



Universidade Federal do Rio de Janeiro
Escola Politécnica
Programa de Projeto de Estruturas

André Vítor Bolanho Rocha de Oliveira

**ANÁLISE NUMÉRICA DE FUNÇÕES DE IMPEDÂNCIA EM SOLO
HETEROGÊNEO**



André Vítor Bolanho Rocha de Oliveira

ANÁLISE NUMÉRICA DE FUNÇÕES DE IMPEDÂNCIA EM SOLO HETEROGÊNEO

Dissertação de Mestrado apresentada ao Programa de Projeto de Estruturas, Escola Politécnica, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Projeto de Estruturas.

Orientadores:

Sergio Hampshire de Carvalho Santos

Marco Felipe Fialho Santos

Rio de Janeiro

2025

Oliveira, André Vitor Bolanho Rocha de

Análise Numérica de Funções de Impedância em Solo Heterogêneo/ André Vitor Bolanho Rocha de Oliveira. – Rio de Janeiro: UFRJ/ ESCOLA POLITÉCNICA - 2024.

V; 159p. il; 29,7cm.

Dissertação (Mestrado em Projeto de Estruturas) – Universidade Federal do Rio de Janeiro, Escola Politécnica, Programa de Projeto de Estruturas, Rio de Janeiro, 2024.

Orientadores: Prof. Sergio Hampshire de Carvalho Santos, D.Sc.
Prof. Marco Felipe Fialho Santos, D.Sc.

Referências Bibliográficas: p. 103-105

1.Fundações de Máquinas. 2. Funções de Impedância. 3. Modelo Numérico. 4. Solo Heterogêneo. 5. Análise Dinâmica.

I. Santos, Sergio Hampshire de Carvalho; Santos, Marco Felipe Fialho II. Universidade Federal do Rio de Janeiro, UFRJ, Escola Politécnica, Programa de Projetos de Estruturas. III. Título



ANÁLISE NUMÉRICA DE FUNÇÕES DE IMPEDÂNCIA EM SOLO HETEROGÊNEO

André Vítor Bolanho Rocha de Oliveira

Orientadores:

Sergio Hampshire de Carvalho Santos

Marco Felipe Fialho Santos

Dissertação de Mestrado apresentada ao Programa de Projeto de Estruturas, Escola Politécnica, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Projeto de Estruturas.

Aprovada pela Banca:

Prof. Sergio Hampshire de Carvalho Santos, D.Sc, UFRJ

Prof. Marco Felipe Fialho Santos, D.Sc, UFRJ

Prof. Silvio de Souza Lima, D.Sc, UFRJ

Prof. José Antonio Fontes Santiago, D.Sc, UFRJ

Rio de Janeiro

2025

AGRADECIMENTOS

Gostaria de expressar meus mais sinceros agradecimentos ao Professor Sergio Hampshire de Carvalho Santos, cuja orientação ao longo dos anos foi essencial para o desenvolvimento desta pesquisa. Sua confiança depositada, bem como sua presença e disponibilidade, foram fundamentais para que eu pudesse enfrentar os desafios e alcançar os objetivos propostos. Agradeço também ao Professor Marco Felipe Fialho Santos, pela coorientação dedicada, cujas valiosas contribuições enriqueceram este estudo, além do tempo e da atenção dedicados ao longo do percurso.

À Banca Avaliadora, Professor Silvio de Souza Lima e Professor José Antonio Fontes Santiago, manifesto minha gratidão pela disponibilidade, pelas considerações construtivas e pelo tempo investido.

Agradeço igualmente a todos os professores do Programa de Projeto de Estruturas, pelo conhecimento transmitido e pela atenção sempre que requisitada, enriquecendo a experiência acadêmica com orientações que ampliaram minha compreensão sobre o tema.

À Universidade Federal do Rio de Janeiro e à Escola Politécnica, deixo meu reconhecimento por seu papel imprescindível no fomento à educação e na disseminação de um conhecimento técnico de excelência.

Agradeço também à coordenação e à diretoria da Casagrande Engenharia, que reconheceram a importância do mestrado, me oferecendo flexibilidade para conciliar trabalho e estudos, e por apresentarem desafios profissionais que me incentivaram a continuar aprimorando meu conhecimento em engenharia estrutural.

Aos meus amigos, sou grato pelo suporte constante, pelas discussões produtivas e pelas palavras de encorajamento. Agradeço por estarem presentes, mesmo à distância, contribuindo para tornar a jornada mais leve e gratificante, e por sempre ajudarem a transformar os dias de estudo e trabalho em momentos mais descontraídos.

À Polyanna, expresso meu agradecimento por seu carinho, paciência, atenção e apoio desde o início desta jornada na Engenharia Civil. Estendo esses agradecimentos a toda a sua família, que me acolheu no Rio de Janeiro e sempre demonstrou apoio e compreensão.

Por fim, dedico meu mais profundo reconhecimento aos meus familiares, especialmente aos meus pais, Neusa e Paulo. Obrigado por tudo. Pela criação, pelo incentivo contínuo, pela confiança e pelo suporte incondicional. Mesmo quando distantes, sempre vibraram e torceram por mim. Este trabalho não teria sido possível sem tudo que vocês me proporcionaram.

A todos, meu mais sincero e profundo agradecimento.

RESUMO

OLIVEIRA, André Vítor Bolanho Rocha de. **Análise Numérica de Funções de Impedância em Solo Heterogêneo**. Dissertação (Mestrado) – Programa de Projeto de Estruturas, Escola Politécnica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2025

Muitas dúvidas ainda existem a respeito do comportamento da interação solo-estrutura no projeto de fundações de máquinas sujeitas a carregamentos dinâmicos. Correntemente assumem-se diversas hipóteses simplificadoras. Uma das dúvidas diz respeito à obtenção das funções de impedância, de rigidez e amortecimento do solo. Essas funções são aqui analisadas com a variação da frequência excitadora para o problema do solo heterogêneo, utilizando-se de formulação que pode ser aplicada a múltiplas camadas. No presente trabalho, os ensaios numéricos consistem em situações de duas camadas, variando as propriedades da segunda camada. É aplicada uma formulação matematicamente robusta e é mostrado como em modelos de interface empregá-la a uma fundação superficial de formato retangular. Em seguida é apresentada uma comparação dos valores de rigidez e de amortecimento obtidos com os de outras formulações clássicas consolidadas de Richart (1970) e de Wolf (1994), apresentadas neste trabalho. A análise dos resultados permitiu concluir que o amortecimento e a rigidez seguem distribuições distintas e geometricamente variáveis na interface solo-fundação com comportamento sensível às diferentes frequências excitadoras e propriedades das camadas de solo.

Palavras- chave: Fundações de Máquinas, Funções de Impedância, Modelo Numérico, Solo Heterogêneo, Análise Dinâmica.

ABSTRACT

OLIVEIRA, André Vítor Bolanho Rocha de. **Análise Numérica de Funções de Impedância em Solo Heterogêneo**. Thesis (Master's) – Programa de Projeto de Estruturas, Escola Politécnica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2025

There are still many questions regarding the behavior of soil-structure interaction in the design of machine foundations subjected to dynamic loads. Various simplifying assumptions are commonly made. One of the questions concerns the determination of soil impedance functions, stiffness, and damping. These functions are herein analyzed with the variation of the excitation frequency for the problem of heterogeneous soil, using a formulation that can be applied to multiple layers. In this work, numerical tests consist of two-layer cases, varying the properties of the second layer. A mathematically robust formulation is applied, and it is shown how to apply the proposed formulation in interface models on a rectangular surface foundation. A comparison of the stiffness and damping values obtained with those from other established formulations by Richart (1970) and Wolf (1994) presented in this work is then shown. The analysis of the results led to the conclusion that damping and stiffness exhibit distinct and variable distributions along the soil-foundation interface, with behavior sensitive to different excitation frequencies and soil layer properties.

Keywords: Machine Foundations, Impedance Functions, Numerical Model, Heterogeneous Soil, Dynamic Analysis.

SUMÁRIO

Capítulo 1	1
1.1. Motivação	2
1.2. Objetivo	2
1.3. Materiais e Métodos	3
1.4. Apresentação e Estrutura do Trabalho	3
1.5. Revisão Bibliográfica	4
Capítulo 2	7
2. FUNDAMENTAÇÃO TEÓRICA	7
2.1. Propagação de Ondas	7
2.1.1. Ondas de Volume	7
2.1.2. Ondas de Superfície	9
2.1.3. Propagação Unidimensional de Ondas em uma Barra	10
2.1.4. Condições de Contorno Na Barra Horizontal	15
2.1.5. Propagação Tridimensional de Ondas no Meio Contínuo	17
2.2. A Equação Diferencial do Movimento	20
2.3. Avaliação das Propriedades do Solo	24
2.4. Coeficientes de Rigidez e Amortecimento para Fundações Diretas	28
2.4.1. Consideração de Blocos Rígidos e Blocos Flexíveis	30
2.4.2. Fundações sobre Solo Homogêneo	30
2.4.3. Fundações sobre Solo com Estratificação Horizontal	38
2.4.4. Fundações sobre Solo Heterogêneo	40
2.4.5. A consideração da Hipótese de Winkler	43
2.4.6. A consideração da Matriz de Rigidez Condensada do Solo	44
Capítulo 3	46
3. DESENVOLVIMENTO DO PROBLEMA	46
3.1. Aplicação da Metodologia	46
3.2. Modelagem do Solo para o Problema Estudado	48
3.3. Implementação	49
3.4. Algoritmo de análise	58
Capítulo 4	61
4. EXEMPLOS DE APLICAÇÃO	61
4.1. Validação da função da influência de flexibilidade	61
4.2. Distribuição das funções de impedância na interface solo-fundação	63
4.2.1. Caso 1: Solo Homogêneo	64
4.2.2. Caso 2: Solo Heterogêneo	69

4.2.3.	<i>Caso 3: Base em Rocha</i>	73
4.3.	Variação das funções de impedância para diferentes frequências	78
4.3.1.	<i>Caso 1: Solo Homogêneo</i>	79
4.3.2.	<i>Caso 2: Solo Heterogêneo</i>	80
4.3.3.	<i>Caso 3: Base em Rocha</i>	81
4.3.4.	<i>Comparação e Resumo dos Casos</i>	81
4.4.	Estudo da amplificação dinâmica	83
4.5.	Aplicação prática	89
Capítulo 5		99
5.	CONCLUSÕES	99
5.1.	SUGESTÕES PARA TRABALHOS FUTUROS	102
	REFERÊNCIAS BIBLIOGRÁFICAS	103
	APÊNDICES	106
I.	CÓDIGO DESENVOLVIDO EM C++	106

LISTA DE FIGURAS

Figura 2.1 – Ondas P, primárias ou de compressão (Adaptado de <i>Encyclopædia Britannica</i> , 2021).	9
Figura 2.2 – Ondas S, secundárias ou de cisalhamento (Adaptado de <i>Encyclopædia Britannica</i>).....	9
Figura 2.3 – Ondas de <i>Love</i> , ou ondas Q (Adaptado de <i>Encyclopædia Britannica</i> , 2021).	10
Figura 2.4 – Ondas de <i>Rayleigh</i> , ou ondas R (Adaptado de <i>Encyclopædia Britannica</i> , 2021).	10
Figura 2.5 – Deformações e tensões no sentido longitudinal de uma barra horizontal (SANTOS, 2024).	11
Figura 2.6 – Comprimento de onda (SANTOS, 2024).....	12
Figura 2.7 – Direção da propagação de ondas (SANTOS,2024).	13
Figura 2.8 – Deslocamentos das ondas para $t=0$ e $t=\Delta t$ (SANTOS,2024).....	13
Figura 2.9 – Propagação de onda em direção a extremidade engastada (SANTOS,2024).....	14
Figura 2.10 – Propagação de onda em direção a extremidade livre (SANTOS,2024).	14
Figura 2.11 – Reflexão e refração de ondas na interface de meios desiguais (SANTOS,2024).....	15
Figura 2.12 – Barra horizontal infinita (SANTOS, 2024).	15
Figura 2.13 – Barra horizontal com amortecedor para representação do meio infinito (SANTOS, 2024).	16
Figura 2.14 – Tensões em elemento prismático infinitesimal em meio elástico infinito (RICHART, 1970).	17
Figura 2.15 – Ensaio <i>Crosshole</i> (adaptado de ABMS/ ABEF, 2002).....	26
Figura 2.16 – Representação da interação Máquina x Fundação x Solo (SANTOS, 2024).	29
Figura 2.17 – Modelo numérico de um grau de liberdade para Blocos Rígidos (SANTOS, 2024).	29
Figura 2.18 – Orientação dos eixos conforme formulação de fundações circulares de Richart et al. (1970).....	31
Figura 2.19 – Orientação dos eixos conforme formulação de fundações retangulares de Wolf (1994).	36
Figura 2.20 – Orientação dos eixos conforme formulação de fundações de forma qualquer de Wolf (1994).....	37

Figura 2.21 – Fundação Circular em Estrato sobre Camada Rígida (GAZETAS,1983).	39
Figura 2.22 – Vista em corte transversal do modelo estudado por Luco (1974).	41
Figura 2.23 – Vista superior do modelo axissimétrico estudado por Luco (1974).	41
Figura 2.24 – Fundação Circular assente em Estrado sobre Camada infinita (GAZETAS,1983).	42
Figura 2.25 – Pressões de contato em placa rígida sobre argila (ABMS/ABEF, 2002).	43
Figura 3.1 – Malha do modelo de estudo.	47
Figura 3.2 – Área de influência de um ponto nodal.	55
Figura 3.3 – Montagem da matriz de flexibilidade.	56
Figura 4.1 – Parte real da função de influência de flexibilidade vertical FWW .	62
Figura 4.2 – Parte imaginária da função de influência de flexibilidade vertical FWW ...	62
Figura 4.3 – Parte real da amplitude de deslocamento vertical $W(r)$.	63
Figura 4.4 – Parte imaginária da amplitude de deslocamento vertical $W(r)$	63
Figura 4.5 – Distribuição dos Coeficientes de Rigidez para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=0$	65
Figura 4.6 – Variação da Rigidez para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=0$	65
Figura 4.7 – Distribuição dos Coeficientes de Amortecimento para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=0$	66
Figura 4.8 – Variação do Amortecimento para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=0$	66
Figura 4.9 – Distribuição dos Coeficientes de Rigidez para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=3,0$.	67
Figura 4.10 – Variação da Rigidez para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=3,0$.	67
Figura 4.11 – Distribuição dos Coeficientes de Amortecimento para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=3,0$	68
Figura 4.12 – Variação do Amortecimento para Caso 1 – Solo Homogêneo ($G2/G1 = 1$) e com $\alpha0=3,0$.	68
Figura 4.13 – Distribuição dos Coeficientes de Rigidez para Caso 2 – Solo Heterogêneo ($G2/G1 = 1,838$) e com $\alpha0=0$.	70
Figura 4.14 – Variação da Rigidez para Caso 2 – Solo Heterogêneo ($G2/G1 = 1,838$) e com $\alpha0=0$	70

Figura 4.15 – Distribuição dos Coeficientes de Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $\alpha_0=0$.	71
Figura 4.16 – Variação do Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $\alpha_0=0$.	71
Figura 4.17 – Distribuição dos Coeficientes de Rigidez para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $\alpha_0=3,0$.	72
Figura 4.18 – Variação da Rigidez para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $\alpha_0=3,0$.	72
Figura 4.19 – Distribuição dos Coeficientes de Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $\alpha_0=3,0$.	73
Figura 4.20 – Variação do Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $\alpha_0=3,0$.	73
Figura 4.21 – Distribuição dos Coeficientes de Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=0$.	74
Figura 4.22 – Variação da Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=0$.	75
Figura 4.23 – Distribuição dos Coeficientes de Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=0$.	75
Figura 4.24 – Variação do Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=0$.	76
Figura 4.25 – Distribuição dos Coeficientes de Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=3,0$.	76
Figura 4.26 – Variação da Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=3,0$.	77
Figura 4.27 – Distribuição dos Coeficientes de Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=3,0$.	77
Figura 4.28 – Variação do Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $\alpha_0=3,0$.	78
Figura 4.29 – Comparação entre os valores de rigidez adimensionais encontrados e os de referência (LUCO, 1974) – Caso 1: Solo Homogêneo.	79
Figura 4.30 – Comparação entre os valores de amortecimento adimensionais encontrados e os de referência (LUCO, 1974) – Caso 1: Solo Homogêneo.	79
Figura 4.31 – Comparação entre os valores de rigidez adimensionais encontrados e os de referência (LUCO, 1974) – Caso 2: Solo Heterogêneo.	80
Figura 4.32 – Comparação entre os valores de amortecimento adimensional encontrados e os de referência (LUCO, 1974) – Caso 2: Solo Heterogêneo.	80

Figura 4.33 – Comparação entre os valores de rigidez adimensionais encontrados e os de referência (LUCO, 1974) – Caso 3: Base em Rocha.....	81
Figura 4.34 – Comparação entre os valores de amortecimento adimensional encontrados e os de referência (LUCO, 1974) – Caso 3: Base em Rocha.	81
Figura 4.35 – Coeficientes de Rigidez Adimensionais Verticais.....	83
Figura 4.36 – Coeficientes de Amortecimento Adimensionais Verticais.	83
Figura 4.37 – Amplificação dinâmica para fundação leve ($m = 1,0$).	84
Figura 4.38 – Amplificação dinâmica para fundação pesada ($m = 5,0$).	85
Figura 4.39 – Diferenças absolutas da amplificação dinâmica de cada configuração de espessura das camadas em relação ao caso do solo homogêneo para fundação leve.	86
Figura 4.40 – Diferenças absolutas da amplificação dinâmica de cada configuração de espessura das camadas em relação ao caso do solo homogêneo para fundação pesada.....	86
Figura 4.41 – Amplificação dinâmica para fundação leve ($m = 1,0$).	88
Figura 4.42 – Amplificação dinâmica para fundação pesada ($m = 5,0$).	88
Figura 4.43 – Geometria do bloco de fundação estudado.....	90
Figura 4.44 – Tela principal para solo homogêneo (BlockSolver).	91
Figura 4.45 – Entrada de dados de geometria (BlockSolver).	91
Figura 4.46 – Entrada de dados de massas (BlockSolver).	92
Figura 4.47 – Entrada de dados do solo homogêneo (BlockSolver).	92
Figura 4.48 – Entrada de dados de cargas dinâmicas (BlockSolver).	93
Figura 4.49 – Entrada de dados da frequência de operação (BlockSolver).	93
Figura 4.50 – Deslocamentos verticais para o solo homogêneo (BlockSolver).	94
Figura 4.51 – Resultados para a frequência da máquina no solo homogêneo (BlockSolver).	94
Figura 4.52 – Tela principal para solo heterogêneo (BlockSolver).	96
Figura 4.53 – Entrada de dados equivalentes para solo heterogêneo (BlockSolver). ..	96
Figura 4.54 – Deslocamentos verticais para o solo heterogêneo (BlockSolver).	97
Figura 4.55 – Resultados para a frequência da máquina no solo heterogêneo (BlockSolver).	97
Figura 4.56 – Variação do deslocamento vertical com a frequência do carregamento.	98

LISTA DE TABELAS

Tabela 2.1 – Valores típicos de ρ , ν e ε (SANTOS,2024).	28
Tabela 4.1 – Valores calculados numericamente e valores teóricos tradicionais.....	82
Tabela 4.2 – Valores da amplificação dinâmica para a variação da espessura das camadas do solo.	87
Tabela 4.3 – Valores da amplificação dinâmica para a variação das propriedades do subespaço.	89

NOTAÇÃO

a_0 – Frequência adimensional;

a – Metade do lado de maior dimensão de uma fundação retangular;

b – Metade do lado de menor dimensão de uma fundação retangular;

B_z – Fração de massa na direção vertical;

B_x – Fração de massa na direção horizontal;

B_ψ – Fração de massa rotacional;

B_θ – Fração de massa torcional;

c – Coeficiente adimensional de amortecimento;

C – Matriz de amortecimento do sistema;

C_c – Amortecimento crítico;

D – Amplificação dinâmica;

e – Índice de vazios;

$F(t)$ – Vetor de forças aplicadas;

$F_{ww}(k)$ – Função de flexibilidade vertical;

$f_{v_{m,n}}$ – Valor da flexibilidade para o termo m,n da matriz de flexibilidade vertical;

G – Módulo de elasticidade transversal;

G^* – Módulo de deformação transversal complexo;

G_f – Matriz de flexibilidade dinâmica;

$g(s)$ – Função de influência de flexibilidade do semiespaço estratificado;

i – Unidade imaginária;

I_ψ – Momento de inércia rotacional;

I_θ – Momento de inércia torcional;

J_0 – Função de Bessel de primeiro tipo de ordem 0;

J_1 – Função de Bessel de primeiro tipo de ordem 1;

k_o – Número de onda;

k – Coeficiente adimensional de rigidez;
 K – Matriz de rigidez do sistema;
 K_z – Coeficiente de rigidez na direção vertical;
 K_x – Coeficiente de rigidez na direção horizontal X;
 K_y – Coeficiente de rigidez na direção horizontal Y;
 K_ψ – Coeficiente de rigidez rotacional;
 K_{xx} – Coeficiente de rigidez na direção rotacional XX;
 K_{yy} – Coeficiente de rigidez na direção rotacional YY;
 K_θ – Coeficiente de rigidez torcional;
 K_{zz} – Coeficiente de rigidez na direção torcional ZZ;
 K_0 – Coeficiente de empuxo no repouso;
 M – Matriz de massa do sistema;
 m – Massa da fundação;
 $\bar{m}$ – Massa adimensional;
 p – Intensidade de carga aplicada;
 r – Raio equivalente de um elemento da malha;
 r_0 – Amplitude constante na direção vertical;
 S_z – Matriz de rigidez dinâmica;
 T – Matriz de rigidez dinâmica da interface;
 $u(t)$ – Vetor de deslocamentos;
 $\dot{u}(t)$ – Vetor de velocidades;
 $\ddot{u}(t)$ – Vetor de acelerações;
 u_{st} – Deslocamento estático;
 V_v – Volume de vazios;

V_s – Volume de sólidos;

V_p – Velocidade de onda de principal ou de compressão;

V_s – Velocidade de onda secundária ou de cisalhamento;

$w(r)$ – Amplitude de deslocamento vertical;

ν – Coeficiente de Poisson;

λ – Comprimento de onda;

δ_v – Deslocamento vertical distante de um ponto de aplicação de uma carga vertical P;

δ_h – Deslocamento horizontal distante de um ponto de aplicação de uma carga vertical P;

ε – Fração de amortecimento crítico;

ε_z – Fração de amortecimento crítico na direção vertical;

ε_x – Fração de amortecimento crítico na direção horizontal;

ε_ψ – Fração de amortecimento crítico na direção rotacional;

ε_θ – Fração de amortecimento crítico na direção torcional;

$\bar{\omega}$ – Frequência circular própria;

ω – Frequência de excitação;

ρ – Massa específica;

σ_1 – Tensão vertical efetiva, para o solo homogêneo;

σ_2 – Tensão horizontal efetiva, igual a $K_0 \cdot \sigma_1$;

σ_3 – Tensão horizontal efetiva, igual a $K_0 \cdot \sigma_1$;

σ_m – Tensão octaédrica média;

ξ_{hist} – Fração de amortecimento histerético do material.

Capítulo 1

1. INTRODUÇÃO

Problemas relativos à dinâmica das estruturas têm tomado maior importância, e diversos projetos industriais não se utilizam dessas análises de maneira adequada, simplificando as cargas dinâmicas em estáticas equivalentes.

Muitos dos problemas que concernem as análises dinâmicas realizadas nas estruturas, em especial no projeto de fundações de máquinas, não possuem respostas conhecidas ou consolidadas, para algumas de suas questões. Assim, o comportamento dessas estruturas nem sempre é bem representado, sendo sujeito a diversas simplificações.

O projeto das fundações de máquinas tem como objetivo limitar as amplitudes de oscilação a valores admissíveis, de modo a evitar danos estruturais ou desconforto para profissionais que trabalhem próximo às máquinas, e garantir o funcionamento e integridade do equipamento. Essa limitação da amplitude, está relacionada ao afastamento das frequências oscilatórias e excitadoras das frequências naturais.

Comparando-se as fundações de máquinas aos demais problemas envolvendo fundações que não estão submetidas aos carregamentos dinâmicos, ou seja, problemas estáticos, ou demais problemas geotécnicos e de análise das estruturas de modo geral, verifica-se que é um assunto muito complexo e que ainda há muito a ser estudado e a evoluir.

Uma das dúvidas mais pertinentes no que diz respeito a estes projetos é sobre o comportamento e a distribuição das funções de impedância, ou seja, se o amortecimento segue a mesma distribuição geométrica na base da fundação e o mesmo comportamento que a rigidez, com a variação de frequências excitadoras.

No trabalho de OLIVEIRA (2021), esse problema foi avaliado considerando a hipótese simplificadora de que o solo pode ser modelado como homogêneo. Entretanto, para uma caracterização mais precisa do solo, é necessário considerar as diversas propriedades e camadas subjacentes e sua influência no cálculo das funções de impedância.

Esta pesquisa objetiva estudar a modelagem da interface entre solo e estrutura em fundações de máquinas, a distribuição e o comportamento das funções de impedância, amortecimento e rigidez, considerando o solo como heterogêneo em um semiespaço, por meio de modelos numéricos, em uma malha discretizada, utilizando uma formulação matematicamente robusta.

Nesta formulação, o problema é resolvido aplicando-se a metodologia de Wolf (1985), que propõe equações básicas para a modelagem da matriz de rigidez dinâmica do solo, de acordo com o cálculo de funções de influência de flexibilidade. É considerada a discretização de molas e amortecedores na base, centradas nos elementos de uma malha e obtém-se uma distribuição não-uniforme das funções de impedância.

A adequação das respostas obtidas é comprovada pela comparação com valores de rigidez e de amortecimento obtidos com formulações clássicas utilizadas em projeto, como as de Richart et al. (1970) e de Wolf (1994).

1.1. Motivação

Motivado pela suposição adotada em diversos trabalhos, como o de Albuquerque (2015), em que se observa que em uma formulação teórica seria possível modelar amortecedores seguindo uma variação similar à adotada para os coeficientes de rigidez, surgiu a principal indagação para essa pesquisa: se é correto assumir que os amortecedores, podem ter um comportamento similar ao de rigidez, na sua distribuição na interação solo-estrutura de uma fundação, ou se as distribuições são distintas e tais aproximações não são válidas.

O trabalho de Oliveira (2021) estudou a distribuição geométrica e o comportamento das funções de impedância considerando um modelo simplificado, onde o solo é tratado como homogêneo. No entanto, o estudo da distribuição de molas e amortecedores, visando representar de forma mais realista a interação solo-estrutura para um solo com variação de propriedades entre camadas de diferentes espessuras, ainda é necessário. Este estudo é relevante para verificar se as conclusões e hipóteses do modelo homogêneo se mantêm válidas em solos com configurações mais complexas.

Segundo Wolf (1985), as análises estruturais dinâmicas possuem métodos consolidados para determinar as tensões e deslocamentos aos quais a estrutura está sujeita. No entanto, a estrutura irá interagir com o solo no entorno da mesma e apenas a análise da estrutura não é suficiente. A correta modelagem do solo é de elevada importância.

1.2. Objetivo

Esta pesquisa tem como objetivo estudar a modelagem da interface entre solo e estrutura em fundações de máquinas, com foco na análise da distribuição das funções

de impedância e comportamento de um modelo de fundação composto por elementos de área circular. O estudo examina as funções de impedância, avaliando a relação entre amortecimento e rigidez, considerando o solo como um meio heterogêneo, estratificado em camadas horizontais. Para isso, se utiliza um modelo numérico com malha discretizada e se aplica o método dos elementos de interface para capturar com precisão a interação entre solo e estrutura em diferentes condições de estratificação e propriedades do solo.

Com esta formulação, o problema é resolvido aplicando-se a metodologia de Wolf (1985), que propõe equações básicas para modelagem da matriz de rigidez dinâmica do solo, de acordo com o cálculo de funções de influência de flexibilidade.

É considerada a organização de molas e amortecedores na base centradas em cada elemento da malha definida em cada modelo, de modo a validar uma distribuição não-uniforme das funções de impedância, em solo heterogêneo, para estes modelos.

1.3. Materiais e Métodos

A análise comparativa entre o comportamento das funções de impedância foi realizada por meio de código desenvolvido em linguagem de programação C++, que é amplamente utilizada para resolução de problemas de engenharia, devido à sua eficiência e desempenho, favorecendo uma melhor precisão e velocidade nos cálculos.

O principal método de pesquisa e estudo a ser utilizado no desenvolvimento deste trabalho é a bibliometria, por meio da qual a literatura acadêmica disponível é analisada para identificar padrões, tendências e soluções mais pertinentes para o problema de estudo.

Além da Dissertação de Mestrado *Modelagem Numérica da Fundação de uma Máquina Rotativa Considerando Rigidez e Amortecimento do Solo com Distribuição não Homogênea* (ALBUQUERQUE, 2015), estudou-se principalmente a obra de Wolf, *Dynamic Soil-Structure Interaction* (WOLF, 1985), o material Fundações de Máquinas (SANTOS, 2024) e os estudos desenvolvidos no projeto de graduação *Análise Numérica de Funções de Impedância em Solo Homogêneo* (OLIVEIRA, 2021).

Também foram estudadas as normas DIN 4024/Parte 1 (DIN, 1988) e ACI 351.3R – *Foundation for Dynamic Equipment* (ACI, 2018), além da norma N-1848 – Projeto de Fundações de Máquinas da PETROBRAS (2023).

1.4. Apresentação e Estrutura do Trabalho

O trabalho é composto por três partes principais: a fundamentação teórica, o problema numérico a ser analisado e os exemplos de aplicação.

Na primeira parte (Capítulo 2), aborda-se a propagação de ondas elásticas em solos e a solução da equação diferencial do movimento clássico em termos de deslocamentos. Em seguida, são discutidos os métodos para obtenção dos parâmetros do solo, com base na sua modelagem, e soluções clássicas para os coeficientes de rigidez e amortecimento no contexto de interação solo-estrutura em fundações circulares, conforme proposto por Richart *et al.* (1970) para solo homogêneo. Além disso, incluem-se coeficientes de rigidez para fundações de geometria retangular e outras formas, conforme definidos por Wolf (1994), também para solo homogêneo.

Posteriormente, explora-se a formulação dos coeficientes das funções de impedância proposta por Luco (1974) para solos com estratificação horizontal, assim como as contribuições de Gazetas (1983), Gazetas *et al.* (1991), aprimoradas por Wolf (1994) para situações mais gerais. Essas soluções clássicas são estruturadas para permitir a comparação com soluções numéricas baseadas em modelos inspirados no Método das Soluções Fundamentais de Green, desenvolvidos para solos heterogêneos, e discutidos na segunda parte, a partir do Capítulo 3.

No Capítulo 3, é desenvolvido o problema estudado, com a descrição da metodologia de modelagem e análise, bem como sua implementação no algoritmo desenvolvido.

No Capítulo 4, analisam-se as funções de impedância na modelagem do solo sujeito a ações dinâmicas, assumindo-se a hipótese de solo heterogêneo em um semiespaço. Aplica-se uma formulação matemática robusta, com uma comparação dos valores de rigidez e amortecimento com os de outras formulações clássicas discutidas no Capítulo 2. Também se utiliza a modelagem de uma fundação retangular para avaliar a distribuição geométrica das funções de impedância e sua variação com a frequência. A validação é feita com base nos resultados obtidos por Luco (1974) e Wolf (1985).

O Capítulo 5 apresenta as conclusões obtidas dos estudos realizados e as direções sugeridas para continuidade da pesquisa. Em seguida, são listadas as referências bibliográficas do trabalho e, por fim, o apêndice com o código desenvolvido em C++, contendo a metodologia do Capítulo 4 para a obtenção dos resultados discutidos no Capítulo 5.

1.5. Revisão Bibliográfica

Richart *et al.* (1970) foram os primeiros a introduzir e a se dedicar ao problema das fundações submetidas a carregamentos dinâmicos de máquinas rotativas em solos homogêneos. O seu trabalho compilou os conhecimentos existentes até então

relacionados aos procedimentos de análise, projeto e medições da resposta das fundações às cargas dinâmicas e transmissão das vibrações pelo solo.

Seu trabalho também exhibe as formulações para as funções de impedância, independentes da frequência excitadora, utilizadas até o presente para fundações circulares. As expressões obtidas para o amortecimento têm maior relevância, pois os trabalhos seguintes utilizaram-se desses resultados obtidos na obra, atualizando apenas as formulações para coeficientes de rigidez.

Os mesmos autores supramencionados desenvolveram a formulação dos coeficientes de rigidez de fundações retangulares, enquanto as expressões para o amortecimento foram as mesmas encontradas por Richart para fundações circulares, porém utilizando raios equivalentes para transformar áreas e inércias para a geometria circular. Essa formulação de fundações retangulares para solo homogêneo foi superada pelos estudos de Wolf (1994).

Poulos (1974) foi o pioneiro no estudo de estacas sujeitas a carregamentos dinâmicos, sobretudo na consideração do efeito de grupo em que o deslocamento de uma estaca provoca deslocamentos nas demais.

Ainda em 1974, Luco dedicou-se ao estudo da relação entre forças e deslocamentos nas fundações, ao expandir a solução do problema das funções de impedância para fundação rígida, com a consideração da estratificação horizontal do solo.

Em 1977, Novak desenvolveu o PILAY, programa computacional escrito em linguagem *Fortran* para cálculo da rigidez e do amortecimento dinâmicos de estacas em solo estratificado.

Gazetas (1983) e Gazetas, *et al.* (1991) propuseram novas formulações para coeficientes de rigidez das estacas, destacando a importância do efeito de grupo, baseados no trabalho de Poulos. Wolf (1994), baseado no trabalho de Gazetas, além de elaborar uma solução diferente para as fundações retangulares, superando a proposta por Richart *et al.* (1970) expandiu o problema para fundações de geometria quaisquer e para solos heterogêneos.

Além dessas novas formulações para os coeficientes de rigidez, Wolf (1985) apresentou a metodologia principal para o desenvolvimento deste trabalho, cuja abordagem é aplicada tanto para problemas bidimensionais quanto para problemas tridimensionais, transformando o problema do domínio da frequência em um problema no domínio do número de onda, formulando o cálculo da amplitude de deslocamentos, com uma discretização em elementos circulares.

Neste trabalho, Wolf (1985) estudou os sistemas dinâmicos discretizados, com diversos estudos paramétricos e principalmente o problema relacionado à modelagem

do solo, utilizando o Método das Soluções Fundamentais de Green, método computacional para solução de equações diferenciais, para calcular os coeficientes de rigidez dinâmica das fundações.

Até a presente data, não há nenhuma norma brasileira específica para o projeto de fundações de máquinas. Entretanto, a indústria no país adota principalmente as recomendações prescritas na norma N-1848 – Projeto de Fundações de Máquinas da PETROBRAS (2023).

Em nível internacional, há respaldo da norma DIN 4024:1988, da DIN (*Deutsches Institut für Normung*), da ACI 351 3R:2018 – *Foundation for Dynamic Equipment*, da ACI – *American Concrete Institute*, e da ISO 2372:2018, da ISO (*International Organization for Standardization*).

Também é possível citar como referência o *Eurocode 8 - EN 1998-6 "Design of structures for earthquake resistance – Part 6: Towers, masts and chimneys"*, que apesar de tratar especificamente de torres, mastros e chaminés, também apresenta diretrizes para modelagem dinâmica de fundações de estruturas esbeltas e equipamentos industriais.

Capítulo 2

2. FUNDAMENTAÇÃO TEÓRICA

O projeto de fundações de máquinas sujeitas a carregamentos dinâmicos envolve uma investigação precisa do solo para obter suas propriedades da maneira correta, além da determinação correta das características da estrutura e da máquina em si, o que normalmente é fornecido pelo fabricante.

O principal critério de projeto é limitar as amplitudes de deslocamento da fundação, devidas à vibração da máquina, de modo a não prejudicar o bom funcionamento da mesma e nem provocar sensação de insegurança, ansiedade, medo ou desconforto nas pessoas próximas à estrutura. Essa classificação pode ser feita conforme a ISO 2372, que separa as máquinas por classes, e com base na frequência de operação, classificando a amplitude em faixas ótimas, aceitáveis, toleráveis e inaceitáveis para cada classe de máquina.

Adicionalmente, o projeto objetiva evitar o fenômeno da ressonância, mantendo a frequência de operação da máquina distante da frequência própria do sistema estrutura-solo. O controle do afastamento entre essas frequências é realizado basicamente com alterações na massa e na rigidez do sistema máquina-estrutura-solo. Isso pode ser facilmente compreendido ao se observar a solução da equação do movimento, conforme o item 2.2 do presente trabalho.

2.1. Propagação de Ondas

Existem dois principais tipos de ondas no solo, as ondas de volume, classificadas como primárias (ou de compressão), e ondas secundárias (ou de cisalhamento) e as ondas superficiais, que podem ser de *Love* ou de *Rayleigh*. Essas ondas têm sua origem nas solicitações dinâmicas na superfície do solo, como na base de fundações de máquinas, ou podem ser geradas por eventos naturais, como sismos.

2.1.1. Ondas de Volume

As ondas de volume dividem-se entre ondas primárias e secundárias. As ondas primárias são também denominadas ondas de compressão, ondas de dilatação volumétrica, ou ondas P e caracterizam-se fundamentalmente pela propagação das partículas do solo na mesma direção da propagação de onda, conforme mostra a Figura 2.1.

Conforme afirma Santos (2024), o entendimento de como ocorre a propagação dessas ondas pode ser realizado a partir do estudo da propagação axial em uma barra, uma vez que o movimento das partículas ocorre na mesma linha da propagação das ondas.

Já as ondas secundárias podem ser chamadas de ondas de cisalhamento ou ondas S e tem como característica o movimento das partículas do solo de maneira perpendicular à direção da propagação da onda (SANTOS, 2024), vide Figura 2.2. Também pode-se compreender a propagação dessas ondas estudando uma barra, cujas partículas se deslocam perpendicularmente à direção da propagação das ondas.

Matematicamente pode-se definir a velocidade de onda de cisalhamento como:

$$V_S = \sqrt{\frac{G}{\rho}} \quad (2.1)$$

Já a velocidade de onda de compressão é definida por:

$$V_P = V_S \cdot \sqrt{\frac{2 - 2\nu}{1 - 2\nu}} \quad (2.2)$$

Em que,

V_P – Velocidade de onda de principal ou de compressão;

V_S – Velocidade de onda secundária ou de cisalhamento;

G – Módulo de elasticidade transversal;

ν – Coeficiente de Poisson;

ρ – Massa específica;

A dedução dessa formulação será realizada no item 2.1 desta dissertação.

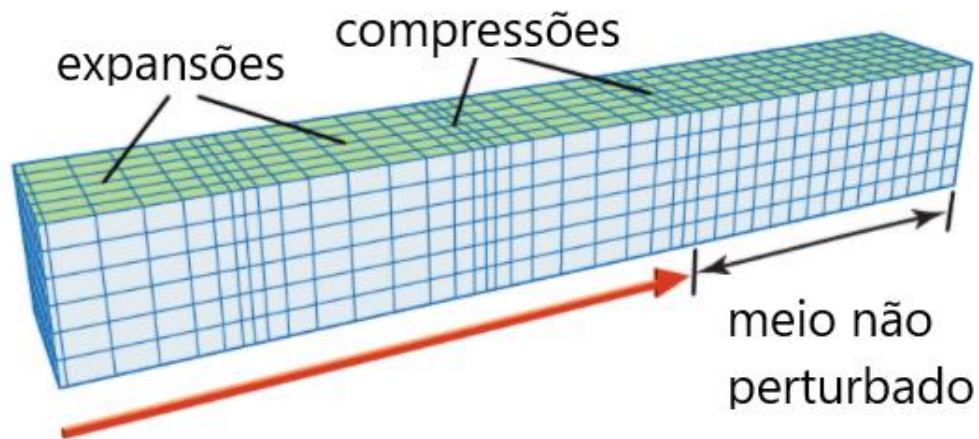


Figura 2.1 – Ondas P, primárias ou de compressão (Adaptado de *Encyclopædia Britannica*, 2021).

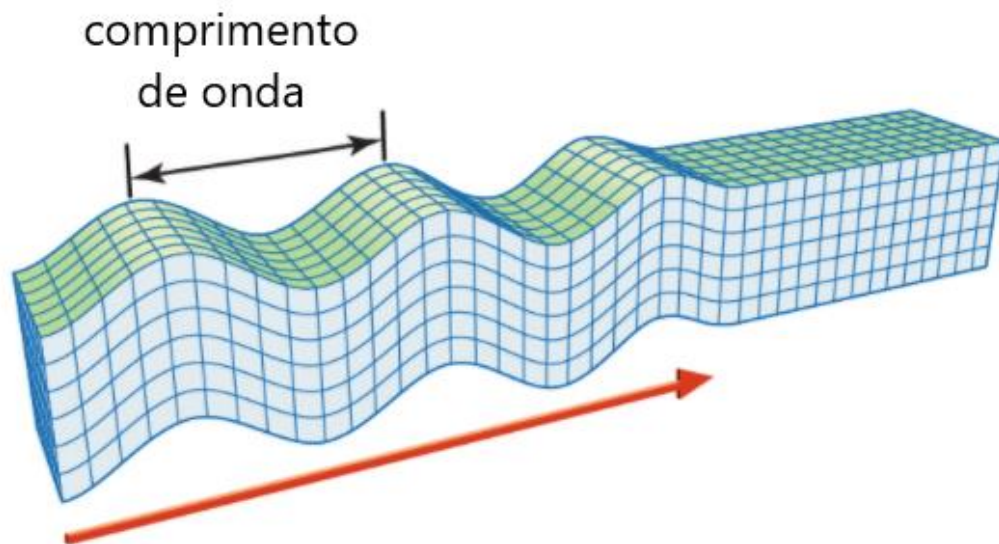


Figura 2.2 – Ondas S, secundárias ou de cisalhamento (Adaptado de *Encyclopædia Britannica*).

2.1.2. Ondas de Superfície

As ondas de Love, ou ondas Q, surgem nos limites horizontais entre camadas de solo diferentes e são caracterizadas pelo deslocamento das partículas do solo perpendicularmente ao eixo horizontal de propagação da onda, no próprio plano horizontal, conforme afirma SANTOS (2024).

Na superfície livre, manifestam-se as ondas de Rayleigh, ou ondas R, que se caracterizam pelo deslocamento das partículas em trajetórias elípticas, no plano vertical da propagação da onda.

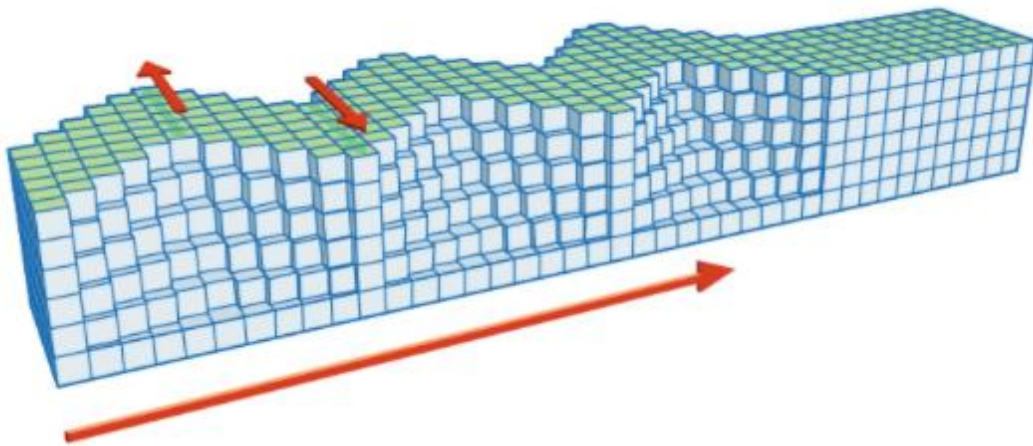


Figura 2.3 – Ondas de Love, ou ondas Q (Adaptado de *Encyclopædia Britannica*, 2021).

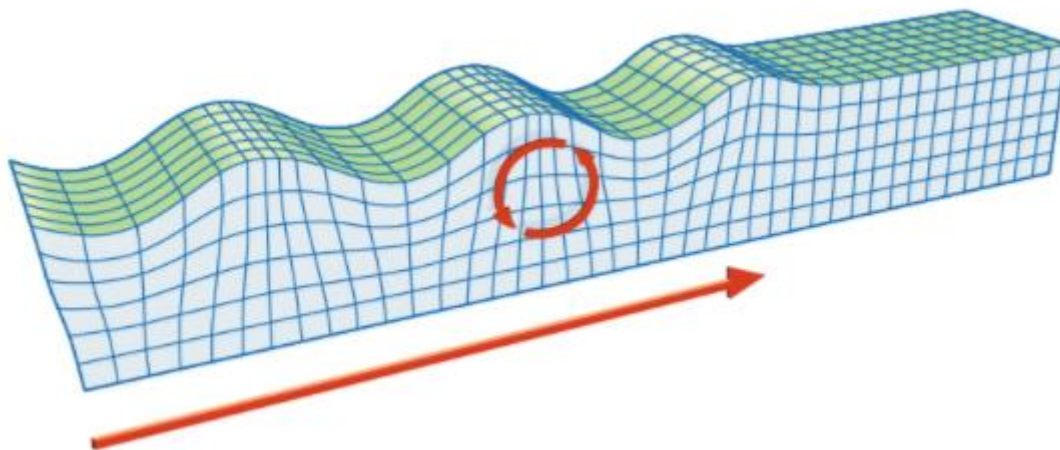


Figura 2.4 – Ondas de Rayleigh, ou ondas R (Adaptado de *Encyclopædia Britannica*, 2021).

2.1.3. Propagação Unidimensional de Ondas em uma Barra

Conforme deduzido por Clough e Penzien (1993), a figura 2.5, apresenta as deformações e tensões no sentido longitudinal de uma barra horizontal, sendo N o esforço normal.

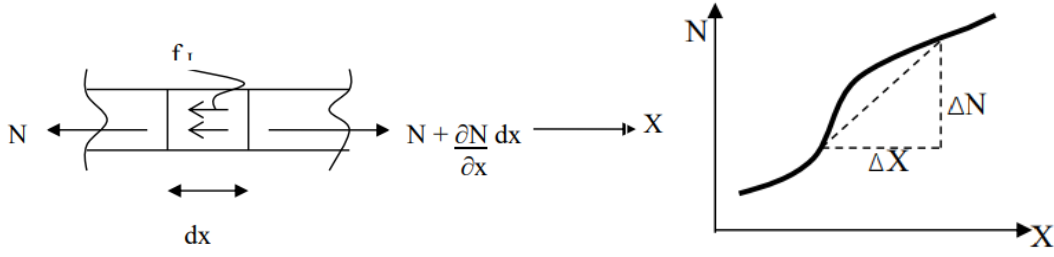


Figura 2.5 – Deformações e tensões no sentido longitudinal de uma barra horizontal (SANTOS, 2024).

Pelo equilíbrio de forças no sentido axial, escreve-se a equação do movimento, segundo o eixo X .

$$N + f_I dx - \left(N + \frac{\partial N}{\partial x} dx \right) = 0 \quad (2.3)$$

Onde f_I é a força de inércia distribuída por unidade de comprimento da barra, definida por:

$$f_I = \bar{m} \frac{\partial^2 u}{\partial t^2} \quad (2.4)$$

Substituindo (2.4) na equação (2.3), tem-se:

$$\bar{m} \frac{\partial^2 u}{\partial t^2} - \frac{\partial N}{\partial x} = 0 \quad (2.5)$$

Explorando a relação entre força axial e deslocamento:

$$\bar{m} \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x} \left(EA \frac{\partial u}{\partial x} \right) = 0 \quad (2.6)$$

Considerando EA como sendo constante ao longo do comprimento horizontal, tem-se que:

$$\bar{m} \frac{\partial^2 u}{\partial t^2} - EA \frac{\partial^2 u}{\partial x^2} = 0 \quad (2.7)$$

Definimos a constante C , com grandeza de uma velocidade como:

$$C = \sqrt{\frac{EA}{m}} = \sqrt{\frac{E}{\rho}} \quad (2.8)$$

Dividimos a expressão (2.7) por $\overline{m}$ substituímos C na expressão (2.9):

$$\frac{\partial^2 u}{\partial t^2} - C^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad (2.9)$$

Com isso, tem-se que a solução geral da equação (2.9) acima é:

$$u(x, t) = u_1(x + Ct) + u_2(x - Ct) \quad (2.10)$$

Define-se o comprimento de onda para excitação harmônica, como na figura 2.6:

$$\Delta e = v \cdot \Delta t = \frac{C}{f} \quad (2.11)$$

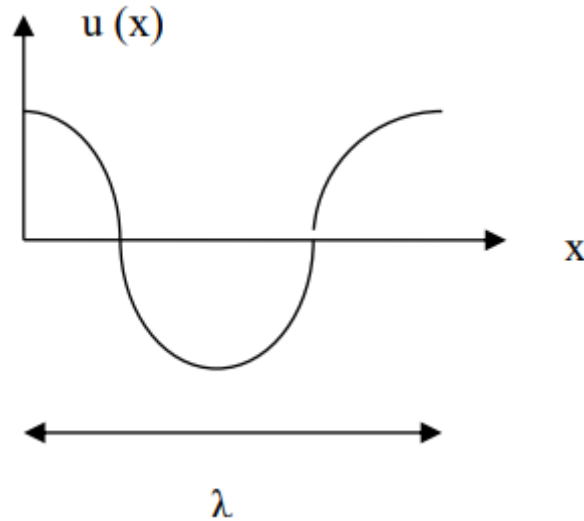


Figura 2.6 – Comprimento de onda (SANTOS, 2024).

Fisicamente, pode-se dizer que a solução expressa em (2.10) representa duas parcelas de ondas propagando: com u_1 no sentido negativo do eixo X e uma u_2 no sentido positivo.

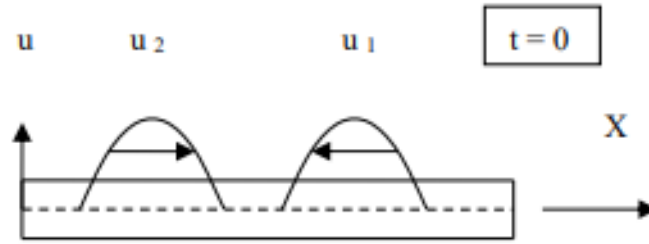


Figura 2.7 – Direção da propagação de ondas (SANTOS,2024).

Para o instante $t = 0$, o campo de deslocamentos é função somente da posição.

$$u = 2(x + Ct) + (x - Ct)^2 \quad (t \geq 0 \text{ e } u \geq 0) \quad (2.12)$$

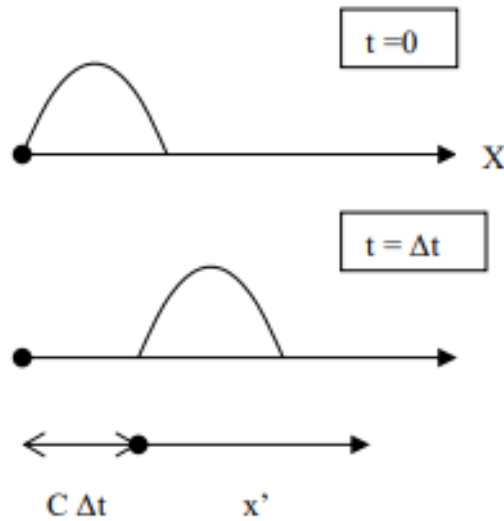


Figura 2.8 – Deslocamentos das ondas para $t=0$ e $t=\Delta t$ (SANTOS,2024).

Então, para o instante $t = 0$, tem-se a equação dos deslocamentos, com $u \geq 0$:

$$u = 2x + x^2 \quad (2.13)$$

Quando $t = \Delta t$, com $u \geq 0$

$$u = 2(x + C\Delta t) + (x - C\Delta t)^2 \quad (2.14)$$

Tomando $x' = x - C\Delta t$:

$$u = 2x' + x'^2 \quad (2.15)$$

O que permite concluir que a curva que representa os deslocamentos é a mesma com defasagem $C\Delta t$ no espaço. Portanto, em um período Δt , a onda percorre o espaço $C\Delta t$, onde C é a velocidade de propagação da onda, conforme a fórmula clássica de velocidade média, pois segundo a cinemática $V_m = \Delta S / \Delta t$.

Tem-se que as tensões são iguais a:

$$\sigma(x, t) = E \frac{\partial u(x, t)}{\partial x} \quad (2.16)$$

Considerando-se as condições de contorno em uma extremidade, uma frente de onda se propagaria no sentido positivo do eixo horizontal X pela barra real, enquanto outra realizaria o deslocamento na mesma direção, mas em sentido contrário em uma barra fictícia simétrica à real. Realizando as seguintes considerações de condições de contorno em uma extremidade:

a) Onda que se propaga em direção a uma extremidade engastada:

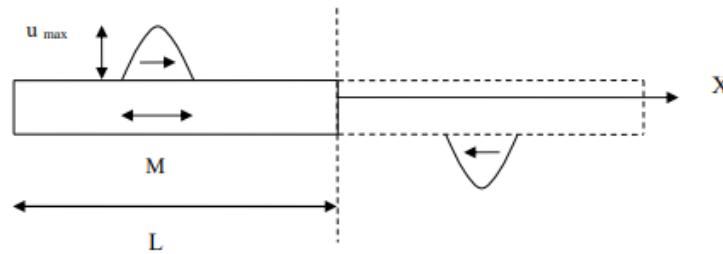


Figura 2.9 – Propagação de onda em direção a extremidade engastada (SANTOS,2024).

b) Onda que se propaga em direção a uma extremidade livre:

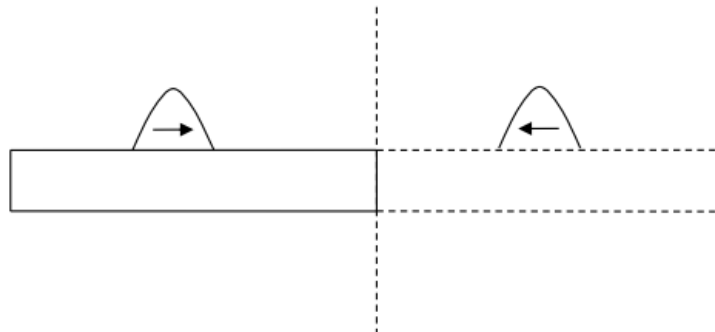


Figura 2.10 – Propagação de onda em direção a extremidade livre (SANTOS,2024).

c) Reflexão e refração de ondas na interface de meios desiguais:

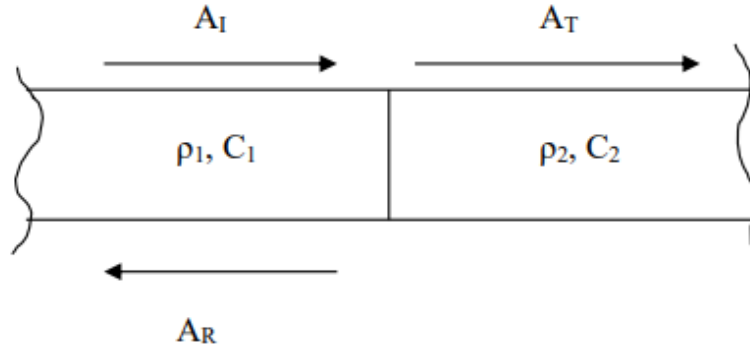


Figura 2.11 – Reflexão e refração de ondas na interface de meios desiguais (SANTOS,2024).

Onde A_I , A_R e A_T são as amplitudes dos deslocamentos, definidos por:

$$A_R = \frac{\alpha - 1}{\alpha + 1} A_I \quad (2.17)$$

$$A_T = \frac{2}{\alpha + 1} A_I \quad (2.18)$$

Em que,

$$\alpha = \frac{\rho_2 \cdot C_2}{\rho_1 \cdot C_1} = \sqrt{\frac{G_2 \cdot \rho_2}{G_1 \cdot \rho_1}} \quad (2.19)$$

Tal que $\alpha \geq 1$ e $v_1 = v_2$.

2.1.4. Condições de Contorno Na Barra Horizontal

Ao aplicar-se um amortecedor viscoso, como condição de contorno, em um ponto da barra do modelo, é possível simular, a situação real de um meio semi-infinito. Assim a barra da Figura 2.12 pode ser substituída por um modelo equivalente, com um amortecedor, conforme a Figura 2.13.

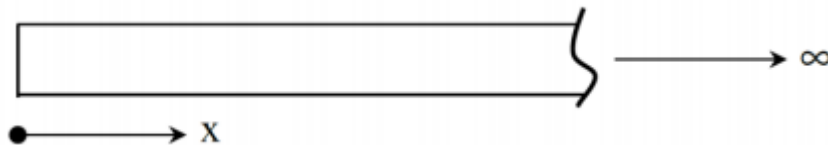


Figura 2.12 – Barra horizontal infinita (SANTOS, 2024).

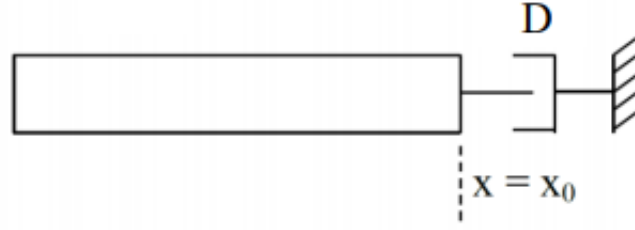


Figura 2.13 – Barra horizontal com amortecedor para representação do meio infinito (SANTOS, 2024).

Supõe-se que:

$$u = u_0 \cdot \text{sen}(x - Ct) \quad (2.20)$$

E, substituindo (2.20) na equação (2.16),

$$\sigma = E \cdot u_0 \cdot \text{sen}(x - Ct) \quad (2.21)$$

A força para $x = x_0$ é:

$$F(x_0) = A \cdot \sigma(x_0) = E \cdot A \cdot u_0 \cdot \cos(x_0 - Ct) \quad (2.22)$$

Com um amortecedor D, em módulo, para $x = x_0$:

$$F(x_0) = D \cdot \frac{\partial u}{\partial t}(x_0) = D \cdot C \cdot u_0 \cdot \cos(x_0 - Ct) \quad (2.23)$$

Igualando as expressões (2.22) e (2.23), tem-se:

$$D \cdot C = E \cdot A \rightarrow D = \frac{E \cdot A}{\sqrt{E/\rho}} = \sqrt{E \cdot \rho} \cdot A \quad (2.24)$$

Para um círculo de raio r :

$$D = \sqrt{E \cdot \rho} \cdot \pi \cdot r^2 = \sqrt{2 \cdot G \cdot (1 - \nu)} \cdot \rho \cdot \pi \cdot r^2 \quad (2.25)$$

Então com o amortecedor de constante igual a D, é reproduzida a solução exata.

2.1.5. Propagação Tridimensional de Ondas no Meio Contínuo

Conforme Richart *et al.* (1970), as equações de equilíbrio no elemento prismático infinitesimal inserido em um meio elástico infinito, conforme na Figura 2.14, abaixo, são formuladas a seguir.

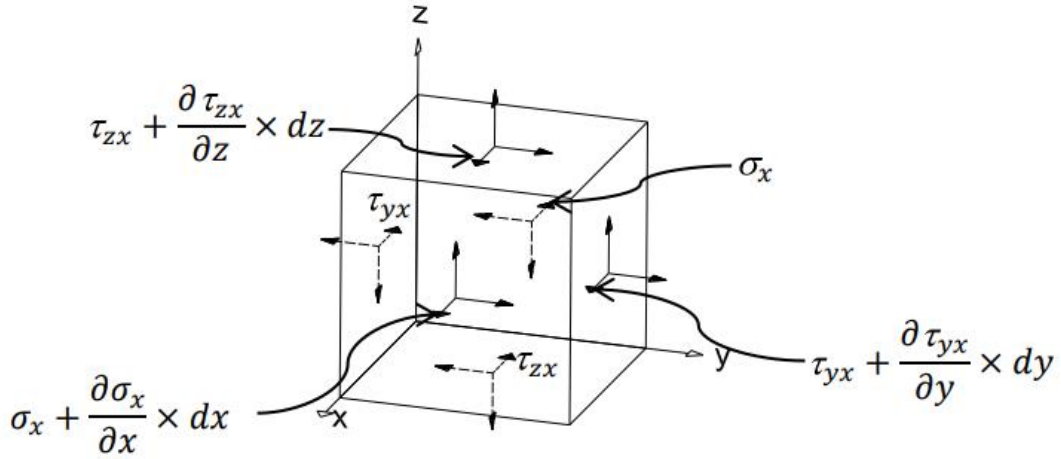


Figura 2.14 – Tensões em elemento prismático infinitesimal em meio elástico infinito (RICHART, 1970).

Realizando o equilíbrio na direção do eixo X, considerando todas as seis parcelas representadas na Figura 2.14:

$$\begin{aligned}
 & \left(\sigma_x + \frac{\partial \sigma_x}{\partial x} dx \right) dy dz - \sigma_x dy dz + \left(\tau_{xy} + \frac{\partial \tau_{xy}}{\partial y} dy \right) dx dz \\
 & - \tau_{xy} dx dz + \left(\tau_{xz} + \frac{\partial \tau_{xz}}{\partial z} dz \right) dx dy - \tau_{xz} dx dy \quad (2.26) \\
 & = \rho(dx dy dz) \frac{\partial^2 u}{\partial t^2}
 \end{aligned}$$

Considerando-se as três direções, X, Y e Z, chegamos às seguintes expressões:

$$\rho \frac{\partial^2 u}{\partial t^2} = (\lambda + G) \frac{\partial \varepsilon}{\partial x} + G \nabla^2 u \quad (2.27)$$

$$\rho \frac{\partial^2 v}{\partial t^2} = (\lambda + G) \frac{\partial \varepsilon}{\partial y} + G \nabla^2 v \quad (2.28)$$

$$\rho \frac{\partial^2 w}{\partial t^2} = (\lambda + G) \frac{\partial \varepsilon}{\partial z} + G \nabla^2 w \quad (2.29)$$

Sendo que, pela expressão do módulo do módulo de deformação transversal, expresso em (2.81):

$$\lambda = \frac{\nu \cdot E}{(1 + \nu)(1 - 2\nu)} = \frac{2 \cdot \nu \cdot G}{(1 - 2\nu)} \quad (2.30)$$

E,

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2.31)$$

$$\varepsilon = \varepsilon_x + \varepsilon_y + \varepsilon_z = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \quad (2.32)$$

Tem-se que a primeira solução é obtida ao se diferenciar respectivamente por x , y e z e somar as equações (2.27) a (2.29):

$$\rho \frac{\partial^2 \varepsilon}{\partial t^2} = (\lambda + 2G) \nabla^2 \varepsilon \quad (2.33)$$

Definindo-se a velocidade de onda primária, ou de compressão, V_p :

$$V_p^2 = \frac{\lambda + 2G}{\rho} \quad (2.34)$$

Substituindo-se (2.34) em (2.33), tem-se a equação de propagação de ondas de dilatação volumétrica ε , em que o deslocamento das partículas do solo se desenvolvem na mesma linha em que a onda se propaga (SANTOS, 2024):

$$\frac{\partial^2 \varepsilon}{\partial t^2} = V_p^2 \nabla^2 \varepsilon \quad (2.35)$$

A segunda solução é obtida ao se diferenciar a equação (2.28), em relação a z , e (2.29), em relação a y , e subtraindo-se uma à outra:

$$\rho \frac{\partial^2}{\partial t^2} \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right) = G \cdot \nabla^2 \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right) \quad (2.36)$$

Denotando as rotações em função dos deslocamentos por:

$$2\omega_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \quad (2.37)$$

Substituindo (2.37) em (2.36):

$$\rho \frac{\partial^2}{\partial t^2} \omega_x = G \cdot \nabla^2 \omega_x \quad (2.38)$$

Definindo-se a velocidade de onda secundária, ou de cisalhamento, V_s como:

$$V_s^2 = \frac{G}{\rho} \quad (2.39)$$

Substituindo-se (2.37) em (2.38), tem-se a equação de propagação de ondas de distorção ω , caracterizadas pelo movimento das partículas do solo se desenvolverem perpendicularmente à direção propagação da onda (SANTOS,2024):

$$\frac{\partial^2 \omega_x}{\partial t^2} = V_s^2 \cdot \nabla^2 \omega_x \quad (2.40)$$

O que também permite escrever,

$$\frac{\partial^2 \omega_y}{\partial t^2} = V_s^2 \cdot \nabla^2 \omega_y \quad (2.41)$$

$$\frac{\partial^2 \omega_z}{\partial t^2} = V_s^2 \cdot \nabla^2 \omega_z \quad (2.42)$$

Da relação entre as velocidades de onda Primária e Secundária tiramos:

$$\frac{V_P}{V_s} = \sqrt{\frac{\lambda + 2G}{G}} = \sqrt{\frac{2\nu}{(1 - 2\nu)} + 2} = \sqrt{\frac{2 - 2\nu}{(1 - 2\nu)}} \quad (2.43)$$

O que nos dá:

$$V_P = V_s \sqrt{\frac{2 - 2\nu}{1 - 2\nu}} \quad (2.44)$$

2.2. A Equação Diferencial do Movimento

O sistema de equações diferenciais de movimento é representado por:

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = F(t) \quad (2.45)$$

Em que,

M – Matriz de massa do sistema;

C – Matriz de amortecimento do sistema;

K – Matriz de rigidez do sistema;

$F(t)$ – Vetor de forças aplicadas;

$u(t)$ – Vetor de deslocamentos;

$\dot{u}(t)$ – Vetor de velocidades;

$\ddot{u}(t)$ – Vetor de acelerações;

Assumindo-se um sistema de um grau de liberdade sujeito a um carregamento harmônico, a força aplicada pode ser expressa pela seguinte expressão, em que ω é a frequência circular de excitação, pode ser escrito:

$$F(t) = F_0 \cos(\omega t) = \frac{F_0}{2} (e^{i\omega t} + e^{-i\omega t}) \quad (2.46)$$

O deslocamento em função do tempo t é expresso conforme (2.47) a partir da solução permanente da equação (2.45) de movimento, também expressa na forma complexa:

$$u(t) = A \cos(\omega t) + B \sin(\omega t) = C_A e^{i\omega t} + C_B e^{-i\omega t} \quad (2.47)$$

Na expressão acima, A e B são constantes determinadas a partir das condições de contorno, e C_A e C_B constantes em variáveis complexas. Aplicam-se as transformações de (2.48) a (2.51) na expressão (2.47) anterior e, pela transformação de Euler, a equação é reescrita em (2.52).

$$e^{i\omega t} = \cos(\omega t) + i \cdot \sin(\omega t) \quad (2.48)$$

$$e^{-i\omega t} = \cos(-\omega t) + i \cdot \sin(-\omega t) = \cos(\omega t) - i \cdot \sin(\omega t) \quad (2.49)$$

$$C_A = C_1 + i \cdot D_1 \quad (2.50)$$

$$C_B = C_2 + i \cdot D_2 \quad (2.51)$$

Tem-se que:

$$u(t) = (C_1 + i \cdot D_1)[\cos(\omega t) + \sin(\omega t)] + (C_2 + i \cdot D_2)[\cos(\omega t) - \sin(\omega t)] \quad (2.52)$$

Pode-se reorganizar a equação (2.52) da seguinte forma:

$$u(t) = (C_1 + C_2)\cos(\omega t) - (D_1 - D_2)\sin(\omega t) + i[(D_1 + D_2)\cos(\omega t) + (C_1 - C_2)\sin(\omega t)] \quad (2.53)$$

De modo a anular a parte imaginária, obtém-se que as constantes devem ser definidas como:

$$C_1 = C_2 = C \quad (2.54)$$

$$D_1 = -D_2 = D \quad (2.55)$$

Igualando as equações (2.47) e (2.53) obtém-se C e D em função de A e B.

$$A = C_1 + C_2 = 2C \rightarrow C = \frac{A}{2} \quad (2.56)$$

$$B = D_1 + D_2 = -2D \rightarrow D = -\frac{B}{2} \quad (2.57)$$

Substituindo-se, então os valores das constantes C e D encontrados nas expressões (2.56) e (2.57), respectivamente, na expressão (2.47), obtém-se a equação:

$$u(t) = \frac{A - iB}{2} e^{i\omega t} + \frac{A + iB}{2} e^{-i\omega t} \quad (2.58)$$

Se anularmos B, pela transformação de Euler é possível obter os valores das constantes A e B para a expansão de $A \cos(\omega t)$. Considerando cada parcela de forma separada, obtém-se a solução para a equação do movimento.

Começamos pela primeira parcela:

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = \frac{F_0}{2}e^{i\omega t} \quad (2.59)$$

$$U = (U_1 + iU_2)e^{i\omega t} \quad (2.60)$$

$$(-\omega^2 M + i\omega C + K)(U_1 + iU_2)e^{i\omega t} = \frac{F_0}{2}e^{i\omega t} \quad (2.61)$$

Criam-se as constantes K_1 e K_2 , separando a parte real da imaginária tal que:

$$K_1 = K - \omega^2 M \quad (2.62)$$

$$K_2 = \omega C \quad (2.63)$$

Assim, tem-se:

$$U_1 + iU_2 = \frac{F_0/2}{(-\omega^2 M + i\omega C + K)} = \frac{F_0/2}{K_1 + iK_2} \quad (2.64)$$

Pode-se reescrever a equação (2.64) conforme as expressões (2.65) e (2.66):

$$U_1 = \frac{K_1 F_0}{2(K_1^2 + K_2^2)} \quad (2.65)$$

$$U_2 = \frac{-K_2 F_0}{2(K_1^2 + K_2^2)} \quad (2.66)$$

Para a segunda parcela, é realizado o mesmo procedimento.

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = \frac{F_0}{2}e^{-i\omega t} \quad (2.67)$$

$$U = (U_1 + iU_2)e^{-i\omega t} \quad (2.68)$$

$$(-\omega^2 M - i\omega C + K)(U_1 + iU_2)e^{-i\omega t} = \frac{F_0}{2}e^{-i\omega t} \quad (2.69)$$

$$K_1 = K - \omega^2 M \quad (2.70)$$

$$K_2 = -\omega C \quad (2.71)$$

$$U_1 + iU_2 = \frac{F_0/2}{(-\omega^2 M - i\omega C + K)} = \frac{F_0/2}{K_1 + iK_2} \quad (2.72)$$

Pode-se, então, reescrever a equação (2.72) conforme as expressões (2.73) e (2.74), da mesma forma em que foi realizado no procedimento anterior:

$$U_1 = \frac{K_1 F_0}{2(K_1^2 + K_2^2)} \quad (2.73)$$

$$U_2 = \frac{K_2 F_0}{2(K_1^2 + K_2^2)} \quad (2.74)$$

O deslocamento total trata-se da soma de (2.60) e (2.68).

$$U_{TOTAL} = (U_1 + iU_2)e^{i\omega t} + (U_1 + iU_2)e^{-i\omega t} \quad (2.75)$$

Substituindo os valores encontrados para U_1 e U_2 :

$$U_{TOTAL} = \left[\frac{K_1 F_0}{2(K_1^2 + K_2^2)} + i \frac{-K_2 F_0}{2(K_1^2 + K_2^2)} \right] e^{i\omega t} + \left[\frac{K_1 F_0}{2(K_1^2 + K_2^2)} + i \frac{K_2 F_0}{2(K_1^2 + K_2^2)} \right] e^{-i\omega t} \quad (2.76)$$

Voltando ao problema inicial, substituímos os valores nas expressões (2.48) e (2.49), o que resulta em (2.77):

$$U_{TOTAL} = \left[\frac{K_1 F_0}{2(K_1^2 + K_2^2)} + i \frac{-K_2 F_0}{2(K_1^2 + K_2^2)} \right] (\cos(\omega t) + i \cdot \sin(\omega t)) + \left[\frac{K_1 F_0}{2(K_1^2 + K_2^2)} + i \frac{K_2 F_0}{2(K_1^2 + K_2^2)} \right] (\cos(\omega t) - i \cdot \sin(\omega t)) \quad (2.77)$$

A solução é então o deslocamento dado pela expressão (2.78):

$$u(t) = F_0 \frac{K_1 \cos(\omega t) + K_2 \sin(\omega t)}{K_1^2 + K_2^2} \quad (2.78)$$

Ainda é possível escrever $u(t)$ em função do seu valor máximo u_0 , substituindo nas expressões (2.70) e (2.71).

$$u(t) = u_0 \cdot e^{i\omega t} = \frac{F_0 e^{i\omega t}}{-\omega^2 M + i\omega C + K} \quad (2.79)$$

Pode-se concluir então que resolver cada parcela separadamente é possível. Ou seja, pode-se resolver apenas a parcela de $\cos(\omega t)$ para a parte real e $\sin(\omega t)$ para parte imaginária. Isso torna possível a análise matricial de fundações de máquinas utilizando as variáveis complexas, a partir da definição do módulo de deformação transversal complexo, como será explicitado no capítulo 3.

Reescreve-se a expressão da amplitude como:

$$|u| = \frac{F_0}{\sqrt{K_1^2 + K_2^2}} \quad (2.80)$$

Onde,

$$K_1 = K - \omega^2 M$$

$$K_2 = \omega C$$

Conforme explicitado anteriormente, o objetivo do projeto de fundações de máquinas é a limitação da amplitude, de modo a evitar o fenômeno da ressonância. Pela formulação acima demonstrada, é necessário determinar os valores dos parâmetros M , C e K , que são respectivamente a massa do sistema, o amortecimento do sistema e a rigidez do sistema. Assim, pode-se observar a importância da determinação correta desses parâmetros e o conhecimento de seu comportamento e distribuição.

É importante lembrar que para múltiplos graus de liberdade, essa equação diferencial e sua solução se tornam matriciais.

2.3. Avaliação das Propriedades do Solo

Definida a expressão da amplitude dos deslocamentos no item 2.2, e conhecendo que os deslocamentos existentes em fundações de máquinas são

relativamente pequenos, correspondendo a distorções da ordem de $10^{-4} \%$ a $10^{-3} \%$, pode-se concluir que as deformações no solo também são muito pequenas. Isso proporciona a utilização de soluções elástico-lineares de forma a modelar o comportamento do solo. (SANTOS, 2024).

As propriedades do solo que precisam ser definidas para a realização do projeto, em cada camada do solo, são o módulo de deformação transversal, ou módulo de elasticidade transversal (G), o coeficiente de Poisson (ν), a massa específica (ρ) e a fração de amortecimento crítico (ε).

O módulo de deformação transversal G é expresso pela equação (2.81), em função do módulo de deformação longitudinal E :

$$G = \frac{E}{2(1 + \nu)} \quad (2.81)$$

O módulo de deformação transversal é um parâmetro muito importante, pois está presente nas diversas expressões a serem utilizadas no projeto, e é um parâmetro com vasta variedade de expressões empíricas para relacionar os ensaios mais comumente realizados, como o *Standard Penetration Test* (ou ensaio SPT), ao módulo de deformação transversal em si.

Embora a utilização de resultados de ensaios SPT permita avaliar o parâmetro G , o valor estimado apresenta menor precisão do que aqueles obtidos por outros ensaios de caracterização do solo mais específicos. Estes ensaios, por sua vez, são usualmente realizados para avaliar o parâmetro G em casos de fundações submetidas a carregamentos elevados.

A D4428M.14 (ASTM, 2014) - *Standard Test Methods for Crosshole Seismic Testing* normatiza o ensaio de *Crosshole*, que é realizado em campo e permite resultados muito mais precisos, comparado com a caracterização obtida via ensaios SPT.

Neste ensaio, são abertos três furos no subsolo, que devem estar alinhados e revestidos com calda de cimento. São realizados ensaios ao longo da profundidade dos furos, excitando ondas sísmicas no primeiro furo a partir de um golpe com um martelo. Nos outros dois furos, na mesma profundidade em que a fonte gerou a onda, são medidos, em geofones, os tempos de chegada das ondas de compressão (P) e de cisalhamento (S).

Com a diferença dos tempos de chegada de cada onda nos dois geofones e conhecendo-se a distância entre os furos, são medidas as velocidades V_P e V_S das ondas de compressão e de cisalhamento.

Pela teoria de propagação tridimensional de ondas:

$$G = \rho \cdot V_S^2 \quad (2.82)$$

$$\nu = \frac{1}{2} \cdot \frac{(V_P^2 - 2 \cdot V_S^2)}{V_P^2 - V_S^2} \quad (2.83)$$

A Figura 2.15 mostra um esquema do ensaio de *Crosshole* adaptado da Associação Brasileira de Mecânica dos Solos e Engenharia Geotécnica/ Associação Brasileira de Empresas de Engenharia de Fundações e Geotecnia (ABMS/ABEF, 2002).

Este é um dos principais ensaios para a avaliação de G , que deve ser determinado de forma empírica, caso não seja possível realizar este ensaio.

Em solos não coesivos, ou seja, solos compostos de pedras, pedregulhos, cascalhos e areias, o módulo de deformação transversal é influenciado pela dimensão dos grãos e seu formato, além de ser influenciado pelo índice de vazios (e), que é a relação entre o volume de vazios e volume de sólidos, e pela tensão efetiva média.

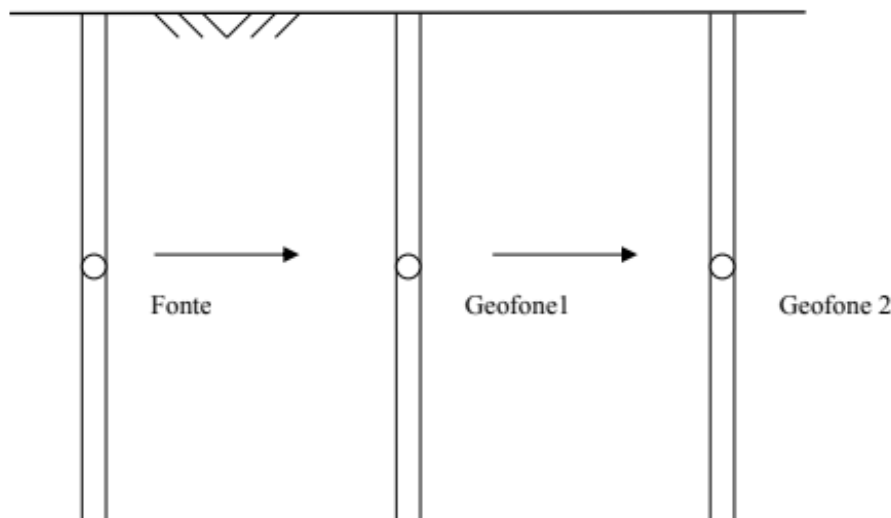


Figura 2.15 – Ensaio *Crosshole* (adaptado de ABMS/ ABEF, 2002).

Os valores típicos para o índice de vazios em argilas normalmente adensadas são:

Argila Mole: $e \geq 1,5$;

Argila Média: $e = 0,6$;

Argila Rija: $e = 0,3$;

Argila Dura: $e = 0,2$.

Já para areias, podem ser considerados os valores extremos:

Areias muito compactas: $e = 0,35$;

Areias muito fofas: $e = 1,05$;

Areia média: $e = 0,70$.

A tensão octaédrica média é variável para as profundidades do solo e é determinada conforme a expressão (2.84):

$$\sigma_m = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3} \quad (2.84)$$

Em que,

$$\sigma_2 = \sigma_3 = K_0 \cdot \sigma_1;$$

K_0 é o coeficiente de empuxo no repouso (tipicamente 0,4~0,5 em areias e 0,5~0,8 em argilas).

σ_1 , para solos homogêneos, é a tensão efetiva vertical, ou seja, seu valor numérico é igual à pressão efetiva no solo em uma dada profundidade. Esse valor é função da pressão imposta na superfície do solo pela fundação e da pressão efetiva devida ao peso de terra.

Richart *et al.* (1970) propuseram as expressões (2.85) e (2.86) a seguir, para a determinação do parâmetro G em areias, limitadas a deformações distorcionais de 0,01%.

A expressão (2.85) apresenta a estimativa para o módulo de deformação transversal para areias de grãos arredondados, cujo índice de vazios encontra-se no intervalo entre 0,35 e 0,85. Deve-se utilizar a tensão octaédrica média (2.84).

$$G(MPa) = 218 \sigma_m^{1/2} \times \frac{(2,17 - e)^2}{1 + e} \quad (2.85)$$

Para areias de grãos angulosos, cujo índice de vazios encontra-se no intervalo entre 0,60 e 1,30, o módulo de deformação transversal G pode ser calculado conforme a expressão (2.86).

$$G(MPa) = 102 \times \sigma_m^{1/2} \times \frac{(2,97 - e)^2}{1 + e} \quad (2.86)$$

Fora as expressões de Richart *et al.* (1970) também existem as expressões empíricas a seguir para obtenção dos parâmetros do solo a partir dos valores de N_{SPT} , obtidos pelo ensaio SPT.

Para o módulo de deformação transversal G :

Ohsaki e Iwasaki, (1973) *apud* SANTOS, (2024) (recomendado para valores de G de até 100 MPa):

$$G(\text{MPa}) = 11,5 (N_{SPT})^{0,8} \quad (2.87)$$

Imai e Tonouchi (1982) *apud* SANTOS, (2024):

$$G(\text{MPa}) = 14,07 (N_{SPT})^{0,68} \quad (2.88)$$

Seed *et al.* (1983) *apud* SANTOS, (2024) (válido somente para areias):

$$G(\text{MPa}) = 6,22 N_{SPT} \quad (2.89)$$

Para o valor da velocidade de cisalhamento V_S :

Ohta e Goto (1978) *apud* SANTOS, (2024):

$$V_S(\text{m/s}) = 85,3 (N_{SPT})^{0,341} \quad (2.90)$$

Sykora e Stokoe (1983) *apud* SANTOS, (2024):

$$V_S(\text{m/s}) = 101 (N_{SPT})^{0,29} \quad (2.91)$$

A avaliação de G sempre envolverá incertezas, o que torna indispensável uma análise paramétrica para a avaliação da variação das propriedades do solo.

Pode-se observar os valores típicos para a massa específica (ρ), coeficiente de Poisson e fração de amortecimento crítico (ε) na Tabela 2.1, para cada tipo de solo conforme SANTOS (2024). Para o amortecimento, é considerada uma faixa usual de projetos de fundações de máquinas, que corresponde a uma deformação distorcional de 10⁻⁴%.

Tabela 2.1 – Valores típicos de ρ , ν e ε (SANTOS,2024).

Tipo de solo	ρ (Mg/m ³)	ν	ε (%)
Areia	1,70 a 2,10	0,3 a 0,4	2,0
Argila	1,40 a 2,00	0,4 a 0,5	2,0
Rocha	2,50 a 2,70	0,15 a 0,3	2,0

2.4. Coeficientes de Rigidez e Amortecimento para Fundações Diretas

O problema físico de um bloco rígido é apresentado na Figura 2.16. Na figura pode-se observar três componentes essenciais para o problema: a máquina, a fundação e o solo, modelado conforme os parâmetros discretizados no item 2.3.

A Figura 2.16 pode ser rerepresentada por um modelo de um grau de liberdade vertical, conforme a Figura 2.17, em que a interação solo-estrutura é simulada através de molas (K) e amortecedores (C), definidos em função das propriedades do solo (G , ρ , ν) e da frequência circular excitadora (ω).

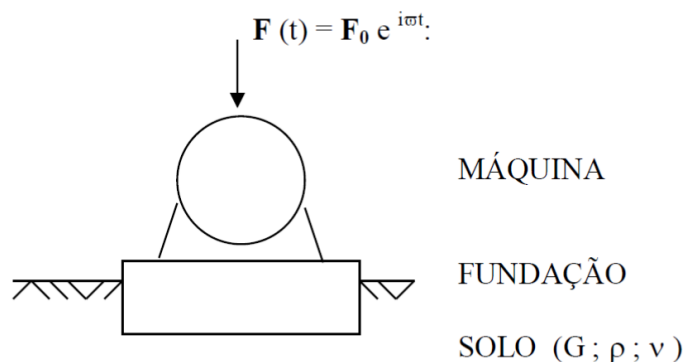


Figura 2.16 – Representação da interação Máquina x Fundação x Solo (SANTOS, 2024).

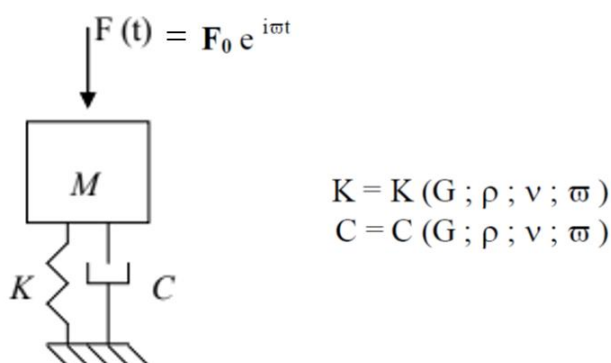


Figura 2.17 – Modelo numérico de um grau de liberdade para Blocos Rígidos (SANTOS, 2024).

O grau de liberdade representado na Figura 2.17 é apenas o vertical, sem perda de generalidade, apenas para efeito de simplificação da figura. A força $F(t)$ dinâmica é aplicada ao sistema, de massa M , que transmite esta força ao terreno.

Vale ressaltar que essa é apenas uma simplificação do problema, que originalmente é composto por seis graus de liberdade: três translacionais e três rotacionais. Aqui é representado apenas o grau de liberdade vertical.

Além da força vertical, também existe uma força horizontal, que provoca um momento relativo a base. Teoricamente existem então três forças determinadas, porém ao se realizarem medições, surgem forças em diversos sentidos e forças em frequências diferentes da frequência de operação da máquina. Então, para este estudo, é considerado um modelo teórico com seis graus de liberdade.

É necessário estimar, ou verificar nos desenhos de fabricação, qual é a massa desequilibrada, que é a parte que gera a força. Com essa informação, as normas DIN 4024:1988 e ACI 351 3R:20 fornecem excentricidades a serem consideradas, com valores obtidos pela DIN aproximadamente 2,5 vezes maiores do que da ACI.

A excentricidade, que irá ocorrer ao longo do desgaste da máquina, provoca uma força desbalanceada associada a uma condição de operação, e gera a força dinâmica para os dados de entrada do problema.

2.4.1. Consideração de Blocos Rígidos e Blocos Flexíveis

Os modelos massa-mola consideram o bloco de fundação como sendo rígido, de modo que todos os seus pontos se movimentam simultaneamente e de forma síncrona.

A análise pode então ser simplificada, de modo que é possível concentrar a massa das máquinas sobre o bloco de fundação e a massa do próprio bloco em pontos que representem o sistema estudado. Pode-se observar pela Figura 2.17 que o bloco rígido de massa M concentra as propriedades do sistema e transmite sua força para o terreno pela mola e pelo amortecedor. Observa-se que originalmente o problema possui seis graus de liberdade: três translações e três rotações.

Para garantir a validade dessa hipótese, é necessário garantir que as deformações causadas no bloco, devido os carregamentos dinâmicos oriundos das máquinas possam ser desprezadas, o que é possível de ser admitido desde que o bloco tenha altura suficiente.

De modo a representar corretamente o comportamento dos blocos flexíveis, o modelo da fundação deve ser feito baseado no Método das Soluções Fundamentais, com uma malha discretizada, representando todos os graus de liberdade, considerando propriedades de rigidez e amortecimento translacionais, bem como as rotacionais, em todas as direções, já que essas propriedades são correlacionadas entre si.

2.4.2. Fundações sobre Solo Homogêneo

Pode-se realizar a modelagem do solo de maneira aproximada, considerando um semiespaço homogêneo. Essa abordagem pode ser adotada desde que a fundação se encontre assente em camada que possa ser aproximada por uma camada homogênea e que seja profunda.

Segundo Gazetas (1983) essa aplicação leva a resultados compatíveis com a observação experimental. A seguir serão descritas formulações para as funções de impedância obtidas por diferentes autores para fundações apoiadas sobre solo homogêneo, de forma a representar a interação solo-estrutura.

Considerando o sistema massa-mola-amortecedor representado na Figura 2.17, é necessário definir os valores dos coeficientes de rigidez e amortecimento (K e C). Conforme Richart *et al.* (1970), é possível considerar esses parâmetros como

independentes da frequência de excitação ω , quando se trata do caso de solos homogêneos.

Entretanto, as situações em que a consideração de que o solo possui um perfil perfeitamente homogêneo, em que se pode considerar as propriedades elásticas do solo como constantes ao longo de toda sua profundidade, são muito incomuns na prática.

a) Fundações Circulares

Soluções em fundações circulares são pouco usuais em projetos de fundações de máquinas. Porém, o estudo dos parâmetros deduzidos para fundações circulares assentes em solo homogêneo tem relevância para as demais formulações das funções de impedância que são mais recentes, uma vez que os coeficientes de amortecimento são adotados conforme as expressões (2.107) a (2.110) de Richart, com devidas correções para os raios em cada grau de liberdade.

Somente para solo homogêneo, será possível se considerar diretamente soluções aproximadas independentes da frequência de excitação ω

A seguir são reproduzidos os coeficientes de rigidez (K) amortecimento (C) e os coeficientes adimensionais de fração de massa (B), utilizados para cálculo dos coeficientes de amortecimento, a partir da fração de amortecimento crítico, propostos por Richart *et al.* (1970), para fundações circulares assentes em solo homogêneo, aproveitados nas formulações de Wolf (1994), para os graus de liberdade translacionais (vertical e horizontal) e rotacionais:

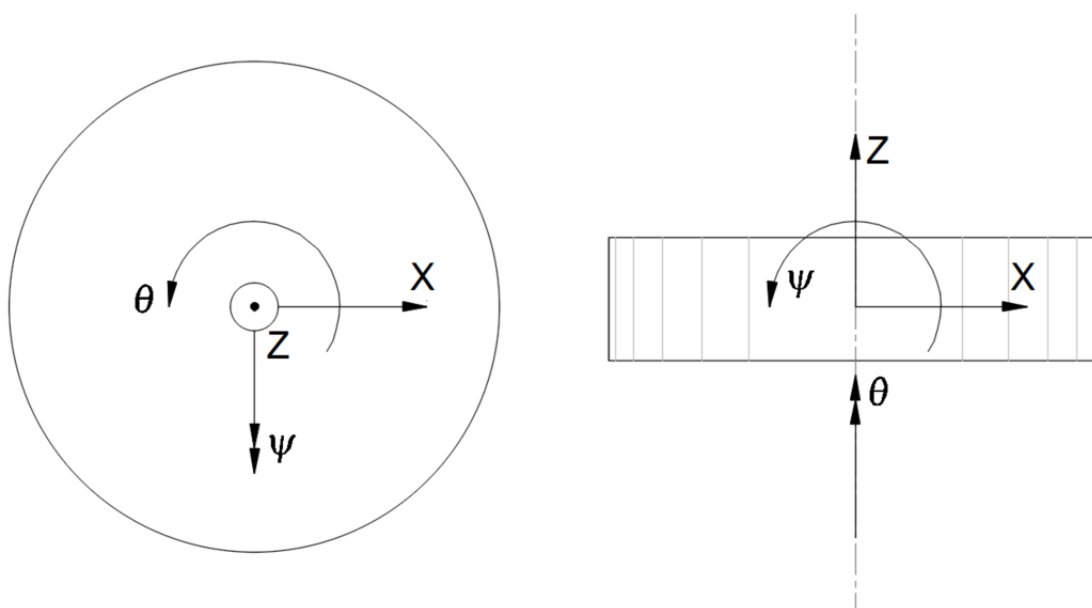


Figura 2.18 – Orientação dos eixos conforme formulação de fundações circulares de Richart *et al.* (1970).

Coeficientes de Rigidez:Vertical (K_Z):

$$K_Z = \frac{4Gr}{1 - \nu} \quad (2.92)$$

Horizontal (K_X):

$$K_X = \frac{32(1 - \nu)Gr}{7 - 8\nu} \quad (2.93)$$

Rotacional (K_ψ):

$$K_\psi = \frac{8Gr^3}{3(1 - \nu)} \quad (2.94)$$

Torcional (K_θ):

$$K_\theta = \frac{16Gr^3}{3} \quad (2.95)$$

Fração de massa:Vertical (B_Z):

$$B_Z = \frac{1 - \nu}{4} \cdot \frac{M}{\rho r^3} \quad (2.96)$$

Horizontal (B_X):

$$B_X = \frac{7 - 8\nu}{32(1 - \nu)} \cdot \frac{M}{\rho r^3} \quad (2.97)$$

Rotacional (B_ψ):

$$B_\psi = \frac{3(1 - \nu)I_\psi}{8 \rho r^5} \quad (2.98)$$

Torcional (B_θ):

$$B_\theta = \frac{I_\theta}{\rho r^5} \quad (2.99)$$

Para um disco circular,

$$I_\psi = \frac{\pi r^4}{4} \rho h = \frac{Mr^2}{4} \quad (2.100)$$

E,

$$I_\theta = 2I_\psi \quad (2.101)$$

Fração de amortecimento:

Para obter-se o amortecimento, tem-se a relação para a fração de amortecimento crítico (ε):

$$\varepsilon = \frac{C}{C_c} = \frac{C}{2\sqrt{KM}} \quad (2.102)$$

Onde,

C – Amortecimento do sistema;

C_c – Amortecimento crítico;

K – Rigidez do sistema;

M – Massa do sistema;

Vertical (ε_z):

$$\varepsilon_z = \frac{0,425}{\sqrt{B_z}} \quad (2.103)$$

Horizontal (ε_x):

$$\varepsilon_x = \frac{0,288}{\sqrt{B_x}} \quad (2.104)$$

Rotacional (ε_ψ):

$$\varepsilon_\psi = \frac{0,15}{(1 + B_\psi)\sqrt{B_\psi}} \quad (2.105)$$

Torcional (ε_θ):

$$\varepsilon_\theta = \frac{0,5}{1 + 2B_\theta} \quad (2.106)$$

Onde,

G – Módulo de elasticidade transversal;

I_ψ – Momento de inércia rotacional;

I_θ – Momento de inércia torcional;

r – Raio;

ν – Coeficiente de Poisson;

ρ – Massa específica;

As expressões para os amortecedores são definidas a partir dos valores de ε , K e M como:

Vertical (C_z):

$$C_z = \frac{3,4r^2}{1 - \nu} \sqrt{\rho G} \quad (2.107)$$

Horizontal (C_x):

$$C_x = \frac{18,4 (1 - \nu)r^2}{7 - 8\nu} \sqrt{\rho G} \quad (2.108)$$

Rotacional (C_ψ):

$$C_\psi = \frac{0,8r^4}{(1 - \nu) \cdot (1 + B_\psi)} \sqrt{\rho G} \quad (2.109)$$

Torcional (C_θ):

$$C_\theta = \frac{1}{1 + 2B_\theta} \sqrt{\frac{16Gr^3 \cdot I_\theta}{3}} \quad (2.110)$$

b) Fundações Retangulares

Richart *et al.* (1970) também obtiveram expressões para coeficientes de rigidez para o caso de fundações de forma retangular, formulação que foi superada por WOLF (1994).

Para os coeficientes de amortecimento de fundações retangulares, Richart *et al.* (1970) indicam que devem ser calculados utilizando-se as mesmas fórmulas para fundações circulares, que foram apresentadas nas expressões (2.107) a (2.110), porém com valores de raios equivalentes para cada grau de liberdade, convertendo a área e a inércia, conforme as expressões (2.111) a (2.113) a seguir:

Raio equivalente para translações:

$$r = \left(\frac{4cd}{\pi} \right)^{\frac{1}{2}} \quad (2.111)$$

Raio equivalente para rotação:

$$r = \left(\frac{16cd^3}{3\pi} \right)^{\frac{1}{4}} \quad (2.112)$$

Raio equivalente para torção:

$$r = \left(\frac{16cd(c^2 + d^2)}{6\pi} \right)^{\frac{1}{4}} \quad (2.113)$$

Wolf (1994), valendo-se dos resultados obtidos por Gazetas (1983), propôs novas formulações para os cálculos dos coeficientes de rigidez de fundações retangulares, conforme as expressões de (2.114) a (2.119). A Figura 2.18 representa a orientação dos eixos consideradas, com “a” sendo a metade do lado de maior dimensão, e “b” a metade do lado de menor dimensão.

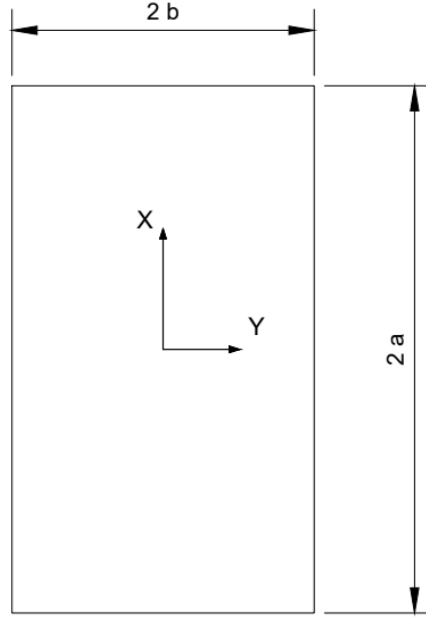


Figura 2.19 – Orientação dos eixos conforme formulação de fundações retangulares de Wolf (1994).

$$K_X \left(\frac{kN}{m} \right) = \frac{Gb}{2-\nu} \left(6,8 \left(\frac{a}{b} \right)^{0,65} + 2,4 \right) \quad (2.114)$$

$$K_Y \left(\frac{kN}{m} \right) = \frac{Gb}{2-\nu} \left(6,8 \left(\frac{a}{b} \right)^{0,65} + 0,8 \frac{a}{b} + 1,6 \right) \quad (2.115)$$

$$K_Z \left(\frac{kN}{m} \right) = \frac{Gb}{1-\nu} \left(3,1 \left(\frac{a}{b} \right)^{0,75} + 1,6 \right) \quad (2.116)$$

$$K_{XX} \left(\frac{kN}{m^2} \right) = \frac{Gb^3}{1-\nu} \left(3,2 \left(\frac{a}{b} \right) + 0,8 \right) \quad (2.117)$$

$$K_{YY} \left(\frac{kN}{m^2} \right) = \frac{Gb^3}{1-\nu} \left(3,73 \left(\frac{a}{b} \right)^{2,4} + 0,27 \right) \quad (2.118)$$

$$K_{ZZ} \left(\frac{kN}{m^2} \right) = Gb^3 \left(4,25 \left(\frac{a}{b} \right)^{2,45} + 4,06 \right) \quad (2.119)$$

Em que,

G – Módulo de elasticidade transversal do solo;

ν – Coeficiente de Poisson do solo;

a, b – dimensões da fundação, que seguem a orientação da Figura 2.19, tal que

$a \geq b$.

c) *Fundações de Geometria Qualquer*

Wolf (1994) apresentou uma formulação para fundações de geometria qualquer, que está expressa pelas equações de (2.120) a (2.125).

A Figura 2.20 mostra a consideração da geometria de fundações de forma qualquer que se utilizam dessa formulação. A fundação, representada pela parte hachurada da figura, deve ser circunscrita por um retângulo cujo lado maior tem dimensão igual a $2L$, e o menor de dimensão $2B$. Ou seja, a dimensão “ L ” corresponde à metade do lado do retângulo de maior dimensão, e “ B ” à metade do lado de menor dimensão.

O eixo X está posicionado de forma a ser paralelo ao lado de dimensão maior do retângulo, de valor $2L$, enquanto o eixo Y é paralelo ao menor lado do retângulo, de dimensão $2B$.

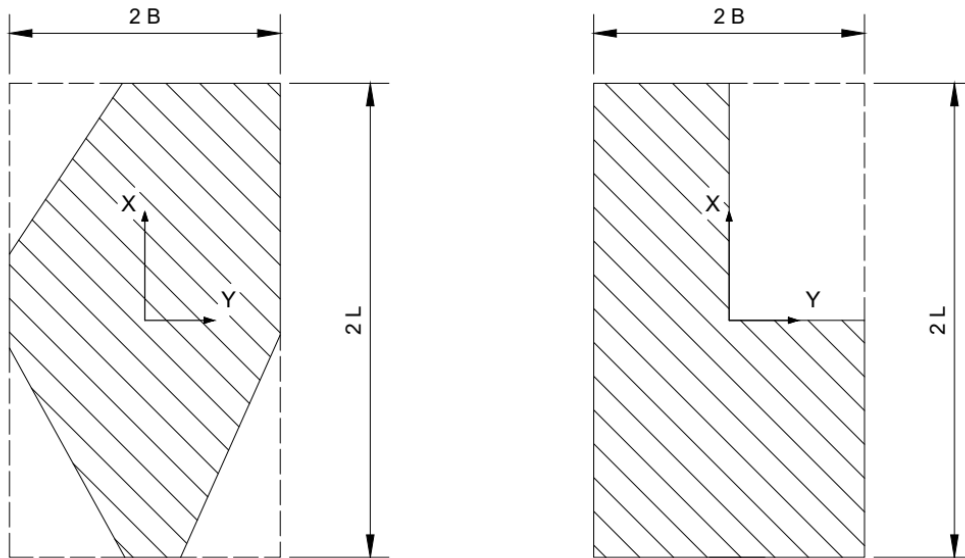


Figura 2.20 – Orientação dos eixos conforme formulação de fundações de forma qualquer de Wolf (1994).

$$K_x \left(\frac{kN}{m} \right) = K_y - \frac{0,2}{0,75 - \nu} GL \left(1 - \frac{B}{L} \right) \quad (2.120)$$

$$K_y \left(\frac{kN}{m} \right) = \frac{2GL}{2 - \nu} \left(2 + 2,5 \left(\frac{A_b}{4L^2} \right)^{0,85} \right) \quad (2.121)$$

$$K_z \left(\frac{kN}{m} \right) = \frac{2GL}{1-\nu} \left(0,73 + 1,54 \left(\frac{A_b}{4L^2} \right)^{0,75} \right) \quad (2.122)$$

$$K_{xx} \left(\frac{kN}{m^2} \right) = \frac{G}{1-\nu} I b_x^{0,75} \left(\frac{L}{B} \right)^{0,25} \left(2,4 + 0,5 \frac{B}{L} \right) \quad (2.123)$$

$$K_{yy} \left(\frac{kN}{m^2} \right) = \frac{2,9G}{1-\nu} I b_y^{0,75} \left(\frac{L}{B} \right)^{0,15} \quad (2.124)$$

$$K_{zz} \left(\frac{kN}{m^2} \right) = 3,5G I b_z^{0,75} \left(\frac{B}{L} \right)^{0,4} \left(\frac{I b_z}{B^4} \right)^{0,2} \quad (2.125)$$

$I b_x$, $I b_y$ e $I b_z$ são os momentos de inércia (em m^4), respectivamente com relação aos eixos X , Y e Z . Ou seja, $I b_y$ é o maior momento de inércia, e $I b_x$ é o menor momento de inércia. $I b_z$ é a soma de $I b_x$ e $I b_y$. Esses valores podem ser calculados conforme as expressões (2.126) a (2.128), para fundações retangulares.

$$I b_x = \frac{4}{3} B^3 L \quad (2.126)$$

$$I b_y = \frac{4}{3} L^3 B \quad (2.127)$$

$$I b_z = I b_x + I b_y \quad (2.128)$$

2.4.3. Fundações sobre Solo com Estratificação Horizontal

Gazetas (1983), a partir de resultados de Luco (1974), desenvolveu uma formulação para os coeficientes de rigidez do solo, obtidos para fundações circulares sobre uma camada de solo, apoiada em base rígida, cujos coeficientes são apresentados nas expressões (2.129) a (2.132) a seguir. É importante ressaltar que para cada expressão existe uma faixa de validade que deve ser respeitada. O problema pode ser ilustrado pela Figura 2.21.

Nota-se que a principal diferença dessa formulação para a formulação de Richart *et al.* (1970), é que aquela considera a máquina assente em uma camada de espessura tão grande que o valor de sua dimensão vertical possa ser considerado infinita, e assim pode ser aproximada a uma camada homogênea, enquanto essa formulação considera a faixa de validade em função da profundidade H e do raio R da fundação.

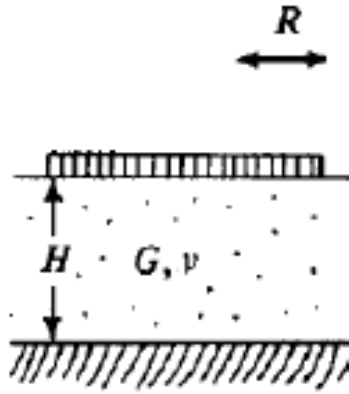


Figura 2.21 – Fundação Circular em Estrato sobre Camada Rígida (GAZETAS,1983).

Vertical (K_v):

$$K_v = \frac{4GR}{1-\nu} \left(1 + 1,28 \frac{R}{H} \right) \quad (2.129)$$

Faixa de Validade:

$$\frac{H}{R} > 2$$

Horizontal (K_h):

$$K_h = \frac{8GR}{2-\nu} \left(1 + \frac{1R}{2H} \right) \quad (2.130)$$

Faixa de Validade:

$$\frac{H}{R} > 1$$

Rotacional (K_r):

$$K_r = \frac{8GR^3}{3(1-\nu)} \quad (2.131)$$

Faixa de Validade:

$$4 > \frac{H}{R} > 1$$

Torcional (K_t):

$$K_t = \frac{16}{3} GR^3 \quad (2.132)$$

Faixa de Validade:

$$\frac{H}{R} > 1,25$$

2.4.4. Fundações sobre Solo Heterogêneo

Um dos pontos de maior importância na avaliação dos efeitos da interação solo-estrutura em uma análise dinâmica é a relação entre forças e deslocamentos da estrutura. Esses termos estão disponíveis em forma das funções de impedância, mas a maioria dos resultados são restritos a uma modelagem de um subespaço homogêneo e elástico.

As formulações apresentadas não fornecem resultados para os coeficientes de amortecimento, e apenas consideram a mesma formulação adotada por Richart (1970) para essa função de impedância.

Luco (1974) conseguiu obter soluções gerais, cuja formulação considera uma fundação rígida circular assente na superfície de um meio viscoelástico estratificado e sujeita a vibrações harmônicas forçadas. Esta abordagem permite a obtenção não apenas de coeficientes de rigidez, mas também os coeficientes de amortecimento do solo, que representam a dissipação da energia do sistema pelo solo, por radiação.

Em um contexto de solo heterogêneo, expressões independentes da frequência de excitação são inviáveis devido à variação das propriedades do solo, que influenciam sua resposta dinâmica a carregamentos externos.

A complexa interação entre essas camadas, cada uma com comportamento distinto em resposta a frequências específicas, torna a obtenção de soluções analíticas simplificadas difícil e envolve diversas incertezas e imprecisões.

Nesse cenário, abordagens baseadas em métodos numéricos são essenciais para capturar com precisão as variações de frequência e as propriedades do solo ao longo da profundidade.

Portanto, uma análise robusta da fundação de máquinas em solo heterogêneo demanda considerações detalhadas da interação solo-estrutura e o uso de técnicas avançadas de modelagem e simulação. Entretanto, é possível considerar formulações simplificadas dentro de intervalos de validade.

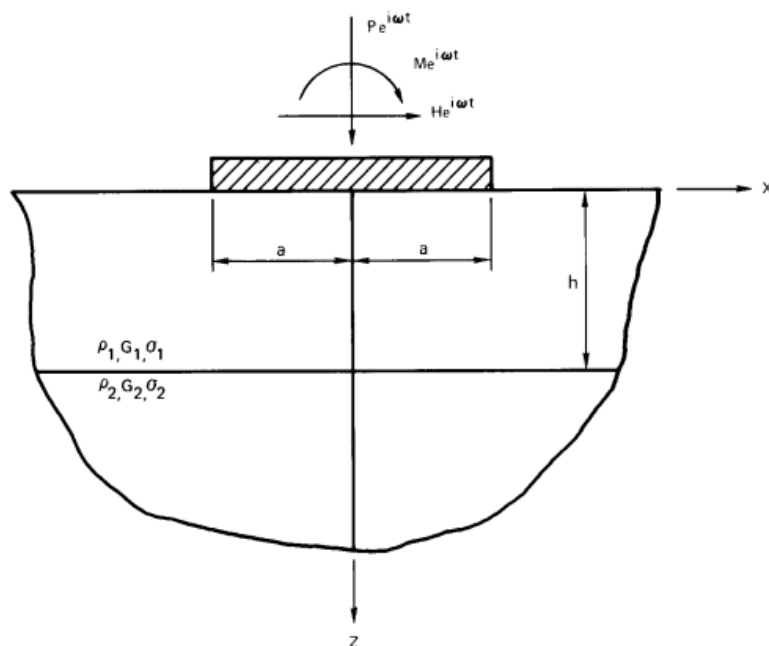


Figura 2.22 – Vista em corte transversal do modelo estudado por Luco (1974).

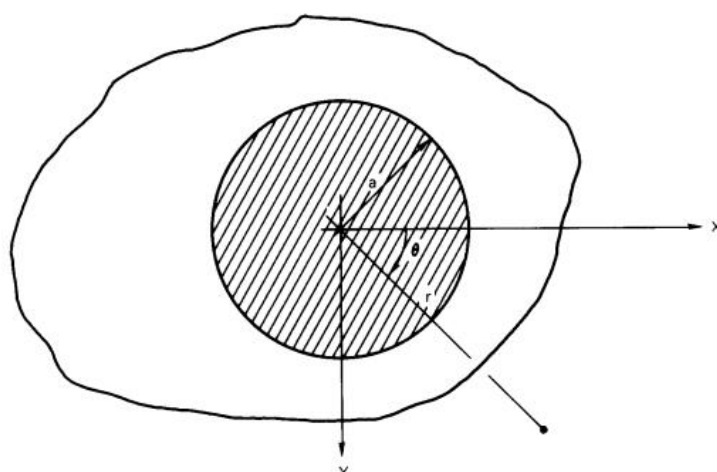


Figura 2.23 – Vista superior do modelo axissimétrico estudado por Luco (1974).

Gazetas (1983) desenvolveu uma formulação considerando a estratificação horizontal sobre camadas-infinitas do solo, para fundações sobre uma base rígida.

Considera-se agora uma fundação circular, adicionando-se na formulação a estratificação do solo, com essa fundação sobre uma camada cujo módulo de elasticidade transversal possui valor G_1 , que, por sua vez, se apoia sobre uma camada -infinita de solo cujo módulo de elasticidade transversal possui o valor G_2 .

Sendo $G_2 > G_1$, e o solo representado conforme ilustrado pela Figura 2.21, as expressões para os parâmetros de rigidez estática da formulação referida são apresentadas nas expressões (2.133) a (2.135).

Vale salientar que nessa formulação também existem faixas de validade para a utilização das expressões, que devem ser respeitadas.

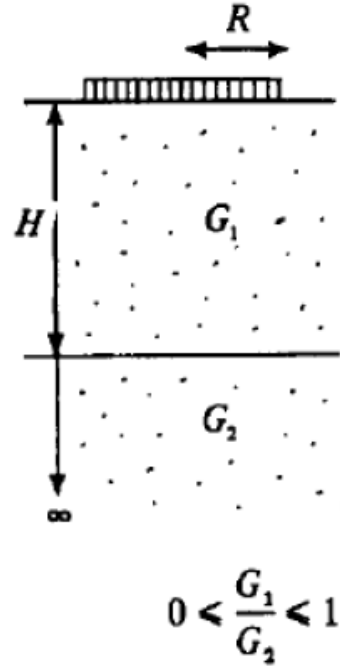


Figura 2.24 – Fundação Circular assente em Estrado sobre Camada infinita (GAZETAS,1983).

Vertical (K_v):

$$K_v = \frac{4G_1R}{1-\nu_1} \cdot \left(\frac{1 + 1,28 \frac{R}{H}}{1 + 1,28 \frac{R}{H} \frac{G_1}{G_2}} \right) \quad (2.133)$$

Faixa de Validade:

$$1 \leq \frac{H}{R} < 5$$

Horizontal (K_h):

$$K_h = \frac{8G_1R}{2-\nu_1} \cdot \left(\frac{1 + \frac{1}{2} \cdot \frac{R}{H}}{1 + \frac{1}{2} \cdot \frac{R}{H} \cdot \frac{G_1}{G_2}} \right) \quad (2.134)$$

Faixa de Validade:

$$1 \leq \frac{H}{R} < 4$$

Rotacional (K_r):

$$K_r = \frac{8G_1 R^3}{3(1-\nu_1)} \cdot \left(\frac{1 + \frac{1}{6} \cdot \frac{R}{H}}{1 + \frac{1}{6} \cdot \frac{R}{H} \cdot \frac{G_1}{G_2}} \right) \quad (2.135)$$

Faixa de Validade:

$$\frac{3}{4} \leq \frac{H}{R} < 2$$

2.4.5. A consideração da Hipótese de Winkler

Pela hipótese de Winkler (ABMS/ABEF, 2002), tem-se que a aplicação de cargas, ou pressões verticais na superfície do solo, provoca deslocamentos proporcionais a essas pressões, somente no ponto de aplicação das mesmas, ignorando a continuidade do meio, sendo este ponto qualquer um da superfície de contato entre a fundação e o solo.

Isso permite a consideração de que a modelagem da interface solo-estrutura possa ser realizada por meio de molas com rigidez equivalentes e amortecedores.

A Figura 2.25 a seguir mostra como é o comportamento dos deslocamentos causados pela aplicação das cargas verticais em uma placa rígida. Observa-se que nas bordas as tensões são maiores.

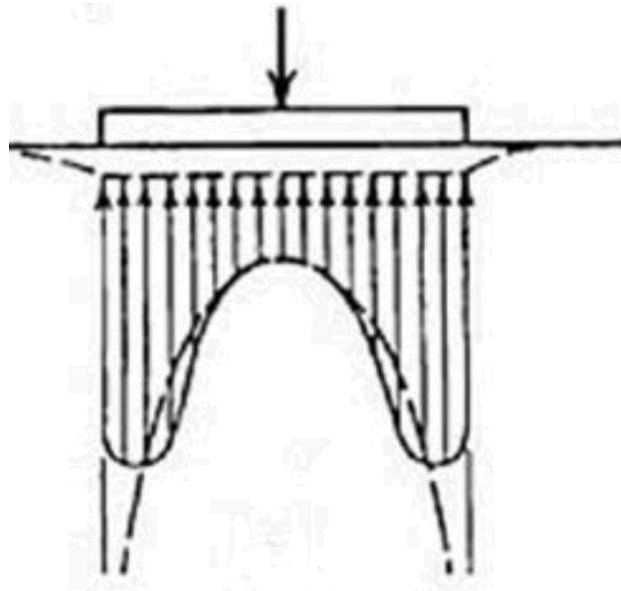


Figura 2.25 – Pressões de contato em placa rígida sobre argila (ABMS/ABEF, 2002).

A linha tracejada representa a curva teórica, que desconsidera a plastificação do solo, e é obtida pela Teoria da Elasticidade. São encontradas tensões maiores nas bordas, que tendem ao infinito, o que é incompatível com o caso físico real.

Já a linha cheia é a representação real da distribuição de tensões, que apresenta uma suavização da curva, causada pela plastificação do solo na região próxima às bordas, uma vez que não existe tensão infinita ou deslocamento infinito. Isso demonstra uma inconsistência desta hipótese.

A expressão (2.136), segundo a ABMS/ABEF (2002), permite calcular uma distribuição de pressões (p) sobre uma placa rígida retangular corrida de largura $2B$, com distância x do centro sujeita a uma pressão uniforme q em um ponto qualquer dela.

$$p = \frac{2q}{\pi \left[1 - \left(\frac{x}{B} \right)^2 \right]^{1/2}} \quad (2.136)$$

2.4.6. A consideração da Matriz de Rigidez Condensada do Solo

Conforme Santos e Guimarães (1988) propuseram e posteriormente Ribeiro (2014) e Albuquerque (2015) utilizaram, os parâmetros de rigidez do solo podem ser determinados considerando-se a modelagem do solo como um semiespaço homogêneo e elástico a partir de uma matriz das funções de impedância da interface fundação-solo.

Considerando o bloco de fundação como flexível, de tal modo que sejam consideradas suas deformações, o sistema possui infinitos graus de liberdade e, conseqüentemente, infinitos coeficientes a serem determinados para as funções de impedância.

Pode-se discretizar a superfície de contato da interface solo-estrutura em uma malha de elementos retangulares em que as propriedades podem ser consideradas como uniformes e concentradas em seu ponto geométrico central. Considera-se que o ponto central de cada elemento representa o valor médio dos deslocamentos no próprio elemento, em sua área de influência.

Calculam-se os deslocamentos em cada ponto da malha, com as fórmulas de Boussinesq (2.137), para o deslocamento vertical, baseada na Teoria da Elasticidade, e de Cerruti (2.138) para o deslocamento horizontal. Aplicam-se de forças unitárias no centro de cada elemento ponto, e calculam-se os deslocamentos em todos os demais pontos da malha discretizada, obtendo a influência da aplicação dessa força em cada ponto, distantes de r .

$$\delta_v = \frac{(1 - \nu)}{2 \cdot \pi \cdot r \cdot G} \quad (2.137)$$

$$\delta_h = \frac{(1 - \nu)}{2 \cdot \pi \cdot r \cdot G} \left(1 - \nu + \frac{x^2}{r^2} \nu \right) \quad (2.138)$$

Onde,

G – Módulo de elasticidade transversal;

r – Distância do ponto de aplicação da força unitária ao ponto central do elemento da malha analisado;

δ_v – Deslocamento vertical distante r de um ponto de aplicação de uma carga vertical unitária;

δ_h – Deslocamento horizontal distante r de um ponto de aplicação de uma carga horizontal unitária;

ν – Coeficiente de Poisson;

A partir dos deslocamentos calculados, é criada uma matriz de flexibilidade da interface fundação-solo e, em seguida, as matrizes de rigidez e de amortecimento na interface fundação-solo são obtidas pela inversão da matriz de flexibilidade.

É possível observar que as expressões (2.137) e (2.138) são funções apenas dos parâmetros do solo e da distância da aplicação da força r . Logo, como independe da frequência excitadora, essa solução é recomendável apenas para solos homogêneos.

Capítulo 3

3. DESENVOLVIMENTO DO PROBLEMA

Em um projeto de fundações de máquinas é imprescindível uma análise cuidadosa da resposta da fundação ao carregamento dinâmico previsto e calculado, como afirma Gazetas (1983), de forma que a amplitude de vibração seja limitada para assegurar o bom funcionamento da máquina, e evitar o desconforto dos trabalhadores próximos às máquinas.

Dentre as principais etapas do projeto de fundação de máquinas estão a determinação de características da estrutura, do solo e do equipamento, estas geralmente fornecidas pelo fabricante. Então a determinação dos parâmetros do solo, como o módulo de deformação transversal G , o módulo de elasticidade E , o coeficiente de Poisson e as frequências, são fundamentais no projeto para que sejam determinados os valores para as funções de impedância, para então modelar as molas e amortecedores corretamente.

Somente conhecendo corretamente a distribuição dessas funções é possível realizar a análise de maneira correta e precisa, visando um afastar as frequências de operação do equipamento das frequências próprias da estrutura, evitando assim o fenômeno da ressonância.

Quanto mais próxima a frequência de operação for da frequência natural da estrutura, os deslocamentos resultantes podem ser grandes, inviabilizando o projeto. Para que a máquina opere em uma faixa segura de frequências, procura-se alterar a rigidez, assim como a massa do conjunto de forma que sejam alteradas as suas frequências naturais, sendo novamente destacada a importância do conhecimento do comportamento das funções de impedância para a otimização do projeto.

3.1. Aplicação da Metodologia

Neste trabalho é analisada a interface entre o solo e a face inferior de uma fundação rígida, criando uma malha que representa a interface da superfície do solo, de modo que sejam criados elementos de interface estrutura-solo. Cria-se um modelo analítico de forma a representar a interface entre o terreno e o bloco da fundação. Apesar de que o exemplo apresentado seja de uma fundação retangular, o modelo pode ser aplicado a fundações de formato qualquer, como mostrado por OLIVEIRA (2021).

A quantidade de elementos utilizados para discretizar a interface da fundação com o solo tem relação com a maior frequência de excitação analisada, que neste caso será $\alpha_0 = 3,0$.

Wolf (1985), indica que são necessários no mínimo 6 pontos para representar adequadamente a curva senoide de uma onda cujo comprimento de onda é λ . Da equação fundamental das ondas, com V_S como a velocidade de onda de cisalhamento da primeira camada, em contato com a interface da fundação e ω como a frequência de excitação:

$$\lambda = \frac{2 \cdot \pi \cdot V_S}{\omega} \quad (3.1)$$

Assim, para que uma fundação seja bem representada, o maior lado Δ_l de um elemento deverá ter uma dimensão não superior a:

$$\Delta_l = \frac{\pi \cdot V_S}{3 \cdot \omega} \quad (3.2)$$

No exemplo aqui apresentado, é estudada uma fundação quadrada de dimensões 4,00m x 4,00m, com malha discretizada em 36 elementos quadrados de 0,667m, vide a Figura 3.1, que representa adequadamente a fundação de acordo com o critério apresentado por WOLF (1985). Para uma discretização mais refinada, a variação nos resultados é mínima, enquanto a demanda computacional aumenta significativamente.

Dados os modelos elaborados, são calculados, para essa geometria, os coeficientes de rigidez e os coeficientes de amortecimento de acordo com as formulações apresentadas. Esses valores são comparados aos valores obtidos por meio da formulação de WOLF (1985), a ser desenvolvida a seguir.

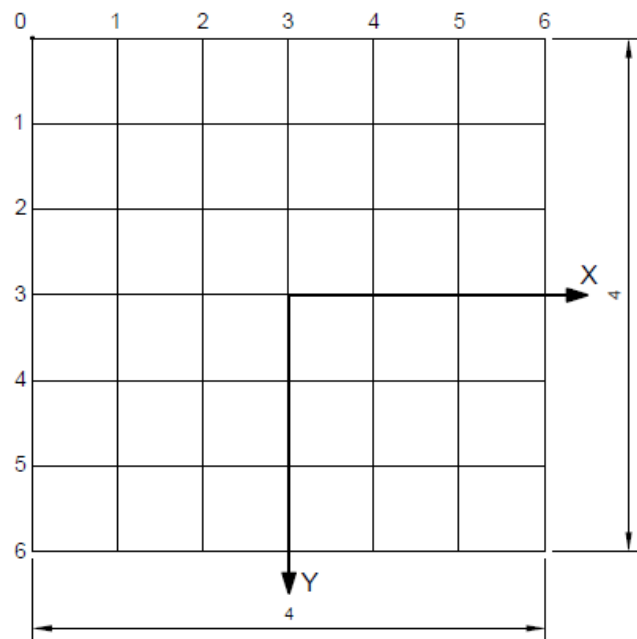


Figura 3.1 – Malha do modelo de estudo.

3.2. Modelagem do Solo para o Problema Estudado

O solo é modelado como um meio heterogêneo, seguindo comportamento elástico-linear, utilizando-se molas e amortecedores posicionados no centro de cada elemento da malha, formando um único elemento reticulado. Essa distribuição é realizada de maneira não uniforme, obedecendo à consideração da matriz condensada do solo.

Conforme essa hipótese, no exemplo de estudo, considera-se o solo composto por duas camadas, representado por uma matriz de rigidez que relaciona as forças aplicadas aos deslocamentos.

Analisando o problema harmônico, utilizar as variáveis complexas é o único meio de se representar matematicamente o amortecimento viscoso do solo, e o amortecimento histerético da estrutura.

Os amortecedores do tipo viscoso serão montados na matriz de maneira similar à matriz de rigidez. O amortecimento do material, que é da dissipação de energia na estrutura, do tipo histerético, pode ser considerado através dos modos complexos, por:

$$E^* = E(1 + \xi_{hist} \cdot i) \quad (3.3)$$

Analogamente, o módulo de deformação transversal complexo é definido por:

$$G^* = G(1 + 2 \cdot \xi_{hist} \cdot i) \quad (3.4)$$

Onde,

E^*, G^* – Módulos de deformação longitudinal e transversal complexos;

E, G – Módulos de deformação longitudinal e transversal;

ξ_{hist} – Fração de amortecimento histerético do material;

i – Unidade imaginária.

O amortecimento histerético pode ser obtido a partir da relação de equivalência a seguir:

$$\text{Força de amortecimento viscoso} = C \cdot \dot{u}(t) \quad (3.5)$$

$$\text{Força de amortecimento histerético} = 2 \cdot \xi_{hist} \cdot K \cdot u(t) \quad (3.6)$$

Como no problema harmônico $\dot{u}(t) = \omega \cdot u(t)$,

Igualando-se as duas expressões de forças, tem-se:

$$\xi_{hist} = \xi_{visc} \cdot \beta \quad (3.7)$$

Sendo que β é definido por:

$$\beta = \frac{\omega}{\bar{\omega}} \quad (3.8)$$

Com,

$$\bar{\omega} = \sqrt{\frac{K}{M}} \quad (3.9)$$

Onde,

$\bar{\omega}$ – Frequência circular própria

ω – Frequência de excitação

Quando β igual à 1, ocorre a ressonância. Então para β próximo de 1, o amortecimento viscoso se aproxima do histerético.

Visando realizar a análise numérica para um solo homogêneo típico, e de modo a comparar os resultados com os obtidos por Wolf (1985), adotam-se os seguintes valores na modelagem da camada superior:

$$\nu = 0,33; G_1^* = 1 + 4 \times 10^{-6}i; \rho = 1 \text{ Mg/m}^3; r_0 = 1 \mu\text{m}.$$

Para a camada inferior, do semiespaço, são adotados os mesmos valores de ν e ρ , porém é variado o valor de G_2 , de modo que diferentes condições de contorno sejam avaliadas para validar a aplicação da metodologia para três casos:

- 1) Solo homogêneo ($G_2/G_1 = 1$);
- 2) Solo heterogêneo com semiespaço infinito ($G_2/G_1 > 1$);
- 3) Semiespaço rígido com base em rocha ($G_2/G_1 = \infty$).

Especificamente para o caso 2, é adotada a razão $G_2/G_1 = 1,838$, para comparar ao caso 2 de Luco (1974).

3.3. Implementação

Conforme a metodologia apresentada por WOLF (1985), são calculadas as velocidades de onda de cisalhamento e de onda primária pelas expressões (2.1) e (2.2), respectivamente e, em seguida, são calculados os parâmetros $s(k_o)$ e $t(k_o)$, que

transformam o problema do domínio da frequência para o domínio do número de onda (k):

$$s(k_o) = -i \cdot \sqrt{1 - \frac{\omega^2}{V_P^2 \cdot k_o^2}} \quad (3.10)$$

$$t(k_o) = -i \cdot \sqrt{1 - \frac{\omega^2}{V_S^2 \cdot k_o^2}} \quad (3.11)$$

Onde,

i – Unidade imaginária;

k_o – Número de onda;

V_P – Velocidade de onda de principal ou de compressão;

V_S – Velocidade de onda secundária ou de cisalhamento;

ω – Frequência circular natural.

A função de flexibilidade que representa o deslocamento vertical devida a uma pressão unitária para o problema axissimétrico em solo homogêneo, em uma fundação circular é dada por:

$$F_{ww}(k) = \frac{1}{k_o \cdot G} \cdot \frac{-i \cdot s(k_o)(1 + t(k_o)^2)}{(1 - t(k_o)^2)^2 + 4 \cdot s(k_o) \cdot t(k_o)} \quad (3.12)$$

Para o solo heterogêneo, adicionalmente à metodologia utilizada para o solo homogêneo descrita no trabalho de Oliveira (2021), utiliza-se a matriz de rigidez dinâmica da camada do solo, apresentada por Wolf (1985) e aqui denominada por “S”, cujos termos são apresentados nas expressões (3.13) a (3.28) a seguir:

$$\bar{S}_{1,1} = \frac{1}{t} \cos(k_o s d) \cdot \sin(k_o t d) + s \cdot \sin(k_o s d) \cdot \cos(k_o t d) \quad (3.13)$$

$$\bar{S}_{1,2} = \frac{3-t^2}{1+t^2} (1 - \cos(k_o s d) \cdot \cos(k_o t d)) + \frac{1+2s^2t^2-t^2}{s \cdot t(1+t^2)} \sin(k_o s d) \cdot \sin(k_o t d) \quad (3.14)$$

$$\bar{S}_{1,3} = -s \cdot \sin(k_o s d) - \frac{1}{t} \sin(k_o t d) \quad (3.15)$$

$$\bar{S}_{1,4} = \cos(k_o s d) - \cos(k_o t d) \quad (3.16)$$

$$\bar{S}_{2,1} = \frac{3-t^2}{1+t^2} (1 - \cos(k_o s d) \cdot \cos(k_o t d)) + \frac{1+2s^2t^2-t^2}{s \cdot t(1+t^2)} \sin(k_o s d) \cdot \sin(k_o t d) \quad (3.17)$$

$$\bar{S}_{2,2} = \frac{1}{s} \sin(k_o s d) \cdot \cos(k_o t d) + t \cdot \cos(k_o s d) \cdot \sin(k_o t d) \quad (3.18)$$

$$\bar{S}_{2,3} = -\cos(k_o s d) + \cos(k_o t d) \quad (3.19)$$

$$\bar{S}_{2,4} = -\frac{1}{s} \sin(k_o s d) - t \cdot \sin(k_o t d) \quad (3.20)$$

$$\bar{S}_{3,1} = -s \cdot \text{sen}(k_o sd) - \frac{1}{t} \text{sen}(k_o td) \quad (3.21)$$

$$\bar{S}_{3,2} = -\cos(k_o sd) + \cos(k_o td) \quad (3.22)$$

$$\bar{S}_{3,3} = \frac{1}{t} \cos(k_o sd) \cdot \text{sen}(k_o td) + s \cdot \text{sen}(k_o sd) \cdot \cos(k_o td) \quad (3.23)$$

$$\bar{S}_{3,4} = \frac{t^2-3}{1+t^2} (1 - \cos(k_o sd) \cdot \cos(k_o td)) + \frac{t^2-2s^2t^2-1}{s \cdot t(1+t^2)} \text{sen}(k_o sd) \cdot \text{sen}(k_o td) \quad (3.24)$$

$$\bar{S}_{4,1} = \cos(k_o sd) - \cos(k_o td) \quad (3.25)$$

$$\bar{S}_{4,2} = -\frac{1}{s} \text{sen}(k_o sd) - t \cdot \text{sen}(k_o td) \quad (3.26)$$

$$\bar{S}_{4,3} = \frac{t^2-3}{1+t^2} (1 - \cos(k_o sd) \cdot \cos(k_o td)) + \frac{t^2-2s^2t^2-1}{s \cdot t(1+t^2)} \text{sen}(k_o sd) \cdot \text{sen}(k_o td) \quad (3.27)$$

$$\bar{S}_{4,4} = \frac{1}{s} \text{sen}(k_o sd) \cdot \cos(k_o td) + t \cdot \cos(k_o sd) \cdot \text{sen}(k_o td) \quad (3.28)$$

Com as expressões acima, é montada a matriz de rigidez dinâmica da camada em análise:

$$S = \frac{(1+t^2)k_o G^*}{D} \begin{bmatrix} \bar{S}_{1,1} & \bar{S}_{1,2} & \bar{S}_{1,3} & \bar{S}_{1,4} \\ \bar{S}_{2,1} & \bar{S}_{2,2} & \bar{S}_{2,3} & \bar{S}_{2,4} \\ \bar{S}_{3,1} & \bar{S}_{3,2} & \bar{S}_{3,3} & \bar{S}_{3,4} \\ \bar{S}_{4,1} & \bar{S}_{4,2} & \bar{S}_{4,3} & \bar{S}_{4,4} \end{bmatrix} \quad (3.29)$$

Com D definido por (3.29):

$$D = 2(1 - \cos(k_o sd) \cdot \cos(k_o td)) + \left(s \cdot t + \frac{1}{s \cdot t} \right) \text{sen}(k_o sd) \cdot \text{sen}(k_o td) \quad (3.30)$$

Em que,

k_o – Número de onda;

s – Parâmetro $s(k_o)$;

t – Parâmetro $t(k_o)$;

G^* – Módulo de deformação transversal complexo da camada;

Já a matriz do subespaço (S_{sub}), é definida por:

$$S_{sub} = k_o G^* \begin{bmatrix} \frac{i \cdot s(1+t^2)}{1+s \cdot t} & 2 - \frac{1+t^2}{1+s \cdot t} \\ 2 - \frac{1+t^2}{1+s \cdot t} & \frac{i \cdot t(1+t^2)}{1+s \cdot t} \end{bmatrix} \quad (3.31)$$

Em que,

i – Unidade imaginária;

k_o – Número de onda;

s – Parâmetro $s(k_o)$;

t – Parâmetro $t(k_o)$;

G^* – Módulo de deformação transversal complexo da camada;

Tratando-se de um modelo composto por duas camadas de solo, sendo a segunda camada a do subespaço, tem-se que a matriz de rigidez dinâmica é definida somando-se a matriz da expressão (3.31) aos termos 3,3; 3,4; 4,3 e 4,4 da matriz (3.29), conforme ilustrado na matriz abaixo.

$$S' = \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} & S_{1,4} \\ S_{2,1} & S_{2,2} & S_{2,3} & S_{2,4} \\ S_{3,1} & S_{3,2} & S_{3,3} + S_{sub\ 1,1} & S_{3,4} + S_{sub\ 1,2} \\ S_{4,1} & S_{4,2} & S_{4,3} + S_{sub\ 2,1} & S_{4,4} + S_{sub\ 2,2} \end{bmatrix} \quad (3.32)$$

Visando realizar o cálculo para o problema específico do semiespaço rígido, é possível simplificar a matriz acima, de forma a poupar tempo de processamento, eliminando os termos das linhas e colunas 3 e 4 da matriz (3.32) acima, gerando a matriz a seguir:

$$S'_{semispaço\ rígido} = \begin{bmatrix} S_{1,1} & S_{1,2} \\ S_{2,1} & S_{2,2} \end{bmatrix} \quad (3.33)$$

Após somados os devidos termos, a matriz de rigidez final é então invertida para se obter a matriz de flexibilidade. Para a formulação adotada, o termo de flexibilidade vertical, $F_{ww}(k)$, será dado pelo termo de índice 2,2 da matriz invertida ($S'^{-1}_{2,2}$).

Nota-se que para o caso do solo homogêneo, com $G_2/G_1 = 1$, esse valor de $F_{ww}(k)$ deve ser o mesmo do obtido pela expressão (3.12)

Conforme explicitado por Wolf (1985), por eficiência computacional, é assumida a carga atuante num subdisco circular permitindo o uso de coordenadas cilíndricas. Essa carga é expandida em série de Fourier na direção radial.

Assume-se que o carregamento vertical atuando em um círculo possui amplitude constante. A amplitude de deslocamento $w(r)$ que surge a uma distância r do centro do elemento circular é dada pela seguinte expressão:

$$w(r) = a \left[\int_{k_o=0}^{\infty} J_1(k_o \cdot a) \cdot F_{ww}(k_o) \cdot J_0(k_o \cdot r) dk_o \right] r_0 = g(r)r_0 \quad (3.34)$$

Onde,

a – Raio do elemento;

J_1 – Função de Bessel de ordem 1;

J_0 – Função de Bessel de ordem 0;

r_0 – Amplitude constante na direção vertical.

As funções de Bessel realizam a transformação do estado plano para o estado axissimétrico. E como k_o , o número de onda, varia de 0 a infinito, todos os tipos de onda são capturados nesta formulação (WOLF, 1985).

Em um ponto qualquer na superfície entre a fundação e o solo simbolicamente denotado por 's', a carga e os deslocamentos nos nós da fundação podem ser descritos em termo das funções de interpolação da amplitude do carregamento $L(s)$ e pela função de influência de flexibilidade do semiespaço estratificado $g(s)$, conforme definido em (3.34). O deslocamento na fundação é distribuído na superfície pela função de forma $N(s)$.

Tem-se que a flexibilidade G_f multiplicada pela intensidade da carga aplicada p deve ser igual à rigidez T , da interface, vezes o deslocamento u , o que resulta na seguinte relação que representa a ação que a fundação impõe sobre o solo e a reação do solo sobre a fundação, para compatibilizar as tensões e deformações na interface

$$G_f \cdot p = T \cdot u \quad (3.35)$$

Wolf (1985) define a matriz de flexibilidade G_f e de rigidez T da seguinte forma:

$$G_f = \int_S L(s)^T \cdot g(s) ds \quad (3.36)$$

$$T = \int_S L(s)^T \cdot N(s) ds \quad (3.37)$$

E daí tem-se que, considerando as funções de interpolação, a rigidez dinâmica é dada por:

$$Sz = T^T \cdot G_f^{-1} \cdot T \quad (3.38)$$

Para os cálculos, são consideradas as hipóteses de que o carregamento é uniformemente distribuído na fundação, e de que a fundação é rígida. Ao realizar o cálculo para um único elemento circular, a operação é escalar. Considerando essas hipóteses, $L(s)$ e $N(s)$ são unitários, e T é uma função no domínio, que resulta em πa^2 e G_f depende de $g(s)$, associado à $w(r)$.

Portanto, pode-se reescrever G_f e T conforme as equações (3.39) e (3.40) a seguir:

$$G_f = \int_0^a 2\pi \cdot r \cdot g(r) dr \quad (3.39)$$

$$T = \pi \cdot a^2 \quad (3.40)$$

A partir de (3.38), pode-se escrever o termo de rigidez dinâmica, substituindo (3.39) e (3.40) que é dado por:

$$S_z = \frac{\pi \cdot a^4}{2 \int_0^a w(r) \cdot r \, dr} \quad (3.41)$$

A parte real da equação (3.41) é associada ao termo de rigidez (k_{S_z}), enquanto a parcela imaginária é relacionada ao amortecimento (c_{S_z}). Para calcular os coeficientes adimensionais, Wolf (1986) propõe a rigidez estática definida por:

$$K_s = 1,82 \cdot \pi \cdot G \cdot a \quad (3.42)$$

Pode-se escrever essas funções conforme a seguir.

$$k_{S_z} = \text{Real}(S_z) \quad (3.43)$$

$$c_{S_z} = \frac{\text{Imaginário}(S_z)}{a_0} \quad (3.44)$$

Os pontos nodais da matriz condensada do solo localizam-se no centro de cada elemento da malha. Uma vez que é possível considerar como de boa qualidade o nível de refinamento da malha que representa a fundação, simplificada, é considerado o ponto central como representativo

Admite-se que os deslocamentos verticais, em cada área de influência do elemento, são aproximadamente constantes e iguais ao do ponto nodal. Isto é, considera-se o valor do deslocamento no ponto nodal como valor médio do deslocamento em cada elemento, em sua área de influência conforme a Figura 3.2 a seguir.

Para se obter a matriz de flexibilidade, a interface entre solo e fundação é discretizada em elementos quadrados de lado com dimensões de 0,667 m, conforme supramencionado.

A área é calculada como de uma circunferência de raio equivalente, de modo que resulte em uma área igual ao do elemento quadrado da malha.

Considera-se o desacoplamento entre os três graus de liberdade translacionais de cada nó para obter-se a matriz de flexibilidade, com o cálculo apenas dos deslocamentos no sentido vertical, de forma simplificada.

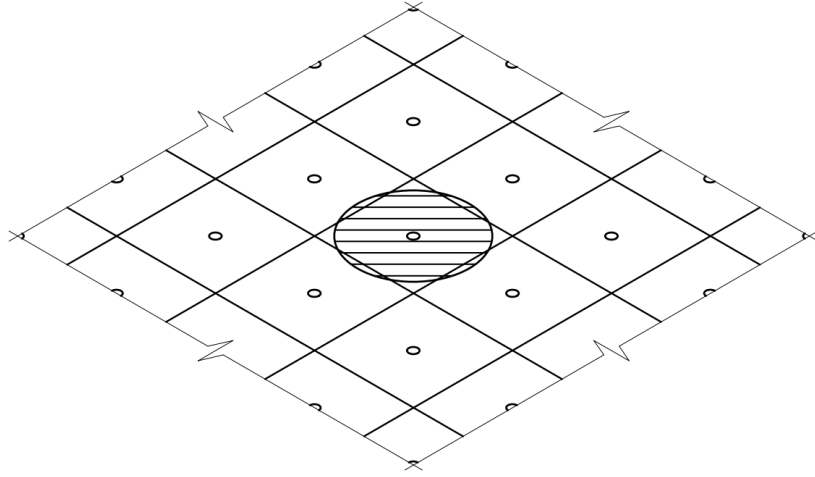


Figura 3.2 – Área de influência de um ponto nodal.

Supondo um elemento circular de área aproximadamente igual ao do elemento da malha, o deslocamento vertical distante r do ponto de aplicação de uma carga vertical P , é calculado pela expressão dos deslocamentos verticais (3.34) de Wolf (1985), cuja parte real é equivalente ao valor obtido para deslocamentos verticais pela fórmula de Boussenesq, em (2.137).

Esta fórmula é para uma carga uniforme unitária em um círculo de raio a , ou seja, para se ter os termos de flexibilidade fora da diagonal, deve-se dividir por $\pi \cdot a^2$.

É preciso ressaltar que, segundo a teoria, os deslocamentos para os elementos da diagonal principal tenderiam ao infinito, o que não seria correto. Ou seja, a expressão (2.137) é válida apenas para elementos fora da diagonal principal.

Nestes elementos, para se obter a correlação entre a força aplicada em um ponto e os deslocamentos provocados em sua própria área de influência, deve ser empregada a solução da rigidez dinâmica de placas circulares rígidas sob a ação de forças verticais (expressão (3.41), decomposta em (3.43) e (3.44)), sendo os termos diagonais numericamente iguais ao inverso desses coeficientes.

Ou seja, os termos de flexibilidade na diagonal devem ser obtidos aplicando uma carga unitária na rigidez complexa obtida com o raio real, com k_{sz} e c_{sz} obtidos a partir do cálculo da rigidez dinâmica. Então, para se obter uma matriz de flexibilidade $m \times n$ utiliza-se a seguinte regra:

$$f_{v_{m,n}} = \begin{cases} \frac{w(r_{m,n})}{\pi \cdot a^2} & \text{se } m \neq n \\ \frac{1}{k_{sz} + i \cdot \omega \cdot c_{sz}} & \text{se } m = n \end{cases} \quad (3.45)$$

A matriz de rigidez do solo é obtida, então, pela parte real da sua matriz de flexibilidade invertida. Já a matriz de amortecimento é obtida da mesma forma, porém com a parte imaginária da matriz inversa da matriz de flexibilidade.

Esse processo trata-se da criação de um elemento de interface, empregando um subdivisão da superfície entre solo e estrutura, o que torna esse método mais eficiente do que o Método dos Elementos Finitos na análise de problemas de fundações tridimensionais.

Genericamente, pode-se mostrar o procedimento da montagem da matriz de flexibilidade pela Figura 3.3 a seguir, em que é representado, para a aplicação de uma carga unitária, o campo de deslocamentos no próprio ponto de aplicação e nos demais pontos, a ele adjacentes. O deslocamento é maior no próprio ponto de aplicação e conforme se afasta desse ponto, os deslocamentos são menores.

Para a carga unitária, o deslocamento é equivalente à flexibilidade e monta-se a matriz de flexibilidade com os coeficientes conforme a Figura 3.3, até completar as m linhas e n colunas da matriz. A matriz de flexibilidade f_v é invertida e sua parte real corresponde à rigidez, enquanto a parcela imaginária corresponde ao amortecimento.

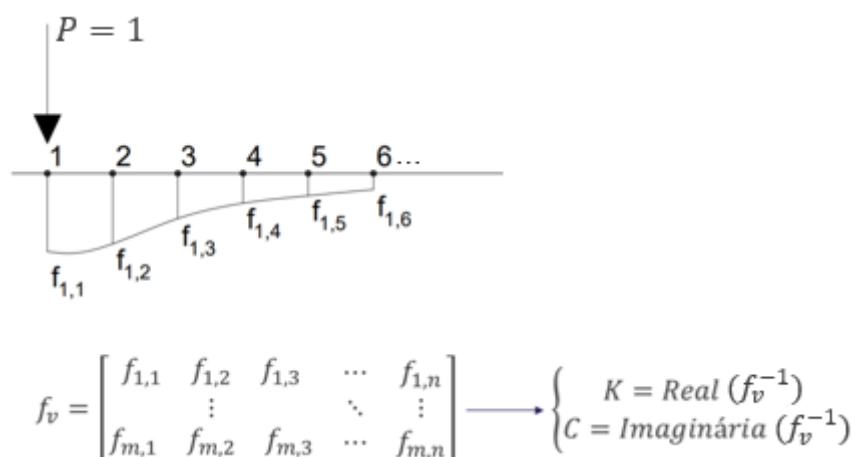


Figura 3.3 – Montagem da matriz de flexibilidade.

Ao multiplicar as matrizes de rigidez e amortecimento por um vetor de forças cujos elementos são todos iguais a 1, devido à carga unitária, o resultado equivale à soma das linhas da matriz de rigidez. Dessa forma, obtém-se a rigidez total de cada elemento da malha discretizada e, ao somar todos esses valores, obtém-se a rigidez total do sistema.

Isso vale para o amortecimento: a soma das linhas da matriz de amortecimento fornece o amortecedor de cada elemento, e a soma destes valores resulta no amortecimento total.

Os resultados são avaliados por meio de uma análise adimensional para viabilizar melhores comparações e de caráter mais geral. Wolf (1985) decompõe a matriz de rigidez dinâmica conforme a seguinte expressão:

$$[S_z] = [K](k + i \cdot a_0 \cdot c) \quad (3.46)$$

A matriz K contém os coeficientes de rigidez estática, ou seja, a rigidez obtida para frequência excitadora nula ($\omega = 0$). As matrizes k e c são os coeficientes de mola e amortecimentos adimensionais.

As expressões dos parâmetros adimensionais de k e c , relacionam respectivamente a rigidez e amortecimento no sentido vertical em relação aos valores obtidos conforme as expressões (3.47) e (3.48) de Luco (1974). Importante é ressaltar que o coeficiente de amortecimento adimensional deve ser dividido não só pelo valor teórico, mas por ω , uma vez que a parte complexa é multiplicada por esse valor.

$$k = \frac{K_{calculado}}{K_{teórico}} \quad (3.47)$$

$$c = \frac{C_{calculado}}{\omega \cdot C_{teórico}} \quad (3.48)$$

Nas expressões acima, $K_{teórico}$ é expresso por (2.116) e $C_{teórico}$ é expresso por (3.49).

$$C_{teórico} = \frac{4 \cdot r^2}{1 - \nu} \cdot \sqrt{\rho \cdot G} \quad (3.49)$$

Esses valores de k e c são calculados com a variação do parâmetro a_0 que representa a frequência adimensional.

$$a_0 = \frac{\omega \cdot r}{V_S} \quad (3.50)$$

Onde,

a_0 – Frequência adimensional;

ω – Frequência circular natural;

r – Raio equivalente de um elemento da malha;

V_S – Velocidade de onda de cisalhamento.

Para analisar a amplitude do deslocamento vertical, é possível adimensionalizar essa grandeza, dividindo o resultado da expressão (2.80) pelo deslocamento estático $u_{st} = F_0/K$ para se obter o fator de amplificação dinâmica, em que a fração de amortecimento crítico ε dada por (2.102) e β dado por (3.8), no caso de coeficientes independentes da frequência:

$$D = \frac{u}{u_{st}} = \frac{1}{\sqrt{(1 - \beta^2)^2 + (2 \cdot \beta \cdot \varepsilon)^2}} \quad (3.51)$$

Para o caso em estudo, essa equação deve ser reescrita, conforme sugerido por Wolf (1985), da seguinte forma com a incorporação dos coeficientes adimensionais de rigidez, amortecimento e a frequência adimensional.

$$\frac{w}{w_{st}} = \frac{1}{1 - \frac{a_0^2 \cdot \bar{m}}{\pi} \cdot [k + i \cdot a_0 \cdot c]} \quad (3.52)$$

Em que $\bar{m}$ é a massa adimensional do sistema, dada por:

$$\bar{m} = \frac{m}{\rho \cdot a \cdot b} \quad (3.53)$$

Com m sendo a massa da fundação, ρ a massa específica, e a sendo a metade do lado de maior dimensão, e b a metade do lado de menor dimensão, em um elemento retangular.

3.4. Algoritmo de análise

Para a realização dos cálculos, é desenvolvido um algoritmo utilizando a linguagem C++, devido à sua eficiência e desempenho, favorecendo uma melhor precisão e velocidade nos cálculos.

A sequência lógica do algoritmo de resolução do problema é apresentada a seguir:

1. Apresentação do programa;
2. Leitura dos dados das camadas da subsuperfície (G_1 , ν_1 , ρ_1 , ξ_1 , G_2 , ν_2 , ρ_2 , ξ_2);
3. Leitura dos dados da geometria da fundação (A , B , Divisões em x , Divisões em y);

4. Inicialização do arquivo de saída para registro dos resultados gerados pelo algoritmo;
5. Cálculo dos parâmetros para a consideração do amortecimento de histerese para a primeira camada e para o subespaço ($G_1^*, c_{s_1}^*, c_{p_1}^*, G_2^*, c_{s_2}^*, c_{p_2}^*$);
6. Rotina que realiza o cálculo para todos os valores de frequência adimensional (α_0) determinados. Ela segue a sequência:
 - 6.a. Determinação da dimensão do elemento da malha, do raio da fundação e de cada elemento, além das funções $s(k_o)$ e $t(k_o)$;
 - 6.b. Cálculo da função da amplitude de deslocamento vertical, $w(r)$, com implementação de rotina para variar o valor da distância do ponto de aplicação da carga, r , de 0 a 1 para gerar os gráficos de $w(r) \times r$;
 - 6.c. Cálculo do parâmetro da rigidez dinâmica S_z ;
 - 6.d. Cálculo dos coeficientes de rigidez (k_{S_z}) e amortecimento (c_{S_z}) como preparação para entrada na matriz de flexibilidade;
 - 6.e. Montagem da matriz de flexibilidade, $f_{v_{m,n}}$, e inversão;
 - 6.f. Separação da matriz em parte real e imaginária com o cálculo da rigidez e do amortecimento do problema;
 - 6.g. Cálculo dos valores teóricos da literatura existente;
 - 6.h. Cálculo dos coeficientes adimensionais;
7. Criação das tabelas no arquivo de saída com os resultados obtidos;
8. Cálculo da amplificação dinâmica e impressão dos resultados no arquivo de saída;
9. Finalização e fechamento dos arquivos abertos, com encerramento das operações do programa.

O programa solicita como valores de entrada:

- a. Módulo de Elasticidade Transversal das duas camadas;
- b. Coeficiente de Amortecimento Histerético das duas camadas;
- c. Coeficiente de Poisson das duas camadas;
- d. Massa Específica das duas camadas;
- e. Profundidade da Primeira Camada de Solo;
- f. Dimensão 'A' e 'B' da fundação 'A x B';
- g. Número de Divisões da malha da fundação em x;
- h. Número de Divisões da malha da fundação em y;
- i. Frequências adimensionais ' α_0 ' a serem analisadas;
- j. Número de onda limite para análise.

E são criadas 4 tabelas no arquivo de saída:

1. Parte Real de $W(r)$ para todas as frequências adimensionais analisadas
2. Parte Imaginária de $W(r)$ para todas as frequências adimensionais analisadas
3. Coeficientes adimensionais calculados
4. Amplificação dinâmica para todas as frequências adimensionais analisadas, com $\bar{m} = 1$, para representar fundação leve, e $\bar{m} = 5$, para representar fundação pesada.

Capítulo 4

4. EXEMPLOS DE APLICAÇÃO

4.1. Validação da função da influência de flexibilidade

Como verificação inicial, do código e da metodologia, foi avaliada a consistência dos resultados com problemas abordados previamente por SOLDAN (1999) e por SANTOS (2018), para o problema do estado plano de tensões, e que foi descrito primeiramente por WOLF (1985). Aqui o problema é tratado para o problema axissimétrico, e considera uma camada de solo sobre um subespaço a partir dos seguintes dados de entrada:

- Módulo de elasticidade transversal da camada superior: $G_1 = 20000 \text{ kPa}$;
- Coeficiente de Poisson da camada superior: $\nu_1 = 0,33$;
- Coeficiente de amortecimento histerético da camada superior: $\xi_1 = 0,001$;
- Massa específica da camada superior: $\rho_1 = 2 \text{ t/m}^3$;
- Módulo de elasticidade transversal do subespaço: $G_2 = 80000 \text{ kPa}$;
- Coeficiente de amortecimento histerético do subespaço: $\xi_2 = 0,001$;
- Coeficiente de Poisson do subespaço: $\nu_2 = 0,33$;
- Massa específica do subespaço: $\rho_2 = 2 \text{ t/m}^3$;
- Frequência circular de análise: $\omega = 35 \text{ rad/s}$;
- Espessura da primeira camada: $d = 20 \text{ m}$;
- Dimensão do raio da área de aplicação da carga: $R = 0,0625d = 1,25 \text{ m}$;
- Número de onda limite: $k_{olim} = 0,6 \text{ rad/m}$;
- Passo incremental do número de onda: $\Delta_{k_o} = 0,001 \text{ rad/m}$.

A comparação, é realizada a partir da adimensionalização dos parâmetros apresentados, com verificação da função de influência de flexibilidade $FWW(k_o)$. As figuras a seguir representam a variação das componentes real e imaginária da função avaliada.

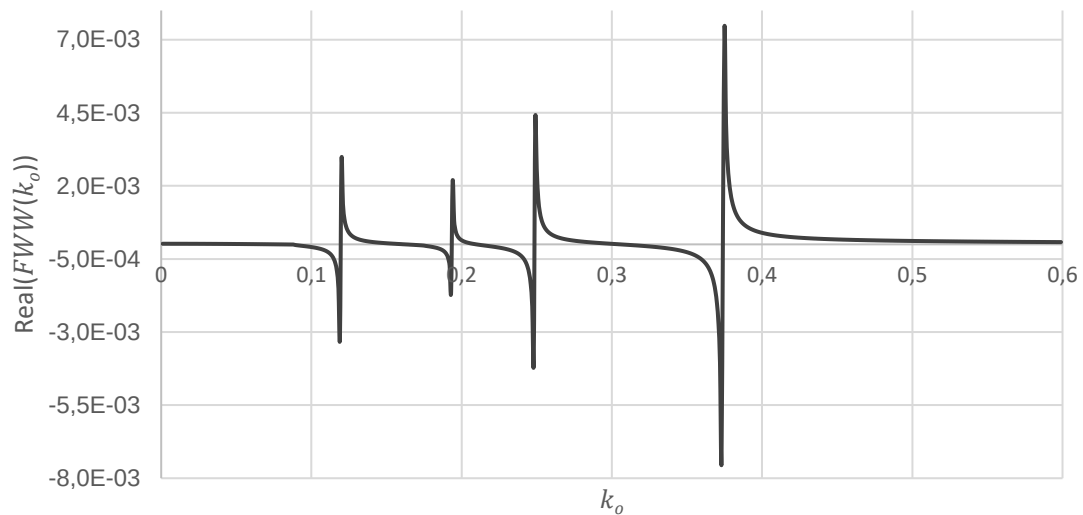


Figura 4.1 – Parte real da função de influência de flexibilidade vertical FWW .

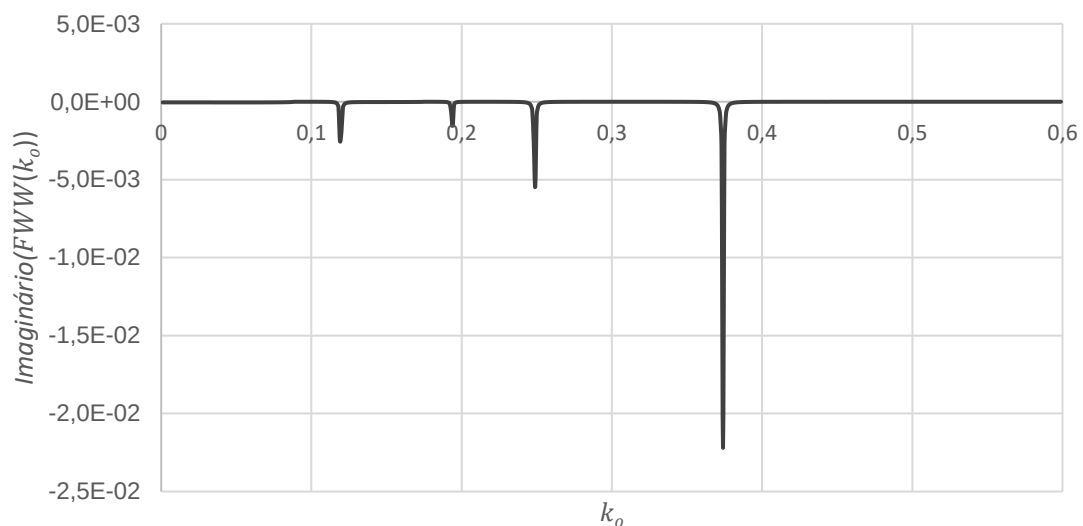


Figura 4.2 – Parte imaginária da função de influência de flexibilidade vertical FWW .

Os resultados obtidos foram compatíveis com os valores obtidos pelos demais autores citados.

Com base no exemplo mencionado, foram avaliadas as componentes da amplitude do deslocamento vertical. Os valores obtidos estão na mesma ordem de grandeza dos testes realizados pelos mesmos autores. No entanto, observa-se que os deslocamentos são maiores para menores valores de r .

Essa diferença é coerente com o fato de que, no problema de estudo, foi adicionada uma nova dimensão, o que altera a natureza do problema de um plano para um cenário tridimensional. A inclusão da flexibilidade fora do plano, em um problema axissimétrico, pode indicar um aumento na flexibilidade da fundação, refletindo assim um comportamento mais complexo e flexível do sistema analisado.

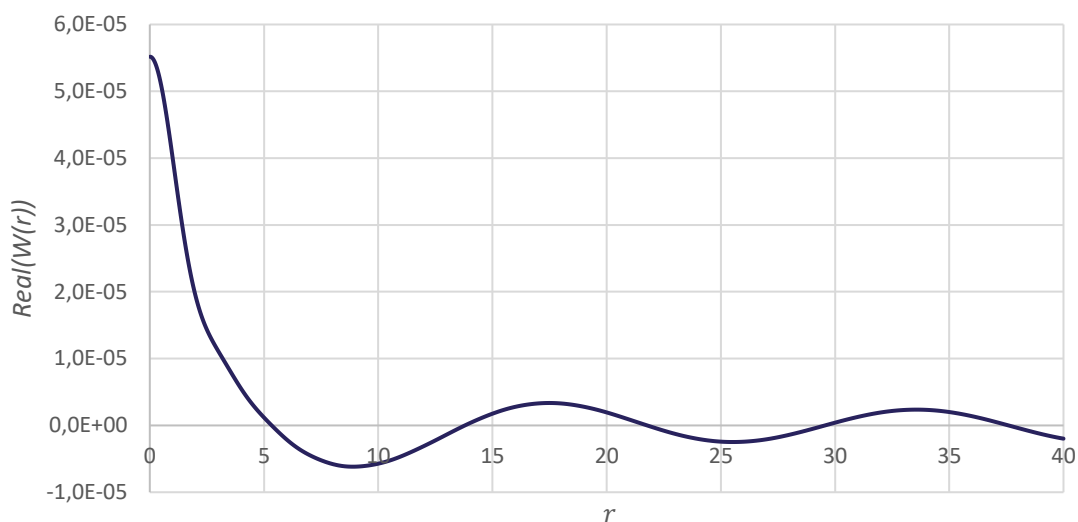


Figura 4.3 – Parte real da amplitude de deslocamento vertical $W(r)$.

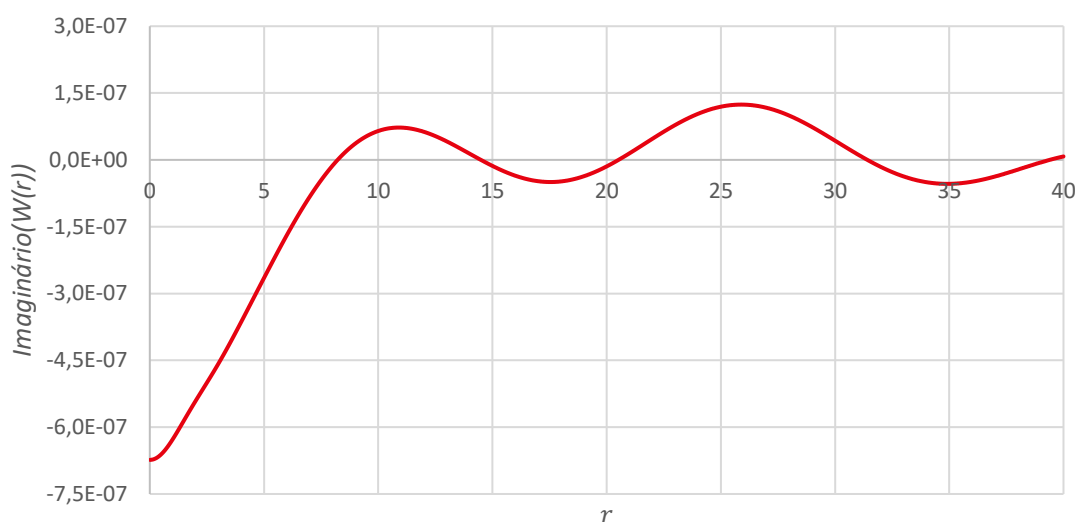


Figura 4.4 – Parte imaginária da amplitude de deslocamento vertical $W(r)$.

4.2. Distribuição das funções de impedância na interface solo-fundação

A partir dos dados iniciais descritos no item 3.2 e da formulação detalhada, são obtidos e avaliados a seguir os resultados para a rigidez e amortecimento verticais referentes ao exemplo de estudo.

A fim de realizar uma comparação da distribuição das funções de impedância ao longo da fundação, são representados a seguir os valores para os coeficientes de rigidez e de amortecimento nos três casos de estudo, cada um para duas frequências distintas: considerando o caso estático, com frequências nulas ($\omega = 0, a_0 = 0$); e para frequências de 1,33 rad/s ($a_0 = 3$).

Adicionalmente, são determinadas seções de análise nas regiões das bordas (Seção 1-1) e mais ao centro do modelo (Seção 2-2), conforme indicado nas figuras a seguir.

4.2.1. Caso 1: Solo Homogêneo

Para o caso 1, do solo homogêneo, é possível observar entre a Figura 4.5 e a Figura 4.12 que as funções de impedância apresentam maiores valores nas bordas da fundação, e com o aumento da frequência excitadora os valores de rigidez, próximo às bordas, e de amortecimento aumentam em módulo. Entretanto, a rigidez reduz de valor nas regiões centrais e pode inverter de sinal.

É notável que o efeito de enrijecimento nas extremidades é aplicável ao amortecimento, para o caso estático. Além disso, observa-se que o amortecimento e rigidez possuem comportamentos distintos para diferentes frequências.

Para o caso estático, é observada uma simetria em K e C , que indica uma distribuição uniforme das propriedades mecânicas na interface solo-fundação, o que sugere que o solo reage de maneira uniforme na área de contato, o que é típico de um comportamento de quando não há vibrações externas significativas.

O maior valor das funções encontra-se nos cantos e diminuem ao centro. O amortecimento é baixo em toda a interface, o que é esperado no caso estático, já que não há forças dinâmicas significativas.

Para a condição dinâmica de uma maior frequência excitadora, com $\alpha_0 = 3,0$, a rigidez apresenta uma alteração mais significativa, com valores negativos nas regiões centrais, resultando em uma distribuição mais complexa. Esse comportamento pode levar à instabilidade em frequências específicas e precisa ser cuidadosamente considerado no projeto, para evitar amplificação irreal de vibrações. Por outro lado, as bordas continuam a absorver grande parte da carga.

O amortecimento possui um aumento considerável em comparação ao caso estático, o que é indicativo de que, em frequências mais altas, o sistema dissipa mais energia devido às vibrações, o que é esperado. O mais importante é que a função agora se concentra principalmente nas regiões centrais, com alguns picos nas bordas, apesar de as distribuições das funções se manterem simétricas.

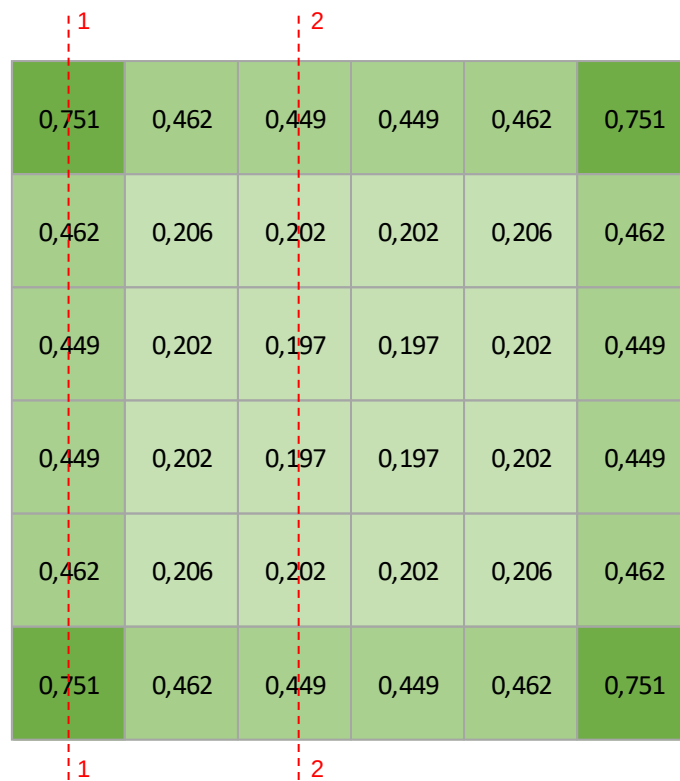


Figura 4.5 – Distribuição dos Coeficientes de Rigidez para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=0$.

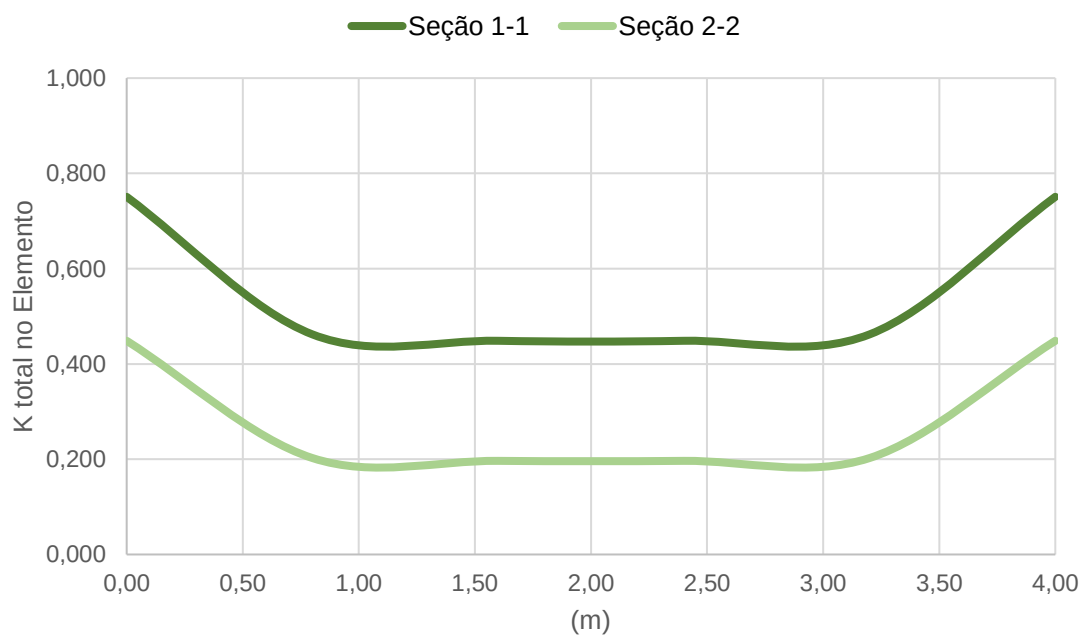


Figura 4.6 – Variação da Rigidez para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=0$.

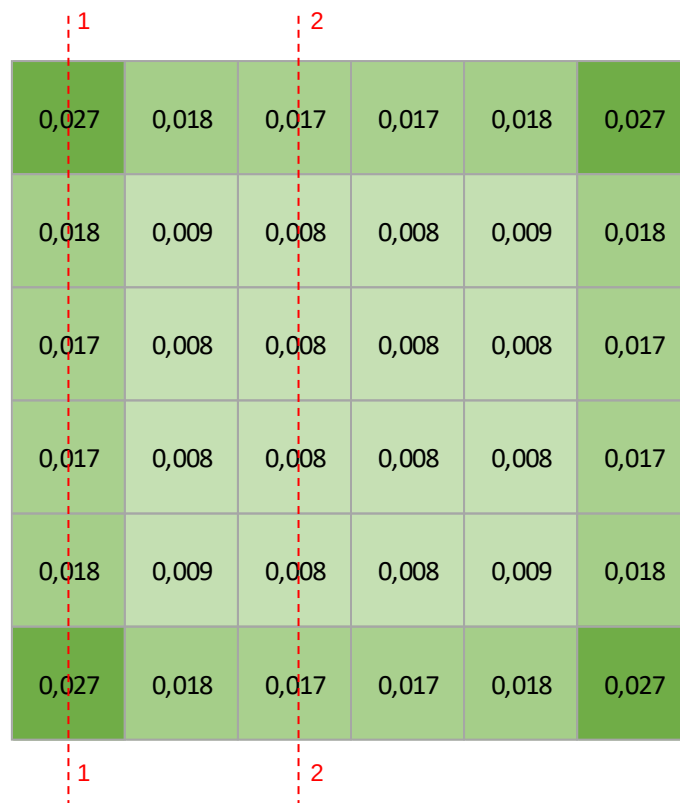


Figura 4.7 – Distribuição dos Coeficientes de Amortecimento para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=0$.

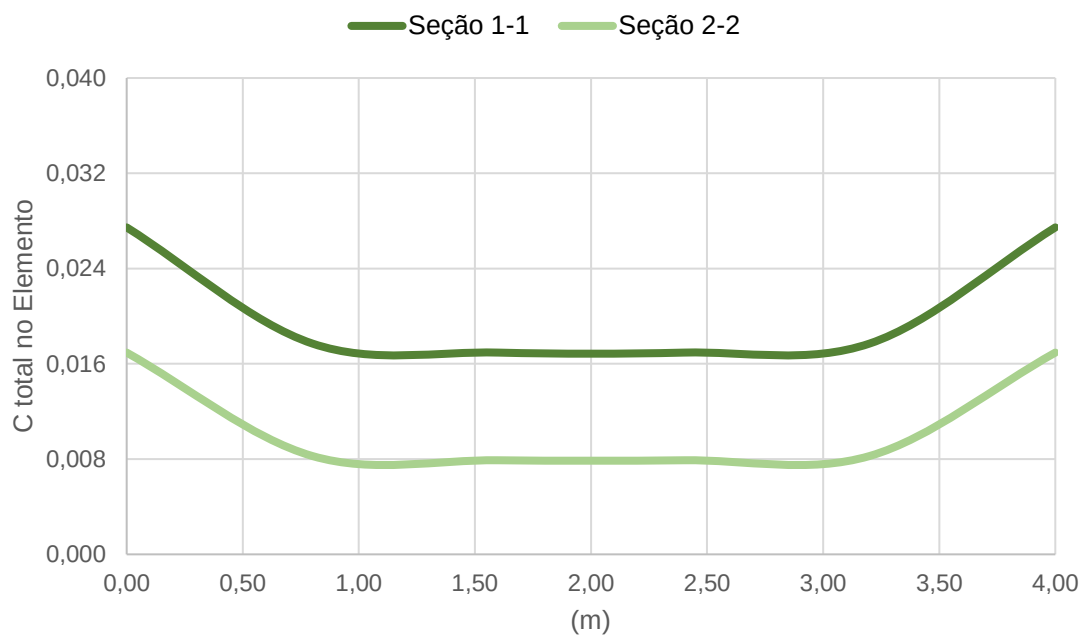


Figura 4.8 – Variação do Amortecimento para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=0$.

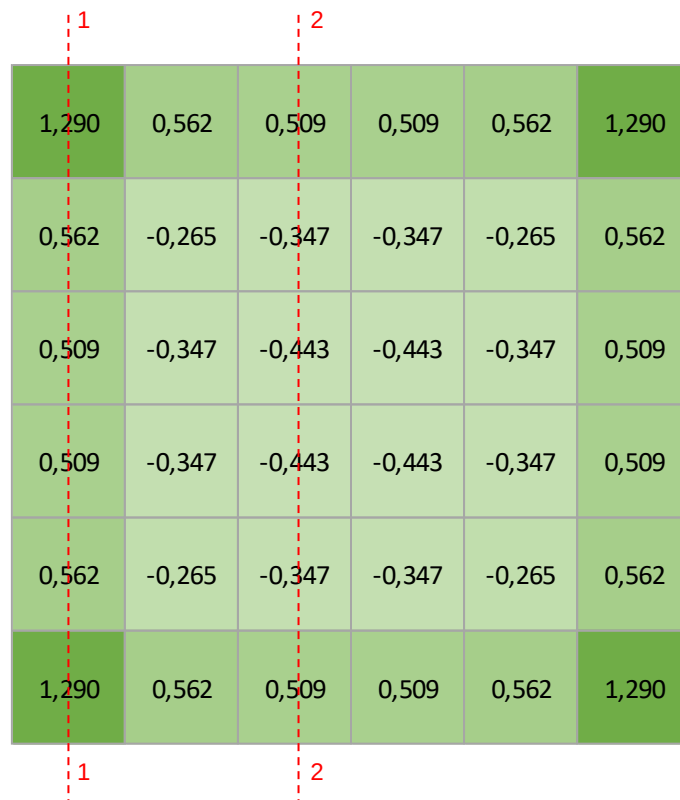


Figura 4.9 – Distribuição dos Coeficientes de Rigidez para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=3,0$.

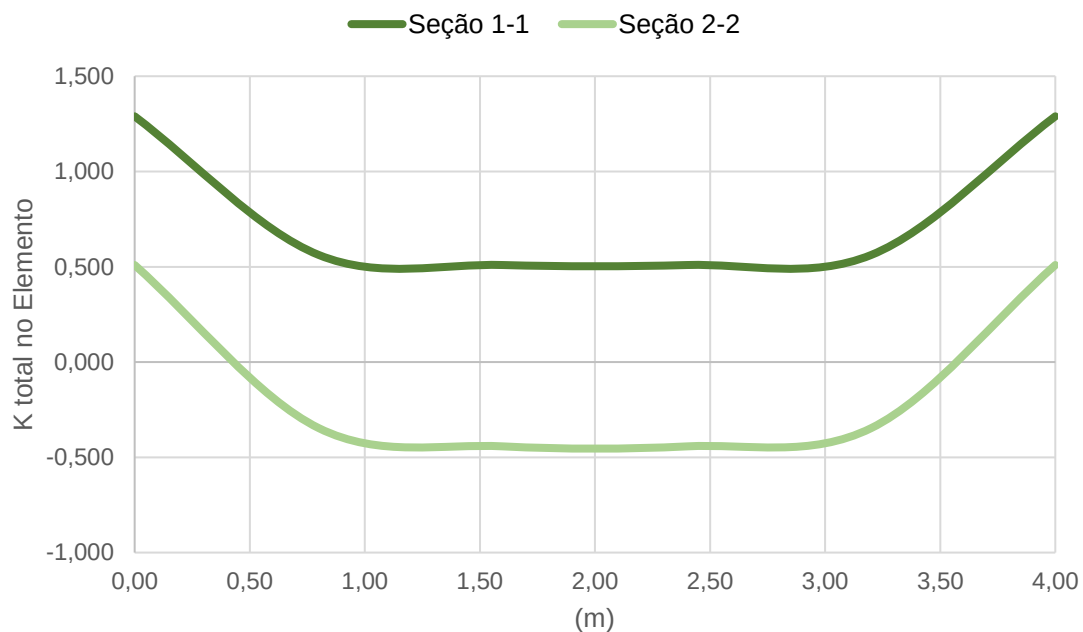


Figura 4.10 – Variação da Rigidez para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=3,0$.



Figura 4.11 – Distribuição dos Coeficientes de Amortecimento para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=3,0$.

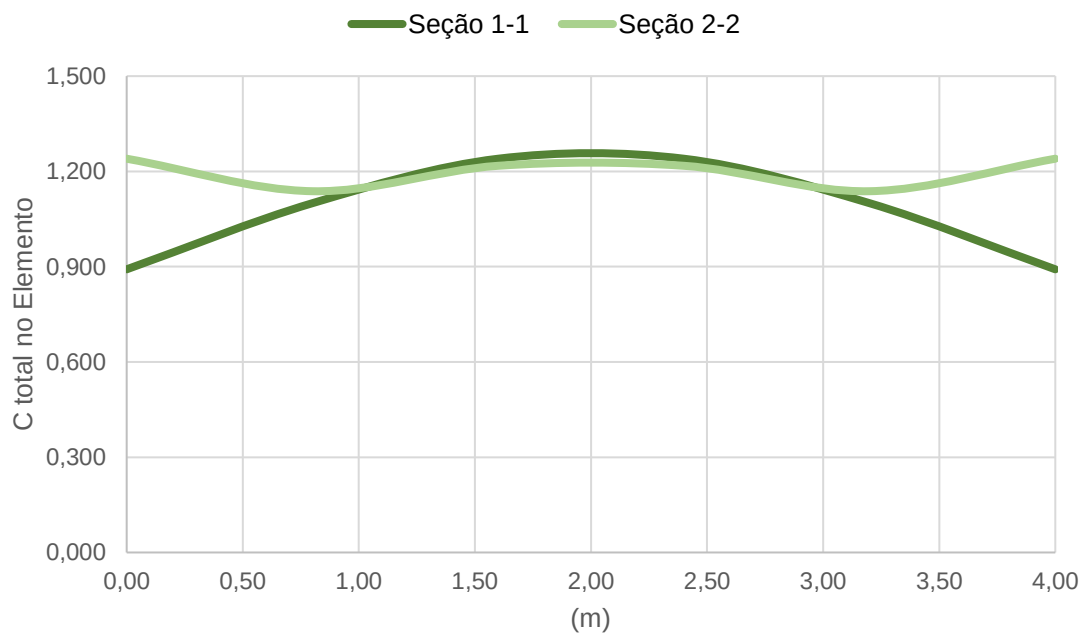


Figura 4.12 – Variação do Amortecimento para Caso 1 – Solo Homogêneo ($G_2/G_1 = 1$) e com $a_0=3,0$.

4.2.2. Caso 2: Solo Heterogêneo

Considerando agora o segundo caso, para o solo heterogêneo, com $G_2/G_1 = 1,838$, verifica-se entre a Figura 4.13 e a Figura 4.20 que a variação da rigidez e do amortecimento é mais acentuada do que para o primeiro caso, que considera o solo como homogêneo. Entretanto, também se pode concluir que as funções apresentam maiores valores nas bordas da fundação, para o caso estático.

Com o aumento da frequência excitadora, os valores de amortecimento e valores de rigidez próximo das bordas aumentam, porém a rigidez ao centro reduz de valor e inverte de sinal. É possível perceber que, para o amortecimento, há uma concentração de maiores valores ao centro da fundação, e nas bordas centrais, fora dos cantos, para maiores frequências.

Essa concentração do amortecimento ao centro é mais acentuada do que para o caso homogêneo, com resposta mais localizada devido às diferentes camadas que compõem o solo.

Em relação ao Caso 1, do solo homogêneo, o Caso 2 apresenta valores de rigidez relativamente maiores, porém o amortecimento apresenta valores menores. As distribuições das funções continuam simétricas, porém com inversão de sinal para rigidez e picos nas bordas para o amortecimento.

Pode-se concluir que o efeito de enrijecimento nas extremidades não é aplicável ao amortecimento, e que as funções possuem comportamentos distintos para diferentes frequências. O solo heterogêneo apresenta uma maior variabilidade nas distribuições de rigidez e amortecimento quando comparado ao homogêneo, ao longo da fundação, especialmente no regime dinâmico. As camadas com diferentes propriedades afetam a resposta da fundação, levando a um comportamento menos previsível.

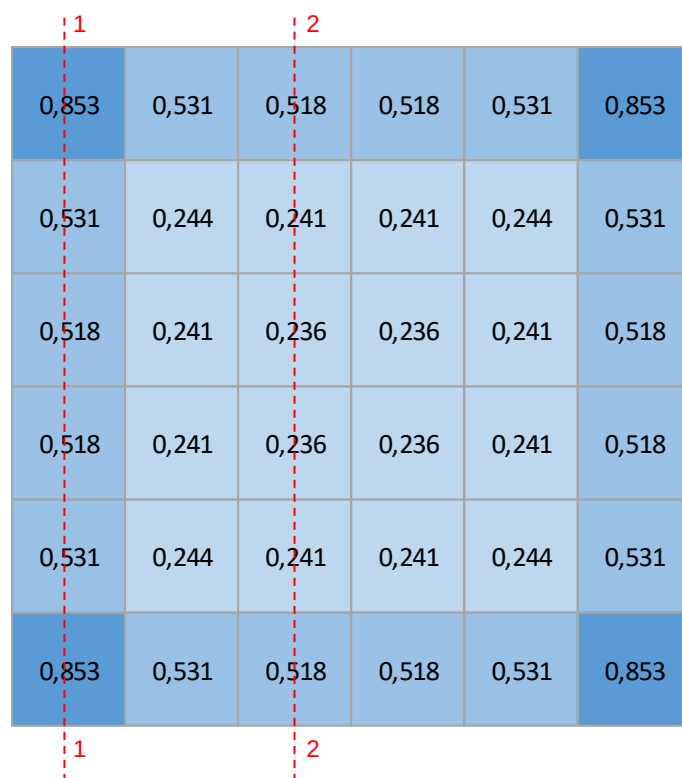


Figura 4.13 – Distribuição dos Coeficientes de Rigidez para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=0$.

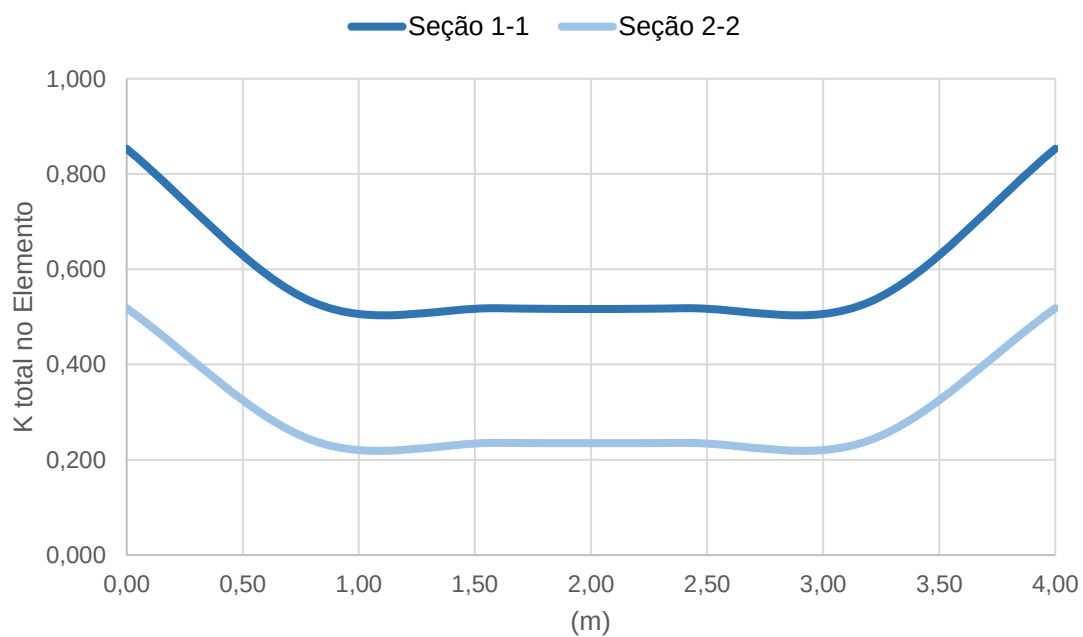


Figura 4.14 – Variação da Rigidez para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=0$.

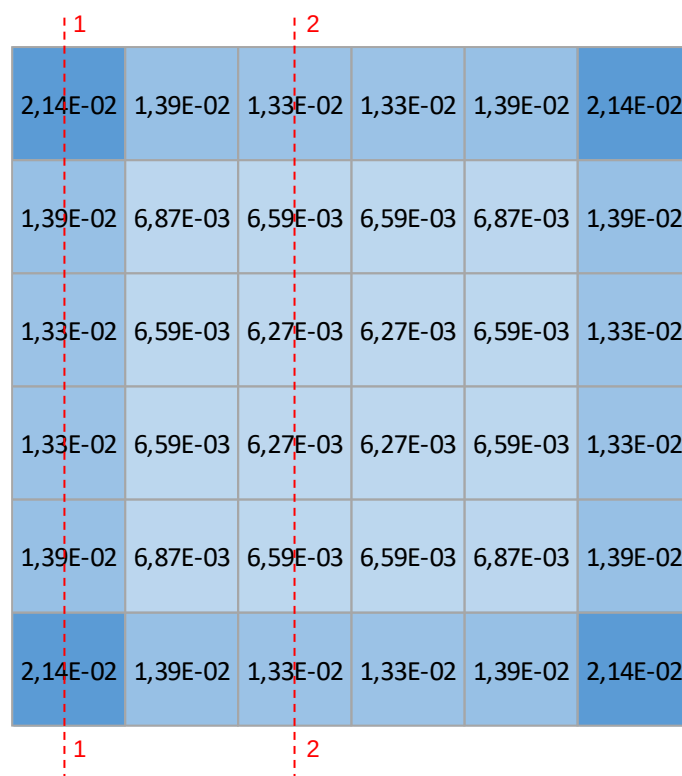


Figura 4.15 – Distribuição dos Coeficientes de Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=0$.

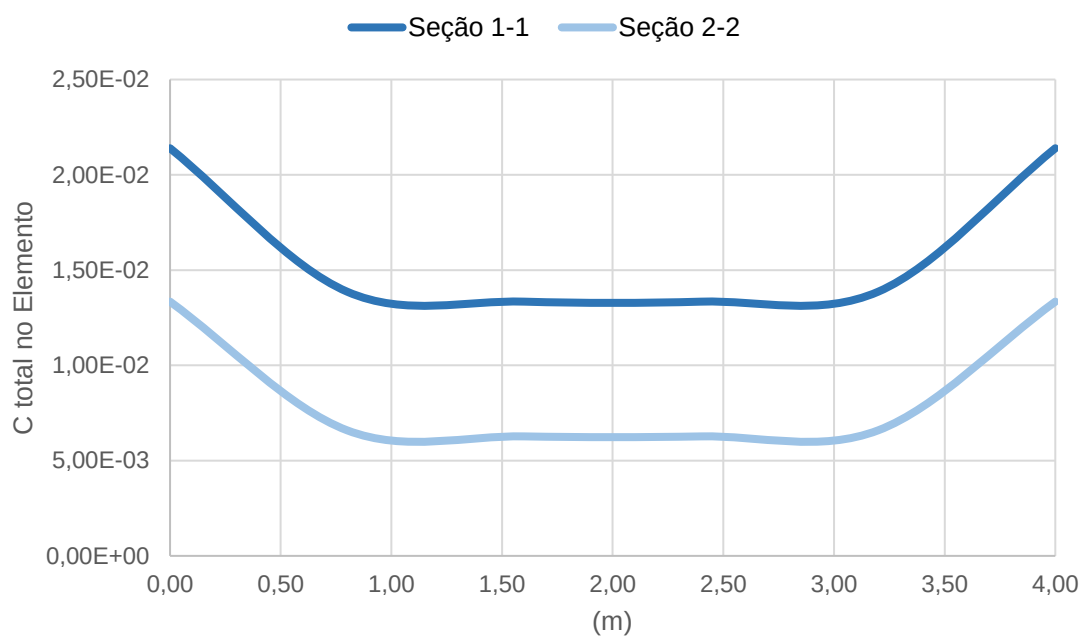


Figura 4.16 – Variação do Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=0$.

1	1,342	0,663	0,636	0,636	0,663	1,342	2
	0,663	-0,135	-0,199	-0,199	-0,135	0,663	
	0,636	-0,199	-0,276	-0,276	-0,199	0,636	
	0,636	-0,199	-0,276	-0,276	-0,199	0,636	
	0,663	-0,135	-0,199	-0,199	-0,135	0,663	
	1,342	0,663	0,636	0,636	0,663	1,342	
1							2

Figura 4.17 – Distribuição dos Coeficientes de Rigidez para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=3,0$.

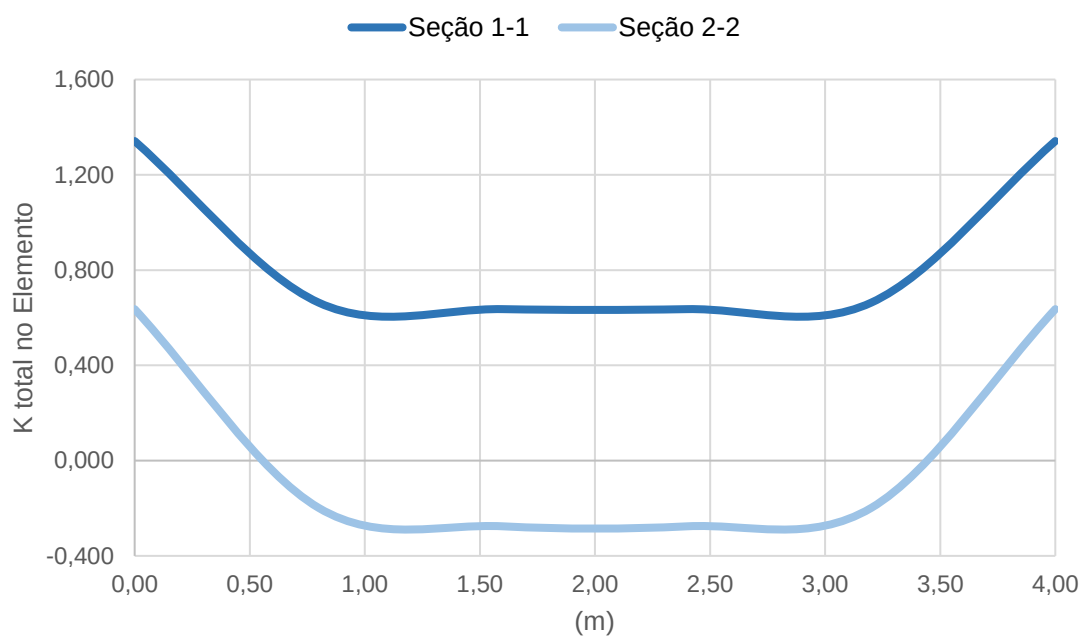


Figura 4.18 – Variação da Rigidez para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=3,0$.

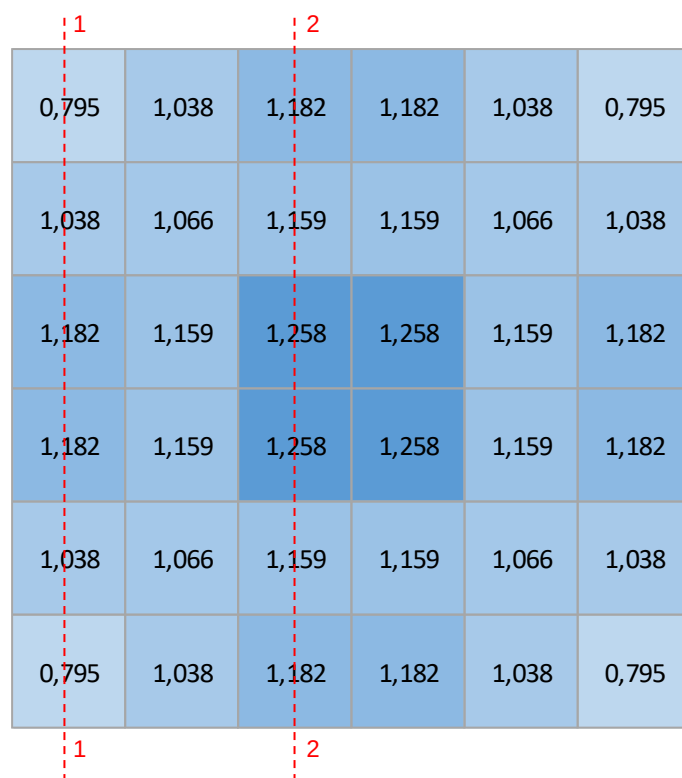


Figura 4.19 – Distribuição dos Coeficientes de Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=3,0$.

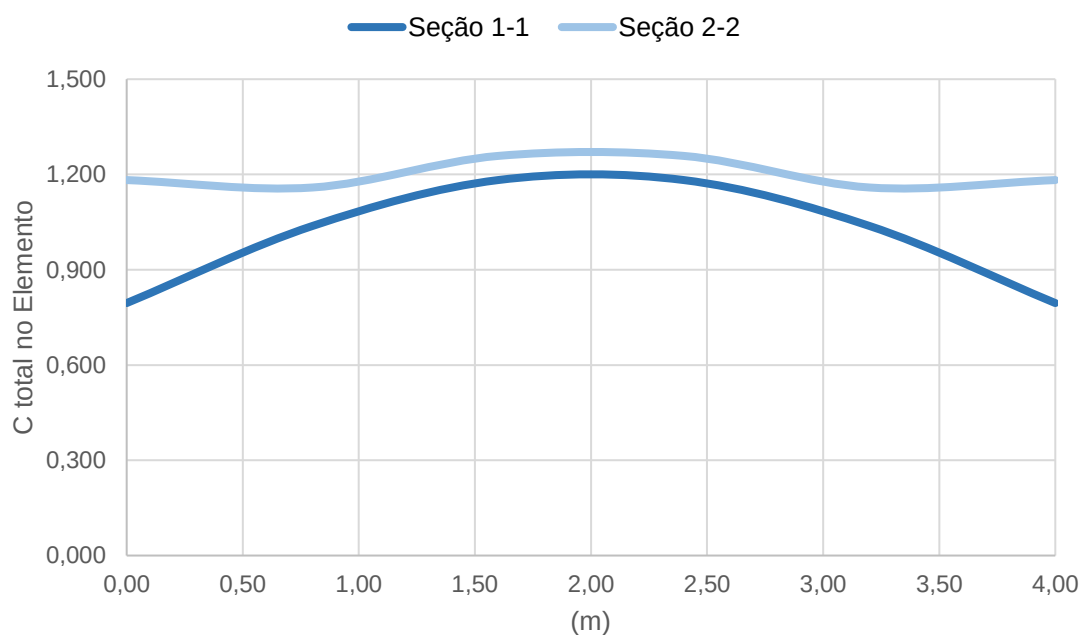


Figura 4.20 – Variação do Amortecimento para Caso 2 – Solo Heterogêneo ($G_2/G_1 = 1,838$) e com $a_0=3,0$.

4.2.3. Caso 3: Base em Rocha

Já para o caso 3 com o semiespaço infinitamente mais rígido do que a camada superior, ou seja para base em rocha com $G_2/G_1 = \infty$, verifica-se da Figura 4.21 a Figura 4.28, que a variação da rigidez e do amortecimento também é mais acentuada do que

para o caso homogêneo, assim como é mais acentuada do que para o caso heterogêneo.

As funções ainda apresentam maiores valores nas bordas da fundação, mas ao aumentar a frequência, os valores da rigidez diminuem, enquanto do amortecimento aumentam, analisando apenas para $a_0=0$ e $a_0=3$. Nota-se que os valores do amortecimento são praticamente nulos no caso estático.

Ao se analisar o caso estático, observa-se que a rigidez apresenta valores superiores em relação aos demais casos analisados, enquanto os valores de amortecimento são muito inferiores, o que confirma a tendência observada no Caso 2.

Para o caso de $a_0 = 3,0$, a fundação se apresenta muito mais rígida nas extremidades, e os valores ao centro ainda encontram inversão de sinal, mas com valores mais próximos de zero e mais concentrados. Para o amortecimento, a variação é mais pronunciada comparada aos demais casos, com menores valores nas bordas, porém com maiores resultados na região central, sem apresentar os mesmos picos nas extremidades.

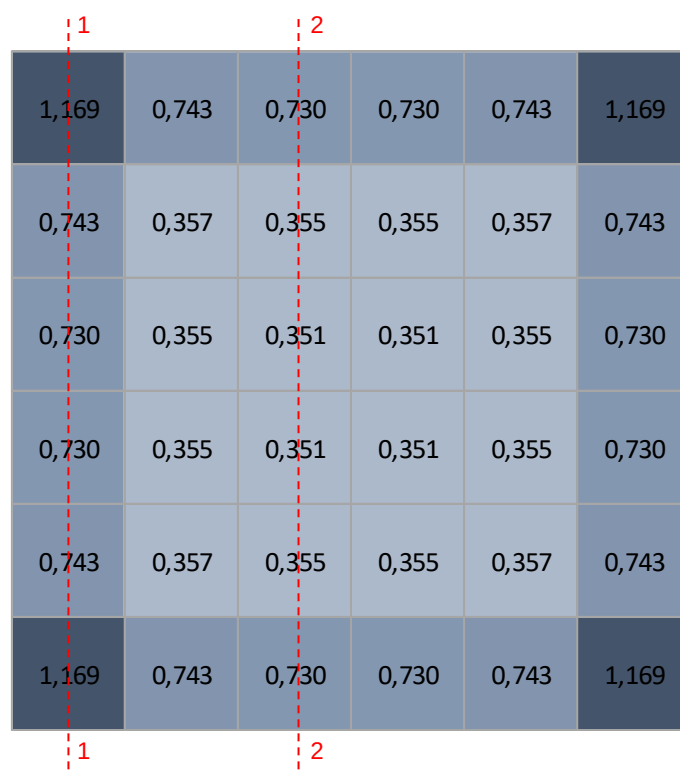


Figura 4.21 – Distribuição dos Coeficientes de Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=0$.

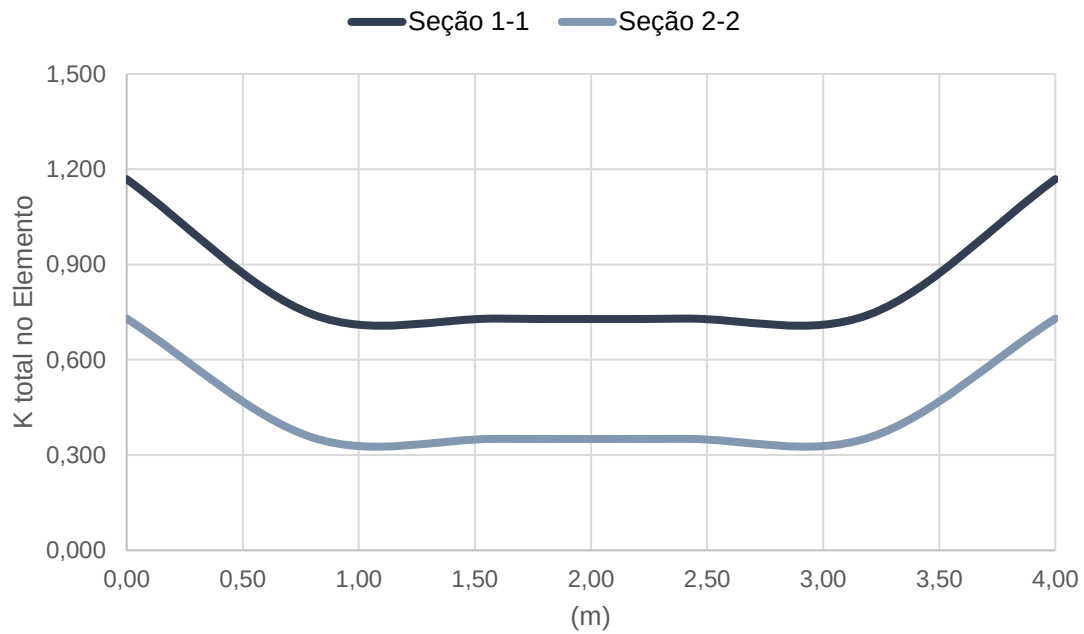


Figura 4.22 – Variação da Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=0$.

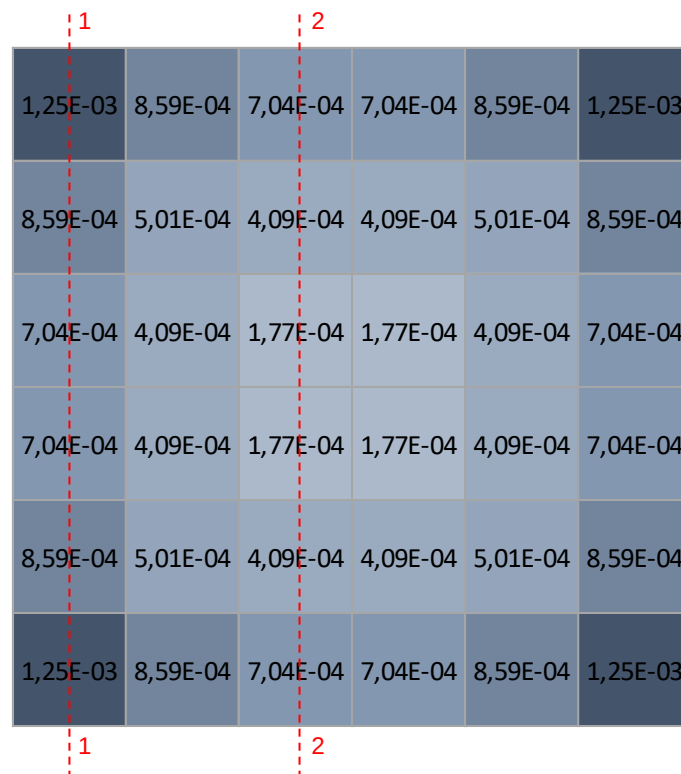


Figura 4.23 – Distribuição dos Coeficientes de Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=0$.

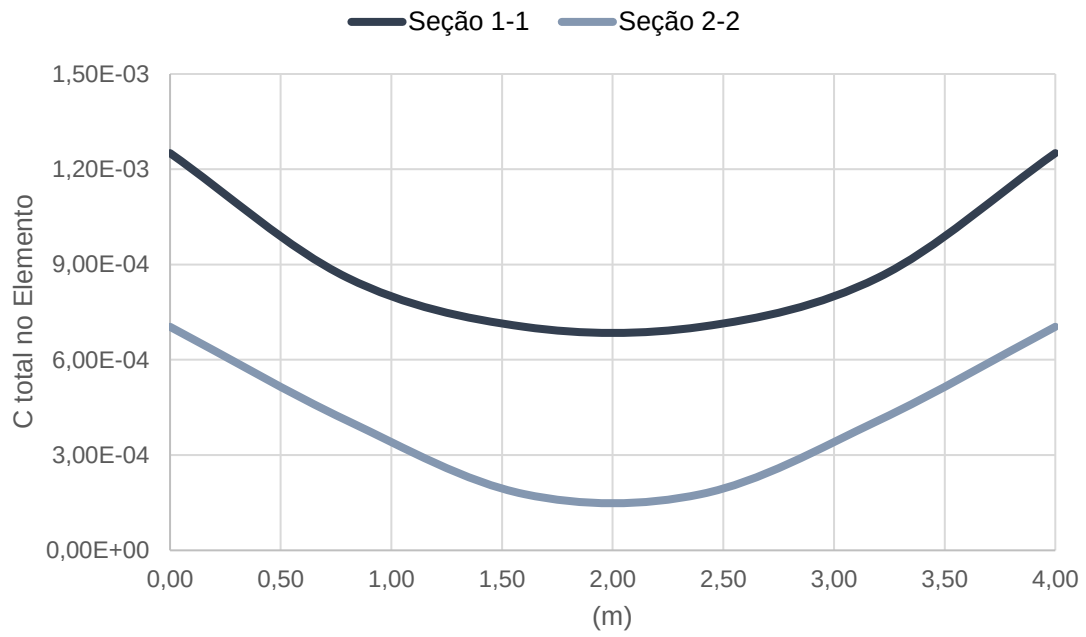


Figura 4.24 – Variação do Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=0$.

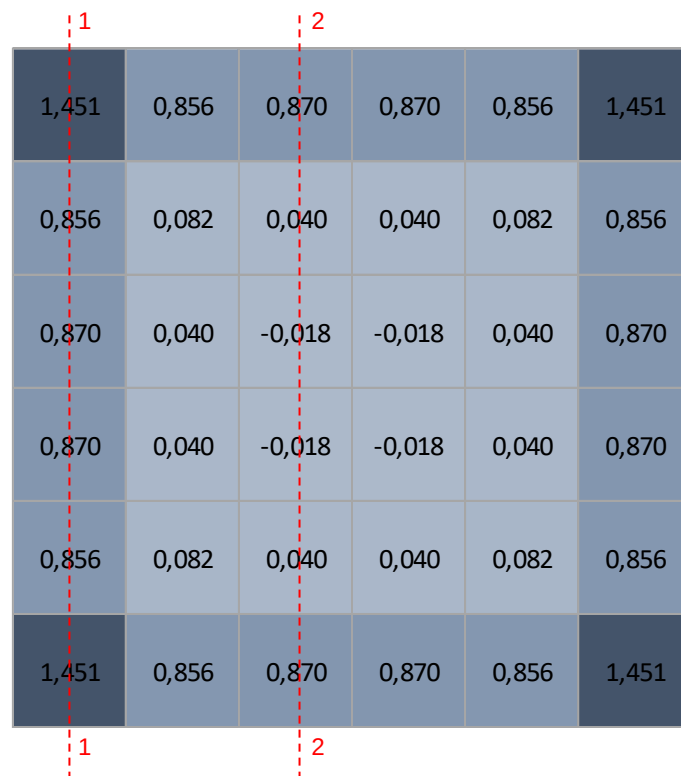


Figura 4.25 – Distribuição dos Coeficientes de Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=3,0$.

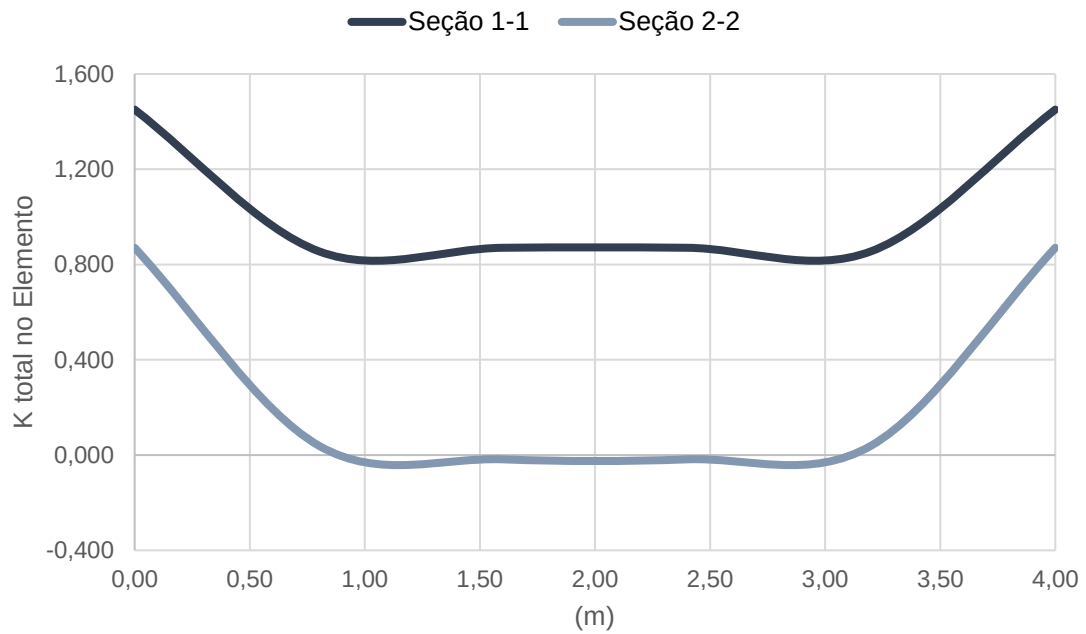


Figura 4.26 – Variação da Rigidez para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=3,0$.

0,644	0,976	1,143	1,143	0,976	0,644
0,976	1,139	1,270	1,270	1,139	0,976
1,143	1,270	1,414	1,414	1,270	1,143
1,143	1,270	1,414	1,414	1,270	1,143
0,976	1,139	1,270	1,270	1,139	0,976
0,644	0,976	1,143	1,143	0,976	0,644

Figura 4.27 – Distribuição dos Coeficientes de Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=3,0$.

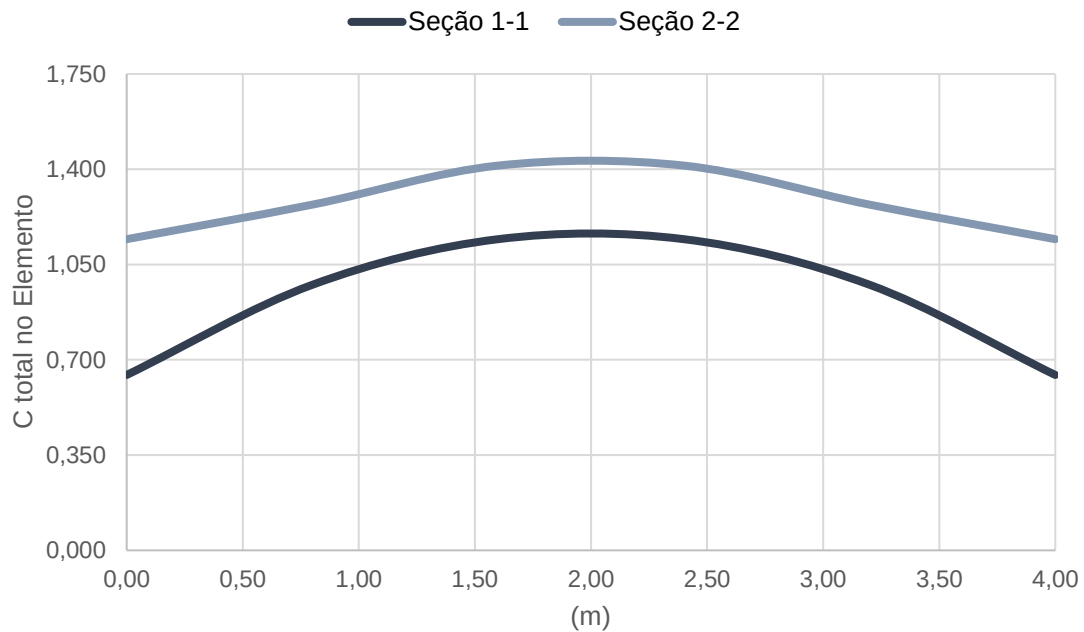


Figura 4.28 – Variação do Amortecimento para Caso 3 – Semiespaço Rígido ($G_2/G_1 = \infty$) e com $a_0=3,0$.

4.3. Variação das funções de impedância para diferentes frequências

A seguir são expressos resultados em termos dos coeficientes adimensionais das expressões (3.47), (3.48) e das frequências adimensionais conforme (3.50). Esses valores são condizentes com os valores obtidos por Luco (1974), conforme comparado da Figura 4.29 a Figura 4.34.

É importante destacar que os valores de referência apresentam uma quantidade significativamente maior de pontos para gerar os gráficos, em comparação com a análise realizada neste estudo. Para os pontos obtidos, os valores se aproximam de forma consistente aos valores de referência, evidenciando a precisão dos dados obtidos.

4.3.1. Caso 1: Solo Homogêneo

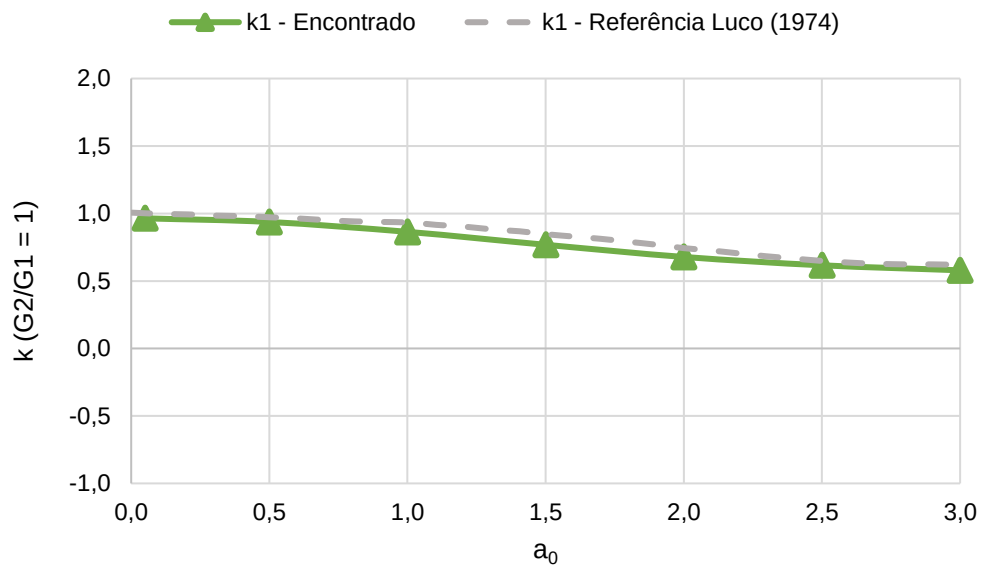


Figura 4.29 – Comparação entre os valores de rigidez adimensionais encontrados e os de referência (LUCO, 1974) – Caso 1: Solo Homogêneo.

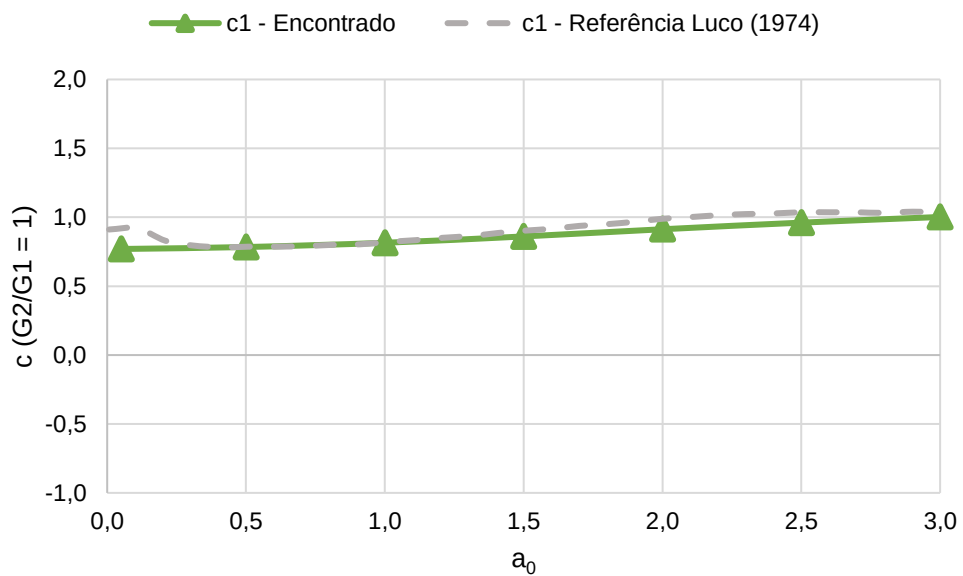


Figura 4.30 – Comparação entre os valores de amortecimento adimensionais encontrados e os de referência (LUCO, 1974) – Caso 1: Solo Homogêneo.

4.3.2. Caso 2: Solo Heterogêneo

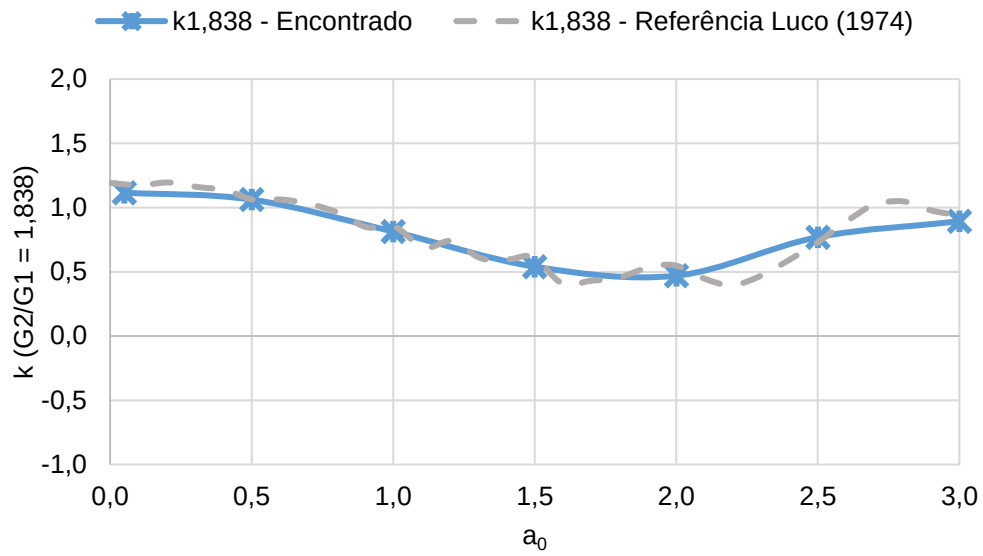


Figura 4.31 – Comparação entre os valores de rigidez adimensionais encontrados e os de referência (LUCO, 1974) – Caso 2: Solo Heterogêneo.

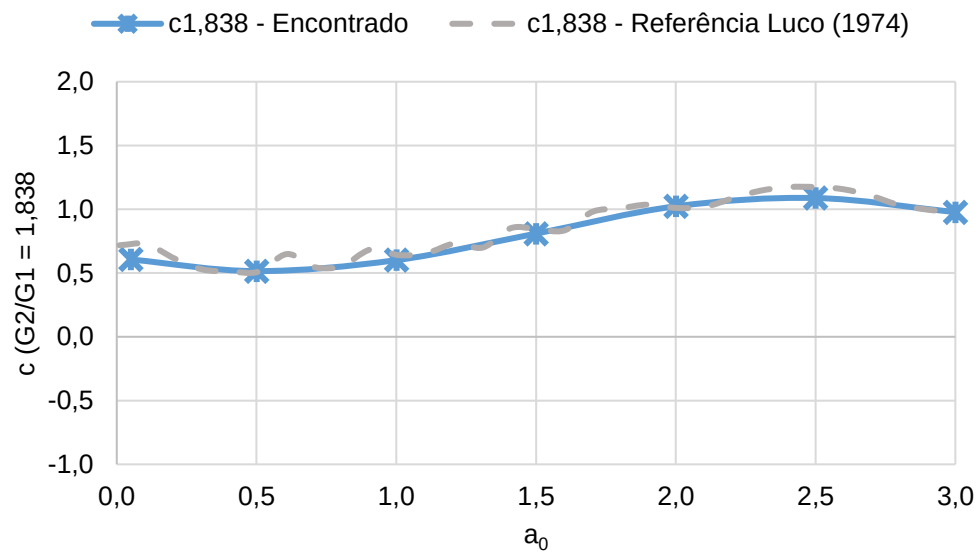


Figura 4.32 – Comparação entre os valores de amortecimento adimensional encontrados e os de referência (LUCO, 1974) – Caso 2: Solo Heterogêneo.

4.3.3. Caso 3: Base em Rocha

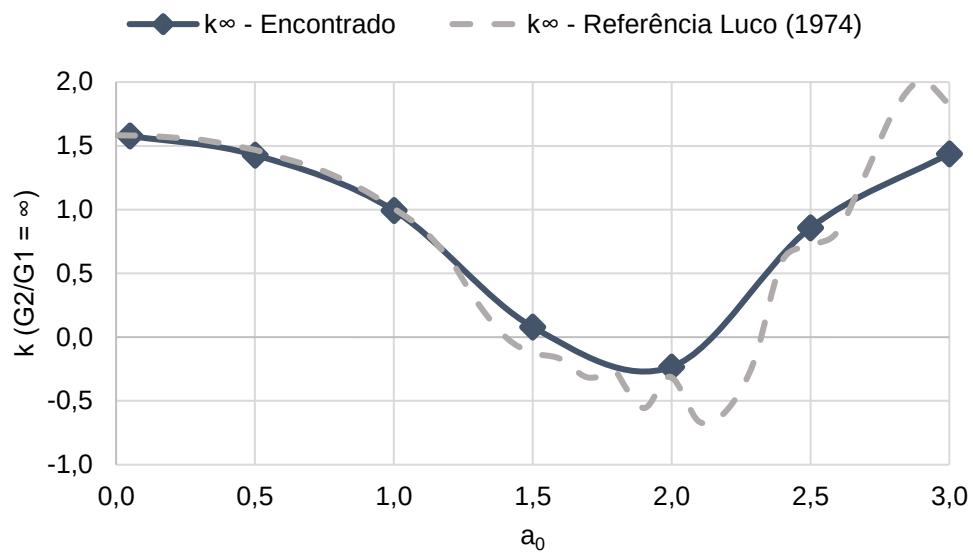


Figura 4.33 – Comparação entre os valores de rigidez adimensionais encontrados e os de referência (LUCO, 1974) – Caso 3: Base em Rocha.

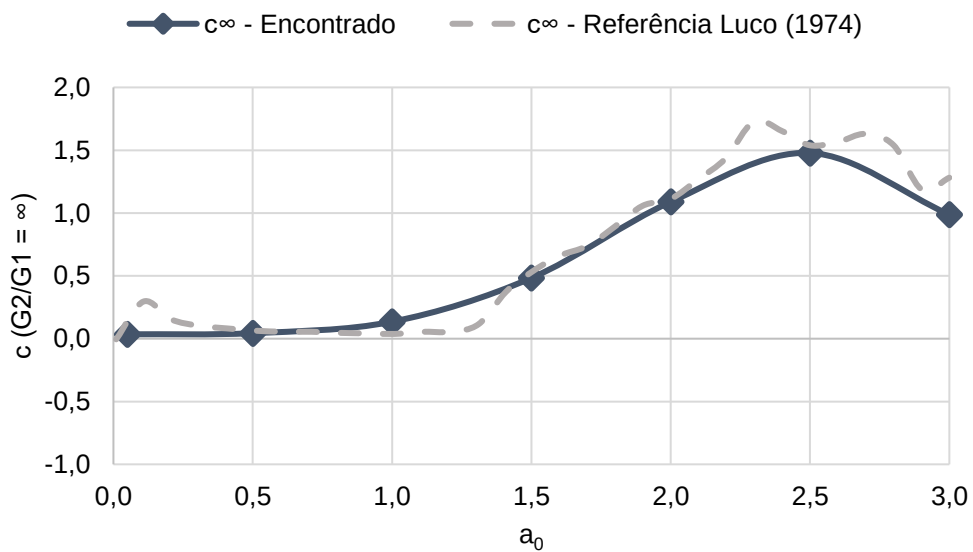


Figura 4.34 – Comparação entre os valores de amortecimento adimensional encontrados e os de referência (LUCO, 1974) – Caso 3: Base em Rocha.

4.3.4. Comparação e Resumo dos Casos

A seguir, na Figura 4.35 e na Figura 4.36 são expressos os mesmos resultados, em termos dos coeficientes e das frequências adimensionais. Essas figuras permitem uma comparação clara entre os diferentes casos analisados.

Para todos os casos é possível observar que os comportamentos das funções de rigidez e amortecimento são distintos para o incremento de frequências.

Entretanto, para o caso homogêneo, a variação é muito pequena e confirma a suposição usual de que, para o solo homogêneo, os coeficientes de rigidez e de amortecimento são praticamente independentes da frequência aplicada.

Para os demais casos é notável que há uma maior variação dos valores das funções de impedância para diferentes frequências, o que indica que a consideração da frequência excitadora não deve ser negligenciada no cálculo dos coeficientes.

As variações no caso 3 são mais extremas e inclusive apresentam valores negativos para a rigidez. Essa instabilidade no resultado é esperada ao trabalhar com valores infinitos.

No caso homogêneo as curvas são bem comportadas, com perda de rigidez e aumento do amortecimento para o incremento de frequências. A rigidez inicia próximo de $k = 1,0$ e $c = 0,80$, confirmando a formulação teórica amplamente utilizada em projetos.

Para os casos 2 e 3, há uma perda de rigidez que apresenta uma recuperação a partir de $a_0 = 2,5$ e um aumento de amortecimento, com queda a partir do $a_0 = 3,0$. Para o Caso 3, as variações são maiores em relação ao Caso 2. Nota-se que no caso do semiespaço rígido, o amortecimento inicia em 0 e segue praticamente nulo até $a_0=1,5$.

Isso permite concluir que para diferentes configurações das camadas e variadas frequências, os comportamentos de rigidez e amortecimento são completamente distintos.

A tabela a seguir resume os resultados da Figura 4.35 e da Figura 4.36.

Tabela 4.1 – Valores calculados numericamente e valores teóricos tradicionais.

Caso	a_0	0,05	0,50	1,00	1,50	2,00	2,50	3,00
	ω	0,02	0,22	0,44	0,66	0,89	1,11	1,33
Caso 1 - Solo Homogêneo ($G_2/G_1 = 1$)	k	0,96	0,94	0,86	0,77	0,68	0,62	0,58
	c	0,77	0,78	0,82	0,86	0,91	0,96	1,00
Caso 2 - Solo Heterogêneo ($G_2/G_1 = 1,838$)	k	1,12	1,06	0,81	0,54	0,47	0,77	0,89
	c	0,61	0,52	0,60	0,81	1,02	1,09	0,98
Caso 3 - Semiespaço Rígido ($G_2/G_1 = \infty$)	k	1,58	1,43	0,99	0,08	-0,23	0,86	1,44
	c	0,04	0,04	0,14	0,48	1,09	1,48	0,99
Teórico	k	14,03	-					
	c	30,41						

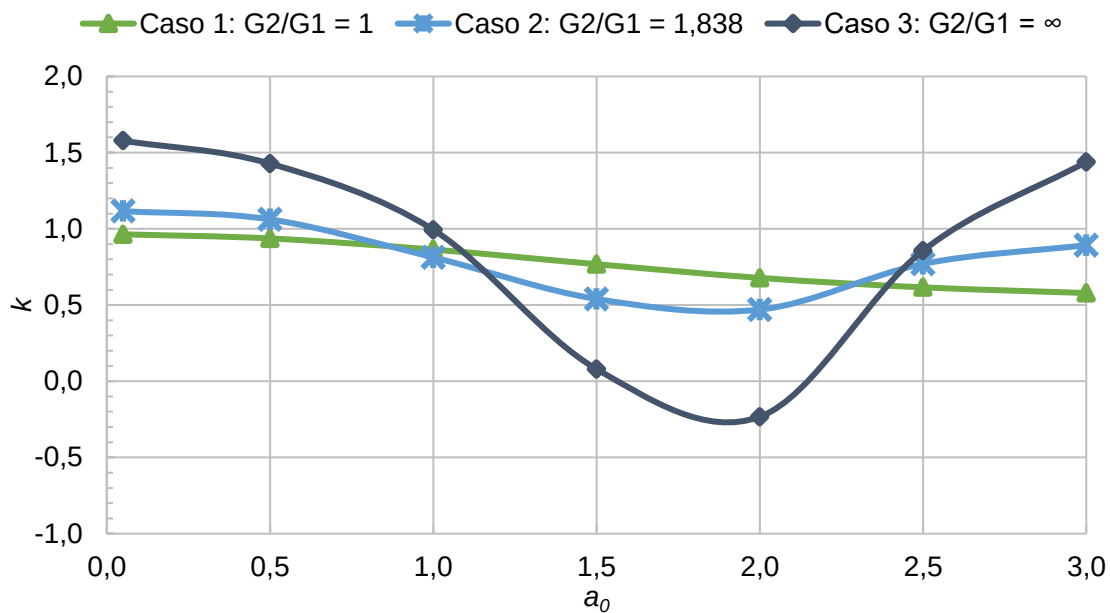


Figura 4.35 – Coeficientes de Rigidez Adimensionais Verticais.

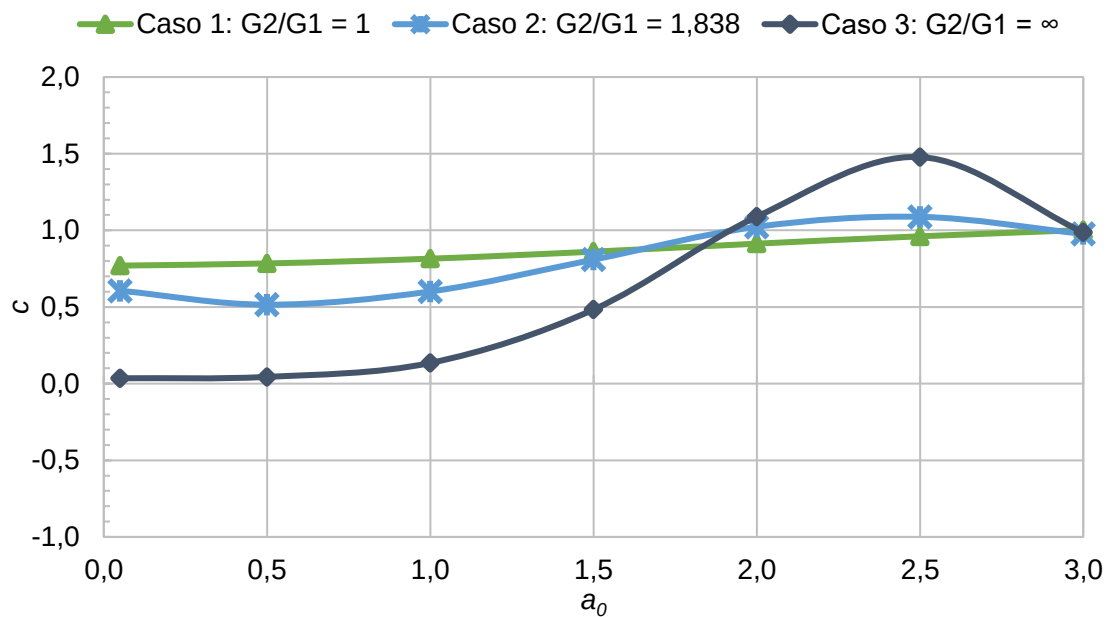


Figura 4.36 – Coeficientes de Amortecimento Adimensionais Verticais.

4.4. Estudo da amplificação dinâmica

Para realização do estudo da amplificação dinâmica, foi considerada a formulação apresentada em (3.52). São avaliados dois tipos de fundação para os três casos de análise: uma fundação “leve” ($\bar{m} = 1,0$) e uma “pesada” ($\bar{m} = 5,0$).

Inicialmente, analisa-se a influência da profundidade da primeira camada do solo. Observa-se que uma menor espessura desta camada aumenta significativamente a amplitude de deslocamento vertical da fundação, assim como a participação do subespaço.

Foram avaliadas espessuras de 2 m, 4 m, 8 m, 12 m, 16 m e 32 m para um solo heterogêneo, onde o módulo de elasticidade transversal do subespaço é 4 vezes maior que o da primeira camada. Nos casos de 2 m e 4 m, há uma amplificação de deslocamento muito maior em comparação com as demais espessuras, enquanto para camadas mais espessas, a diferença é mínima.

À medida que a espessura da camada inicial aumenta, o resultado converge de forma não monotônica para o comportamento de um solo homogêneo, estabilizando-se quando a camada inicial é suficientemente espessa para eliminar a influência do subespaço.

A seguir são expressas as curvas para todas as espessuras de camada analisadas e a curva para o solo homogêneo em linha tracejada.

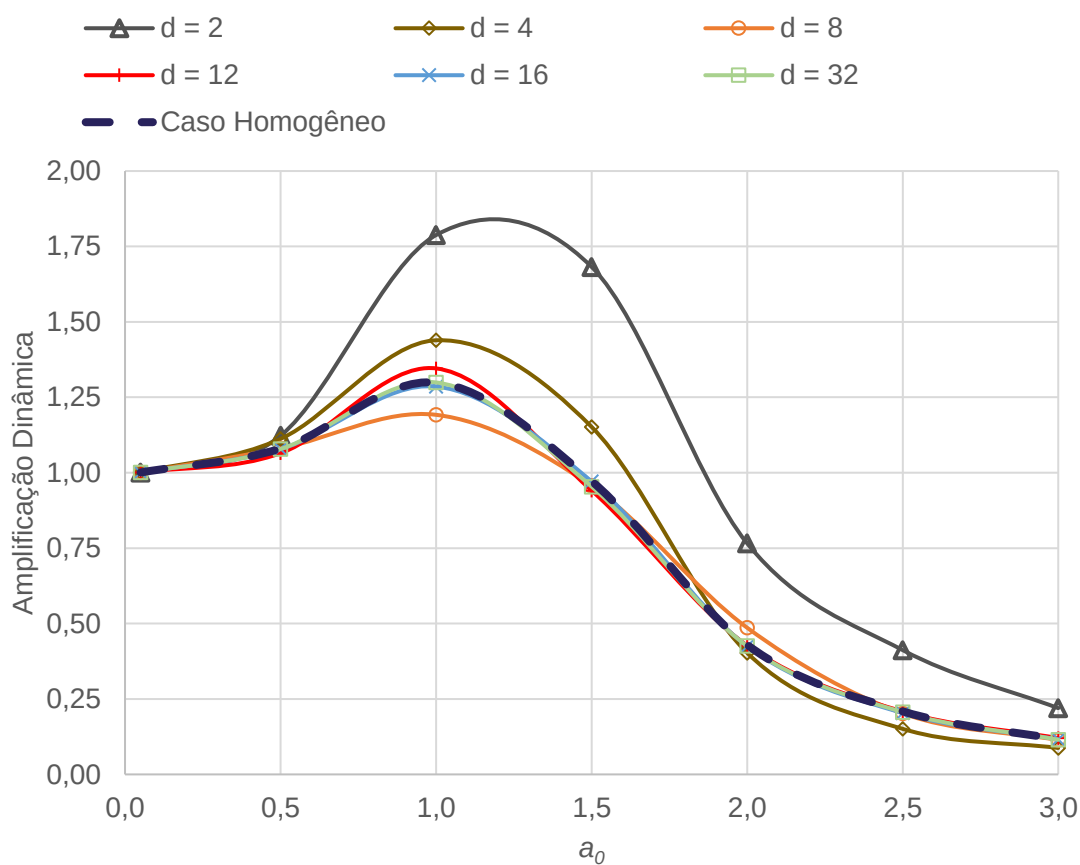


Figura 4.37 – Amplificação dinâmica para fundação leve ($\bar{m} = 1,0$).

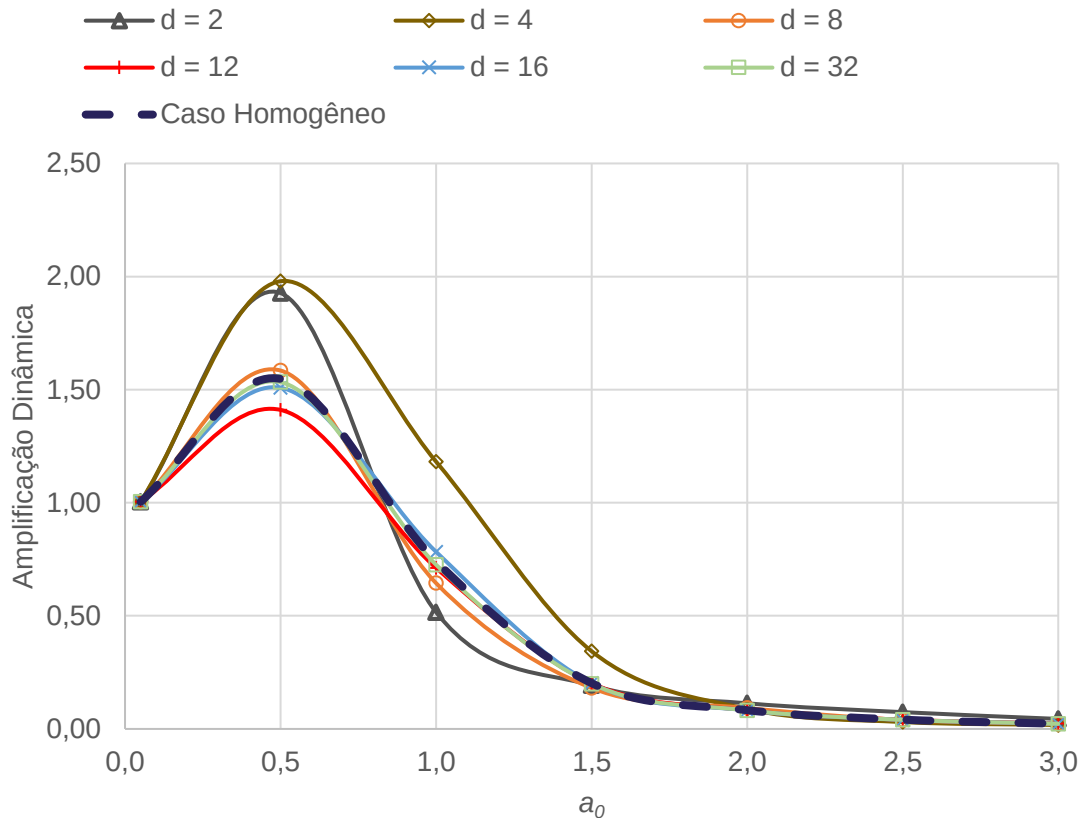


Figura 4.38 – Amplificação dinâmica para fundação pesada ($\bar{m} = 5,0$).

Para ilustrar a convergência, apresentam-se as diferenças entre as curvas de amplificação dinâmica, Δ_{AD} , de cada espessura em relação à curva do solo homogêneo na Figura 4.39 e na Figura 4.40. Observa-se que, a cada aumento na espessura da camada inicial, essas diferenças aproximam-se cada vez mais de zero, indicando a tendência do modelo em convergir para o comportamento de um solo homogêneo.

Observa-se que, para fundações mais leves, as curvas correspondentes às espessuras de 16 m e 32 m são praticamente coincidentes entre si e em relação ao caso homogêneo. Para fundações mais pesadas, a convergência ao comportamento homogêneo é mais precisa na curva de 32 m, especialmente próximo à frequência de amplitude máxima.

Além disso, à medida que se afasta das frequências de amplitude máxima de deslocamento, os resultados para todas as espessuras de camada aproximam-se do comportamento homogêneo.

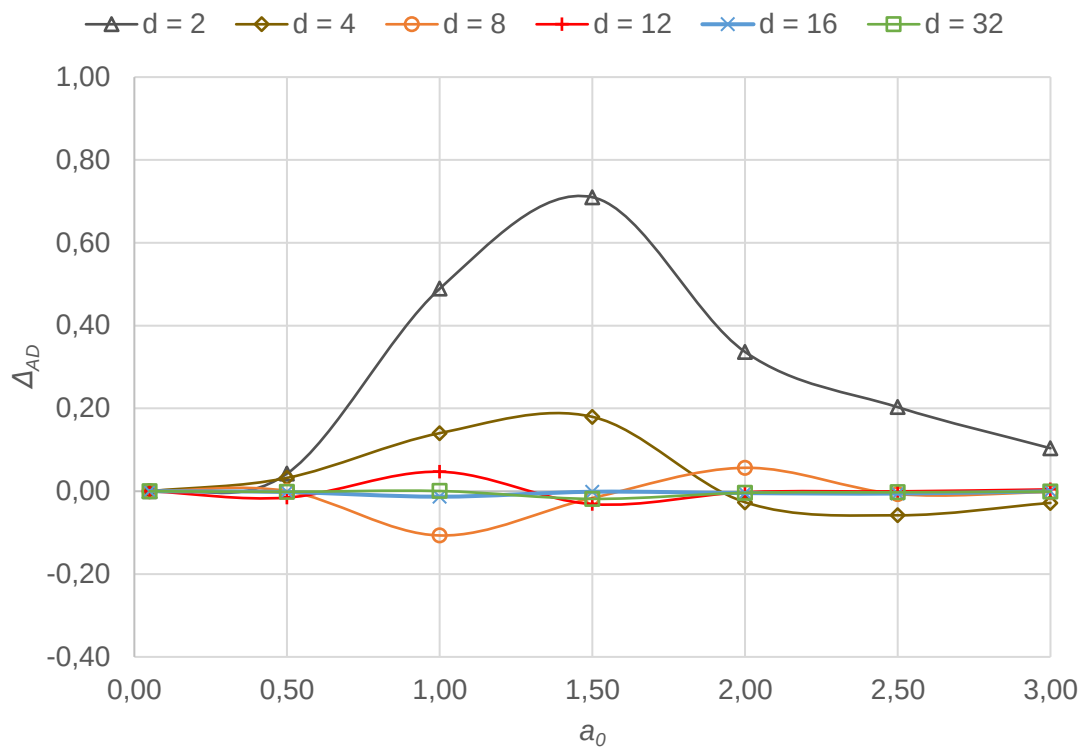


Figura 4.39 – Diferenças absolutas da amplificação dinâmica de cada configuração de espessura das camadas em relação ao caso do solo homogêneo para fundação leve.

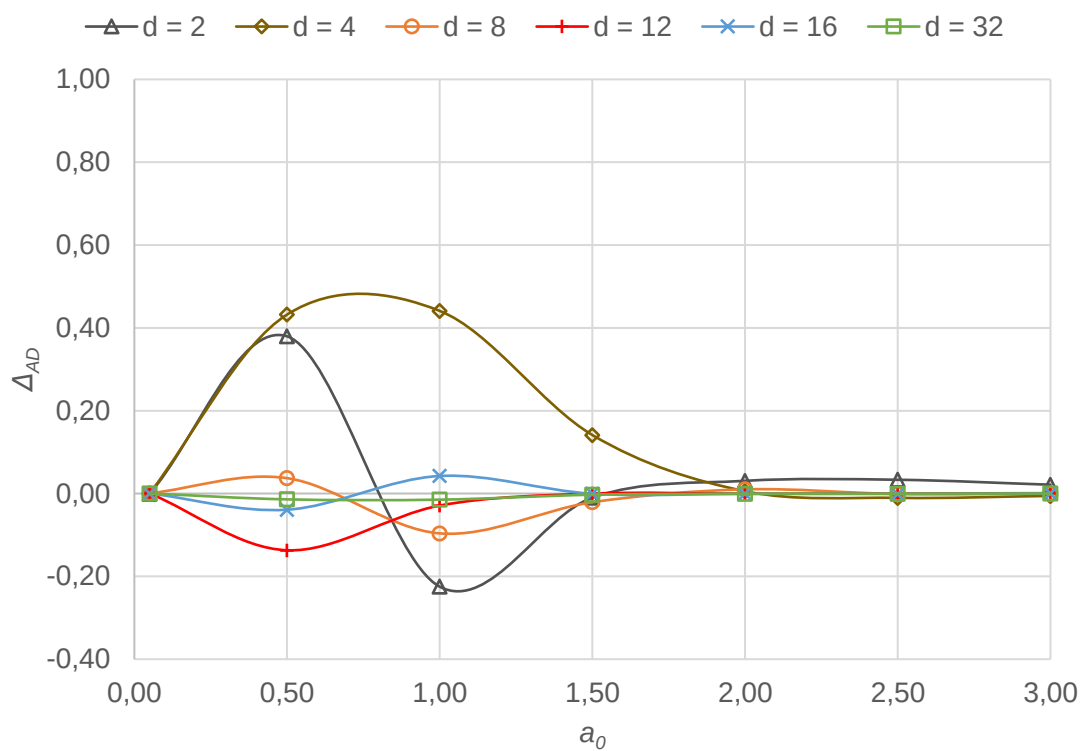


Figura 4.40 – Diferenças absolutas da amplificação dinâmica de cada configuração de espessura das camadas em relação ao caso do solo homogêneo para fundação pesada.

Os resultados são resumidos pela tabela a seguir:

Tabela 4.2 – Valores da amplificação dinâmica para a variação da espessura das camadas do solo.

Espessura da Primeira Camada	a_0	0,05	0,50	1,00	1,50	2,00	2,50	3,00
2 m	$\bar{m} = 1$	1,00	1,12	1,79	1,68	0,77	0,41	0,22
	$\bar{m} = 5$	1,00	1,93	0,52	0,19	0,11	0,07	0,04
4 m	$\bar{m} = 1$	1,00	1,11	1,44	1,15	0,40	0,15	0,09
	$\bar{m} = 5$	1,00	1,98	1,18	0,34	0,09	0,03	0,02
8 m	$\bar{m} = 1$	1,00	1,08	1,19	0,96	0,49	0,20	0,11
	$\bar{m} = 5$	1,00	1,58	0,64	0,18	0,09	0,04	0,02
12 m	$\bar{m} = 1$	1,00	1,06	1,35	0,94	0,43	0,21	0,12
	$\bar{m} = 5$	1,00	1,41	0,71	0,20	0,08	0,04	0,02
16 m	$\bar{m} = 1$	1,00	1,08	1,29	0,97	0,42	0,20	0,12
	$\bar{m} = 5$	1,00	1,51	0,78	0,20	0,08	0,04	0,02
32 m	$\bar{m} = 1$	1,00	1,08	1,30	0,95	0,43	0,21	0,11
	$\bar{m} = 5$	1,00	1,53	0,73	0,20	0,08	0,04	0,02
Caso Homogêneo	$\bar{m} = 1$	1,00	1,08	1,30	0,97	0,43	0,21	0,12
	$\bar{m} = 5$	1,00	1,55	0,74	0,20	0,08	0,04	0,02

Também foi analisado o comportamento da amplificação dinâmica para a variação das propriedades do solo, considerando os mesmos casos apresentados anteriormente: solo homogêneo, solo heterogêneo e subespaço como base rígida. Ao se analisar a influência dos módulos de elasticidade transversal das camadas do solo, é possível observar que para a fundação mais leve há menor influência das camadas do solo na amplificação, apesar de se apresentarem maiores deslocamentos para maior relação entre os módulos de elasticidade transversal das camadas.

Para a fundação mais pesada, há uma maior resposta para os três casos de estudo apresentados ao longo deste capítulo, e o subespaço rígido provoca maiores deslocamentos que o heterogêneo, que por sua vez causa maiores amplitudes que no caso homogêneo.

Em fundações mais leves, os efeitos da interação dinâmica se distribuem de forma mais ampla ao longo da curva de amplificação em função da frequência adimensional, mas se atenuam conforme a frequência aumenta. Já em fundações pesadas, esses efeitos são maiores e localizados próximos à frequência de ressonância, dissipando-se rapidamente à medida que se afastam dessa faixa de frequência. Nota-se que quanto mais pesada a fundação, menor a frequência necessária para atingir o pico da amplificação dinâmica.

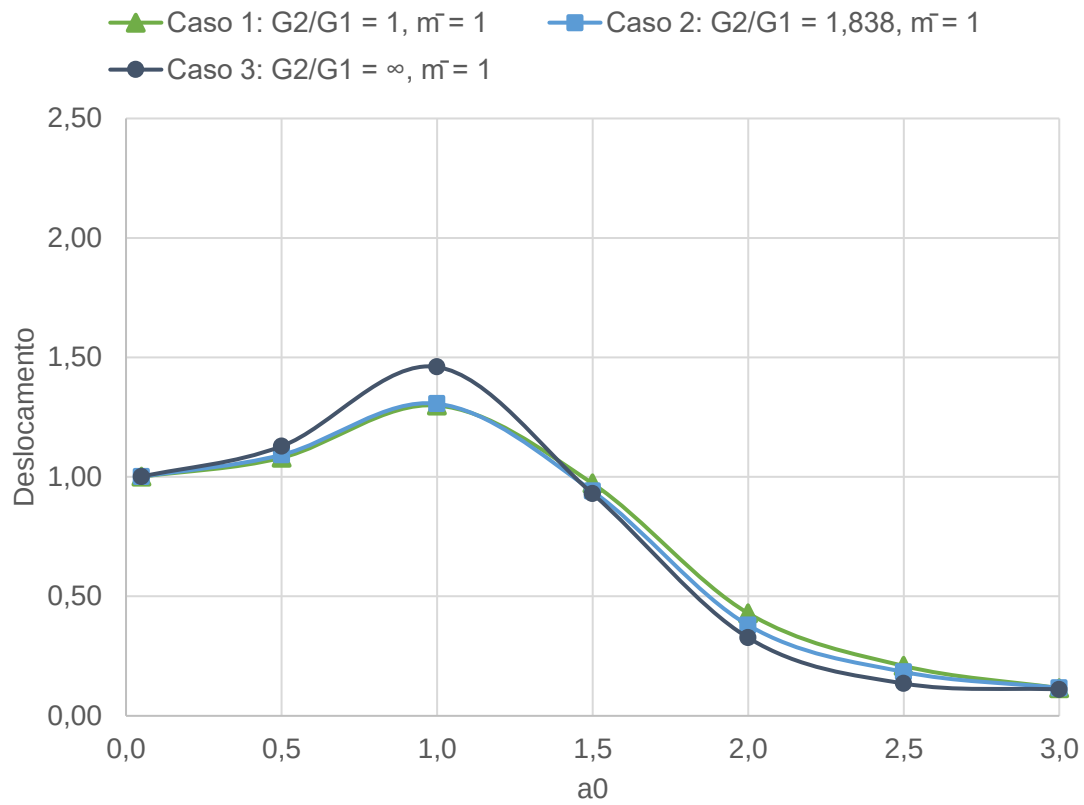


Figura 4.41 – Amplificação dinâmica para fundação leve ($\bar{m} = 1,0$).

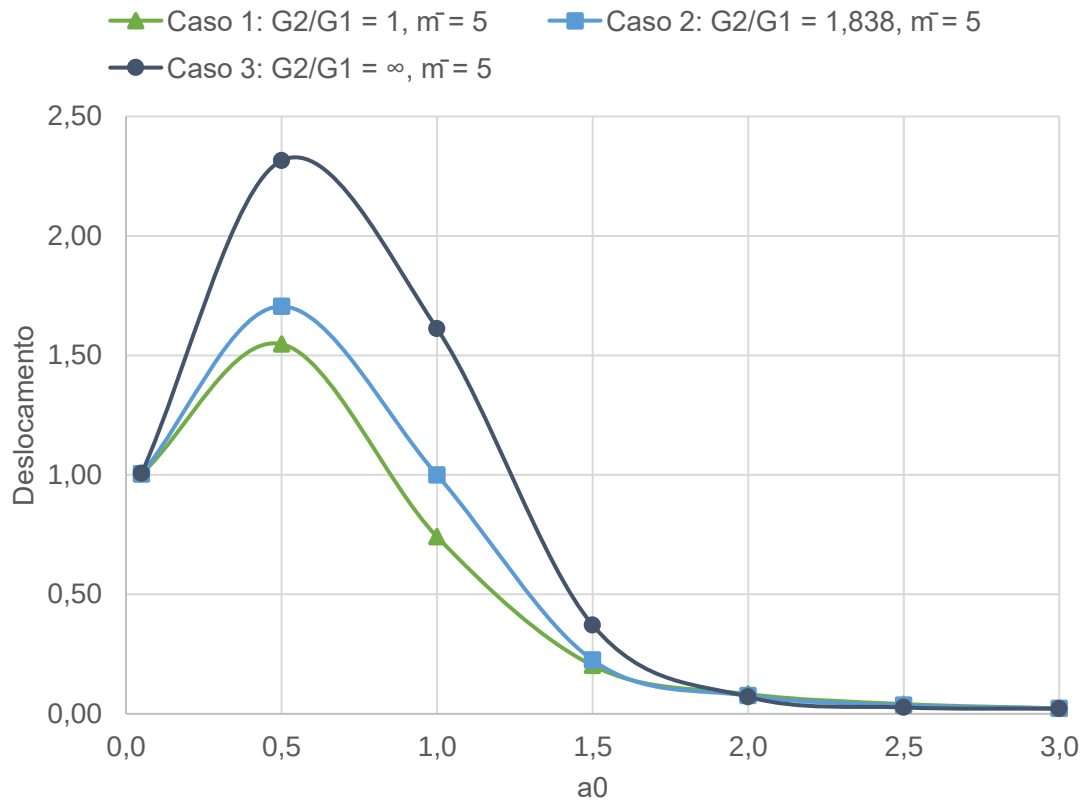


Figura 4.42 – Amplificação dinâmica para fundação pesada ($\bar{m} = 5,0$).

A tabela a seguir resume os resultados da Figura 4.41 e da Figura 4.42.

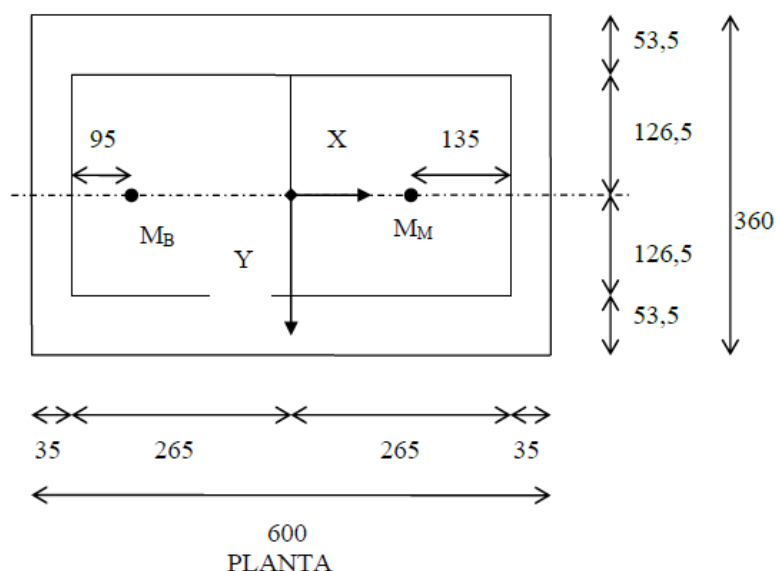
Tabela 4.3 – Valores da amplificação dinâmica para a variação das propriedades do subespaço.

Caso	a_0	0,05	0,50	1,00	1,50	2,00	2,50	3,00
	ω	0,02	0,22	0,44	0,66	0,89	1,11	1,33
Caso 1 - Solo Homogêneo ($G_2/G_1 = 1$)	$\bar{m} = 1$	1,00	1,08	1,30	0,97	0,43	0,21	0,12
	$\bar{m} = 5$	1,00	1,55	0,74	0,20	0,08	0,04	0,02
Caso 2 - Solo Heterogêneo ($G_2/G_1 = 1,838$)	$\bar{m} = 1$	1,00	1,09	1,31	0,94	0,38	0,18	0,12
	$\bar{m} = 5$	1,00	1,71	1,00	0,22	0,08	0,04	0,02
Caso 3 - Semiespaço Rígido ($G_2/G_1 = \infty$)	$\bar{m} = 1$	1,00	1,13	1,46	0,93	0,33	0,14	0,11
	$\bar{m} = 5$	1,01	2,32	1,61	0,37	0,07	0,03	0,02

4.5. Aplicação prática

A seguir é apresentado um exemplo de aplicação prática das análises desenvolvidas e do algoritmo elaborado.

O problema de estudo trata-se de uma fundação direta que recebe as cargas dinâmicas de uma motobomba, cuja geometria pode ser observada na Figura 4.43. Este exemplo é adaptado do item 4.6 do material de SANTOS (2024), em que é utilizado o aplicativo BlockSolver (COUTINHO e MENDES, 2007).



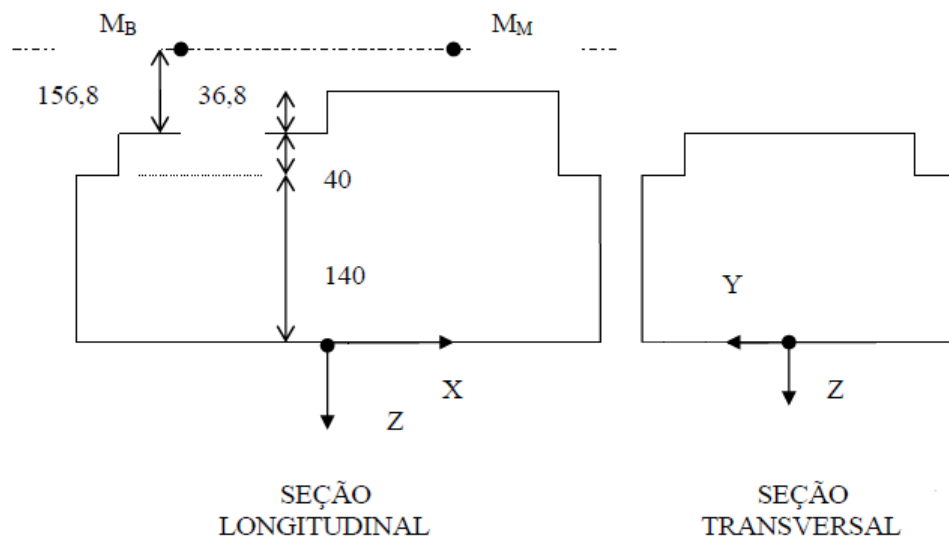


Figura 4.43 – Geometria do bloco de fundação estudado.

São considerados os seguintes parâmetros do solo heterogêneo para análise:

- Módulo de elasticidade transversal da camada superior: $G_1 = 45 \text{ MPa}$;
- Coeficiente de Poisson da camada superior: $\nu_1 = 0,30$;
- Coeficiente de amortecimento histerético da camada superior: $\xi_1 = 0,002$;
- Massa específica da camada superior: $\rho_1 = 1,8 \text{ t/m}^3$;
- Módulo de elasticidade transversal do subespaço: $G_2 = 134 \text{ MPa}$;
- Coeficiente de amortecimento histerético do subespaço: $\xi_2 = 0,002$;
- Coeficiente de Poisson do subespaço: $\nu_2 = 0,30$;
- Massa específica do subespaço: $\rho_2 = 1,8 \text{ t/m}^3$;
- Frequência de operação: $\omega = 710 \text{ rpm} = 11,83 \text{ Hz} = 74,35 \text{ rad/s}$;
- Espessura da primeira camada: $d = 6 \text{ m}$;
- Dimensão do raio da fundação: $R = 2,62 \text{ m}$;
- Massa da bomba: $M_B = 8,27 \text{ t}$;
- Massa do motor: $M_M = 11,70 \text{ t}$;
- Força na bomba: $F_B = 9,39 \text{ kN}$;
- Força no motor: $F_M = 13,29 \text{ kN}$.

Para o caso do solo homogêneo, o BlockSolver é capaz de resolver o problema apresentado de maneira automática. É adotado o valor de G_1 .

BlockSolver

Projeto Ferramentas Propriedades Cargas Análise Ajuda

Matriz de Massa

115.147	0	0	0	0	0
0	115.147	0	0	0	0
0	0	115.147	0	0	0
0	0	0	221.248	0	0
0	0	0	0	390.955	0
0	0	0	0	0	353.888

Matriz Rigidez

565945	0	0	0	756271	0
0	591357	0	-790229	0	0
0	0	711324	0	0	0
0	-790229	0	3.35545E6	0	0
756271	0	0	0	5.87639E6	0
0	0	0	0	0	4.95446E6

Matriz Amortecimento

5479	0	0	0	7322	0
0	5479	0	-7322	0	0
0	0	9504	0	0	0
0	-7322	0	17442	0	0
7322	0	0	0	32099	0
0	0	0	0	0	10928

Cargas

	XX	Y	Z	XY	YY	ZZ
REAL	0	22.68	-6.1474E-19	-46.079	-2.7635E-19	-10.195
IMAGINÁRIA	0	0	22.68	0	-10.195	0

Tipo de Fundação

☒ Fundação Direta ☐ Fundação Estaqueada

OK

Frequência de Operação da Máquina (Hz) 11.830

Frequência Inicial (Hz) 0.000

Incremento de Frequência (Hz) 0.500

Número de Incremento de Frequências (Hz) 80

Centro Geométrico da Base

X (m)	Y (m)	Z (m)
0.000	0.000	0.000

Centro de Massa

X (m)	Y (m)	Z (m)
-0.0084	0.0000	1.3363

ESCOLA POLITÉCNICA - UNIVERSIDADE FEDERAL DO RIO DE JANEIRO - GRUPO DE ANÁLISE DINÂMICA E ENGENHARIA SÍSMICA

Figura 4.44 – Tela principal para solo homogêneo (BlockSolver).

Geometria

Número de Blocos 3

Dimensão da Base

BX (m) 6 BY (m) 3.6

Coordenadas do Centro Geométrico da Base

Xb (m) 0 Yb (m) 0 Zb (m) 0

Blocos

Dimensões

	X (m)	Y (m)	Z (m)
Bloco 1	6.0	3.6	1.4
Bloco 2	5.3	2.53	.40
Bloco 3	2.65	2.53	.368

Coordenadas do Centro de Gravidade

	X (m)	Y (m)	Z (m)
Bloco 1	0	0	0.7
Bloco 2	0	0	1.6
Bloco 3	1.325	0	1.984

A base da fundação deverá ter a sua maior dimensão alinhada com o eixo X sendo a geometria dos restantes blocos definida em concordância com esta orientação.

OK Cancelar

Figura 4.45 – Entrada de dados de geometria (BlockSolver).

Massas

Número de Massas Concentradas

Pesos Volumétricos dos Blocos

	(Ton/m3)
Bloco 1	2.5
Bloco 2	2.5
Bloco 3	2.5

Coordenadas das Massas e Respetivos Valores

	X	Y	Z	(Ton)
Massa 1	1.30	0	3.368	8.27
Massa 2	-1.7	0	3.368	11.70

OK Cancelar

Figura 4.46 – Entrada de dados de massas (BlockSolver).

Solo

Módulo de Deformação Transversal (KN/m2)

Coeficiente de Poisson

Massa Volúmica (ton/m3)

OK Cancelar

Figura 4.47 – Entrada de dados do solo homogêneo (BlockSolver).

Amplitudes

Amplitudes Máximas de Carregamento

	X (KN)	Y (KN)	Z (KN)	XX (KN.m)	YY (KN.m)	ZZ (KN.m)
Massa Concentrada 1	0	9.39	9.39	0	0	0
Massa Concentrada 2	0	13.29	13.29	0	0	0

Ângulos de Fase (graus)

	X	Y	Z	XX	YY	ZZ
Massa Concentrada 1	00	0	90	0	0	0
Massa Concentrada 2	0	0	90	0	0	0

OK Cancelar

Figura 4.48 – Entrada de dados de cargas dinâmicas (BlockSolver).

Frequências

Frequência de Operação da Máquina (Hz) 11.830

Frequência Inicial (Hz) 0.00

Incremento de Frequência (Hz) 0.50

Número de Incremento de Frequências (Hz) 80

OK Cancelar

Figura 4.49 – Entrada de dados da frequência de operação (BlockSolver).

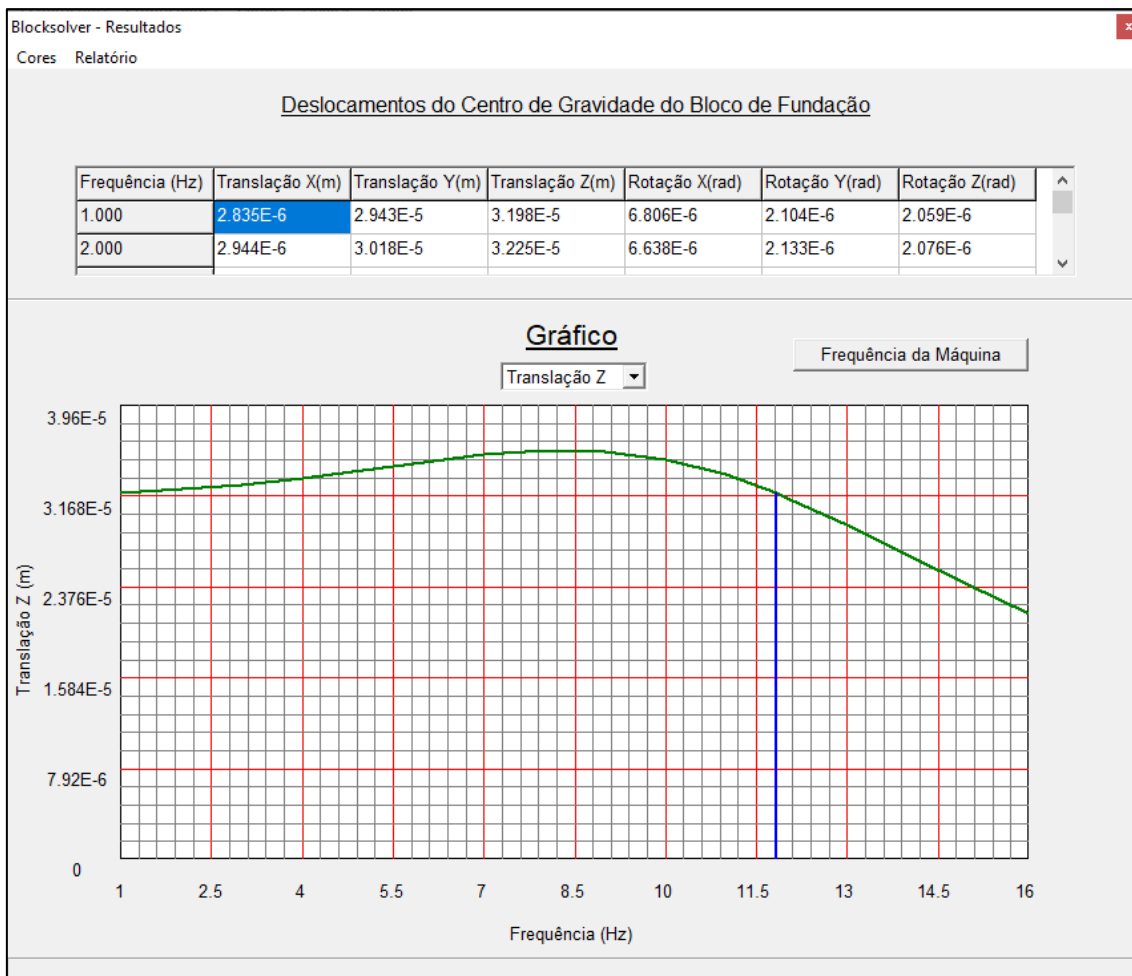


Figura 4.50 – Deslocamentos verticais para o solo homogêneo (BlockSolver).

BlockSolver

Resultados para a Frequência da Máquina

Frequência de Operação:

Translações:

X (m):

Y (m):

Z (m):

Rotações:

X (rad):

Y (rad):

Z (rad):

 Fechar

Figura 4.51 – Resultados para a frequência da máquina no solo homogêneo (BlockSolver).

Portanto, para o solo homogêneo, o BlockSolver retorna um deslocamento vertical de 31,9 μm .

Para realizar os cálculos no BlockSolver para o solo heterogêneo, será necessário realizar uma alteração nas propriedades do solo a partir do cálculo das funções de impedância. Esses valores são obtidos a partir do algoritmo e metodologia desenvolvidos.

Para a frequência de operação, a partir da expressão (3.50) tem-se:

$$a_0 = \frac{\frac{710}{60} \times 2\pi \cdot \sqrt{\frac{6,0 \times 3,6}{\pi}}}{\sqrt{\frac{45000}{1,80}}} = 1,23$$

Pelo programa LUCO, conforme resultado apresentado por SANTOS (2024), para a frequência de 11,83 Hz:

- $K_{LUCO} = 774148 \text{ kN/m};$
- $C_{LUCO} = 4357 \text{ kN/m}.$

Pelo algoritmo desenvolvido, o resultado se aproxima do valor esperado:

- $K = 739260 \text{ kN/m};$
- $C = 338549/(2\pi \times 11,83) = 4555 \text{ kN/m}.$

Esses dados são para calcular valores equivalentes de G_0 e ρ_0 , utilizados como nova entrada no programa. Para isso, são utilizadas as expressões (2.92) e (2.107), respectivamente para a rigidez e para o amortecimento, para fundações circulares.

$$K_z = \frac{4Gr}{1-\nu} \rightarrow G_0 = \frac{K \cdot (1-\nu)}{4r} = \frac{739260 \cdot (1-0,30)}{4 \times 2,62} = 49378 \text{ kN/m}^3$$

$$C_z = \frac{3,4r^2}{1-\nu} \sqrt{\rho G} \rightarrow \rho_0 = \left(\frac{C \cdot (1-\nu)}{3,4r^2} \right)^2 \cdot \frac{1}{G_0} = 0,378 \text{ t/m}^3$$

A seguir são apresentados os resultados com a utilização dos valores representados acima.

BlockSolver

Projeto Ferramentas Propriedades Cargas Análise Ajuda

Matriz de Massa

115.147	0	0	0	0	0
0	115.147	0	0	0	0
0	0	115.147	0	0	0
0	0	0	221.248	0	0
0	0	0	0	390.955	0
0	0	0	0	0	353.888

Matriz Rigidez

621031	0	0	0	829882	0
0	648916	0	-867145	0	0
0	0	780559	0	0	0
0	-867145	0	3.68205E6	0	0
829882	0	0	0	6.44902E6	0
0	0	0	0	0	5.44767E6

Matriz Amortecimento

2630	0	0	0	3514	0
0	2630	0	-3514	0	0
0	0	4563	0	0	0
0	-3514	0	6761	0	0
3514	0	0	0	11253	0
0	0	0	0	0	2930

Cargas

	X	Y	Z	XX	YY	ZZ
REAL	0	22.68	-6.1474E-19	46.079	-2.7635E-19	-10.195
IMAGINÁRIA	0	0	22.68	0	-10.195	0

Tipo de Fundação

☒ Fundação Direta ☐ Fundação Entalheada

OK

Frequência de Operação da Máquina (Hz) 11.830

Frequência Inicial (Hz) 0.000

Incremento de Frequência (Hz) 0.500

Número de Incremento de Frequências (Hz) 80

Centro Geométrico da Base

X (m)	Y (m)	Z (m)
0.000	0.000	0.000

Centro de Massa

X (m)	Y (m)	Z (m)
-0.0084	0.0000	1.3363

ESCOLA POLITÉCNICA - UNIVERSIDADE FEDERAL DO RIO DE JANEIRO - GRUPO DE ANÁLISE DINÂMICA E ENGENHARIA SÍSMICA

Figura 4.52 – Tela principal para solo heterogêneo (BlockSolver).

Solo

Módulo de Deformação Transversal (KN/m²) 49378

Coefficiente de Poisson 0.30

Massa Volúmica (ton/m³) 0.378

OK Cancelar

Figura 4.53 – Entrada de dados equivalentes para solo heterogêneo (BlockSolver).

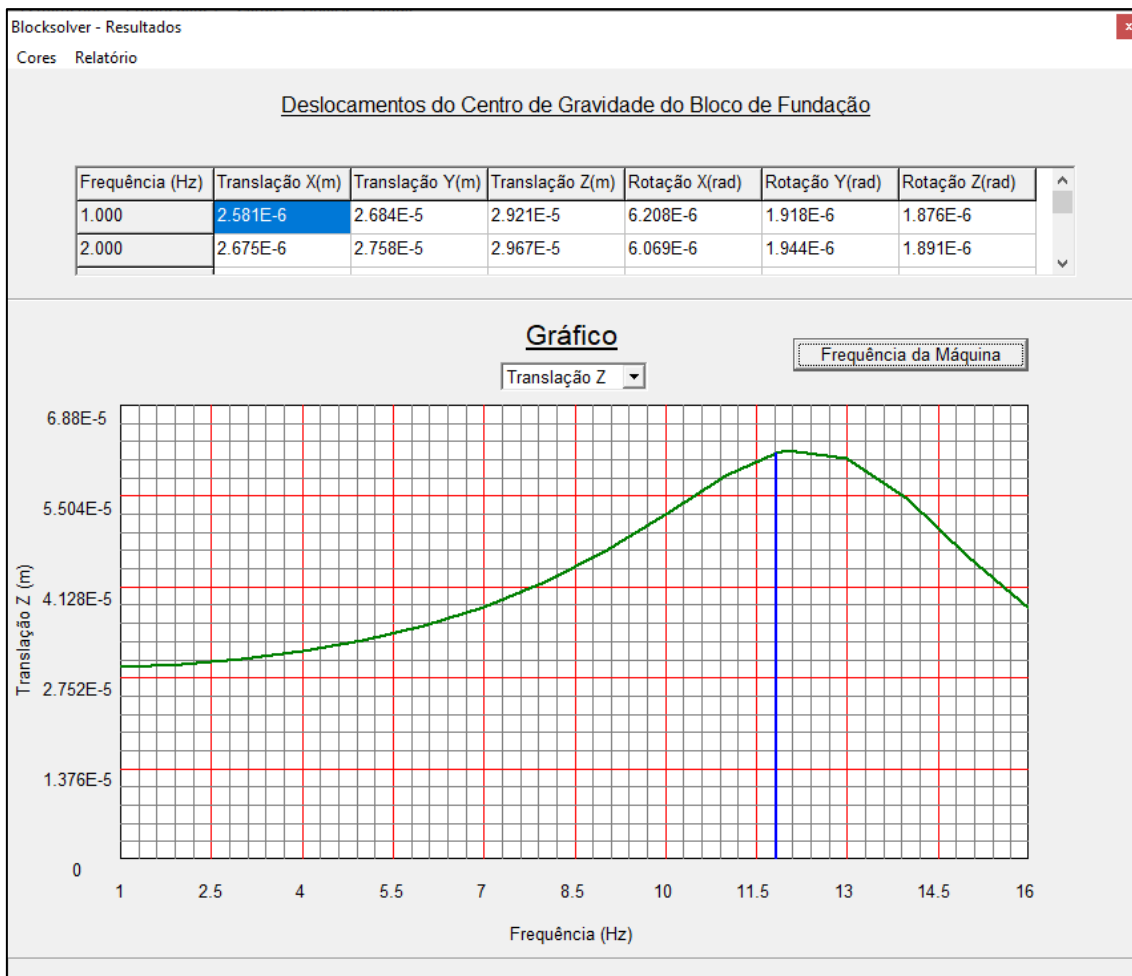


Figura 4.54 – Deslocamentos verticais para o solo heterogêneo (BlockSolver).

BlockSolver

Resultados para a Frequência da Máquina

Frequência de Operação: 11.830

Translações:

X (m): 9.485E-6

Y (m): 2.108E-5

Z (m): 6.154E-5

Rotações:

X (rad): 2.572E-5

Y (rad): 2.138E-6

Z (rad): 2.914E-6

Fechar

Figura 4.55 – Resultados para a frequência da máquina no solo heterogêneo (BlockSolver).

Portanto, para o solo heterogêneo, o BlockSolver retorna um deslocamento vertical de $61,5 \mu m$, o que representa uma faixa aceitável, enquanto para o solo homogêneo ($31,9 \mu m$) está em uma faixa ótima, de acordo com os limites definidos pela ISO 2372.

A seguir são representados os deslocamentos verticais para os casos homogêneo e heterogêneo.

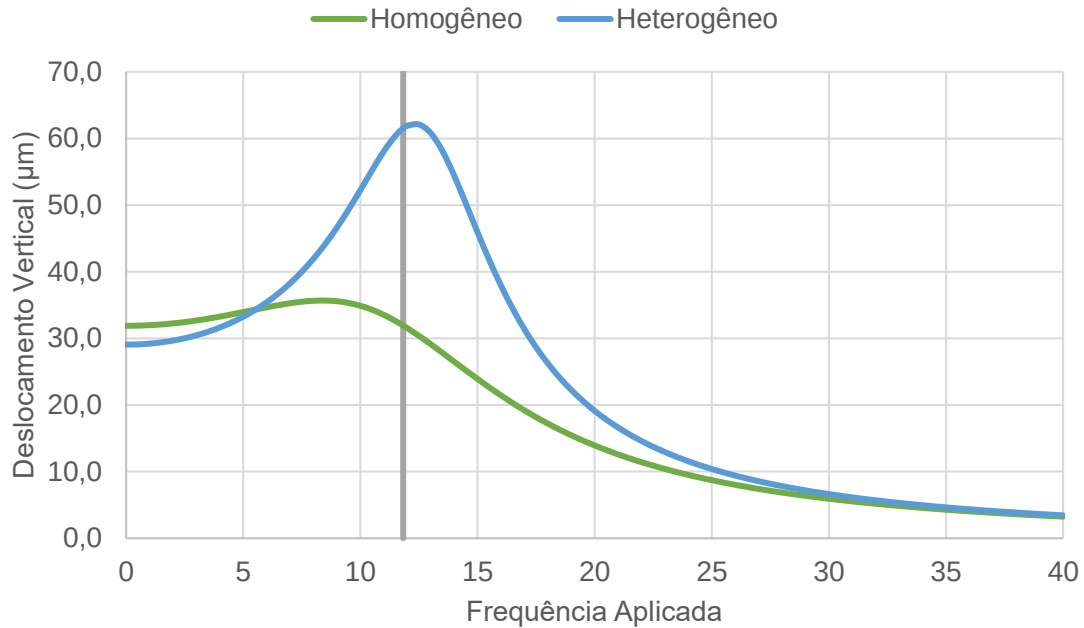


Figura 4.56 – Variação do deslocamento vertical com a frequência do carregamento.

O resultado pode ser comparado a partir os resultados da amplificação dinâmica, calculada conforme (3.52), em que $w_{st} = F_{total}/K$.

Por (3.53), massa adimensional do sistema $\bar{m}$ é:

$$\bar{m} = \frac{m}{\rho \cdot a \cdot b} = \frac{8,27 + 11,70}{1,80 \times \frac{6,0}{2} \times \frac{3,6}{2}} = 2,05$$

Logo, para o solo heterogêneo à amplificação dinâmica resulta em $w/w_{st} = 2,02$ e o deslocamento estático é:

$$w_{st} = \frac{9,39 + 13,29}{739260} = 30,6 \mu m$$

Isso resulta em um deslocamento vertical igual a:

$$w = 2,02 \times 30,6 \mu m = 61,8 \mu m$$

Ou seja, isso representa uma diferença de 0,5% em relação ao cálculo com o BlockSolver.

Capítulo 5

5. CONCLUSÕES

Os resultados obtidos permitem concluir que a definição dos coeficientes de rigidez e de amortecimento do solo devem buscar uma distribuição que reproduza, de forma mais realista possível, as tensões na interface entre o solo e a fundação.

Neste trabalho, foi apresentado como considerar o comportamento elástico de um semiespaço qualquer, por meio de uma matriz de rigidez e de amortecimento, determinadas de forma precisa para um apoio de um bloco sobre uma superfície, com a determinação prévia da matriz de flexibilidade por meio de um modelo de elementos de interface. Depois foi aplicado o método de rigidez invertendo a matriz de flexibilidade.

Para esse cálculo é necessário utilizar um programa computacional, pois há singularidades a serem eliminadas, decorrentes das expressões dos campos de deslocamentos provocados por forças concentradas em diversas direções e em diferentes pontos na superfície e no interior do semiespaço contínuo elástico-linear. Tais singularidades se manifestam principalmente em problemas mais complexos.

Em todos os casos observou-se que as funções de impedância apresentam maiores valores nas bordas da fundação, para o caso estático. Entretanto, a rigidez e o amortecimento exibem comportamentos distintos para diferentes frequências e para configurações de camadas diferentes.

No primeiro cenário, que considera um solo homogêneo, à medida que a frequência excitadora aumenta, tanto os valores de rigidez quanto os de amortecimento próximo às bordas aumentam em módulo. No entanto, a rigidez reduz de valor nas regiões centrais e inverte de sinal. Observa-se que, apesar de aumentar o valor dos coeficientes de amortecimento nas extremidades, essa função se concentra nas regiões centrais e apresenta picos nas bordas.

No segundo caso, que considera um solo heterogêneo, verifica-se uma variação mais acentuada da rigidez e do amortecimento em comparação com o solo homogêneo. As funções ainda apresentam maiores valores nas bordas da fundação. Com o aumento da frequência excitadora, os valores de amortecimento e rigidez próximo das bordas aumentam, mas a rigidez ao centro reduz de valor e inverte de sinal. O efeito de enrijecimento nas extremidades é aplicável ao amortecimento, porém não de maneira

uniforme em todas as regiões, com maiores valores ao centro, assim como no caso homogêneo.

No terceiro caso, que considera um semiespaço rígido, assim como no segundo cenário, as variações da rigidez e do amortecimento são mais intensas em comparação à do solo homogêneo e as funções ainda apresentam maiores valores nas bordas da fundação.

Com o aumento da frequência excitadora, os valores da rigidez diminuem ao centro e aumentam nas bordas, enquanto os do amortecimento são maiores ao centro e reduzem nas bordas. Nota-se que os valores do amortecimento são nulos no caso estático.

Os coeficientes adimensionais das funções de impedância e as frequências adimensionais apresentados no estudo indicam que, para o solo homogêneo, os coeficientes de rigidez e amortecimento são praticamente independentes da frequência aplicada, confirmando a suposição usual de projeto.

No entanto, para os casos heterogêneo e de semiespaço rígido, há uma variação significativa dos valores das funções de impedância para diferentes frequências, destacando a importância de se considerar a frequência excitadora no cálculo dos coeficientes.

Em particular, as variações no caso do semiespaço rígido são mais extremas, apresentando inclusive valores negativos para a rigidez total da fundação, para determinadas frequências, o que pode ser justificado ao lidar com valores infinitos.

O amortecimento calculado reflete a energia da onda que se propaga em resposta ao carregamento variável aplicado à superfície. É importante considerar que esse fenômeno está intimamente ligado ao confinamento do material, uma vez que um maior nível de amortecimento é observado quando o solo está mais confinado. Nesse contexto, o amortecimento depende do confinamento da coluna de solo que se estende até o semiespaço.

A concentração do coeficiente de amortecimento no centro da fundação é uma abordagem lógica, pois é nesse ponto que o solo apresenta um melhor confinamento. Nas bordas da fundação, a propagação da energia é menos eficaz, uma vez que a energia transmitida pelas bordas é menos absorvida pelo solo devido à ausência de vizinhança que poderia contribuir para o confinamento. Assim, a energia tende a se dissipar menos nessas regiões.

Por outro lado, ao se considerar o interior da fundação, onde se busca dispersar a energia, é essencial reconhecer que um entorno significativo também está interagindo com a carga. Portanto, é plausível que os resultados obtidos para o amortecimento nesse contexto sejam coerentes, refletindo uma disposição adequada da função.

Além disso, é importante destacar que a utilização de modelos numéricos, em algumas situações, pode apresentar resultados pouco conservadores. Evidencia-se a importância de um estudo cuidadoso de cada situação a ser modelada, dada a complexidade do problema dinâmico.

Em condições estáticas, o sistema exibe um comportamento relativamente simples e previsível, com a maior parte da rigidez concentrada nas bordas da interface. Sob condições dinâmicas, é importante considerar as regiões de rigidez negativa e o aumento no amortecimento, já que isso pode afetar a estabilidade da fundação e a dissipação de energia nas frequências de operação da máquina.

Essas observações são essenciais para o projeto de fundações de máquinas, já que as variações dinâmicas podem exigir ajustes na estrutura para evitar efeitos indesejáveis, como ressonância ou perda de estabilidade.

Para o estudo da amplificação dinâmica, foram avaliados dois tipos de fundação: “leve” e “pesada”. No cenário considerado, foi adotado um módulo de cisalhamento para o semiespaço quatro vezes maior do que aquele da primeira camada. A análise mostrou que, quanto menor a espessura da primeira camada do solo, maior é a amplitude de deslocamento vertical da fundação e a influência do subespaço.

Para camadas de 2 m e 4 m, observa-se uma amplificação significativa, enquanto para espessuras maiores essa influência é mínima. À medida que a espessura aumenta, o comportamento converge de forma não monotônica para o de um solo homogêneo, com as diferenças entre as curvas se aproximando de zero. A diferença entre as amplitudes é maior para frequências próximas das amplitudes máximas.

Observa-se que, para a fundação mais leve, as diferentes propriedades das camadas do solo influenciam pouco na amplificação, apesar dos maiores deslocamentos associados a uma maior diferença entre os módulos de elasticidade das camadas. Já para a fundação mais pesada, ocorre maior resposta dinâmica, com o semiespaço rígido provocando deslocamentos maiores que o solo heterogêneo, que, por sua vez, gera amplitudes maiores que o solo homogêneo.

Quanto mais pesada a fundação, menor a frequência necessária para atingir o pico de amplificação. Os coeficientes adimensionais e as frequências analisadas

confirmam a formulação teórica, mostrando que, para o solo homogêneo, os coeficientes de rigidez e amortecimento são praticamente independentes da frequência.

Nos casos heterogêneos e de semiespaço rígido, a variação desses coeficientes é mais acentuada, indicando que a consideração da frequência excitadora é fundamental.

5.1. SUGESTÕES PARA TRABALHOS FUTUROS

Entre as possíveis sugestões para trabalhos futuros, é possível citar:

- Considerar, na elaboração da matriz de flexibilidade, os três graus de liberdade translacionais, não apenas o vertical e avaliar a influência desses deslocamentos nos resultados;
- Avaliar, para o solo heterogêneo, o comportamento das funções de impedância rotacionais;
- Avaliar a distribuição e comportamento das funções de impedância para diversas geometrias diferentes;
- Realizar uma análise de sensibilidade para outros tipos de solo com diferentes camadas;
- Investigar o comportamento dinâmico de fundações considerando efeitos não-lineares no solo, como plasticidade ou efeitos de grandes deformações, para representar melhor situações reais;
- O desenvolvimento de uma interface gráfica, que torne o programa mais didático e de fácil manuseio e entendimento, de maneira que qualquer usuário possa, com a devida base teórica, calcular os valores para coeficientes das molas e dos amortecedores, e que assim possa ser dada a entrada em um programa comercial de análise estrutural.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, Igor Mastrianni de. ***Modelagem Numérica da Fundação de uma Máquina Rotativa considerando Rigidez e Amortecimento do Solo com Distribuição não homogênea***. UFRJ - Universidade Federal do Rio de Janeiro/ Escola Politécnica, Programa de Projeto de Estruturas, Rio de Janeiro-RJ, Brasil, 2015.

ASSOCIAÇÃO BRASILEIRA DE MECÂNICA DOS SOLOS E ENGENHARIA GEOTÉCNICA (ABMS) e ASSOCIAÇÃO BRASILEIRA DE EMPRESAS DE ENGENHARIA DE FUNDAÇÕES E GEOTECNIA (ABEF), ***Fundações, Teoria e Prática***, 2ª ed., Editora Pini, São Paulo, SP, Brasil, 2002.

AMERICAN CONCRETE INSTITUTE (ACI), ***ACI.351.3R.18: Foundation for Dynamic Equipment***, Farmington Hills – MI, USA, 2018.

CLOUGH, R.W. e PENZIEN, J., *Dynamics of Structures*, Second Edition, McGraw-Hill, New York, 1993.

COUTINHO, D. S. A, MENDES, C. H. F. ***Projecto Automatizado de Fundações de Máquinas***, Rio de Janeiro. Universidade Federal do Rio de Janeiro (UFRJ)/Escola Politécnica, 2007.

DEUTSCHES INSTITUT FÜR NORMUNG (DIN). ***DIN 4024 - Machine foundations, Flexible structures that support machines with rotating elements, Part 1***, 1988.

_____. ***Machine foundations, Rigid foundations for machinery subjected to periodic vibration, Part 2***, 1991.

ENCYCLOPÆDIA Britannica. ***Seismic Wave: Main Types***. Disponível em: <https://www.britannica.com/science/seismograph/Applications.of.the-ismograph#/media/1/532943/210259>. Acesso em 10 de novembro de 2024.

GAZETAS, G., DOBRY, R., TASSOULAS, J.L., ***Vertical Response of Arbitrarily Shaped Embedded Foundations***, Journal of Geotechnical Engineering, ASCE, Vol. 111, No. 6, Junho 1985.

GAZETAS, G., TASSOULAS, J. L., ***Horizontal Stiffness of Arbitrarily Shaped Embedded Foundations***, Journal of Geotechnical Engineering, ASCE, Vol. 113, No. 5, Maio 1987.

GAZETAS, G. et al., ***Dynamic Interaction Factors for Floating Pile Groups***, Journal of Geotechnical Engineering, ASCE, Vol. 117, Nº10. October 1991, p.1531.1548.

GAZETAS, G., ***Analysis of Machine Foundation Vibrations: State of the Art, Soil Dynamics and Earthquake Engineering***, Vol. 2, Nº. 1, 1983.

ISO 2372. ***Mechanical Vibration of Machines with Operation Speeds from 10 to 200 rev/s***, Grundlang for vurdering, 1983

LUCO, J. E., ***Impedance Functions for a Rigid Foundation on a Layered Medium***, Nuclear Engineering and Design Vol.31, 1974, p.204.217.

NOVAK, M., ABOUL.ELLA, F., PILAY. ***A Computer Program for Calculation of Stiffness and Damping of Piles in Layered Media, Systems Analysis Control and Design Activity (SACDA)***, The University of Western Ontario, London, Canada, December 1977.

OHTA, Yutaka; GOTO, Noritoshi. Empirical shear wave velocity equations in terms of characteristic soil indexes. ***Earthquake engineering & structural dynamics***, v. 6, n. 2, p. 167-187, 1978.

OLIVEIRA, A. V. B. R., ***Análise Numérica de Funções de Impedância em Solo Homogêneo***. Rio de Janeiro. Universidade Federal do Rio de Janeiro (UFRJ)/Escola Politécnica, 2021.

OLIVEIRA, A. V. B. R., SANTOS, S. H. C., ***Análise Numérica de Funções de Impedância em Solo Homogêneo***. XIV Congresso Brasileiro de Pontes e Estruturas, 2023.

PETROBRAS, ***N.1848 - Projeto de Fundações de Máquinas***, Rio de Janeiro, Brasil, 2023.

POULOS, H. G., DAVIS, E. H., ***Elastic Solutions for Soil and Rock Mechanics***, New York, John Wiley & Sons, 1974 (Series in soil engineering).

POULOS, H. G., DAVIS, E. H., ***Pile Foundation Analysis and Design***, New York, John Wiley & Sons, 1980.

RIBEIRO, Marco Antônio Amâncio. ***Análise de Fadiga em Estrutura de Fundação de Torre de Turbina Eólica***. Rio de Janeiro. Universidade Federal do Rio de Janeiro (UFRJ)/Escola Politécnica, 2014.

RICHART, F.E., et al.. ***Vibrations of Soils and Foundations***, Prentice-Hall, INC., Englewood Cliffs, NJ, 1970.

SANTOS, M. F. F., ***Estudo Bidimensional da Interação Dinâmica Estrutura-Solo-Estrutura***, Rio de Janeiro. Universidade Federal do Rio de Janeiro (UFRJ)/Escola Politécnica, 2018.

SANTOS, S.H.C e GUIMARÃES, J.T.G., ***Análise de Placas de Fundação considerando a Matriz de Rigidez do Solo Condensada na Superfície***, IX Congresso Latino-Americano e Ibérico sobre Métodos Computacionais para Engenharia, Córdoba, Argentina, 1988.

SANTOS, S.H.C., ***Fundações de Máquinas***, Universidade Federal do Rio de Janeiro (UFRJ), Escola Politécnica, Departamento de Estruturas, 2024.

SEED, H. Bolton; IDRIS, Izzat M.; ARANGO, Ignacio. Evaluation of liquefaction potential using field performance data. ***Journal of geotechnical engineering***, v. 109, n. 3, p. 458-482, 1983.

SOLDAN, Carla. ***Estudo Bidimensional da Rigidez do Solo Estratificado Sobre o Semi-Espaço para Análise Dinâmica***, Rio de Janeiro. Universidade Federal do Rio de Janeiro (UFRJ)/COPPE, 1999.

STOKOE, D. Sykora and KI. Correlations of in situ measurements in sands of shear wave velocity, soil characteristics, and site conditions. ***Geotechnical Engineering Report***, 1983.

SYKORA, David Wayne. ***Correlations of in situ measurements in sands of shear wave velocity, soil characteristics, and site conditions***. 1983. Tese de Doutorado. University of Texas at Austin.

WOLF, J.P., ***Dynamic Soil-Structure Interaction***, New Jersey, Prentice Hall, 1985.

WOLF, J.P., ***Foundation Vibration Analysis Using Simple Physical Models***, New Jersey, Prentice Hall, 1994.

APÊNDICES

I. CÓDIGO DESENVOLVIDO EM C++

I.I. main

```
#define _USE_MATH_DEFINES // Para usar constantes matemáticas como Pi

#include <iostream> // Para std
#include <cmath> // Funções matemáticas como 'sqrt' e 'pow'
#include <string> // Para manipulação de strings
#include <complex> // Para manipulação de números complexos
#include <windows.h> // Para utilizar a função SetConsoleOutputCP(GetACP())
#include <vector> // Para operações com vector
#include <Eigen/Dense> // Para operações matriciais
#include <Eigen/Core> // Para operações matriciais e vetoriais
#include <fstream> // Para manipular arquivos
#include <iomanip> // Para manipular formatação
#include <locale> // Para trocar saída de ponto para vírgula
#include <chrono> // Para marcar tempo de execução

// Chamando arquivos de referência
#include "inputs.h"
#include "outputs.h"
#include "funcoes.h"
#include "wolf.h"

using namespace std;
using namespace Eigen;

// WOLF Programa principal
int main()
{
    // Marca o tempo de início
    auto inicio = chrono::high_resolution_clock::now();

    SetConsoleOutputCP(GetACP()); // Para aceitar caracteres latinos

    cout << " +-----+
-----+\\n";
    cout << " | Cálculo de Funções de Impedância Por Meio da Função de Influência
de Flexibilidade |\\n";
    cout << " +-----+
-----+\\n";
    cout << " |
|\\n";
    cout << " | Desenvolvido por: André Vítor Bolanho Rocha de Oliveira
|\\n";
    cout << " | Orientação: Sergio Hampshire de Carvalho Santos, D.Sc.
|\\n";
    cout << " | Coorientação: Marco Felipe Fialho Santos, D.Sc.
|\\n";
    cout << " |
|\\n";
    cout << " | Universidade Federal do Rio de Janeiro (UFRJ)
|\\n";
    cout << " | Escola Politécnica
|\\n";
    cout << " | Programa de Projeto de Estruturas
|\\n";
```

```

cout << " +-----+
-----+\\n\\n";

int opcao = escolherOpcaoEntrada();

// Entrada dos dados do solo:
double g01, xi_hist1, poisson1, rho1, g02, xi_hist2, poisson2, rho2, d;
entradaDadosSolo(g01, xi_hist1, poisson1, rho1, g02, xi_hist2, poisson2,
rho2, d, opcao);

// Entrada dos dados da geometria da fundação:
double A, B, x_div, y_div;
entradaDadosGeometria(A, B, x_div, y_div, opcao);

// Executa o código para diversas frequências:

// Abre um arquivo para saída
ofstream outputFile("output.txt");

// Verifique se o arquivo foi aberto corretamente
if (!outputFile.is_open()) {
    cerr << "Erro ao abrir o arquivo de saída." << "\\n";
    return 1; // ou faça algo para lidar com o erro
}

// Vetor de valores específicos de a0
vector<double> valores_a0;
double klim;
entradaAux(valores_a0, klim, opcao);

// Declaração de vetores para armazenar os valores de a0, k e c
vector<double> a0_values;
vector<double> k_values;
vector<double> c_values;
vector<double> r_values;
vector<vector<complex<double>>> Wr_real;
vector<vector<complex<double>>> Wr_imag;

// Loop para diversos valores de a0
for (double a0 : valores_a0)
{
    cout << "Executando cálculos para a0 = " << a0 << "\\n";
    executar_calculos(a0, A, B, x_div, y_div, g01, xi_hist1, poisson1, rho1,
g02, xi_hist2, poisson2, rho2, d, klim, outputFile, a0_values, k_values,
c_values, r_values, Wr_real, Wr_imag);
    cout << "Cálculos finalizados para a0 = " << a0 << "\\n\\n";
}

// Gera as tabelas no arquivo de saída
gerar_tabelas(outputFile, a0_values, k_values, c_values, r_values, Wr_real,
Wr_imag);
//gerar_csv("output.csv", a0_values, k_values, c_values, r_values, Wr_real,
Wr_imag);

// Fecha o arquivo após terminar
outputFile.close();

cout << "\\nCálculos concluídos. Consulte o arquivo 'output.txt'\\n";

// Marca o tempo de término
auto fim = chrono::high_resolution_clock::now();

```

```

// Calcula a duração em minutos
chrono::duration<double, ratio<60>> duracao = fim - inicio;

// Exibe o tempo de execução
cout << "Tempo de execução: " << duracao.count() << " minutos" << "\n";

cout << "\n";
cout << "\n";
cout << "\n";
//Define string 'Tecla' para encerrar programa ao final do código
string Tecla;
cout << "Pressione qualquer tecla e em seguida 'ENTER' para encerrar o
programa\n";
cin >> Tecla;

return 0;
}

```

I.II.wolf.h

```

#ifndef WOLF_H
#define WOLF_H

#include <iostream>
#include <cmath>
#include <complex>
#include <vector>
#include <Eigen/Dense>
#include <fstream>

using namespace std;

// Declaração da função que realizará os cálculos
void executar_calculos(
    double a0, double A, double B, double x_div, double y_div,
    double g01, double xi_hist1, double poisson1, double rho1,
    double g02, double xi_hist2, double poisson2, double rho2,
    double profundidade_primeira_camada, double klim,
    ofstream& outputFile,
    vector<double>& a0_values,
    vector<double>& k_values,
    vector<double>& c_values,
    vector<double>& r_values,
    vector<vector<complex<double>>>& Wr_real,
    vector<vector<complex<double>>>& Wr_imag);

#endif // WOLF_H

```

I.III. wolf.cpp

```

#define _USE_MATH_DEFINES // Para usar constantes matemáticas como Pi

#include "wolf.h"
#include "funcoes.h"
#include "fww.h"
#include "integracao.h"
#include "w(r).h"
#include "s_z.h"
#include "pontos_malha.h"
#include "matriz_flexibilidade.h"
#include "inverte_matriz.h"
#include "coeficientes_adimensionais.h"

```

```

#include <boost/math/special_functions/bessel.hpp>
#include "outputs.h"

using namespace std;
using namespace Eigen;

using namespace boost::math;

// Função principal dentro do loop que chama as funções principais e realiza os cálculos
void executar_calculos(
    double a0, double A, double B, double x_div, double y_div,
    double g01, double xi_hist1, double poisson1, double rho1,
    double g02, double xi_hist2, double poisson2, double rho2,
    double profundidade_primeira_camada, double klim,
    ofstream& outputFile,
    vector<double>& a0_values,
    vector<double>& k_values,
    vector<double>& c_values,
    vector<double>& r_values,
    vector<vector<complex<double>>>& Wr_real,
    vector<vector<complex<double>>>& Wr_imag)
{
    // Chama as funções relacionadas à geometria da fundação
    double raio = raiofund(A, B);

    pair<double, double> dim_elem = dimelem(A, B, x_div, y_div);
    double tx = dim_elem.first;
    double ty = dim_elem.second;

    double a = raioelem(A, B, tx, ty);
    //cout << "Raio de cada elemento da fundação: " << a << "\n";

    // Chama as funções relacionadas aos dados do solo
    complex<double> g1(g01, 2 * xi_hist1);
    complex<double> g2(g02, 2 * xi_hist2);
    complex<double> cs1 = cs(g1, rho1);
    complex<double> cs2 = cs(g2, rho2);
    complex<double> cs_real = real(cs1);
    complex<double> cp1 = cp(g1, rho1, poisson1);
    complex<double> cp2 = cp(g2, rho2, poisson2);

    // Chama as funções relacionadas à frequência e número de onda
    double w = a0 * real(cs_real) / raio;
    complex<double> co_lambda = comp_onda(g1, rho1, w);
    // Tamanho máximo do elemento
    double raio_max = a_max(g1, rho1, w);

    // Profundidade da primeira camada
    double d = profundidade_primeira_camada;

    // Calcula integral da amplitude de deslocamento vertical W(r)
    double lim_inf = 1e-3;
    double lim_sup = klim;
    int n = 500;

    vector<complex<double>> Wr_values;

    for (double r = 0.0; r <= 1.0; r += 0.1)
    {

```

```

        complex<double> W_r = calcular_W_r(r, a, w, g1, g2, rho1, poisson1, rho2,
        poisson2, A, B, d, lim_inf, lim_sup, n);
        Wr_values.push_back(W_r);

        // Se for o primeiro valor de a0, adicione os valores de r ao vetor
r_values
        if (a0_values.empty()) {
            r_values.push_back(r);
        }
    }

    // Adiciona os valores de W(r) para a0 atual
    Wr_real.push_back(Wr_values);
    Wr_imag.push_back(Wr_values);

    // Rigidez dinâmica segundo WOLF (1985)
    complex<double> Sz = calcular_Sz(a, w, g1, g2, rho1, poisson1, rho2,
    poisson2, A, B, d, lim_inf, lim_sup, n);

    // Rigidez estática segundo WOLF (1985)
    double Ks = 1.82 * M_PI * g01 * a;

    // Coeficiente de Rigidez adimensional
    double k0 = real(Sz);

    // Coeficiente de Amortecimento adimensional
    double c0 = imag(Sz) / a0;

    // Cálculo da matriz de flexibilidade
    vector<vector<complex<double>>> matriz_fv = calcular_matriz_fv(a, w, g1, g2,
    rho1, poisson1, rho2, poisson2, A, B, d, lim_inf, lim_sup, n, k0, c0, x_div,
    y_div);

    // Inversão da matriz de flexibilidade
    MatrixXd fv_inversa = inverter_matriz(matriz_fv);

    // Cálculo das matrizes de rigidez e de amortecimento, além dos valores
    totais
    MatrixXd K, C;
    VectorXd K_soma, C_soma;
    double K_calculado, C_calculado;
    calcular_rigidez_amortecimento(fv_inversa, K, C, K_soma, C_soma, K_calculado,
    C_calculado);

    // Calculando o raio equivalente
    double r = sqrt((4 * A / 2 * B / 2) / M_PI);

    // Calculando os valores teóricos
    double K_teorico = calcular_K_teorico(g01, poisson1, A / 2, B / 2);
    double C_teorico = calcular_C_teorico(r, poisson1, rho1, g01);

    // Calculando k e c
    double k = calcular_k(K_calculado, K_teorico);
    double c = calcular_c(C_calculado, C_teorico, w);

    // Adiciona os valores de a0, k e c aos vetores
    a0_values.push_back(a0);
    k_values.push_back(k);
    c_values.push_back(c);
}

```

I.IV. inputs.h

```
#ifndef INPUTS_H
#define INPUTS_H

#include <iostream>
#include <fstream> // Para leitura de arquivos ('ifstream')
#include <sstream> // Para usar stringstream na leitura de arquivos
#include <iomanip>
#include <vector>
#include "confirmacao.h"

using namespace std;

int escolherOpcaoEntrada();

void entradaDadosSolo(
    // inputs
    double& g01, double& xi_hist1, double& poisson1, double& rho1,
    double& g02, double& xi_hist2, double& poisson2, double& rho2,
    double& profundidade_primeira_camada,
    int opcao);

void entradaDadosGeometria(
    //inputs
    double& A, double& B, double& x_div, double& y_div,
    int opcao);

void entradaAux(vector<double>& a0, double& klim, int opcao);

#endif // INPUTS_H
```

I.V. inputs.cpp

```
#include "confirmacao.h"
#include "inputs.h"

using namespace std;

int escolherOpcaoEntrada()
{
    cout << "\nEscolha a opção de entrada:\n";
    cout << "[1] Digitar manualmente\n";
    cout << "[2] Ler de um arquivo de texto\n";
    cout << "Opção: ";
    int opcao;

    // Verifica se a leitura foi bem-sucedida e se o próximo caractere no buffer
    não é uma quebra de linha
    while (!(cin >> opcao) || cin.peek() != '\n')
    {
        // Limpa o buffer de entrada
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        cout << "\nOpção inválida.\nPor favor, digite novamente (1 ou 2): ";
    }
    return opcao;
}
```



```

void entradaDadosSolo(double& g01, double& xi_hist1, double& poisson1, double&
rho1, double& g02, double& xi_hist2, double& poisson2, double& rho2, double&
profundidade_primeira_camada, int opcao)
{
    int confirma;

    do
    {
        if (opcao == 1)
        {
            cout << "\nEntrada dos dados do solo:\n\n";

            cout << "Módulo de Elasticidade Transversal da Camada Superior: ";
            cin >> g01;
            cout << "Coeficiente de Amortecimento Histerético da Camada Superior: ";
            cin >> xi_hist1;
            cout << "Coeficiente de Poisson da Camada Superior: ";
            cin >> poisson1;
            cout << "Massa Específica da Camada Superior: ";
            cin >> rho1;

            cout << "Módulo de Elasticidade Transversal do Subespaço: ";
            cin >> g02;
            cout << "Coeficiente de Amortecimento Histerético do Subespaço: ";
            cin >> xi_hist2;
            cout << "Coeficiente de Poisson do Subespaço: ";
            cin >> poisson2;
            cout << "Massa Específica do Subespaço: ";
            cin >> rho2;

            cout << "Profundidade da Primeira Camada de Solo: ";
            cin >> profundidade_primeira_camada;
        }
        else if (opcao == 2)
        {
            ifstream arquivo("inputs.txt");
            if (!arquivo)
            {
                cerr << "Erro ao abrir o arquivo." << "\n";
                return;
            }

            string linha;
            while (getline(arquivo, linha))
            {
                if (linha.find("Módulo de Elasticidade Transversal da Camada
Superior") != string::npos)
                {
                    stringstream ss(linha.substr(linha.find('=') + 1));
                    ss >> g01;
                }
                else if (linha.find("Coeficiente de Amortecimento Histerético da
Camada Superior") != string::npos)
                {
                    stringstream ss(linha.substr(linha.find('=') + 1));
                    ss >> xi_hist1;
                }
                else if (linha.find("Coeficiente de Poisson da Camada Superior")
!= string::npos)
                {
                    stringstream ss(linha.substr(linha.find('=') + 1));

```

```

        ss >> poisson1;
    }
    else if (linha.find("Massa Específica da Camada Superior") !=
string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> rho1;
    }
    else if (linha.find("Módulo de Elasticidade Transversal do
Subespaço") != string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> g02;
    }
    else if (linha.find("Coeficiente de Amortecimento Histerético do
Subespaço") != string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> xi_hist2;
    }
    else if (linha.find("Coeficiente de Poisson do Subespaço") !=
string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> poisson2;
    }
    else if (linha.find("Massa Específica do Subespaço") !=
string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> rho2;
    }
    else if (linha.find("Profundidade da Primeira Camada de Solo") !=
string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> profundidade_primeira_camada;
    }
}
cout << "\nEntrada dos dados do solo lidos do arquivo:\n\n";
cout << "Módulo de Elasticidade Transversal da Camada Superior: " <<
g01 << "\n";
cout << "Coeficiente de Amortecimento Histerético da Camada Superior:
" << xi_hist1 << "\n";
cout << "Coeficiente de Poisson da Camada Superior: " << poisson1 <<
"\n";
cout << "Massa Específica da Camada Superior: " << rho1 << "\n";
cout << "Módulo de Elasticidade Transversal do Subespaço: " << g02 <<
"\n";
cout << "Coeficiente de Amortecimento Histerético do Subespaço: " <<
xi_hist2 << "\n";
cout << "Coeficiente de Poisson do Subespaço: " << poisson2 << "\n";
cout << "Massa Específica do Subespaço: " << rho2 << "\n";
cout << "Profundidade da Primeira Camada de Solo: " <<
profundidade_primeira_camada << "\n";

    arquivo.close();
}
else
{
    cout << "\nOpção inválida. Por favor, escolha 1 ou 2\n";
}

```

```

        opcao = escolherOpcaoEntrada(); // Solicita ao usuário para escolher
novamente
        continue; // Volta ao início do loop para tratar a nova opção
    }
    confirma = confirmaDados();

    // Se a confirmação for 2 (ou seja, o usuário deseja refazer a entrada),
    // e a opção original era 2 (leitura do arquivo), alteramos a opção para
1
    // para que a próxima iteração do loop solicite os dados manualmente.
    if (confirma == 2 && opcao == 2)
    {
        opcao = 1;
    }

    } while (confirma != 1); // Continua solicitando dados até que o usuário
confirme
}

```

```

void entradaDadosGeometria(double& A, double& B, double& x_div, double& y_div,
int opcao)
{
    int confirma;

    do
    {
        if (opcao == 1)
        {
            cout << "\nDimensões da fundação:\n\n";

            cout << "A = ";
            cin >> A;
            cout << "B = ";
            cin >> B;
            cout << "Número de Divisões em X = ";
            cin >> x_div;
            cout << "Número de Divisões em Y = ";
            cin >> y_div;
        }
        else if (opcao == 2)
        {
            ifstream arquivo("inputs.txt");
            if (!arquivo) {
                cerr << "Erro ao abrir o arquivo." << "\n";
                return;
            }

            string linha;
            while (getline(arquivo, linha))
            {
                if (linha.find("Dimensão 'A' da fundação 'A x B'") !=
string::npos)
                {
                    stringstream ss(linha.substr(linha.find('=') + 1));
                    ss >> A;
                }
                else if (linha.find("Dimensão 'B' da fundação 'A x B'") !=
string::npos)
                {
                    stringstream ss(linha.substr(linha.find('=') + 1));
                    ss >> B;
                }
            }
        }
    } while (confirma != 1);
}

```

```

    }
    else if (linha.find("Número de Divisões em X") != string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> x_div;
    }
    else if (linha.find("Número de Divisões em Y") != string::npos)
    {
        stringstream ss(linha.substr(linha.find('=') + 1));
        ss >> y_div;
    }
}

cout << "\nDimensões da fundação lidos do arquivo:\n\n";
cout << "A = " << A << "\n";
cout << "B = " << B << "\n";
cout << "Número de Divisões em X = " << x_div << "\n";
cout << "Número de Divisões em Y = " << y_div << "\n";

arquivo.close();
}

else {
    cout << "\nOpção inválida. Por favor, escolha 1 ou 2\n";
    opcao = escolherOpcaoEntrada(); // Solicita ao usuário para escolher
novamente
    continue; // Volta ao início do loop para tratar a nova opção
}
confirma = confirmaDados();

// Se a confirmação for 2 (ou seja, o usuário deseja refazer a entrada),
// e a opção original era 2 (leitura do arquivo), alteramos a opção para
1
// para que a próxima iteração do loop solicite os dados manualmente.
if (confirma == 2 && opcao == 2)
{
    opcao = 1;
}

} while (confirma != 1); // Continua solicitando dados até que o usuário
confirme
}

void entradaAux(vector<double>& a0, double& klim, int opcao)
{
    int confirma;

    do
    {
        if (opcao == 1)
        {
            cout << "\nDigite os valores de a0 separados por vírgula: \n\n";
            cout << "a0 = ";
            string input;
            cin.ignore(); // Limpa o buffer de entrada
            getline(cin, input);

            stringstream ss(input);
            string valor;

            while (getline(ss, valor, ','))
            {

```

```

        a0.push_back(stod(valor));
    }

    cout << "\nDigite o valor de klim (Número de onda limite): ";
    cin >> klim;
}
else if (opcao == 2)
{
    ifstream arquivo("inputs.txt");
    if (!arquivo)
    {
        cerr << "Erro ao abrir o arquivo." << "\n";
        return;
    }

    string linha;
    while (getline(arquivo, linha))
    {
        if (linha.find("Frequências adimensionais 'a0' analisadas
(separadas por vírgula)") != string::npos)
        {
            stringstream ss(linha.substr(linha.find('=') + 1));
            string valor;

            while (getline(ss, valor, ','))
            {
                a0.push_back(stod(valor));
            }
        }
        else if (linha.find("Número de onda limite (limite superior para
integração 'klim')") != string::npos)
        {
            stringstream ss(linha.substr(linha.find('=') + 1));
            ss >> klim;
        }
    }

    cout << "\nValores de a0 lidos do arquivo:\n";
    for (size_t i = 0; i < a0.size(); ++i)
    {
        cout << fixed << setprecision(1) << a0[i];
        if (i < a0.size() - 1)
        {
            cout << ", ";
        }
    }
    cout << "\n";
    cout << "\nNúmero de onda limite para integração ('klim'): " << klim
<< "\n";

    arquivo.close();
}
else
{
    cout << "\nOpção inválida. Por favor, escolha 1 ou 2\n";
    opcao = escolherOpcaoEntrada(); // Solicita ao usuário para escolher
novamente
    continue; // Volta ao início do loop para tratar a nova opção
}
confirma = confirmaDados();

```

```

        // Se a confirmação for 2 (ou seja, o usuário deseja refazer a entrada),
        // e a opção original era 2 (leitura do arquivo), alteramos a opção para
1
        // para que a próxima iteração do loop solicite os dados manualmente.
        if (confirma == 2 && opcao == 2)
        {
            opcao = 1;
        }

    } while (confirma != 1); // Continua solicitando dados até que o usuário
confirme
}

```

I.VI. outputs.h

```

#ifndef OUTPUTS_H
#define OUTPUTS_H

#include <iostream>
#include <vector>
#include <complex>
#include <iomanip>
#include <fstream>
#include <string>
#include <locale>

#include "amplificacao_dinamica.h"

using namespace std;

// Função para gerar linha delimitadora de tabela
string gerar_linha_delimitadora(size_t num_colunas, size_t largura_coluna);

// Função para gerar cabeçalho de tabela
string gerar_cabecalho_tabela(const vector<double>& valores, size_t
largura_coluna);

// Função para gerar as tabelas no arquivo de saída
void gerar_tabelas(
    ofstream& outputFile,
    const vector<double>& a0_values,
    const vector<double>& k_values,
    const vector<double>& c_values,
    const vector<double>& r_values,
    const vector<vector<complex<double>>>& Wr_real,
    const vector<vector<complex<double>>>& Wr_imag);

void trocar_ponto_por_virgula(ofstream& outputFile);

string formatar_numero(double valor);

void gerar_csv(
    const string& nome_arquivo,
    const vector<double>& a0_values,
    const vector<double>& k_values,
    const vector<double>& c_values,
    const vector<double>& r_values,
    const vector<vector<complex<double>>>& Wr_real,
    const vector<vector<complex<double>>>& Wr_imag);

#endif // OUTPUTS_H

```

I.VII. outputs.cpp

```
#include "outputs.h"
#include "windows.h"

using namespace std;

// Função para gerar linha delimitadora de tabela
string gerar_linha_delimitadora(size_t num_colunas, size_t largura_coluna)
{
    string linha = "+-----+";
    for (size_t i = 0; i < num_colunas-1; ++i) {
        linha += string(largura_coluna+1, '-') + "+";
    }
    return linha + "\n";
}

// Função para gerar cabeçalho de tabela
string gerar_cabecalho_tabela(const vector<double>& valores, size_t
largura_coluna)
{
    string cabecalho = "|      r      |";
    for (const auto& valor : valores) {
        stringstream ss;
        ss << fixed << setw(10) << setprecision(2) << setw(10) << valor;
        cabecalho += ss.str() + " |";
    }
    cabecalho += "\n";
    cabecalho = gerar_linha_delimitadora(valores.size() + 1, largura_coluna) +
cabecalho;
    return cabecalho;
}

// Função para gerar as tabelas no arquivo de saída
void gerar_tabelas(
    ofstream& outputFile,
    const vector<double>& a0_values,
    const vector<double>& k_values,
    const vector<double>& c_values,
    const vector<double>& r_values,
    const vector<vector<complex<double>>>& Wr_real,
    const vector<vector<complex<double>>>& Wr_imag)
{
    SetConsoleOutputCP(GetACP()); // Para aceitar caracteres latinos

    size_t largura_coluna = 10;

    // Trocar ponto por vírgula no arquivo de saída
    trocar_ponto_por_virgula(outputFile);

    // Escreve a Tabela 1: Parte Real de W(r)
    outputFile << "Tabela 1: Parte Real de W(r)\n";
    outputFile << gerar_cabecalho_tabela(a0_values, largura_coluna);
    for (size_t j = 0; j < r_values.size(); ++j) {
        outputFile << "| " << fixed << setw(5) << setprecision(1) << setw(7) <<
r_values[j] << " |";
        for (size_t i = 0; i < Wr_real.size(); ++i) {
            outputFile << setw(largura_coluna) << setprecision(5) <<
real(Wr_real[i][j]) << " |";
        }
        outputFile << "\n";
    }
    outputFile << gerar_linha_delimitadora(a0_values.size() + 1, largura_coluna);
}
```

```

outputFile << "\n";

// Escreve a Tabela 2: Parte Imaginária de W(r)
outputFile << "Tabela 2: Parte Imaginária de W(r)\n";
outputFile << gerar_cabecalho_tabela(a0_values, largura_coluna);
for (size_t j = 0; j < r_values.size(); ++j) {
    outputFile << "| " << fixed << setw(5) << setprecision(1) << setw(7) <<
r_values[j] << " |";
    for (size_t i = 0; i < Wr_imag.size(); ++i) {
        outputFile << setw(largura_coluna) << setprecision(5) <<
imag(Wr_imag[i][j]) << " |";
    }
    outputFile << "\n";
}
outputFile << gerar_linha_delimitadora(a0_values.size() + 1, largura_coluna);
outputFile << "\n";

// Escreve a tabela de a0, k e c
outputFile << "Tabela 3: Coeficientes dimensionais\n";
outputFile << "+-----+-----+-----+\n";
outputFile << "| a0 | k | c | \n";
for (size_t i = 0; i < a0_values.size(); ++i) {
    outputFile << "| " << fixed << setprecision(1) << setw(4) << a0_values[i]
<< " | "
    << setw(8) << setprecision(5) << k_values[i] << " | "
    << setw(8) << setprecision(5) << c_values[i] << " |\n";
}
outputFile << "+-----+-----+-----+\n";
outputFile << "\n";

outputFile << "Tabela 4: Amplificação Dinâmica\n";
outputFile << "+-----+-----+-----+\n";
outputFile << "| a0 | Amp m=1 | Amp m=5 | \n";

for (size_t i = 0; i < a0_values.size(); ++i) {
    double a0 = a0_values[i];
    double k = k_values[i];
    double c = c_values[i];

    // Calcular amplificação para m = 1 e m = 5
    double amplificacao_m1 = amplificacao_dinamica(a0, k, c, 1.0);
    double amplificacao_m5 = amplificacao_dinamica(a0, k, c, 5.0);

    outputFile << "| " << fixed << setprecision(2) << setw(5) << a0 << " | "
    << setw(9) << setprecision(5) << amplificacao_m1 << " | "
    << setw(9) << setprecision(5) << amplificacao_m5 << " |\n";
}

outputFile << "+-----+-----+-----+\n";
}

// Função para trocar ponto por vírgula
void trocar_ponto_por_virgula(ofstream& outputFile) {
    class CommaDecimal : public numpunct<char> {
    protected:
        char do_decimal_point() const override {
            return ','; // Return comma as the decimal separator
        }
    };

    locale comma_locale(locale::classic(), new CommaDecimal);

```



```

        outputFile.imbue(comma_locale);
    }

    // Função para formatar números com vírgula como separador decimal
    string formatar_numero(double valor) {
        ostreamstream ss;
        ss.imbue(locale(ss.getloc(), new numpunct<char>()));
        ss << fixed << setprecision(2) << valor;
        return ss.str();
    }

    class custom_numpunct : public std::numpunct<char> {
    protected:
        char do_decimal_point() const override {
            return ',';
        }
        char do_thousands_sep() const override {
            return '.';
        }
        std::string do_grouping() const override {
            return "\003";
        }
    };

    // Função para gerar CSV
    void gerar_csv(
        const string& nome_arquivo,
        const vector<double>& a0_values,
        const vector<double>& k_values,
        const vector<double>& c_values,
        const vector<double>& r_values,
        const vector<vector<complex<double>>>& Wr_real,
        const vector<vector<complex<double>>>& Wr_imag)
    {
        ofstream arquivo(nome_arquivo);

        if (!arquivo) {
            cerr << "Não foi possível criar o arquivo " << nome_arquivo << "\n";
            return;
        }

        // Configura o local e o formato de ponto decimal
        arquivo.imbue(locale(arquivo.getloc(), new custom_numpunct));

        // Tabela 1: Parte Real de W(r)
        arquivo << "Tabela 1: Parte Real de W(r)\n";
        arquivo << "r";
        for (const auto& valor : a0_values) {
            arquivo << ";" << formatar_numero(valor);
        }
        arquivo << "\n";

        for (size_t j = 0; j < r_values.size(); ++j) {
            arquivo << formatar_numero(r_values[j]);
            for (size_t i = 0; i < Wr_real.size(); ++i) {
                arquivo << ";" << formatar_numero(real(Wr_real[i][j]));
            }
            arquivo << "\n";
        }

        // Tabela 2: Parte Imaginária de W(r)
        arquivo << "\nTabela 2: Parte Imaginária de W(r)\n";
    }

```

```

arquivo << "r";
for (const auto& valor : a0_values) {
    arquivo << ";" << formatar_numero(valor);
}
arquivo << "\n";

for (size_t j = 0; j < r_values.size(); ++j) {
    arquivo << formatar_numero(r_values[j]);
    for (size_t i = 0; i < Wr_imag.size(); ++i) {
        arquivo << ";" << formatar_numero(imag(Wr_imag[i][j]));
    }
    arquivo << "\n";
}

// Tabela 3: Coeficientes adimensionais
arquivo << "\nTabela 3: Coeficientes adimensionais\n";
arquivo << "a0;k;c\n";
for (size_t i = 0; i < a0_values.size(); ++i) {
    arquivo << formatar_numero(a0_values[i]) << ";"
        << formatar_numero(k_values[i]) << ";"
        << formatar_numero(c_values[i]) << "\n";
}

// Tabela 4: Amplificação Dinâmica
arquivo << "\nTabela 4: Amplificação Dinâmica\n";
arquivo << "a0;k;c;Amplificação m=1;Amplificação m=5\n";

for (size_t i = 0; i < a0_values.size(); ++i) {
    double a0 = a0_values[i];
    double k = k_values[i];
    double c = c_values[i];

    // Calcular amplificação para m = 1 e m = 5
    double amplificacao_m1 = amplificacao_dinamica(a0, k, c, 1.0);
    double amplificacao_m5 = amplificacao_dinamica(a0, k, c, 5.0);

    arquivo << formatar_numero(a0) << ";"
        << formatar_numero(amplificacao_m1) << ";"
        << formatar_numero(amplificacao_m5) << "\n";
}

arquivo.close();
}

```

I.VIII. confirmacao.h

```

#ifndef CONFIRMACAO_H
#define CONFIRMACAO_H

int confirmaDados();

#endif // CONFIRMACAO_H

```

I.IX. confirmacao.cpp

```

#include <iostream>
#include "confirmacao.h"

using namespace std;

int confirmaDados() {
    int opcao;

```

```

bool confirmacaoFeita = false;

while (!confirmacaoFeita)
{
    cout << "\nConfirma os dados?\nSIM [1] ou NÃO [2] \n";
    if (cin >> opcao) {
        // A leitura foi bem sucedida
        switch (opcao) {
            case 1:
                cout << "SIM" << '\n' << '\n';
                confirmacaoFeita = true;
                break;
            case 2:
                cout << "NÃO, por favor insira os dados novamente." << '\n' <<
'\n';
                return opcao; // para voltar para antes do início do while
            default:
                cout << "Opção Inválida. Por favor, insira 1 ou 2." << "\n";
                // Limpa o buffer de entrada para evitar loops infinitos
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }
    else {
        // A leitura falhou (um caractere inválido foi digitado)
        cout << "Entrada inválida. Por favor, insira um número inteiro
correspondente às opções [1] ou [2]." << endl;
        // Limpa o buffer de entrada para evitar loops infinitos
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
}

return opcao;
}

```

I.X. funcoes.h

```

#ifndef FUNCOES_H
#define FUNCOES_H

#include <complex>

using namespace std;

// Funções relacionadas à geometria da fundação
double raiofund(double A, double B);
pair<double, double> dimelem(double A, double B, double x_div, double y_div);
double raioelem(double A, double B, double& tx, double& ty);

// Funções relacionadas aos dados do solo
complex<double> cs(complex<double> g, double rho);
complex<double> cp(complex<double> g, double rho, double poisson);
complex<double> comp_onda(complex<double> g, double rho, double w);
complex<double> s(double k, double w, complex<double> g, double rho, double
poisson);
complex<double> t(double k, double w, complex<double> g, double rho);
double a_max(complex<double> g, double rho, double w);

#endif // FUNCOES_H

```

I.XI. funcoes.cpp

```
#define _USE_MATH_DEFINES

#include <cmath>
#include <complex>
#include <iostream>

#include "funcoes.h"

using namespace std;

// Cálculo do raio da fundação
double raiofund(double A, double B)
{
    double raio = sqrt(A * B / M_PI);
    return raio;
}

// Cálculo das dimensões em x e y de cada elemento da malha
pair<double, double> dimelem(double A, double B, double x_div, double y_div)
{
    double tx = A / x_div;
    double ty = B / y_div;
    return make_pair(tx, ty);
}

// Cálculo do raio de um elemento da malha
double raioelem(double A, double B, double& tx, double& ty)
{
    double raio_elem = sqrt(tx * ty / M_PI);
    return raio_elem;
}

// Cálculo da velocidade de onda secundária ou de cisalhamento
complex<double> cs(complex<double> g, double rho)
{
    complex<double> cs = sqrt(g / rho);
    return cs;
}

// Cálculo da velocidade de onda primária ou de compressão
complex<double> cp(complex<double> g, double rho, double poisson)
{
    complex<double> cs0 = cs(g, rho);
    complex<double> cp = cs0 * sqrt((2 - 2 * poisson) / (1 - 2 * poisson));
    return cp;
}

// Cálculo do comprimento de onda
complex<double> comp_onda(complex<double> g, double rho, double w)
{
    complex<double> cs0 = real(cs(g, rho));
    complex<double> co_lambda = cs0 * (2.0 * M_PI / w);
    return co_lambda;
}

// Cálculo da função s(k)
complex<double> s(double k, double w, complex<double> g, double rho, double poisson)
{

```

```

        complex<double> s = -1i * sqrt(1.0 - (pow(w, 2) / (pow(cp(g, rho, poisson),
2) * pow(k, 2))));
        return s;
    }

    // Cálculo da função t(k)
    complex<double> t(double k, double w, complex<double> g, double rho)
    {
        complex<double> t = -1i * sqrt(1.0 - (pow(w, 2) / (pow(cs(g, rho), 2) *
pow(k, 2))));
        return t;
    }

    // Cálculo da máxima dimensão do elemento
    double a_max(complex<double> g, double rho, double w)
    {
        double cs0 = real(cs(g, rho));
        return (M_PI * cs0) / (6 * w);
    }
}

```

I.XII. fww.h

```

#ifndef FWW_H
#define FWW_H

#include <complex>
#include <fstream>
#include <iostream>

using namespace std;

// Função de influência de flexibilidade vertical para o solo HOMOGÊNEO
complex<double> fww_homogeneo(double k, double w, complex<double> g, double rho,
double poisson);

// Função que calcula a matriz S
void S(
    //inputs
    double k, double w, complex<double> g1, complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2,
    double A, double B, double d,
    //return
    complex<double> S_matrix[4][4]);

// Função de influência de flexibilidade vertical para o solo HETEROGÊNEO
complex<double> fww_estratificado(double k, double w, complex<double> g1,
complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2,
    double A, double B, double d);

// Função para organizar valores do gráfico de FWW vs. k
void calcularFww_x_k(double klim, double delta_k, double w, complex<double> g1,
complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2, double A, double
B, double d, ofstream& output_file);

#endif // FWW_H

```

I.XIII. fww.cpp

```
#include <cmath>
#include <complex>
#include <iostream>
#include <Eigen/Dense>
#include <cstdlib>
#include <iomanip>

#include "fww.h"
#include "funcoes.h"
#include "inverte_matriz.h"

using namespace std;
using namespace Eigen;

// Função de influência de flexibilidade vertical para o solo HOMOGÊNEO
complex<double> fww_homogeneo(double k, double w, complex<double> g, double rho,
double poisson)
{
    complex<double> s_val = s(k, w, g, rho, poisson);
    complex<double> t_val = t(k, w, g, rho);

    // Componentes individuais
    complex<double> numerador = 1i * s_val + 1i * s_val * pow(t_val, 2);
    complex<double> denominador = pow(1.0 - pow(t_val, 2), 2) + 4.0 * s_val *
t_val;
    complex<double> fator = -1.0 / (k * g);

    // Resultado final
    complex<double> fww = fator * numerador / denominador;

    return fww;
}

// Rigidez

// Função que calcula a matriz S
void S(double k, double w, complex<double> g1, complex<double> g2,
double rho1, double poisson1, double rho2, double poisson2,
double A, double B, double d, complex<double> S_matrix[4][4])
{
    // Cálculo dos valores s e t
    complex<double> s1 = s(k, w, g1, rho1, poisson1);
    complex<double> s2 = s(k, w, g2, rho2, poisson2);
    complex<double> t1 = t(k, w, g1, rho1);
    complex<double> t2 = t(k, w, g2, rho2);

    // Definição de outras variáveis necessárias
    complex<double> D = 2.0 * (1.0 - cos(k * s1 * d) * cos(k * t1 * d)) +
(s1 * t1 + 1.0 / (s1 * t1)) * sin(k * s1 * d) * sin(k * t1 * d);

    // Matriz S

    // Cálculo dos elementos da matriz
    // Linha 0
    S_matrix[0][0] = (1.0 + pow(t1, 2)) * k * g1 / D * (1.0 / t1 * cos(k * s1 *
d) * sin(k * t1 * d) + s1 * sin(k * s1 * d) * cos(k * t1 * d));
    S_matrix[0][1] = (1.0 + pow(t1, 2)) * k * g1 / D * ((3.0 - pow(t1, 2)) / (1.0
+ pow(t1, 2)) * (1.0 - cos(k * s1 * d) * cos(k * t1 * d)) + (1.0 + 2.0 * pow(s1,
2) * pow(t1, 2) - pow(t1, 2)) / (s1 * t1 * (1.0 + pow(t1, 2))) * sin(k * s1 * d)
* sin(k * t1 * d));
```

```

    S_matrix[0][2] = (1.0 + pow(t1, 2)) * k * g1 / D * (-s1 * sin(k * s1 * d) -
1.0 / t1 * sin(k * t1 * d));
    S_matrix[0][3] = (1.0 + pow(t1, 2)) * k * g1 / D * (cos(k * s1 * d) - cos(k *
t1 * d));

    // Linha 1
    S_matrix[1][0] = S_matrix[0][1];
    S_matrix[1][1] = (1.0 + pow(t1, 2)) * k * g1 / D * (1.0 / s1 * sin(k * s1 *
d) * cos(k * t1 * d) + t1 * cos(k * s1 * d) * sin(k * t1 * d));
    S_matrix[1][2] = (1.0 + pow(t1, 2)) * k * g1 / D * (-cos(k * s1 * d) + cos(k
* t1 * d));
    S_matrix[1][3] = (1.0 + pow(t1, 2)) * k * g1 / D * (-1.0 / s1 * sin(k * s1 *
d) - t1 * sin(k * t1 * d));

    // Linha 2
    S_matrix[2][0] = S_matrix[0][2];
    S_matrix[2][1] = S_matrix[1][2];
    S_matrix[2][2] = S_matrix[0][0] + k * g2 * ((1i * s2 * (1.0 + pow(t2, 2))) /
(1.0 + s2 * t2));
    S_matrix[2][3] = (1.0 + pow(t1, 2)) * k * g1 / D * ((pow(t1, 2) - 3.0) / (1.0
+ pow(t1, 2)) * (1.0 - cos(k * s1 * d) * cos(k * t1 * d)) + (pow(t1, 2) - 2.0 *
pow(s1, 2) * pow(t1, 2) - 1.0) / (s1 * t1 * (1.0 + pow(t1, 2))) * sin(k * s1 * d)
* sin(k * t1 * d)) + k * g2 * (2.0 - (1.0 + pow(t2, 2)) / (1.0 + s2 * t2));

    // Linha 3
    S_matrix[3][0] = S_matrix[0][3];
    S_matrix[3][1] = S_matrix[1][3];
    S_matrix[3][2] = S_matrix[2][3];
    S_matrix[3][3] = S_matrix[1][1] + k * g2 * ((1i * t2 * (1.0 + pow(t2, 2))) /
(1.0 + s2 * t2));
}

// Função de influência de flexibilidade vertical para o solo HETEROGÊNEO
complex<double> fww_estratificado(double k, double w, complex<double> g1,
complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2,
    double A, double B, double d)
{
    // Calcula a matriz S
    complex<double> S_matrix[4][4];
    S(k, w, g1, g2, rho1, poisson1, rho2, poisson2, A, B, d, S_matrix);

    // Calcula a inversa da matriz S
    // Converte a matriz S para o formato Eigen
    MatrixXcd S_eigen(4, 4);
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 4; ++j) {
            S_eigen(i, j) = S_matrix[i][j]; // Não precisa mais acessar .real()
        }
    }
    // Calcula a inversa usando Eigen
    MatrixXcd S_inv_eigen;
    if (!inverseUsingEigen(S_eigen, S_inv_eigen)) {
        cout << "Erro ao inverter a matriz S. Provável singularidade.\n Por
favor, revise os dados de entrada declarados" << "\n";
        exit(EXIT_FAILURE);
    }
    // Converte a matriz inversa de volta para o tipo complex<double>
    complex<double> S_inv[4][4];
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 4; ++j) {
            S_inv[i][j] = S_inv_eigen(i, j);
        }
    }
}

```

```

    }
}

// Retorna o elemento (2,2) da matriz inversa
return S_inv[1][1];
}

// Função para organizar valores do gráfico de FWW vs. k
void calcularFww_x_k(double klim, double delta_k, double w, complex<double> g1,
complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2, double A, double
B, double d, ofstream& output_file)
{
    output_file << "\tk\tReal(Fww)\tImag(Fww)\n";

    for (double k = 0.001; k = klim; k += delta_k)
    {
        complex<double> Fww = fww_estratificado(k, w, g1, g2, rho1, poisson1,
rho2, poisson2, A, B, d);
        output_file << setw(10) << k << "\t" << setw(10) << setprecision(6) <<
Fww.real() << "\t" << setw(10) << setprecision(6) << Fww.imag() << "\n";
    }
}

/**/

```

I.XIV. **inverte_matriz.h**

```

#ifndef INVERTE_MATRIZ_H
#define INVERTE_MATRIZ_H

#include <complex>
#include <vector>
#include <Eigen/Dense>

using namespace std;
using namespace Eigen;

template<int N>
bool inverse(complex<double> input[N][N], complex<double> output[N][N]);

bool inverseUsingEigen(const MatrixXcd& S_matrix, MatrixXcd& S_inv);

#endif // INVERTE_MATRIZ_H

```

I.XV. **inverte_matriz.cpp**

```

#include <iostream>
#include <complex>
#include <cmath>
#include <Eigen/Dense>

#include "inverte_matriz.h"

using namespace std;
using namespace Eigen;

template<int N>
bool inverse(complex<double> input[N][N], complex<double> output[N][N])
{

```



```

complex<double> temp[N][N * 2];

// Criar uma matriz aumentada [entrada|I]
for (int i = 0; i < N; ++i)
{
    for (int j = 0; j < N; ++j)
    {
        temp[i][j] = input[i][j];
        temp[i][j + N] = (i == j) ? 1.0 : 0.0;
    }
}

// Realizar a eliminação Gauss-Jordan
for (int i = 0; i < N; ++i)
{
    // Encontrar a linha com o elemento máximo na coluna atual
    int maxRow = i;
    for (int k = i + 1; k < N; ++k)
    {
        if (abs(temp[k][i]) > abs(temp[maxRow][i]))
        {
            maxRow = k;
        }
    }

    // Trocar a linha máxima com a linha atual
    for (int k = 0; k < N * 2; ++k)
    {
        swap(temp[maxRow][k], temp[i][k]);
    }

    // Verificar se a matriz é singular
    if (abs(temp[i][i]) == 0.0)
    {
        return false;
    }

    // Normalizar a linha atual
    complex<double> diagElement = temp[i][i];
    for (int k = 0; k < N * 2; ++k)
    {
        temp[i][k] /= diagElement;
    }

    // Eliminar a coluna atual em todas as outras linhas
    for (int k = 0; k < N; ++k)
    {
        if (k != i)
        {
            complex<double> factor = temp[k][i];
            for (int j = 0; j < N * 2; ++j)
            {
                temp[k][j] -= factor * temp[i][j];
            }
        }
    }
}

// Extrair a matriz inversa da matriz aumentada
for (int i = 0; i < N; ++i)
{
    for (int j = 0; j < N; ++j)

```

```

        {
            output[i][j] = temp[i][j + N];
        }
    }

    return true;
}

// Instanciar de maneira específica para tamanhos específicos
template<bool> inverse<4>(complex<double> input[4][4], complex<double>
output[4][4]);

// Função para calcular a inversa da matriz S usando Eigen
bool inverseUsingEigen(const MatrixXcd& S_matrix, MatrixXcd& S_inv) {
    // Calcula a inversa usando Eigen
    S_inv = S_matrix.inverse();

    // Verifica se a inversa foi calculada com sucesso
    return !S_inv.hasNaN();
}

```

I.XVI. integracao.h

```

//Edição 1: Incluído a função para cálculo da integral de Gauss Adaptiva

#ifndef INTEGRACAO_H
#define INTEGRACAO_H

#include <complex>
#include <functional>
#include <vector>

using namespace std;

// Função a ser integrada
complex<double> integrand(double k, double a,
    double r, double w, complex<double> g1, complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2,
    double A, double B, double d);

// Função para calcular a integral usando o método de Simpson
complex<double> simpson(double a, double b, int n,
    function<complex<double>(double)> func);

// Função para calcular a integral usando o método dos trapézios
complex<double> regra_trapezio(double a, double b, int n,
    function<complex<double>(double)> func);

// Função para calcular a integral por Quadratura Gaussiana
complex<double> fIntQuadGauss(double a, double b, int n,
    function<complex<double>(double)> integrando, int NPG = 2);
// Função para calcular a integral adaptiva a partir da Quadratura Gaussiana
complex<double> fIntQuadAdapt(double a, double b, int n,
    function<complex<double>(double)> integrando, int NPG = 2, int NumItMax = 8,
    double TOL = 1e-3, int N0 = 64);

#endif // INTEGRACAO_H

```

I.XVII. integracao.cpp

```
#include <iostream>
#include <complex>
#include <cmath>
#include <boost/math/special_functions/bessel.hpp>
#include <functional>
#include <cstdlib>

#include "integracao.h"
#include "fww.h"

using namespace std;
using namespace boost::math;

// Função a ser integrada
complex<double> integrand(double k, double a,
    double r, double w, complex<double> g1, complex<double> g2,
    double rho1, double poisson1, double rho2, double poisson2,
    double A, double B, double d)
{
    double J1_ka = cyl_bessel_j(1, k * a);
    double J0_kr = cyl_bessel_j(0, k * r);
    complex<double> FWW;

    // Verifica se g1 é igual a g2
    if (g1 == g2)
    {
        // Chama a função para o solo homogêneo
        FWW = fww_homogeneo(k, w, g1, rho1, poisson1);
    }
    else
    {
        // Chama a função para o solo estratificado
        FWW = fww_estratificado(k, w, g1, g2, rho1, poisson1, rho2, poisson2, A,
B, d);
    }

    return a * J1_ka * FWW * J0_kr;
}

// As funções de integração aqui não foram utilizadas na versão final do código,
mas foram testadas. Elas não conseguem captar com precisão os picos do
integrando.

// Função para calcular a integral usando o método de Simpson
complex<double> simpson(double a, double b, int n,
    function<complex<double>(double)> func)
{
    if (n % 2 != 0)
    {
        cerr << "O número de subintervalos n deve ser par!" << "\n";
        exit(EXIT_FAILURE);
    }

    double h = abs(b - a) / static_cast<double>(n);
    complex<double> sum = func(a) + func(b);

    for (int i = 1; i < n; i++)
    {
        double x = a + static_cast<double>(i) * h;
        if (i % 2 == 0)
```

```

        {
            sum += 2.0 * func(x);
        } else {
            sum += 4.0 * func(x);
        }
    }
    return (h / 3.0) * sum;
}

// Função para calcular a integral usando o método dos trapézios
complex<double> regra_trapezio(double a, double b, int n,
function<complex<double>(double)> func) {
    double h = (b - a) / static_cast<double>(n);
    complex<double> sum = (func(a) + func(b)) / 2.0;

    for (int i = 1; i < n; i++) {
        double x = a + static_cast<double>(i) * h;
        sum += func(x);
    }

    return h * sum;
}

// Integração de intervalo [a, b], usando discretização uniforme com n
// elementos de tamanho deltaK, usando NPG pontos de Gauss, para posição r.
complex<double> fIntQuadGauss(double a, double b, int n,
function<complex<double>(double)> integrando, int NPG)
{
    // Tamanho dos elementos 1D
    double deltaK = (b - a) / n;
    complex<double> Integral = 0.0;

    if (NPG == 2)
    {
        // Pontos de Gauss
        double Qsi1 = -1.0 / sqrt(3.0);
        double Qsi2 = 1.0 / sqrt(3.0);

        // Conversão para coordenada global
        double K1 = (0.5 * deltaK) * Qsi1 + (a + 0.5 * deltaK);
        double K2 = (0.5 * deltaK) * Qsi2 + (a + 0.5 * deltaK);

        // Contribuição 1º elemento
        Integral = (integrando(K1) + integrando(K2)) * 0.5 * deltaK;

        // Contribuição demais elementos
        for (int i = 1; i < n; i++)
        {
            K1 += deltaK;
            K2 += deltaK;
            Integral += (integrando(K1) + integrando(K2)) * 0.5 * deltaK;
        }
    }
    else if (NPG == 3)
    {
        // Pontos de Gauss para 3 pontos
        double Qsi1 = -sqrt(3.0 / 5.0);
        double Qsi2 = 0.0;
        double Qsi3 = sqrt(3.0 / 5.0);

        // Pesos de Gauss para 3 pontos
        double w1 = 5.0 / 9.0;

```

```

double w2 = 8.0 / 9.0;
double w3 = 5.0 / 9.0;

// Conversão para coordenada global
double K1 = (0.5 * deltaK) * Qsi1 + (a + 0.5 * deltaK);
double K2 = (0.5 * deltaK) * Qsi2 + (a + 0.5 * deltaK);
double K3 = (0.5 * deltaK) * Qsi3 + (a + 0.5 * deltaK);

// Contribuição 1º elemento
Integral = (integrando(K1) * w1 + integrando(K2) * w2 + integrando(K3) *
w3) * 0.5 * deltaK;

// Contribuição demais elementos
for (int i = 1; i < n; i++)
{
    K1 += deltaK;
    K2 += deltaK;
    K3 += deltaK;
    Integral += (integrando(K1) * w1 + integrando(K2) * w2 +
integrando(K3) * w3) * 0.5 * deltaK;
}
}
else if (NPG == 4)
{
    // Pontos de Gauss para 4 pontos
    double Qsi1 = -sqrt(0.8611);
    double Qsi2 = -sqrt(0.3398);
    double Qsi3 = sqrt(0.3398);
    double Qsi4 = sqrt(0.8611);

    // Pesos de Gauss para 4 pontos
    double w1 = 0.3478;
    double w2 = 0.6521;
    double w3 = 0.6521;
    double w4 = 0.3478;

    // Conversão para coordenada global
    double K1 = (0.5 * deltaK) * Qsi1 + (a + 0.5 * deltaK);
    double K2 = (0.5 * deltaK) * Qsi2 + (a + 0.5 * deltaK);
    double K3 = (0.5 * deltaK) * Qsi3 + (a + 0.5 * deltaK);
    double K4 = (0.5 * deltaK) * Qsi4 + (a + 0.5 * deltaK);

    // Contribuição 1º elemento
    Integral = (integrando(K1) * w1 + integrando(K2) * w2 + integrando(K3) *
w3 + integrando(K4) * w4) * 0.5 * deltaK;

    // Contribuição demais elementos
    for (int i = 1; i < n; i++)
    {
        K1 += deltaK;
        K2 += deltaK;
        K3 += deltaK;
        K4 += deltaK;
        Integral += (integrando(K1) * w1 + integrando(K2) * w2 +
integrando(K3) * w3 + integrando(K4) * w4) * 0.5 * deltaK;
    }
}

return Integral;
}

// Método adaptivo usando a quadratura gaussiana

```

```

complex<double> fIntQuadAdapt(double a, double b, int n,
function<complex<double>(double)> integrando, int NPG, int NumItMax, double TOL,
int N0)
{
    // Contador do número de iterações
    int NumIt = 4;
    // Quantidade de elementos da malha considerada
    int N = N0;
    // Integral das últimas 4 iterações
    vector<complex<double>> Integral = { 0.0, 0.0, 0.0, 0.0 };
    // Diferença relativa das últimas 3 iterações
    vector<double> DifRel = { 0.0, 0.0, 0.0 };
    // Integral das primeiras 4 iterações:
    // 1ª iteração
    Integral[0] = fIntQuadGauss(a, b, N, integrando, NPG);
    // 2ª - 4ª iterações
    for (int i = 1; i < 4; i++)
    {
        // Refinar malha
        N = 2 * N;
        Integral[i] = fIntQuadGauss(a, b, N, integrando, NPG);
        DifRel[i - 1] = abs((Integral[i] - Integral[i - 1]) / Integral[i - 1]);
    }
    // Núcleo do método adaptativo
    while ((NumIt < NumItMax) && (DifRel[0] > TOL || DifRel[1] > TOL || DifRel[2]
> TOL))
    {
        // Atualizar vetores Integral e DifRel nas posições (0, 1, 2) e (0, 1)
        // respectivamente
        Integral[0] = Integral[1];
        Integral[1] = Integral[2];
        Integral[2] = Integral[3];
        DifRel[0] = DifRel[1];
        DifRel[1] = DifRel[2];
        // Refinar malha
        N = 5 * N;
        // Calcular Integral e DifRel da iteração atual
        Integral[3] = fIntQuadGauss(a, b, N, integrando, NPG);
        DifRel[2] = abs((Integral[3] - Integral[2]) / Integral[2]);
        // Atualizar contador de iterações
        NumIt++;
    }
    // Mensagem de êxito ou falha
    if (NumIt >= NumItMax)
    {
        cout << "Método de integração adaptativo não convergiu.\n";
    }
    else
    {
        cout << "Método de integração adaptativo convergiu.\n";
    }
    // Resultado da integração adaptativa
    return Integral[3];
}

```

LXVIII. w(r).h

```

#ifndef W_R_H
#define W_R_H

#include <complex>

```

```

#include <boost/math/quadrature/gauss_kronrod.hpp>
#include <boost/math/quadrature/tanh_sinh.hpp>
#include <complex>
#include <iostream>
#include <functional>

using namespace std;

complex<double> calcular_W_r(double r, double a, double w, const complex<double>&
g1, const complex<double>& g2, double rho1, double poisson1, double rho2, double
poisson2, double A, double B, double d, double lim_inf, double lim_sup, int n);

#endif // W_R_H

```

I.XIX. w(r).cpp

```

#include "w(r).h"
#include "integracao.h"
#include "funcoes.h"

using namespace std;
using namespace boost::math::quadrature;

/**
complex<double> calcular_W_r(double r, double a, double w, const complex<double>&
g1, const complex<double>& g2,
double rho1, double poisson1, double rho2, double poisson2,
double A, double B, double d, double lim_inf, double lim_sup, int n)
{
    // Função lambda para o integrand
    auto integrand_real = [&](double k) -> double
    {
        return integrand(k, a, r, w, g1, g2, rho1, poisson1, rho2, poisson2, A,
B, d).real();
    };

    auto integrand_imag = [&](double k) -> double
    {
        return integrand(k, a, r, w, g1, g2, rho1, poisson1, rho2, poisson2, A,
B, d).imag();
    };

    gauss_kronrod<double, 15> integrator;
    double error_real = 0.0, error_imag = 0.0;

    // Divisão do intervalo de integração
    int num_subintervals = 70;
    double subinterval_size = (lim_sup - lim_inf) / num_subintervals;
    double result_real = 0.0, result_imag = 0.0;

    for (int i = 0; i < num_subintervals; ++i)
    {
        double local_lim_inf = lim_inf + i * subinterval_size;
        double local_lim_sup = local_lim_inf + subinterval_size;

        double local_result_real = integrator.integrate(integrand_real,
local_lim_inf, local_lim_sup, 15, 1e-3, &error_real);
        double local_result_imag = integrator.integrate(integrand_imag,
local_lim_inf, local_lim_sup, 15, 1e-3, &error_imag);

        result_real += local_result_real;
    }
}

```

```

        result_imag += local_result_imag;
    }

    return complex<double>(result_real, result_imag);
}
/**/

```

I.XX. s_z.h

```

#ifndef S_Z_H
#define S_Z_H

#include <complex>
#include <boost/math/quadrature/gauss_kronrod.hpp>
#include <boost/math/quadrature/tanh_sinh.hpp>
#include <complex>
#include <cmath>
#include <functional>
#include <iostream>

using namespace std;

complex<double> calcular_Sz(double a, double w, const complex<double>& g1, const
complex<double>& g2, double rho1, double poisson1, double rho2, double poisson2,
double A, double B, double d, double lim_inf, double lim_sup, int n);

#endif // S_Z_H

```

I.XXI. s_z.cpp

```

#define _USE_MATH_DEFINES

#include "s_z.h"
#include "integracao.h"
#include "funcoes.h"

using namespace std;
using namespace boost::math::quadrature;

/**
complex<double> calcular_Sz(double a, double w, const complex<double>& g1, const
complex<double>& g2,
double rho1, double poisson1, double rho2, double poisson2,
double A, double B, double d, double lim_inf, double lim_sup, int n)
{
    // Função lambda para o integrando
    auto integrand_w_r = [&](double r) -> complex<double>
    {
        auto f_lambda_real = [&](double k) -> double
        {
            return integrand(k, a, r, w, g1, g2, rho1, poisson1, rho2, poisson2,
A, B, d).real();
        };

        auto f_lambda_imag = [&](double k) -> double
        {
            return integrand(k, a, r, w, g1, g2, rho1, poisson1, rho2, poisson2,
A, B, d).imag();
        };
    };
}

```



```

    gauss_kronrod<double, 15> integrator;
    double error_real = 0.0, error_imag = 0.0;

    int num_subintervals = 70;
    double subinterval_size = (lim_sup - lim_inf) / num_subintervals;
    double result_real = 0.0, result_imag = 0.0;

    for (int i = 0; i < num_subintervals; ++i)
    {
        double local_lim_inf = lim_inf + i * subinterval_size;
        double local_lim_sup = local_lim_inf + subinterval_size;

        double local_result_real = integrator.integrate(f_lambda_real,
local_lim_inf, local_lim_sup, 15, 1e-3, &error_real);
        double local_result_imag = integrator.integrate(f_lambda_imag,
local_lim_inf, local_lim_sup, 15, 1e-3, &error_imag);

        result_real += local_result_real;
        result_imag += local_result_imag;
    }

    return complex<double>(result_real, result_imag) * r;
};

gauss_kronrod<double, 15> integrator;
double error_real_wr = 0.0, error_imag_wr = 0.0;

int num_subintervals_r = 70;
double subinterval_size_r = a / num_subintervals_r;
double result_real_wr = 0.0, result_imag_wr = 0.0;

for (int i = 0; i < num_subintervals_r; ++i)
{
    double local_lim_inf_r = i * subinterval_size_r;
    double local_lim_sup_r = local_lim_inf_r + subinterval_size_r;

    double local_result_real_wr = integrator.integrate([&(double r) { return
integrand_w_r(r).real(); }, local_lim_inf_r, local_lim_sup_r, 15, 1e-3,
&error_real_wr);
    double local_result_imag_wr = integrator.integrate([&(double r) { return
integrand_w_r(r).imag(); }, local_lim_inf_r, local_lim_sup_r, 15, 1e-3,
&error_imag_wr);

    result_real_wr += local_result_real_wr;
    result_imag_wr += local_result_imag_wr;
}

complex<double> integral_w_r(result_real_wr, result_imag_wr);

return (M_PI * pow(a, 4)) / (2.0 * integral_w_r);
}
//*/

```

I.XXII. pontos_malha.h

```

#ifndef PONTOS_MALHA_H
#define PONTOS_MALHA_H

#include <iostream>
#include <vector>
#include <algorithm>

```

```

#include <iomanip>
#include <sstream>

using namespace std;

struct Point {
    double x;
    double y;
};

template<typename T>
string to_string(const T& value)
{
    ostringstream oss;
    oss << value;
    return oss.str();
}

vector<Point> calcula_pontos(double A, double B, int x_div, int y_div);

void print_pontos(const vector<Point>& centers);

vector<vector<double>> calcula_distancias(const vector<Point>& centers);

void print_distancias(const vector<vector<double>>& distances);

#endif // PONTOS_MALHA_H

```

I.XXIII. pontos_malha.cpp

```

#include "pontos_malha.h"

using namespace std;

vector<Point> calcula_pontos(double A, double B, int x_div, int y_div)
{
    vector<Point> centers;

    double largura = A / x_div;
    double altura = B / y_div;

    for (int j = 0; j < y_div; ++j) {
        for (int i = 0; i < x_div; ++i) {
            double centerX = (i + 0.5) * largura;
            double centerY = (j + 0.5) * altura;
            centers.push_back({ centerX, centerY });
        }
    }

    // Ordenar os pontos centrais em ordem crescente primeiro em y e depois em x
    sort(centers.begin(), centers.end(), [](const Point& a, const Point& b) {
        if (a.y != b.y) return a.y < b.y;
        return a.x < b.x;
    });

    return centers;
}

void print_pontos(const vector<Point>& centers)
{
    for (const auto& center : centers) {
        cout << "p = (" << center.x << ", " << center.y << ")\n";
    }
}

```

```

    }
}

vector<vector<double>> calcula_distancias(const vector<Point>& centers)
{
    vector<vector<double>> distances;
    distances.resize(centers.size(), vector<double>(centers.size(), 0.0));

    for (size_t i = 0; i < centers.size(); ++i) {
        for (size_t j = i + 1; j < centers.size(); ++j) {
            double distance = sqrt(pow(centers[j].x - centers[i].x, 2) +
pow(centers[j].y - centers[i].y, 2));
            distances[i][j] = distances[j][i] = distance;
        }
    }

    return distances;
}

void print_distancias(const vector<vector<double>>& distances)
{
    // Encontra o maior número de dígitos para alinhar corretamente as colunas
    size_t max_largura = 0;
    for (size_t i = 0; i < distances.size(); ++i) {
        for (size_t j = 0; j < distances[i].size(); ++j) {
            size_t largura = to_string(distances[i][j]).size();
            if (largura > max_largura) {
                max_largura = largura;
            }
        }
    }

    // Imprime a matriz com alinhamento
    for (size_t i = 0; i < distances.size(); ++i) {
        for (size_t j = 0; j < distances[i].size(); ++j) {
            cout << setw(max_largura) << fixed << setprecision(2) <<
distances[i][j] << " ";
        }
        cout << "\n";
    }
}

```

I.XXIV. matriz_flexibilidade.h

```

#ifndef MATRIZ_FLEXIBILIDADE_H
#define MATRIZ_FLEXIBILIDADE_H

#include <iostream>
#include <vector>
#include <complex>
#include <Eigen/Dense>
#include <iostream>
#include <iomanip>
#include <cmath>
#include <set> // Para utilizar o std::set

#include "pontos_malha.h"
#include "w(r).h"

using namespace std;

```

```

using namespace Eigen;

vector<vector<complex<double>>> calcular_matriz_fv(double raio, double w,
complex<double> g1, complex<double> g2, double rho1, double poisson1, double
rho2, double poisson2, double A, double B, double d, double lim_inf, double
lim_sup, int n, double k0, double c0, double x_div, double y_div);

MatrixXcd inverter_matriz(const vector<vector<complex<double>>>& fv);

void imprimir_matriz_complexa_vector(const vector<vector<complex<double>>>&
matriz, const string& nome);

void imprimir_matriz_complexa_matrixxd(const MatrixXcd& matriz, const string&
nome);

void calcular_rigidez_amortecimento(const MatrixXcd& fv_inversa, MatrixXd& K,
MatrixXd& C, VectorXd& K_soma, VectorXd& C_soma, double& K_calculado, double&
C_calculado);

#endif // MATRIZ_FLEXIBILIDADE_H

```

I.XXV. matriz_flexibilidade.cpp

```

#define _USE_MATH_DEFINES // Para usar constantes matemáticas como Pi

#include "matriz_flexibilidade.h"
#include "pontos_malha.h"
#include "w(r).h"

using namespace std;
using namespace Eigen;

/**
// Função para calcular a matriz fv
vector<vector<complex<double>>> calcular_matriz_fv(double raio, double w,
complex<double> g1, complex<double> g2, double rho1, double poisson1, double
rho2, double poisson2, double A, double B, double d, double lim_inf, double
lim_sup, int n, double k0, double c0, double x_div, double y_div)
{
    // Calcular matriz de distâncias
    vector<Point> centers = calcula_pontos(A, B, x_div, y_div);
    size_t size = centers.size();
    vector<vector<double>> distancias = calcula_distancias(centers);

    // Criar um conjunto para armazenar valores únicos de r
    set<double> valores_unicos_r;
    for (size_t i = 0; i < size; ++i) {
        for (size_t j = 0; j < size; ++j) {
            if (i != j) {
                valores_unicos_r.insert(distancias[i][j]);
            }
        }
    }

    // Calcular valores de W(r) para cada valor único de r
    vector<complex<double>> valores_W_r;
    for (const double& r : valores_unicos_r) {
        valores_W_r.push_back(calcular_W_r(r, raio, w, g1, g2, rho1, poisson1,
rho2, poisson2, A, B, d, lim_inf, lim_sup, n));
    }
}

```

```

        // Mapear r -> W(r)
        vector<vector<complex<double>>> fv(size, vector<complex<double>>(size, { 0, 0
    }));
    for (size_t i = 0; i < size; ++i) {
        for (size_t j = 0; j < size; ++j) {
            if (i != j) {
                double r = distancias[i][j];
                // Encontrar o valor de W(r) correspondente
                auto it = find(valores_unicos_r.begin(), valores_unicos_r.end(),
r);
                if (it != valores_unicos_r.end()) {
                    size_t index = distance(valores_unicos_r.begin(), it);
                    fv[i][j] = valores_W_r[index] / (M_PI * pow(raio, 2));
                }
            }
            else {
                fv[i][j] = 1.0 / complex<double>(k0, w * c0);
            }
        }
    }

    return fv;
}
/**/

MatrixXcd inverter_matriz(const vector<vector<complex<double>>>& fv)
{
    // Calcular tamanho da matriz
    size_t size = fv.size();

    // Calcular a inversa da matriz fv
    MatrixXcd fv_matrix(size, size); // Criar uma matriz Eigen de números
complexos
    for (size_t i = 0; i < size; ++i) {
        for (size_t j = 0; j < size; ++j) {
            fv_matrix(i, j) = complex<double>(fv[i][j].real(), fv[i][j].imag());
        }
    }
    // Preencher a matriz Eigen com os valores de fv
    return fv_matrix.inverse();
}

void imprimir_matriz_complexa_vector(const vector<vector<complex<double>>>&
matriz, const string& nome)
{
    cout << "\n" << nome << ":\n";
    for (size_t i = 0; i < matriz.size(); ++i) {
        for (size_t j = 0; j < matriz[i].size(); ++j) {
            cout << setw(20) << setprecision(7) << matriz[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "\n";
}

void imprimir_matriz_complexa_matrixxd(const MatrixXcd& matriz, const string&
nome)
{
    cout << "\n" << nome << ":\n";
    for (int i = 0; i < matriz.rows(); ++i) {
        for (int j = 0; j < matriz.cols(); ++j) {
            cout << setw(20) << setprecision(7) << matriz(i, j) << " ";
        }
    }
}

```

```

    }
    cout << "\n";
    cout << "\n";
}
}

void calcular_rigidez_amortecimento(const MatrixXcd& fv_inversa, MatrixXd& K,
MatrixXd& C, VectorXd& K_soma, VectorXd& C_soma, double& K_calculado, double&
C_calculado)
{
    // Calcular a matriz K (parte real) e C (parte imaginária)
    K = fv_inversa.real();
    C = fv_inversa.imag();

    // Calcular a soma das linhas de K e C
    K_soma = K.rowwise().sum();
    C_soma = C.rowwise().sum();

    // Calcular a rigidez e o amortecimento totais
    K_calculado = K.sum();
    C_calculado = C.sum();
}

```

I.XXVI. coeficientes_adimensionais.h

```

#ifndef COEFICIENTES_ADIMENSIONAIS_H
#define COEFICIENTES_ADIMENSIONAIS_H

#include <iostream>
#include <cmath>

#include "coeficientes_adimensionais.h"

using namespace std;

// Função para calcular K_teorico
double calcular_K_teorico(double g01, double poisson1, double a, double b);

// Função para calcular C_teorico
double calcular_C_teorico(double r_eq, double poisson1, double rho1, double g01);

// Função para calcular k
double calcular_k(double K_calculado, double K_teorico);

// Função para calcular c
double calcular_c(double C_calculado, double C_teorico, double w);

#endif // !COEFICIENTES_ADIMENSIONAIS_H

```

I.XXVII. coeficientes_adimensionais.cpp

```

#include <iostream>
#include <cmath>

#include "coeficientes_adimensionais.h"

using namespace std;

```

```

// Função para calcular K_teórico
double calcular_K_teorico(double g01, double poisson1, double a, double b)
{
    return g01 * b / (1 - poisson1) * (3.1 * pow(a / b, 0.75) + 1.6);
}

// Função para calcular C_teórico
double calcular_C_teorico(double r_eq, double poisson1, double rho1, double g01)
{
    return (4 * pow(r_eq, 2)) / (1 - poisson1) * sqrt(rho1 * g01);
}

// Função para calcular k
double calcular_k(double K_calculado, double K_teorico)
{
    return K_calculado / K_teorico;
}

// Função para calcular c
double calcular_c(double C_calculado, double C_teorico, double w)
{
    return C_calculado / (w * C_teorico);
}

```

I.XXVIII. amplificacao_dinamica.h

```

#ifndef AMPLIFICACAO_DINAMICA_H
#define AMPLIFICACAO_DINAMICA_H

#include <iostream>
#include <complex>

double amplificacao_dinamica(double a0, double k, double c, double m);

#endif // !AMPLIFICACAO_DINAMICA_H

```

I.XXIX. amplificacao_dinamica.cpp

```

#define _USE_MATH_DEFINES // Para usar constantes matemáticas como Pi
#include "amplificacao_dinamica.h"

using namespace std;

double amplificacao_dinamica(double a0, double k, double c, double m)
{
    complex<double> numerador(1.0, 0.0); // Numerador é 1
    complex<double> denominador(1.0 - (pow(a0, 2) * m / M_PI * k), (pow(a0, 3) *
m / M_PI * c)); // Denominador complexo
    complex<double> resultado = numerador / denominador; // Divisão complexa
    return abs(resultado); // Retorna o valor absoluto (módulo)
}

```