

UNIVERSIDADE FEDERAL DE ALAGOAS
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA CIVIL

WEVERTON MARQUES DA SILVA

**MODELAGEM E DESENVOLVIMENTO DE UMA APLICAÇÃO DE UMA
APLICAÇÃO PARA ANÁLISE DUTOS EM VÃO-LIVRE**

Maceió-AL
Setembro de 2019

WEVERTON MARQUES DA SILVA

**MODELAGEM E DESENVOLVIMENTO DE UMA APLICAÇÃO DE UMA
APLICAÇÃO PARA ANÁLISE DUTOS EM VÃO-LIVRE**

Dissertação apresentada como requisito parcial
para obtenção do grau de Mestre pelo Programa
de Pós-Graduação em Engenharia Civil do
Centro de Tecnologia da Universidade Federal
de Alagoas.

Orientador: Adeildo Soares Ramos Júnior
Coorientador: Eduardo Setton S. da Silveira

Maceió-AL

Setembro de 2019

LISTA DE TABELAS

SUMÁRIO

| | | |
|------------|---|----------|
| 1 | INTRODUÇÃO | 4 |
| 1.1 | Objetivos | 4 |
| 2 | MODELAGEM DA COMPUTACIONAL | 5 |
| 2.1 | Boas práticas para computação científica | 5 |
| | REFERÊNCIAS | 7 |

1 INTRODUÇÃO

Recentemente, as indústrias *offshore* e submarina experimentaram uma revolução técnica no processo de projeto. Métodos avançados e ferramentas de análise permitiram uma abordagem mais sofisticada ao projeto, que aproveita os materiais modernos e os códigos de projeto revisados, que dão suporte aos conceitos de estado limite de projeto e aos métodos de confiabilidade. A nova abordagem é chamada de Projeto Através de Análise (*Design Through Analysis* – DTA), onde o Método dos Elementos Finitos é usado para simular o comportamento global dos dutos, bem como os esforços estruturais locais.

O processo de duas etapas é usado de maneira complementar para determinar os estados limites normativos e otimizar um projeto específico.

Devido a quantidade de fatores envolvidos, a análise requer o uso de métodos numéricos robustos para seu tratamento. O Método dos Elementos Finitos (MEF) é amplamente usado nessa tarefa. De modo a representar adequadamente as condições de campo, é necessário modelar desde a etapa de instalação até a operação do duto, assim como considerar efeito de carregamentos dos diferentes valores de pressões internas e externas nas respectivas etapas.

1.1 Objetivos

2 MODELAGEM DA COMPUTACIONAL

Neste capítulo serão apresentados os aspectos para o desenvolvimento da ferramenta, como os pré-requisitos, escolhas das ferramentas, decisões de modelagem, e as práticas adotadas no processo.

2.1 Boas práticas para computação científica

Wilson et al. (2014) apresenta um conjunto de boas práticas a serem adotadas no desenvolvimento de softwares de cunho científico. A seguir é apresentado o resumo do autor sobre essas práticas:

1. Escreva programas para pessoas, não para computadores.
 - a) Um programa não deve exigir que seus leitores mantenham mais de um punhado de fatos na memória de uma só vez.
 - b) Torne os nomes consistentes, distintos e significativos.
 - c) Tornar consistente o estilo e a formatação do código.
2. Deixe o computador fazer o trabalho.
 - a) Faça o computador repetir tarefas.
 - b) Salve comandos recentes em um arquivo para reutilização.
 - c) Use uma ferramenta de construção para automatizar fluxos de trabalho.
3. Faça alterações incrementais.
 - a) Trabalhe em pequenos passos com feedback frequente e correção de rumo.
 - b) Use um sistema de controle de versão.
 - c) Coloque tudo o que foi criado manualmente no controle de versão.
4. Não se repita (ou repita outros).
 - a) Todos os dados devem ter uma única representação oficial no sistema.
 - b) Modularize o código em vez de copiar e colar.
 - c) Reutilize o código em vez de reescrevê-lo.
5. Planeje erros.
 - a) Adicione asserções aos programas para verificar seu funcionamento.
 - b) Use uma biblioteca de testes unitários pronta para uso.

- c) Transforme erros em casos de teste.
 - d) Use um depurador simbólico.
- 6. Otimize o software somente depois que ele funcionar corretamente.
 - a) Use um *profiler* para identificar gargalos.
 - b) Escreva o código na linguagem de nível mais alto possível.
- 7. Documente design e finalidade, não a mecânica.
 - a) Documente interfaces e razões, não implementações.
 - b) Refatore o código, em vez de explicar como ele funciona.
 - c) Incorpore a documentação do software no próprio software.
- 8. Colabore.
 - a) Use revisões de código de *pré-merge*.
 - b) Use a programação em pares ao interar alguém novo e ao lidar com problemas particularmente difíceis.
 - c) Use uma ferramenta de acompanhamento de problemas.

Em observância à estas práticas algumas decisões foram tomadas quanto ao desenvolvimento desta ferramenta:

REFERÊNCIAS

WILSON, G. et al. Best Practices for Scientific Computing. *PLoS Biology*, Public Library of Science, v. 12, n. 1, p. e1001745, jan 2014. ISSN 1545-7885. Disponível em: <<https://dx.plos.org/10.1371/journal.pbio.1001745>>. Citado na página 5.