

1 APLICAÇÕES E RESULTADOS

Este capítulo apresenta um exemplo básico de aplicação do *framework*, apresentando as funções principais para geração de modelos (.inp) e para simulação no ABAQUS, chamada do ABAQUS, pós-processamento e visualização dos resultados. Serão apresentados tanto os resultados obtidos com a simulação quanto os códigos básicos utilizados geração desses resultados, bem como trechos de código demonstrando a forma de utilização das principais funcionalidades implementadas.

Embora não haja restrições do *framework* quanto ao perfil da batimetria, este exemplo será um caso de uma batimetria simples, uma vez que o intuito é apresentar modelo cujos resultados (mais especificamente, os valores das frequências obtidas pela análise modal no ABAQUS) podem validades com os resultados previstos (calculados pela FatFree). Vale salientar que os dados utilizados são de domínio público ou fictícios, mas não representam um caso real, até porque esse tipo de dado é mantido sob sigilo devido a aspectos relacionados à segurança, competitividade tecnológica e propriedade intelectual das empresas do setor.

1.1 Exemplo: análise de vida a fadiga de um pequeno trecho de duto

O modelo consiste um pequeno trecho de duto com comprimento total de 1000 m com um vão de 17 m posicionado no centro. A grande extensão do duto a direita e à esquerda do vão é usada simplesmente para reduzir a influência das condições de contorno nas extremidades.

Para iniciar as análises, todos esses dados devem estar devidamente estruturados num arquivo de entrada no formato JSON. Para carregar esses dados e criar uma instância da classe `Model`, o que pode ser feito com a função `load_json`, como mostrado na Figura 1.

Antes de iniciar a simulação, que pode ser um processo custoso, é importante revisar se o modelo está minimamente de acordo com o desejado. Pode-se, por exemplo, visualizar o perfil da batimetria tomando partido de uma das funções gráficas implementadas no módulo `plots`. O código para geração dessa figura está na Figura 2, cujo resultado é mostrado na Figura 3.

A simulação realizada tem a seguinte sequência de passos de carga:

1. Aplica-se o peso do duto vazio;
2. Aplica-se a pressão externa;

Figura 1 – Código para carregamento dos dados de entrada.

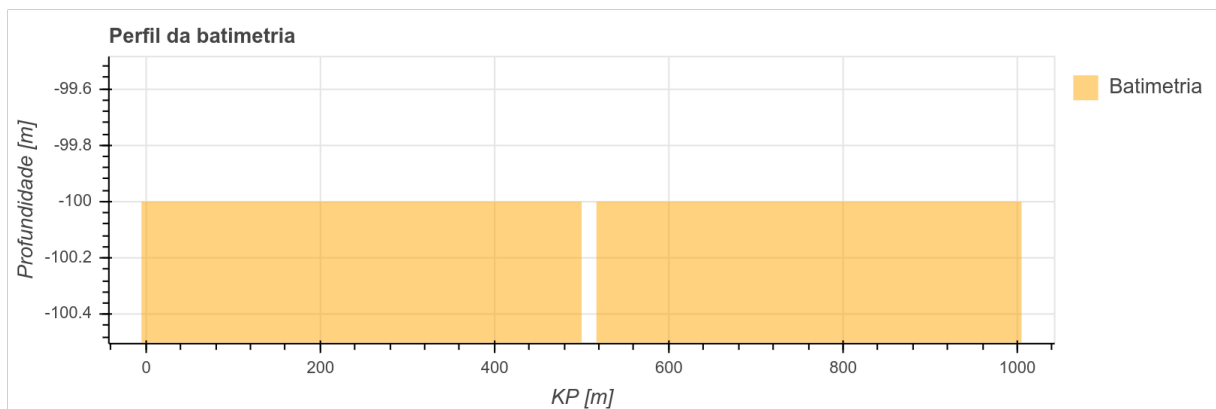
```
1 from integrispan.model_generator.model_generator import load_json
2 model = load_json("inputs.json")
```

Figura 2 – Código para geração de Figura 3 com o perfil da batimetria

```
1 from integrispan.plots import plots
2 bathymetry_plot = plots.bathymetry(model.bat)
3 bathymetry_plot.save("perfil_batimetria.html")
```

Fonte: Autor (2020)

Figura 3 – Perfil do modelo.



Fonte: Autor (2020)

3. Aplica-se a tração de lançamento;
4. Aplica-se o deslocamento vertical e assenta-se o duto;
5. Restaura-se o atrito axial;
6. Ativa-se as molas;
7. Remove a tração de lançamento;
8. Aplica-se a pressão do teste hidrostático;
9. Remove-se a pressão do teste hidrostático;
10. Aplica-se a pressão operacional;
11. Obtêm-se os modos de vibração (análise modal).

A geração dos arquivos para simulação no ABAQUS contém as instruções para toda essa sequência de passo é feita usando o método `write_inps`: `model.write_inps()`. Isso deve gerar dois arquivos dentro de um diretório chamado `exemplo`: o arquivo principal `exemplo.inp`, e `bt_exemplo.inp`, com as coordenadas que define o perfil de batimetria.

O método `run_abaqus` do objeto `model` é responsável por executar a chamada do ABAQUS de maneira programática para iniciar a simulação. Nesse método, ocorre a leitura do arquivo de registro da simulação e o seu conteúdo é exibido na tela do console a cada 5 s.

Uma vez terminada a simulação, a extração de dados do arquivo `odb` gerado pelo ABAQUS e o pós-processamento pode ser feito com chamada do método `post_processing` do objeto `model`: `model.post_processing()`. Para cada tipo de dados extraído há uma função para representação gráfica desses resultados. Nas figuras a seguir são apresentados os gráficos para alguns deles.

Em geral, a primeira forma de validação é uma inspeção visual da configuração deformada do duto sobre a batimetria. Desta forma é possível ver se a simulação consegue reproduzir a situação *in-loco*, especialmente os vãos. Uma vez que a etapa anterior tenha gerado os arquivos `deformed_shape.csv`, com as coordenadas da configuração deformada do duto, e `seabed.csv`, com as coordenadas da batimetria. O código para geração do gráfico desejado seria semelhante ao exposto na Figura 4, e o gráfico resultante na Figura 5. Esse e outros gráficos mais comuns são gerados automaticamente com uma chamada do método `make_plots` do objeto `model`: `model.make_plots()`.

Figura 4 – Código para geração do gráfico do perfil da configuração deformada do duto sobre a batimetria.

```

1 deformed_shape = load_csv(model.final_results_folder / "deformed_shape.csv")
2 pipe_plot = plots.pipe_profile(deformed_shape, legend="Eixo do duto")
3 seabed = load_csv(model.final_results_folder / "seabed.csv")
4 pipe_plot.title = "Configuração deformada"
5 seabed_plot = plots.bathymetry(seabed)
6 seabed_pipe_plot = pipe_plot + seabed_plot
7 seabed_pipe_plot.set_range(y=(-0.06, 0.06), x=(400, 600))
8 seabed_pipe_plot.save("configuração_deformada.html")

```

Fonte: Autor (2020)

Vale destaque o uso do operador de adição (+) na linha 6. Com ele é possível combinar dois gráficos (instâncias da classe `Plot`) sobrepondo-os em um mesmo gráfico.

O próximo passo pode ser a visualização dos modos de vibração com o objetivo de selecionar os que irão ser utilizados para o cálculo de fadiga. A Figura 6 e a Figura 7 apresentam todos os modos de vibração computados pela análise modal do ABAQUS, separando-os em *in-line* e *cross-flow*, onde a região que compreende a extensão do vão está destacada com o fundo cinza.

Figura 5 – Configuração deformada do duto após a simulação.

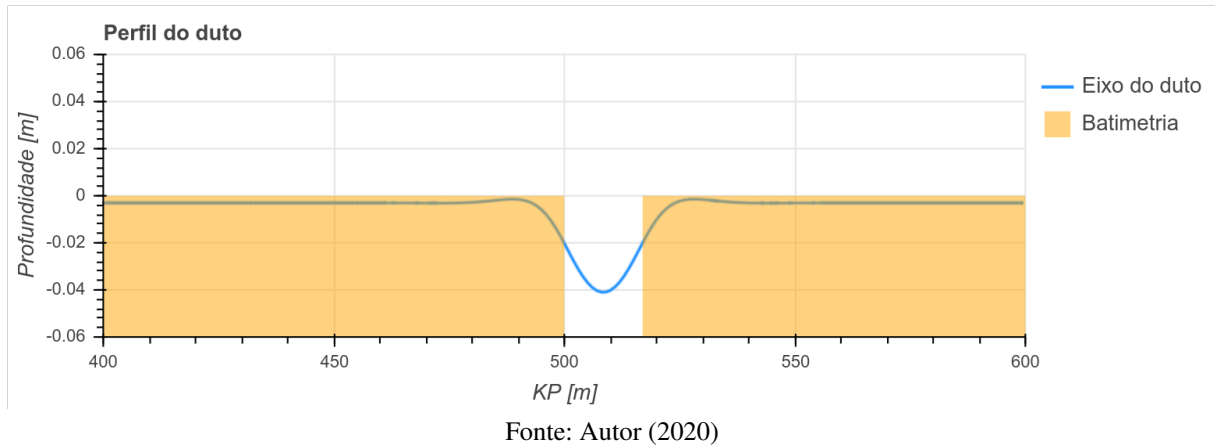


Figura 6 – Todos os modos de vibração *in-line* geradas na análise modal.

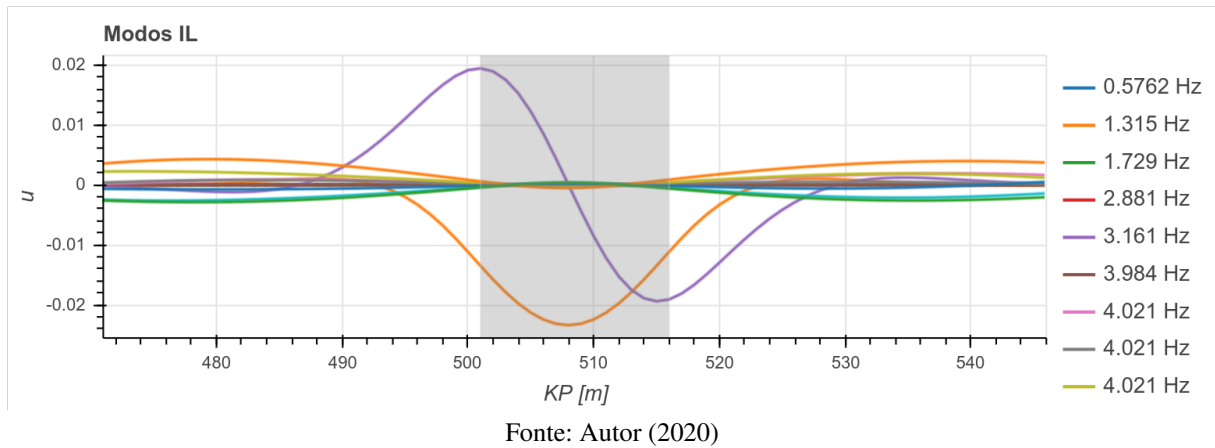
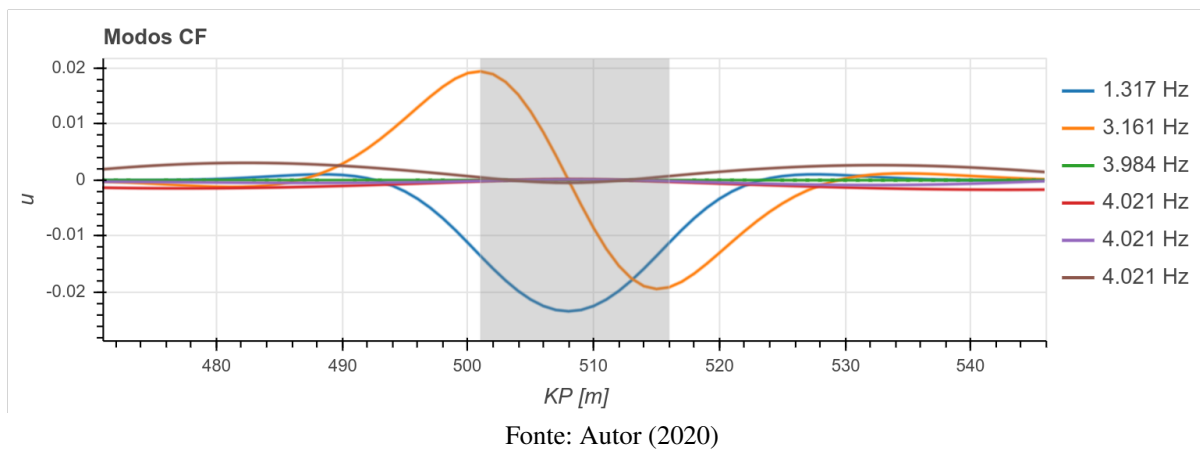


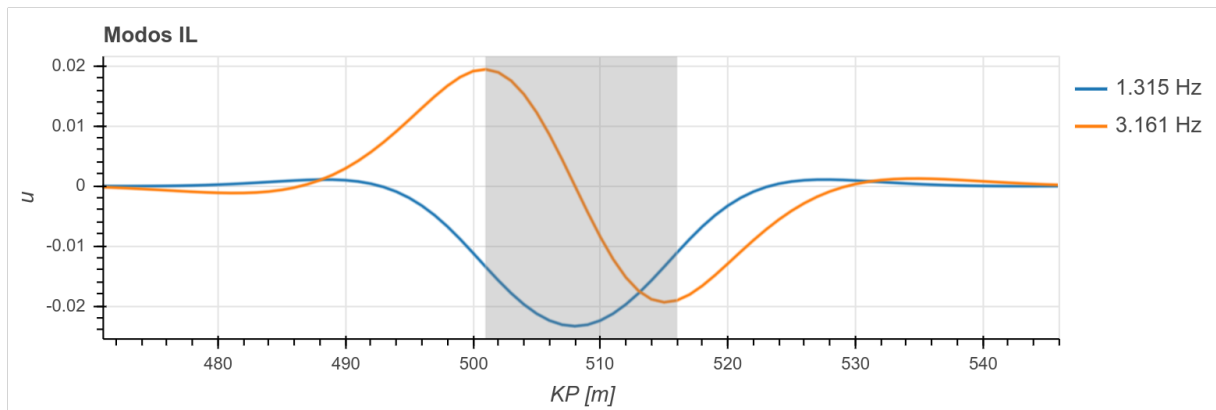
Figura 7 – Todos os modos de vibração *cross-flow* geradas na análise modal.



Como é possível ver nessas figuras, há muitos modos espúrios, que devem ser desconsiderados da análise de fadiga. Esse é o papel do método `select_modes` da classe `Span`. Este

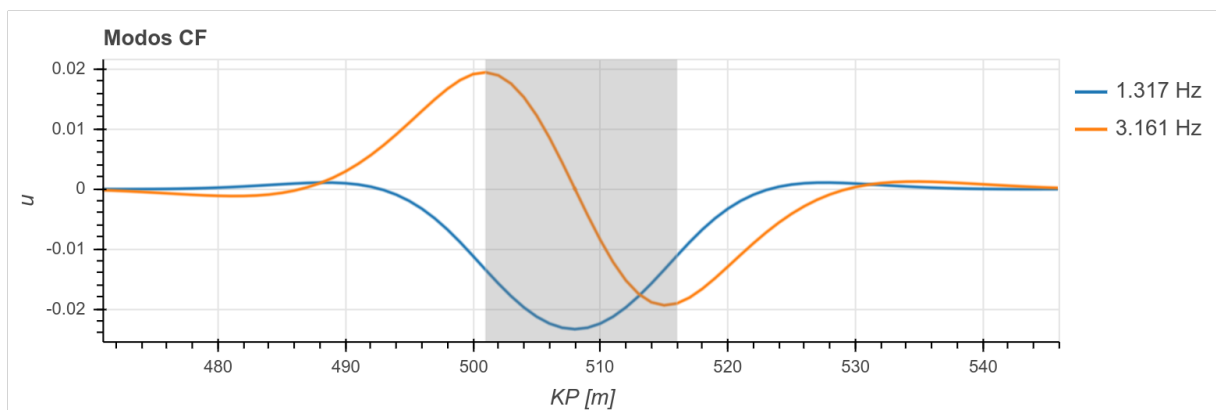
método consiste, boa parte, na implementação dos processos de seleção de modos recomendados pela DNVGL-RP-F105, apresentados na . Para o exemplo aqui apresentado, o resultado são os modos exibidos na Figura 8 e Figura 9.

Figura 8 – Modos de vibração *in-line* selecionados pelo algoritmo implementado.



Fonte: Autor (2020)

Figura 9 – Modos de vibração *cross-flow* selecionados pelo algoritmo implementado.



Fonte: Autor (2020)

Estando satisfeito com os modos selecionados na etapa anterior, o usuário pode chamar o método `run_fatfree`, para que seja feita o preenchimento de uma instância da planilha com os dados específicos do vão, modos, e dados mais gerais — como condições ambientais e coeficientes de segurança. Um exemplo da massiva entrada de dados necessária nesse processo acontece na aba *Multimode* da FatFree, onde se faz a entrada das coordenadas dos modos de vibração, conforme é possível ver na Figura 10.

Com essa instância da planilha, tem-se outro ponto para validação dos resultados. Segundo o item 6.7.4 da DNVGL-RP-F105, a análise de elementos finitos com para um único vão com força axial efetiva igual a zero e $L/D_s \approx 60$, as frequências naturais de *in-line* e *cross-flow* e as faixas de tensão devem mostrar valores semelhantes com margem de $\pm 5\%$. Para o modelo em questão, a comparação entre os valores da primeira frequência obtidos pela análise modal do ABAQUS e a FatFree é apresentada na Tabela 1.

Figura 10 – Dados inseridos na aba *Multimode* da FatFree.

A	B	C	D	E	F	G	H	I	J
Number of in-line		2		Number of evaluation points		18		Format Sheet	
Number of cross-flow		2		Number of span areas		1			
Stress Amplitude calculation option						Static stress option for Von Mises stress calculation		<input type="checkbox"/> Sensitivity on individual mode	
Mode shape input only		<input type="checkbox"/> Disregard static stress							
Span and modal input									
Location [m]	IN-LINE				CROSS-FLOW				
	IL Mode 1		IL Mode 2		CF Mode 1		CF Mode 2		
	<input type="checkbox"/> Deactivate		<input type="checkbox"/> Deactivate		<input type="checkbox"/> Deactivate		<input type="checkbox"/> Deactivate		
	f	1.3153	f	3.161	f	1.3165	f	3.161	
	A IL	153.54663	A IL	341.66245	A CF	153.496154	A CF	341.6502	
	Mode Shape	Stress Amplitude	Mode Shape	Stress Amplitude	Mode Shape	Stress Amplitude	Mode Shape	Stress Amplitude	
500	-0.47752	0.001	0.984205	0.001	-0.4777062	0.001	0.984133	0.001	
501	-0.57727	0.001	0.999995	0.001	-0.5774294	0.001	0.999924	0.001	
502	-0.6746	51.331743	0.973812	341.66245	-0.674744	51.3549717	0.973744	341.6471	
503	-0.76514	80.854772	0.902425	337.3906	-0.7652559	80.8635004	0.902362	337.3761	
504	-0.84496	106.10885	0.786383	306.4768	-0.8450419	106.100902	0.786327	306.4685	
505	-0.91069	126.45517	0.629768	252.20913	-0.9107467	126.434109	0.62972	252.204	
506	-0.95963	141.37592	0.439759	179.15955	-0.9596672	141.341064	0.43972	179.1602	
507	-0.98981	150.48487	0.226026	92.961151	-0.9898215	150.439804	0.225996	92.96488	
508	-1	153.54663	-1.6E-05	0.0059713	-1	153.496154	-3.8E-05	0.000986	
509	-0.9898	150.48378	-0.22606	92.972565	-0.9897965	150.430393	-0.22607	92.96348	
510	-0.95963	141.37219	-0.43979	179.16974	-0.9596185	141.319852	-0.4398	179.1592	
511	-0.91068	126.45113	-0.62979	252.2166	-0.9106771	126.405428	-0.6298	252.2052	
512	-0.84494	106.10213	-0.7864	306.48325	-0.8449551	106.064105	-0.7864	306.4703	
513	-0.76513	80.847702	-0.90244	337.39105	-0.7651569	80.8234916	-0.90244	337.3786	
514	-0.67458	51.322707	-0.97382	341.66131	-0.674638	51.310631	-0.97382	341.6502	
515	-0.57725	18.261118	-1	317.43829	-0.5773224	18.2644031	-1	317.4294	
516	-0.47751	0.001	-0.98421	0.001	-0.4776038	0.001	-0.98421	0.001	
517	-0.38012	0.001	-0.9335	0.001	-0.3802401	0.001	-0.93351	0.001	

Fonte: Autor 2020

Tabela 1 – Comparação entre valores para a primeira frequência.

Direção	ABAQUS	FatFree	Diferença relativa
<i>In-line</i>	1,315	1,351	-2,66%
<i>Cross-flow</i>	1,317	1,357	-2,95%

Uma vez que se tenha atingido sucesso na tarefa de modelar o problema de modo a condizentes, pode-se agora usar a classe `FatFreeResults` para extrair os resultados de uma dada instância da FatFree. Por exemplo, pode-se construir um gráfico da vida a fadiga ao longo vão com perfil do duto com a batimetria abaixo dele (ver Figura 5), como mostra a Figura 11.

Figura 11 – Código paa geração do gráficos de vida à fadiga.

```

1 from integrispan.dnv.fatfree_results import FatFreeResults
2 fatfree_results = FatFreeResults("FatFree_kp_508.5.xls", "./")
3 fatigue_life_plot = fatfree_results.plot_fatigue_life(baseline=20)
4 dashboard = fatigue_life_plot / seabed_pipe_plot
5 dashboard.save("fatigue_pipe_dashboard.html")

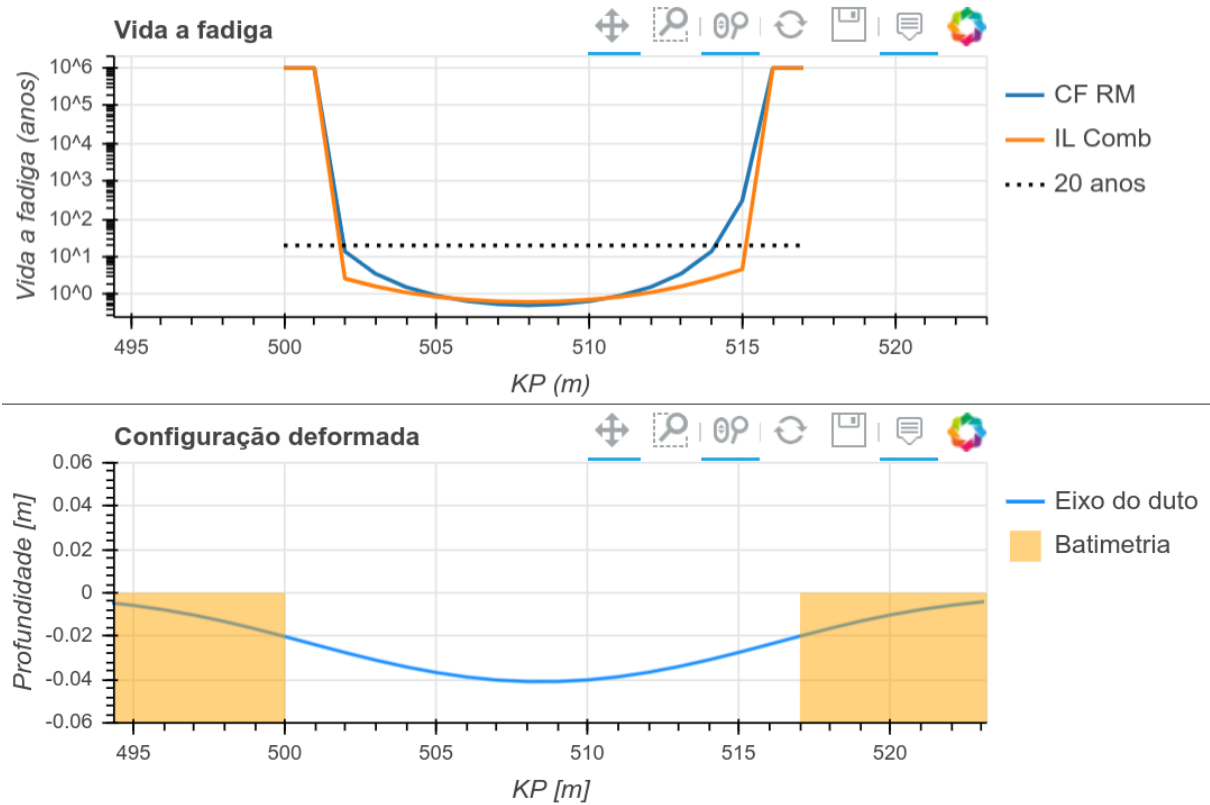
```

Fonte: Autor (2020)

Vale destaque para o uso do operador de divisão (/) na linha 4. Com ele é possível combinar dois gráficos (instâncias da classe `Plot`) posicionando-os um sobre o outro em um mesmo

gráfico. O gráfico resultante é exibido na Figura 12, onde se tem a vida a fadiga na direções *n-line* (IL Comb) e *cross-flow* (CF RM) na região do vão.

Figura 12 – Gráfico de vida a fadiga e perfil do duto.



Fonte: Autor (2020)