740E-01	-10,33714	7,41464E-01	-1,34632E-01
13	-0,55647	6,74971E-01	-1,82233E-01	-0,60508	7,37742E-01	-2,05145E-01
15	3,87738	6,74555E-01	-1,53896E-01	4,23848	7,37306E-01	-1,74137E-01
17	3,89655	6,72649E-01	-9,45310E-02	4,25936	7,35092E-01	-1,09256E-01
19	-0,49872	6,72217E-01	-6,56567E-02	-0,54207	7,34638E-01	-7,76639E-02
21	-9,34703	6,69148E-01	-1,29077E-01	-10,23137	7,31028E-01	-1,47005E-01
23	-16,01677	5,36325E-01	-1,05485E-01	-17,55576	5,85810E-01	-1,20265E-01
25	-18,41401	3,57153E-01	-6,96152E-02	-20,18975	3,90021E-01	-7,87731E-02
27	-15,87557	1,78695E-01	-3,39646E-02	-17,38909	1,95067E-01	-3,75640E-02
29	-9,10387	4,78574E-02	-1,08118E-02	-9,9454	5,21770E-02	-1,13907E-02
31	-0,00003	-5,95745E-10	-1,00366E-07	-0,00001	-1,59460E-10	-2,92058E-08

Tabela 4-28 – Resultados do algoritmo "PEFNL-2D" versão 2.1 e 2.2

Nó	Algoritmo desenvolvido versão 2.1			Algoritmo desenvolvido versão 2.2		
	u_0 [°]	u_x [m]	u_y [m]	u_0 [°]	u_x [m]	u_y [m]
1	Não converge para nenhuma solução			-0,00001	7,57859E-10	-2,10327E-08
3				-10,41405	5,58663E-02	-9,00467E-03
5				-17,57882	2,03025E-01	-3,33160E-02
7				-20,11218	3,99243E-01	-7,19746E-02
9				-17,53906	5,95262E-01	-1,10566E-01
11				-10,34205	7,41848E-01	-1,34736E-01
13				-0,60541	7,38124E-01	-2,05282E-01
15				4,24038	7,37690E-01	-1,74262E-01
17				4,26132	7,35475E-01	-1,09352E-01
19				-0,54223	7,35022E-01	-7,77451E-02
21				-10,23599	7,31410E-01	-1,47117E-01
23				-17,56389	5,86116E-01	-1,20357E-01
25				-20,19916	3,90224E-01	-7,88303E-02
27				-17,39711	1,95168E-01	-3,75864E-02
29				-9,94983	5,22031E-02	-1,13942E-02
31				-0,00001	-1,59883E-10	-2,92073E-08

Tabela 4-29 - Resultados da simulação efectuada no Ansys utilizando o elemento de viga BEAM3 e BEAM4.

Nó	Ansys BEAM3			Ansys BEAM4		
	u_0 [°]	u_x [m]	u_y [m]	u_0 [°]	u_x [m]	u_y [m]
1	0	0	0	0	0	0
3	-9,94139	5,33160E-02	-8,78490E-03	-9,48818	5,08710E-02	-8,58460E-03
5	-16,79397	1,93970E-01	-3,15440E-02	-16,0411	1,85270E-01	-2,99170E-02
7	-19,22273	3,81750E-01	-6,74390E-02	-18,3673	3,64900E-01	-6,32620E-02
9	-16,76417	5,69390E-01	-1,03290E-01	-16,0193	5,44430E-01	-9,65740E-02
11	-9,88696	7,09610E-01	-1,25940E-01	-9,44922	6,78510E-01	-1,17830E-01
13	-0,58086	7,06180E-01	-1,93430E-01	-0,55668	6,75350E-01	-1,82360E-01
15	4,05683	7,05750E-01	-1,63770E-01	3,879497	6,74930E-01	-1,54010E-01
17	4,07694	7,03690E-01	-1,01660E-01	3,898634	6,73020E-01	-9,46130E-02
19	-0,52009	7,03240E-01	-7,14340E-02	-0,49891	6,72590E-01	-6,57230E-02
21	-9,78497	6,99910E-01	-1,37790E-01	-9,35239	6,69520E-01	-1,29180E-01
23	-16,77850	5,60930E-01	-1,12670E-01	-16,0256	5,36620E-01	-1,05570E-01
25	-19,29321	3,73500E-01	-7,40670E-02	-18,4246	3,57350E-01	-6,96670E-02
27	-16,62151	1,86830E-01	-3,57160E-02	-15,8847	1,78790E-01	-3,39850E-02
29	-9,52141	5,00050E-02	-1,10940E-02	-9,10888	4,78840E-02	-1,08150E-02
31	0	0	0	0	0	0

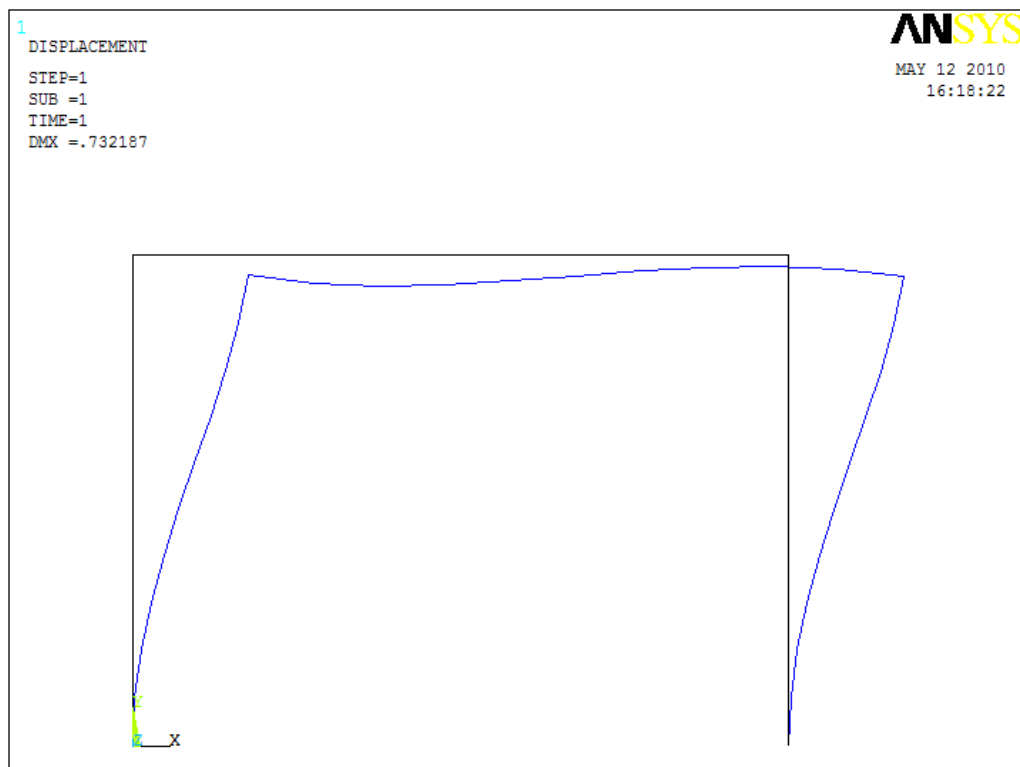


Fig. 4-14 - Configuração inicial e deformada da estrutura

Para comparar os resultados encontrados nas diferentes versões do algoritmo e os encontrados com a simulação no *Ansys*, escolheram-se dois nós que se apresentam como os mais relevantes pela sua posição na estrutura. Para os nós 11 e 21 assinalados na Fig. 4-12 b) tem-se na Tabela 4-30 e na Tabela 4-31 os respectivos erros relativos.

Tabela 4-30 - Comparação de resultados no nó 11 Ansys - algoritmo

Nó 11	Ansys (BEAM4)	Algoritmo			Erro relativo [%]		
		V. 1.1	V. 1.2	V. 2.2	V. 1.1	V. 1.2	V. 2.2
u_{θ} [°]	-9,44922	-9,44370	-10,33714	-10,34205	5,84E-02	9,40	9,45
u_x [m]	6,78510E-01	6,78134E-01	7,41464E-01	7,41848E-01	5,54E-02	9,28	9,33
u_y [m]	-1,17830E-01	-1,17740E-01	-1,34632E-01	-1,34736E-01	7,64E-02	14,3	14,3

Tabela 4-31 - Comparação de resultados no nó 21 Ansys - algoritmo

Nó 21	Ansys (BEAM4)	Algoritmo			Erro relativo [%]		
		V. 1.1	V. 1.2	V. 2.2	V. 1.1	V. 1.2	V. 2.2
u_{θ} [°]	-9,35239	-9,34703	-10,23137	-10,23599	5,73E-02	9,40	9,45
u_x [m]	6,69520E-01	6,69148E-01	7,31028E-01	7,31410E-01	5,56E-02	9,19	9,24
u_y [m]	-1,29180E-01	-1,29077E-01	-1,47005E-01	-1,47117E-01	7,97E-02	13,8	13,9

A comparação de resultados é efectuada considerando os resultados obtidos no *Ansys* com o elemento de viga BEAM4 (3D). Para estes resultados, verifica-se que a única versão do algoritmo “PEFNL-2D” que consegue obter resultados com boa precisão é a versão 1.1.

Na Tabela 4-16 é feita a comparação entre dois factores importantes nas diferentes versões do algoritmo são elas: o número de iterações e o número de incrementos em que dividimos as cargas a aplicar.

Tabela 4-32 - N° de iterações e n° de incrementos nas diferentes configurações do programa "PEFNL-2D"
(Pórtico)

	Pórtico			
	V1.1	V1.2	V2.1	V2.2
N° de iterações	24	7	-	7
N° de incrementos	1	1	-	1

5 CONCLUSÕES

A presente dissertação tem como objectivo a elaboração de um algoritmo em linguagem *Matlab* para analisar estruturas planas constituídas por vigas, utilizando a formulação corrotacional. De acordo com as metodologias apresentadas na literatura, encontram-se diferentes modos para o cálculo da rotação de corpo rígido do elemento (β) e para o cálculo dos esforços internos. Estas dão origem às quatro versões do algoritmo construído (*PEFNL-2D*): V1.1, V1.2, V2.1, V2.2.

Para aferir a qualidade do algoritmo desenvolvido, testaram-se diversos exemplos: “Viga com momento”, “Viga com força axial”, “Viga com força transversal” e “Pórtico”. A versão 2.1 não consegue obter resultados satisfatórios para nenhum dos exemplos. As restantes versões produzem resultados com boa precisão para os três primeiros exemplos e apenas a versão 1.1 se mostrou adequada para o exemplo “Pórtico” revelando-se por isso a opção mais adequada.

O cálculo das rotações de corpo rígido é então uma das questões importantes que deve ser convenientemente tratada para a correcta utilização da formulação corrotacional.

O programa *Ansys* [Ans07] sugere que este cálculo se efectue com base nas rotações dos nós, o que apresenta a vantagem de permitir rotações arbitrárias de qualquer magnitude e em qualquer sentido.

A utilização das coordenadas dos nós para efectuar este mesmo cálculo apresenta a ambiguidade indicada na secção 3.1.3. Esta não permite distinguir rotações de corpo rígido negativas de positivas e até mesmo a sua verdadeira grandeza, uma vez que pode ser um múltiplo do valor encontrado, na forma: $\beta + 2k\pi$, com $k \in \mathbb{Z}$. Contudo, foi desenvolvido um procedimento (3.17), simples e eficiente, para resolver este problema, tendo apenas a limitação de forçar a escolha da amplitude máxima da rotação de corpo rígido.

A partir dos resultados obtidos para os exemplos estudados, constata-se que a escolha da metodologia de cálculo das rotações de corpo rígido condiciona o método de cálculo dos esforços internos. Verifica-se que ao calcular a rotação de corpo rígido a partir das rotações dos nós, não se pode efectuar o cálculo directo dos esforços internos, isto é, das versões desenvolvidas para o programa “*PEFNL-2D*”, não deve ser utilizada a versão 2.1. Verifica-se esta incompatibilidade uma vez que o cálculo directo dos esforços internos assume que o referencial local na iteração n passa pelos dois nós do elemento, o que não

acontece se o eixo local for calculado a partir do eixo inicial considerando uma rotação de corpo rígido com base em $\alpha_{MÉDIO}$.

Contudo, quando se calculam os deslocamentos dos nós no referencial local e se utiliza a matriz de rigidez do elemento para calcular os esforços internos, passa a ser possível contabilizar os deslocamentos transversais dos nós e a sua influência para as forças internas no elemento. Nesse caso o algoritmo converge para a solução correcta e torna possível o uso desta forma de cálculo da rotação de corpo rígido.

O cálculo directo através das equações (3.21), presente na versão 1.1, é uma forma mais eficiente de obter os esforços internos quando comparado com o uso da matriz de rigidez, do vector dos deslocamentos e da equação (3.19), da versão 1.2 e 2.2. Nos vários exemplos estudados, a primeira metodologia referida consegue sempre resultados precisos com um número reduzido de iterações/incrementos, sendo a única a produzir resultados idênticos aos do *Ansys* no exemplo “Pórtico”.

No exemplo referido foram testados dois elementos de viga do *Ansys*, BEAM3 (2D) e BEAM4 (3D), estes produzem resultados diferentes, o que atesta as dificuldades de desenvolvimento da formulação corrotacional. No exemplo “Viga com força transversal” estas dificuldades do *Ansys* também se verificam, uma vez que com o elemento de viga BEAM3 só se obtém convergência para os valores mais reduzidos da carga. E também, ao recorrer ao elemento BEAM4 encontram-se resultados inesperados em $FL^2/EI = 7$ e $FL^2/EI = 10$ onde o programa *Ansys* produz erros no cálculo das rotações.

Para todos os exemplos, à excepção do “Pórtico”, a precisão dos resultados e o número de iterações/incrementos é semelhante quando utilizado o cálculo directo (versão 1.1) ou o cálculo da matriz de rigidez e dos deslocamentos nodais (versão 2.1 e 2.2).

Pode desta forma concluir-se que, das quatro versões estudadas, a versão 1.1 representa a solução mais eficaz e eficiente para desenvolver a formulação corrotacional. Esta provou ser a mais adequada para todos os exemplos, convergindo rapidamente para a solução correcta e sendo inclusivamente a única com capacidades para analisar o exemplo “Pórtico”.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Ans07] Release 11.0 Documentation for ANSYS, ANSYS, Inc., 2007.
- [Bab89] Babuska, I., “*Trends in finite element*”, IEEE Transactions on Magnetics, Vol.25, pp. 2799-2803, 1989.
- [Bat02] Battini, J. M., Pacoste, C., “*Corotational beam elements with warping effects in instability problems*”, Comp. Meth. in Applied Mechanics and Engineering, Vol. 191, pp. 1755-1789, 2002.
- [Bat96] Bathe, K. J., “*Finite Element Procedures, Prentice Hall*”, 1996.
- [Bel00] Belytschko, T., Liu, W. K., Moran, B., “*Nonlinear Finite Elements for Continua and Structures*”, John Wiley and Sons, England, 2000.
- [Bis45] Bishopp, K., E., Drucker, D., C., “*Large deflections of cantilever beams*”, Quarterly of Applied Mathematics, Vol. 3, pp. 272-275, 1945.
- [Dia10] Teixeira-Dias, F., Pinho-da-Cruz, J., Fontes Valente, R., A., Alves de Sousa, R., J., “*Método dos Elementos Finitos-Técnicas de Simulação Numérica em Engenharia*”, 1ª Edição, ETEP, Lisboa, 2010.
- [Fel05] Felippa, C., A., Haugen, B., “*A unified formulation of small-strain corotational finite elements: I. Theory*”, Comp. Meth. in Applied Mechanics and Engineering, Vol. 194, pp. 2285-2335, 2005.
- [Har73] Harrison, H.B., “*Computer methods in structural analysis*”, Prentice Hall, Inc., Englewood Cliffs, New Jersey, USA, 1973.

- [Men03] Menin, R., Taylor, W., “*Resposta pós-crítica de pórticos planos discretizados com elementos de viga de Euler-Bernoulli utilizando uma formulação corrotacional*”, XXIV Iberian Latin-American Congress on Computational Methods in Engineering / Cílamce 2003, Ouro Preto / MG, Brasil
- [Men06] Menin, R., “*Aplicação da descrição cinemática co-rotacional na análise não linear geométrica de estruturas discretizadas por elementos finitos de treliças, vigas e cascas*”, Dissertação de Doutorado pela Universidade de Brasília, Maio de 2006.
- [Pac96] Pacoste, C., Eriksson, A., “*Beam element in instability problems*”, *Comp. Meth. in Applied Mechanics and Engineering*, Vol. 156, pp. 163-197, 1996.
- [Par98] Park, Eun-Jae, “*Some recent topics in numerical analysis-finite element methods*”, *Trends in Mathematics*, Vol.1, pp.121-127, 1998.
- [Ran86] Rankin, C. C., and Brogan, F.A., “*An element independent corotational procedure for the treatment of large rotations*”, *Journal of Pressure Vessel Technology*, Vol. 108, pp.165-174, 1986.
- [Rei01] Reis, A., Camotim, D., “*Estabilidade Estrutural*”, McGraw-Hill, Lisboa, 2001.
- [Tim61] Timoshenko, S., Gere, J., M., “*Theory of Elastic Stability*”, 2nd Edition, McGraw-Hill, New York, 1961.
- [Urt05] Urthaler, Y., Reddy, J., N., “*A corotacional finite element formulation for the analysis of planar beams*”, *Communications in Numerical Methods in Engineering*, Vol. 21, pp. 553-570, 2005.

ANEXOS

Algoritmo “PEFNL-2D”:

```
% Programa PEFNL-2D
% Programa de elementos finitos não lineares versão 2D
%
% Versão 1.0 - Realiza análises não lineares elástica com elementos
%             viga 2D com elementos corotacionais
%
% -----
% O ficheiro de dados, '*.inp' deve conter :
% - coordenadas dos nos,
% - definicao dos elementos,
% - caracteristicas dos materiais,
% - caracteristicas das secções
% - forcas aplicadas
% - apoios
%
disp('Programa PEFNL-2D');
%-----
%
% Parametros de controlo do programa e algoritmo
% incremental-iterativo de Newton-Raphson
%
% Estratégia (pode tomar o valor [1 1], [1 2], [2 1], [2 2])
% [1 X] - Calculo das rotações corpo rígido a partir das coordenadas
% [2 X] - Calculo das rotações corpo rígido a partir das rotações dos nós
% [X 1] - Calculo directo dos esforços internos
% [X 2] - Calculo dos esforços internos a partir da matriz rigidez
estrategia= [1 1];
% Numero de incrementos da carga aplicada
ninc= 1;
% Número máximo de iterações
max= 250;
% Erro máximo
tolerancia= 1e-9;
%-----
% Abertura do ficheiro de dados
%
tipo={'*.inp'};
titulo='Elastica: Seleccione o ficheiro de dados';
[nome,caminho]=uigetfile(tipo,titulo);
ficheiro=[caminho,nome];
disp(nome);
if nome == 0 % Verifica se foi seleccionado um ficheiro
    warndlg('Ficheiro não encontrado','Elastica');
else
    %-----
    % Leitura dos dados
    % *****
    fid= fopen(ficheiro,'r');
    nnos= fscanf(fid,'%d',1);
    fprintf('numero de nos= %3d\n',nnos)
    nos= fscanf(fid,'%f',[2 nnos]);
    nelementos= fscanf(fid,'%d',1);
    fprintf('numero de elementos= %3d\n',nelementos)
    elementos= fscanf(fid,'%f',[4 nelementos]);
    nmateriais= fscanf(fid,'%d',1);
    fprintf('numero de materiais= %3d\n',nmateriais)
    materiais= fscanf(fid,'%f',[1 nmateriais]);
    nseccoes= fscanf(fid,'%d',1);
```

```
fprintf('numero de seccoes= %3d\n',nseccoes)
seccoes= fscanf(fid,'%f',[2 nseccoes]);
nforcas= fscanf(fid,'%d',1);
fprintf('numero de forcas= %3d\n',nforcas)
forcas= fscanf(fid,'%f',[4 nforcas]);
napoios= fscanf(fid,'%d',1);
fprintf('numero de apoios= %3d\n',napoios)
apoios= fscanf(fid,'%d',[4 napoios]);
fclose(fid);
%
% Troca a extensão do ficheiro de dados para '*.des'
%
dim=size(ficheiro);
ficheiro(dim(2)-2)='d';
ficheiro(dim(2)-1)='e';
ficheiro(dim(2))='s';
fid= fopen(ficheiro,'w');
fprintf(fid,'Deslocamentos nos nos\n');
%
% Troca a extensão do ficheiro para '*.esf'
%
dim=size(ficheiro);
ficheiro(dim(2)-2)='e';
ficheiro(dim(2)-1)='s';
ficheiro(dim(2))='f';
fie= fopen(ficheiro,'w');
fprintf(fie,'Esforços nos elementos\n');
%
%-----
% Cálculos iniciais
% *****
%
% A matriz nos_actuais(2,nnos) guarda as coordenadas actualizadas
% dos nós. Na primeira iteração elas são iguais 'as iniciais
%
nos_actuais= nos;
%
% O vector alfa(1,nnos) guarda as phiações nos nós e o vector
% ug(2,nnos) as translações nos nós. Estas phiações e translações são
% inicialmente nulas e vão sendo incrementadas durante a solução
% iterativa até convergirem para a solução final
%
alfa= zeros(1,nnos);
ug= zeros(2,nnos);
%
% O vector p guarda a força axial em cada elemento
%
p= zeros(nelementos,1);
%
% O vector rot guarda a rotação de corpo rígido de cada elemento
%
if estrategia(1) == 2
    rot= zeros(nelementos,1);
end
%
% Calcula e guarda o comprimento inicial de cada elemento, bem como a
% rotação inicial do referencial local em relação ao referencial
global
%
ll= zeros(1,nelementos);
teta= zeros(1,nelementos);
%
```

```

for i=1:nelementos
    no1= elementos(1,i);
    no2= elementos(2,i);
    dxi= nos(1,no2)-nos(1,no1);
    dyi= nos(2,no2)-nos(2,no1);
    ll(i)= sqrt( dxi * dxi + dyi * dyi ) ;
    teta(i)= acos(dxi/ll(i));
    if dyi < 0
        teta(i)=-teta(i);
    end
end
%
%-----
% Método incremental-iterativo de Newton-Raphson
% *****
%
% Inicialização do processo incremental
forcas_iniciais=forcas;
inc=1;
iter= 1;
%
for inc=1:ninc
    %
    % Inicialização do processo iterativo
    %
    lambda= inc/ninc; % factor de carga
    fprintf('*** incremento %d , Factor= %f *** \n',inc,lambda);
    fprintf(fid,'*** Incremento %d , Factor= %f *** \n',inc,lambda);
    erro= 9999;
    forcas=[forcas(1,:);forcas_iniciais(2:4,:)*lambda];
    %
    while erro > tolerancia && iter <= max
        %
        %-----
        % Calculo das forcas internas e assemblagem do vector de
        % forças
        *****
        fprintf('*** iteração %d *** \n',iter);
        %
        f= zeros(nnos*3,1);
        %
        % Na primeira iteração não vale a pena calcular as forças
        % internas, pois são nulas.
        %
        if iter > 1
            %fprintf('calcula as forças internas\n');
            fprintf(fie,...
                ' Iter= %d\n El. , phi\n N (nó1)      , V (nó 1)
',iter);

            fprintf(fie,...
                ' M (nó 1)      , N (nó 2)      , V (nó 2)      , M (nó
2)\n');
        %
        for i=1:nelementos
            no1= elementos(1,i);
            no2= elementos(2,i);
            mat= elementos(3,i);
            sec= elementos(4,i);
            ae= seccoes(1,sec)*materiais(1,mat);
            ei= materiais(1,mat)*seccoes(2,sec);
            li= ll(i) ;
            li2= li * li ;

```

```

li3= li2 * li ;
%
%-----
% Calcula a matriz de transformação to para a posição
% do eixo x inicial
%
to= zeros(2);
dx= nos(1,no2)-nos(1,no1);
dy= nos(2,no2)-nos(2,no1);
to(1,1)= dx/li; to(1,2)= dy/li;
to(2,1)= -to(1,2); to(2,2)= to(1,1);
%
%-----
% Calcula a matriz de transformação tn para a posição
% do eixo x actual
%
tn= zeros(2);
dx= nos_actuais(1,no2)-nos_actuais(1,no1);
dy= nos_actuais(2,no2)-nos_actuais(2,no1);
l= sqrt( dx * dx + dy * dy );
tn(1,1)= dx/l; tn(1,2)= dy/l;
tn(2,1)= -tn(1,2); tn(2,2)= tn(1,1);
%
%-----
% Calcula o angulo que define a rotação de corpo
% rígido, phi, como sendo o angulo entre o eixo x
% local inicial e o eixo x local actual
%
if estrategia(1) == 1
    ang= acos(tn(1,1));
    if tn(1,2) < 0
        ang=-ang;
    end
    %
    % O ângulo ang deve ser corrigido considerando o
    % valor actual da rotação nos dois nós.
    alfam= (alfa(no1)+alfa(no2))/2;
    %
    % Escolhe o elemento da família de ângulos que
    % melhor aproxima o ângulo de rotação médio,
    % considerando a possibilidade de existência de
    % rotações entre -1080° e +1080° (3x360°)
    lmin= 9999;
    kmin= 9;
    for k=-3:+3
        %
        angx= ang + 2*k*pi;
        %
        % Calcula a diferença entre os dois ângulos
        d= alfam - angx;
        %
        % Calcula o comprimento de d
        lx= abs(d);
        if lx < lmin
            lmin= lx;
            kmin= k;
        end
    end
    %
    ang= ang + 2*kmin*pi;
    %
    phi= ang-teta(i);

```

```

        %
    end
    %
    %-----
    % Calcula o angulo que define a rotação de corpo
    % rígido com base na média das rotações nos dois nós
    % do elemento, que está guardada no vector rot()
    %
    if estrategia(1) == 2
        %
        phi= rot(i);
        %
    end
    %
    tetal= alfa(no1) - phi;
    teta2= alfa(no2) - phi;
    %
    %-----
    % Calcula os esforços internos directamente pelas
    % equações de Menin
    %
    if estrategia(2) == 1
        d= (l-li);
        N= ae*d/li ;
        V= 6*ei*(tetal+teta2)/l/li;
        M1= 2*ei*(2*tetal+teta2)/li;
        M2= 2*ei*(tetal+2*teta2)/li;
        fe=zeros(6,1);
        fe(1)= -N;
        fe(2)= V;
        fe(3)= M1;
        fe(4)= N;
        fe(5)= -V;
        fe(6)= M2;
    end
    %
    %-----
    % Calcula os deslocamentos dos nos associados às
    % deformações pelas equações do Ansys
    %
    if estrategia(2) == 2
        %
        % Calcula a matriz rn
        %
        rn= zeros(2);
        rn(1,1)=cos(phi); rn(1,2)=sin(phi);
        rn(2,1)=-rn(1,2); rn(2,2)=rn(1,1);
        uedef= zeros(6,1);
        %
        % Calcula as translações nos nós associadas às
        % deformações pela equação (3.43) do ANSYS
        % uedef= to * ( rn ( xv + ug ) - xv ) ;
        %
        uedef(1:2)= to*(rn*nos_actuais(1:2,no1)-
nos(1:2,no1));
        uedef(4:5)= to*(rn*nos_actuais(1:2,no2)-
nos(1:2,no2));

        %
        % Calcula as rotações nos nós associados às
        % deformações subtraindo a rotação de corpo
        % rígido à rotação no nó
        %
    end

```

```

        uedef(3)= teta1;
        uedef(6)= teta2;
        %
        % Calcula os esforços internos a partir das
        % deformações
        %
        k= zeros(6);
        %
        % Matriz de rigidez do elemento viga 2D
        % em coordenadas locais
        %
        k(1,1)= ae / li ;
        k(1,4)= -k(1,1) ;
        k(2,2)= 12 * ei / li3 ;
        k(2,3)= 6 * ei / li2 ;
        k(2,5)= -k(2,2) ;
        k(2,6)= k(2,3) ;
        k(3,2)= k(2,3) ;
        k(3,3)= 4 * ei / li ;
        k(3,5)= -k(3,2) ;
        k(3,6)= k(3,3) / 2.0 ;
        k(4,1)= k(1,4) ;
        k(4,4)= k(1,1) ;
        k(5,2)= k(2,5) ;
        k(5,3)= k(3,5) ;
        k(5,5)= k(2,2) ;
        k(5,6)= -k(2,3) ;
        k(6,2)= k(2,6) ;
        k(6,3)= k(3,6) ;
        k(6,5)= k(5,6) ;
        k(6,6)= k(3,3) ;
        %
        % Calcula os esforços internos fe(6)
        %
        fe= k * uedef ;
    end
    %
    % Guarda a força axial e imprime os esforços
    %
    p(i)= fe(4);
    fprintf(fie,...
        'El= %3d , %12.4e , %12.4e , %12.4e , %12.4e ,
%12.4e , %12.4e\n'...
        ,i,fe(1),fe(2),fe(3),fe(4),fe(5),fe(6));
    %
    %-----
    % Efectua a transformação de coordenadas para
    % obter as forças internas no referencial global
    %
    tnn= eye(6);
    tnn(1:2,1:2)= tn;
    tnn(4:5,4:5)= tn;
    fc= tnn' * fe;
    %-----
    % E adiciona as forças internas ao
    % vector de forças globais
    %
    f(no1*3-2:no1*3)= f(no1*3-2:no1*3)-fc(1:3);
    f(no2*3-2:no2*3)= f(no2*3-2:no2*3)-fc(4:6);
    %
end % for i=1:nelementos
%
```

```
end % if iter > 1
%
% Calculo de K e resolução do sistema Ku=f
% *****
% Calculo da matriz de rigidez global, K, e do vector de
% incrementos nos deslocamentos, u, para a iteração corrente.
% (Na primeira iteração, considera-se o problema elástico
% linear, e calculam-se os deslocamentos devidos às forças
% aplicadas)
%
K= zeros(nnos*3);
u= zeros(nnos*3,1);
%
% Calculo da matriz de rigidez e da matriz geométrica de cada
% elemento e adição à matriz global
%
for i=1:nelementos
%
% inicializa a matriz de rigidez do elemento
k= zeros(6);
g= zeros(6);
tnn= eye(6);
no1= elementos(1,i);
no2= elementos(2,i);
mat= elementos(3,i);
sec= elementos(4,i);
ae= seccoes(1,sec)*materiais(1,mat);
ei= materiais(1,mat)*seccoes(2,sec);
li= ll(i) ;
li2= li * li ;
li3= li2 * li ;
%
% Matriz de rigidez do elemento viga 2D
% em coordenadas locais
k(1,1)= ae / li ;
k(1,4)= -k(1,1) ;
k(2,2)= 12 * ei / li3 ;
k(2,3)= 6 * ei / li2 ;
k(2,5)= -k(2,2) ;
k(2,6)= k(2,3) ;
k(3,2)= k(2,3) ;
k(3,3)= 4 * ei / li ;
k(3,5)= -k(3,2) ;
k(3,6)= k(3,3) / 2.0 ;
k(4,1)= k(1,4) ;
k(4,4)= k(1,1) ;
k(5,2)= k(2,5) ;
k(5,3)= k(3,5) ;
k(5,5)= k(2,2) ;
k(5,6)= -k(2,3) ;
k(6,2)= k(2,6) ;
k(6,3)= k(3,6) ;
k(6,5)= k(5,6) ;
k(6,6)= k(3,3) ;
%
% Matriz geometrica
%
ax1= p(i);
g(2,2)= 6*ax1/(5*li) ;
g(2,3)= ax1/10 ;
g(2,5)= -g(2,2) ;
g(2,6)= g(2,3) ;
```

```

g(3,2)= g(2,3) ;
g(3,3)= 2*axl*li/15 ;
g(3,5)= -g(2,3) ;
g(3,6)= -axl*li/30 ;
g(5,2)= g(2,5) ;
g(5,3)= g(3,5) ;
g(5,5)= g(2,2) ;
g(5,6)= -g(2,3) ;
g(6,2)= g(2,6) ;
g(6,3)= g(3,6) ;
g(6,5)= g(5,6) ;
g(6,6)= g(3,3) ;
%
%-----
% Calcula a matriz de transformação tn para a posição
% do eixo x actual
dx= nos_actuais(1,no2)-nos_actuais(1,no1);
dy= nos_actuais(2,no2)-nos_actuais(2,no1);
l= sqrt( dx * dx + dy * dy ) ;
tnn(1,1)= dx/l; tnn(1,2)= dy/l;
tnn(2,1)= -tnn(1,2); tnn(2,2)= tnn(1,1);
tnn(4:5,4:5)=tnn(1:2,1:2);
kg= tnn'*(k+g)*tnn;
%-----
%
% assemblagem
in=no1*3-2;
jn=no2*3-2;
K(in:in+2,in:in+2)= K(in:in+2,in:in+2)+kg(1:3,1:3);
K(in:in+2,jn:jn+2)= K(in:in+2,jn:jn+2)+kg(1:3,4:6);
K(jn:jn+2,in:in+2)= K(jn:jn+2,in:in+2)+kg(4:6,1:3);
K(jn:jn+2,jn:jn+2)= K(jn:jn+2,jn:jn+2)+kg(4:6,4:6);
% disp(K);
%
end % for i=1:nelementos
%
% Calcula a contribuição das forças concentradas
% para o vector de forcas
%
for i= 1:nforcas
    no= forcas(1,i);
    in=no*3-2;
    f(in:in+2,1)= f(in:in+2,1)+forcas(2:4,i);
end
%
% Penaliza a matriz de rigidez devido a existencia de apoios
%
for i= 1:napoios
    no= apoios(1,i);
    in=no*3-2;
    for j=2:4
        if apoios(j,i) == 1
            K(in+j-2,in+j-2)= K(in+j-2,in+j-2)*1e6;
        end
    end
end
%
% Resolve o sistema de equacoes
%
u= K\f;
%
% Imprime os deslocamentos e actualiza vários vectores e

```

```

% matrizes
%
fprintf(fid,...
        ' Iter= %d\n No ,      X      ,      Y      ,
Teta\n',iter);
for i=1:nnos
%
% Atualiza as translações globais
%
ug(1:2,i)= ug(1:2,i)+u(i*3-2:i*3-1);
%
% Atualiza as rotações globais
%
alfa(i)= alfa(i)+u(i*3);
%
% Imprime os deslocamentos
%
fprintf(fid,' %3d , %15.6e , %15.6e , %15.6e
(%10.5f°)\n',...
        i,ug(1,i),ug(2,i),alfa(i),alfa(i)*180/pi);
end
%
% Atualiza as coordenadas dos nós
%
nos_actuais= nos + ug ;
%
% Atualiza o vector com as rotações de corpo rígido dos
% elementos
%
if estrategia(1) == 2
for i=1:nelementos
no1= elementos(1,i);
no2= elementos(2,i);
delta_alfa= (u(no1*3)+u(no2*3))/2;
rot(i)= rot(i)+delta_alfa;
end
end
%
% Calcula o erro
%
erro= u'*u;
fprintf(fid,'erro= %e\n',erro);
%
% Incrementa o contador de iterações
%
iter= iter + 1;
end % while ...
%
end % for
fclose(fid);
fclose(fie);
helpdlg('O programa executou com sucesso todos os cálculos','PEFNL-
2D');
end
fprintf('n_iter= %d, FIM\n',iter-1);
% fim

```


Ficheiro de input para Matlab “Viga com momento”:

```
21
0      0
0.127  0
0.254  0
0.381  0
0.508  0
0.635  0
0.762  0
0.889  0
1.016  0
1.143  0
1.270  0
1.397  0
1.524  0
1.651  0
1.778  0
1.905  0
2.032  0
2.159  0
2.286  0
2.413  0
2.540  0
20
1  2  1  1
2  3  1  1
3  4  1  1
4  5  1  1
5  6  1  1
6  7  1  1
7  8  1  1
8  9  1  1
9 10  1  1
10 11 1  1
11 12 1  1
12 13 1  1
13 14 1  1
14 15 1  1
15 16 1  1
16 17 1  1
17 18 1  1
18 19 1  1
19 20 1  1
20 21 1  1
1
2.06844e11
1
1.6129e-4  2.16787e-9
1
21 0 0 -1109.23284
1
1 1 1 1
```



```

E, 2, 3
E, 3, 4
E, 4, 5
E, 5, 6
E, 6, 7
E, 7, 8
E, 8, 9
E, 9, 10
E, 10, 11
E, 11, 12
E, 12, 13
E, 13, 14
E, 14, 15
E, 15, 16
E, 16, 17
E, 17, 18
E, 18, 19
E, 19, 20
E, 20, 21
!
! M = pi * E * I / L
!
!      ou
!
! M = 2 * pi * E * I / L
!
F, 21, MZ, -1109.23284
!
D, 1, ALL
!
FINISH
!
!-----! PROBLEMA ESTÁTICO: [K]u=F
!
/SOLU      !*** SOLUÇÃO ***
SOLCONTROL, 0
NEQIT, 150      !Maximo de iterações
ANTYPE, STATIC  !Tipo de análise: estática
NLGEOM, ON      !Grandes deslocamentos
SOLVE         !Resolve sistema
/POST1        !*** PÓS-PROCESSAMENTO ***
PPRSOL        !Mostra reacções nos apoios
PRNSOL, DOF   !Mostra deslocamentos e
rotações
FINISH
!

```

Ficheiro de input para Matlab “Viga com força axial”:

```
11
0.0 0.0
0.0 0.254
0.0 0.508
0.0 0.762
0.0 1.016
0.0 1.270
0.0 1.524
0.0 1.778
0.0 2.032
0.0 2.286
0.0 2.540
10
1 2 1 1
2 3 1 1
3 4 1 1
4 5 1 1
5 6 1 1
6 7 1 1
7 8 1 1
8 9 1 1
9 10 1 1
10 11 1 1
1
2.06844e11
1
1.6129e-4 2.16787e-9
1
11 2.23 -323.09213 0.0
1
1 1 1 1
```



```
FDEL, 11, FX                !Retira a força horizontal
F, 11, FY, FCR*1.063
SOLVE                       !LOAD STEP 2
F, 11, FY, FCR*1.152
SOLVE                       !LOAD STEP 3
F, 11, FY, FCR*1.293
SOLVE                       !LOAD STEP 4
F, 11, FY, FCR*1.518
SOLVE                       !LOAD STEP 5
F, 11, FY, FCR*1.884
SOLVE                       !LOAD STEP 6
FINISH
```

Ficheiro de input para Matlab “Viga com força transversal”:

```
11
 0.0  0.0
 0.254 0.0
 0.508 0.0
 0.762 0.0
 1.016 0.0
 1.27  0.0
 1.524 0.0
 1.778 0.0
 2.032 0.0
 2.286 0.0
 2.54  0.0
10
 1  2  1  1
 2  3  1  1
 3  4  1  1
 4  5  1  1
 5  6  1  1
 6  7  1  1
 7  8  1  1
 8  9  1  1
 9 10  1  1
10 11  1  1
1
2.06844e11
1
6.4516e-4  3.4686e-8
1
11 0 -11120.6383 0.0
1
1 1 1 1
```



```
E, 10, 11
!
! P = 10 * E * I / L2
!
F, 11, FY, -11120.63814
!
D, 1, ALL
!
FINISH
!
!-----! PROBLEMA ESTÁTICO: [K]u=F
!
/SOLU                                !*** SOLUÇÃO ***
SOLCONTROL, 0
NEQIT, 150                            !Maximo de iterações
ANTYPE, STATIC                        !Tipo de análise: estática
NLGEOM, ON                            !Grandes deslocamentos
SOLVE                                  !Resolve sistema
/POST1                                 !*** PÓS-PROCESSAMENTO ***
PRRSOL                                !Mostra reacções nos apoios
PRNSOL, DOF                           !Mostra deslocamentos e rotações
FINISH
!
```

Ficheiro de input para Matlab “Pórtico”:

```
31
0.0  0.0
0.0  0.3
0.0  0.6
0.0  0.9
0.0  1.2
0.0  1.5
0.0  1.8
0.0  2.1
0.0  2.4
0.0  2.7
0.0  3.0
0.4  3.0
0.8  3.0
1.2  3.0
1.6  3.0
2.0  3.0
2.4  3.0
2.8  3.0
3.2  3.0
3.6  3.0
4.0  3.0
4.0  2.7
4.0  2.4
4.0  2.1
4.0  1.8
4.0  1.5
4.0  1.2
4.0  0.9
4.0  0.6
4.0  0.3
4.0  0.0
30
1  2  1  1
2  3  1  1
3  4  1  1
4  5  1  1
5  6  1  1
6  7  1  1
7  8  1  1
8  9  1  1
9  10  1  1
10 11  1  1
11 12  1  1
12 13  1  1
13 14  1  1
14 15  1  1
15 16  1  1
16 17  1  1
17 18  1  1
18 19  1  1
19 20  1  1
```

```
20 21 1 1
21 22 1 1
22 23 1 1
23 24 1 1
24 25 1 1
25 26 1 1
26 27 1 1
27 28 1 1
28 29 1 1
29 30 1 1
30 31 1 1
1
210e9
1
14. 9e-3  252e-6
2
11 2000000  -37500000  0
21      0  -37500000  0
2
  1 1 1 1
31 1 1 1
```



```
N, 26, 4.0, 1.5
N, 27, 4.0, 1.2
N, 28, 4.0, 0.9
N, 29, 4.0, 0.6
N, 30, 4.0, 0.3
N, 31, 4.0, 0.0
!
ET, 1, BEAM4                ! Define tipo de elemento
R, 1, 14.9E-3, 252E-6      ! Constantes reais: área e segundo momento de
área
MP, EX, 1, 210E9          ! Módulo de Young
MP, PRXY, 1, 0.3         ! Coeficiente de Poisson
!
E, 1, 2
E, 2, 3
E, 3, 4
E, 4, 5
E, 5, 6
E, 6, 7
E, 7, 8
E, 8, 9
E, 9, 10
E, 10, 11
E, 11, 12
E, 12, 13
E, 13, 14
E, 14, 15
E, 15, 16
E, 16, 17
E, 17, 18
E, 18, 19
E, 19, 20
E, 20, 21
E, 21, 22
E, 22, 23
E, 23, 24
E, 24, 25
E, 25, 26
E, 26, 27
E, 27, 28
E, 28, 29
E, 29, 30
E, 30, 31
!
F, 11, FX, 2000000
F, 11, FY, -37500000
F, 21, FY, -37500000
!
D, 1, ALL
D, 31, ALL
!
FINISH
!
! -----! PROBLEMA ESTÁTICO: [K]u=F
```

```
!  
/SOLU                                ! *** SOLUÇÃO ***  
SOLCONTROL, 0  
NEQIT, 150                            ! Máximo de iterações  
ANTYPE, STATIC                        ! Tipo de análise: estática  
NLGEOM, ON                            ! Grandes deslocamentos  
SOLVE                                 ! Resolve sistema  
/POST1                                ! *** PÓS-PROCESSAMENTO ***  
PRRSOL                               ! Mostra reacções nos apoios  
PRNSOL, DOF                          ! Mostra deslocamentos e rotações  
FINISH  
!
```