Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia Aeroespacial

Caracterização de incertezas no comportamento dinâmico de mangueiras pressurizadas para aplicações aeroespaciais

Autor: Nivaldo Pereira Lopo Junior
Orientador: Prof. Dr. Sergio Henrique da Silva Carneiro

Brasília, DF
2017

Nivaldo Pereira Lopo Junior

Caracterização de incertezas no comportamento dinâmico de mangueiras pressurizadas para aplicações aeroespaciais

Monografia submetida ao curso de graduação em Engenharia Aeroespacial da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Aeroespacial.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Sergio Henrique da Silva Carneiro

Brasília, DF

2017

Nivaldo Pereira Lopo Junior
Caracterização de incertezas no comportamento dinâmico de mangueiras pressurizadas para aplicações aeroespaciais/ Nivaldo Pereira Lopo Junior. – Brasília, DF, 2017-
121 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Sergio Henrique da Silva Carneiro

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2017.

1. Mangueira Flexível. 2. Vibração. I. Prof. Dr. Sergio Henrique da Silva Carneiro. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Caracterização de incertezas no comportamento dinâmico de mangueiras pressurizadas para aplicações aeroespaciais

CDU 02:141:005.6

Nivaldo Pereira Lopo Junior

Caracterização de incertezas no comportamento dinâmico de mangueiras pressurizadas para aplicações aeroespaciais

Monografia submetida ao curso de graduação em Engenharia Aeroespacial da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Aeroespacial.

Trabalho aprovado. Brasília, DF, 08 de Dezembro de 2017:

Prof. Dr. Sergio Henrique da Silva Carneiro
Orientador

Prof. Dr. Artem Andrianov
Convidado 1

Prof. Dr. Cristian Vendittozzi
Convidado 2

Brasília, DF
2017

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais, Maria do Carmo e Nivaldo Lopo, por tornar todas as conquistas em minha vida possíveis, em especial minha graduação em Engenharia Aeroespacial, me dando todo o suporte necessário com muito amor e dedicação.

Agradeço também à minha fiel companheira Camila Cardador por estar ao meu lado e nos momentos mais dificeis me motivar e me manter focado nessa reta final da gradução, além de me confortar com os melhores abraços.

A todo o corpo docente do curso que me apresentou com maestria o mundo da engenharia aeroespacial me inspirando a seguir esta carreira com paixão e vontade. Ao Professor Sergio Carneiro pela paciência, ajuda e compreensão, pelas conversas e orientações durante a realização desta monografia e aos professores Artem Andrianov, Chantal Cappelletti, Artur Bertoldi, Jungpyo Lee, Olexiy Shynkarenko, Manuel Barcelos, Paolo Gessini, Domenico Simone, Sébastien Rondineau e Cristian Vendittozzi por me ajudarem e compartilharem seus conhecimentos me permitindo também aplicar todo seu ensinamento.

Obrigado aos meus amigos Lui Txai, Luan Henrique, Hermes William, Nícollas Diniz, Beatriz Liberino, Luan Guimarães, Felipe Duerno, Igor Kinoshita, Matheus Cabral, Lucas Germano, José Eduardo e Reges Matheus por todas as conversas e momentos que tenho certeza terem sido essenciais para minha vontade e persistência em perseguir e não desistir deste sonho.

“Isto não é voar, isto é cair com estilo”

Toy Story

Resumo

Uma particularidade de sistemas propulsivos com propelentes líquidos e/ou gasosos é a possível existência de mangueiras flexíveis, estas presentes em locais onde será necessária a movimentação de partes estruturais do foguete. A geometria interna das mangueiras flexíveis é extremamente favorável a possíveis curvas ou pequenos deslocamentos das mesmas, porém tem como grande contrapartida a possível existência de vibrações induzidas pelo escoamento interno. O presente trabalho apresenta um estudo do efeito de incertezas nos parâmetros construtivos e operacionais sobre o comportamento dinâmico das mangueiras flexíveis. A análise é feita a partir da aplicação de simulações de Monte Carlo sobre um método encontrado na literatura que descreve o fenômeno vibratório das mangueiras flexíveis. O método é implementado em um algoritmo computacional e é feita primeiramente uma análise de sensibilidade dos parâmetros utilizados seguida da realização de casos de randomização dos parâmetros geométricos e operacionais da mangueira, tais casos mostram que mesmo operando nominalmente, fora do regime vibratório, mangueiras flexíveis estão suscetíveis ao acoplamento fluido-estrutural devido a incerteza dos parâmetros descritivos das mesmas.

Palavras-chaves: Mangueira Flexível. Vibração. Incerteza. Monte Carlo.

Abstract

A particularity of liquid and/or gaseous propulsive systems is the possible existence of flexible hoses, those present in places where the mobility of some of the rocket structures are needed. The internal geometry of flexible hoses is extremely favorable to possible curves and small displacements of themselves, but has a major counterpart: the possible existence of internal flow-induced vibrations. The present project presents a study about the uncertainty effects on constructive and operational parameters on the dynamic behavior of flexible hoses. The analysis is made from the application of Monte Carlo simulations over a method found in literature that describes the vibrating phenomenon of the flexible hoses. The method is implemented in a computational algorithm and a sensitivity analysis of the parameters used is performed, followed by cases of randomization of the geometric and operational parameters of the hose, such cases showed that even operating in nominal condition, out of the vibration regime, flexible hoses are susceptible to fluid-structural coupling duo to uncertainty of some of their parameters.

Key-words: Flexible Hose. Vibration. Uncertainty. Monte Carlo.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

EUA	Estados Unidos da América
NASA	*National Aeronautics and Space Administration*
MSFC	*Marshall Space Flight Center*
SwRI	*Southwest Research Institute*
TCC	Trabalho de Conclusão de Curso
FNCO	Primeiro número de frequência do modo acústico radial
CEE	Constante elástica específica
MMC	Método de Monte Carlo
FDP	Função de densidade de probabilidade

Lista de símbolos

σ	Comprimento da convolução
λ	Passo entre as convoluções
h	Altura das convoluções
t	Espessura das convoluções
δ	Espaço entre convoluções $= \lambda - \sigma$
a	Raio médio da convolução
S	Número de Strouhal
$S_{\sigma i}$	Limite inferior de Strouhal $= 0.1$
$S_{\sigma c}$	Valor crítico de Strouhal $= 0.2$
$S_{\sigma s}$	Limite superior de Strouhal $= 0.3$
f	Frequência
f_{EF}	Frequência para modo em-fase
f_{FF}	Frequência para modo fora-de-fase
f_{FL}	Frequência para o modo de flexão local
f_{acous}	Frequência primeiro modo ressonância acústica radial
V	Velocidade
V_e	Velocidade do escoamento
V_s	Velocidade do som no meio
V_{acous}	Velocidade primeiro modo ressonância acústica radial
V_I	Limite inferior de velocidade
V_C	Valor critico de velocidade
V_S	Limite superior de velocidade
l	Comprimento específico

k	Constante elástica de meia convolução
K_a	Constante elástica de uma convolução inteira
D_i	Diametro interno da mangueira
D_e	Diametro externo da mangueira
D_m	Diametro médio da mangueira $= (D_i + D_e)/2$
E	Módulo de Elasticidade
N_p	Número de camadas das convoluções
m_m	Massa do material de uma convolução
m_{EF}	Massa de fluido para modo em-fase
m_{FF}	Massa de fluido para modo fora-de-fase
ρ_f	Densidade do fluido
ρ_m	Densidade do material das convoluções
r_i	Raio iterno $= D_i/2$
T_{esc}	Tensão induzida pelo escoamento
$T_{esc}C$	Tensão induzida pelo escoamento corrigida
FC	Fator de correção
P_D	Pressão dinâmica
C_{NP}	Coeficiente de N_p modificador de amortecimento
C_c	Coeficiente de cotovelo amplificador de tensão

Sumário

1 Introdução

Este capítulo tem como objetivo uma apresentação inicial do presente trabalho, desde uma contextualização da problemática aqui envolvida até uma breve descrição dos objetivos desta monografia.

1.1 Contextualização

1.1.1 Mangueiras Flexíveis

Uma particularidade de sistemas propulsivos com propelentes líquidos e/ou gasosos é a possível existência de mangueiras flexíveis, estas presentes em locais onde será necessária a movimentação de partes estruturais do foguete. Durante a concepção de todo o conjunto propulsivo o projetista deve utilizar o menor numero possível de seções flexíveis e posicioná-las de maneira a ter-se a maior movimentação possível do sistema como um todo, com a menor movimentação possível das seções flexíveis, evitando alto níveis de tensão nos componentes (HUZEL; HUANG, 1992).

As mangueiras flexíveis estudadas nesse documento correspondem a uma sequencia de convoluções metálicas envolvidas por uma trança, também metálica, como mostram as Figs. 1 e 2, de forma que o movimento do topo das convoluções é limitado pela trança.

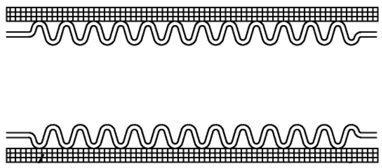

Figura 1 – Esquema da seção transversal da mangueira[1].

Os parâmetros que descrevem as convoluções são: comprimento da convolução, σ, passo entre as convoluções, λ, altura das convoluções, h e a espessura das convoluções, t, como mostra a Fig. 3. Somados a esses parâmetros, tem-se também os diâmetros interno e externo da mangueira, levando em consideração apenas as convoluções, sem a trança externa. As mangueiras podem, também, possuir uma ou mais camadas de convoluções.

[1] Adaptada de Cap, Urquidi e Canzona (2013).

Figura 2 – Mangueira flexível[2].

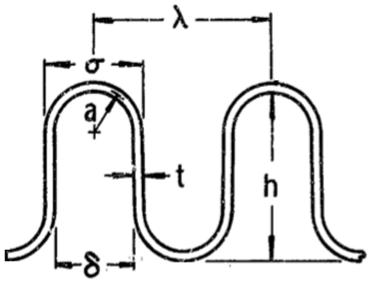

Figura 3 – Parâmetros das convoluções[3].

1.1.2 Vibração induzida pelo escoamento em mangueiras flexíveis

A geometria interna das mangueiras flexíveis é extremamente favorável a possíveis curvas ou pequenos deslocamentos das mesmas, porém tem como grande contrapartida a possível existência de vibrações induzidas pelo escoamento interno nas convoluções da mangueira. Esse tipo de vibração pode levar a falhas por fadiga, forçando desligamento prematuro de sistemas críticos de fluidos (GERLACH, 1969). A importância desse tópico foi descoberta em investigações da causa de falha de uma mangueira flexível no voo AS-502 do foguete *Saturn* (BASS III et al., 1970), onde acredita-se que esse tipo de vibração em seções da linha de combustível causou o mal funcionamento dos motores foguete *Rocketdyne J-2* (DOUGHERTY JR; RAFFERTY, 1968).

Os métodos mais comuns para corrigir aplicações não sucedidas de mangueiras flexíveis são tanto o uso de um forro interno, quando possível, ou o uso de convoluções com camada mais espessa ou mais de uma camada. Este último método, no entanto, nem sempre previne a ocorrência de vibrações induzidas pelo escoamento e a inserção de um forro interno geralmente leva ao acréscimo do peso e custo dos componentes. É desejável, portanto, do ponto de vista do projetista, o uso de um procedimento analítico que permita a predição de zonas de escoamento criticas para certa configuração de convoluções em

[2] Disponivel em: <https://goo.gl/images/aaJ1Q7>.
[3] Adaptado de Gerlach e Schroeder (1969).

uma mangueira flexível, possibilitando, também, o posterior cálculo do nível de tensão resultante da excitação fluídica em tais zonas criticas (GERLACH, 1969).

Dada tamanha importância do bom funcionamento das mangueiras flexíveis em aplicações aeroespaciais, onde a falha do sistema de propulsão pode levar a perca total da missão, ou ainda mais grave em missões tripuladas, ao óbito, foi iniciado pelo *Southwest Research Institute* (SwRI) nos EUA um estudo do fenômeno de vibração das convoluções de linhas flexíveis, os primeiros resultados desse estudo são descritos por Gerlach e Schroeder (1969).

A analise foi realizada por meio da experimentação de diversas amostras de convoluções, com e sem a trança externa, onde cada espécime possuía um medidor de tensão conectado ao topo da convolução. Foi então observado que a causa dos problemas relacionados a vibração induzida pelo escoamento provém de uma instabilidade fluido-elástica, com o acoplamento do derramamento de vórtice nos topos das convoluções com os modos de vibração naturais da estrutura (GERLACH; SCHROEDER, 1969).

Com o interesse, e financiamento, da agencia espacial norte-americana (NASA) essa pesquisa foi mantida com aprimoramento dos métodos para cálculo das zonas criticas de excitação de linhas flexíveis, além de melhoras no método empírico para calculo dos níveis de tensão exercidos nas convoluções, tais estudos são descritos por Johnson et al. (1979) e Tygielski, Smyly e Gerlach (1983). Com todo o conhecimento levantado por meio de tais pesquisas, o *Marshall Space Flight Center* (MSFC) desenvolveu um documento descritivo para a avaliação de linhas flexíveis com vibrações induzidas pelo escoamento (MSFC, 1992), tal documento é utilizado pela NASA em aplicações atuais, como em NASA (2016).

Nos estudos do fenômeno de vibração de linhas flexíveis, realizados por Gerlach e Schroeder (1969), Bass III et al. (1970), Johnson et al. (1979), Tygielski, Smyly e Gerlach (1983) foram analisadas mangueiras flexíveis e foles metálicos, que correspondem a convoluções metálicas sem a presença da trança externa. Nesse trabalho, porém, foi realizada a análise de incertezas somente em mangueiras flexíveis.

1.1.3 Incerteza dos parâmetros utilizados

Para a avaliação dos modos de vibração das mangueiras flexíveis, MSFC (1992) utiliza das dimensões físicas que descrevem as convoluções, além de características especificas tanto do material da mangueira como do fluído. Tais parâmetros possuem uma incerteza em seus valores, esta incerteza provém, no caso das dimensões, do método de fabricação das mangueiras, já em relação as características do fluído podem haver variações de pressão e temperatura ao longo do percurso da mangueira, além de uma indefinição do valor preciso da rigidez do material. Essas incertezas, podem, portanto, influenciar no fenômeno de vibração das linhas flexíveis.

1.2 Objetivos

1.2.1 Objetivo geral

Este trabalho tem por objetivo investigar o efeito de incertezas nos parâmetros construtivos e operacionais sobre o comportamento dinâmico de mangueiras flexíveis pressurizadas utilizadas em sistemas propulsivos de veículos espaciais.

1.2.2 Objetivos específicos

De maneira a alcançar o escopo do trabalho aqui descrito, são estabelecidos os seguintes objetivos específicos:

(a) Estudar o problema de vibração em linhas flexíveis de sistemas propulsivos.

(b) Analisar o atual método implementado pela NASA para validação de linhas flexíveis.

(c) Implementar este método analítico/empírico em linguagem computacional.

(d) Realizar análise de sensibilidade dos parâmetros utilizados no método.

(e) Implementar o método de Monte Carlo em linguagem computacional.

(f) Realizar simulações de Monte Carlo para análise de incertezas nos parâmetros construtivos e operacionais da mangueira flexível.

(g) Analisar distribuições de probabilidade nos parâmetros de entrada e saída do método.

1.3 Organização do trabalho

No capítulo 2 deste trabalho encontra-se toda a descrição do método utilizado para analise de vibrações induzidas pelo escoamento em mangueiras flexíveis, além da descrição do método utilizado para investigação de incertezas nos parâmetros governantes do problema.

No capitulo 3 é apresentada a metodologia aplicada na realização do presente trabalho, citando as ferramentas utilizadas no mesmo.

No capítulo 4 são apresentados os resultados e discussões das analises e simulações realizadas na presente monografia.

O capítulo 5 faz uma síntese do presente trabalho, com uma análise final dos resultados obtidos. Além de apresentar outros possíveis estudos relacionados ao tema aqui analisado.

2 Referencial Teórico

Ao longo deste capitulo será explicado o método utilizado para avaliação do fenômeno de vibração das mangueiras flexíveis, além de uma breve explicação do método utilizado para análise de incerteza dos parâmetros de entrada no cálculo.

2.1 Nomenclatura

Os parâmetros utilizados para descrição da geometria das mangueiras flexíveis que são aplicados nas equações que descrevem o problema de vibração nas mesmas estão especificados na Fig. 4

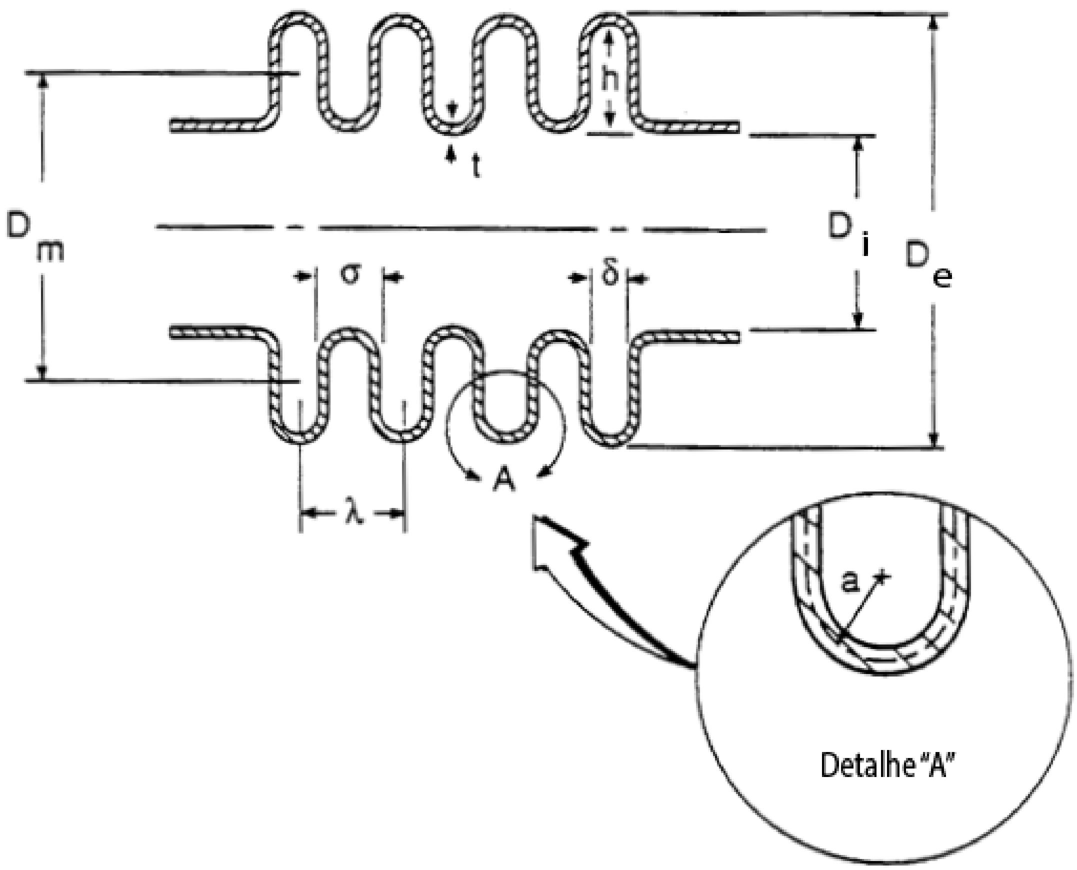

Figura 4 – Nomenclatura das convoluções[1].

[1] Adaptada de MSFC (1992).

$$N_p = \text{Número de camadas}$$
$$D_m = \text{Diâmetro médio}$$
$$D_i = \text{Diâmetro interno}$$
$$D_e = \text{Diâmetro externo}$$
$$t = \text{Espessura da camada}$$
$$\lambda = \text{Passo entre as convoluções}$$
$$\sigma = \text{Comprimento da convolução}$$
$$a = \text{Raio médio da convolução}$$
$$h = \text{Altura das convoluções}$$
$$\delta = \text{Espaço entre convoluções}$$

2.2 Instabilidade Fluido Estrutural

O fenômeno de vibração das convoluções de mangueiras flexíveis trata-se de um problema de instabilidade fluido-estrutural, onde a estrutura (convoluções) deve estar em movimento para causar atividade dinâmica de vórtice, ao passo que é necessária a existência de vórtices para geração de vibração estrutural, formando assim um evento não linear (TYGIELSKI; SMYLY; GERLACH, 1983). Este fenômeno é exemplificado na Fig. 5

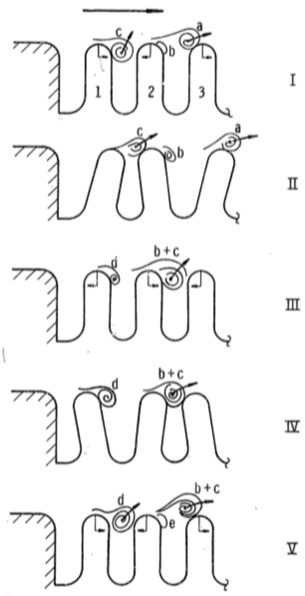

Figura 5 – Sequência de eventos de acoplamento fluido estrutural [2].

[2] Fonte: Gerlach e Schroeder (1969).

2.3 Número de Strouhal

Para determinada mangueira, com uma dada geometria interna, existem certas velocidades de escoamento que proporcionam máxima amplitude de excitação das convoluções em um modo de vibração correspondente. São, portanto, nessas velocidades que o processo de derramamento de vórtice possui melhor circunstância para fornecer energia do escoamento para o processo de vibração. Partindo desse fato, um ótimo método para correlacionar a frequência de vibração com a velocidade do escoamento e a geometria do sistema é através do numero adimensional de Stouhal (S), da forma:

$$S = \frac{fl}{V} \tag{2.1}$$

onde f representa a frequência, V a velocidade do escoamento e l um comprimento especifico, neste estudo será aplicado o comprimento da convolução σ (GERLACH; SCHROEDER, 1969).

2.4 Modelo Mecânico

Durante aplicação as mangueiras flexíveis tem o movimento longitudinal da base de suas convoluções limitado pela presença da trança externa, devido a pressão interna. Dessa forma a movimentação de uma convolução é independente das convoluções vizinhas. Com isso é possível modelar a dinâmica estrutural das convoluções como um sistema massa-mola engastado em ambas extremidades, como mostra a Fig. 6.

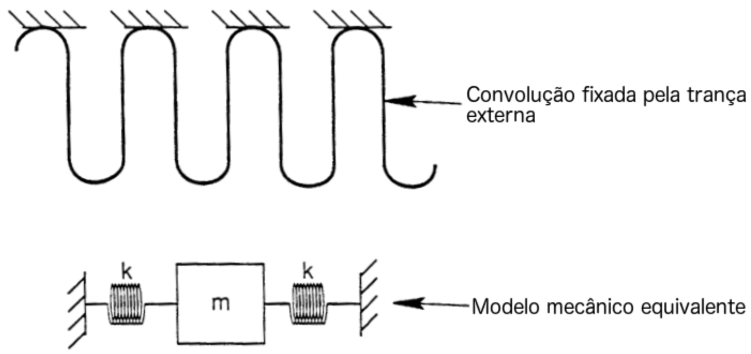

Figura 6 – Modelo mecânico da mangueira flexível[3].

Analisando-se a partir desse modelo seria possível apenas um modo de vibração, no entanto, com a presença do fluido, dois modos de acoplamento fluido-estrutural se tornam possíveis (GERLACH, 1969). Estes modos são: o modo ‘em-fase’, onde todas as convoluções se movimentam de maneira uniforme (Fig. 7), e o modo ‘fora-de-fase’, onde as

[3] Adaptada de MSFC (1992).

convoluções não possuem movimento uniforme, causando aceleração do fluido para dentro e fora das convoluções, com evidenciado na Fig. 8.

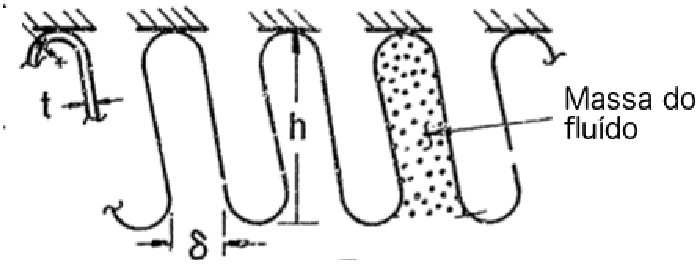

Figura 7 – Modo de vibração em fase[4].

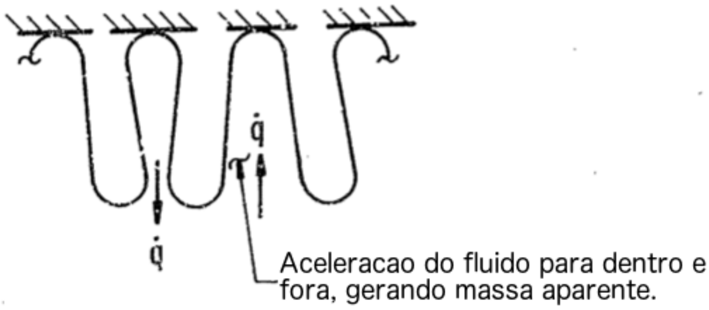

Figura 8 – Modo de vibração fora de fase[5].

Tais modos se diferem na massa atribuída ao modelo, onde: para o modo 'em-fase' a massa adicionada pelo escoamento corresponde apenas a massa contida entre as convoluções, e no modo 'fora-de-fase' a massa não corresponde a um valor real e sim a um valor de massa aparente devido ao movimento do fluido.

Nos experimentos realizados por Gerlach e Schroeder (1969) foi também observado um modo de vibração de maior ordem, onde ocorre flexão local em cada convolução, como ilustra a Fig. 9, nesse modo a massa fluídica utilizada é a mesma no modo fora-de-fase.

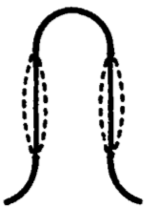

Figura 9 – Ilustração do modo de flexão local[6].

[4] Adaptada de Gerlach e Schroeder (1969).
[5] Adaptada de Gerlach e Schroeder (1969).
[6] Adaptada de Tygielski, Smyly e Gerlach (1983).

2.4.1 Constante elástica (k)

A constante elástica k para o referido modelo mecânico pode ser encontrada através da relação:

$$k = 2K_a \tag{2.2}$$

onde K_a corresponde a constante elástica para uma convolução inteira, preferencialmente determinada através de um ensaio com a mangueira flexível, quando tal ensaio não for possível, K_a pode ser estimado através da equação:

$$K_a = D_m E N_p (\tfrac{t}{h})^3 \tag{2.3}$$

onde E corresponde ao módulo de elasticidade do material da mangueira flexível.

2.4.2 Massa (m)

A massa usada no modelo corresponde a uma soma da massa do material da convolução (m_m) com a massa adicionada pelo escoamento correspondente ao modo de vibração, m_{EF} para o modo em fase e m_{FF} para os modos fora de fase e de flexão local.

$$m_m = \pi \rho_m D_m t N_p [\pi a + h - 2a] \tag{2.4}$$

$$m_{EF} = \frac{\pi}{2} \rho_f D_m h (2a - t N_p) \tag{2.5}$$

$$m_{FF} = \frac{0.68}{\delta} \rho_f D_m h^3 \tag{2.6}$$

2.4.3 Frequências

A frequência para cada modo de vibração da mangueira flexível podem ser calculadas por meio das equações:

$$f_{EF} = \frac{1}{2\pi} \sqrt{\frac{2k}{m_m + m_{EF}}} \tag{2.7}$$

$$f_{FF} = \frac{1}{2\pi} \sqrt{\frac{2k}{m_m + m_{FF}}} \tag{2.8}$$

$$f_{FL} = \frac{1}{2\pi} \sqrt{\frac{8k}{m_m + m_{FF}}} \tag{2.9}$$

onde,

$$f_{EF} = \text{Frequência para o modo em-fase}$$
$$f_{FF} = \text{Frequência para o modo fora-de-fase}$$
$$f_{FL} = \text{Frequência para o modo de flexão local}$$

2.5 Intervalos de excitação

Como mencionado anteriormente, as condições de excitação do fenômeno de vibração induzida pelo escoamento podem ser correlacionadas com o uso do número de Strouhal. Em posse das frequências correspondentes a cada modo de vibração, torna-se possível, com o uso da equação 2.1, o cálculo dos intervalos de velocidade do escoamento para qual cada modo irá ocorrer (GERLACH; SCHROEDER, 1969).

Utilizando o comprimento da convolução (σ) como dimensão especifica, tem-se a seguinte relação de Strouhal para o cálculo da faixa de velocidade (TYGIELSKI; SMYLY; GERLACH, 1983):

$$S_{\sigma i} = \text{Limite inferior de Strouhal} = 0.1$$
$$S_{\sigma c} = \text{Valor crítico de Strouhal} = 0.2$$
$$S_{\sigma s} = \text{Limite superior de Strouhal} = 0.3$$

Logo, para um modo de vibração da mangueira flexível com frequência f, a excitação fluídica pode ocorrer a partir de um limite inferior (V_I) até um limite superior (V_S), passando por um valor crítico (V_C):

$$V_I = \frac{f\sigma}{S_{\sigma s}} \tag{2.10}$$

$$V_C = \frac{f\sigma}{S_{\sigma c}} \tag{2.11}$$

$$V_S = \frac{f\sigma}{S_{\sigma i}} \tag{2.12}$$

2.6 Primeiro modo de ressonância acústica radial

Em mangueiras flexíveis com meio interno gasoso é possível a ocorrência de ressonância acústica radial. Esse modo acústico ocorre além dos modos em-fase, fora-de-fase e de flexão local (MSFC, 1992).

O primeiro modo acústico radial ocorre na frequência dada por:

$$f_{acus} = \frac{(FNCO)V_s}{2\pi r_i} \tag{2.13}$$

Onde V_s é a velocidade do som no meio, determinada pelas propriedades do fluido, r_i corresponde ao raio interno da mangueira flexível e $FNCO$ é um parâmetro empírico dado por:

$$FNCO = 3.8 - 16.72(h/r_i)^2 + 13.67(h/r_i)^3 \qquad \text{para } 0 \leq (h/r_i) < 0.4 \tag{2.14a}$$

$$FNCO = -0.336 + 0.935(h/r_i)^{-1} \qquad \text{para } 0.4 \leq (h/r_i) \leq 1 \tag{2.14b}$$

A velocidade na qual o primeiro modo de ressonância acústica radial ocorre é calculado por:

$$V_{acus} = \frac{f_{acus}\sigma}{S_{\sigma c}} \tag{2.15}$$

O cálculo do primeiro modo de ressonância acústica radial é necessário para a verificação de uma sobreposição deste modo com algum dos outros modos de vibração da mangueira flexível, caso ocorra tal sobreposição, é necessária a aplicação de um fator de correção no valor da tensão exercida no material das convoluções, como evidenciado na seção 2.8.

2.7 Cálculo de tensões

Caso a velocidade do escoamento na mangueira flexível esteja entre qualquer um dos intervalos de velocidades dos modos de vibração, assume-se a presença de vibração induzida pelo escoamento nas convoluções. Logo, é necessário o calculo de tensões exercidas no material da mangueira flexível para inferir sobre a existência de falha por fadiga ou ruptura do material (MSFC, 1992).

Nessa seção será, portanto, exposto o método desenvolvido por Tygielski, Smyly e Gerlach (1983) e descrito em MSFC (1992). Esse método provém da análise de diversos experimentos realizados no *Southwest Research Institute* e no *Marshall Space Flight Center*, dessa maneira, correspondem a equações modeladas empiricamente.

A tensão induzida pelo escoamento nas convoluções da mangueira flexível pode ser calculada através da equação:

$$T_{esc} = (EE)\left[\frac{C^{*}tP_D}{V'(CEE)\delta}\right](E)(C_{NP})(C_c)\left(\frac{1}{N_p}\right) \tag{2.16}$$

Os parâmetros nessa equação são descritos e podem ser calculados pelas relações:

$$EE = 1 + 0.1\left(\frac{2757902.8}{CEE}\right)^2 \tag{2.17}$$

Nota: O número 2757902.8 corresponde a uma constante elástica específica possuindo a dimensão $[N/m^2]$.

$$CEE = \frac{K_a}{D_m N_p} \tag{2.18}$$

Para o modo em-fase e fora-de-fase o calculo de C^* é feito através da equação:

$$C^* = \frac{C_1}{C_2 + (V')^2} + \frac{C_3|\sin(\pi V')|}{C_4 + (V')^2} + C_5 \tag{2.19}$$

Para o modo de flexão local $C^* = 0.4$.

$$V' = \frac{V_C}{V^*} \tag{2.20}$$

V_C = valor crítico da velocidade para cada um dos 3 modos de vibração.

V^* = valor crítico da velocidade para o modo fora-de-fase.

A pressão dinâmica P_D é dada por:

$$P_D = \frac{\rho_f (V_C)^2}{2} \tag{2.21}$$

O parâmetro C_{NP} corresponde a um coeficiente modificador de amortecimento para convoluções com mais de uma camada (TYGIELSKI; SMYLY; GERLACH, 1983), dado por:

$$C_{NP} = \begin{cases} 1, & \text{para } N_p = 1 \\ 1 + \frac{C_6(\sigma/h)}{1+C_7(V')^2}, & \text{para } N_p = 2, 3, ... \end{cases} \tag{2.22}$$

O termo C_c corresponde a um amplificador de tensão utilizado para computar o efeito de um cotovelo possivelmente presente anteriormente à entrada da linha flexível, dado por:

$$C_c = \begin{cases} 1, & \text{para nenhum cotovelo presente.} \\ 1 + \frac{4.7}{2+L/D}, & \text{para um cotovelo presente.} \end{cases} \tag{2.23}$$

Nota: L representa a distancia do cotovelo a entrada da linha e D o diâmetro da linha.

Os parâmetros $C_1, C_2, ..., C_7$ representam coeficientes empíricos sem dimensão derivados dos experimentos realizados por Tygielski, Smyly e Gerlach (1983), e possuem os valores:

$$
\begin{aligned}
C_1 &= 0.13 \\
C_2 &= 0.462 \\
C_3 &= 1.0 \\
C_4 &= 10.0 \\
C_5 &= 0.06 \\
C_6 &= 1.25 \\
C_7 &= 5.5
\end{aligned}
$$

2.8 Fator de correção

Os valores de tensões (T_{esc}) calculados na seção 2.7 são derivados de uma análise puramente empírica do problema, portanto fatores de correção (FC) são aplicados para cálculo de valores de tensões corrigidas ($T_{esc}C$), de maneira que:

$$T_{esc}C = T_{esc} \cdot FC \tag{2.24}$$

Os fatores determinados por MSFC (1992) são:

- 2.0, para uma mangueira onde a constante elástica foi calculada experimentalmente.
- 2.5, para uma mangueira onde a constante elástica é estimada pela equação 2.3.

Caso ocorra sobreposição do modo acústico radial com algum dos modos de vibração, os fatores devem ser multiplicados por 7.5, caso as frequências modais sejam menores que a frequência do primeiro modo de ressonância acústico radial, o valor da tensão não sofre ajuste pelo fator acústico (MSFC, 1992).

2.9 Método de Monte Carlo

O método de Monte Carlo (MMC) é usado comumente em aplicações cientificas e na engenharia como ferramenta matemática na simulação de problemas com representatividade estocástica. Pode ser descrito como um método estatístico, que realiza, a partir da geração de uma sequência de números aleatórios, a simulação de um sistema real (YORIYAZ, 2009).

Para aplicar uma simulação de Monte Carlo é necessária a reprodução de aleatoriedade por um algoritmo de computador, que é determinístico por natureza (JOHANSEN; EVERS, 2008). São, portanto, usados números pseudorrandômicos na utilização do MMC. Segundo Ripley (1987), uma sequência pseudorrandômica de números corresponde a uma sequência determinística de números com as mesmas propriedades estatísticas de uma sequência randômica de valores.

O MMC utiliza cálculos computacionais para a simulação de um grande número de análises com entradas e saídas randômicas. Quando aplicado a avaliação de incertezas, os números randômicos são utilizados para amostrar aleatoriamente o espaço de incerteza dos parâmetros (PAPADOPOULOS; YEUNG, 2001). Dessa forma, cada distribuição randômica dos valores de entrada corresponde a um cenário provável do sistema real.

2.10 Distribuição Gama

A distribuição Gama é caracterizada por ser uma função densidade de probabilidade válida para valores reais positivos e ajustada por dois parâmetros, α de escala e β de forma. Sua formulação geral é escrita como

$$f_{\alpha,\beta}(x) = \frac{x^{\alpha-1}e^{-x/\beta}}{\beta^{\alpha}\Gamma(\alpha)} \tag{2.25}$$

onde $\Gamma = \Gamma(\alpha)$ é uma função gama e α e β devem necessariamente ser maiores que zero (MONTGOMERY; RUNGER, 2010). Os valores de α e β podem ser calculados a partir da média (μ) e do desvio padrão (s), como:

$$\alpha = \frac{\mu^2}{s^2} \qquad \beta = \frac{s^2}{\mu} \tag{2.26}$$

Comparando-se a distribuição gama com a normal, é possível notar que as principais diferenças entre ambas é a validade da primeira apenas para valores reais positivos, quanto que no caso gaussiano a validade é sobre toda a reta real, e no que tange a simetria, já que a distribuição gama possui obliquidade maior que zero, quanto que a distribuição gaussiana é simétrica. Ambas características fazem da distribuição gama adequada para diversas situações onde o parâmetro a ser estudado possui significado físico apenas em valores positivos, como é o caso de distâncias espaciais e o tempo a partir de um evento (CUNHA, 2017).

3 Metodologia

O presente trabalho propõe-se a realizar um estudo de incertezas perante a vibração induzida pelo escoamento em mangueiras flexíveis. Para atingir tal objetivo a metodologia a seguir foi implementada.

3.1 Implementação computacional do método

Primeiramente foi realizado um estudo do método encontrado na pesquisa bibliográfica que permite a análise vibratória de linhas flexíveis. Com o domínio de tal método, este foi implementado na linguagem de programação Python, permitindo uma análise mais ágil do fenômeno, sendo facilmente possível a modificação manual dos parâmetros descritivos da mangueira flexível.

O código foi então validado a partir de exemplos presentes no apêndice B do MSFC (1992) para mangueiras flexíveis. Para validação foram verificados todos resultados do método, como os valores de faixa de velocidades de acoplamento, os valores de tensões para cada modo de vibração além do primeiro modo de ressonância acústica.

3.2 Análise de sensibilidade

Com o codigo validado, foi realizada uma análise de sensibilidade dos parâmetros construtivos e operacionais independentes da mangueira flexível. A análise foi realizada variando separadamente cada valor de entrada em mais e menos vinte porcento de seu valor nominal, calculando nessa faixa os valores de velocidade do escoamento e de limite inferior de velocidade (V_I) para todos os modos de acoplamento.

Os parametros análisados foram: o comprimento da convolução (σ), altura das convoluções (h), espessura da camada (t), diâmetro interno (D_i), módulo de elasticidade (E), densidade do material das convoluções (ρ_m), densidade do fluido (ρ_f) e a vazão mássica ($\dot{m}$).

Os parâmetros que não foram analisados foram: o passo entre as convoluções (λ) por não ser considerado nos cálculos propostos em MSFC (1992), o diâmetro externo (D_e) por ter relação direta com D_i e h, além da pressão e temperatura do fluido por estarem relacionadas a vazão mássica e a densidade do fluído.

3.3 Simulação de Monte Carlo

O módulo *random* da linguagem de programação *Python* foi utilizado para implementação do método de Monte Carlo. A simulação foi feita a partir da elaboração de um algoritmo de geração de mangueiras flexíveis que estabelece os parâmetros de entrada como variáveis aleatórias. A aleatoriedade foi feita com base na função de densidade de probabilidade (FDP) da distribuição gamma, utilizando como valor médio o valor nominal de cada parâmetro e como desvio padrão 5% do valor médio.

O estudo foi realizado estabelecendo isoladamente cada um dos parametros construtivos e operacionais como variaveis aleatórias bem como todos os parâmetros sendo randomizados de uma vez.

A análise do efeito de incertezas foi realizada a partir da comparação entre os valores de velocidade do escoamento e intervalo de velocidades de acoplamento para cada exemplar de mangueira flexível gerado. No exemplo utilizado os valores nominais dos parâmetros não ocasionam em acoplamento. Dessa forma, o efeito de incerteza nos parâmetros construtivos e operacionais da mangueira flexivel é mensurado a partir da porcentagem dos exemplares gerados onde ocorra acoplamento.

3.3.1 Análise de influência do desvio padrão

A partir da simulação de Monte Carlo foi estudado o efeito da variação do desvio padrão (em função do valor nominal do parâmetro) utilizado na geração randômica de um dos parametros de maior sensibilidade estudado.

3.3.2 Geração de histogramas

Foram gerados histogramas para verificação da função de densidade de probabilidade nos valores de entrada e de saída do metodo.

3.4 Valores nominais dos parâmetros

Os valores nominais de cada parâmetro utilizados nas análises realizadas nesse trabalho estão listados no apêndice A.

4 Resultados e Discussão

4.1 Implementação

A implementação do trabalho em *Python* foi realizada a partir da divisão do problema em três classes. Uma classe correspondente a mangueira flexível contendo todos os parâmetros construtivos da mesma, outra classe correspondente ao fluído contendo os parâmetros operacionais do sistema e uma terceira classe que recebe os valores impostos pelas outras duas classes e contém todas as equações que descrevem o fenômeno de vibração induzida pelo escoamento em mangueiras flexíveis.

Com o uso dessas três classes é possível realizar os cálculos de frequências, faixas de velocidades de acoplamento e tensões para todos os modos de vibração de uma dada mangueira flexível. Para tal foi então concebido um *script main.py* que recebe como entrada as três classes e realiza a verificação de acoplamento para cada modo de vibração da linha flexível, além de imprimir na tela os valores de frequência, faixas de velocidade e tensões correspondentes a cada modo. A Fig. 10 mostra um fluxograma representativo de tal *script*.

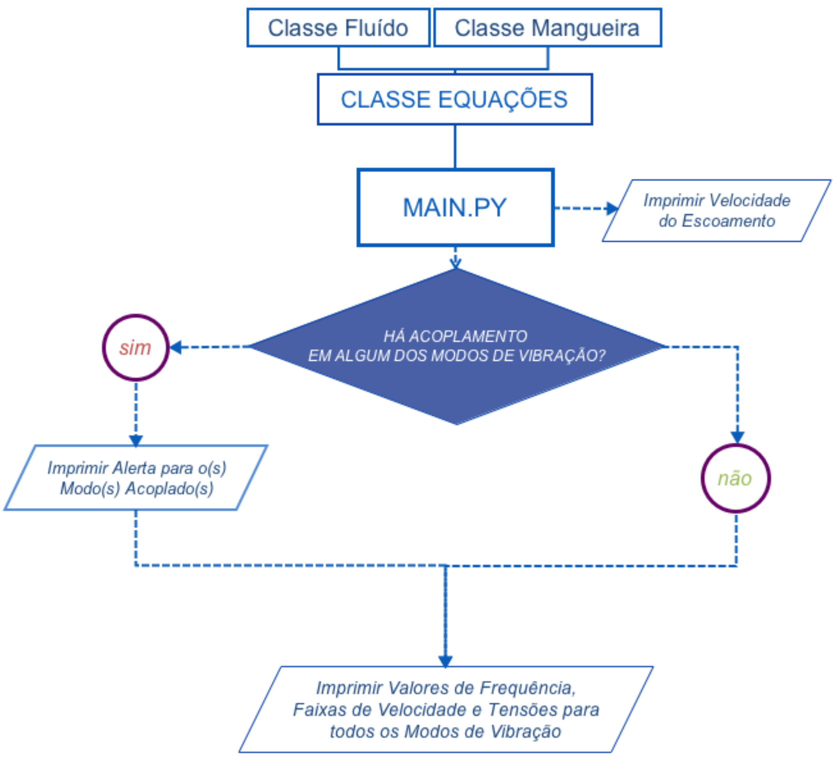

Figura 10 – Fluxograma do *script main.py*

Em posse dos três arquivos de classe e do *script main.py* foi possível realizar a validação da implementação computacional do método de análise utilizado. A Fig. 11 mostra o retorno do *script main.py* na validação do mesmo, utilizando valores provenientes do apêndice B do MSFC (1992), também disponíveis no apêndice A.

```
➜  TCC git:(master) ✗ python main.py
The flow velocity is: 234.84 m/s

There is frequency coupling for the in-phase mode of vibration!!

There is frequency coupling for the out-of-phase mode of vibration!!

There is frequency coupling for the convolute bending mode of vibration!!

There is acoustic resonance phenomenon!!

The coupling frequency for the in-phase mode of vibration is:
f(IP) = 13675.01 Hz
The velocity range for the in-phase mode of vibration is:
Lower limit (IP)    : 83.36 m/s
Critical value (IP) : 125.04 m/s
Upper limit (IP)    : 250.09 m/s

The coupling frequency for the out-of-phase mode of vibration is:
f(OP) = 13640.79 Hz
The velocity range for the out-of-phase mode of vibration is:
Lower limit (OP)    : 83.15 m/s
Critical value (OP) : 124.73 m/s
Upper limit (OP)    : 249.46 m/s

The coupling frequency for the convolute bending mode of vibration is:
f(CB) = 27281.59 Hz
The velocity range for the convolute bending mode of vibration is:
Lower limit (CB)    : 166.31 m/s
Critical value (CB) : 249.46 m/s
Upper limit (CB)    : 498.93 m/s

The first radial acoustic mode frequency is:
fλ : 23755.13 Hz
The first radial acoustic mode velocity is:
Vλ : 217.22 m/s

The predicted Flow-Induced Stress (FIV) for the in-phase mode of vibration is:
FIV(IP) = 9804762.37 Pa
The predicted Flow-Induced Stress (FIV) for the out-of-phase mode of vibration is:
FIV(OP) = 9749196.98 Pa
The predicted Flow-Induced Stress (FIV) for the convolute bending mode of vibration is:
FIV(CB) = 420636614.71 Pa
```

Figura 11 – Retorno do *script main.py*

4.2 Análise de sensibilidade

Realizada a análise de sensibilidade dos parâmetros de entrada, foram obtidos os seguintes gráficos correspondentes as distribuições de velocidades para cada um dos modos de acoplamento.

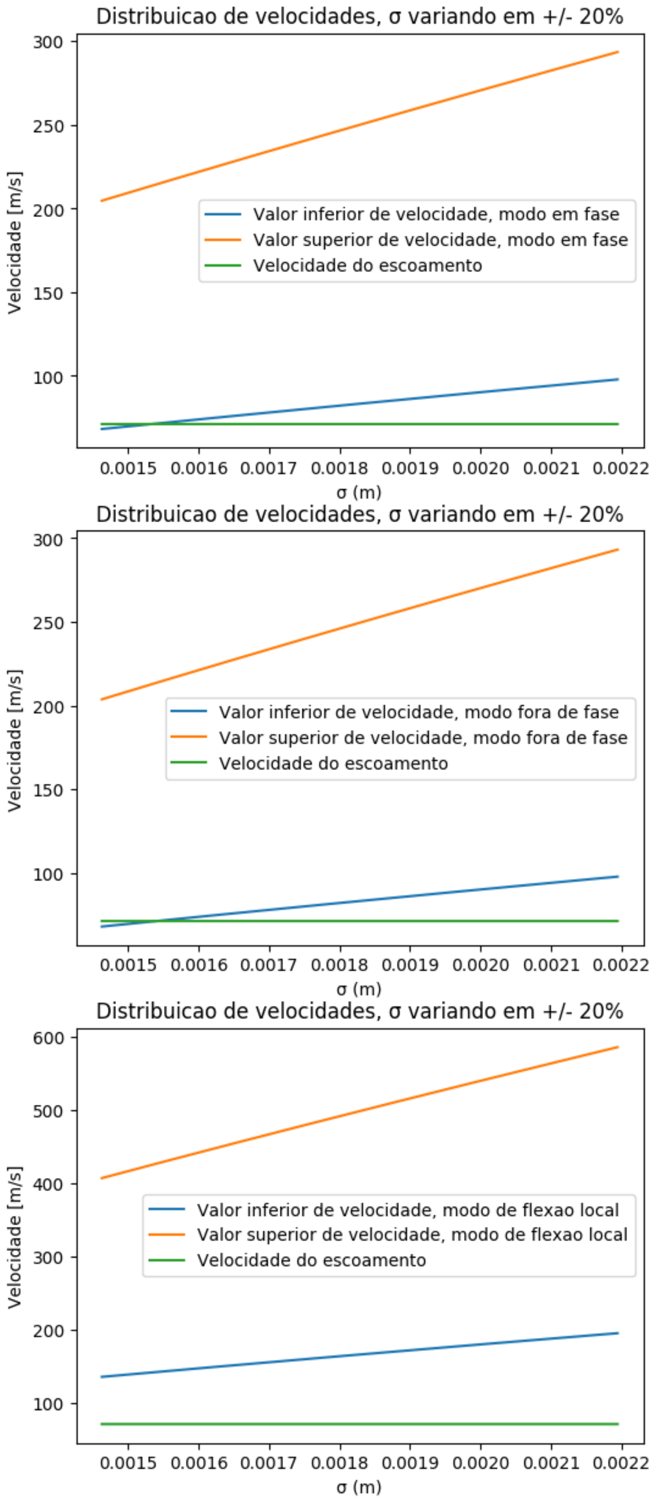

Figura 12 – Distribuição de velocidades na análise de sensibilidade de σ

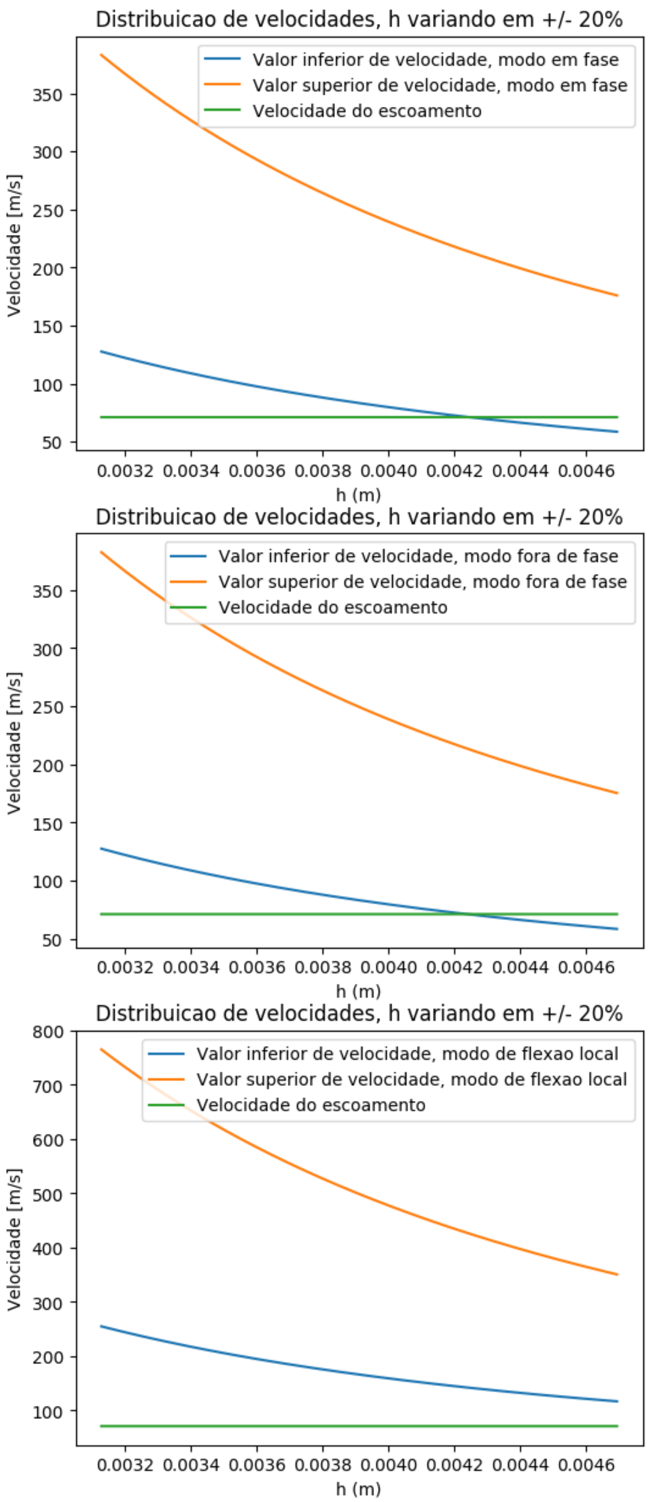

Figura 13 – Distribuição de velocidades na análise de sensibilidade de h

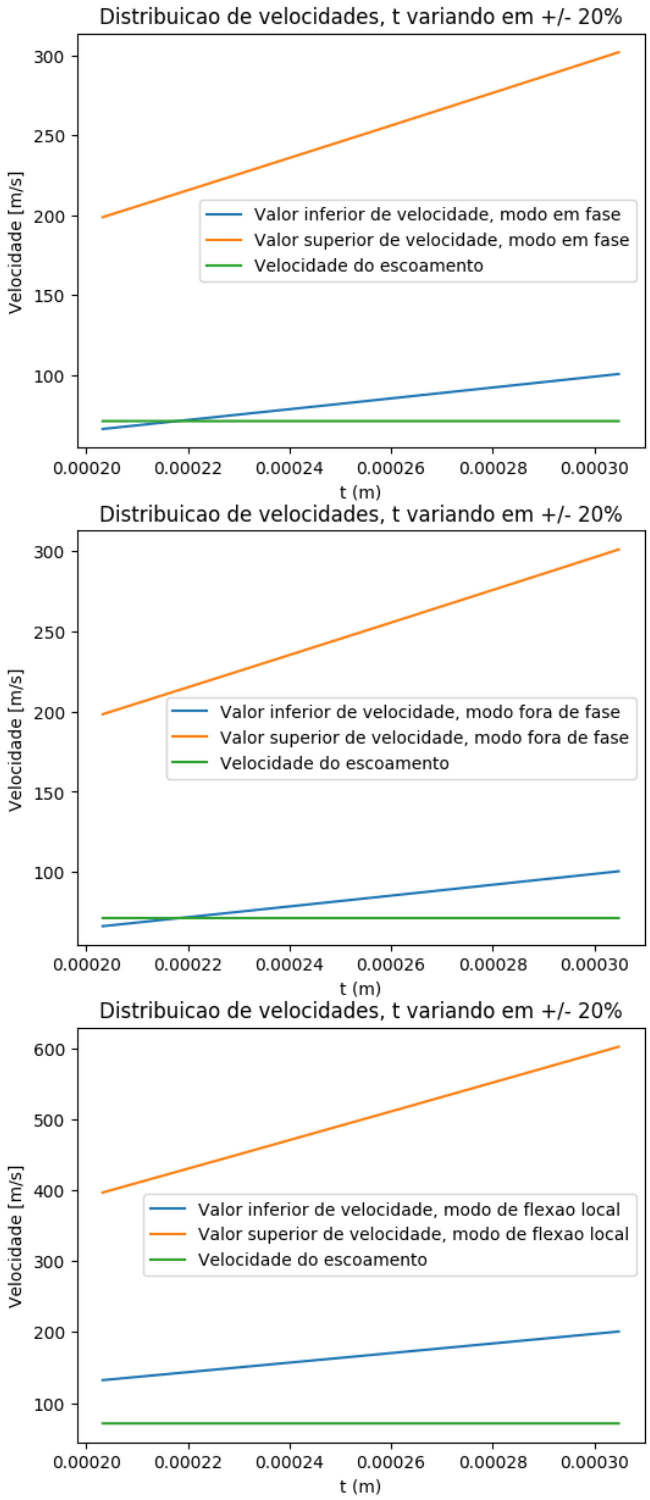

Figura 14 – Distribuição de velocidades na análise de sensibilidade de t

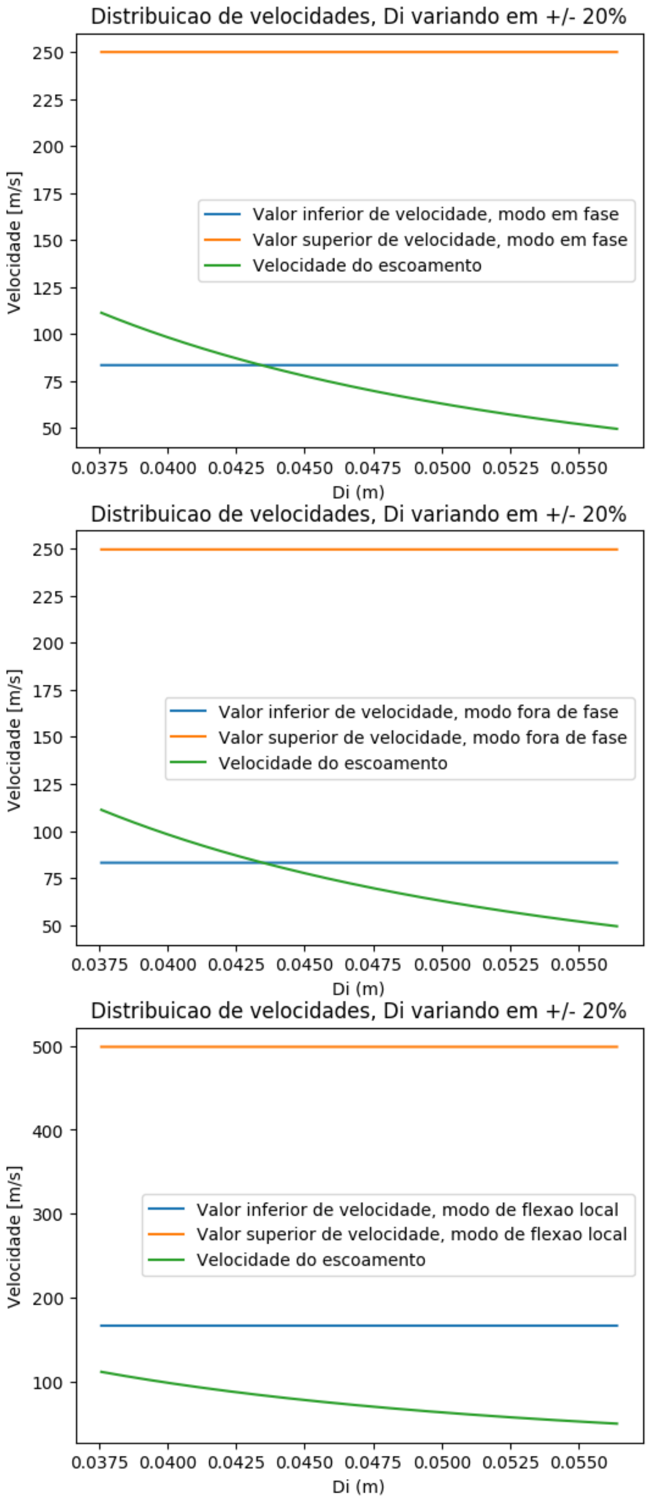

Figura 15 – Distribuição de velocidades na análise de sensibilidade de D_i

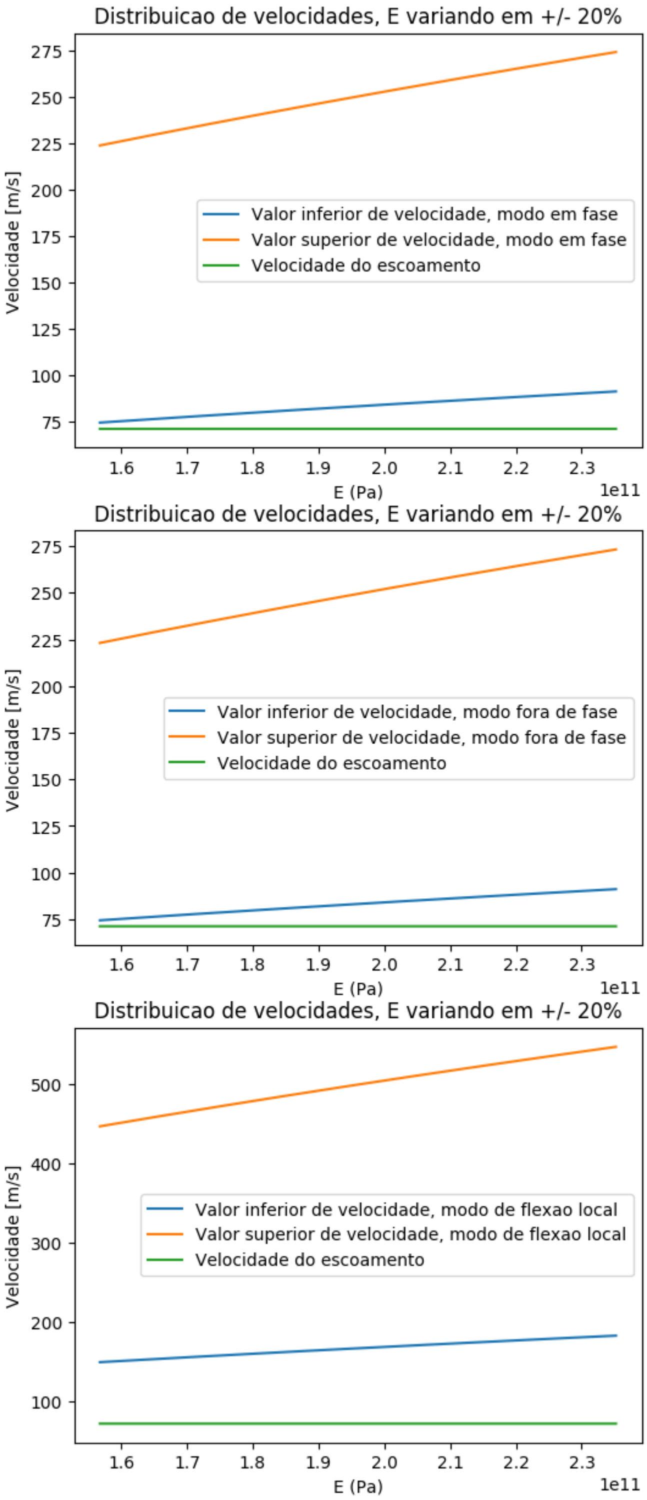

Figura 16 – Distribuição de velocidades na análise de sensibilidade de E

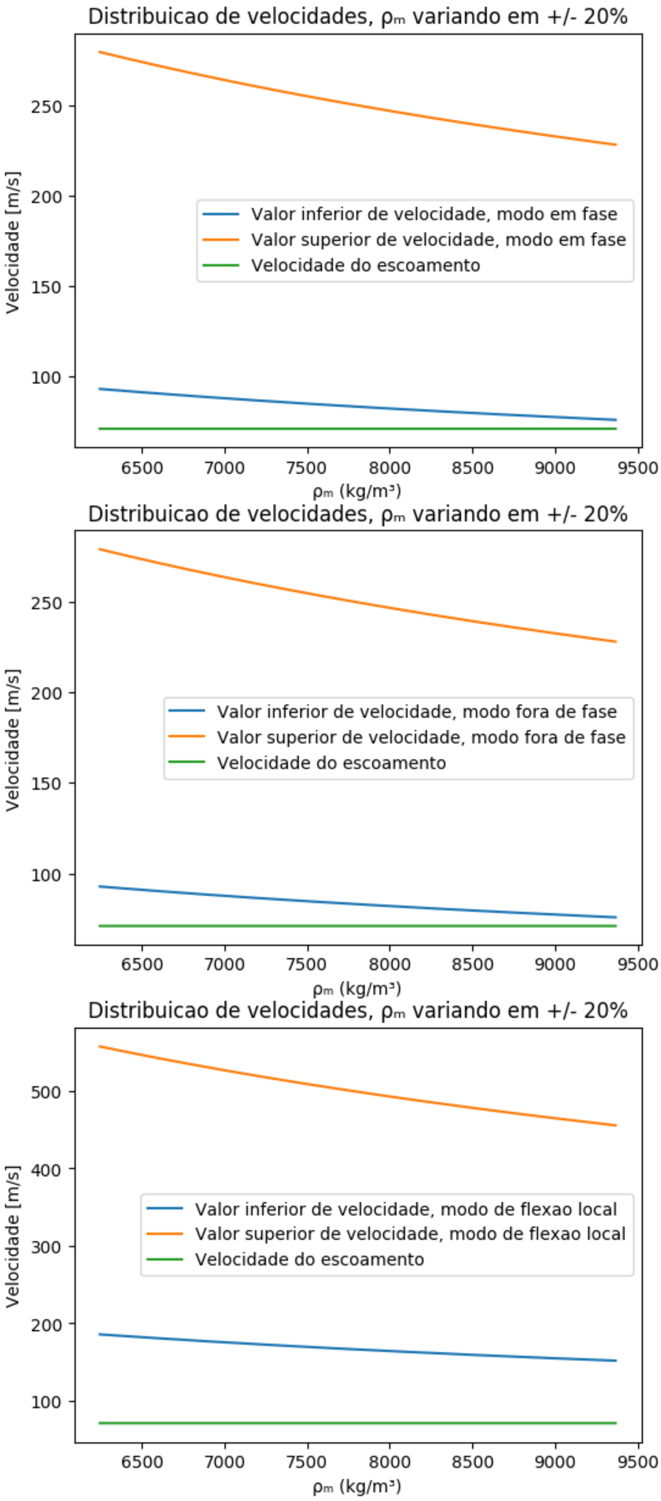

Figura 17 – Distribuição de velocidades na análise de sensibilidade de ρ_m

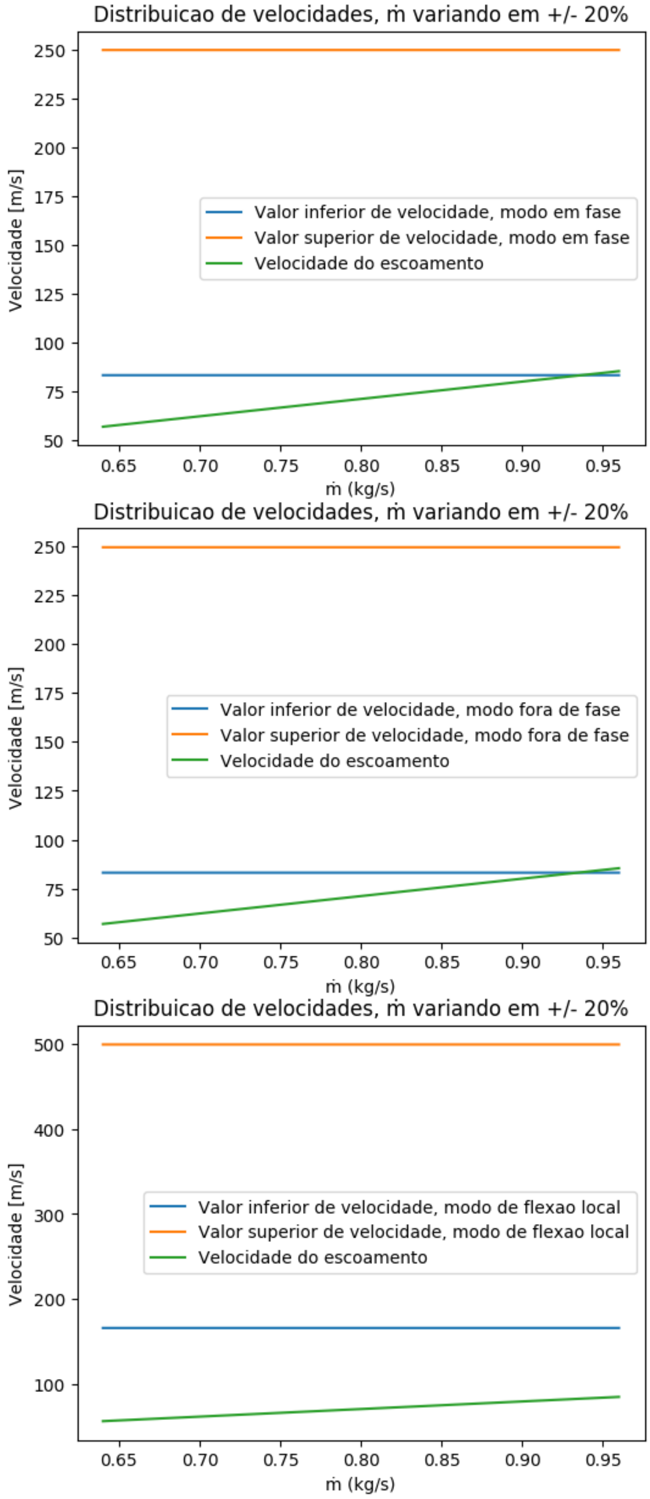

Figura 18 – Distribuição de velocidades na análise de sensibilidade de $\dot{m}$

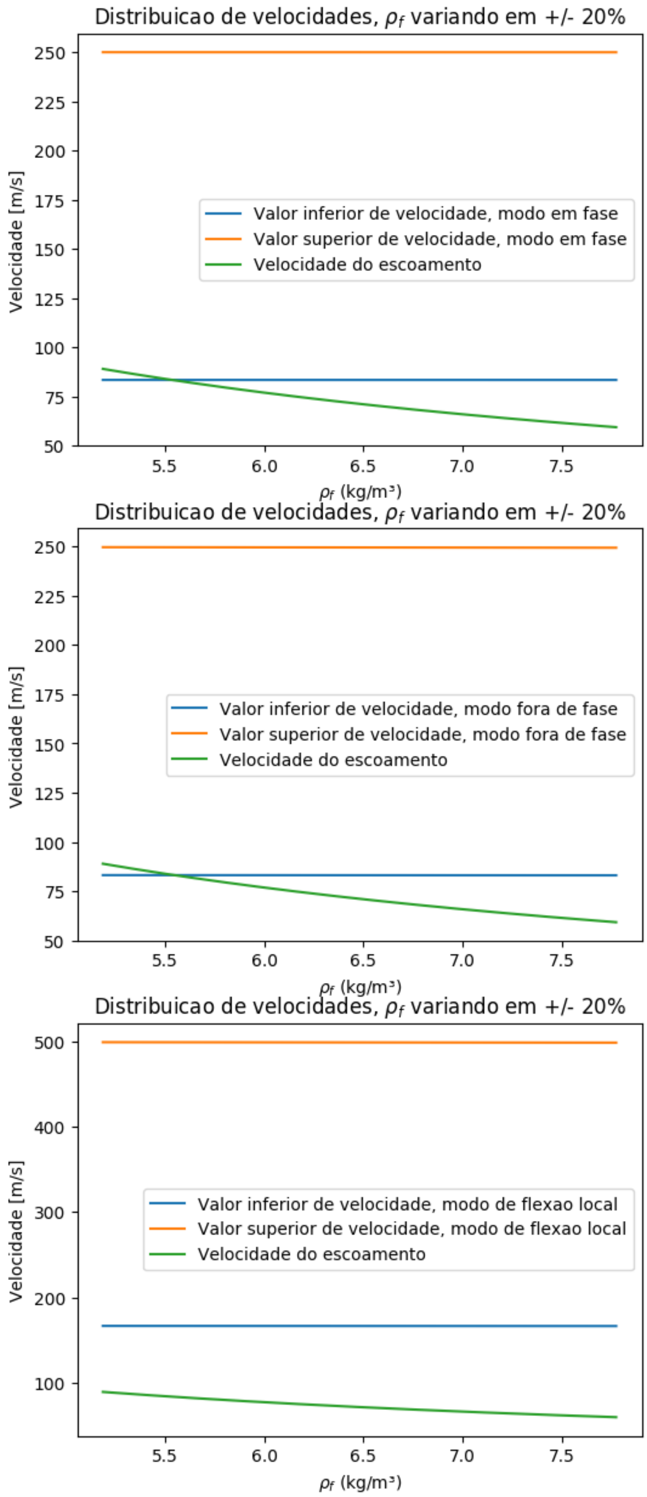

Figura 19 – Distribuição de velocidades na análise de sensibilidade de ρ_f

Através de tais gráficos é possivel notar que a variação dos parâmetros σ, h, t, E e ρ_m provoca uma alteração no valor das velocidades de acoplamento. Já para os parâmetros D_i, $\dot{m}$ e ρ_f a mudança dá-se na velocidade do escoamento.

Além dos gráficos presentes nas figuras 12 a 19 as tabelas 1 a 5 demostram resultados da análise de sensibilidade.

Tabela 1 – Porcentagem de acoplamento em cada modo de vibração

Parâmetro Variado	**Modo em fase (%)**	**Modo fora de fase (%)**	**Modo de flexão local(%)**
σ	9.58	10.58	0.00
h	28.60	29.03	0.00
t	14.46	14.95	0.00
D_i	31.03	31.32	0.00
E	0.00	0.00	0.00
ρ_m	0.00	0.00	0.00
$\dot{m}$	7.25	7.99	0.00
ρ_f	13.49	13.95	0.00

A tab. 1 lista a porcentagem de exemplares de mangueira flexiveis com acoplamento de velocidades para cada um dos modos de vibração.

Tabela 2 – Análise de sensibilidade, velocidade inferior para modo em fase

Parâmetro Variado	**Valor Min.**	**Valor Nominal**	**Valor Máx.**	**Faixa de variação a esquerda (%)**	**Faixa de variação a direita (%)**	**Faixa de variação total (%)**
σ	68.24	83.36	97.86	18.14	17.39	35.53
h	58.69	83.36	127.70	29.60	53.19	82.79
t	66.27	83.36	100.67	20.50	20.77	41.27
D_i	83.36	83.36	83.36	0.00	0.00	0.00
E	74.56	83.36	91.32	10.56	9.54	20.10
ρ_m	76.10	83.36	93.20	8.71	11.80	20.50
$\dot{m}$	83.36	83.36	83.36	0.00	0.00	0.00
ρ_f	83.37	83.36	83.36	0.00	0.00	0.00

Tabela 3 – Análise de sensibilidade, velocidade inferior para modo fora de fase

Parâmetro Variado	**Valor Min.**	**Valor Nominal**	**Valor Máx.**	**Faixa de variação a esquerda (%)**	**Faixa de variação a direita (%)**	**Faixa de variação total (%)**
σ	67.89	83.15	97.72	18.35	17.51	35.87
h	58.46	83.15	127.52	29.69	53.35	83.05
t	66.12	83.15	100.38	20.49	20.71	41.20
D_i	83.15	83.15	83.15	0.00	0.00	0.00
E	74.38	83.15	91.09	10.56	9.54	20.10
ρ_m	75.94	83.15	92.90	8.67	11.73	20.40
$\dot{m}$	83.15	83.15	83.15	0.00	0.00	0.00
ρ_f	83.11	83.15	83.20	0.06	0.06	0.12

Nas tabelas 2, 3 e 4 são listadas as faixas de variações (em porcentagem) a esquerda e direita do valor nominal da velocidade inferior para cada modo de vibração. Especificando os valores minimos e máximos dos limites inferiores de velocidade a partir da variação em mais e menos 20% de cada um dos parâmetros de entrada no sistema.

Tabela 4 – Análise de sensibilidade, velocidade inferior para modo de flexão local

Parâmetro Variado	**Valor Min.**	**Valor Nominal**	**Valor Máx.**	**Faixa de variação a esquerda (%)**	**Faixa de variação a direita (%)**	**Faixa de variação total (%)**
σ	135.79	166.31	195.43	18.35	17.51	35.87
h	116.92	166.31	255.03	29.69	53.35	83.05
t	132.23	166.31	200.75	20.49	20.71	41.20
D_i	166.31	166.31	166.31	0.00	0.00	0.00
E	148.75	166.31	182.18	10.56	9.54	20.10
ρ_m	151.89	166.31	185.81	8.67	11.73	20.40
$\dot{m}$	166.31	166.31	166.31	0.00	0.00	0.00
ρ_f	166.22	166.31	166.40	0.06	0.06	0.12

Tabela 5 – Análise de sensibilidade, velocidade do escoamento

Parâmetro Variado	**Valor Min.**	**Valor Nominal**	**Valor Máx.**	**Faixa de variação a esquerda (%)**	**Faixa de variação a direita (%)**	**Faixa de variação total (%)**
σ	71.19	71.19	71.19	0.00	0.00	0.00
h	71.19	71.19	71.19	0.00	0.00	0.00
t	71.19	71.19	71.19	0.00	0.00	0.00
D_i	49.44	71.19	111.23	30.56	56.25	86.81
E	71.19	71.19	71.19	0.00	0.00	0.00
ρ_m	71.19	71.19	71.19	0.00	0.00	0.00
$\dot{m}$	56.95	71.19	85.43	20.00	20.00	40.00
ρ_f	59.32	71.19	88.99	16.67	25.00	41.67

Na tab. 5 são listadas as faixas de variações (em porcentagem) a esquerda e direita do valor nominal da velocidade do escoamento. Especificando os valores minimos e máximos da velocidade do escoamento a partir da variação em mais e menos 20% de cada um dos parâmetros de entrada no sistema.

A partir da análise de sensibilidade foi possivel observar que a altura das convoluções (h) e o diâmetro interno (D_i) da mangueira flexivel são os parametros com maior sensibilidade no valor da resposta do sistema. Sendo h um parâmetro que intefere no valor da faixa de velocidade de acoplamento para cada um dos modos de vibração e D_i um parâmetro relacionado a velocidade do escoamento na mangueira flexível. A maior sensibilidade de tais parâmetros é também confirmada através da tab. 1 onde a variação de h e D_i proporcionou uma maior porcentagem de acoplamento em relação aos outros parâmetros.

Outra informação que pode ser retirada da análise de sensibilidade é a semelhança no comportamento dos modos de vibração em resposta a variação dos parâmetros de entrada. Em razão disso o método de Monte Carlo foi implementado na análise do modo de vibração em fase, por ser suficiente para compreensão do fenômeno.

4.3 Simulação de Monte Carlo

Validada a implementação do método e feita a análise de sensibilidade dos parâmetros de entrada do sistema, foi criado um novo *script* em *Python* para implementação do metodo de Monte Carlo na análise de incertezas dos parâmetros construtivos e operacionais da mangueira flexível.

Após receber as classes anteriormente criadas, tal *script* inicia um loop para criação de exemplares de mangueiras flexíveis com certos parâmetros randomizados, anexando em uma lista a velocidade do escoamento bem como os valores inferiores (V_I) e superiores (V_S) de velocidade para o modo de vibração em análise. Em cada iteração desse mesmo loop é feita a verificação de acoplamento da velocidade do escoamento com a faixa de velocidade do modo de vibração analisado, utilizando um contador para armazenar o número de mangueiras com acoplamento.

Logo após o término do loop, tem-se o número total de mangueiras geradas e o número de mangueiras sujeitas ao fenômeno vibratório. Tornando possível o cálculo da porcentagem de mangueiras geradas com parâmetros randomizados com acoplamento. Em posse da lista de valores de velocidades é possível a *plotagem* de um gráfico de tais informações, para melhor análise dos dados a lista foi ordenada do menor ao maior valor de velocidade do escoamento.

A Fig. 20 mostra o fluxograma do *script Monte-Carlo-Analysis.py*.

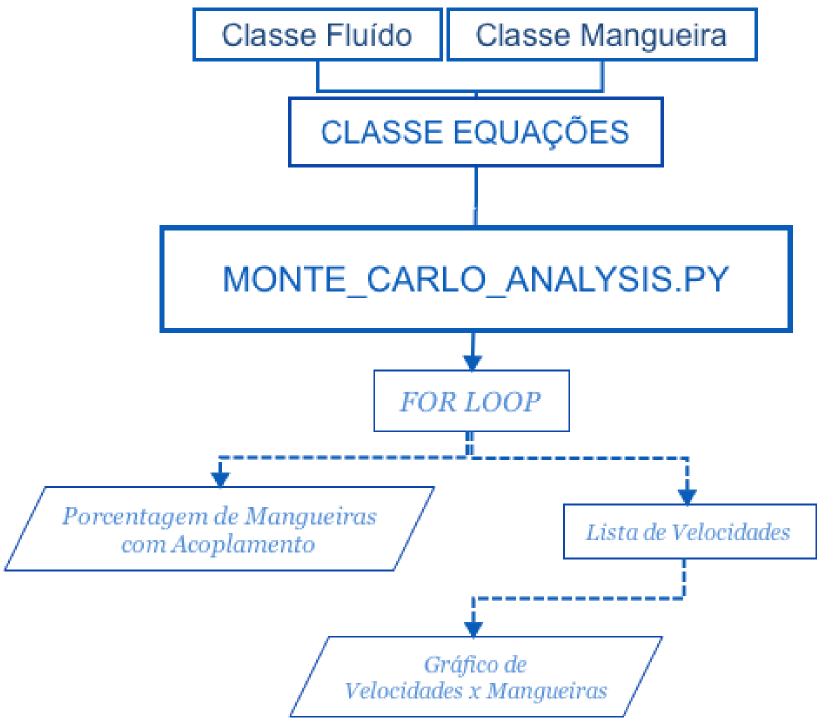

Figura 20 – Fluxograma do *script Monte-Carlo-Analysis.py*

O método de Monte Carlo foi aplicado randomizando cada um dos parametros analisados na análise de sensibilidade separadamente seguido de uma simulação com to-

Tabela 6 – Casos simulados pelo Método de Monte Carlo

Simulações	Descrição da Simulação	$N°$ de exemplares randômicos de mangueira flexível
01	variar σ	1000000
02	variar h	1000000
03	variar t	1000000
04	variar D_i	1000000
05	variar E	1000000
06	variar ρ_m	1000000
07	variar $\dot{m}$	1000000
08	variar ρ_f	1000000
09	variar todos os parâmetros	1000000
10	variar todos os parâmetros	10000000

dos os parâmetros como variáveis aleatórias. A tabela 6 lista os casos simulados e suas descrições.

As figuras 21 a 30 exibem os gráficos gerados em cada uma das simulações.

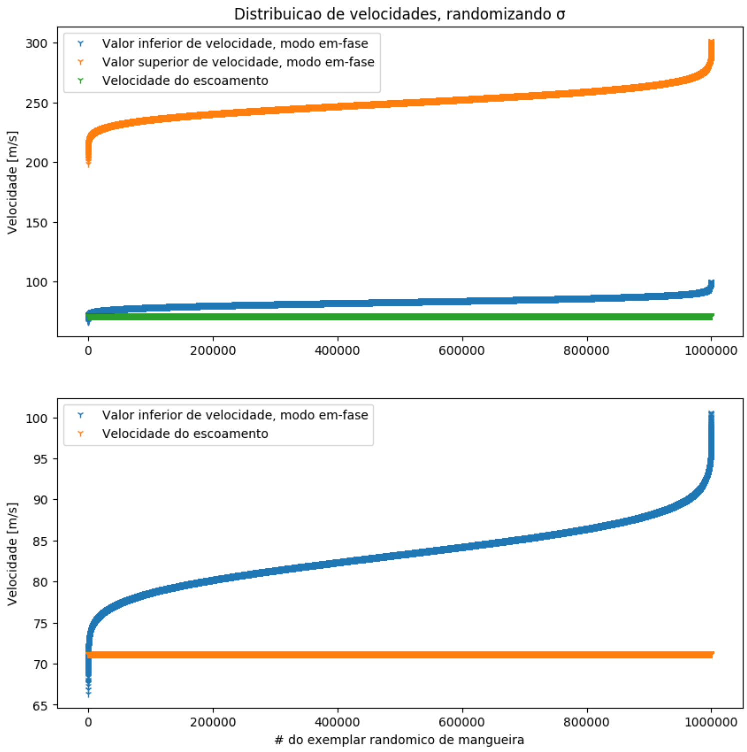

Figura 21 – Distribuição de velocidades na simulação 01

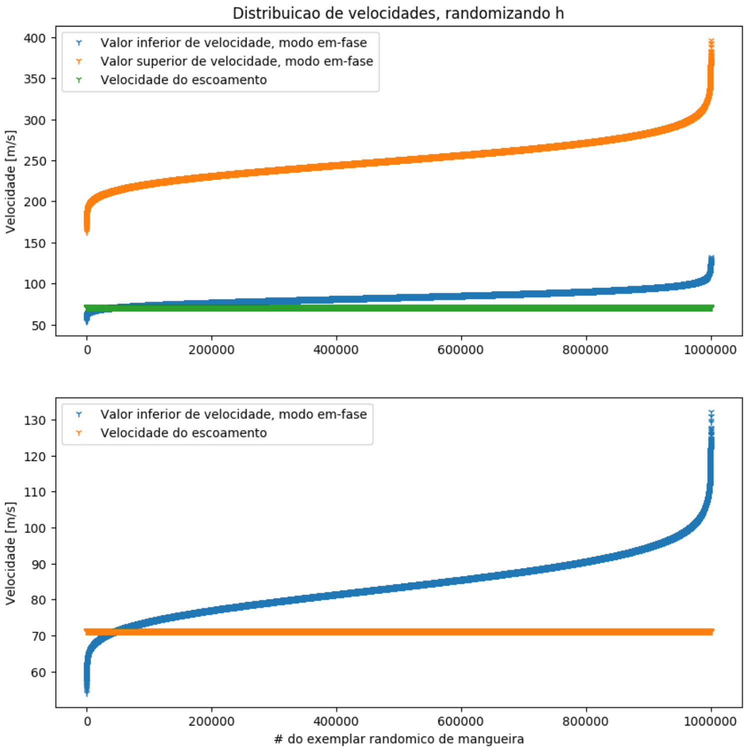

Figura 22 – Distribuição de velocidades na simulação 02

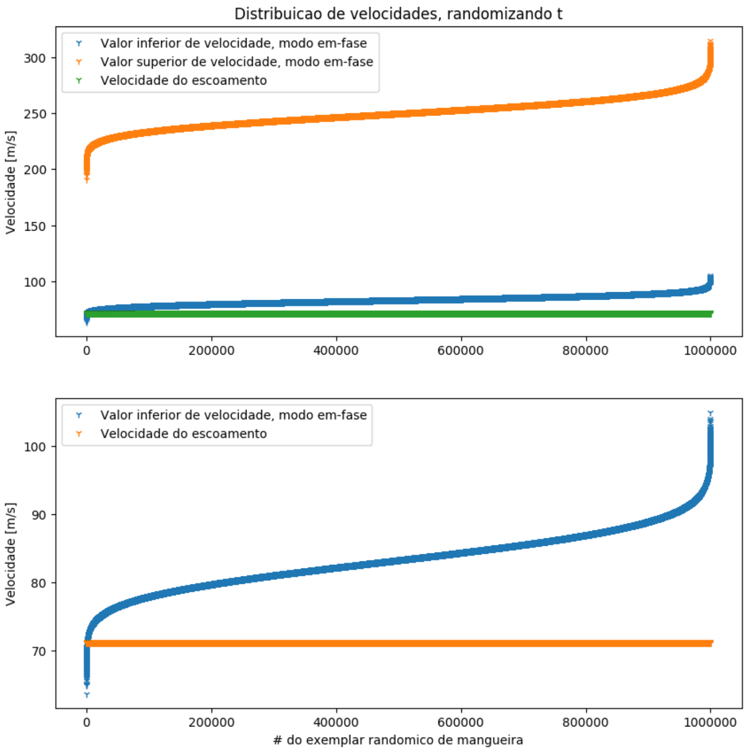

Figura 23 – Distribuição de velocidades na simulação 03

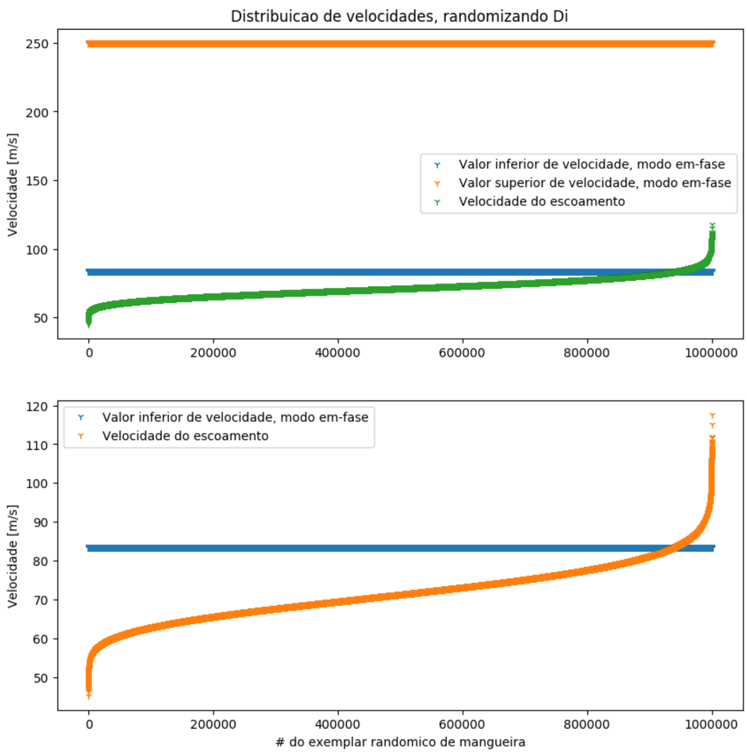

Figura 24 – Distribuição de velocidades na simulação 04

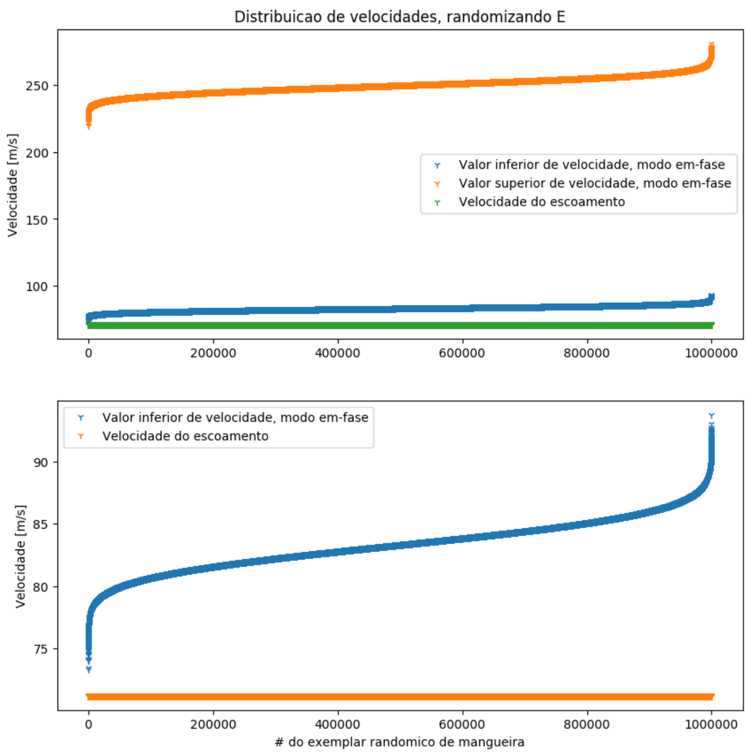

Figura 25 – Distribuição de velocidades na simulação 05

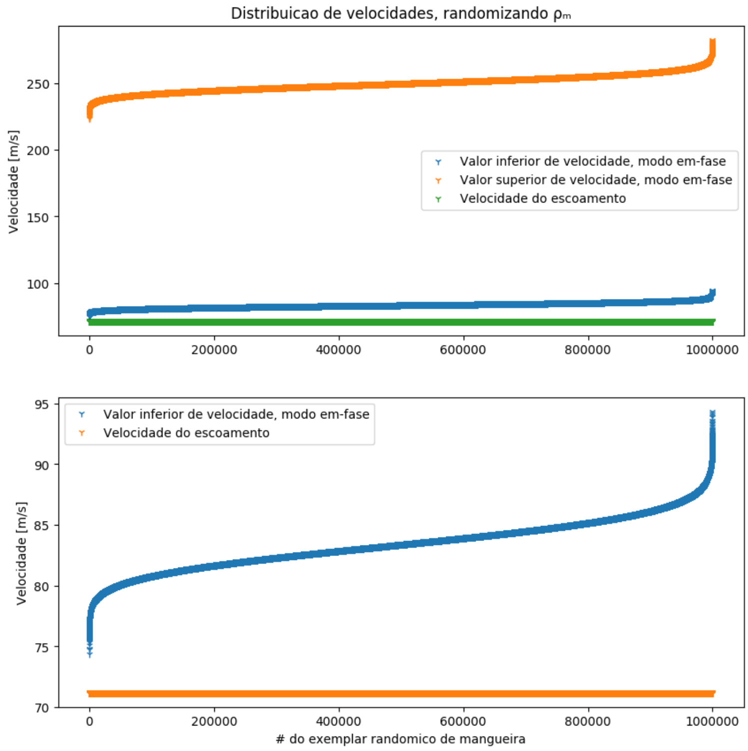

Figura 26 – Distribuição de velocidades na simulação 06

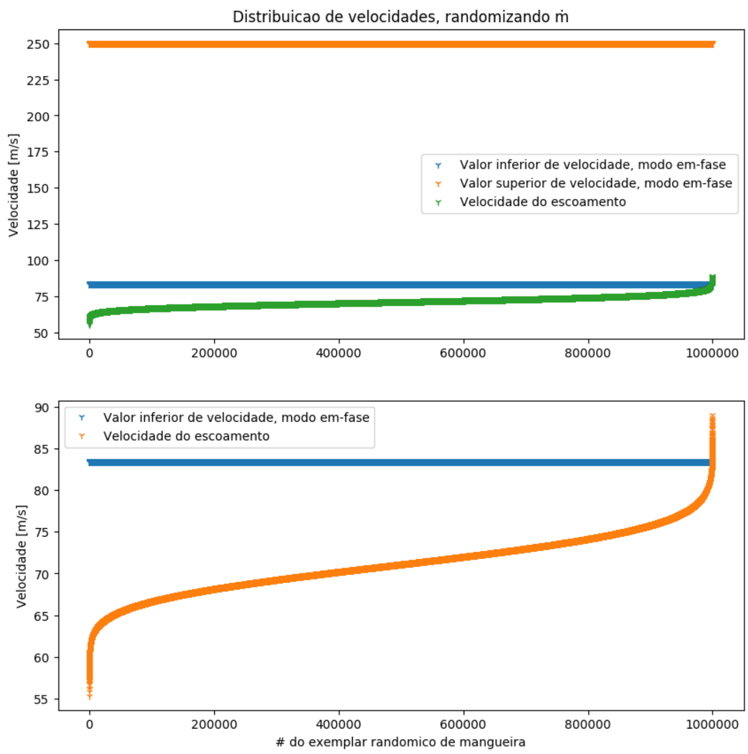

Figura 27 – Distribuição de velocidades na simulação 07

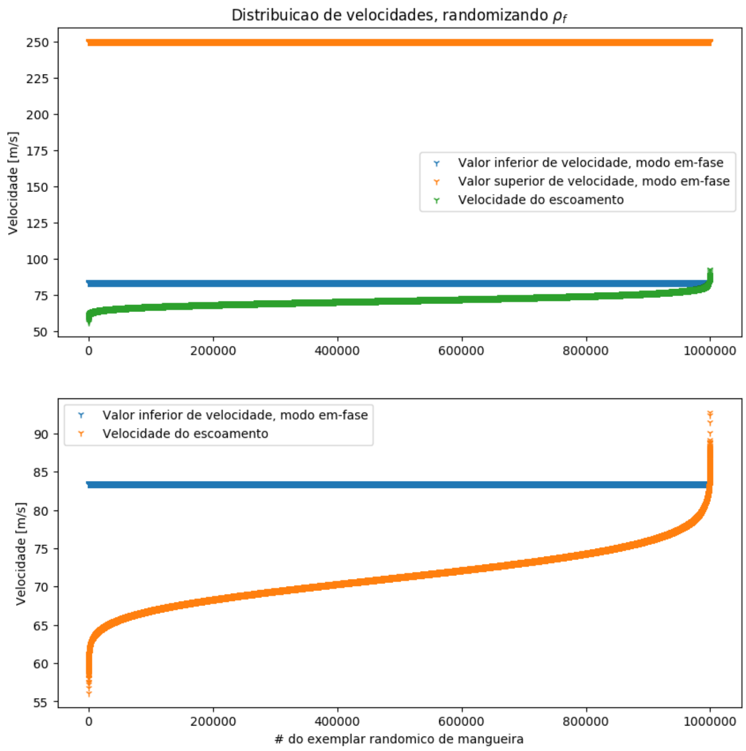

Figura 28 – Distribuição de velocidades na simulação 08

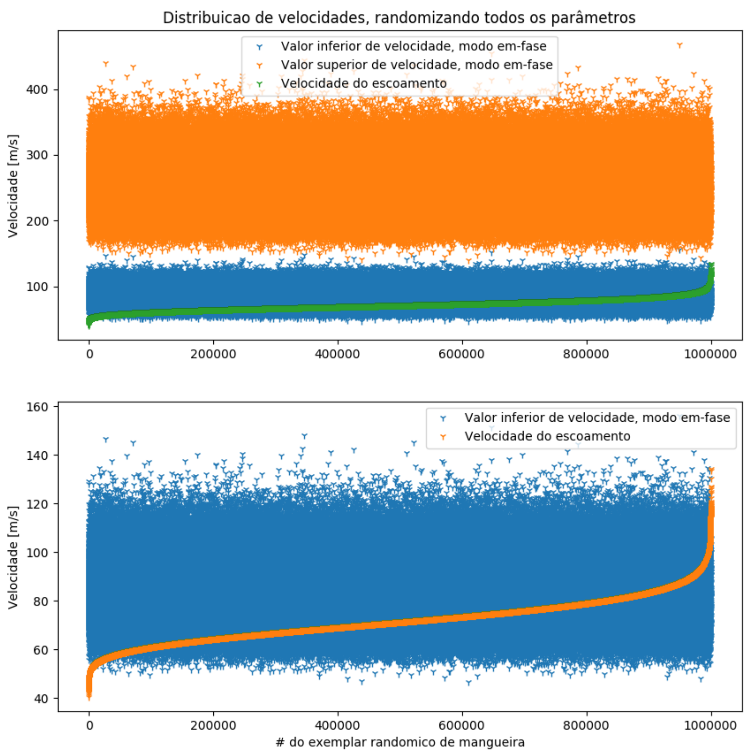

Figura 29 – Distribuição de velocidades na simulação 09

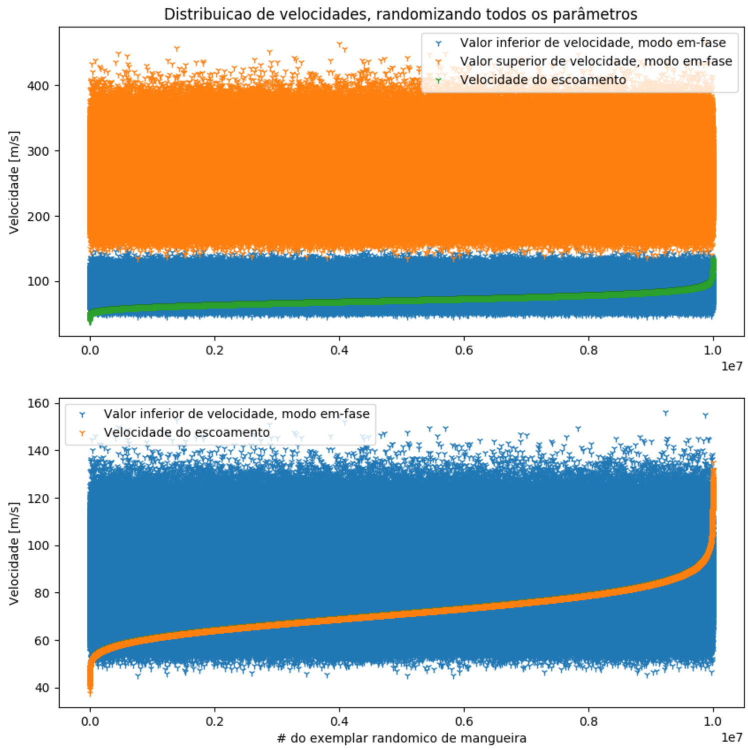

Figura 30 – Distribuição de velocidades na simulação 10

A tab. 7 lista o resultado das simulações de Monte Carlo, contendo a porcentagem de mangueiras com acoplamento para cada um dos casos simulados.

Tabela 7 – Resultado simulações de Monte Carlo

Simulação	**Parâmetro Variado**	**$N°$ de mangueiras com acoplamento**	**Total de exemplares**	**Porcentagem de mangueiras com acoplamento (%)**
01	σ	310	1000000	0.03
02	h	46258	1000000	4.63
03	t	1522	1000000	0.15
04	D_i	61557	1000000	6.16
05	E	0	1000000	0.00
06	ρ_m	0	1000000	0.00
07	$\dot{m}$	561	1000000	0.06
08	ρ_f	1063	1000000	0.11
09	todos	186109	1000000	18.61
10	todos	1854099	10000000	18.54

A simulação 10 foi realizada para garantir que o mesmo numero de exemplares de mangueira flexível variando um ou todos os parâmetros é suficiente para fixar uma porcentagem de acoplamento. Foi observado que mesmo com dez vezes mais exemplares a porcentagem se manteve semelhante.

A partir das simulações de Monte Carlo foi possivel verificar que mesmo operando nominalmente, fora do regime vibratório, mangueiras flexíveis estão suscetíveis ao acoplamento fluido-estrutural devido a incerteza de parâmetros construtivos e operacionais das mesmas.

Em corroboração com a análise de sensibilidade, a aleatoriedade dos parâmetros h e D_i proporcionaram uma maior porcentagem de acoplamento em relação aos outros parâmetros.

4.4 Distribuições de entrada e saída

Os histogramas dos valores de entrada com maior sensibilidade no método (h e D_i) e suas saídas estão nas fig. 31 e 32. Para h é analisado a distribuição gerada na velocidade inferior (V_i) para o modo em fase, já para D_i é analisada a distribuição gerada na velocidade do escoamento (V_e).

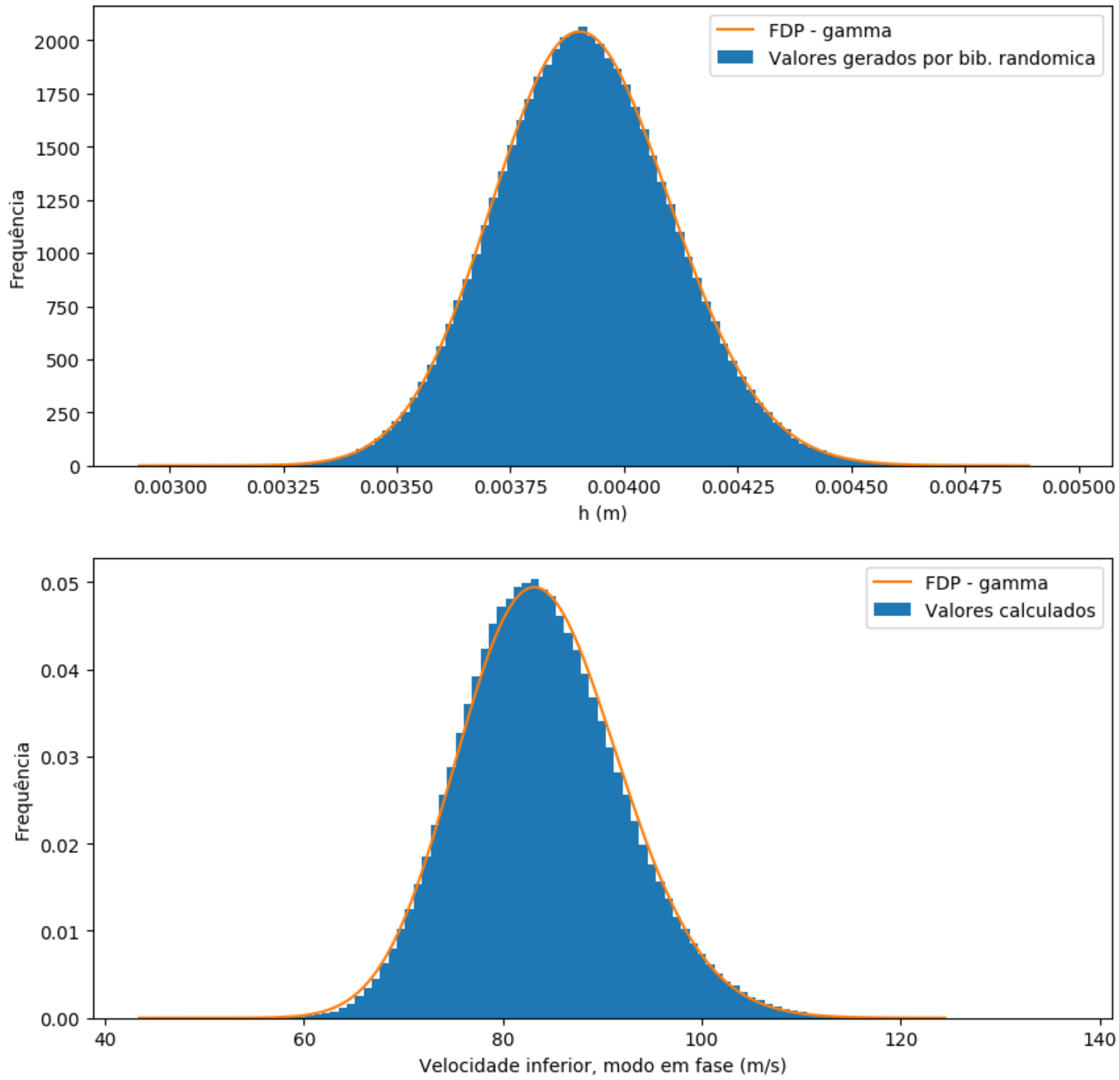

Figura 31 – Histogramas - parâmetro h

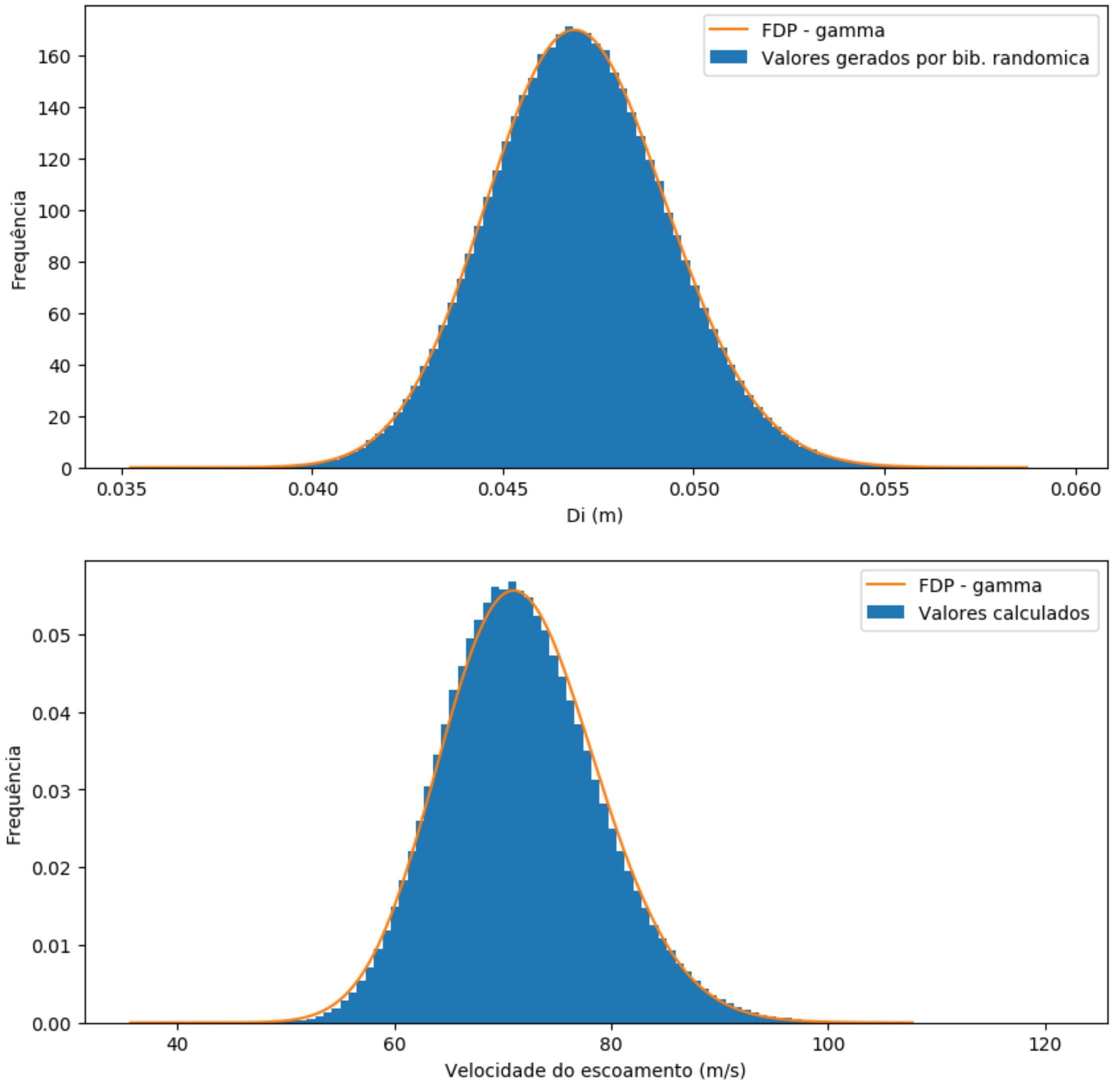

Figura 32 – Histogramas - parâmetro Di

Obeserva-se nos histogramas que as distribuições de saída (V_i para h e V_e para D_i) aproximam-se bastante a função de densidade de probabilidade (FDP) gerada usando o valor médio e o desvio padrão das saídas encontradas como parâmetros de entrada. Dessa forma é possivel afirmar que as saídas são ordenadas em uma distribuição gamma.

4.5 Interfêrencia do desvio padrão da entrada

Os resultados obtidos na análise de interferencia do desvio padrão utilizado na randomização dos valores de entrada das simulações está exposto nas figuras 33 e 34.

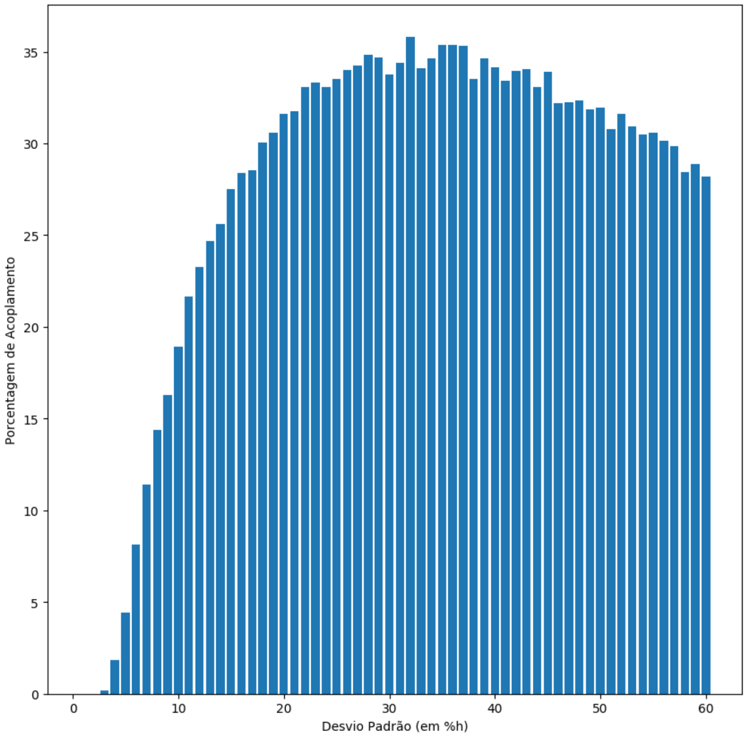

Figura 33 – Relação entre o desvio padrão em h e a porcentagem de acoplamento

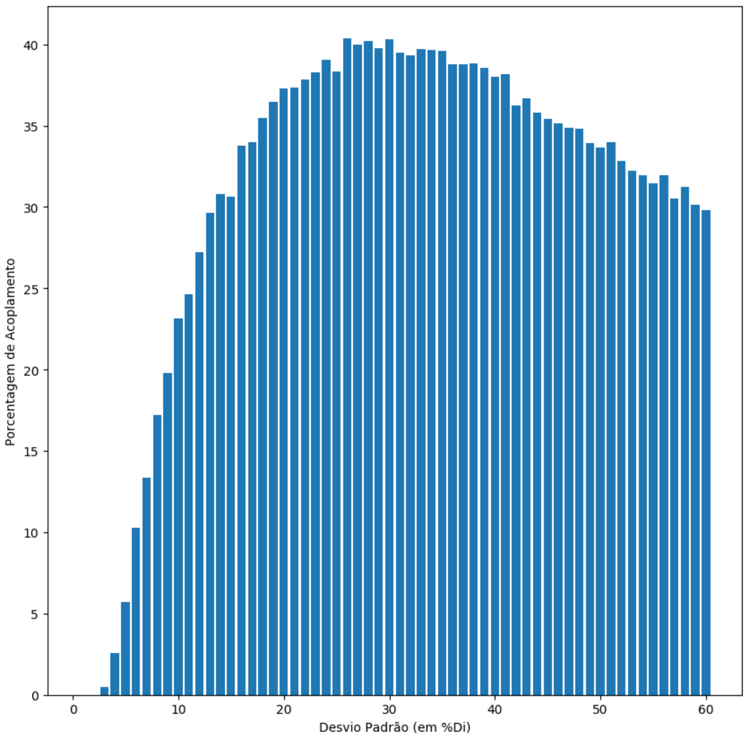

Figura 34 – Relação entre o desvio padrão em D_i e a porcentagem de acoplamento

Observa-se que a porcentagem de acoplamento cresce em relação ao desvio padrão até um valor máximo onde começa a decrescer. Esse fenômeno pode ser melhor analisado ao se observar a distribuição dos valores de saída para o valor de desvio padrão que gera esse máximo (pela fig. 33 aproximadamente 35% do valor de h), como exemplifica a fig. 35.

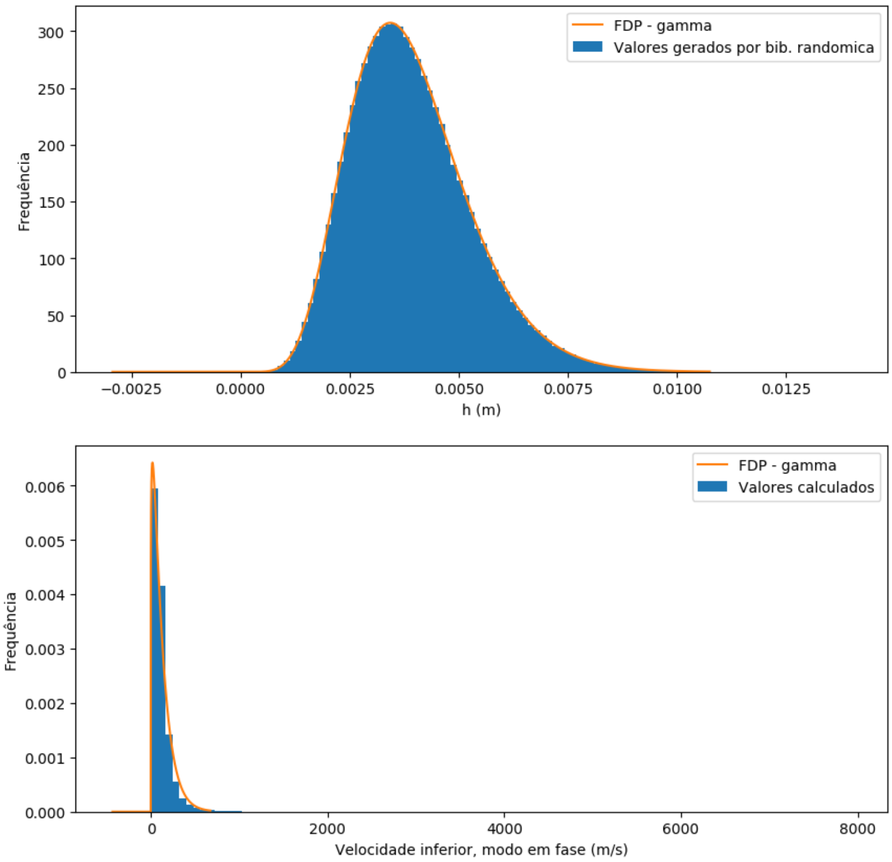

Figura 35 – Histograma variando h usando 35% de h como desvio padrão

Analisando as figuras 33 e 35 é possivel inferir que a porcentagem de acoplamentos começa a decair a partir do momento em que o valor de desvio padrão começa a impor um valor baixo no parâmetro de forma da FDP gamma da distribuição dos valores de saída (V_e e V_i).

5 Conclusão

Esse trabalho de conclusão de curso teve como objetivo a caracterização de incertezas no fenômeno de vibração induzida por escoamento em mangueiras flexíveis para aplicações aeroespaciais. Para tal, foi estudado o método que descreve o fenômeno vibratório, este método foi implementado em linguagem computacional *Python* e devidamente validado. Uma vez validada a implementação do método foi analisada a sensibilidade do mesmo a cada um dos paramentros descritivos e operacionais da mangueira flexível.

Realizou-se também o estudo e implementação do método de simulação de Monte Carlo. Simulações foram rodadas para considerando separadamente cada parâmetro como variaveis aleatórias seguida de uma simulação considerando incertezas em todos os parâmetros da linha flexível.

De maneira a extrair o máximo possivel de informações da análise, foi tambem avaliado o efeito de variação do desvio padrão utilizado na descrição da função de densidade de probabilidade de entrada nas variaveis com maior sensibilidade ao método, bem como qual é a distribuição de probabilidade na saída do sistema.

Todos os códigos desenvolvidos encontram-se nos apêndices do presente trabalho.

Os parâmetros que se mostraram mais sensíveis na validação de linhas flexíveis foram a altura das convoluções (h) com 29.03% de acoplamento no modo fora de fase e o diâmetro interno da mangueira (D_i) com 31.32% de acoplamento para o modo fora de fase, ambos os parâmetros foram variados em mais e menos 20% do valor nominal. Dessa forma tais parâmetros devem receber especial atenção e uma margem de segurança deve ser aplicada nos mesmos para completamente evitar acoplamento fluido estrutural de acordo com MSFC (1992).

As simulações de Monte Carlo de cada parâmetro sendo randomizado isoladamente mostraram baixas porcentagens de acoplamento (6.16% no pior caso), porém ao considerar a incerteza em todos os parâmetros 18.61% dos exemplares entraram em acoplamento, um número significativo. Como as simulações foram realizadas a partir de um exemplo onde nominalmente não há acoplamento o estudo aqui realizado demonstra que para previnir completamente a linha flexível de acoplamento é necessario considerar tambem o efeito das incertezas dos parâmetros.

Possíveis avanços nos estudos realizados na presente monografia podem ser alcançados a partir de analises experimentais do fenômeno, aprimorando o método empírico descrito em MSFC (1992). É possivel também furamente analisar o efeito de incertezas nas estimativas de tensões para cada modo de acoplamento descrito em MSFC (1992).

Referências

BASS III, R. L. et al. *Flow-induced Vibration of Bellows With Internal Cryogenic Fluid Flows*. [S.l.], 1970. Disponível em: <https://ntrs.nasa.gov/search.jsp?R=19710003606>. Citado 2 vezes nas páginas 24 e 25.

CAP, D.; URQUIDI, R.; CANZONA, G. *Nano-grained aluminum alloy bellows*. Google Patents, 2013. US Patent 8,429,894. Disponível em: <http://www.google.com/patents/US8429894>. Citado na página 23.

CUNHA, A. J. Modeling and Quantification of Physical Systems Uncertainties in a Probabilistic Framework. In: ALEMAYEHU, S. E.-O. A. C. G. F. M. (Ed.). *Probabilistic Prognostics and Health Management of Energy Systems*. Springer International Publishing, 2017. p. 127–156. Disponível em: <https://hal.archives-ouvertes.fr/hal-01516295>. Citado na página 38.

DOUGHERTY JR, N. S.; RAFFERTY, C. A. *Altitude Developmental Testing of the J-2 Rocket Engine in Propulsion Engine Test Cell (J-4) (Tests J4-1801-39 through J4-1801-41)*. [S.l.], 1968. Disponível em: <http://www.dtic.mil/get-tr-doc/pdf?AD=AD0848188>. Citado na página 24.

GERLACH, C.; SCHROEDER, E. *Study of Minimum Pressure Loss in High Velocity Duct Systems*. [S.l.], 1969. Disponível em: <https://ntrs.nasa.gov/search.jsp?R=19700016070>. Citado 6 vezes nas páginas 24, 25, 30, 31, 32 e 34.

GERLACH, C. R. Flow-induced vibrations of metal bellows. *Journal of Engineering for Industry*, ASME, v. 91, n. 4, p. 1196–1202, 11 1969. Disponível em: <http://dx.doi.org/10.1115/1.3591771>. Citado 3 vezes nas páginas 24, 25 e 31.

HUZEL, D.; HUANG, D. *Modern Engineering for Design of Liquid-Propellant Rocket Engines*. [S.l.]: American Institute of Aeronautics & Astronautics, 1992. Citado na página 23.

JOHANSEN, A. M.; EVERS, L. *Monte Carlo Methods*. [S.l.]: University of Bristol, 2008. (Lecture Notes of the Department of Mathematics). Citado na página 38.

JOHNSON, J. E. et al. *Bellows Flow-Induced Vibrations*. [S.l.], 1979. Disponível em: <https://ntrs.nasa.gov/search.jsp?R=19810023877>. Citado na página 25.

MONTGOMERY, D.; RUNGER, G. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2010. ISBN 9780470053041. Disponível em: <https://books.google.com.br/books?id=_f4KrEcNAfEC>. Citado na página 38.

NASA MARSHALL SPACE FLIGHT CENTER. *MSFC-DWG-20M02540 - Assessment of Flexible Lines For Flow Induced Vibration*. Hunstsville, Alabama, 1992. 107 p. Citado 9 vezes nas páginas 25, 29, 31, 34, 35, 37, 39, 42 e 65.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. *NASA-STD-5012B - Strenght and Life Assessment Requirements For Liquid-Fueled Space Propulsion System Engines*. [S.l.], 2016. 47 p. Citado na página 25.

PAPADOPOULOS, C. E.; YEUNG, H. Uncertainty estimation and monte carlo simulation method. *Flow Measurement and Instrumentation*, v. 12, n. 4, p. 291 – 298, 2001. ISSN 0955-5986. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0955598601000152>. Citado na página 38.

RIPLEY, B. *Stochastic simulation*. J. Wiley, 1987. (Wiley Series in Probability and Statistics). ISBN 9780471818847. Disponível em: <https://books.google.com.br/books?id=VW1HAAAAMAAJ>. Citado na página 38.

TYGIELSKI, P. J.; SMYLY, H. M.; GERLACH, C. R. *Bellows Flow-Induced Vibrations*. [S.l.], 1983. Disponível em: <https://ntrs.nasa.gov/search.jsp?R=19840014844>. Citado 7 vezes nas páginas 25, 30, 32, 34, 35, 36 e 37.

YORIYAZ, H. Método de monte carlo: princípios e aplicações em física médica. *Revista Brasileira de Física Médica*, p. 141–149, 2009. Citado na página 38.

Apêndices

APÊNDICE A – Valores Nominais Utilizados

A.1 Parâmetros utilizados na validação da implementação em 4.1

$$\begin{aligned}
\text{Comprimento da convolução} &= \sigma = 0.0018288m \\
\text{Passo entre as convoluções} &= \lambda = 0.0026416m \\
\text{Altura das convoluções} &= h = 0.0039116m \\
\text{Espessura da camada} &= t = 0.000254m \\
\text{Número de camadas} &= N_p = 2 \\
\text{Diâmetro interno} &= D_i = 0.04699m \\
\text{Diâmetro externo} &= D_e = Di + 2(h + (Npt))m \\
\text{Módulo de elasticidade} &= E = 1.96e11Pa \\
\text{Densidade do material das convoluções} &= \rho_m = 7805.75kg/m^3 \\
\text{Pressão interna} &= P = 4.137e + 6Pa \\
\text{Temperatura interna} &= T = 297.039K \\
\text{Vazão mássica} &= \dot{m} = 2.63905kg/s \\
\text{Densidade do fluido} &= \rho_f = 6.48kg/m^3 \\
\text{Razão de calores especificos} &= \gamma = (5/3)
\end{aligned}$$

A.2 Valores nominais dos parâmetros utilizados em 4.2, 4.3, 4.4 e 4.5

$$\text{Comprimento da convolução} = \sigma = 0.0018288m$$
$$\text{Passo entre as convoluções} = \lambda = 0.0026416m$$
$$\text{Altura das convoluções} = h = 0.0039116m$$
$$\text{Espessura da camada} = t = 0.000254m$$
$$\text{Número de camadas} = N_p = 2$$
$$\text{Diâmetro interno} = D_i = 0.04699m$$
$$\text{Diâmetro externo} = D_e = Di + 2(h + (Npt))m$$
$$\text{Módulo de elasticidade} = E = 1.96e11Pa$$
$$\text{Densidade do material das convoluções} = \rho_m = 7805.75kg/m^3$$
$$\text{Pressão interna} = P = 4.137e + 6Pa$$
$$\text{Temperatura interna} = T = 297.039K$$
$$\text{Vazão mássica} = \dot{m} = 0.80kg/s$$
$$\text{Densidade do fluido} = \rho_f = 6.48kg/m^3$$
$$\text{Razão de calores especificos} = \gamma = (5/3)$$

APÊNDICE B – flexhose.py

```
from math import pi
from random import gammavariate


class FlexHose:
    # constructor
    def __init__(self, sigma=0.0018288, Lambda=0.0026416, h=0.0039116,
                 t=0.000254, Di=0.04699, De=0, Np=2, Nc=1, E=1.96e11,
                     rho_m=7805.75):
        self.sigma = sigma            # inside convolute width, in meters [m]
        self.Lambda = Lambda          # inside convolute pitch, in meters [m]
        self.h = h                    # convolute height, in meters [m]
        self.t = t                    # ply thickness, in meters [m]
        self.Di = Di                  # internal diameter, in meters [m]
        self.Np = Np                  # number of plys
        self.De = Di + 2*(h+(Np*t))   # external diameter, in meters [m]
        self.Nc = Nc                  # number of convolutions, 1 for FlexHose
        self.E = E                    # Young's module
        self.rho_m = rho_m            # material density, in Kg/m^3

    # return new FlexHose with random input values
    def new_random(self, sd):
        #sd -> standard deviation
        sd_sigma=0.05*self.sigma
        sd_h=sd*self.h
        sd_t=0.05*self.t
        sd_Di=sd*self.Di
        sd_E=0.05*self.E
        sd_rho_m=0.05*self.rho_m
        return FlexHose(
            # sigma = gammavariate((self.sigma/sd_sigma)**2, (sd_sigma**2)/self.sigma),
            # h = gammavariate((self.h/sd_h)**2, (sd_h**2)/self.h),
            # t = gammavariate((self.t/sd_t)**2, (sd_t**2)/self.t),
            Di = gammavariate((self.Di/sd_Di)**2, (sd_Di**2)/self.Di),
            # E = gammavariate((self.E/sd_E)**2, (sd_E**2)/self.E),
            # rho_m = gammavariate((self.rho_m/sd_rho_m)**2, (sd_rho_m**2)/self.rho_m)
        )

    # return new hose for sensibility test
    def new_hose(self, sigma=0.0018288, Lambda=0.0026416, h=0.0039116,
```

```
                t=0.000254, Di=0.04699, De=0.0558292, Np=2, Nc=1, E=1.96e11,
                    rho_m=7805.75):
        return FlexHose(sigma, Lambda, h, t, Di, De, Np, Nc, E, rho_m)

    # mean diameter
    def Dm(self):
        return (self.Di + self.De)/2

    # internal radius of external convolute, in meters [m]
    def a(self):
        return (self.sigma-self.t*self.Np)/2

    # distance between adjacent convolutions
    def delta(self):
        return self.sigma - (2 * (self.t * self.Np))

    # elemental spring rate
    def k(self):
        return 2 * (self.Dm() * self.E * (self.Np/self.Nc) * ((self.t/self.h)**3))

    # material mass
    def m_m(self):
        return pi * self.rho_m * self.t * self.Np * self.Dm() * ((pi * self.a()) + self.h - (2 * self.a()))

    # parameter for acoustic vibration
    def FNCO(self):
        if (self.h/(self.Di/2)) < 0.4 and (self.h/(self.Di/2)) >= 0:
            return 3.8 - (16.72 * ((self.h/(self.Di/2))**2)) + (13.67 * ((self.h/(self.Di/2))**3))
        elif (self.h/(self.Di/2)) >= 0.4 and (self.h/(self.Di/2)) <= 1:
            return -0.336 + 0.935 * ((self.h/(self.Di/2)) ** -1)

    # SSR
    def SSR(self):
        return ((self.k())/2) * self.Nc / (self.Dm() * self.Np)

    # parameter for FIS calc
    def EE(self):
        return 1 + 0.1 * ( 2757902.8/self.SSR()) * (2757902.8 / self.SSR())
```

APÊNDICE C – fluid.py

```
from random import gammavariate

class Fluid:
    # constructor
    def __init__(self, p=4.137e+6, T=297.039, m_dot=0.80, rho_f=6.48, gamma=(5/3)):
        self.p = p             # fluid pressure, in pascal [Pa]
        self.T = T             # fluid temperature, in kelvins [K]
        self.m_dot = m_dot     # mass flow rate, in Kg/s
        self.rho_f = rho_f     # fluid density, in Kg/m^3
        self.gamma = gamma     # Ratio of specific heats

    # speed of sound in the medium
    def S_s(self):
        return (self.gamma * (self.p / self.rho_f))**0.5

    # return new FlexHose with random input values
    def new_random(self):
        #sd -> standard deviation
        sd_m_dot=0.05*self.m_dot
        sd_rho_f=0.05*self.rho_f

        return Fluid(
            # m_dot = gammavariate((self.m_dot/sd_m_dot)**2, (sd_m_dot**2)/self.m_dot),
            # rho_f = gammavariate((self.rho_f/sd_rho_f)**2, (sd_rho_f**2)/self.rho_f)
        )

    # return new fluid for sensibility test
    def new_fluid(self, p=4.137e+6, T=297.039, m_dot=0.80, rho_f=6.48, gamma=(5/3)):
        return Fluid(p, T, m_dot, rho_f, gamma)
```

APÊNDICE D – equations.py

```
from math import pi, sin

class Equations:
    # constructor
    def __init__(self, S_u = 0.3, S_c = 0.2, S_l = 0.1, c1 = 0.13, c2 = 0.462,
                    c3 = 1, c4 = 10, c5 = 0.06, c6 = 1.25, c7 = 5.5, elbow = False, L = 0.034):
        ## velocity calculation Parameters (Strouhal numbers)
        self.S_u = S_u #Strouhal upper limit
        self.S_c = S_c #Strouhal critic value
        self.S_l = S_l #Strouhal lower limit
        ## empirical non-dimensional coefficients for FIS calculation
        self.c1 = c1
        self.c2 = c2
        self.c3 = c3
        self.c4 = c4
        self.c5 = c5
        self.c6 = c6
        self.c7 = c7
        ## definition of equations for flow-induced stress calculation procedure
        self.elbow = elbow  #False for no elbow present upstream, True is elbow present upstream
        self.L = L          #distance from end of elbow to bellow, in meters [m]

    # flow speed in m/s
    def flow_speed(self, f, fh):
        return f.m_dot / f.rho_f / (pi * fh.Di * fh.Di / 4)

    # apparent fluid added mass in convolutions (in-phase)
    def m_IP(self, f, fh):
        return pi * f.rho_f * fh.Dm() * fh.h * ((2 * fh.a())-(fh.t * fh.Np)) * (0.5)

    # apparent fluid added mass in convolutions (out-of-phase)
    def m_OP(self, f, fh):
        return 0.68 * f.rho_f * fh.Dm() * (fh.h**3) * (1/fh.delta())

    # vibration frequency (in-phase)
    def f_IP(self, f, fh):
        return (1/(2 * pi)) * ((2 * fh.k())/(fh.m_m() + self.m_IP(f, fh)))
```

```
**0.5

    # vibration frequency (out-of-phase)
    def f_OP(self, f, fh):
        return (1/(2 * pi)) * ((2 * fh.k())/(fh.m_m() + self.m_OP(f, fh)))
**0.5

    # vibration frequency (convolute bending mode)
    def f_CB(self, f, fh):
        return (1/(2 * pi)) * ((8 * fh.k())/(fh.m_m() + self.m_OP(f, fh)))
**0.5

    # first radial acoustic mode frequency
    def f_acous(self, f, fh):
        return fh.FNCO() * f.S_s() * (1/(2*pi*(fh.Di/2)))

     # first radial acoustic mode velocity
    def v_acous(self, f, fh):
        return (self.f_acous(f, fh) * fh.sigma) / self.S_c

    # velocity range calculation, outputs a vector in which position [0] is
     lower vel, [1] is critical vel, [2] is upper vel
    def vel_range(self, f, fh, freq):
        v_range = [0,0,0]
        v_range[0] = (freq(f, fh) * fh.sigma) / self.S_u
        v_range[1] = (freq(f, fh) * fh.sigma) / self.S_c
        v_range[2] = (freq(f, fh) * fh.sigma) / self.S_l
        return v_range

    # critical frequency, equal to the f_OP
    def f_c(self, f, fh):
        return self.f_OP(f, fh)

    # critical Velocity, equal to the critical vel in OP
    def vel_c(self, f, fh):
        return self.vel_range(f, fh, self.f_OP)[1]

    # V'IP
    def v_prime_IP(self, f, fh):
        return self.vel_range(f, fh, self.f_IP)[1] / self.vel_c(f, fh)

    # V'OP
    def v_prime_OP(self, f, fh):
        return self.vel_range(f, fh, self.f_OP)[1] / self.vel_c(f, fh)

    # V'CB
    def v_prime_CB(self, f, fh):
```

```
        return self.vel_range(f, fh, self.f_CB)[1] / self.vel_c(f, fh)

    # coefficient for number of plys (in-phase)
    def Cnp_IP(self, f, fh):
        if fh.Np == 1:
            return 1
        elif fh.Np > 1:
            return 1 - ((self.c6 * (fh.sigma / fh.h))/(1 + (self.c7 * (self.v_prime_IP(f, fh)**2))))

    # coefficient for number of plys (out-of-phase)
    def Cnp_OP(self, f, fh):
        if fh.Np == 1:
            return 1
        elif fh.Np > 1:
            return 1 - ((self.c6 * (fh.sigma / fh.h))/(1 + (self.c7 * (self.v_prime_OP(f, fh)**2))))

    # coefficient for number of plys (convolute bending mode)
    def Cnp_CB(self, f, fh):
        if fh.Np == 1:
            return 1
        elif fh.Np > 1:
            return 1 - ((self.c6 * (fh.sigma / fh.h))/(1 + (self.c7 * (self.v_prime_CB(f, fh)**2))))

    # parameter for FIS calc (in-phase)
    def C_star_IP(self, f, fh):
        return (self.c1/(self.c2 + (self.v_prime_IP(f, fh)**2))) + ((self.c3 * abs(sin(pi * self.v_prime_IP(f, fh))))/(self.c4 + (self.v_prime_IP(f, fh)**2))) + self.c5

    # parameter for FIS calc (out-of-phase)
    def C_star_OP(self, f, fh):
        return (self.c1/(self.c2 + (self.v_prime_OP(f, fh)**2))) + ((self.c3 * abs(sin(pi * self.v_prime_OP(f, fh))))/(self.c4 + (self.v_prime_OP(f, fh)**2))) + self.c5

    # parameter for FIS calc (convolute bending)
    def C_star_CB(self):
        return 0.4

    # dynamic pressure (in-phase)
    def Pd_IP(self, f, fh):
        return (f.rho_f * (self.vel_range(f, fh, self.f_IP)[1]**2)) / 2

    # dynamic pressure (out-of-phase)
```

```
    def Pd_OP(self, f, fh):
        return (f.rho_f * (self.vel_range(f, fh, self.f_OP)[1]**2)) / 2

    # dynamic pressure (convolute bending)
    def Pd_CB(self, f, fh):
        return (f.rho_f * (self.vel_range(f, fh, self.f_CB)[1]**2)) / 2

    # parameter for FIS calc (in-phase)
    def DD_IP(self, f, fh):
        return (self.C_star_IP(f, fh) * fh.t * self.Pd_IP(f, fh)) / (self.v_prime_IP(f, fh) * fh.SSR() * fh.delta())

    # parameter for FIS calc (out-of-phase)
    def DD_OP(self, f, fh):
        return (self.C_star_OP(f, fh) * fh.t * self.Pd_OP(f, fh)) / (self.v_prime_OP(f, fh) * fh.SSR() * fh.delta())

    # parameter for FIS calc (convolute bending)
    def DD_CB(self, f, fh):
        return (self.C_star_CB() * fh.t * self.Pd_CB(f, fh)) / (self.v_prime_CB(f, fh) * fh.SSR() * fh.delta())

     # elbow coefficient
    def Ce(self, f, fh):
        if self.elbow is True:
            return 1 + (4.7 / (2 + (self.L / fh.Di)))
        else:
            return 1

    # flow-induced stress (in-phase)
    def FIS_IP(self, f, fh):
        return fh.EE() * self.DD_IP(f, fh) * fh.E * self.Cnp_IP(f, fh) * self.Ce(f, fh) / fh.Np

    # flow-induced stress (out-of-phase)
    def FIS_OP(self, f, fh):
        return fh.EE() * self.DD_OP(f, fh) * fh.E * self.Cnp_OP(f, fh) * self.Ce(f, fh) / fh.Np

    # flow-induced stress (convolute bending)
    def FIS_CB(self, f, fh):
        return fh.EE() * self.DD_CB(f, fh) * fh.E * self.Cnp_CB(f, fh) * self.Ce(f, fh) / fh.Np

    # uncertainty factor (in-phase)
    def UF_IP(self, f, fh):
        if self.f_IP(f, fh) < self.f_acous(f, fh):
```

```
            return 2.5
        else:
            return 2.5 * 1.5 * 5

    # uncertainty factor (out-of-phase)
    def UF_OP(self, f, fh):
        if self.f_OP(f, fh) < self.f_acous(f, fh):
            return 2.5
        else:
            return 2.5 * 1.5 * 5

    # uncertainty factor (in-phase)
    def UF_CB(self, f, fh):
        if self.f_CB(f, fh) < self.f_acous(f, fh):
            return 2.5
        else:
            return 2.5 * 1.5 * 5

    # corected FISC_IP
    def FISC_IP(self, f, fh):
        return self.FIS_IP(f, fh) * self.UF_IP(f, fh)

    # corected FISC_IP
    def FISC_OP(self, f, fh):
        return self.FIS_OP(f, fh) * self.UF_OP(f, fh)

    # corected FISC_IP
    def FISC_CB(self, f, fh):
        return self.FIS_CB(f, fh) * self.UF_CB(f, fh)
```

APÊNDICE E – main.py

```
import fluid, flexhose, equations

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

print('The flow velocity is: %.2f m/s' %eq.flow_speed(f, fh) + '\n')

## Checking if there is coupling for all modes

coupling_IP = False
coupling_OP = False
coupling_CB = False
coupling_acoustic = False

if eq.vel_range(f, fh, eq.f_IP)[0] <= eq.flow_speed(f, fh) and eq.vel_range
    (f, fh, eq.f_IP)[2] >= eq.flow_speed(f, fh):
    coupling_IP = True
    print('There is frequency coupling for the in-phase mode of vibration!!
     \n')

if eq.vel_range(f, fh, eq.f_OP)[0] <= eq.flow_speed(f, fh) and eq.vel_range
    (f, fh, eq.f_OP)[2] >= eq.flow_speed(f, fh):
    coupling_OP = True
    print('There is frequency coupling for the out-of-phase mode of
    vibration!! \n')

if eq.vel_range(f, fh, eq.f_CB)[0] <= eq.flow_speed(f, fh) and eq.vel_range
    (f, fh, eq.f_CB)[2] >= eq.flow_speed(f, fh):
    coupling_CB = True
    print('There is frequency coupling for the convolute bending mode of
    vibration!! \n')

if eq.v_acous(f, fh) <= eq.flow_speed(f, fh):
    coupling_acoustic = True
    print('There is acoustic resonance phenomenon!! \n')

## Printing the values of velocity range and frequency for all modes

print('The coupling frequency for the in-phase mode of vibration is: \n' +
    \
              'f(IP) = %.2f Hz \n' %eq.f_IP(f, fh) + \
         'The velocity range for the in-phase mode of vibration is: \n' + \
```

```
            'Lower limit (IP)    : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_IP)[0] + \
            'Critical value (IP) : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_IP)[1] + \
            'Upper limit (IP)    : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_IP)[2])

print('The coupling frequency for the out-of-phase mode of vibration is: \n' + \
            'f(OP) = %.2f Hz \n' %eq.f_OP(f, fh) + \
        'The velocity range for the out-of-phase mode of vibration is: \n' + \
            'Lower limit (OP)    : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_OP)[0] + \
            'Critical value (OP) : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_OP)[1] + \
            'Upper limit (OP)    : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_OP)[2])

print('The coupling frequency for the convolute bending mode of vibration is: \n' + \
            'f(CB) = %.2f Hz \n' %eq.f_CB(f, fh) + \
        'The velocity range for the convolute bending mode of vibration is: \n' + \
            'Lower limit (CB)    : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_CB)[0] + \
            'Critical value (CB) : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_CB)[1] + \
            'Upper limit (CB)    : %.2f m/s \n' %eq.vel_range(f, fh, eq.f_CB)[2])

print('The first radial acoustic mode frequency is: \n' + \
            'f%c : %.2f Hz \n' %(955, eq.f_acous(f, fh)) + \
        'The first radial acoustic mode velocity is: \n' + \
            'V%c : %.2f m/s \n' %(955, eq.v_acous(f, fh)))

## Printing the values of stress for all modes

print('The predicted Flow-Induced Stress (FIV) for the in-phase mode of vibration is: \n' + \
            'FIV(IP) = %.2f Pa \n' %eq.FISC_IP(f, fh) + \
      'The predicted Flow-Induced Stress (FIV) for the out-of-phase mode of vibration is: \n' + \
            'FIV(OP) = %.2f Pa \n' %eq.FISC_OP(f, fh) + \
      'The predicted Flow-Induced Stress (FIV) for the convolute bending mode of vibration is: \n' + \
            'FIV(CB) = %.2f Pa \n' %eq.FISC_CB(f, fh))
```

APÊNDICE F – sensibility-test-vel.py

```
import fluid, equations, flexhose
from numpy import linspace
import matplotlib.pyplot as plt

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

#txt for write percentage of coupling
txt = open('sensibility_test_vel_coup.txt', 'w')
txt.write('Analise de sensibilidade - Porcentagens de acoplamento em cada modo\n')
txt.write('----------------------------------------------------------------------------\n')
txt.write('Par. var. | em fase | fora de fase |  flexao local \n')
txt.write('----------------------------------------------------------------------------\n')
#txt for percentual range inposed in vel. inf. (IP) by variation of variables
log_vel_range_ip = open('perc_comp_ip.txt', 'w')
log_vel_range_ip.write('Analise de sensibilidade - Velocidade Inferior modo em fase\n')
log_vel_range_ip.write('------------------------------------------------------------------------------------------------------------------------\n')
log_vel_range_ip.write('Par. var. |  Valor MIN |  Valor Padrao |  Valor MAX |  Porc. Esq. |   Porc. Dir   | Faixa de var. total(%)\n')
log_vel_range_ip.write('------------------------------------------------------------------------------------------------------------------------\n')
#txt for percentual range inposed in vel. inf. (OP) by variation of variables
log_vel_range_op = open('perc_comp_op.txt', 'w')
log_vel_range_op.write('Analise de sensibilidade - Velocidade Inferior modo fora de fase\n')
log_vel_range_op.write('------------------------------------------------------------------------------------------------------------------------\n')
log_vel_range_op.write('Par. var. |  Valor MIN |  Valor Padrao |  Valor MAX |  Porc. Esq. |  Porc. Dir   | Faixa de var. total(%)\n')
log_vel_range_op.write('------------------------------------------------------------------------------------------------------------------------\n')
#txt for percentual range inposed in vel. inf. (CB) by variation of variables
```

```
log_vel_range_cb = open('perc_comp_cb.txt', 'w')
log_vel_range_cb.write('Analise de sensibilidade - Velocidade Inferior modo de flexao local\n')
log_vel_range_cb.write('------------------------------------------------------------------------------------------------------------\n')
log_vel_range_cb.write('Par. var. |  Valor MIN  |  Valor Padrao  |  Valor MAX  |  Porc. Esq. |  Porc. Dir   | Faixa de var. total(%)\n')
log_vel_range_cb.write('------------------------------------------------------------------------------------------------------------\n')

#txt for percentual range imposed in flow speed by variation of variables
log_flow_speed = open('flow_speed_range.txt', 'w')
log_flow_speed.write('Analise de sensibilidade - Velocidade do escoamento\n')
log_flow_speed.write('------------------------------------------------------------------------------------------------------------\n')
log_flow_speed.write('Par. var. |  Valor MIN |  Valor Padrao |  Valor MAX |  Porc. Esq. |  Porc. Dir   | Faixa de var. total(%)\n')
log_flow_speed.write('------------------------------------------------------------------------------------------------------------\n')

sigma_x = linspace(0.8*fh.sigma, 1.2*fh.sigma, num=10000)
Lambda_x = linspace(0.8*fh.Lambda, 1.2*fh.Lambda, num=10000)
h_x = linspace(0.8*fh.h, 1.2*fh.h, num=10000)
t_x = linspace(0.8*fh.t, 1.2*fh.t, num=10000)
Di_x = linspace(0.8*fh.Di, 1.2*fh.Di, num=10000)
E_x = linspace(0.8*fh.E, 1.2*fh.E, num=10000)
rho_m_x = linspace(0.8*fh.rho_m, 1.2*fh.rho_m, num=10000)
m_dot_x = linspace(0.8*f.m_dot, 1.2*f.m_dot, num=10000)
rho_f_x = linspace(0.8*f.rho_f, 1.2*f.rho_f, num=10000)

# ####################################################################################
#         SIGMA

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
```

```python
hoses_coupling_cb = 0

for i in sigma_x:
    new_hose = fh.new_hose(sigma=i, Lambda=fh.Lambda, h=fh.h, t=fh.t, Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=f.rho_f, gamma=f.gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_hose.sigma)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(sigma_x)) * 100
perc_op = (hoses_coupling_op / len(sigma_x)) * 100
perc_cb = (hoses_coupling_cb / len(sigma_x)) * 100

log_vel_range_ip.write('Sigma      |     %.2f    |      %.2f       |     %.2f    |     %.2f      |       %.2f       |   %.2f\n' %(yip[0][0], eq.vel_range(f, fh, eq.f_IP)[0], yip[9999][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]) + 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('Sigma      |     %.2f    |      %.2f       |     %.2f    |     %.2f      |       %.2f      |  %.2f\n' %(yop[0][0], eq.vel_range(f, fh, eq.
```

```
f_OP)[0], yop[9999][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]) + 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('Sigma    |    %.2f   |     %.2f     |    %.2f   |    %.2f    |     %.2f    |  %.2f\n' %(ycb[0][0], eq.vel_range(f, fh, eq.f_CB)[0], ycb[9999][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]) + 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('Sigma    |    %.2f   |     %.2f     |   %.2f    |    %.2f    |     %.2f     |  %.2f\n' %(y_flow_speed[0], eq.flow_speed(f, fh), y_flow_speed[9999], 100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)), 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)) + 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('Sigma    |    %.2f  |     %.2f     |   %.2f    \n' %(perc_ip, perc_op, perc_cb))

plt.figure(1, figsize=(6, 15))
plt.subplot(311)
plt.plot(sigma_x, [i[0] for i in yip])
plt.plot(sigma_x, [i[1] for i in yip])
plt.plot(sigma_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u03C3 (m)')
plt.title('Distribuicao de velocidades, \u03C3 variando em +/- 20%')

plt.subplot(312)
plt.plot(sigma_x, [i[0] for i in yop])
plt.plot(sigma_x, [i[1] for i in yop])
plt.plot(sigma_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
```

```
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u03C3 (m)')
plt.title('Distribuicao de velocidades, \u03C3 variando em +/- 20%')

plt.subplot(313)
plt.plot(sigma_x, [i[0] for i in ycb])
plt.plot(sigma_x, [i[1] for i in ycb])
plt.plot(sigma_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor
    superior de velocidade, modo de flexao local', 'Velocidade do escoamento
    '])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u03C3 (m)')
plt.title('Distribuicao de velocidades, \u03C3 variando em +/- 20%')
plt.savefig('texto/figuras/graphs/sensibility_test/sigma.png',bbox_inches='
    tight')


#
    ##############################################################################

#        h

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0

for i in h_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=i, t=fh.t,
    Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=f.rho_f,
    gamma=f.gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

```

```
    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_hose.h)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(h_x)) * 100
perc_op = (hoses_coupling_op / len(h_x)) * 100
perc_cb = (hoses_coupling_cb / len(h_x)) * 100

log_vel_range_ip.write('h        |    %.2f    |     %.2f      |   %.2f    |    %.2f     |      %.2f      |  %.2f\n' %(yip[9999][0], eq.vel_range(f, fh, eq.f_IP)[0], yip[0][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[9999][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[0][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[9999][0])/eq.vel_range(f, fh, eq.f_IP)[0]) + 100*((yip[0][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('h        |    %.2f    |     %.2f      |    %.2f   |    %.2f     |      %.2f     |  %.2f\n' %(yop[9999][0], eq.vel_range(f, fh, eq.f_OP)[0], yop[0][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[9999][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[0][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[9999][0])/eq.vel_range(f, fh, eq.f_OP)[0]) + 100*((yop[0][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('h        |    %.2f    |     %.2f      |    %.2f    |     %.2f     |      %.2f     |  %.2f\n' %(ycb[9999][0], eq.vel_range(f, fh, eq.f_CB)[0], ycb[0][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[9999][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[0][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[9999][0])/eq.vel_range(f, fh, eq.f_CB)[0]) + 100*((ycb[0][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0])) ))

```

```
log_flow_speed.write('h          |      %.2f    |       %.2f       |      %.2f    |      %.2f      |       %.2f       |   %.2f\n' %(y_flow_speed[0], eq.flow_speed(f, fh), y_flow_speed[9999], 100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)), 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)) + 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('h          |      %.2f   |       %.2f       |    %.2f      \n' %(perc_ip, perc_op, perc_cb))

plt.figure(3, figsize=(6, 15))
plt.subplot(311)
plt.plot(h_x, [i[0] for i in yip])
plt.plot(h_x, [i[1] for i in yip])
plt.plot(h_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('h (m)')
plt.title('Distribuicao de velocidades, h variando em +/- 20%')

plt.subplot(312)
plt.plot(h_x, [i[0] for i in yop])
plt.plot(h_x, [i[1] for i in yop])
plt.plot(h_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('h (m)')
plt.title('Distribuicao de velocidades, h variando em +/- 20%')

plt.subplot(313)
plt.plot(h_x, [i[0] for i in ycb])
plt.plot(h_x, [i[1] for i in ycb])
plt.plot(h_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor superior de velocidade, modo de flexao local', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('h (m)')
plt.title('Distribuicao de velocidades, h variando em +/- 20%')

```

```
plt.savefig('texto/figuras/graphs/sensibility_test/h.png',bbox_inches='
    tight')

#
    ###############################################################################

#         t

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0

for i in t_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=fh.h, t=i,
    Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=f.rho_f,
    gamma=f.gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_hose.t)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

```

```
perc_ip = (hoses_coupling_ip / len(t_x)) * 100
perc_op = (hoses_coupling_op / len(t_x)) * 100
perc_cb = (hoses_coupling_cb / len(t_x)) * 100

log_vel_range_ip.write('t | %.2f | %.2f | %.2f | %.2f | %.2f | %.2f\n' %(yip[0][0], eq.vel_range(f, fh, eq.f_IP)[0], yip[9999][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]) + 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('t | %.2f | %.2f | %.2f | %.2f | %.2f | %.2f\n' %(yop[0][0], eq.vel_range(f, fh, eq.f_OP)[0], yop[9999][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]) + 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('t | %.2f | %.2f | %.2f | %.2f | %.2f | %.2f\n' %(ycb[0][0], eq.vel_range(f, fh, eq.f_CB)[0], ycb[9999][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]) + 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('t | %.2f | %.2f | %.2f | %.2f | %.2f | %.2f\n' %(y_flow_speed[0], eq.flow_speed(f, fh), y_flow_speed[9999], 100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)), 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)) + 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('t | %.2f | %.2f | %.2f \n' %(perc_ip, perc_op, perc_cb))

plt.figure(4, figsize=(6, 15))
plt.subplot(311)
plt.plot(t_x, [i[0] for i in yip])
plt.plot(t_x, [i[1] for i in yip])
```

```
plt.plot(t_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('t (m)')
plt.title('Distribuicao de velocidades, t variando em +/- 20%')

plt.subplot(312)
plt.plot(t_x, [i[0] for i in yop])
plt.plot(t_x, [i[1] for i in yop])
plt.plot(t_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('t (m)')
plt.title('Distribuicao de velocidades, t variando em +/- 20%')

plt.subplot(313)
plt.plot(t_x, [i[0] for i in ycb])
plt.plot(t_x, [i[1] for i in ycb])
plt.plot(t_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor superior de velocidade, modo de flexao local', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('t (m)')
plt.title('Distribuicao de velocidades, t variando em +/- 20%')

plt.savefig('texto/figuras/graphs/sensibility_test/t.png',bbox_inches='tight')

# ##########################################################################################

#          Di

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0
```

```
for i in Di_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=fh.h, t=fh.t
    , Di=i, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=f.rho_f,
    gamma=f.gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_hose.Di)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(Di_x)) * 100
perc_op = (hoses_coupling_op / len(Di_x)) * 100
perc_cb = (hoses_coupling_cb / len(Di_x)) * 100

log_vel_range_ip.write('Di        |    %.2f    |      %.2f      |     %.2f    |
      %.2f       |      %.2f      |  %.2f\n' %(yip[0][0], eq.vel_range(f, fh,
    eq.f_IP)[0], yip[9999][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip
    [0][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[9999][0]-eq.
    vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((
    eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)
    [0]) + 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(
    f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('Di        |    %.2f    |      %.2f      |     %.2f    |
      %.2f       |      %.2f      |  %.2f\n' %(yop[0][0], eq.vel_range(f, fh,
    eq.f_OP)[0], yop[9999][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop
```

```
    [0][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[9999][0]-eq.
    vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((
    eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)
    [0]) + 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(
    f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('Di      |   %.2f  |    %.2f    |   %.2f  |
       %.2f     |     %.2f     |  %.2f\n' %(ycb[0][0], eq.vel_range(f, fh,
    eq.f_CB)[0], ycb[9999][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb
    [0][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[9999][0]-eq.
    vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((
    eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)
    [0]) + 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(
    f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('Di      |   %.2f  |    %.2f    |   %.2f  |
    %.2f    |     %.2f    |  %.2f\n' %(y_flow_speed[9999], eq.flow_speed(f,
    fh), y_flow_speed[0], 100*((eq.flow_speed(f, fh)-y_flow_speed[9999])/eq.
    flow_speed(f, fh)), 100*((y_flow_speed[0]-eq.flow_speed(f, fh))/eq.
    flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[9999])/eq.
    flow_speed(f, fh)) + 100*((y_flow_speed[0]-eq.flow_speed(f, fh))/eq.
    flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('Di      |   %.2f  |    %.2f    |   %.2f    \n' %(perc_ip,
    perc_op, perc_cb))

plt.figure(5, figsize=(6, 15))
plt.subplot(311)
plt.plot(Di_x, [i[0] for i in yip])
plt.plot(Di_x, [i[1] for i in yip])
plt.plot(Di_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior
    de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('Di (m)')
plt.title('Distribuicao de velocidades, Di variando em +/- 20%')

plt.subplot(312)
plt.plot(Di_x, [i[0] for i in yop])
plt.plot(Di_x, [i[1] for i in yop])
plt.plot(Di_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor
    superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
```

```
plt.xlabel('Di (m)')
plt.title('Distribuicao de velocidades, Di variando em +/- 20%')

plt.subplot(313)
plt.plot(Di_x, [i[0] for i in ycb])
plt.plot(Di_x, [i[1] for i in ycb])
plt.plot(Di_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor superior de velocidade, modo de flexao local', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('Di (m)')
plt.title('Distribuicao de velocidades, Di variando em +/- 20%')

plt.savefig('texto/figuras/graphs/sensibility_test/Di.png',bbox_inches='tight')

# ##########################################################################################
# E

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0

for i in E_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=fh.h, t=fh.t, Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=i, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=f.rho_f, gamma=f.gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
```

```
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_hose.E)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(E_x)) * 100
perc_op = (hoses_coupling_op / len(E_x)) * 100
perc_cb = (hoses_coupling_cb / len(E_x)) * 100

log_vel_range_ip.write('E          |     %.2f    |      %.2f       |     %.2f    |     %.2f     |      %.2f       |  %.2f\n' %(yip[0][0], eq.vel_range(f, fh, eq.f_IP)[0], yip[9999][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]) + 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('E          |     %.2f    |      %.2f       |     %.2f    |     %.2f     |      %.2f      |  %.2f\n' %(yop[0][0], eq.vel_range(f, fh, eq.f_OP)[0], yop[9999][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]) + 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('E          |     %.2f    |      %.2f       |     %.2f    |     %.2f     |      %.2f      |  %.2f\n' %(ycb[0][0], eq.vel_range(f, fh, eq.f_CB)[0], ycb[9999][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]) + 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('E          |     %.2f    |      %.2f       |     %.2f    |
```

```
    %.2f     |      %.2f      |   %.2f\n' %(y_flow_speed[0], eq.flow_speed(f, fh), y_flow_speed[9999], 100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)), 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)) + 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('E          |     %.2f   |      %.2f      |    %.2f     \n' %(perc_ip, perc_op, perc_cb))

plt.figure(6, figsize=(6, 15))
plt.subplot(311)
plt.plot(E_x, [i[0] for i in yip])
plt.plot(E_x, [i[1] for i in yip])
plt.plot(E_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('E (Pa)')
plt.title('Distribuicao de velocidades, E variando em +/- 20%')

plt.subplot(312)
plt.plot(E_x, [i[0] for i in yop])
plt.plot(E_x, [i[1] for i in yop])
plt.plot(E_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('E (Pa)')
plt.title('Distribuicao de velocidades, E variando em +/- 20%')

plt.subplot(313)
plt.plot(E_x, [i[0] for i in ycb])
plt.plot(E_x, [i[1] for i in ycb])
plt.plot(E_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor superior de velocidade, modo de flexao local', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('E (Pa)')
plt.title('Distribuicao de velocidades, E variando em +/- 20%')

plt.savefig('texto/figuras/graphs/sensibility_test/E.png',bbox_inches='
```

```
    tight')

#
    ##############################################################################

#        rho_m

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0

for i in rho_m_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=fh.h, t=fh.t
    , Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=i)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=f.rho_f,
    gamma=f.gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_hose.rho_m)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(rho_m_x)) * 100
```

```
perc_op = (hoses_coupling_op / len(rho_m_x)) * 100
perc_cb = (hoses_coupling_cb / len(rho_m_x)) * 100

log_vel_range_ip.write('rho_m      |     %.2f    |      %.2f       |     %.2f    |      %.2f       |       %.2f       |   %.2f\n' %(yip[9999][0], eq.vel_range(f, fh, eq.f_IP)[0], yip[0][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[9999][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[0][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[9999][0])/eq.vel_range(f, fh, eq.f_IP)[0]) + 100*((yip[0][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('rho_m      |     %.2f    |      %.2f       |     %.2f    |      %.2f       |       %.2f       |   %.2f\n' %(yop[9999][0], eq.vel_range(f, fh, eq.f_OP)[0], yop[0][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[9999][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[0][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[9999][0])/eq.vel_range(f, fh, eq.f_OP)[0]) + 100*((yop[0][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('rho_m      |     %.2f    |      %.2f       |     %.2f    |      %.2f       |       %.2f       |   %.2f\n' %(ycb[9999][0], eq.vel_range(f, fh, eq.f_CB)[0], ycb[0][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[9999][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[0][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[9999][0])/eq.vel_range(f, fh, eq.f_CB)[0]) + 100*((ycb[0][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('rho_m      |     %.2f    |      %.2f       |     %.2f    |      %.2f      |       %.2f       |   %.2f\n' %(y_flow_speed[0], eq.flow_speed(f, fh), y_flow_speed[9999], 100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)), 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.flow_speed(f, fh)) + 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('rho_m      |     %.2f   |      %.2f       |    %.2f     \n' %(perc_ip, perc_op, perc_cb))

plt.figure(7, figsize=(6, 15))
plt.subplot(311)
plt.plot(rho_m_x, [i[0] for i in yip])
plt.plot(rho_m_x, [i[1] for i in yip])
plt.plot(rho_m_x, y_flow_speed)
```

```

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior
    de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u03C1\u2098 (kg/m\u00B3)')
plt.title('Distribuicao de velocidades, \u03C1\u2098 variando em +/- 20%')

plt.subplot(312)
plt.plot(rho_m_x, [i[0] for i in yop])
plt.plot(rho_m_x, [i[1] for i in yop])
plt.plot(rho_m_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor
    superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u03C1\u2098 (kg/m\u00B3)')
plt.title('Distribuicao de velocidades, \u03C1\u2098 variando em +/- 20%')

plt.subplot(313)
plt.plot(rho_m_x, [i[0] for i in ycb])
plt.plot(rho_m_x, [i[1] for i in ycb])
plt.plot(rho_m_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor
    superior de velocidade, modo de flexao local', 'Velocidade do escoamento
    '])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u03C1\u2098 (kg/m\u00B3)')
plt.title('Distribuicao de velocidades, \u03C1\u2098 variando em +/- 20%')

plt.savefig('texto/figuras/graphs/sensibility_test/rho_m.png',bbox_inches='
    tight')

#
    ##################################################################################

#         m_dot

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0

```

```
for i in m_dot_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=fh.h, t=fh.t
    , Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=i, rho_f=f.rho_f, gamma=f.
    gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_fluid.m_dot)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(m_dot_x)) * 100
perc_op = (hoses_coupling_op / len(m_dot_x)) * 100
perc_cb = (hoses_coupling_cb / len(m_dot_x)) * 100

log_vel_range_ip.write('m_dot     |    %.2f    |      %.2f      |     %.2f    |
      %.2f       |       %.2f       |   %.2f\n' %(yip[0][0], eq.vel_range(f, fh,
    eq.f_IP)[0], yip[9999][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip
    [0][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[9999][0]-eq.
    vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((
    eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)
    [0]) + 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(
    f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('m_dot     |    %.2f    |      %.2f      |     %.2f    |
      %.2f       |       %.2f       |   %.2f\n' %(yop[0][0], eq.vel_range(f, fh,
    eq.f_OP)[0], yop[9999][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop
    [0][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[9999][0]-eq.
```

```
    vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((
    eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)
    [0]) + 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(
    f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('m_dot     |     %.2f    |       %.2f       |     %.2f    |
       %.2f       |       %.2f       |   %.2f\n' %(ycb[0][0], eq.vel_range(f, fh,
    eq.f_CB)[0], ycb[9999][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb
    [0][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[9999][0]-eq.
    vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((
    eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)
    [0]) + 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(
    f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('m_dot     |     %.2f    |       %.2f       |     %.2f    |
      %.2f       |       %.2f       |   %.2f\n' %(y_flow_speed[0], eq.flow_speed(f, fh
    ), y_flow_speed[9999], 100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.
    flow_speed(f, fh)), 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.
    flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[0])/eq.
    flow_speed(f, fh)) + 100*((y_flow_speed[9999]-eq.flow_speed(f, fh))/eq.
    flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('m_dot     |     %.2f   |       %.2f       |    %.2f     \n' %(perc_ip,
    perc_op, perc_cb))

plt.figure(8, figsize=(6, 15))
plt.subplot(311)
plt.plot(m_dot_x, [i[0] for i in yip])
plt.plot(m_dot_x, [i[1] for i in yip])
plt.plot(m_dot_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior
    de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u1E41 (kg/s)')
plt.title('Distribuicao de velocidades, \u1E41 variando em +/- 20%')

plt.subplot(312)
plt.plot(m_dot_x, [i[0] for i in yop])
plt.plot(m_dot_x, [i[1] for i in yop])
plt.plot(m_dot_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor
    superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u1E41 (kg/s)')
```

```
plt.title('Distribuicao de velocidades, \u1E41 variando em +/- 20%')

plt.subplot(313)
plt.plot(m_dot_x, [i[0] for i in ycb])
plt.plot(m_dot_x, [i[1] for i in ycb])
plt.plot(m_dot_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor
    superior de velocidade, modo de flexao local', 'Velocidade do escoamento
    '])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('\u1E41 (kg/s)')
plt.title('Distribuicao de velocidades, \u1E41 variando em +/- 20%')

plt.savefig('texto/figuras/graphs/sensibility_test/m_dot.png',bbox_inches='
    tight')

#
    ##########################################################################################

#        rho_f

y1 = [] #to check if value are changing
yip = [] #will import value of freq. of the in-phase mode
yop = [] #will import value of freq. of the out-of-phase mode
ycb = [] #will import value of freq. of the conv. bending mode
y_flow_speed = [] #will import values of flow speed
hoses_coupling_ip = 0
hoses_coupling_op = 0
hoses_coupling_cb = 0

for i in rho_f_x:
    new_hose = fh.new_hose(sigma=fh.sigma, Lambda=fh.Lambda, h=fh.h, t=fh.t
    , Di=fh.Di, De=fh.De, Np=fh.Np, Nc=fh.Nc, E=fh.E, rho_m=fh.rho_m)
    new_fluid = f.new_fluid(p=f.p, T=f.T, m_dot=f.m_dot, rho_f=i, gamma=f.
    gamma)

    vell_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    vell_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[0]
    velu_op = eq.vel_range(new_fluid, new_hose, eq.f_OP)[2]
    vell_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[0]
    velu_cb = eq.vel_range(new_fluid, new_hose, eq.f_CB)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)

    #assessment of hose coupling for the in-phase mode
    if vell_ip < flow_speed and flow_speed < velu_ip:
```

```
        hoses_coupling_ip += 1
    #assessment of hose coupling for the out-of-phase-phase mode
    if vell_op < flow_speed and flow_speed < velu_op:
        hoses_coupling_op += 1
    #assessment of hose coupling for the conv. bending mode
    if vell_cb < flow_speed and flow_speed < velu_cb:
        hoses_coupling_cb += 1

    y1.append(new_fluid.rho_f)
    yip.append([vell_ip, velu_ip])
    yop.append([vell_op, velu_op])
    ycb.append([vell_cb, velu_cb])
    y_flow_speed.append(flow_speed)

perc_ip = (hoses_coupling_ip / len(rho_f_x)) * 100
perc_op = (hoses_coupling_op / len(rho_f_x)) * 100
perc_cb = (hoses_coupling_cb / len(rho_f_x)) * 100

log_vel_range_ip.write('rho_f     |     %.2f    |      %.2f      |     %.2f    |      %.2f      |      %.2f      |  %.2f\n' %(yip[0][0], eq.vel_range(f, fh, eq.f_IP)[0], yip[9999][0], 100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]), 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0]), (100*((eq.vel_range(f, fh, eq.f_IP)[0]-yip[0][0])/eq.vel_range(f, fh, eq.f_IP)[0]) + 100*((yip[9999][0]-eq.vel_range(f, fh, eq.f_IP)[0])/eq.vel_range(f, fh, eq.f_IP)[0])) ))

log_vel_range_op.write('rho_f     |     %.2f    |      %.2f      |     %.2f    |      %.2f      |      %.2f      | %.2f\n' %(yop[0][0], eq.vel_range(f, fh, eq.f_OP)[0], yop[9999][0], 100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]), 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0]), (100*((eq.vel_range(f, fh, eq.f_OP)[0]-yop[0][0])/eq.vel_range(f, fh, eq.f_OP)[0]) + 100*((yop[9999][0]-eq.vel_range(f, fh, eq.f_OP)[0])/eq.vel_range(f, fh, eq.f_OP)[0])) ))

log_vel_range_cb.write('rho_f     |     %.2f    |      %.2f      |     %.2f    |      %.2f      |      %.2f      | %.2f\n' %(ycb[0][0], eq.vel_range(f, fh, eq.f_CB)[0], ycb[9999][0], 100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]), 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0]), (100*((eq.vel_range(f, fh, eq.f_CB)[0]-ycb[0][0])/eq.vel_range(f, fh, eq.f_CB)[0]) + 100*((ycb[9999][0]-eq.vel_range(f, fh, eq.f_CB)[0])/eq.vel_range(f, fh, eq.f_CB)[0])) ))

log_flow_speed.write('rho_f     |     %.2f    |      %.2f      |     %.2f    |      %.2f      |      %.2f      | %.2f\n' %(y_flow_speed[9999], eq.flow_speed(f,
```

```
fh), y_flow_speed[0], 100*((eq.flow_speed(f, fh)-y_flow_speed[9999])/eq.flow_speed(f, fh)), 100*((y_flow_speed[0]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh)), (100*((eq.flow_speed(f, fh)-y_flow_speed[9999])/eq.flow_speed(f, fh)) + 100*((y_flow_speed[0]-eq.flow_speed(f, fh))/eq.flow_speed(f, fh))) ))

#write values of percentage of coupling for variable analysed
txt.write('rho_f     |    %.2f   |     %.2f     |   %.2f    \n' %(perc_ip, perc_op, perc_cb))

plt.figure(9, figsize=(6, 15))
plt.subplot(311)
plt.plot(rho_f_x, [i[0] for i in yip])
plt.plot(rho_f_x, [i[1] for i in yip])
plt.plot(rho_f_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo em fase', 'Valor superior de velocidade, modo em fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('$\u03C1_f$ (kg/m\u00B3)')
plt.title('Distribuicao de velocidades, $\u03C1_f$ variando em +/- 20%')

plt.subplot(312)
plt.plot(rho_f_x, [i[0] for i in yop])
plt.plot(rho_f_x, [i[1] for i in yop])
plt.plot(rho_f_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo fora de fase', 'Valor superior de velocidade, modo fora de fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('$\u03C1_f$ (kg/m\u00B3)')
plt.title('Distribuicao de velocidades, $\u03C1_f$ variando em +/- 20%')

plt.subplot(313)
plt.plot(rho_f_x, [i[0] for i in ycb])
plt.plot(rho_f_x, [i[1] for i in ycb])
plt.plot(rho_f_x, y_flow_speed)

plt.legend(['Valor inferior de velocidade, modo de flexao local', 'Valor superior de velocidade, modo de flexao local', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('$\u03C1_f$ (kg/m\u00B3)')
plt.title('Distribuicao de velocidades, $\u03C1_f$ variando em +/- 20%')

plt.savefig('texto/figuras/graphs/sensibility_test/rho_f.png',bbox_inches='tight')
```

APÊNDICE G – Monte-Carlo-Analysis.py

```
import time
import matplotlib.pyplot as plt
import fluid, equations, flexhose

#starting calculation of computing time
start_time = time.time()

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

# y[0] = flow-speed
# y[1] = lower velocity in-phase random
# y[2] = upper velocity in-phase random

txt = open('Monte-Carlo-Simulation.txt', 'a')
# txt.write('Parametro | qnt acoplamento | total de mangueiras | Porcentagem de acoplamento\n')
txt.write('------------------------------------------------------------------------------------------\n')

y = []
num_hoses_coupling = 0
x = range(10000000) #number of hoses (samples)

#loop for generation of random hoses
for i in x:
    new_hose = fh.new_random()
    new_fluid = f.new_random()
    vell_ip_random = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
    velu_ip_random = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
    flow_speed = eq.flow_speed(new_fluid, new_hose)
    y.append([flow_speed, vell_ip_random, velu_ip_random])
    #assessment of hose coupling for the in-phase mode
    if vell_ip_random < flow_speed and flow_speed < velu_ip_random:
        num_hoses_coupling += 1

perc_coup = (num_hoses_coupling / len(x)) * 100
# print('Number of hoses coupling: ' + str(num_hoses_coupling) + '\n' +
#       'Number of random hoses: ' + str(len(x)))
# print('The porcentage of hoses coupling is: %.2f %%' %perc_coup)

```

```
txt.write('todas10m | '+str(num_hoses_coupling)+' | '+str(len(x))+' | %.2f\
    n' %perc_coup)
y = sorted(y)

plt.figure(1, figsize=(10, 10))
plt.subplot(211)
plt.plot(x, [i[1] for i in y], '1')
plt.plot(x, [i[2] for i in y], '1')
plt.plot(x, [i[0] for i in y], '1')

plt.legend(['Valor inferior de velocidade, modo em-fase', 'Valor superior
    de velocidade, modo em-fase', 'Velocidade do escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.title('Distribuicao de velocidades, randomizando todos os parametros')

plt.subplot(212)
plt.plot(x, [i[1] for i in y], '1')
plt.plot(x, [i[0] for i in y], '1')

plt.legend(['Valor inferior de velocidade, modo em-fase', 'Velocidade do
    escoamento'])
plt.ylabel('Velocidade [m/s]')
plt.xlabel('# do exemplar randomico de mangueira')

elapsed_time = time.time() - start_time
print('Execution time: %.2f s' %elapsed_time)
plt.savefig('texto/figuras/graphs/Monte_Carlo/todas10m.png',bbox_inches='
    tight')
# plt.show()
```

APÊNDICE H – histograma-io-h.py

```
import fluid, equations, flexhose
from numpy import linspace, std, mean
import matplotlib.pyplot as plt
import scipy.stats as stats

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

#standard deviation
sd_h = 0.05*fh.h
#alpha and beta for h value
alpha_h = (fh.h/sd_h)**2
beta_h = (sd_h**2)/fh.h

y_vel = [] #stores values of vell_ip_random
y_h = [] #stores values of h
num=1000000 #number of hoses (samples)
x = range(num) #number of hoses (samples)

#loop for generation of random hoses
for i in x:
    new_hose = fh.new_random(sd_h/fh.h)
    vell_ip_random = eq.vel_range(f, new_hose, eq.f_IP)[0]
    y_h.append(new_hose.h)
    y_vel.append(vell_ip_random)

#finds standard deviation and mean of vell_ip_random values
sd = std(y_vel)
mean = mean(y_vel)

vell_ip_x = linspace(mean-5*sd, mean+5*sd, num=num)
h_x = linspace(fh.h-5*sd_h, fh.h+5*sd_h, num=num)

#alpha and beta for vell_ip_random value
alpha_vel = (mean/sd)**2
beta_vel = (sd**2)/mean

y_vel = sorted(y_vel)
y_h = sorted(y_h)

#calculates pdf for h (in)
y_h_gamma = stats.gamma.pdf(h_x, alpha_h, scale=beta_h)
```

```
# y_h_gauss = stats.norm.pdf(h_x, loc=fh.h, scale=sd_h)
#calculates pdf for velu_ip_random values (out)
y_vel_gamma = stats.gamma.pdf(vell_ip_x, alpha_vel, scale=beta_vel)
# y_vel_gauss = stats.norm.pdf(vell_ip_x, loc=mean, scale=sd)

plt.figure(1, figsize=(10, 10))
#plot histogram for in (h)
plt.subplot(211)
plt.hist(y_h, 100, density=True)
plt.plot(h_x, y_h_gamma)
# plt.plot(h_x, y_h_gauss)

plt.legend(['FDP - gamma', 'Valores gerados por bib. randomica'])
plt.ylabel('Frequ\u00EAncia')
plt.xlabel('h (m)')

#plots histogram for out (vell_ip_random)
plt.subplot(212)
plt.hist(y_vel, 100, density=True)
plt.plot(vell_ip_x, y_vel_gamma)
# plt.plot(vell_ip_x, y_vel_gauss)

plt.legend(['FDP - gamma', 'Valores calculados'])
plt.ylabel('Frequ\u00EAncia')
plt.xlabel('Velocidade inferior, modo em fase (m/s)')

plt.savefig('texto/figuras/graphs/histograma/h.png',bbox_inches='tight')
# plt.show()
```

APÊNDICE I – histograma-io-Di.py

```
import fluid, equations, flexhose
from numpy import linspace, std, mean
import matplotlib.pyplot as plt
import scipy.stats as stats

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

#standard deviation
sd_Di = 0.05*fh.Di
#alpha and beta for h value
alpha_Di = (fh.Di/sd_Di)**2
beta_Di = (sd_Di**2)/fh.Di

y_vel = [] #stores values of flow_speed
y_Di = [] #stores values of Di
num=1000000 #number of hoses (samples)
x = range(num) #number of hoses (samples)

#loop for generation of random hoses
for i in x:
    new_hose = fh.new_random(sd_Di/fh.Di)
    flow_speed = eq.flow_speed(f, new_hose)
    y_Di.append(new_hose.Di)
    y_vel.append(flow_speed)

#finds standard deviation and mean of flow_speed values
sd = std(y_vel)
mean = mean(y_vel)

vell_ip_x = linspace(mean-5*sd, mean+5*sd, num=num)
Di_x = linspace(fh.Di-5*sd_Di, fh.Di+5*sd_Di, num=num)

#alpha and beta for flow_speed value
alpha_vel = (mean/sd)**2
beta_vel = (sd**2)/mean

y_vel = sorted(y_vel)
y_Di = sorted(y_Di)

#calculates pdf for Di (in)
y_Di_gamma = stats.gamma.pdf(Di_x, alpha_Di, scale=beta_Di)
```

```
y_Di_gauss = stats.norm.pdf(Di_x, loc=fh.Di, scale=sd_Di)
#calculates pdf for velu_ip_random values (out)
y_vel_gamma = stats.gamma.pdf(vell_ip_x, alpha_vel, scale=beta_vel)
y_vel_gauss = stats.norm.pdf(vell_ip_x, loc=mean, scale=sd)

plt.figure(1, figsize=(10, 10))
#plot histogram for in (Di)
plt.subplot(211)
plt.hist(y_Di, 100, density=True)
plt.plot(Di_x, y_Di_gamma)
# plt.plot(Di_x, y_Di_gauss)

plt.legend(['FDP - gamma', 'Valores gerados por bib. randomica'])
plt.ylabel('Frequ\u00EAncia')
plt.xlabel('Di (m)')

#plots histogram for out (flow_speed)
plt.subplot(212)
plt.hist(y_vel, 100, density=True)
plt.plot(vell_ip_x, y_vel_gamma)
# plt.plot(vell_ip_x, y_vel_gauss)

plt.legend(['FDP - gamma', 'Valores calculados'])
plt.ylabel('Frequ\u00EAncia')
plt.xlabel('Velocidade do escoamento (m/s)')

plt.savefig('texto/figuras/graphs/histograma/Di.png',bbox_inches='tight')
# plt.show()
```

APÊNDICE J – SD-Analysis-h.py

```
import time
import matplotlib.pyplot as plt
import fluid, equations, flexhose

#starting calculation of computing time
start_time = time.time()

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

# y[0] = flow-speed
# y[1] = lower velocity in-phase random
# y[2] = upper velocity in-phase random

txt = open('Monte-Carlo-Analysis-SD-var-h.txt', 'w')
txt.write('Des. Padrao (%h) | qnt acoplamento | total de mangueiras |
    Porcentagem de acoplamento\n')
txt.write('
    ----------------------------------------------------------------------------------------\
    n')

x = range(10000) #number of hoses (samples)
SD_h = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11,
    0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22, 0.23,
    0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.30, 0.31, 0.32, 0.33, 0.34, 0.35,
    0.36, 0.37, 0.38, 0.39, 0.40, 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47,
    0.48, 0.49, 0.50, 0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59,
    0.60]#, 0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.80,
    0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92,
    0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1]



perc_coup_hoses = [] #array for percentage of hoses for each SD of h
sd = [] #array for SD value (in percentage of h)

for sd_h in SD_h:
    y = []
    num_hoses_coupling = 0

    #loop for generation of random hoses
    for i in x:
```

```
        new_hose = fh.new_random(sd_h)
        new_fluid = f.new_random()
        vell_ip_random = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
        velu_ip_random = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
        flow_speed = eq.flow_speed(new_fluid, new_hose)
        y.append([flow_speed, vell_ip_random, velu_ip_random])
        #assessment of hose coupling for the in-phase mode
        if vell_ip_random < flow_speed and flow_speed < velu_ip_random:
            num_hoses_coupling += 1

    perc_coup = (num_hoses_coupling / len(x)) * 100

    perc_coup_hoses.append(perc_coup)
    sd.append(sd_h*100)

    txt.write('        %.1f           |  '%(sd_h*100)+str(num_hoses_coupling)+' |
    '+str(len(x))+' | %.2f\n' %(perc_coup))
    y = sorted(y)

    plt.figure(1, figsize=(10, 10))
    plt.subplot(211)
    plt.plot(x, [i[1] for i in y], '1')
    plt.plot(x, [i[2] for i in y], '1')
    plt.plot(x, [i[0] for i in y], '1')

    plt.legend(['Valor inferior de velocidade, modo em-fase', 'Valor
   superior de velocidade, modo em-fase', 'Velocidade do escoamento'])
    plt.ylabel('Velocidade [m/s]')
    plt.title('hstribuicao de velocidades, Desvio Padrao de h = %.2f %% de
   h\n' %(sd_h*100))

    plt.subplot(212)
    plt.plot(x, [i[1] for i in y], '1')
    plt.plot(x, [i[0] for i in y], '1')

    plt.legend(['Valor inferior de velocidade, modo em-fase', 'Velocidade
   do escoamento'])
    plt.ylabel('Velocidade [m/s]')
    plt.xlabel('# do exemplar randomico de mangueira')

    elapsed_time = time.time() - start_time
    print('Execution time: %.2f s' %elapsed_time)

    flag = '%s' % str(sd_h*100)
    plt.savefig('texto/figuras/graphs/Monte_Carlo/SD_Analysis/h/sd-%s.png'
   %flag,bbox_inches='tight')
    plt.clf()
```

```

plt.figure(1)
plt.bar(sd, perc_coup_hoses, width=0.8)
plt.xlabel('Desvio Padrao (em %h)')
plt.ylabel('Porcentagem de Acoplamento')
# plt.title('Relacao entre o desvio padrao e a porcentagem de acoplamento
    em h')
plt.savefig('texto/figuras/graphs/Monte_Carlo/SD_Analysis/h/h-SD-Analysis.
    png',bbox_inches='tight')
```

APÊNDICE K – SD-Analysis-Di.py

```
import time
import matplotlib.pyplot as plt
import fluid, equations, flexhose

#starting calculation of computing time
start_time = time.time()

f = fluid.Fluid()
fh = flexhose.FlexHose()
eq = equations.Equations()

# y[0] = flow-speed
# y[1] = lower velocity in-phase random
# y[2] = upper velocity in-phase random

txt = open('Monte-Carlo-Analysis-SD-var-Di.txt', 'w')
txt.write('Des. Padrao (%Di) | qnt acoplamento | total de mangueiras |
    Porcentagem de acoplamento\n')
txt.write('
    -----------------------------------------------------------------------\
    n')

x = range(10000) #number of hoses (samples)
SD_DI = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11,
    0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22, 0.23,
    0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.30, 0.31, 0.32, 0.33, 0.34, 0.35,
    0.36, 0.37, 0.38, 0.39, 0.40, 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47,
    0.48, 0.49, 0.50, 0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59,
    0.60] #0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.80, 0.81,
     0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93,
     0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1]



perc_coup_hoses = [] #array for percentage of hoses for each SD of Di
sd = [] #array for SD value (in percentage of Di)

for sd_di in SD_DI:
    y = []
    num_hoses_coupling = 0

    #loop for generation of random hoses
    for i in x:
```

```
        new_hose = fh.new_random(sd_di)
        new_fluid = f.new_random()
        vell_ip_random = eq.vel_range(new_fluid, new_hose, eq.f_IP)[0]
        velu_ip_random = eq.vel_range(new_fluid, new_hose, eq.f_IP)[2]
        flow_speed = eq.flow_speed(new_fluid, new_hose)
        y.append([flow_speed, vell_ip_random, velu_ip_random])
        #assessment of hose coupling for the in-phase mode
        if vell_ip_random < flow_speed and flow_speed < velu_ip_random:
            num_hoses_coupling += 1

    perc_coup = (num_hoses_coupling / len(x)) * 100

    perc_coup_hoses.append(perc_coup)
    sd.append(sd_di*100)

    txt.write('       %.1f          |  '%(sd_di*100)+str(num_hoses_coupling)+'  |  '+str(len(x))+' | %.2f\n' %(perc_coup))
    y = sorted(y)

    plt.figure(1, figsize=(10, 10))
    plt.subplot(211)
    plt.plot(x, [i[1] for i in y], '1')
    plt.plot(x, [i[2] for i in y], '1')
    plt.plot(x, [i[0] for i in y], '1')

    plt.legend(['Valor inferior de velocidade, modo em-fase', 'Valor superior de velocidade, modo em-fase', 'Velocidade do escoamento'])
    plt.ylabel('Velocidade [m/s]')
    plt.title('Distribuicao de velocidades, Desvio Padrao de Di = %.2f %% de Di\n' %(sd_di*100))

    plt.subplot(212)
    plt.plot(x, [i[1] for i in y], '1')
    plt.plot(x, [i[0] for i in y], '1')

    plt.legend(['Valor inferior de velocidade, modo em-fase', 'Velocidade do escoamento'])
    plt.ylabel('Velocidade [m/s]')
    plt.xlabel('# do exemplar randomico de mangueira')

    elapsed_time = time.time() - start_time
    print('Execution time: %.2f s' %elapsed_time)

    flag = '%s' % str(sd_di*100)
    plt.savefig('texto/figuras/graphs/Monte_Carlo/SD_Analysis/Di/sd-%s.png' %flag ,bbox_inches='tight')
    plt.clf()
```

```
plt.figure(1)
plt.bar(sd, perc_coup_hoses, width=0.8)
plt.xlabel('Desvio Padrao (em %Di)')
plt.ylabel('Porcentagem de Acoplamento')
# plt.title('Relacao entre o desvio padrao e a porcentagem de acoplamento em Di')
plt.savefig('texto/figuras/graphs/Monte_Carlo/SD_Analysis/Di/Di-SD-Analysis.png',bbox_inches='tight')
```