

32

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E
SISTEMAS MECÂNICOS**

Nota final
8,3 (oitos e tres)
WAm

VIBRAÇÃO DE VIGAS ALINHADAS NO PLANO

Fernando D'Agosto Toledo

São Paulo
2005

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E
SISTEMAS MECÂNICOS**

VIBRAÇÃO DE VIGAS ALINHADAS NO PLANO

Trabalho de formatura apresentado à Escola
Politécnica da Universidade de São Paulo para
obtenção do título de Graduação em Engenharia

Fernando D'Agosto Toledo

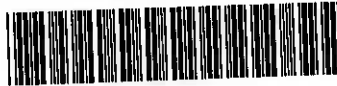
Orientador: Carlos Alberto Nunes Dias

Área de Concentração:
Engenharia Mecatrônica

São Paulo
2005

326

DEDALUS - Acervo - EPMN



31600011837

14939990

FICHA CATALOGRÁFICA

Toledo, Fernando D Agosto

Vibração de vigas alinhadas no plano / F.D A. Toledo. -- São Paulo, 2005.

50 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Vibrações 2.Vigas I.Universidade de São Paulo. Escola Poli-
técnica. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos II.t.

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE
TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO
PARA FINS DE ESTUOD E PESQUISA, DESDE QUE CITADA A FONTE.

Resumo

Estudou-se neste trabalho toda fundamentação teórica necessária para que um programa que calculasse exatamente as frequências naturais e os modos de vibrar de uma estrutura plana formada vigas fosse desenvolvido. O programa proposto trabalha com as hipóteses necessárias para garantir um modelo contínuo e linear e sua programação em MATLAB R12 permitiu manter o foco do trabalho nos diversos problemas de vibração possíveis ao invés de se prender aos detalhes de algoritmos de cálculo numérico. A opção por este software desempenhou um papel importante ao garantir a utilização do programa em diversas plataformas computacionais sem que fossem necessárias recompilações e adaptações do código fonte, nem a criação de um pacote de instalação complexo. A relevância deste trabalho reside na não existência de valores tabelados para o cálculo destas propriedades em estruturas com tal complexidade e os valores calculados pelo programa são a solução exata do modelo matemático que nos propusemos a resolver. O programa resultante pode ser utilizado em pequenos projetos de engenharia ou como referência para programas que utilizam métodos aproximados como é o caso de elementos finitos. Diversos casos típicos foram avaliados e tiveram seus resultados comparados aos obtidos com um software de elementos finitos amplamente utilizado no mercado, comprovando a funcionalidade do programa.

Palavras-chave: vibração; vigas; frequência natural; modo de vibrar.

Abstract

In this work all theory needed for the development of a program that calculates the natural frequencies and the mode shapes of a plain structure formed by beams was studied. This program was made based on the achieved theory and the necessary hypothesis to guarantee that the model could be considered linear and continuous. Programming in MATLABR12 allowed us keeping the focus in the possible vibration problems instead of algorithms development. The option for this software played an important role by guaranteeing its use over different platforms without recompilations and adaptations of its source code, nor the creation of a complex installation package. This work is relevant because we can't find these properties (mode shapes and natural frequencies) tabled in books. Therefore this program solves the exact mathematical model. The resultant program can be used in small engineering projects or as a benchmark for commercial programs as finite elements ones. The results obtained by the program for typical structures were presented in comparison with widely used commercial finite element software, proving the program's functionality.

Keywords: vibration; beam; mode shape; natural frequencies.

Lista de Ilustrações

Ilustração 1, Narrows Bridge, Tacoma, WA , USA– oscilações induzidas pelo vento.	9
Ilustração 2, Ilustração de uma viga e os esforços nas extremidades.....	12
Ilustração 3, de contorno típicas	13
Ilustração 4, Vértice da estrutura.....	17
Ilustração 5, Interface Gráfica com o Usuário.....	25
Ilustração 6, Ilustração da Output Window.....	27
Ilustração 7, Screenshot do Programa em Execução	28

Lista de Tabelas

Tabela 1, Condições de contorno.....	13
Tabela 2, Codificação das Condições de Apoio	25
Tabela 3, Descrição das Variáveis.....	26
Tabela 4, Comparação de Resultados	29
Tabela 5, dos resultados considerando-se as massas concentradas	30

Sumário

RESUMO.....	4
ABSTRACT.....	5
LISTA DE ILUSTRAÇÕES.....	6
LISTA DE TABELAS.....	7
SUMÁRIO	8
1. INTRODUÇÃO	9
2. REVISÃO BIBLIOGRÁFICA.....	11
2.1. FLEXÃO.....	11
2.2. AXIAL.....	12
2.3. MASSAS E MOLAS CONCENTRADAS NAS EXTREMIDADES	15
2.4. ESTRUTURA DE VIGAS	16
2.4.1. COMPATIBILIDADE DE ESFORÇOS E DESLOCAMENTOS NUM VÉRTICE DA ESTRUTURA.....	17
2.4.2. OBTENDO O SISTEMA DE EQUAÇÕES.....	18
3. EXEMPLO DE IMPLEMENTAÇÃO.....	19
3.1. MONTANDO O SISTEMA DE EQUAÇÕES	20
3.2. ALGORITMO DE RESOLUÇÃO	22
3.3. PROGRAMAÇÃO	23
3.4. DESENVOLVIMENTO DA IGU (INTERFACE GRÁFICA COM O USUÁRIO).....	24
4. RESULTADOS.....	29
CONCLUSÕES.....	34
REFERÊNCIAS BIBLIOGRÁFICAS.....	35
ANEXO 1 – CODIGO FONTE DO PROGRAMA PROPOSTO.....	36

1. Introdução

O estudo de vibrações contempla o movimento oscilatório de sistemas e os esforços a eles associados. No projeto dos mais diversos tipos de estruturas e mecanismos em engenharia não se pode negligenciar o estudo do comportamento oscilatório. Um exemplo clássico, ilustrado na figura abaixo, é o caso da ponte “Narrows Bridge” que colapsou devido a vibrações induzidas pelo vento.

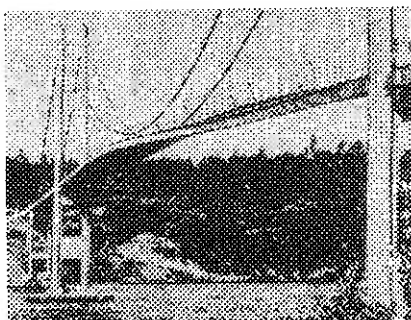


Ilustração 1, Narrows Bridge, Tacoma, WA , USA– oscilações induzidas pelo vento.

Os fenômenos relacionados à vibração dos corpos influenciam na segurança, precisão, funcionamento e conforto de máquinas, construções, etc., e atualmente temos à disposição diversos tipos de ferramentas para auxiliar na previsão destes efeitos e projetar considerando este aspecto.

Os sistemas podem ser caracterizados como lineares ou não lineares e existem métodos numéricos e analíticos para solução destes modelos. Uma ferramenta muito utilizada atualmente é o método dos elementos finitos que é, no entanto, aproximado na solução deste modelo matemático. A proposta deste trabalho é desenvolver uma fundamentação teórica que possa ser utilizada para resolver analiticamente problemas de vibrações em estruturas planas formadas por vigas. Assim é possível obter a solução exata do modelo matemático que descreve este fenômeno.

Baseando-se na fundamentação teórica levantada foi proposto um software que recebe os dados de uma estrutura plana de vigas e calcula suas frequências

naturais e seus respectivos modos de vibrar. Ele resolve problemas com vigas alinhadas e vigas em ângulo, também solucionando alguns casos de pórticos planos. Para estes casos mais complexos o software dispõe de estruturas pré-programadas que foram testadas e para as quais devemos providenciar alguns parâmetros.

Gerar uma interface que permita ao usuário entrar com uma estrutura complexa genérica requer uma dedicação à programação de interfaces gráficas muito além do escopo do presente trabalho. Isto fica como uma sugestão para futuros trabalhos.

2. Revisão Bibliográfica

Apresentar-se-á a seguir a teoria envolvida na vibração de uma viga no plano e de estruturas formadas por vigas delgadas:

2.1. Flexão

A vibração de uma viga delgada à flexão ocorre segundo as seguintes hipóteses: viga longa e delgada; material isotrópico e homogêneo; deslocamentos dentro do regime elástico do material (THOMSON). Assim garantimos a linearidade do modelo.

Na vibração de uma viga, cada trecho infinitesimal realiza um movimento harmônico síncrono ao longo da viga, em torno de sua posição de equilíbrio. Neste caso consideraremos a flecha $v(x, t)$ é função da posição x ao longo do comprimento da viga e do tempo t . A amplitude do movimento ao longo da viga é dada por uma função $\phi(x)$, a qual se denomina modo de vibrar. A flecha no movimento harmônico síncrono é descrita, portanto, por:

$$v(x, t) = \phi(x) \cdot \text{sen}(\omega t) \quad (1)$$

Onde " ω " corresponde à frequência natural do sistema. Agora, a partir da teoria da resistência dos materiais, baseando-se na viga descrita por Timoshenko, obtêm-se os esforços solicitantes na estrutura a partir da flecha:

$$\theta(x, t) = \frac{\partial}{\partial x} v(x, t) = v'(x, t) = \phi'(x) \cdot \text{sen}(\omega t); \quad (2)$$

$$M(x, t) = E.I \cdot v''(x, t) = E.I \cdot \phi''(x) \cdot \text{sen}(\omega t); \quad (3)$$

$$Q(x, t) = E.I \cdot v'''(x, t) = E.I \cdot \phi'''(x) \cdot \text{sen}(\omega t); \quad (4)$$

Onde “ θ ” é a rotação de cada elemento infinitesimal da viga. “ M ” é o momento fletor e “ Q ” é a força cortante, atuantes na estrutura. Ainda podemos obter uma forma para o modo de vibrar $\phi(x)$ que satisfaça estas equações diferenciais ao fazermos o equilíbrio do sistema:

$$\phi(x) = C_1 \cdot \sin(\xi \cdot \lambda) + C_2 \cdot \cos(\xi \cdot \lambda) + C_3 \cdot \sinh(\xi \cdot \lambda) + C_4 \cdot \cosh(\xi \cdot \lambda); \quad (5)$$

Com:

$$\beta = \frac{\lambda}{L}; \quad (6)$$

$$\omega^2 = \beta^4 \cdot \left(\frac{EI}{m} \right); \quad (7)$$

$$\xi = \frac{x}{L}; \quad (8)$$

Sendo “ L ” o comprimento da viga, “ E ” o módulo de elasticidade e “ I ” o momento de inércia da seção transversal. Ainda temos “ ξ ”, “ β ” e “ λ ”, variáveis auxiliares.

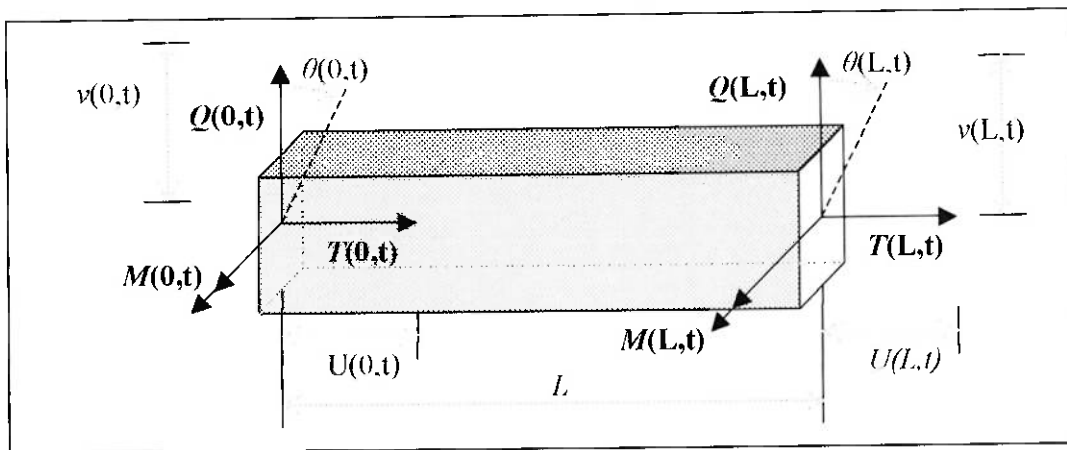


Ilustração 2, Ilustração de uma viga e os esforços nas extremidades

2.2. Axial

Na vibração axial, analogamente, o deslocamento axial é dado por:

$$u(x,t) = \phi_t(x) \cdot \sin(\omega t) \quad (9)$$

Sendo a tração obtida por

$$T(x,t) = EA.\phi_1'(x).\text{sen}(\omega t) \quad (10)$$

E o modo de vibrar axial:

$$\phi_1(x) = A.\text{sen}(\beta_1.x) + B.\cos(\beta_1.x) \quad (11)$$

Com:

$$\omega^2 = \beta^2 \cdot \left(\frac{E.A}{m} \right) \quad (12)$$

Sendo “A” a área da seção transversal e “m” a massa distribuída ao longo do comprimento. Assim, dadas as condições de contorno das extremidades, podemos determinar para cada viga o equilíbrio do sistema e resolver o sistema de equações diferenciais resultantes, obtendo assim as frequências naturais e os modos de vibrar.

As condições de contorno aplicadas tornam alguns dos deslocamentos e esforços conhecidos nas extremidades da viga:

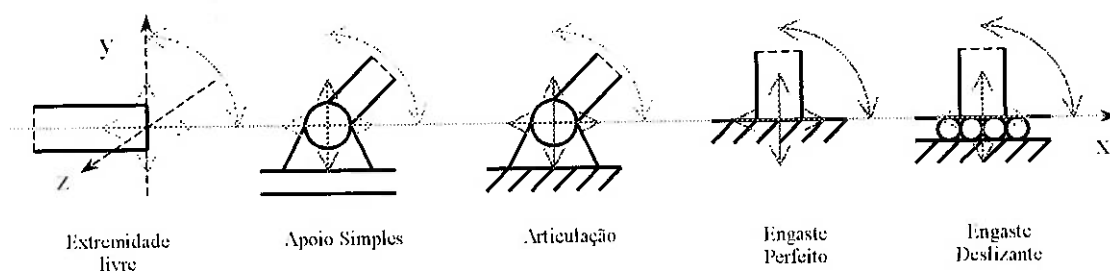


Ilustração 3, de contorno típicas

Para estas condições de apoio temos os seguintes esforços e deslocamentos conhecidos:

Tabela 1, Condições de contorno

Apoio	Flecha	Rotação	Desloc. Axial	Momento Fletor	Força Cortante	Tração
-------	--------	---------	---------------	----------------	----------------	--------

Extremid. Livre	-	-	-	zero	zero	zero
Apoio Simples	zero	-	-	zero	-	zero
Articul.	zero	-	zero	zero	-	-
Engaste Deslizante	-	zero	zero	-	zero	-
Engaste	zero	zero	zero	-	-	-

Substituindo estas condições nas equações diferenciais que descrevem os deslocamentos e esforços (1), (3), (4), (5), (9) e (10), obtemos um sistema possível e indeterminado nas variáveis “ C_n ”, “ A ” e “ B ”, da forma $[A].\{x\}=\{0\}$, sendo a matriz quadrada $[A]$ função somente de “ ω ” e o vetor $\{x\}$ os coeficientes de $\phi(x)$ e $\psi(x)$.

Desta maneira, o sistema só apresenta solução não trivial $\{0\}$ quando o determinante de $[A]$ for nulo. Neste caso, a variável “ ω ” encontra-se no argumento de funções trigonométricas e hiperbólicas, caracterizando um problema transcendental, ou seja, a variável não pode ser isolada. Porém, computacionalmente podemos considerar $\det([A])$ como uma função e, portanto, aplicar métodos numéricos para obter suas raízes. Cada raiz corresponde a uma frequência natural do sistema e para ela podemos voltar ao sistema $[A].\{x\}=\{0\}$ e obter os coeficientes correspondentes àquele modo de vibrar.

Ainda devemos considerar que ao respeitarmos as hipóteses levantadas no princípio da revisão garantimos a linearidade do modelo e assim os efeitos de flexão e tração são independentes. Isto posto, podemos para o caso de uma viga ou de um

sistema formado por vigas alinhadas ignorar o efeito da vibração axial, com magnitude infinitamente menor aos efeitos impostos pela flexão.

2.3. Massas e Molas concentradas nas extremidades

Podemos, a partir das derivadas temporais de segunda ordem de (1), (2) e (9), obter facilmente as acelerações na direção da flecha, rotacionais e axiais, ao longo da viga:

$$v(x,t) = \phi(x).sen(\omega t) \Leftrightarrow \ddot{v}(x,t) = \omega^2[\phi(x).sen(\omega t)] \quad (13)$$

$$\theta(x,t) = \phi'(x).sen(\omega t) \Leftrightarrow \ddot{\theta}(x,t) = \omega^2[\phi'(x).sen(\omega t)] \quad (14)$$

$$u(x,t) = \phi_t(x).sen(\omega t) \Leftrightarrow \ddot{u}(x,t) = \omega^2[\phi_t(x).sen(\omega t)] \quad (15)$$

Então, para estabelecer o equilíbrio de um vértice da viga, devem-se igualar os esforços solicitantes da estrutura ao produto da rigidez pelo deslocamento mais o produto da inércia pela aceleração (seja de rotação ou translação). Ou seja, para uma massa “Mt”, com inércia de rotação “Mr” concentradas num vértice da estrutura juntamente às molas de rotação e translação “Kr” e “Kt”, obtemos as seguintes expressões:

$$Q(x,t) = M_t.\ddot{v}(x,t) + K_r.v(x,t) \quad (16)$$

$$M(x,t) = M_r.\ddot{\theta}(x,t) + K_r.\theta(x,t) \quad (17)$$

$$T(x,t) = M_t.\ddot{u}(x,t) + K_{t,a}.u(x,t) \quad (18)$$

Sendo as equações (16), (17) e (18) referentes aos graus de liberdade de translação na direção da flecha, rotação e deslocamento axial estando livres. “Kt.A”

refere-se à rigidez atuante na direção axial da viga e “ K_t ” à rigidez na direção da flecha.

2.4. Estrutura de vigas

No caso de estruturas de vigas, sabe-se que é a frequência natural comum a todo sistema. Ainda, ao surgirem vigas em ângulo, os esforços e deslocamentos axiais de uma viga influenciam flexionalmente as vigas adjacentes, não podendo ser omitidos, neste caso, os efeitos da vibração axial.

Devemos utilizar agora novas condições de contorno que dizem respeito à compatibilidade dos esforços e deslocamentos nas extremidades das vigas que formam a estrutura. Ou seja, num vértice de conexão entre duas vigas devemos garantir a compatibilidade dos esforços de uma viga em relação às outras.

Uma estrutura de “ N ” vigas é formada por “ $N+1$ ” vértices que correspondem às uniões de duas vigas adjacentes da estrutura. Cada viga possui seu próprio modo de vibrar $\phi_i(x)$ e $\phi_{t,i}(x)$, que compõe o modo de vibrar da estrutura. Consideraremos os apoios, massas e molas concentradas sendo aplicados somente aos vértices. Assim, para estabelecer o equilíbrio da estrutura basta estabelecer o equilíbrio de cada vértice.

2.4.1. Compatibilidade de esforços e deslocamentos num vértice da estrutura

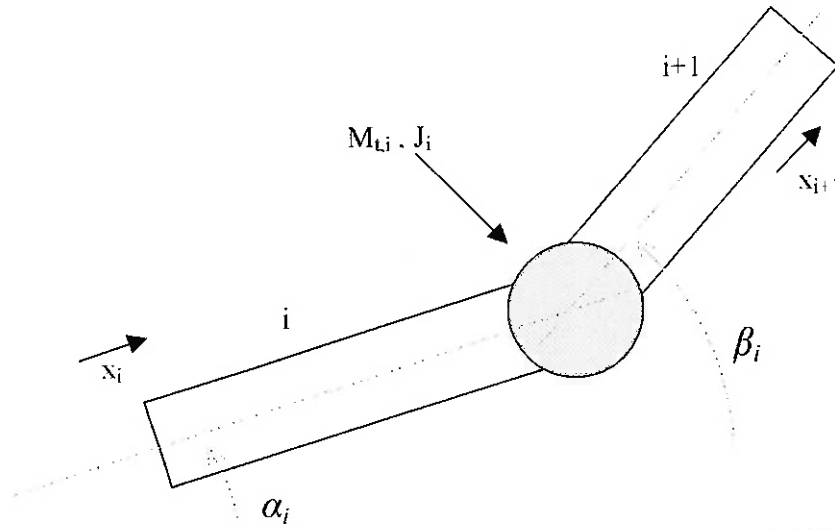


Ilustração 4, Vértice da estrutura

Agora serão apresentadas as equações de compatibilidade de deslocamentos e esforços, aplicáveis aos graus de liberdade presos de uma viga em relação a sua vizinha, obedecendo a seguinte notação simplificadora:

$$u_i(x,t) =_x u_i \quad (19)$$

Temos:

$$_L u_i \cdot \cos(\alpha_i) -_L v_i \cdot \sin(\alpha_i) =_0 u_{i+1} \cdot \cos(\alpha_{i+1}) -_0 v_{i+1} \cdot \sin(\alpha_{i+1}) \quad (20)$$

$$_L u_i \cdot \sin(\alpha_i) +_L v_i \cdot \cos(\alpha_i) =_0 u_{i+1} \cdot \sin(\alpha_{i+1}) +_0 v_{i+1} \cdot \cos(\alpha_{i+1}) \quad (21)$$

$$_L \theta_i =_L v'_i =_0 \theta_{i+1} =_0 v'_{i+1} \quad (22)$$

$$_L M_i -_0 M_{i+1} = J_{i+1,0} \ddot{\theta}_{i+1} \quad (23)$$

$$\begin{aligned}
& -_LT_i \cdot \cos(\alpha_i) - _LQ_i \cdot \sin(\alpha_i) + _0T_{i+1} \cdot \cos(\alpha_{i+1}) + _0Q_{i+1} \cdot \sin(\alpha_{i+1}) - \\
& -M_{i,i+1} \left[_0\ddot{u}_{i+1} \cdot \cos(\alpha_{i+1}) - _0\ddot{v}_{i+1} \cdot \sin(\alpha_{i+1}) \right]
\end{aligned} \tag{24}$$

$$\begin{aligned}
& -_LT_i \cdot \sin(\alpha_i) + _LQ_i \cdot \cos(\alpha_i) + _0T_{i+1} \cdot \sin(\alpha_{i+1}) - _0Q_{i+1} \cdot \cos(\alpha_{i+1}) - \\
& -M_{i,i+1} \left[_0\ddot{u}_{i+1} \cdot \sin(\alpha_{i+1}) - _0\ddot{v}_{i+1} \cdot \cos(\alpha_{i+1}) \right]
\end{aligned} \tag{25}$$

2.4.2 Obtendo o sistema de equações

Como podemos observar, para cada viga, temos seis condições de contorno necessárias para resolver as equações diferenciais de equilíbrio. Estas condições provêm das condições de apoio já descritas e da compatibilidade dos esforços e deslocamentos nos vértices da estrutura também ilustrados.

Agora, sabendo as condições de contorno, basta substituir as funções $\phi_i(x)$ e $\phi_{i+1}(x)$ e suas derivadas nas equações de equilíbrio para obter o sistema $[A] \cdot \{x\} = \{0\}$ e aplicar os métodos descritos no item seguinte para encontrar as frequências naturais que garantam que o sistema não tenha solução trivial e resolvê-lo, obtendo os coeficientes do modo de vibrar.

3. Exemplo de Implementação

Agora utilizaremos o equacionamento dos graus de liberdade desenvolvido acima para determinar as equações derivadas da aplicação de cada condição de contorno:

Extremidade livre:

-Rotação livre:

$$C_3 \cdot (\lambda \cdot \sinh(\lambda \xi_i) - f_{R,i} \cdot \cosh(\lambda \xi_i)) + C_4 \cdot (\lambda \cdot \cosh(\lambda \xi_i) - f_{R,i} \cdot \sinh(\lambda \xi_i)) = 0$$

-Translação livre:

$$C_1 \cdot (-\lambda^3 \cdot \cos(\lambda \xi_i) - f_{T,i} \cdot \sin(\lambda \xi_i)) + C_2 \cdot (\lambda^3 \cdot \sin(\lambda \xi_i) - f_{T,i} \cdot \cos(\lambda \xi_i)) + \\ C_3 \cdot (\lambda^3 \cdot \cosh(\lambda \xi_i) - f_{T,i} \cdot \sinh(\lambda \xi_i)) + C_4 \cdot (\lambda^3 \cdot \sinh(\lambda \xi_i) - f_{T,i} \cdot \cosh(\lambda \xi_i)) = 0$$

Engaste:

-Rotação presa:

$$C_1 \cdot \cos(\lambda \xi_i) - C_2 \cdot \sin(\lambda \xi_i) + C_3 \cdot \cosh(\lambda \xi_i) - C_4 \cdot \sinh(\lambda \xi_i) = 0$$

-Translação presa:

$$C_1 \cdot \sin(\lambda \xi_i) + C_2 \cdot \cos(\lambda \xi_i) + C_3 \cdot \sinh(\lambda \xi_i) + C_4 \cdot \cosh(\lambda \xi_i) = 0$$

Articulação:

-Rotação livre:

$$C_3 \cdot (\lambda \cdot \sinh(\lambda \xi_i) - f_{R,i} \cdot \cosh(\lambda \xi_i)) + C_4 \cdot (\lambda \cdot \cosh(\lambda \xi_i) - f_{R,i} \cdot \sinh(\lambda \xi_i)) = 0$$

- Translação presa:

$$C_1 \cdot \cos(\lambda \xi_i) - C_2 \cdot \sin(\lambda \xi_i) + C_3 \cdot \cosh(\lambda \xi_i) - C_4 \cdot \sinh(\lambda \xi_i) = 0$$

Engaste Deslizante:

-Rotação presa:

$$C_1 \cdot \cos(\lambda \xi_i) - C_2 \cdot \sin(\lambda \xi_i) + C_3 \cdot \cosh(\lambda \xi_i) - C_4 \cdot \sinh(\lambda \xi_i) = 0$$

-Translação livre:

$$C_1 \cdot (-\lambda^3 \cdot \cos(\lambda \xi_i) - f_{T,i} \cdot \sin(\lambda \xi_i)) + C_2 \cdot (\lambda^3 \cdot \sin(\lambda \xi_i) - f_{T,i} \cdot \cos(\lambda \xi_i)) + \\ C_3 \cdot (\lambda^3 \cdot \cosh(\lambda \xi_i) - f_{T,i} \cdot \sinh(\lambda \xi_i)) + C_4 \cdot (\lambda^3 \cdot \sinh(\lambda \xi_i) - f_{T,i} \cdot \cosh(\lambda \xi_i)) = 0$$

3.1 Montando o sistema de equações

Para uma viga delgada temos duas extremidades e, com as condições de apoio de cada uma, determinamos as quatro equações que nos dão as condições de contorno necessárias. Dados estas condições determinamos o seguinte sistema $[A] \cdot \{C\} = \{0\}$:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \otimes \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Para excluir a solução trivial devemos fazer $\det\{A\} = 0$, obtendo valores de λ associados aos respectivos modos de vibrar. Os valores de λ obtidos desta maneira (autovalores de A) fornecerão as frequências naturais de vibração do nosso sistema. ou seja, para quais frequências de excitação nosso sistema vibra segundo $v(x,t)$. E para um dado valor de λ podemos resolver o sistema (*) e obter os valores das constantes C_i de $\phi(x)$ e determinar a forma do modo de vibrar da viga numa determinada frequência.

Foi feita a organização das equações da seguinte maneira:

$$\begin{matrix} x = x_1 \\ x = x_2 \end{matrix} \begin{cases} \text{rotação} \\ \text{translação} \end{cases} \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \otimes \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Ou seja, as duas primeiras linhas relacionam as condições de contorno da extremidade esquerda da viga e as duas últimas relacionam as da extremidade da direita. Ainda posicionamos as equações de cada extremidade sendo a primeira equação relativa a rotação daquela extremidade e a segunda relativa a sua translação.

Fazendo este arranjo notamos que as equações de um mesmo grau de liberdade (translação ou rotação) tem forma muito semelhante e podem ser relacionados por uma variável $\Psi_{t,i}$ assumindo o valor 0 ou 1, para o grau de liberdade t livre ou preso na extremidade i .

Obtemos, fazendo $x_1 = 0$, e $x_2 = L$, os elementos da matriz $[A]$:

Primeira linha, correspondente à primeira equação (rotação na extremidade esquerda):

$$\begin{aligned} a_{1,1} &= f_{R,1} \\ a_{1,2} &= \lambda \cdot \Psi_{R,1} \\ a_{1,3} &= f_{R,1} \\ a_{1,4} &= -\lambda \cdot \Psi_{R,1} \end{aligned}$$

Segunda linha, correspondente à segunda equação (translação na extremidade esquerda):

$$\begin{aligned} a_{2,1} &= \lambda^3 \cdot \Psi_{T,1} \\ a_{2,2} &= f_{T,1} \\ a_{2,3} &= -\lambda^3 \cdot \Psi_{T,1} \\ a_{2,4} &= f_{T,1} \end{aligned}$$

Terceira linha, correspondente à terceira equação (rotação na extremidade direita):

$$\begin{aligned}a_{3,1} &= f_{R,2} \cdot \cos \lambda + \lambda \cdot \sin \lambda \cdot \Psi_{R,2} \\a_{3,2} &= -f_{R,2} \cdot \sin \lambda + \lambda \cdot \cos \lambda \cdot \Psi_{R,2} \\a_{3,3} &= f_{R,2} \cdot \cosh \lambda - \lambda \cdot \sinh \lambda \cdot \Psi_{R,2} \\a_{3,4} &= f_{R,2} \cdot \sinh \lambda - \lambda \cdot \cosh \lambda \cdot \Psi_{R,2}\end{aligned}$$

Quarta linha, correspondente à quarta equação: (translação na extremidade direita):

$$\begin{aligned}a_{4,1} &= f_{R,2} \cdot \sin \lambda + \lambda^3 \cdot \cos \lambda \cdot \Psi_{R,2} \\a_{4,2} &= f_{R,2} \cdot \cos \lambda - \lambda^3 \cdot \sin \lambda \cdot \Psi_{R,2} \\a_{4,3} &= f_{R,2} \cdot \sinh \lambda - \lambda^3 \cdot \cosh \lambda \cdot \Psi_{R,2} \\a_{4,4} &= f_{R,2} \cdot \cosh \lambda - \lambda^3 \cdot \sinh \lambda \cdot \Psi_{R,2}\end{aligned}$$

Os termos $\Psi_{t,i}$ são dados por:

$$\Psi_{R,i} \begin{cases} = 0, & \text{se a rotação no nó } i \text{ for presa.} \\ = 1, & \text{se a rotação no nó } i \text{ for livre.} \end{cases}$$

$$\Psi_{T,i} \begin{cases} = 0, & \text{se a translação no nó } i \text{ for presa.} \\ = 1, & \text{se a translação no nó } i \text{ for livre.} \end{cases}$$

3.2 Algoritmo de resolução

Para resolver o problema de determinar as frequências naturais e os modos de vibrar de uma determinada viga devemos:

- Obter os dados relativos à viga, parâmetros físicos e suas condições de apoio;
- Dados os parâmetros calcular os adimensionais $\mu_{R,i}$, $\mu_{T,i}$, $\alpha_{R,i}$ e $\alpha_{T,i}$;

- Resolver a equação $\det[A](\lambda) = 0$, variando λ à partir de zero com passo ε até ser encontradas N raízes;
- Para cada raiz λ_n ($n = 1, 2, \dots, N$) obtida desta maneira calcular a frequência natural relacionada a ela e resolver o sistema $[A] \cdot \{C\} = \{0\}$, obtendo os valores das constantes $C_{i,n}$.

3.3 Programação

Variáveis de entrada:

REAIS (dupla precisão):

L – Comprimento da viga (vão livre):

E – Módulo de elasticidade do material:

I – Momento de inércia à flexão:

A – Área da Secção Transversal:

Ro - Massa específica da viga:

Mr,i – Massa concentrada de rotação: (Extremidade $i = 1, 2$)

Mt,i – Massa concentrada de translação: (Extremidade $i = 1, 2$)

Kr,i – Mola concentrada de rotação: (Extremidade $i = 1, 2$)

Kt,i – Mola concentrada de translação: (Extremidade $i = 1, 2$)

LÓGICAS:

BT,i – Translação da extremidade i na direção y :

BR,i – Rotação da extremidade i na direção z :

Variáveis de saída:

REAIS (dupla precisão):

C1, C2, C3, C4 – Coeficientes da equação diferencial;

Raízes(N) – Array que armazena os autovalores de $[A]$ (relacionado com as Freqüências naturais):

3.4 Desenvolvimento da IGU (Interface gráfica com o usuário)

Pretendemos agora obter uma interface que permita ao usuário entrar com os dados do problema a ser resolvido facilmente (clareza no layout da interface gráfica), e organizar estes dados a fim de que o programa tenha acesso a eles com o menor esforço possível.

Para a “coleta” dos dados dispomos de janelas com ferramentas como Caixas de Texto (onde o usuário pode digitar textos e o programa pode utilizar diretamente estes dados, sejam eles numéricos ou não), Check-Boxes, onde o usuário seleciona algumas opções e botões que podem acionar funções e procedimentos.

Com o objetivo de facilitar o acesso aos dados do problema eles serão armazenados num array real de dupla precisão $DADOS[7]$, na seguinte ordem:

- [1] Comprimento da Viga;
- [2] Massa Específica do material;
- [3] Área da secção transversal da viga;
- [4] Momento de Inércia da secção;
- [5] Módulo de Elasticidade do Material;
- [6] Massa Concentrada na extremidade esquerda (rotação) – MR,1
- [7] Massa Concentrada na extremidade esquerda (translação) – MT,1
- [8] Massa Concentrada na extremidade direita (rotação) – MR,2
- [9] Massa Concentrada na extremidade direita (translação) – MT,2
- [10] Mola Concentrada na extremidade esquerda (rotação) – KR,1
- [11] Mola Concentrada na extremidade esquerda (translação) – KT,1
- [12] Mola Concentrada na extremidade direita (rotação) – KR,2
- [13] Mola Concentrada na extremidade direita (translação) – KT,2

A condição de contorno será armazenada através de duas variáveis BR_i e BT_i que estão relacionadas respectivamente à liberdade de rotação na extremidade i e à liberdade de translação do nó i . Estas variáveis assumem o valor 1 para um grau de liberdade preso e valor 0 para um grau de liberdade livre, conforme a tabela:

Tabela 2. Codificação das Condições de Apoio

Apoio	BR_i	BT_i
Articulação	0	1
Engaste	1	1
Extremidade livre	0	0
Engaste deslizante	1	0

São diversas as maneiras de se organizar uma janela. Foi desenvolvida a interface gráfica ilustrada na figura abaixo:

Ilustração 5. Interface Gráfica com o Usuário

Há uma figura meramente ilustrativa de uma barra que tem por objetivo tornar clara como está sendo fixada cada condição de contorno. A condição de contorno será determinada pelas variáveis BT_i e BR_i a partir de check-boxes e os

demais dados em caixas de texto. Isto Permite determinar as condições de contorno do problema e os seus parâmetros físicos e o programa pode, a cada modificação, destes dados realizar uma nova “coleta” e uma nova bateria de cálculos.

Na tabela abaixo estão descritos quais as soluções utilizadas para cada variável, uma vez que cada uma delas tem atributos que devem ser tratados de formas diferentes.

Tabela 3, Descrição das Variáveis

Dado / Variável	Recurso Utilizado	Justificativa
Condição de Contorno / BR.i e BT.i	Check Box	São 10 possíveis combinações das condições de contorno diferentes umas das outras. Cada opção é associada a valores de B. Podemos dessa forma também determinar as equações utilizadas em tempo de execução.
Comprimento / DADOS[1]	Caixa de Texto	Estes valores podem assumir magnitudes muito diferentes e não é possível determinar valores ou intervalos para utilizarmos Combo Boxes ou Barras de Rolagem. Ainda utilizando este recurso podemos, com uma condicional, limitar os valores que forem necessários. Caixas de texto também são vantajosas, pois podemos utilizar diretamente os valores numéricos nelas digitados sem ter de ser feito alguma alteração de sua natureza (como mudar de variável real para inteira).
Massa Espec. / DADOS[2]		
Secção / DADOS[3]		
M. Secção / DADOS[4]		
Módulo Elast. / DADOS[5]		
Massas e molas de concentradas nas extremidades / DADOS[6 até 13]		

Assim, a janela que será utilizada no programa tem o layout apresentado na figura 4 deste relatório e como podemos observar é de fácil compreensão e utilização tanto pelo usuário como pelo computador. Os dados são digitados e selecionados diretamente e podem ser acessados pelo programa sempre que for necessário.

Finalmente, obtidos os valores das constantes C_i podemos plotar o gráfico de $\phi(x)$ utilizando as rotinas gráficas disponíveis no Visual Studio. Para isto foi utilizada uma pequena janela gráfica cujas funções utilizadas foram as de LINETO, MOVETO, RECTANGLE e FLOODFILL.

Foi necessária a criação de uma escala que relacionasse as coordenadas do gráfico com as coordenadas da janela gráfica (output window), pois o eixo y da output window tem direção contrária ao do gráfico e a densidade de pixels na vertical é diferente da densidade de pixels na horizontal:

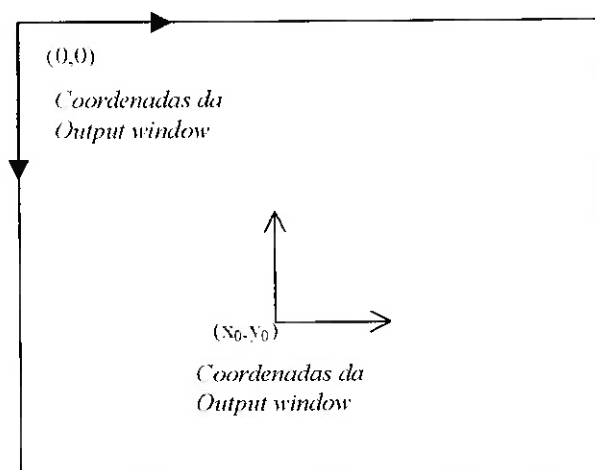


Ilustração 6, Ilustração da Output Window

Utilizou-se, portanto uma escala que relaciona as unidades do gráfico com os pixels da seguinte maneira: no eixo horizontal do gráfico o comprimento da viga L será representado por uma linha de comprimento 400 pixels e no eixo vertical do gráfico o comprimento máximo (que é unitário) será representado por 50 pixels. A origem do gráfico está na posição (50,125) da window outoput. A dimensão dessa escala foi estimada observando-se valores que resultassem numa boa visualização do gráfico.

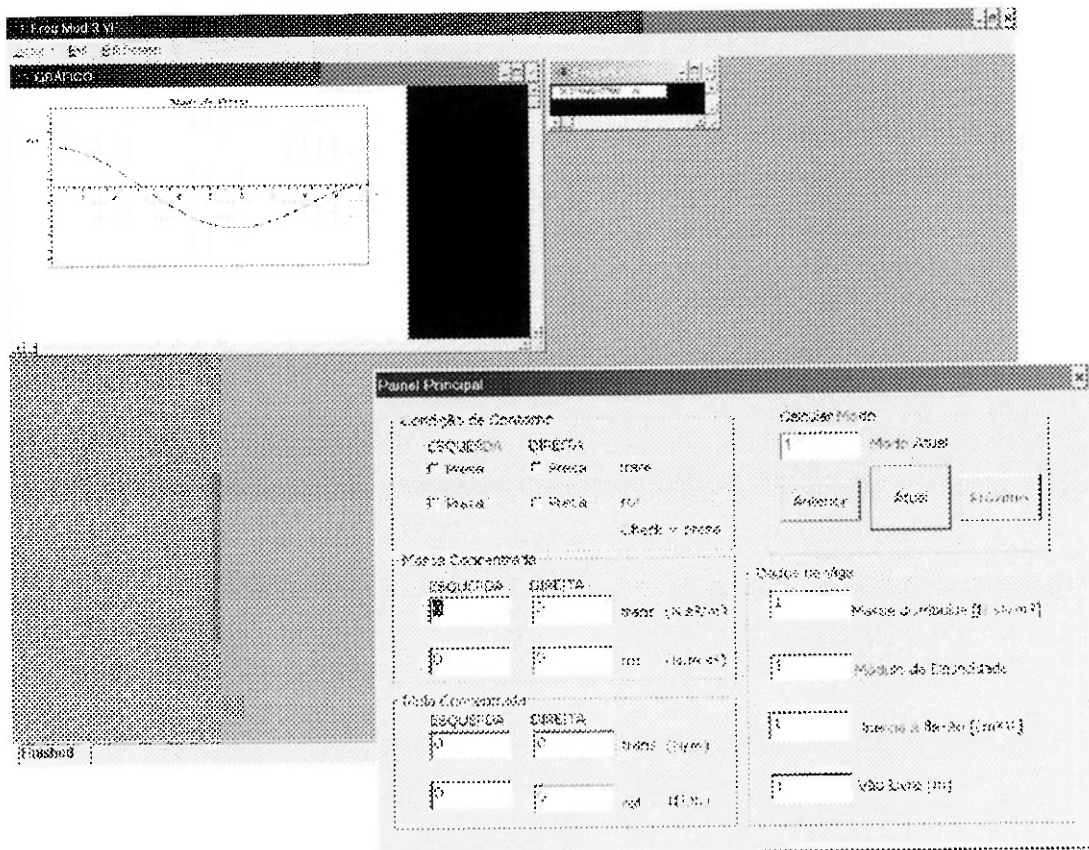


Ilustração 7, Screenshot do Programa em Execução

Primeiramente apresentaremos um screenshot do programa em execução para que seja possível uma apreciação de seu visual.

4. Resultados

A seguir compararemos os resultados obtidos no programa com os resultados encontrados na literatura. Para compara-los foi utilizada uma tabela (Thomson William T., pág. 277) que apresenta as frequências naturais para algumas condições de apoio típicas.

Comparam-se os valores λ_i , que dependem somente das condições de contorno, para os três primeiros modos de vibrar, obtiveram-se os seguintes resultados:

Tabela 4, Comparação de Resultados

	$(\lambda_1)^2$ Modo 1	$(\lambda_2)^2$ Modo 2	$(\lambda_3)^2$ Modo 3
Apoiada Simplesmente			
Valor Tabelado	9.87	39.5	88.9
Programa	9.8696	39.4784	88.8264
Em Balanço			
Valor Tabelado	3.52	22.4	61.7
Programa	3.5160	22.0344	61.6972
Duplamente Livre			
Valor Tabelado	22.4	61.7	121.0
Programa	22.3732	61.6728	120.9033
Duplamente Engastada			
Valor Tabelado	22.4	61.7	121.0
Programa	22.3732	61.6728	120.9033
Engastada-Articulada			
Valor Tabelado	15.4	50.0	104.0
Programa	15.4182	49.9648	104.2476
Articulada-Livre			
Valor Tabelado	15.4	50	104.0
Programa	15.4182	49.9648	104.2476

A tabela seguinte leva em consideração apenas a massa concentrada nas extremidades, que quando tende a um valor muito elevado, comporta-se como uma restrição de deslocamento ou rotação (consideraremos a inércia à rotação separadamente a inércia ao deslocamento). Assim, uma massa de deslocamento

infinita simula uma articulação (pois o deslocamento deste ponto fica praticamente zero) e uma “massa” de rotação infinita gera um engaste deslizante (pois a rotação do ponto fica praticamente zero). Portanto unindo-se uma massa infinita a uma inércia à rotação infinita obtém-se uma condição de contorno equivalente a um engaste. O caso da rigidez de molas de translação e rotação é análogo. Utilizaremos este método, pois não existem valores tabelados para frequências naturais em muitos casos que envolvem massas concentradas.

Tabela 5, dos resultados considerando-se as massas concentradas

	$(\lambda_1)^2$ Modo 1	$(\lambda_2)^2$ Modo 2	$(\lambda_3)^2$ Modo 3
Apoiada Simplesmente			
Valor Tabelado	9.87	39.5	88.9
Programa	9.8895	39.4983	88.8464
Em Balanço			
Valor Tabelado	3.52	22.4	61.7
Programa	3.5323	22.0560	61.6972
Duplamente Livre			
Valor Tabelado	22.4	61.7	121.0
Programa	-	-	-
Duplamente Engastada			
Valor Tabelado	22.4	61.7	121.0
Programa	22.3415	-	-
Engastada-Articulada			
Valor Tabelado	15.4	50.0	104.0
Programa	-	-	-
Articulada-Livre			
Valor Tabelado	15.4	50	104.0
Programa	15.7185	-	-

Este segundo método de comparação não permitiu a obtenção de todos os resultados, pois a inclusão de valores muito grandes de massas e molas causou problemas numéricos, impedindo o cálculo das frequências naturais.

Foram realizados mais 3 testes, utilizando fórmulas obtidas no manual “Formulas for Natural Frequency and Mode Shape” (pág.168-170). Neste manual foram obtidas fórmulas para o primeiro modo de vibrar dos seguintes casos: viga

Articulada-Livre com uma mola de rotação presa na articulação; viga Articulada-Livre, com uma mola de rotação presa na articulação e uma mola de translação presa à extremidade livre; e finalmente Articulada-Livre com uma mola de rotação presa na articulação e uma massa presa à extremidade livre.

Caso 1

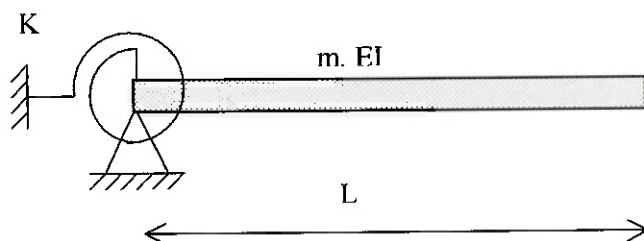


Figura 1 - Ilustração do Caso 1

Para este caso, a frequência para o primeiro modo de vibrar, com $m=6,35$ (massa distribuída, igual ao produto da área da seção transversal pela densidade do material), $L=7,5$, $EI=1500$ e $K=200$ é calculada em $0,425564$ rad/s. O programa nos dá o resultado de $0,42551$ rad/s. Verificamos um desvio na quinta casa decimal, que é o esperado devido à precisão do programa.

Caso 2

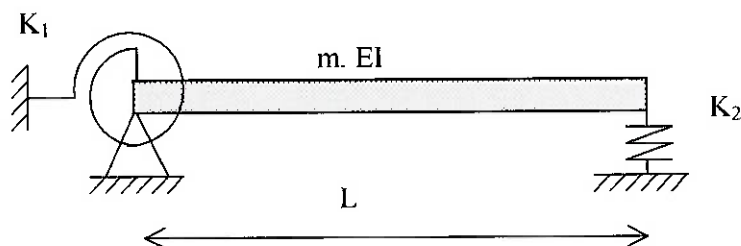


Figura 2 - Ilustração do Caso 2

Para este caso, a frequência para o primeiro modo de vibrar, com $m=6.35$ (massa distribuída, igual ao produto da área da seção transversal pela densidade do material), $L=7.5$, $EI=1500$, $K_1=200$ e $K_2=355.5$ é calculada em 2,64003716 rad/s. O programa nos dá o resultado de 2,640031 rad/s. Verificamos novamente um desvio na quinta casa decimal, que é o esperado devido à precisão do programa.

Caso 3

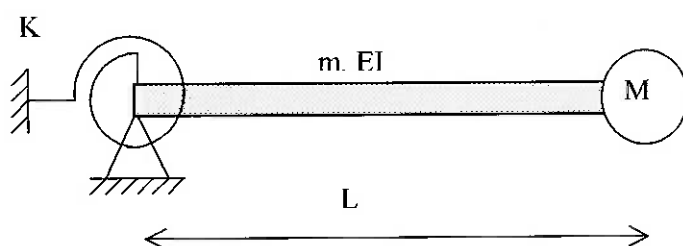


Figura 3 - Ilustração do Caso 3

Para este caso, a frequência para o primeiro modo de vibrar, com $m=6.35$ (massa distribuída, igual ao produto da área da seção transversal pela densidade do material), $L=7.5$, $EI=1500$, $K=200$ e $M=0.4763$ é calculada em 0,41883 rad/s. O programa nos dá o resultado de 0,41887 rad/s. Verificamos novamente um desvio na quinta casa decimal, que é o esperado devido à precisão do programa.

Também devido a erros numéricos de truncamento os valores obtidos pelo programa são apenas uma aproximação do valor real, com uma precisão de 10^{-5} , e ainda, não foi possível calcular frequências naturais além do 4o modo, pois os valores das frequências aumentam a ponto de causar problemas de overflow.

Assim restringimos na interface gráfica o cálculo de frequências naturais até o 4o modo, e a alocação de massas e molas concentradas também. Criando estas limitações não permitimos que o usuário obtenha resultados sem significado.

CONCLUSÕES

O objetivo do trabalho foi alcançado com sucesso. Toda base teórica necessária para o desenvolvimento do programa foi estudada e foi gerada uma primeira versão funcional e validada, cumprindo a proposta inicial.

No desenvolvimento deste programa notou-se que a grande maioria dos problemas que foram enfrentados está relacionada a graves problemas de condicionamento numérico. Isto ocorre por dois principais motivos: Matrizes de ordem elevada e a discrepância entre os valores dos elementos da matriz $A(\omega)$. Operações com matrizes de ordem elevada geram problemas numéricos devido ao fato de o computador utilizar um número finito de casas decimais resultando em problemas de truncamento que se tornam mais graves com o aumento da dimensão do problema.

Porém o mau condicionamento numérico é um problema muito mais grave, que acaba inviabilizando a aplicação de diversos algoritmos de cálculo numérico e até, em alguns casos, a própria solução do problema. Isso ocorre devido à presença de elementos derivados de funções trigonométricas e funções hiperbólicas.

Pode-se destacar, como uma sugestão para futuros estudos, a necessidade de um trabalho extensivo na melhoria da interface gráfica para permitir o cálculo de estruturas genéricas e a utilização de uma linguagem e algoritmos mais robustos para permitir o cálculo de estruturas mais complexas. Podemos ainda ponderar acerca da extensão ao caso tridimensional onde surgem também os efeitos da vibração torcional.

REFERÊNCIAS BIBLIOGRÁFICAS

Nunes, C. A. D., APOSTILA Fundamentos da Dinâmica das Estruturas Oceânicas. p 1-28.

Thomson, William T., Theory of vibration with applications, Edição 4. ed., Cheltenham: Stanley Thornes Ltd, 1998.

Humes, A.F.P. de C., Melo, I.S.H. de, Yoshida, L.K., Martins, W.T., NOÇÕES DE CÁLCULO NUMÉRICO, McGraw-Hill do Brasil, 1984.

Meirovitch, L., Elements of Vibration Analysis, Second Edition, McGraw-Hill International Editions., 1986.

Martins, J. F., Influência da Inércia de Rotação e da Força Cortante nas Frequências Naturais e na Resposta Dinâmica de Estruturas de Barras. 1998. 159 p. Tese (doutorando) – Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, 1998.

ANEXO 1 – CODIGO FONTE DO PROGRAMA PROPOSTO

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                     !!
!! FREQMOD3.f90                       !!!!
!!                                     !!
!! VERSÃO 3 COM CÁLCULO DOS CASOS DE MASSAS E MOLAS      !!
!!             CONCENTRADAS NAS EXTREMIDADES             !!
!!                                     !!
!!                                     !!
!! Entradas: L comprimento da Viga                       !!
!!             I Momento de Inércia da seção Transversal !!
!!             E Modulo de elasticidade                 !!
!!             A Área da ST                             !!
!!             rô massa especifica                      !!
!!             Kr Coef. de mola de Rotação Concentrada  !!
!!             Kt Coef. de mola de Translação Concentrada !!
!!             Mt Massa Concentrada: Translação         !!
!!             Mr MASSa Concentrada: Rotação            !!
!!                                     !!
!!             Modo a ser visualizado                   !!
!!                                     !!
!!             Condições de apoio (Liberdade da         !!
!!             rotação e, ou, translação das            !!
!!             extremidades).                           !!
!!                                     !!
!! Saídas:                                              !!
!!             Frequências Naturais ( Junto ao gráfico ) !!
!!                                     !!
!!             Modo de Vibrar (Gráfico plotado na janela !!
!!             do programa)                             !!
!!                                     !!
!!!!                                                !!
!! !                                                !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! MENU
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
LOGICAL(4) FUNCTION InitialSettings()
!!
USE DFLIB
!!
external dodialog
!!
LOGICAL(4) 14
!!
!14 = appendmenuqq(1, $MENUENABLED, 'Dialogo'C, DoDialog)
!!
14 = appendmenuqq(2, $MENUENABLED, 'Exit'C, WINEXIT)
!!
!14 = appendmenuqq(3, $MENUENABLED, 'Full Screen'C,
WINFULLSCREEN)!!
InitialSettings = 14
!!

```

```

return
!!
end
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!

```

```

program Freqmod3
use dflogm
use DFLIB
USE DFNLIS
implicit none
external Dodialog
include 'resource.fd'
type (dialog) dlgDummy

call DoDialog( dlgDummy, 0, 0 )
  do while (.true.)
    end do
end program

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! DoDialog                                     !
!!                                             !
!! Inicializa e abre a janela do aplicativo   !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

recursive subroutine DoDialog( dlgParent, id, callbacktype )
use dflogm
implicit none
type (dialog) dlgParent
integer id
integer callbacktype
include 'resource.fd'
integer retint
logical retlog
type (dialog) dlg
external ProcedurePrincipal
integer local_id, local_callbacktype
type (dialog) local_dlg
local_id = id
local_callbacktype = callbacktype
local_dlg = dlgParent
if ( .not. DlgInit( idd_painel, dlg ) ) then
write (*,*) "error: resource not found"
else
! Inicia as Caixas de Texto
retlog = DlgSet( dlg, EDIT_BT1, .false. )
retlog = DlgSet( dlg, EDIT_BT2, .false. )
retlog = DlgSet( dlg, EDIT_BT3, .false. )
retlog = DlgSet( dlg, EDIT_BT4, .false. )
retlog = DlgSet( dlg, EDIT_MT1, "0" )
retlog = DlgSet( dlg, EDIT_MT2, "0" )
retlog = DlgSet( dlg, EDIT_MR1, "0" )
retlog = DlgSet( dlg, EDIT_MR2, "0" )
retlog = DlgSet( dlg, EDIT_KT1, "0" )
retlog = DlgSet( dlg, EDIT_KT2, "0" )

```

```

retlog = DlgSet( dlg, EDIT_KR1, "0" )
retlog = DlgSet( dlg, EDIT_KR2, "0" )
retlog = DlgSet( dlg, EDIT_M, "1" )
retlog = DlgSet( dlg, idc_Nmodo, "1" )
retlog = DlgSet( dlg, EDIT_E, "1" )
retlog = DlgSet( dlg, EDIT_I, "1" )
retlog = DlgSet( dlg, EDIT_L, "1" )

! Associa a ProcedurePrincipal ao evento Change nas caixas de
texto
!call ProcedurePrincipal( dlg, EDIT_Nmodo, dlg_change)
!retlog = DlgSetSub( dlg, EDIT_NMOD0, ProcedurePrincipal )

retlog = DlgSetSub( dlg, IDC_BUTTON_calcular1,
ProcedurePrincipal )
retlog = DlgSetSub( dlg, IDC_BUTTON_calcular2,
ProcedurePrincipal )
retlog = DlgSetSub( dlg, IDC_BUTTON_calcular3,
ProcedurePrincipal )

retint = DlgModal( dlg )

call DlgUninit( dlg )

end if
end subroutine DoDialog

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                     !
!   Subrotinas do Programa - FreqMod3   !
!                                     !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

subroutine ProcedurePrincipal( dlg, id, callbacktype )
  use dflogm
  use dflib
  implicit none
  type (dialog) dlg
!   **** DECLARAÇÃO DAS VARIÁVEIS

! Build as QuickWin or Standard Graphics

  INTEGER(2) result !variáveis usadas na composição do gráfico
  INTEGER(2) style,dummy,status
  TYPE (xycoord) xy
  INTEGER(4) i,j,grstat
  TYPE (xycoord) xys
  TYPE (qwinfo) winfo
  REAL K,z
!VARIÁVEIS DO DIALOG
  integer id ! controle que chamou a função, ex: Botao_1
  integer callbacktype
  include 'resource.fd'
  integer retint

```

```

character*30 text

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Variáveis usadas na solução!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
integer NMODO

!Parâmetros concentrados
REAL*8 MT(20),MR(20)!massas
REAL*8 KR(20),KT(20)!molas

! DADOS da viga: (1)L;(2)MASSA DISTRIBUIDA;(3)E;(4)I !
REAL*8 DADOS(40)
LOGICAL BT(20),BR(20) ! APOIO NAS EXTREMIDADES

!matrizes e vetores usados para calcular raizes
REAL*8 x0,x1,y0,y1,h0,h1,h2,raizes(150)
REAL*8 det ! função det(A) que calcula o determinante da matriz
A
!Fatores calculados à partir dos parâmetros concentrados
REAL*8 Fr,Ft
REAL*8 AlfaT(20),AlfaR(20),MiT(20),MiR(20)!fatores diretamente
utilizados para montar o sistema Ac=B

!Variáveis para solução do sistema Ac=B
REAL*8 A(4,4),B(4),C(4)
REAL*8 tol,AUX,Cmax,X
REAL*8 AMAX
REAL*8 fatorT(20),FatorR(20)
INTEGER m,n
INTEGER ordem(40)
INTEGER err,modosaida

logical retlog
integer local_callbacktype

local_callbacktype = callbacktype

! **** ARMAZENAMENTO DOS DADOS DIGITADOS NO DIALOG **** !
retlog =DlgGet( dlg, IDC_NMODO, text )
read (text, *, iostat=retint) NMODO

Select Case(ID) ! verifica qual botão foi utilizado para chamar
a rotina: Avançar, atual ou voltar

case (IDC_BUTTON_calcular1)
nmodo=nmodo-1
write(text,*) nmodo
retlog = dlgSet (dlg, IDC_nmodo, text)
if (nmodo.lt.1) then
nmodo=1
! CALL BEEPQQ(500,200)
! CALL BEEPQQ(900,200)
! CALL BEEPQQ(700,200)
! CALL BEEPQQ(300,200)
CALL BEEPQQ(400,400)

```

```

retlog = dlgSet (dlg, idc_nmodo, "1")
end if

CASE (IDC_BUTTON_calcular3)
nmodo=nmodo+1
write(text,*) nmodo
retlog = dlgSet (dlg, IDC_nmodo, text)
if (nmodo.gt.4) then
nmodo=1
CALL BEEPQQ(1000,200)
retlog = dlgSet (dlg, idc_nmodo, "1")
end if

end select

retlog = DlgGet( dlg, EDIT_BT1, BT(1) )
retlog = DlgGet( dlg, EDIT_BT2, BT(2) )
retlog = DlgGet( dlg, EDIT_BT3, BR(1) )
retlog = DlgGet( dlg, EDIT_BT4, BR(2) )
retlog = DlgGet( dlg, EDIT_MT1, text )
read (text, *, iostat=retint) MT(1)
retlog = DlgGet( dlg, EDIT_MT2, text )
read (text, *, iostat=retint) MT(2)
retlog = DlgGet( dlg, EDIT_MR1, text )
read (text, *, iostat=retint) MR(1)
retlog = DlgGet( dlg, EDIT_MR2, text )
read (text, *, iostat=retint) MR(2)
retlog = DlgGet( dlg, EDIT_KT1, text )
read (text, *, iostat=retint) KT(1)
retlog = DlgGet( dlg, EDIT_KT2, text )
read (text, *, iostat=retint) KT(2)
retlog = DlgGet( dlg, EDIT_KR1, text )
read (text, *, iostat=retint) KR(1)
retlog = DlgGet( dlg, EDIT_KR2, text )
read (text, *, iostat=retint) KR(2)
retlog = DlgGet( dlg, EDIT_L, text )
read (text, *, iostat=retint) DADOS(1)
retlog = DlgGet( dlg, EDIT_M, text )
read (text, *, iostat=retint) DADOS(2)
retlog = DlgGet( dlg, EDIT_I, text )
read (text, *, iostat=retint) DADOS(3)
retlog = DlgGet( dlg, EDIT_E, text )
read (text, *, iostat=retint) DADOS(4)

!
!
!
! **** PARÂMETROS BÁSICOS - Primeiros Parâmetros que são
constantes ao longo do programa **** !
do i=1,2
  AlfaT(i)=( (dados(1)**3) * KT(i) ) / ( dados(3)*dados(4) )

  AlfaR(i)=( dados(1) * KR(i) ) / ( dados(3)*dados(4) )

  MiT(i)=MT(i)/( dados(1) * dados(2) )

  MiR(i)=MR(i)/( (dados(1)**3) * dados(2) )

```



```

end do

!
!
!
! **** CALCULO DAS RAIZES **** !
! OPEN(1,FILE='malha.csv',FORM='FORMATTED',
MODE='READWRITE',STATUS='UNKNOWN') !ARQUIVO DE SAIDA
! REWIND 1
n=5
x0=2e-6
x1=3e-6
i=1

321 DO WHILE (i .Le. n)
!FAZ A PRIMEIRA APROXIMAÇÃO DO INTERVALO QUE CONTÉM A RAIZ
y0=Det(X0,alfaT,alfaR,miT,miR,Bt,br)/(x-1e-6)
!write(1,*)
x0,', ',y0,', ',(x0/DADOS(1))*2*DSQRT(DADOS(4)*DADOS(3)/DADOS(2))
y1=Det(X1,alfaT,alfaR,miT,miR,Bt,br)/(x-1e-6)
if (y0*y1 .LT. 0 .and. y0*y1.ne.'nan') then
H0=X0
H1=X1
H2=(X0+X1)/2

DO WHILE (ABS(H1-H2) .GE. 1E-5)

!APROXIMA A RAIZ ATÉ UMA PRECISÃO RELATIVA DE 1E-30
PELO MÉTODO DAS SECANTES
H2=H1-Det(H1,alfaT,alfaR,miT,miR,Bt,br)*((H1-H0)/
(Det(H1,alfaT,alfaR,miT,miR,Bt,br)-
Det(H0,alfaT,alfaR,miT,miR,Bt,br)))
H0=H1
H1=H2

END DO

Raizes(i)=H2
i=i+1

end if
x0=x1
x1=x1+0.001
END DO

!
!
!
!
! **** CALCULA E IMPRIME MODO DE VIBRAR (X,Y) NO ARQUIVO DE SAÍDA
**** !
modosaida=NMODO
X=raizes(modosaida)

!montando a matriz A(4,4)
do i=1,2

```

```

        IF (BT(I)) then
            FATORT(I)=0
        ELSE
            FATORT(I)=1
        end if
        if (bR(i)) then
            fatorR(i)=0
        else
            fatorR(i)=1
        end if
    end do

    A(1,1)= Fr(bR(1),1,X,AlfaR(1),MiR(1))
    A(1,2)= X*fatorR(1)
    A(1,3)= Fr(bR(1),1,X,AlfaR(1),MiR(1))
    A(1,4)= -X*fatorR(1)

    A(2,1)= X**3*fatorT(1)
    A(2,2)= -Ft(bt(1),1,X,AlfaT(1),MiT(1))
    A(2,3)= -X**3*fatorT(1)
    A(2,4)= -Ft(bt(1),1,X,AlfaT(1),MiT(1))

    A(3,1)= -Fr(bR(2),2,X,AlfaR(2),MiR(2))*DCOS(X)
    +X*DSIN(X)*fatorR(2)
    A(3,2)= Fr(bR(2),2,X,AlfaR(2),MiR(2))*DSIN(X)
    +X*DCOS(X)*fatorR(2)
    A(3,3)= -Fr(bR(2),2,X,AlfaR(2),MiR(2))*DCOSH(X)-
    X*DSINH(X)*fatorR(2)
    A(3,4)= -Fr(bR(2),2,X,AlfaR(2),MiR(2))*DSINH(X)-
    X*DCOSH(X)*fatorR(2)

    A(4,1)= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DSIN(X)
    +X**3*DCOS(X)*fatorT(2)
    A(4,2)= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DCOS(X) -
    X**3*DSIN(X)*fatorT(2)
    A(4,3)= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DSINH(X)-
    X**3*DCOSH(X)*fatorT(2)
    A(4,4)= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DCOSH(X)-
    X**3*DSINH(X)*fatorT(2)

    do I=1,n
        b(i)=0
    end do
    n=4

!determinação da tolerância!
    AMAX=0
    DO I=1,N
        DO J=1,N
            IF (AMAX.LT.ABS(A(I,J))) AMAX=ABS(A(I,J))
        END DO
    END DO
    TOL=AMAX/1E6

! inicializacao da ordem das incógnitas
    do i=1,n
        ORDEM(i)=i
    end do

```

```

! triangularizacao
  do k=1,n-1
    do i=k+1,n

      if (abs(a(k,k)).LE.tol) then
!----- permutacao para remover pivo nulo
!      de colunas:
        err=1
        m=0
        do while (abs(a(k,k)).LE.tol.AND.(k+m).LT.n)
          m=m+1
          if (a(k,k+m).NE.tol) then
            do j=1,n
              aux=a(j,k+m)
              a(j,k+m)=a(j,k)
              a(j,k)=aux
            end do
            AUX=ORDEM(k+m)
            ORDEM(k+m)=ORDEM(k)
            ORDEM(k)=AUX
            err=0
          end if
        end do

!      de linhas, se necessario:
        if (err.NE.0) then
          m=0
          Do while (abs(a(k,k)).LE.tol)
            m=m+1;
            if (m+k.GT.n) then
              stop 'nao tem solucao'
            end if
            if (abs(a(k+m,k)).GE.tol) then
              do j=1,n
                x=a(k+m,j)
                a(k+m,j)=a(k,j)
                a(k,j)=x
              end do
              AUX=b(k+m)
              b(k+m)=b(k)
              b(k)=AUX
              err=0
            end if
          end do
        end if
      end if

      if (abs(a(k,k)).GE.tol) then
        AUX=-a(i,k)/a(k,k);
        do j=1,n
          a(i,j)=a(i,j)+AUX*a(k,j)
        end do
        b(i)=b(i)+AUX*b(k)
      end if

    end do
  end do

```

!retrosubstituicao

```
C(ORDEM(n))=1
DO i=n-1,1,-1

    AUX=0
    DO m=i+1,n
        AUX=AUX+A(i,m)*C(ORDEM(m))
    END DO
    C(ORDEM(i))=(b(i)-aux)/A(i,i)

END DO
```

!Normalização

```
amax=0
do i=1,4
    if (abs(amax).le.abs(c(i))) amax=c(i)
end do
do i=1,4
    c(i)=c(i)/abs(amax)
end do
```

```
!
!
!
!
!***** IMPRIME O MODO DE VIBRAR *****!
!! **** UTILIZAÇÃO DA PARTE GRÁFICA **** !!
```

```
!!!!!!!!!!!!!!!!!!!!!!
!FREQUÊNCIA NATURAL!!
!!!!!!!!!!!!!!!!!!!!!!
```

```
close(3)
OPEN(3, file='user', title='Frequência Natural')
winfo=TYPE = QWIN$MAX
result = SETWSIZEQQ(QWIN$FRAMEWINDOW, winfo)
!Adjust child window parameters
winfo.H = 2
winfo.W = 22
winfo.y = 0
winfo.x = 76
winfo.TYPE = QWIN$SET
status = SETWSIZEQQ(3, winfo)
! Escreve a frequência natural
result = setcolorrgb(#c0c0c0)
result = FLOODFILL (1, 1, 1)

result = setcolorrgb(#ff0000)
dummy = RECTANGLE( $GFillInterior , int2(0), int2(0), int2(150),
int2(20) )

result = setcolorrgb(#ffffff)
```

```

dummy = RECTANGLE( $GFILLINTERIOR , int2(3), int2(3), int2(147),
int2(17) )

result = INITIALIZEFONTS()
result = SETFONT('t'Arial'h12w5pvib')
grstat=SETCOLORrgb(#000000)

write(text,*) (
raizes(modosaida)**2*SQRT(DADOS(3)*DADOS(4)/DADOS(2)) / DADOS(1)**2
),'rad/s'

CALL MOVETO(INT2(3),INT2(4),xys)
CALL OUTGTEXT(text)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!GRÁFICO!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
close(4)
OPEN(4, file='user', title='GRÁFICO')
!Maximize Window
winfo+TYPE = QWIN$MAX
result = SETWSIZEQQ(QWIN$FRAMEWINDOW, winfo)
!Adjust child window parameters
winfo.H = 16
winfo.W = 73
winfo.y = 0
winfo.x = 0
winfo.TYPE = QWIN$SET
status = SETWSIZEQQ(4, winfo)

!desenha o fundo cinza e o retângulo branco
result = setcolorrgb(#C0c0c0)

result = FLOODFILL (1, 1, 1)
result = setcolorrgb(#ffffff)
dummy = RECTANGLE( $GFILLINTERIOR , int2(10), int2(10),
int2(500), int2(250) )

!gráfico animado

do j=0,1000,+10

result = setcolorrgb(#000000)
CALL MOVETO(INT2(50), INT2(30-5), xy) !EIXO Y
status = LINETO(INT2(50), INT2(230-5))
CALL MOVETO(INT2(50), INT2(125), xy) !EIXO X
status = LINETO(INT2(450), INT2(125))

RESULT= SETCOLORrgb(#dd0000)
CALL MOVETO(INT2(50),
INT2(125+50*(C(1)*DSIN(0/(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+
C(2)*DCOS(0/(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(3)*DSINH(0/
(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(4)*DCOSH(0/(400/dados(1)
))*RAIZES(MODOSAIDA)/dados(1))), xy)
DO x=0,400
z=3.14*j/100

K=125+sin(z)*50*(C(1)*DSIN(X/(400/dados(1))*RAIZES(MODOSAIDA)/dados(

```

```

1)) + C(2) * DCOS(X / (400 / dados(1))) * RAIZES(MODOSAIDA) / dados(1)) + C(3) * DSIN
H(X / (400 / dados(1))) * RAIZES(MODOSAIDA) / dados(1)) + C(4) * DCOSH(X / (400 / dad
os(1))) * RAIZES(MODOSAIDA) / dados(1)))
    STATUS = LINETO(INT2(50+x), INT2(K))
    END DO

    result = setcolorrrgb(#0)
    CALL MOVETO(INT2(50), INT2(30-5), xy) !EIXO Y
    status = LINETO(INT2(50), INT2(230-5))
    CALL MOVETO(INT2(50), INT2(125), xy) !EIXO X
    status = LINETO(INT2(450), INT2(125))

    result = setcolorrrgb(#ffffff)
    dummy = RECTANGLE( $GFillInterior , int2(00), int2(00),
int2(500), int2(500) )

end do

!BORDA DO GRAFICO + grade tracejada

    result = setcolorrrgb(#c0c0c0)
    call setlinestyle(#AA3C )
    do i=1,20
    call moveto(int2(50+20*i),int2(25),xys)
    status = lineto(int2(50+20*i),int2(225))
    end do
    do i=1,10
    call moveto(int2(50),int2(25+20*i),xys)
    status = lineto(int2(450),int2(25+20*i))
    end do

    CALL SETLINESTYLE(#ffff)

    result = setcolorrrgb(#0)
    dummy = RECTANGLE( $GBORDER , int2(50), int2(25), int2(450),
int2(225))

!desenha os eixos
    grstat=setcolorrrgb(#0)
    do i=3,23 !TRACEJADO NO EIXO Y
    CALL MOVETO(INT2(50), INT2(i*10-5), xy)
    status = LINETO(INT2(45), INT2(i*10-5))
    end do

    do i=0,40 !TRACEJADO DO EIXO X
    CALL MOVETO(INT2(50+10*i), INT2(130), xy)
    status = LINETO(INT2(50+10*i), INT2(125))
    end do

    result = INITIALIZEFONTS() ! ESCRIVE OS VALORES NA ABSCISSA
    result = SETFONT('t' 'Arial' 'h12w4pvib')
    grstat=SETCOLORrgb(#0)
    do i=0,40
    CALL MOVETO(INT2(55+10*i), INT2(133), xy)

```

```

write(text,*) 1+I/4
IF(MOD(I,4).EQ.0 .AND. I.LT.34) THEN
CALL OUTGTEXT(text)
END IF
end do

CALL MOVETO(INT2(50), INT2(30-5), xy) !EIXO Y
status = LINETO(INT2(50), INT2(230-5))
CALL MOVETO(INT2(50), INT2(125), xy) !EIXO X
status = LINETO(INT2(450), INT2(125))

! desenha a função
RESULT= SETCOLORrgb(#0000f0)
CALL MOVETO(INT2(50),
INT2(125+50*(C(1)*DSIN(0/(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+
C(2)*DCOS(0/(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(3)*DSINH(0/
(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(4)*DCOSH(0/(400/dados(1)
))*RAIZES(MODOSAIDA)/dados(1)))), xy)
DO x=0,400

K=125+50*(C(1)*DSIN(X/(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(2)
)*DCOS(X/(400/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(3)*DSINH(X/(40
0/dados(1))*RAIZES(MODOSAIDA)/dados(1))+C(4)*DCOSH(X/(400/dados(1))*
RAIZES(MODOSAIDA)/dados(1)))
STATUS = LINETO(INT2(50+x), INT2(K))
END DO

! escreve os textos
result = INITIALIZEFONTS()
result = SETFONT('t' 'Times New Roman' 'h12w7pvib')
CALL MOVETO(INT2(200),INT2(12),xys)
grstat=SETCOLORrgb(#0)
CALL OUTGTEXT('Modo de Vibrar')

result = INITIALIZEFONTS()
result = SETFONT('t' 'verdana' 'h10w5pvib')
CALL MOVETO(INT2(20),INT2(60),xys)
grstat=SETCOLORrgb(#0)
CALL OUTGTEXT('v(x)')

result = INITIALIZEFONTS()
result = SETFONT('t' 'verdana' 'h10w5pvib')
CALL MOVETO(INT2(460),INT2(130),xys)
grstat=SETCOLORrgb(#0)
CALL OUTGTEXT('x')

```

666 end subroutine ProcedurePrincipal

```

!***** FIM DO PROGRAMA PRINCIPAL *****!
!*****!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!

```

```

!      Funções do Programa - FreqMod3      !
!                                          !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!Função que calcula o valor da função Ft(i)
Real(8) Function Ft(Bt,i,lbd,alfaT,miT)
Implicit None
Real*8 lbd,alfaT,miT
logical Bt
Integer i
    if (bt) then
        Ft=1
    else
        Ft=(alfaT-(lbd**4)*miT)
    end if
Return

end function

!Função que calcula o valor da função Fr(i)
Real(8) Function Fr(Br,i,lbd,alfaR,miR)
Implicit None
Real*8 lbd,alfaR,miR
LOGICAL Br
integer i
    if (Br) then
        Fr=1
    else
        Fr=(alfaR-(lbd**4)*miR)
    end if
Return

end function

!Função que calcula o determinante da matriz
Real(8) Function Det(X,alfaT,alfaR,miT,miR,Bt,br)
Implicit None
EXTERNAL Fr,Ft
Integer z,fatorT(2),fatorR(2)
Logical bt(20),Br(20)
Real*8
X,a,b,c,d,e,f,g,h,i,j,l,m,n,o,p,q,r,s,t,u,alfaR(2),alfaT(2),miR(2),m
it(2),Fr,Ft

    do i=1,2
        IF(BT(I)) then
            FATORT(I)=0
        ELSE
            FATORT(I)=1
        end if
        if(bR(i)) then
            fatorR(i)=0
        else
            fatorR(i)=1
        end if
    end do

    a= Fr(bR(1),1,X,AlfaR(1),MiR(1))

```



```

b= X*fatorR(1)
c= Fr(bR(1),1,X,AlfaR(1),MiR(1))
d= -X*fatorR(1)

e= X**3*fatorT(1)
f= -Ft(bt(1),1,X,AlfaT(1),MiT(1))
g= -X**3*fatorT(1)
h= -Ft(bt(1),1,X,AlfaT(1),MiT(1))

i= -Fr(bR(2),2,X,AlfaR(2),MiR(2))*DCOS(X) +X*DSIN(X)*fatorR(2)
j= Fr(bR(2),2,X,AlfaR(2),MiR(2))*DSIN(X) +X*DCOS(X)*fatorR(2)
l= -Fr(bR(2),2,X,AlfaR(2),MiR(2))*DCOSH(X)-X*DSINH(X)*fatorR(2)
m= -Fr(bR(2),2,X,AlfaR(2),MiR(2))*DSINH(X)-X*DCOSH(X)*fatorR(2)

n= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DSIN(X)
+X**3*DCOS(X)*fatorT(2)
o= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DCOS(X) -
X**3*DSIN(X)*fatorT(2)
p= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DSINH(X)-
X**3*DCOSH(X)*fatorT(2)
q= Ft(bt(2),2,X,AlfaT(2),MiT(2))*DCOSH(X) -
X**3*DSINH(X)*fatorT(2)

!calcula o determinante / modificar para o método da
triangularização!
r=a*(f*(l*q-p*m) - g*(j*q-o*m) + h*(j*p-o*l))
s=b*(e*(l*q-p*m) - g*(i*q-n*m) + h*(i*p-n*l))
t=c*(e*(j*q-o*m) - f*(i*q-n*m) + h*(i*o-n*j))
u=d*(e*(j*p-o*l) - f*(i*p-n*l) + g*(i*o-n*j))
det=r-s+t-u
Return

end function

```

Fernando D'Agosto Toledo

Vibração de Vigas Alinhadas no Plano

Graduação
EPUSP
2005