UNIVERSIDADE FEDERAL DE PERNAMBUCO
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DESENVOLVIMENTO DE FERRAMENTAS COMPUTACIONAIS PARA MODELAGEM E ANÁLISE AUTOMÁTICA DE DEFEITOS DE CORROSÃO EM DUTOS

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE PERNAMBUCO PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA MECÂNICA.

AUTOR: HÉLDER LIMA DIAS CABRAL
ORIENTADOR: RAMIRO BRITO WILLMERSDORF
CO-ORIENTADORA: SILVANA MARIA BASTOS AFONSO DA SILVA

Recife, Fevereiro de 2007.

"DESENVOLVIMENTO DE FERRAMENTAS COMPUTACIONAIS PARA MODELAGEM E ANÁLISE AUTOMÁTICA DE DEFEITOS DE CORROSÃO EM DUTOS".

HÉLDER LIMA DIAS CABRAL

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA MECÂNICA

ÁREA DE CONCENTRAÇÃO: MECÂNICA COMPUTACIONAL
APROVADA EM SUA FORMA FINAL PELO
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA/CTG/EEP/UFPE

Prof. Dr. RAMIRO BRITO WILLMERSDORF
ORIENTADOR/PRESIDENTE

Profa. Dra. SILVANA MARIA BASTOS AFONSO DA SILVA
CO-ORIENTADORA

Profa. Dra. ANA ROSA MENDES PRIMO
COORDENADORA DO CURSO

BANCA EXAMINADORA:

Prof. Dr. RAMIRO BRITO WILLMERSDORF (UFPE)

Profa. Dra. SILVANA MARIA BASTOS AFONSO DA SILVA (UFPE)

Prof. Dr. PAULO ROBERTO MACIEL LYRA (UFPE)

Dr. ADILSON CARVALHO BENJAMIN (PETROBRÁS/RJ)

C117d **Cabral, Hélder Lima Dias**

Desenvolvimento de ferramentas computacionais para modelagem e análise automática de defeitos de corrosão em dutos / Hélder Lima Dias Cabral. – Recife: O Autor, 2007.

xvii, 143 f.; il. color., gráfs., tabs.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Mecânica, 2007.

Inclui referências e apêndices.

1. Engenharia Mecânica. 2. Dutos – Defeitos de Corrosão. 3. Elementos Finitos. 4. Modelagem Automática. 5. PATRAN – Interface Gráfica. I. Título.

621 CDD (22.ed.) **UFPE/BCTG/2007-34**

DEDICATÓRIA

Dedico esta dissertação de mestrado aos meus pais, José Ademar e Maria Lenauda, que são os meus verdadeiros orientadores e responsáveis pela minha formação pessoal e profissional. Eles que sempre acompanharam de perto as minhas batalhas, e sempre foram para mim o exemplo máximo de humildade, determinação e fé.
Sou eternamente grato.

AGRADECIMENTOS

A Deus, minha eterna fonte de inspiração e fé. Ele que está sempre ao meu lado, em todos os momentos da minha vida. Sem ele, nada disso seria possível.

Gostaria de agradecer ao meu orientador, Professor Ramiro Brito Willmersdorf, pela confiança depositada, pelos desafios que foram impostos a mim durante o projeto de pesquisa o qual originou esta dissertação e pelo incentivo, orientação e paciência no acompanhamento dos trabalhos desde o período de iniciação científica. Sua extrema competência e capacidade de solucionar problemas (raramente vistas) tornaram este trabalho menos árduo.

Eu gostaria também de expressar os meus sinceros agradecimentos à minha co-orientadora, Professora Silvana Maria Bastos Afonso da Silva, pelo incentivo, orientação e pela oportunidade de trabalhar em conjunto no projeto sob sua coordenação, o qual originou esta dissertação. Juntamente com a professora Silvana, gostaria de agradecer ao Professor Paulo Roberto Maciel Lyra que foi pra mim também o meu co-orientador. Ambos colaboraram bastante fornecendo suas valiosas críticas e sugestões não somente para este trabalho de dissertação, mas também para os dois trabalhos publicados em congressos que participei ao longo deste mestrado (CONEM e CILAMCE).

Não podia deixar de mencionar os meus sinceros agradecimentos ao CENPES/PETROBRÁS, em nome dos Engenheiros Adilson Carvalho Benjamin e Edmundo Queiroz de Andrade, por permitirem que esse trabalho fosse publicado. Em especial, gostaria de agradecer pelas valiosas discussões dos procedimentos de solução e geração dos modelos que aqui foram implementados baseados no padrão adotado pelo CENPES. Esses procedimentos foram resultados de projetos de consultoria e pesquisa financiados e monitorados pela PETROBRÁS e foram muito importantes para que o grupo PADMEC utilizasse o "*know how*" já adquirido pela equipe do CENPES. Gostaria também de agradecer em nome da equipe PADMEC pelos valiosos artigos fornecidos e que foram utilizados neste trabalho contendo os dados experimentais e numéricos fundamentais para a validação das ferramentas aqui desenvolvidas.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), à Financiadora de Estudos e Projetos (FINEP/CTPETRO) e ao CENPES/PETROBRÁS pela ajuda financeira recebida durante o período deste trabalho. Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela ajuda financeira recebida durante o período de graduação (iniciação científica).

Ao Professor José Maria Andrade Barbosa, pela oportunidade de trabalhar, ainda na graduação, em um dos seus projetos de pesquisa ("Danos em Dutos") no qual tive o primeiro contato com as ferramentas de pesquisa científica. Em particular, o Método dos Elementos Finitos e linguagens de programação. Naquela época, os meus amigos Gustavo de Novaes Pires Leite e Renato Bezerra Pajeú me ajudaram e incentivaram bastante nas horas de estudo e pesquisa na área computacional.

Ao Professor Ocke Meister, pela oportunidade de participar de atividades de pesquisa junto ao Instituto de Pesquisas Aeronáuticas da Universidade Técnica de Braunschweig (TU-BS) durante período de intercâmbio na graduação. Foi justamente durante este período que tive a oportunidade de trabalhar com o software MSC.PATRAN, principal ferramenta utilizada neste trabalho.

A Flávio e Juliana, pelas orientações e ajuda durante os projetos nos quais participaram e que antecederam este. Também gostaria de agradecer à Manoela e Renato por se envolverem neste projeto dando continuidade a este trabalho com muita competência.

À minha namorada, Ana Karina Pessoa da Silva, pelo incentivo, compreensão e muita paciência nesse longo período de pesquisa e muito trabalho.

Às minhas irmãs: Nádia, Edna e Leane, que são uma referência e estímulo para mim.

Aos meus amigos e companheiros de pós-graduação com quem tive o privilégio de conviver e estudar. Em especial os meus amigos: André, Andrea, Brito, Carlos, Daniel, De-

metrium, Eliane, Gilson, Hélder (catota), Jane, Gisele, Guaraci, Henrique, Marcelo, Silvana, Wellington e Xistófanes.

Aos membros do grupo PADMEC pelas preciosas reuniões que passaram a ser chamadas de “paredão”. Gostaria também de agradecer aos companheiros do laboratório de computação (LABCOM) pelos momentos de descontração e pela amizade. Em especial: “Papola”, Bruno, Michel, Rodrigo, Fred, Rafael, João e Manassés.

A todos os meus amigos da minha turma de graduação que, mesmo após dois anos de formados, continuamos a nos encontar nos bares, churrascos e casas de amigos o que contribui para mantermos essa forte amizade que existe entre nós. Isso é muito importante para mim.

E agradeço a todos que aqui não foram citados, mas que direta ou indiretamente contribuíram para que este trabalho fosse concluído com muito esforço e dedicação.

"No one believes the numerical results except the one who performed the calculation, and everyone believes the experimental results except the ones who performed the experiment." (Saying in aerodynamics community)

RESUMO

A segurança operacional da malha de dutos de transporte de hidrocarbonetos é uma grande preocupação de todas as companhias de petróleo, devido aos imensos danos econômicos, sociais e em termos da imagem da companhia que um acidente de grande porte com um duto pode causar. Esta malha deve ser monitorada continuamente e problemas encontrados devem ser avaliados de forma confiável, a fim de analisar o comprometimento da integridade estrutural do duto e permitir que reparos necessários sejam realizados com segurança, antes que estes defeitos causem um acidente. No caso de defeitos causados por corrosão, a análise computacional com o Método dos Elementos Finitos (MEF) tem se mostrado uma das ferramentas mais eficientes para a avaliação correta da integridade estrutural de dutos com defeitos. Estas ferramentas permitem considerar diretamente os fenômenos físicos envolvidos no processo de falha do duto resultando assim em resultados mais precisos que os encontrados por meio de modelos semi-empíricos e bem mais rápidos e econômicos que os obtidos através de experimentos em laboratório.

A análise via o MEF, no entanto, requer uma grande especialização e um treinamento específico que não são característicos de todos os engenheiros de tubulações. O processo para a criação de bons modelos computacionais para um defeito, que inclui a modelagem fiel da geometria deste defeito e a geração de uma malha apropriada, demanda uma interação manual constante do engenheiro, é demorado e muito repetitivo, e por estas razões muito propenso a erros. Normalmente, este procedimento é repetido desde o início para cada novo problema analisado, em um patente desperdício de recursos humanos qualificados.

A principal proposta deste trabalho foi desenvolver um conjunto de ferramentas computacionais que produzem automaticamente modelos de dutos com defeitos, prontos para serem analisados em programas comerciais que implementam o MEF, a partir de alguns parâmetros que localizem e forneçam as dimensões principais do defeito ou de uma série de defeitos. Estas ferramentas são baseadas no programa comercial de pré e pós-processamento MSC.PATRAN e foram produzidas por meio da linguagem de programação PCL (Patran Command Language). O programa de geração automática de modelos (denominado programa PIPEFLAW) tem interface gráfica simplificada e personalizada, de forma que um engenheiro, com noções básicas de simulação computacional com elementos finitos, possa gerar rapidamente modelos que resultem em simulações precisas e confiáveis.

A realização das análises não-lineares foi feita por meio de um "script" implementado na linguagem de programação PYTHON a partir do qual toda a análise é gerenciada através da execução automática de tarefas pré-determinadas. Assim, a cada iteração da análise, o solver "ANSYS" é acionado pelo "script" e em seguida, os resultados gerados durante aquela iteração são lidos e interpretados pelo "script" possibilitando que os critérios de convergência e incremento de carga, pré-definidos pelo usuário, sejam aplicados automaticamente, diferentemente do procedimento usual quando se ativa os critérios de convergência e de incremento de carga determinados pelo "solver".

Por fim, são apresentados alguns exemplos de modelos de dutos gerados automaticamente pelo programa PIPEFLAW. Os resultados de análises numéricas, realizadas utilizando as ferramentas desenvolvidas neste trabalho, são comparados com resultados empíricos, numéricos e experimentais disponíveis na literatura com o objetivo de validar as ferramentas aqui apresentadas.

Palavras-chave: Dutos, Defeitos de Corrosão, Elementos Finitos, Modelagem Automática, PATRAN, Interface Gráfica.

ABSTRACT

The operational safety of hydrocarbon transport pipelines is a major concern of all oil companies, due to the enormous economic, social and public image damage that can arise from a major pipeline accident. These pipelines must be monitored continuously and potential problems must be evaluated reliably, to assess the structural integrity of the compromised pipe and to allow that repair be effected safely, before these defects cause an accident. The Finite Element Method (FEM) is one of the most efficient tools to quantify reliably the remaining strength of corroded pipes. These tools allow the direct simulation of the physical phenomena involved in the failure of the pipe, providing more precise results than the ones found through semi-empirical methods and much faster and cheaper results than the ones from experiments.

FEM analysis requires, however, specific knowledge and training that are not characteristic of all pipeline engineers. The process of creating good computational models for a defect, which includes precise representation of the geometry of the defect and the generation of an appropriate mesh demand intense manual labor from the engineer, and it is also slow and extremely repetitive, therefore it is very error prone. Normally, this process is repeated from the very beginning for each new defect to be analyzed, in a clear waste of qualified human resources.

The main purpose of this work was to develop a set of computational tools to produce automatically models of pipes with defects, ready to be analyzed with commercial FEM programs, starting from a few parameters that locate and provide the main dimensions of the defect or a series of defects. These tools were based on MSC.PATRAN pre- and post-processing program, and were written with PCL (Patran Command Language). The program for the automatic generation of models (PIPEFLAW) has a simplified and customized graphical interface, so that an engineer with basic notions of computational simulation with the FEM can generate rapidly models that result in precise and reliable simulations.

The non-linear analyses were run by a Python “script”, which manages all analysis though the execution of preprogrammed tasks. At each iteration of the non-linear analysis, the “ANSYS” solver is run by the script; following that, the results of that iteration are read and interpreted by the script, allowing that convergence criteria and load increments, predefined by the user, be applied automatically, in contrast with the usual procedure where automatic convergence criteria and load increments are determined by the solver.

Finally, some examples of models of pipes with defects generated by the PIPEFLAW system are shown, and results of numerical analysis, done with the tools presented in this work are compared with experimental, numeric and empiric results available in the literature, to validate these tools.

Keywords: Pipelines, Corrosion Defects, Finite Elements, Automatic Modeling, PATRAN, GUI.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

1 INTRODUÇÃO

1.1 Considerações Iniciais

A segurança operacional da malha de dutos de transporte de hidrocarbonetos é uma grande preocupação de todas as companhias de petróleo, devido aos imensos danos econômicos, sociais e em termos da imagem da companhia que um acidente de grande porte com um duto pode causar. No Brasil, em particular, a malha dutoviária tem dezenas de milhares de quilômetros de extensão e uma vida operacional considerável. Esta malha deve ser monitorada continuamente e problemas encontrados devem ser avaliados de forma confiável, a fim de analisar o comprometimento da integridade estrutural do duto e permitir que reparos necessários sejam realizados com segurança, antes que estes defeitos causem um acidente. No caso de defeitos causados por corrosão, a análise computacional com o Método dos Elementos Finitos (MEF) tem se mostrado uma das ferramentas mais eficientes para a avaliação correta da integridade estrutural de dutos com defeitos (Chouchaoui et al, 1992; Fu & Kirkwood, 1995; Bathe, 1996; Cronin, 2002; Andrade et al, 2006). No entanto, é válido lembrar que os resultados das simulações numéricas via o MEF só têm confiabilidade se os modelos de elementos finitos foram previamente validados por meio da comparação com testes de laboratório. A avaliação da segurança de dutos com defeitos de corrosão é normalmente feita por meio de métodos semi-empíricos disponibilizados através de normas bastante utilizadas pelo setor, tal como as normas BS 7910 (BS 7910, 1999) e DNV RP-F101 (DNV, 1999). O uso de normas, no entanto, sempre implica em uma grave simplificação na geometria dos defeitos reais, o que pode levar a resultados imprecisos. A simulação computacional pelo MEF permite uma representação muito mais fiel destes defeitos, e além disto, por considerar diretamente os fenômenos físicos envolvidos no processo de falha do duto, produz resultados mais precisos que os encontrados por meio de métodos semi-empíricos e bem mais rápidos e econômicos que os obtidos através de experimentos em laboratório.

A análise via o MEF, no entanto, requer uma grande especialização e um treinamento específico que não são característicos de todos os engenheiros de tubulações. O processo para a criação de bons modelos computacionais para um duto com defeito, que inclui a modelagem fiel da geometria do defeito e a geração de uma malha apropriada, demanda uma interação manual constante do engenheiro, é demorado e muito repetitivo, e por estas razões muito propenso a erros. Normalmente, este procedimento é repetido desde o início para cada novo problema analisado, em um patente desperdício de recursos humanos qualificados.

A principal proposta deste trabalho foi desenvolver um conjunto de ferramentas computacionais que produzem automaticamente modelos de dutos com defeitos de corrosão, prontos para serem analisados em programas comerciais que implementam o MEF, a partir de alguns parâmetros que localizem e forneçam as dimensões principais do defeito ou de uma série de defeitos. Estas ferramentas são baseadas no programa comercial de pré e pós-processamento MSC.PATRAN (PATRAN, 2005) e foram produzidas por meio da linguagem de programação PCL (Patran Command Language). O programa de geração automática de modelos (denominado programa PIPEFLAW) tem interfaces gráficas simplificadas e personalizadas, de forma que um engenheiro, com noções básicas de simulação computacional com elementos finitos, possa gerar rapidamente modelos que resultem em simulações precisas e confiáveis. Até o momento, o programa PIPEFLAW gera automaticamente modelos de dutos com defeitos simples ou múltiplos alinhados (longitudinalmente ou circunferencialmente), com geometria retangular, localizados na superfície interna ou externa do duto. O programa PIPEFLAW deverá comportar também outras funcionalidades como geração de múltiplos defeitos em posição arbitrária, geração de defeitos superpostos (malhas não-estruturadas) e deverá incluir defeitos com geometria elíptica.

Os softwares comerciais de elementos finitos atualmente existentes no mercado possuem ferramentas poderosas para a execução de análises não-lineares, possibilitando que o

usuário estabeleça diversos parâmetros para o controle da análise. Em particular, no ANSYS (ANSYS, 2004), o usuário pode optar pelo recurso "Automatic Time Step". Este recurso ativa um esquema automático de forma a garantir que a variação do incremento de carga não seja nem muito grande (o que resulta em muitas bisseções e re-análises) e nem também muito conservador (o que resulta em um incremento de carga muito pequeno).

No entanto, algumas situações particulares requerem um controle mais determinístico da aplicação dos incrementos de carga. Uma alternativa viável no ANSYS é o uso do recurso "Save/Restart" que possibilita que o usuário reinicialize a análise não-linear a cada incremento de carga a partir de um passo imediatamente anterior. Apesar de ser um procedimento manual, este tipo de recurso possibilita que o usuário controle a cada incremento de carga os resultados gerados durante a análise e estabeleça critérios próprios de convergência baseados nos valores de variáveis físicas quaisquer, o que não ocorre no recurso "Automatic Time Step" no qual o controle da análise é limitado somente a alguns parâmetros pré-definidos pelo "solver".

Além das análises não-lineares já serem bastante demoradas, o uso do procedimento "Save/Restart" padrão fornecido pelo ANSYS exige que o engenheiro detenha boa parte do seu tempo coletando e interpretando os resultados obtidos a cada passo de incremento de carga, o que torna o trabalho bastante repetitivo e demorado, além de ser muito propenso a erros.

A fim de tornar este processo mais rápido e mais confiável, foi criado um procedimento automático, usando o recurso "Save/Restart", para controlar análises não-lineares no ANSYS baseado no procedimento padrão pré-estabelecido pelo CENPES/PETROBRÁS (Benjamin & Andrade, 2005). Para isso, foi implementado um programa interpretado ("script"), na linguagem de programação Python (PYTHON, 2005), a partir do qual toda a análise é gerenciada por meio da execução automática de tarefas pré-determinadas. Assim, a cada iteração da análise não-linear, o solver (ANSYS) é acionado pelo "script" e em seguida, os resultados gerados durante aquela iteração, são extraídos por meio de manipulação de arquivos (Cabral et al, 2006a).

A interpretação desses resultados pelo "script", possibilita que os critérios de convergência e de incremento de carga, pré-determinados pelo usuário, sejam verificados e calculados automaticamente, diferentemente do procedimento usual quando se ativa os critérios de convergência e de incremento de carga determinados pelo "solver". No final da análise, é gerado também automaticamente um resumo do histórico da análise não-linear em um arquivo no formato Excel.

1.2 Motivação

As ferramentas computacionais aqui apresentadas foram produzidas pelo grupo de pesquisa PADMEC por indução do CENPES/PETROBRÁS no projeto "Geração Automática de Defeitos de Corrosão em Dutos - MOSINEIP" da Rede Norte-Nordeste de Pesquisa Cooperativa em Modelagem Computacional (RPCMOD-Rede9) com o apoio financeiro do FINEP/CTPETRO e CENPES/PETROBRÁS.

Entendendo que a análise de defeitos via o MEF já é uma das principais ferramentas para a avaliação da integridade estrutural de dutos, os benefícios obtidos pela companhia operadora de dutos ao utilizar essas ferramentas automáticas são vários:

- *Redução no tempo de criação do modelo* – Atualmente, a geração de um bom modelo de defeito pode levar muitos dias. Claramente, isto dificulta que simulações computacionais sejam usadas na tomada de decisão sobre a segurança de um duto específico, já que raramente o engenheiro pode dar-se ao luxo de exigir que um duto suspeito seja retirado de operação por vários dias, enquanto elabora modelos computacionais. Com a aceleração deste processo, visualizamos um futuro no qual o engenheiro pode apelar para ferramentas computacionais e tomar decisões baseadas nos seus resultados.

- *Redução de erros de modelagem* – A geração automática, por requisitar muito menor intervenção manual, é muito menos propensa a erros do que a geração manual, sendo possível também implementar alguns mecanismos de verificação automática que diminuem a probabilidade de geração de modelos inadequados mesmo que o usuário assim o requisite. As conseqüências de um modelo computacional incorreto, que está sendo usado para avaliar a segurança de um duto, podem ser catastróficas.
- *Uso eficiente de mão de obra especializada* – Os engenheiros de tubulações são, em geral, extremamente competentes em sua área de atuação. Tê-los sentados na frente de um computador, simplesmente repetindo um procedimento mecânico que pode ser automatizado, é um desvio de função. Os engenheiros devem concentrar-se em controlar a simulação computacional, avaliando a confiabilidade dos seus resultados e verificando sua validade, e amparados por estes resultados, mas baseados também em sua experiência e conhecimentos, propor soluções para os problemas encontrados. Além disto, um processo de geração automática de modelos permite que outros engenheiros, que não tenham treinamento específico no programa de modelagem, possam realizar análises computacionais.
- *Economia e segurança* – Claramente, as vantagens acima se traduzem em economia de recursos e no aumento da segurança operacional. Com o aumento da velocidade de tomada de decisão sobre a condição de um duto pode-se diminuir o tempo de parada de um sistema comprometido. Além disto, uma avaliação mais precisa da gravidade de um defeito permite que, por exemplo, se opere o duto à pressão reduzida, com segurança, até que uma parada pré-programada seja atingida, sendo preservada pelo menos parte da produção.

1.3 Objetivos

O objetivo final desta dissertação é desenvolver ferramentas computacionais, baseadas no programa comercial MSC.PATRAN, para a geração automática de modelos de dutos com defeitos de corrosão.

O presente trabalho constitui parte do projeto de pesquisa "Geração Automática de Defeitos de Corrosão em Dutos - MOSINEIP" da Rede Norte-Nordeste de Pesquisa Cooperativa em Modelagem Computacional (RPCMOD-Rede9). Os defeitos serão modelados, neste projeto, através de dois tipos de geometria simplificada: defeitos retangulares e defeitos elípticos. As dimensões do defeito e sua localização serão fornecidas pelo usuário via interface gráfica. Serão tratados defeitos individuais e conjuntos de defeitos.
As malhas de elementos finitos geradas para os modelos devem ser mapeadas, isto é, estruturadas por blocos, e compostas de elementos hexaédricos lineares.

Alguns objetivos desta dissertação são partes das metas físicas previstas para este projeto. Os desenvolvimentos específicos para este trabalho de dissertação estão listados abaixo:

- Geração automática de dutos com defeitos retangulares em superfície interna ou externa:
  a) Gerar um defeito isolado em posição arbitrária no duto.
  b) Gerar "n" defeitos alinhados longitudinalmente no duto.
  c) Gerar "n" defeitos alinhados circunferencialmente no duto.
- Implementar interface gráfica que gerencia os "scripts" de geração automática dos modelos com defeitos retangulares.
- Automatizar o procedimento padrão de análises não-lineares proposto pelo CENPES/PETROBRÁS.
- Apresentar um conjunto de exemplos de modelos gerados automaticamente pelo programa PIPEFLAW com o objetivo de testar a robustez e limitação do mesmo.
- Validar as ferramentas desenvolvidas através de simulações numéricas que serão comparadas com resultados experimentais e numéricos disponíveis na literatura.

1.4 Organização da Dissertação

Este trabalho de dissertação divide-se em seis capítulos. Este capítulo mostrou uma visão geral da dissertação, bem como descreveu a sua organização.

O **capítulo 2 – Revisão Bibliográfica** – apresenta de forma básica e sucinta os principais tópicos relacionados com a integridade estrutural de dutos. São apresentados algumas definições e tipos de defeitos que comumente aparecem nos dutos bem como os principais métodos de controle, inspeção e monitoramento dos mesmos. Este capítulo descreve ainda os principais métodos semi-empíricos, utilizados na avaliação da resistência residual de dutos corroídos e apresenta algumas contribuições de trabalhos na área de simulações numéricas e ensaios experimentais envolvendo defeitos de corrosão em dutos.

O **capítulo 3 – Automatização da Modelagem de Defeitos de Corrosão em Dutos –** descreve quais e como foram desenvolvidas as ferramentas de interface gráfica e modelagem automática utilizadas no programa PIPEFLAW.

O **capítulo 4 – Automatização da Análise Não-Linear –** descreve quais e como foram desenvolvidas as ferramentas utilizadas para automatizar o procedimento de análises não-lineares utilizado pelo CENPES/PETROBRÁS para a obtenção da pressão de ruptura de dutos com defeitos causados por corrosão.

O **capítulo 5 – Resultados –** são apresentados alguns exemplos de modelos de dutos gerados automaticamente pelo programa PIPEFLAW. Ainda neste capítulo, alguns resultados de simulações numéricas, realizadas utilizando as ferramentas desenvolvidas neste trabalho, são comparados com resultados empíricos, numéricos e experimentais disponíveis na literatura com o objetivo de validar as ferramentas aqui apresentadas.

Finalmente, no **capítulo 6 – Conclusões e Trabalhos Futuros –** são feitas as conclusões e considerações finais sobre a aplicação das ferramentas computacionais desenvolvidas, mostrando as principais contribuições deste trabalho. Neste capítulo, são apresentados os principais trabalhos em andamentos nesta linha de pesquisa conduzidos pelo grupo PADMEC. Algumas sugestões para trabalhos futuros são também fornecidas.

2 REVISÃO BIBLIOGRÁFICA

Uma sofisticada estrutura de transporte de petróleo e seus derivados interligam as fontes de produção, refinarias e centros de consumo em qualquer país. No Brasil, onde a Petrobrás é a principal empresa do ramo petrolífero e dispõe de uma malha dutoviária de mais de 30.300Km[1], os dutos exercem um papel fundamental como meio de transporte e sua demanda vem aumentando significativamente devido ao conseqüente aumento da produção e consumo no país.

O transporte por meio de dutos apresenta várias vantagens que justificam o seu amplo uso principalmente quando se trata de grandes quantidades de fluido a ser transportado. O custo de transporte por dutos é, na maioria das vezes, inferior se comparado com outros tipos de transporte conforme Figura 2.1.

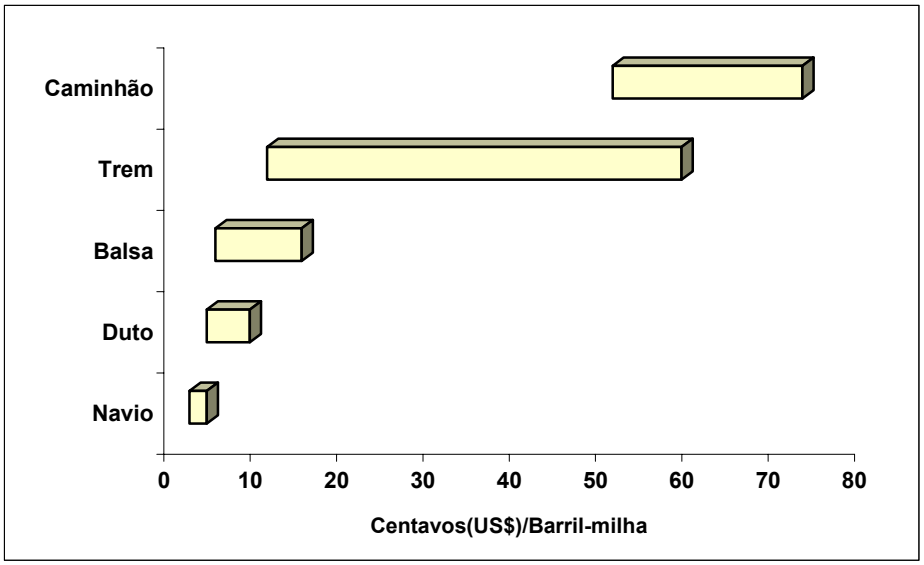


Figura 2.1 – Custos de diferentes tipos de transporte de petróleo e derivados. Fonte: Adaptado de Kennedy (1993).

Os dutos também se destacam por apresentarem um baixo consumo de energia (energia consumida por tonelada-km de carga transportada) em relação a outros tipos de transporte como caminhões e trens. Além disso, agridem menos o meio ambiente (quando não há acidentes), e possuem alta confiabilidade devido ao seu alto grau de automação que permite que redes de dutos operem continuamente sem serem afetadas pela ação da natureza, pois a maioria dos dutos é enterrada. (Liu, 2003).

O desenvolvimento de novas ligas que conferem melhores propriedades mecânicas ao material tem contribuído para a redução de custos na produção dos dutos, tornando possível selecionar menores espessuras de parede, mantendo-se a mesma pressão de operação (Fedele, 2002).

Como exemplo, podemos citar uma especificação do API (American Petroleum Institute) denominada API 5L-X. Esta é uma especificação para tubos com ou sem costura, de aços-carbono de alta resistência e baixa liga, especiais para oleodutos e gasodutos abrangendo vários graus de material, indo desde o grau X42 até o grau X80 (Telles, 1997).

O grau do aço, segundo a norma API (API, 2000), reflete a tensão mínima de escoamento do material em [ksi], ou seja, o grau X70 tem esta tensão no valor igual a 70 ksi. Nos últimos anos, tem sido testada com sucesso a inserção de elementos de liga como o molibdê-

[1] Dados de fev. 2006, Petrobrás em Números. (http://www2.petrobras.com.br/portal/Petrobras.htm)

nio, o cobre e o níquel, além de processos modificados de resfriamento controlado, possibilitando o desenvolvimento de aços com graus X100 e X120 (Santos Neto, 2003).

2.1 Integridade de Dutos

Apesar dos dutos serem uma das formas mais seguras e confiáveis para o transporte de petróleo e derivados, as companhias do setor se preocupam constantemente com a segurança operacional dos dutos, pois sabem que um acidente de grande porte pode causar imensos danos econômicos, sociais e em termos da imagem da companhia. No Brasil, em particular, a malha dutoviária tem dezenas de milhares de quilômetros de extensão e uma vida operacional considerável. Esta malha deve ser monitorada continuamente e problemas encontrados devem ser avaliados de forma confiável, a fim de avaliar o comprometimento da integridade estrutural do duto e permitir que reparos necessários sejam feitos com segurança, antes que ocorra um acidente.

A integridade de dutos é a atividade responsável pela certificação de que um determinado duto esteja operando com alto grau de segurança e envolve vários aspectos e informações das áreas de projeto, operação, inspeção e manutenção. A aplicação de um programa de gerenciamento da integridade de dutos assume um papel fundamental para sistematizar as informações obtidas destas diversas áreas mantendo uma visão integrada do problema (Souza 2003).

Para manter a integridade de um duto temos que saber quais são as causas que levam um duto a falhar. Hopkins (2002b) mostra que a maior causa de falhas em dutos (óleo e gás) nos EUA é devida a fatores externos (danos provocados por ação de terceiros) seguida de falhas devido à corrosão. A CETESB-SP (Companhia de Tecnologia de Saneamento Ambiental de São Paulo) realizou um estudo para apurar as causas de vazamentos em dutos, durante o período de 1980-2002. A classificação utilizada para causa dos acidentes envolvendo dutos foi:

- Causas Naturais: eventos associados com ação da natureza tais como erosão, deslizamentos de terra ou movimentação do solo;
- Ação de Terceiros: eventos associados com perfuração não intencional da linha, atos de vandalismo, entre outros;
- Falhas Operacionais: eventos associados com falhas dos operadores decorrente de atividades indevidas durante operação;
- Falhas Mecânicas: eventos associados a defeitos ou mau funcionamento de válvulas, flanges, juntas, bem como desgaste ou fadiga do material;
- Falhas na Manutenção: eventos associados durante os trabalhos de manutenção das linhas;
- Corrosão: eventos associados à ação da corrosão.

De acordo com os dados obtidos, entre as causas que puderam ser apuradas, a maioria dos vazamentos dos dutos foram motivados por corrosão (17%) e pela ação de terceiros (11%), conforme ilustrado na Figura 2.2. Os resultados reforçam os argumentos de Hopkins de que para melhorar a segurança operacional de dutos é preciso manter um sistema de controle e gerenciamento da integridade do duto durante toda a sua vida útil. Isto significa que geralmente um duto não falha devido somente à corrosão, mas sim devido a uma falha do sistema de controle de corrosão por completo. A Figura 2.3 mostra um caso de oleoduto rompido devido corrosão no ano de 1990 na cidade de Campinas-SP.

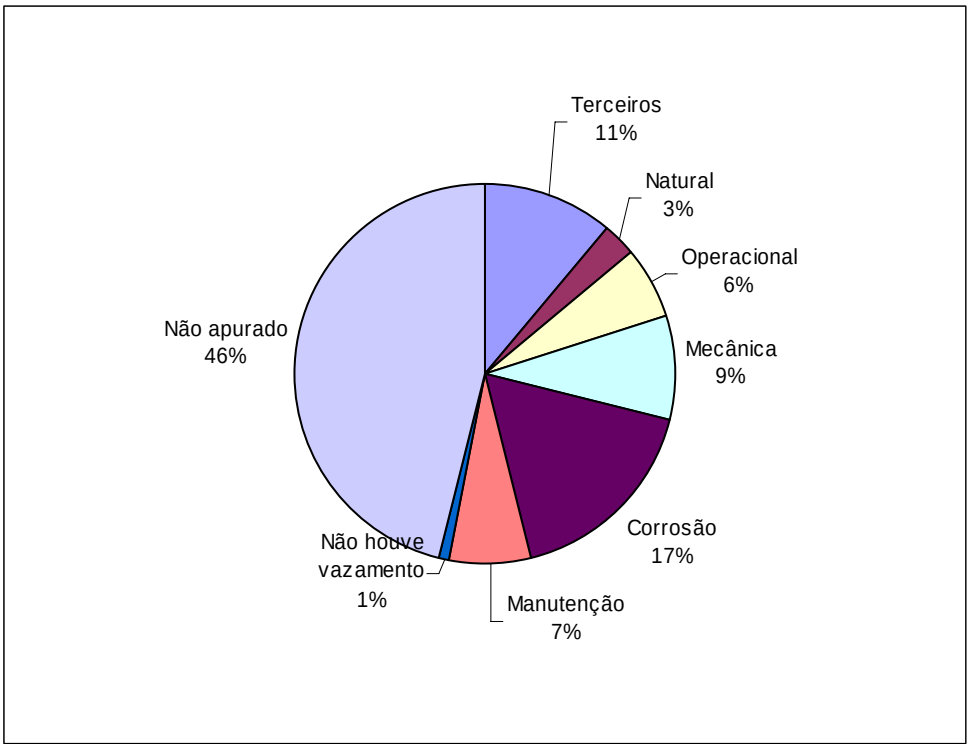


Figura 2.2 – Causas das ocorrências envolvendo acidentes com dutos no Estado de São Paulo: Total de 149 casos registrados pela CETESB.[2]

Figura 2.3 – Oleoduto rompido por corrosão em Campinas-SP.

2.2 Definição e Tipos de Defeitos

Um dos pré-requisitos para que um duto continue operando de forma segura é garantir um alto nível de confiabilidade da integridade estrutural do mesmo durante sua vida útil. Tal integridade pode ser ameaçada por defeitos introduzidos no duto durante o processo de fabricação, instalação ou operação (MMS, 2000).

Um defeito é uma descontinuidade ou irregularidade do material ou geométrica que é detectada por inspeção de acordo com o requerimento de vários códigos e normas que definem os limites de rejeição de defeitos.

[2] fonte: http://www.cetesb.sp.gov.br/emergencia/acidentes/dutos/aa_causas.asp.

Um defeito é considerado não aceitável quando sua magnitude for suficiente para garantir a rejeição baseada nos requerimentos de códigos, normas ou outros métodos usados para avaliação do defeito (MMS, 2000).

Os principais defeitos encontrados em dutos podem ser agrupados em três categorias de acordo com suas causas, conforme descrito logo abaixo (MMS, 2000).

a) Defeitos de Corrosão

- Corrosão Uniforme ou Generalizada: Perda uniforme ou gradual da espessura de parede do duto ao longo de uma extensa área.
- Corrosão por "Pite": Corrosão localizada, com grandes profundidades que diminuem consideravelmente a espessura do duto.
- Trincamento sob Tensão em meio corrosivo: Acontece quando um material, submetido à tensões de tração (aplicadas ou residuais), é colocado em contato com um meio corrosivo específico.
- Fissuração por Hidrogênio: Ocorre quando o hidrogênio migra para o interior do material e acumula-se em falhas existentes, provocando falha a baixos níveis de tensão.

b) Danos Mecânicos

- Mossa ou amassamento ("Dent"): Mossa causada por um evento que produz uma variação visível na curvatura da parede do duto ou componente sem que ocorra variação na espessura de parede do duto.
- Rasgos Superficiais ("Gouge"): Imperfeição na superfície causada pela remoção mecânica de material ou deslocamento de material provocando a redução da espessura de parede do duto.
- Ranhuras ("Groove"): Uma ranhura pode causar concentração de tensões em um determinado ponto podendo assim ser considerado um defeito.
- Trincas Superficiais: Trincas geradas na superfície do duto.

c) Defeitos de Soldagem

- Penetração Incompleta: Quando a raiz da junta a ser soldada não é fundida e preenchida completamente.
- Fusão Incompleta: Refere-se à ausência da união por fusão entre passes adjacentes da solda e o metal de base.
- Inclusões de Impurezas ("Slag"): Ocorre quando partículas de óxido e outros sólidos não-metálicos ficam aprisionados entre passes de solda ou entre a solda e o metal de base.
- Porosidade: A porosidade é formada pela evolução de gases, na parte posterior da poça de fusão, durante a solidificação da solda.
- Mordedura ("Undercut"): São reentrâncias agudas formadas pela ação da fonte de calor do arco entre um passe de solda e o metal de base ou um outro passe adjacente.

A Figura 2.4 e a Figura 2.5 ilustram alguns exemplos de defeitos de corrosão, mecânicos e de soldagem comumente encontrados em dutos.

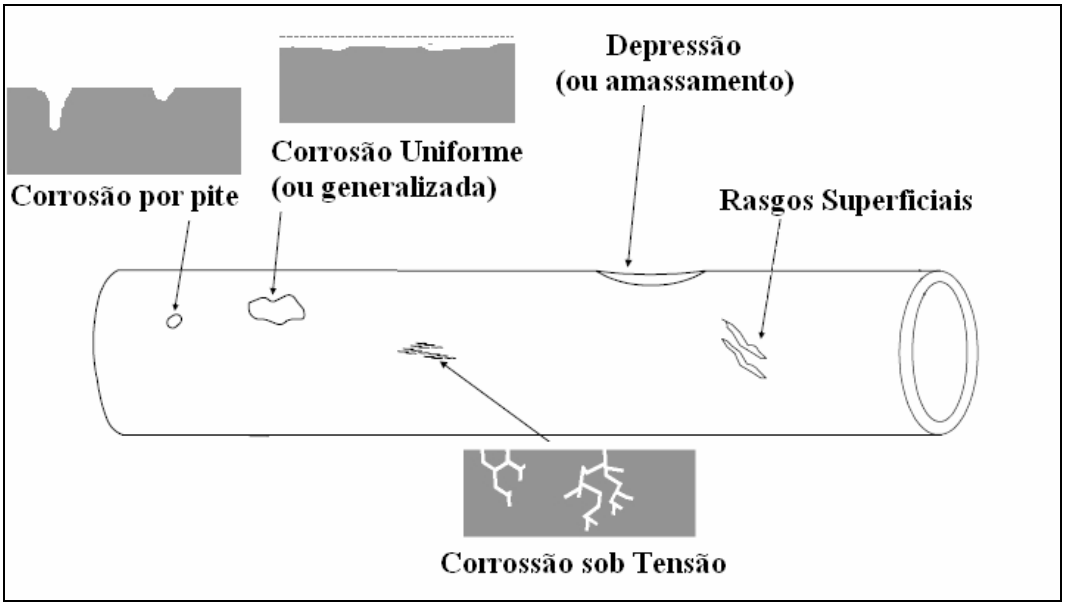


Figura 2.4 – Exemplos de tipos de defeitos de corrosão e danos mecânicos em dutos.[3]

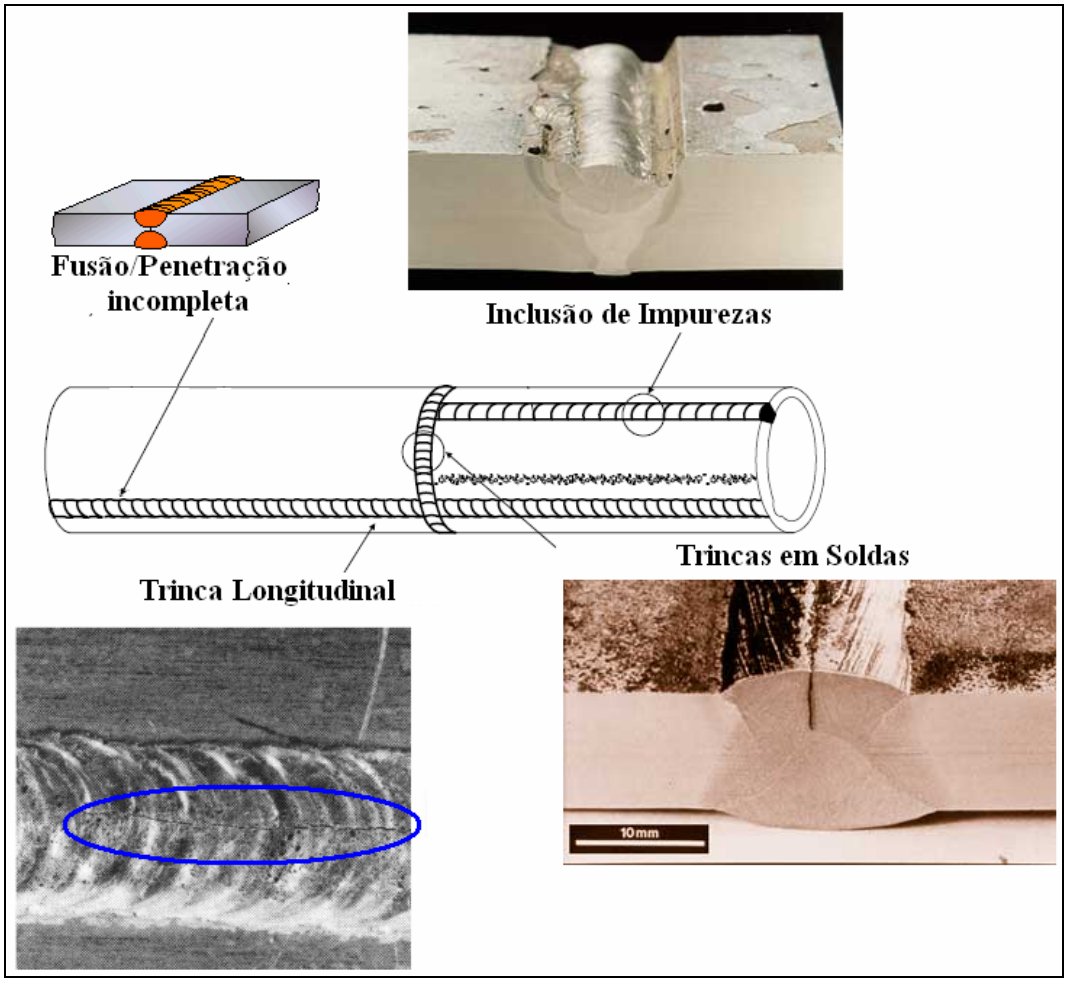


Figura 2.5 – Exemplos de defeitos de soldagem.

[3] fontes: http://www.corrosion-club.com/forms.htm, http://www.twi.co.uk/j32k/protected/band_3/jk40.html e http://www.esabna.com/EUWeb/MIG_handbook/592mig10_1.htm e MMS (2000).

2.3 Métodos de Controle da Corrosão

Em seu princípio mais básico, a corrosão nada mais é, do que o processo inverso da metalurgia onde estruturas metálicas enterradas ou submersas tendem a retornar ao seu estado mineral, ou seja, trata-se da deterioração de metais e ligas por ação química do meio ambiente conforme ilustrado na Figura 2.6.

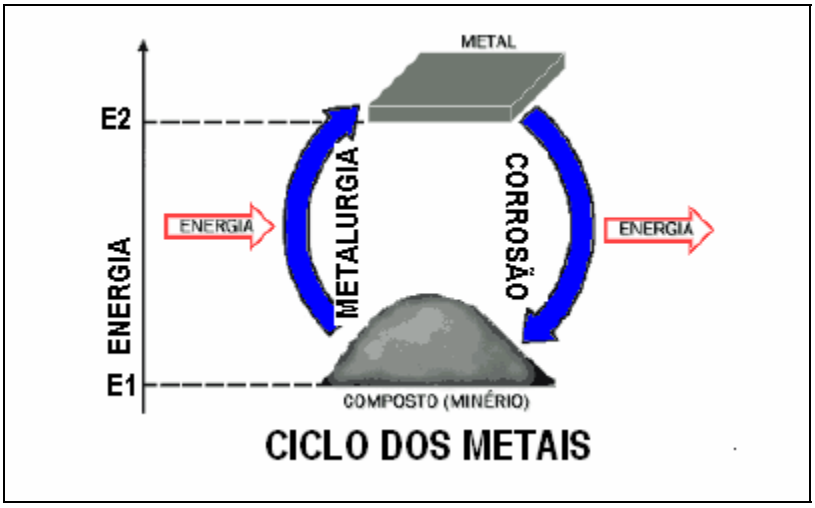


Figura 2.6 – Ciclo de energia dos metais ao passarem pelo processo de corrosão.[4]

Um duto enterrado (ou submerso na água) é essencialmente um pedaço de metal envolvido por um eletrólito (solo ou água). Ao longo do tempo, os potenciais elétricos podem variar de um ponto da tubulação para outro, como resultado da existência de áreas anódicas e catódicas. Estas áreas de diferentes potenciais elétricos são a base para a formação de uma célula de corrosão.

Para que uma célula de corrosão seja criada, é necessário que as seguintes condições sejam satisfeitas:

- Existência de um anodo e de um catodo;
- Existência de um potencial elétrico entre o anodo e o catodo;
- Existência de um caminho metálico conectando eletricamente o anodo e o catodo;
- O anodo e o catodo devem estar imersos num eletrólito eletricamente condutivo (solo ou água).

Uma vez presente estas condições, uma célula de corrosão é criada, uma corrente elétrica fluirá e metal será consumido no anodo. Se uma dessas quatro condições for removida, a corrosão é interrompida. No caso de um duto, o anodo, o catodo e o caminho elétrico metálico estão presentes na própria tubulação. A Figura 2.7 mostra de forma esquemática exemplos de células de corrosão (simples e múltiplas células) ao longo de um duto enterrado. Algumas áreas atuam como catodos, outras como anodo, e a tubulação entre elas age como um circuito conector. Dentre as várias razões pelas quais uma parte da tubulação atua como um anodo em relação à outra, podemos citar:

a) Duto novo e duto velho: quando um duto novo é inserido numa tubulação velha, ele assume o papel de anodo, pois seu potencial é maior que o do duto velho.

b) Corrosão resultante de solos dissimilares: o potencial natural de um metal pode variar com as diferenças na composição do solo (eletrólito).

[4] Fonte: www.abraco.org.br.

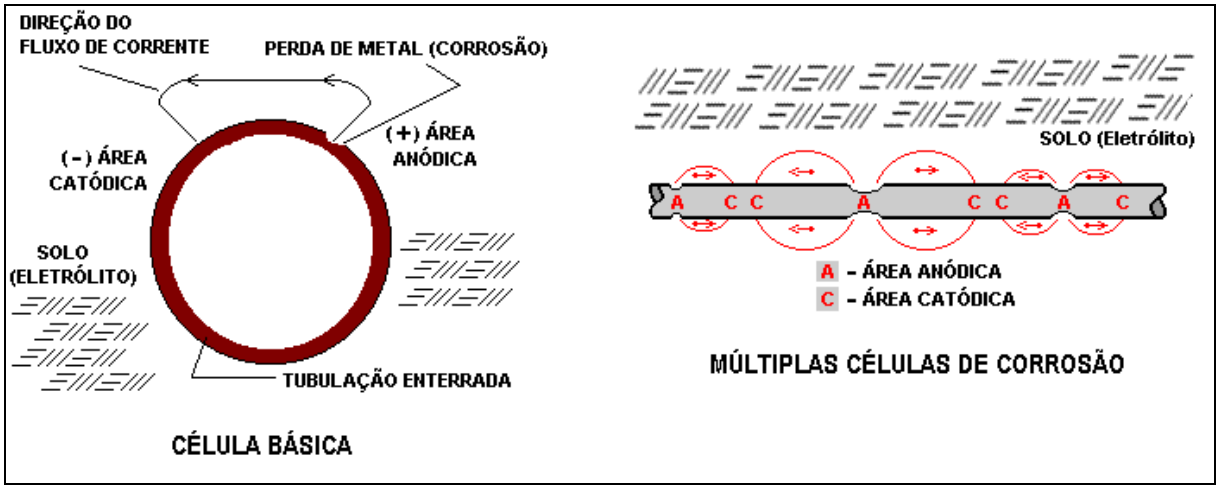


Figura 2.7 – Representação esquemática de células de corrosão em dutos.[5]

Existem várias técnicas empregadas para controle da corrosão. Dois dos principais métodos de proteção de dutos contra corrosão são a aplicação de revestimento (interno e/ou externo) e o uso de técnicas de proteção catódica, conforme descritos a seguir.

2.3.1 Revestimentos

Os dutos enterrados e submersos devem receber revestimento interno e/ou externo contra corrosão como uma forma de controlar a ação eletrolítica de correntes elétricas subterrâneas que caminham pela tubulação, em conseqüência das diferenças de potencial entre o duto e o solo, de um ponto para outro do próprio solo ou mesmo embaixo d'água. Dependendo da intensidade dessas correntes, a corrosão resultante é às vezes violenta, perfurando completamente a parede do duto em pouco tempo. Além da função anticorrosiva, os revestimentos também servem para atenuar possíveis danos mecânicos e isolar termicamente o duto.

Existem vários tipos de materiais usados como revestimentos para dutos, sendo alguns dos mais empregados os revestimentos com: esmalte de alcatrão de hulha (coal-tar), asfaltos, fitas plásticas (PVC, poliéster, polietileno), entre outros (Telles,1997).

2.3.2 Proteção Catódica

Um outro método complementar à proteção anticorrosiva do revestimento citado anteriormente é o uso de sistemas de proteção catódica que faz uso de uma corrente elétrica induzida com o objetivo de tornar toda a estrutura de aço (duto) em uma área catódica ao invés de anódica.

Existem dois métodos de executar a proteção catódica. O primeiro é chamado de Proteção por Corrente Impressa ou Forçada e é ilustrado na Figura 2.8. Neste caso, uma corrente elétrica (fornecida de uma fonte geradora de corrente contínua) é forçada à circular entre a estrutura (duto enterrado) e o anodo fazendo com que o anodo seja consumido ao invés do duto.

Outro método similar é a chamada Proteção Galvânica ou Proteção por Anodo de Sacrifício (Figura 2.9). Neste caso, o fluxo de corrente elétrica fornecido é originado devido à diferença de potencial existente entre o metal a ser protegido (duto) e o outro escolhido como anodo, com potencial mais negativo (Gentil, 2003).

[5] fonte: www.abraco.org.br.

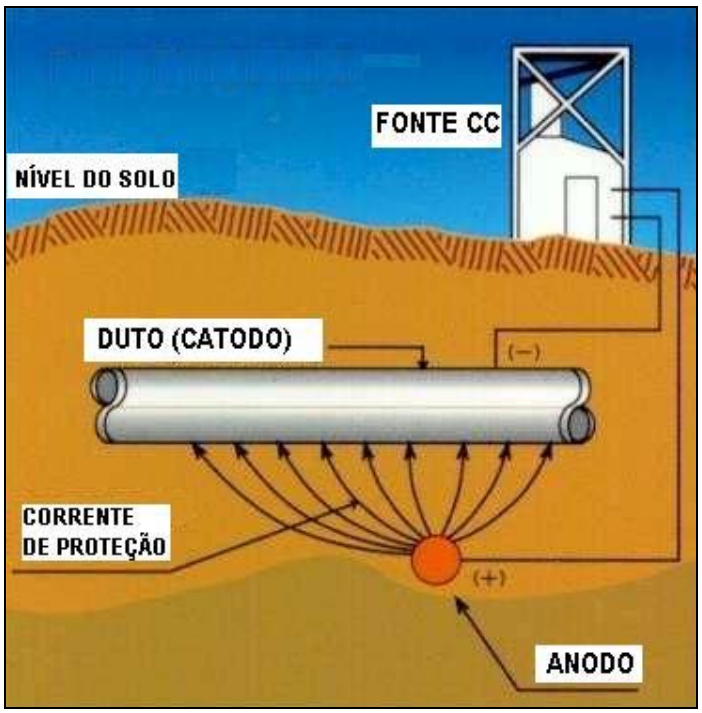


Figura 2.8 – Proteção catódica pelo método da Corrente Impressa (ou forçada).[6]

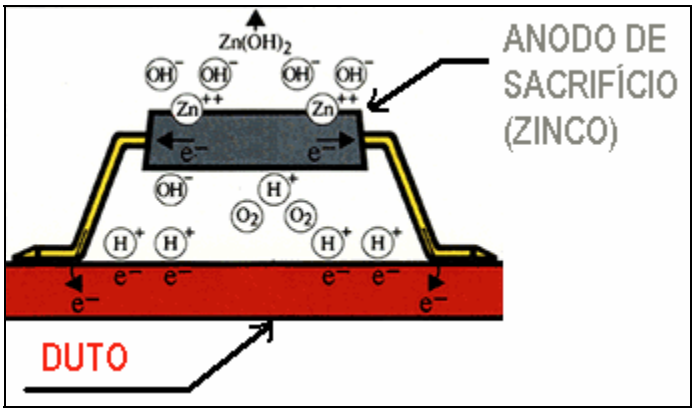


Figura 2.9 – Proteção catódica pelo método da Proteção Galvânica (Anodo de sacrifício).

2.4 Técnicas de Inspeção e Monitoramento de Dutos

Existem inúmeras técnicas de inspeção e monitoramento de dutos que aumentam a segurança e eficiência do sistema e minimizam potenciais fontes de acidentes e interrupções da linha. Essas técnicas incluem uma grande variedade de medidas que traduzem a condição atual de integridade do duto e do ambiente o qual está localizado, permitindo identificar, localizar e mapear possíveis defeitos no duto antes que se transformem em vazamentos ou causem grandes acidentes.

A maioria das modernas empresas possui suas redes de dutos automatizadas e controladas remotamente por sofisticados sistemas de controle que incluem três partes principais:

[6] fonte: adaptado de http://www.windsun.com/CP_Stuff/What_is_CP.htm.

- O sistema SCADA (Supervisão, Controle e Aquisição de Dados), que serve como cérebro para coletar, receber, processar dados e enviar sinais de comandos para controle de vários equipamentos de forma remota;
- Os meios de comunicação (satélites, torres de micro-ondas, cabos de fibra óptica, etc.), que interligam o sistema SCADA com as estações remotas (por exemplo, estações de bombeamento, válvulas, ect.);
- As unidades locais de controle (RTU-"Remote Terminal Units"), que enviam dados das estações locais para o sistema SCADA e recebem comandos do SCADA para controle remoto de equipamentos (por exemplo, fechamento de válvulas, etc.).

É possível inspecionar e avaliar o revestimento e a eficiência do sistema de proteção catódica do duto aplicando-se uma combinação de técnicas que permitem avaliar e mapear as perdas de corrente na proteção. Entre as técnicas mais utilizadas, podemos citar: CIS ("Close Interval Survey") também chamada de passo-a-passo, cuja técnica consiste na medição contínua dos potenciais duto/solo, medidos em espaços próximos; PCM ("Pipeline Current Mapper"), que se baseia na atenuação da corrente de proteção; e a técnica DCVG ("Direct Current Voltage Gradient"), que efetua leituras e analisa os gradientes de potencial no eletrólito (solo) determinando a direção do fluxo de corrente.

2.4.1 Inspeção por "Pigs"

"Pigs" são equipamentos que, inseridos dentro do duto, viajam por toda a sua extensão, impulsionados pela própria vazão do fluido podendo executar uma grande variedade de funções. Em geral, os "pigs" que realizam função de limpeza, separação de produtos, ou remoção de água são denominados de "Utility Pigs". Por outro lado, os "pigs" que fornecem informações das condições da linha (por exemplo, localização de amassamentos e ovalizações, detecção de vazamentos ou pontos onde há redução da espessura de parede do duto) são denominados "pigs" instrumentados, ou "smart pigs". Estes últimos, informam com boa precisão a localização e extensão de defeitos existentes no duto (Caldwell et al, 2001; Gentil, 2003; Tiratsoo, 1992).

A técnica de inspeção de dutos por "Pigs" é uma forma bastante utilizada para mapear defeitos causados pela corrosão em um duto ao longo dos anos. Sua grande vantagem é possibilitar a investigação em toda a extensão do duto, o que seria, usando outra técnica, inviável economicamente, no caso de dutos enterrados de grandes extensões (Gentil, 2003). A Figura 2.10 ilustra esquema de "pig" inserido dentro de um duto. A estrutura mecânica é composta por uma cápsula cilíndrica apoiada entre dois suportes de borracha. Dentro da cápsula estão os circuitos eletrônicos e as baterias do "pig". Os suportes de borracha mantêm a cápsula centralizada na tubulação. A pressão do fluido atua sobre o suporte traseiro e impulsiona o "pig" ao longo do duto.

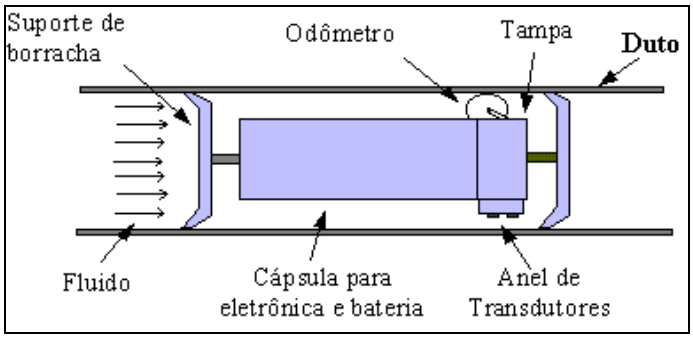


Figura 2.10 – Exemplo de "Pig" instrumentado inserido no duto.[7]

[7] fonte: http://www.poli.usp.br/Pig/descricao.html.

O tipo de "pig" mais comum para inspeção de defeitos de corrosão é o "pig" de perda de espessura (ou perda de massa) que é capaz de detectar e dimensionar pontos em que há redução da espessura de parede do duto e informar com boa precisão a localização destes defeitos. Existem duas técnicas diferentes para "pigs" de perda de espessura. A mais utilizada atualmente é a técnica de fuga de fluxo magnético (MFL- "Magnetic Flux Leakage"). Esta técnica consiste na geração de um campo magnético por meio de dois anéis circunferenciais contendo pólos norte e sul de um ímã. Este campo magnético é usado para saturar o duto. Ao encontrar qualquer defeito ou anomalia, as linhas de campo são desviadas para dentro e para fora do duto devido à menor espessura disponível para sua passagem (Gentil,2003). Esses defeitos são medidos por meio de anéis de sensores e suas dimensões armazenadas através do sistema "onboard" de armazenamento de dados do "pig". A Figura 2.11 ilustra exemplo de um "pig" para detecção de defeitos de corrosão fabricado pela companhia brasileira PipeWay.

Figura 2.11 – Exemplo de "pig" de perda de espessura (MFL) sendo inserido em um duto.[8]

A outra técnica utilizada é por meio de ultra-som. O "pig" ultra-sônico possui uma grande quantidade de cabeçotes que fazem a medição direta de espessura do duto de maneira a varrer toda a circunferência do mesmo.

Embora os "pigs" de perda de espessura magnéticos de alta resolução sejam bastante precisos, é necessário conhecer suas limitações ao analisar os dados. Tims & Wilson (2002) mostraram um caso no qual a presença de restos metálicos causaram falsos resultados de corrosão o que resultou numa substituição desnecessária do duto. Com isso, Tims & Wilson enfatizam que o uso dessa técnica pode conduzir a resultados mau interpretados em certas circunstâncias onde não há prática de outras técnicas complementares.

A decisão de qual método utilizar deve ser tomada com base nos recursos das duas técnicas, no tipo de corrosão esperada e em uma avaliação econômica. Em ambas as técnicas o engenheiro deve estar consciente de suas limitações em detectar e dimensionar defeitos.

[8] fonte: http://www.pipeway.com/cgi/cgilua.exe/sys/start.htm?infoid=116&op=pt&sid=35.

2.4.2 Mapeamento de Defeitos

Os "pigs" instrumentados realizam as medições (via fuga de fluxo magnético, por exemplo) e armazenam esses dados para posterior interpretação e caracterização do perfil do defeito. Os dados obtidos nas leituras feitas pelos "pigs" instrumentados são posteriormente processados por meio de softwares específicos e avaliados por analistas especializados que irão tentar diferenciar as anomalias encontradas observando as características do sinal. Os analistas fazem uso de experiência própria, "in-house" algoritmos e softwares para decidir o perfil de corrosão (tipo de defeito e dimensões), pois infelizmente, estas interpretações e análises não são normalizadas (Palmer-Jones et al, 2002).

Com o mapeamento do defeito, vários perfis podem ser montados. Segundo Kiefner & Vieth (1989), o perfil a ser considerado para a avaliação do defeito requer um julgamento baseado na experiência e quando isso não é possível, o procedimento mais recomendável é combinar todos os perfis e assumir o caso mais desfavorável.

Palmer-Jones et al (2002) apontam alguns problemas que podem surgir quando há interação entre defeitos contidos nos relatórios de inspeção por "pig". Os defeitos individuais são armazenados como "caixas" cujo comprimento, largura e profundidade são iguais ao máximo comprimento, largura e profundidade do defeito. No caso onde há a interação entre defeitos próximos, as caixas são agrupadas segundo regras de interação próprias de cada empresa. Por exemplo, as caixas de defeitos interagentes podem ser tratadas como um defeito simples cujo comprimento é igual à distância da primeira até a última "caixa" e cuja profundidade é assumida como sendo a maior profundidade entre as "caixas" existentes dentro do grupo, conforme Figura 2.12.

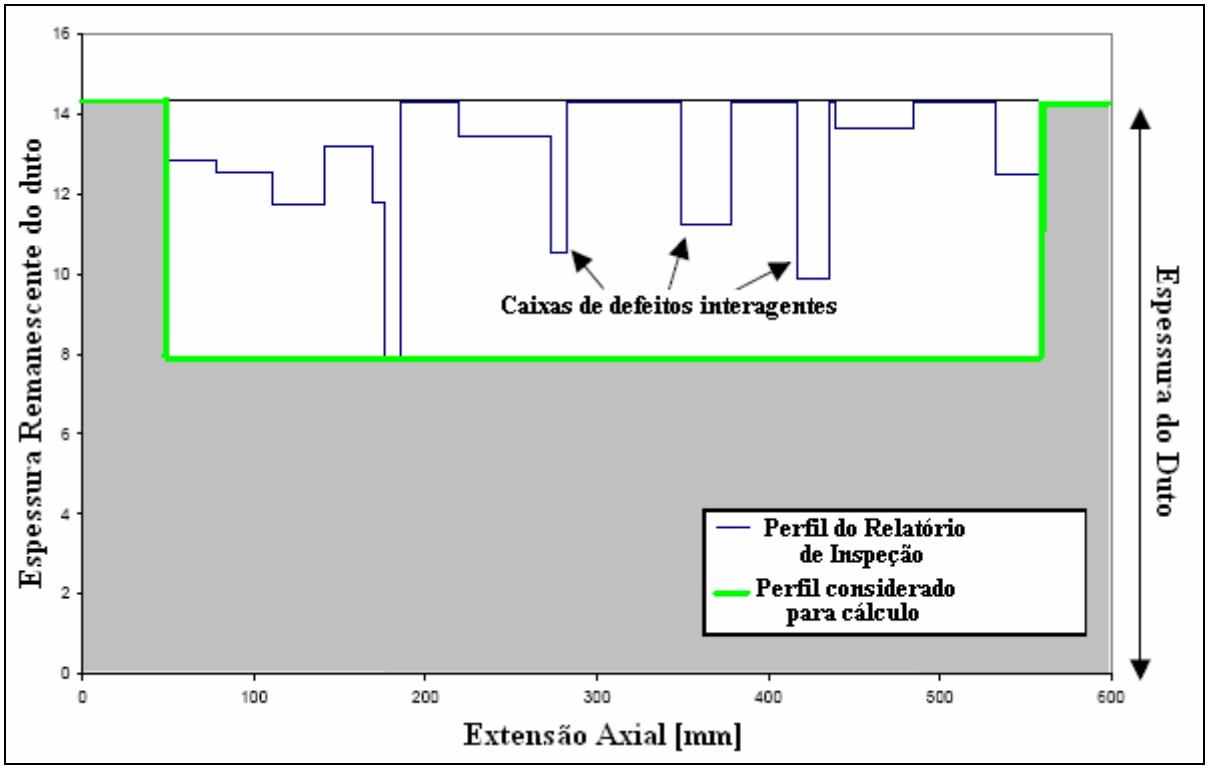


Figura 2.12 – Perfil de "caixas" de defeitos gerado pelo relatório de inspeção e perfil definido por regras de interação para posterior estimativa da pressão de ruptura.[9]

[9] fonte: Adaptado de Palmer-Jones et al (2002).

Este método é bastante utilizado e já está bem consolidado. No entanto, antigamente os "pigs" de inspeção só eram capazes de detectar defeitos com profundidades maiores que 10% ou 20% da espessura de parede do duto. Hoje em dia, os "pigs" são capazes de detectar defeitos muito rasos. Isto pode resultar num perfil de defeito extremamente conservador. Um exemplo disso é mostrado na Figura 2.13: a profundidade do defeito é considerada como sendo a maior profundidade das três caixas (A,B e C). No entanto, o comprimento total das três caixas é usado como sendo o comprimento do defeito, significando assim, que se trata de um defeito longo e profundo, onde na realidade existe apenas um defeito curto e profundo. Para o caso onde há a previsão de ruptura e não vazamento, a avaliação será bastante conservadora.

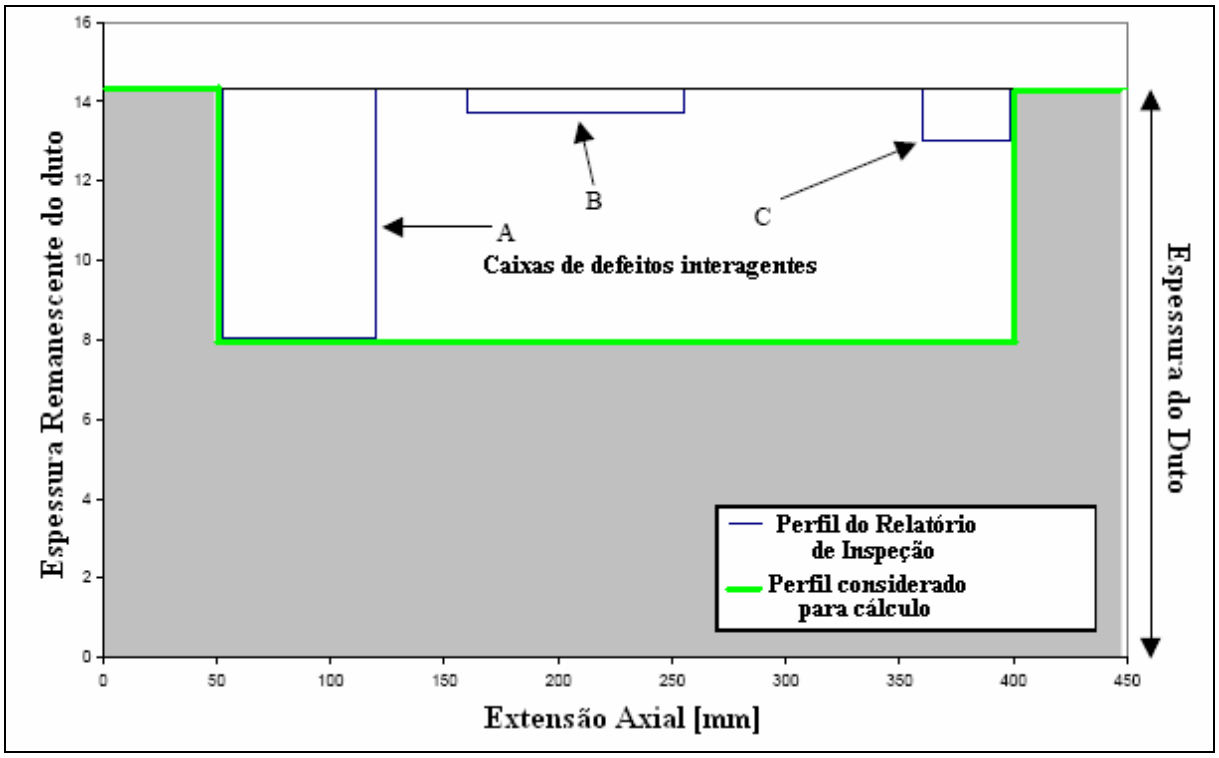


Figura 2.13 – Exemplos de perfil de defeito conservador gerado por regras de interação.[10]

2.5 Métodos de Avaliação da Resistência Residual de Dutos Corroídos

Ao receber um relatório de inspeção por "pig" instrumentado com uma lista de defeitos caracterizados sob forma e dimensão, o analista precisa ter uma metodologia para avaliá-los e saber se aquela configuração do defeito é aceitável ou não para que o duto continue operando de forma segura (Souza 2003; Costa, 2004). Para isso, utilizam-se normas, códigos e recomendações desenvolvidas por grandes empresas do setor como a Det Norske Veritas (DNV, 1999), ou são desenvolvidos procedimentos para analisar defeitos específicos para atender à necessidade de uma determinada empresa (Noronha Jr et al., 2002; Benjamin & Andrade, 2003a). Estes procedimentos semi-empíricos desenvolvidos ainda são bastante empregados, no entanto, com o passar dos tempos, testes em escala real têm sido feitos para comparar as normas existentes e propor correções quando necessário. Além disso, métodos mais sofisticados estão cada vez mais sendo aplicados como meio de avaliação da resistência de dutos corroídos.

[10] fonte: Adaptado de Palmer-Jones et al (2002).

Entre eles, podemos citar análise numérica 3-D não-linear via método dos elementos finitos (tema desta dissertação), análise limite (via MEF), análise probabilística, análise de risco e análise de confiabilidade.

2.5.1 Avaliação de Defeitos por Níveis de Complexidade

Dependendo do método de cálculo escolhido, a avaliação da resistência residual do duto corroído poderá se tornar relativamente simples, extremamente sofisticado ou até mesmo inviável do ponto de vista econômico.

Uma boa prática é avaliar os defeitos em níveis crescentes de complexidade de análise. O método a ser usado depende do objetivo da avaliação, do tipo de defeito, das condições de carregamento e da qualidade de dados disponível.

Cosham & Hopkins (2001) propuseram no "The Pipeline Defect Assessment Manual (PDAM)" a avaliação de defeitos por níveis de complexidade, que pode ser aplicado para defeitos de corrosão. A Figura 2.14 resume os cinco diferentes níveis de avaliação de defeitos e seus respectivos dados necessários.

- Nível 1: Normas internas de empresas operadoras ou regras práticas para aprovar ou reprovar defeitos de corrosão com informações apenas do tipo do defeito e dimensões.

- Nível 2: Neste nível de análise, utiliza-se os métodos de fácil aplicação e que em muitas situações podem apresentar resultados excessivamente conservadores. Podemos citar métodos tais como o ASME B31.G, RSTRENG 0,85dL, RPA, DNV RP-F101 (para defeitos isolados) e BS-7910 (para defeitos isolados). Para aplicar estes métodos, é preciso conhecer o comprimento e a maior profundidade do defeito. Além disso, é necessário conhecer o grau do aço, o diâmetro e a espessura do duto.

- Nível 3: Neste nível de análise, além dos dados até agora citados, é necessário conhecer o perfil de corrosão do defeito. Os principais métodos que podem ser aplicados neste nível são o "Effective Area", o DNV RP-F101 (para defeitos de geometria complexa). Existem programas comerciais restritos para uso das companhias que permitem facilmente aplicar estes métodos cuja maior dificuldade é a obtenção do perfil de corrosão do defeito. Neste nível de classificação proposto por Cosham & Hopkins (2001), podemos incluir também a norma BS 7910 (para defeitos interagentes) a qual requer o conhecimento do espaçamento axial e angular entre os defeitos e as larguras dos mesmos.

- Nível 4: Este nível consiste em realizar análise não-linear de elementos finitos ou executar testes experimentais em escala real para o problema. A utilização do método dos elementos finitos, além de depender de pessoal extremamente qualificado, exige muito tempo no processo de modelagem e determinação da solução do problema. Reproduzir o defeito em escala real e realizar testes destrutivos é também uma alternativa viável, embora demande bastante tempo, pessoal qualificado e aparelhagem adequada.

- Nível 5: Neste nível, a análise requer mais dados do duto em relação aos demais métodos. É necessário ter a distribuição estatística da geometria (defeito e duto), das propriedades do material e entre outras para quantificar as incertezas embutidas na avaliação e, quando conjugadas com a análise de risco, subsidiar a tomada de decisão em aceitar ou não um defeito.

Geralmente, a avaliação dos defeitos é conduzida até o nível 3. Caso o defeito seja reprovado até este ponto, é necessário a utilização de métodos mais sofisticados de avaliação do

defeito (nível 4 ou nível 5), redução da pressão de operação ou, em último caso, há a necessidade de reparo do duto.

A possibilidade de se avaliar os defeitos de corrosão com mais de um tipo de método é bastante grande e poderá ser uma ferramenta poderosa para que haja uma redução no número de reparos desnecessários resultando, portanto, numa imensa economia de recursos financeiros, sem comprometer a segurança do duto.

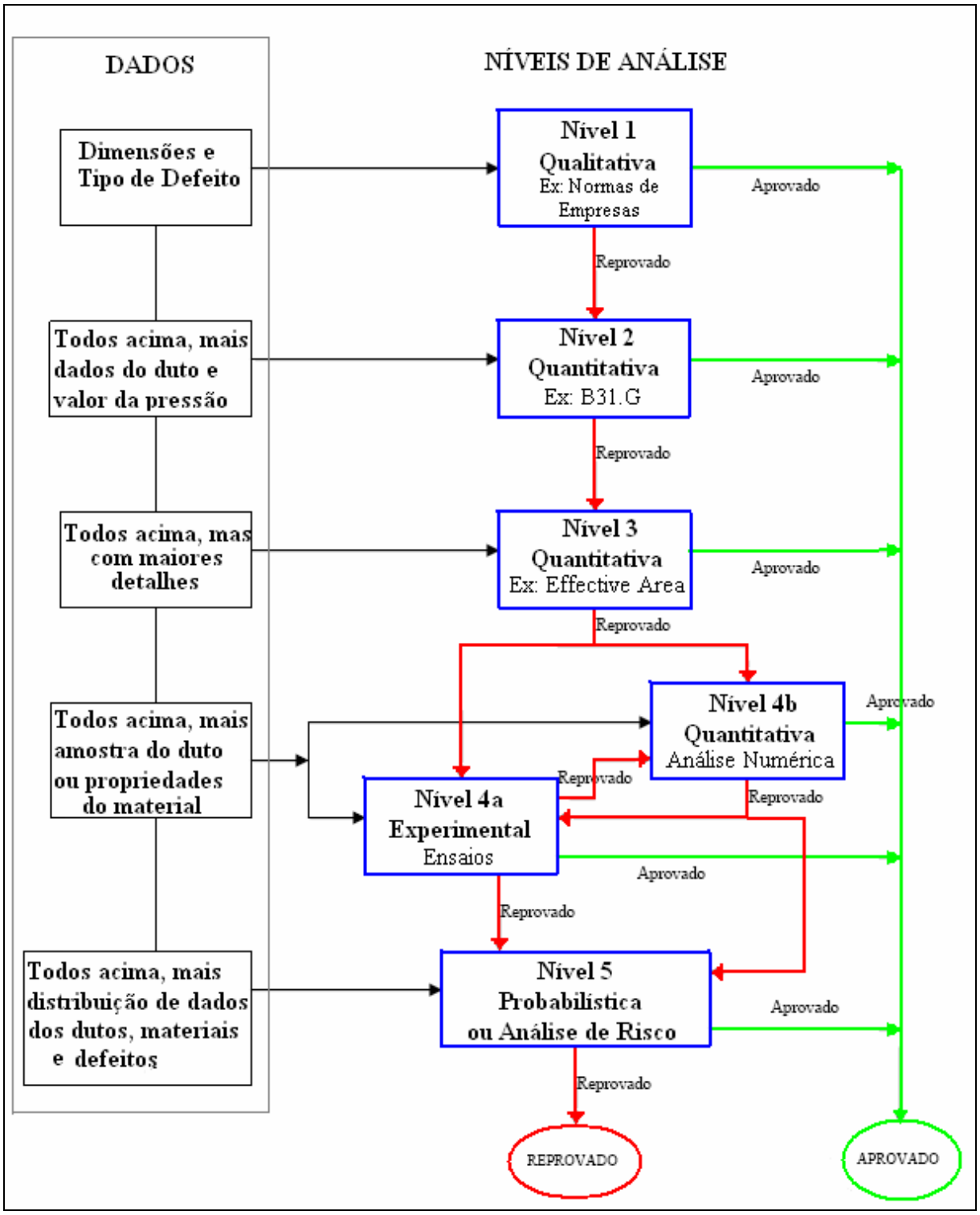


Figura 2.14 – Métodos de avaliação de defeitos por níveis de complexidade. [11]

[11] fontes: Adaptado de Cosham & Hopkins (2001) e Souza (2003).

2.6 Métodos Semi-Empíricos

2.6.1 Introdução

Os principais métodos existentes para avaliação de defeitos de corrosão em dutos utilizam conceitos da Mecânica da Fratura que, modificados por dados empíricos, resultam em expressões semi-empíricas que, se aplicadas dentro de seus limites de validação, permitem estimar a pressão de ruptura de dutos com defeitos.

É importante lembrar que estes métodos de avaliação, denominados "fitness-for-purpose"[12], devem ser abordados holisticamente. Isto significa que todos os aspectos da integridade de um duto devem ser considerados, não tratando-se simplesmente em executar um procedimento mecânico de inserção de dados das dimensões do defeito (obtidos via inspeção por "pig") em uma equação para simples estimativa da pressão de ruptura (Palmer-Jones et al, 2002).

Pesquisadores do Battelle Memorial Institute, sob supervisão do comitê de pesquisa da AGA (American Gas Assotiation), estudaram de forma pioneira e extensiva a falha de defeitos em dutos de aço através de estudos teóricos e ensaios experimentais em escala real. O principal objetivo desses estudos era determinar, de forma quantitativa, a relação entre o valor da pressão de ruptura (obtidos de testes hidrostáticos) com o número e tamanho dos defeitos (Cosham & Hopkins, 2002).

A equação gerada neste estudo ficou conhecida como NG-18 Surface Flaw Equation a qual formou a base dos métodos subseqüentes tais como ASME B31G, RSTRENG e DNV RP-F101. A equação tem a seguinte forma:

$$\sigma_{rup} = \sigma_{flow} \cdot \left[\frac{1 - \frac{A}{A_0}}{1 - \frac{A}{A_0} M^{-1}} \right] \tag{2.1}$$

onde:

σ_{rup} - Tensão circunferencial da parede do duto no instante da ruptura, numa região fora do defeito.

σ_{flow} - Tensão de escoamento média do material ("flow stress").

A - Área longitudinal de material perdido.

A_0 - Área longitudinal original da região corroída.

M - Fator de dilatação ("bulging factor" ou fator de Folias)

O fator de dilatação M ("Folias factor") foi criado para levar em consideração a influência da deformação, em forma de uma bolha, na tensão circunferencial que está sendo aplicada na região corroída. O fator de dilatação é expresso por:

$$M = \sqrt{1 + 0,6275 \cdot \left(\frac{L^2}{D \cdot t} \right) - 0,003375 \cdot \left(\frac{L^2}{D \cdot t} \right)^2} \tag{2.2}$$

[12] Nota: "Fitness-for-purpose" são métodos que se baseiam no princípio de que, uma determinada estrutura (neste caso, o duto) está adequada para seu uso, se as condições para que ocorra uma falha não sejam atingidas (Anon, 1999).

Onde:
L - Comprimento longitudinal do defeito.
D - Diâmetro externo do duto.
t - Espessura de parede do duto.

A tensão de escoamento média (σ_{flow}) foi definida como uma tensão compreendida entre a resistência ao escoamento e a resistência à tração do material, ou seja, $\sigma_{esc} < \sigma_{flow} < \sigma_u$.

A área longitudinal "A" de material perdido (Figura 2.15a) pode ser determinada por meio da técnica de projeção (Cronin & Pick, 2000a; Souza et al, 2005), na qual o ponto de maior perda de espessura de cada linha circunferencial de dados (dentro do defeito) é projetado no plano longitudinal que corta a parede do duto.

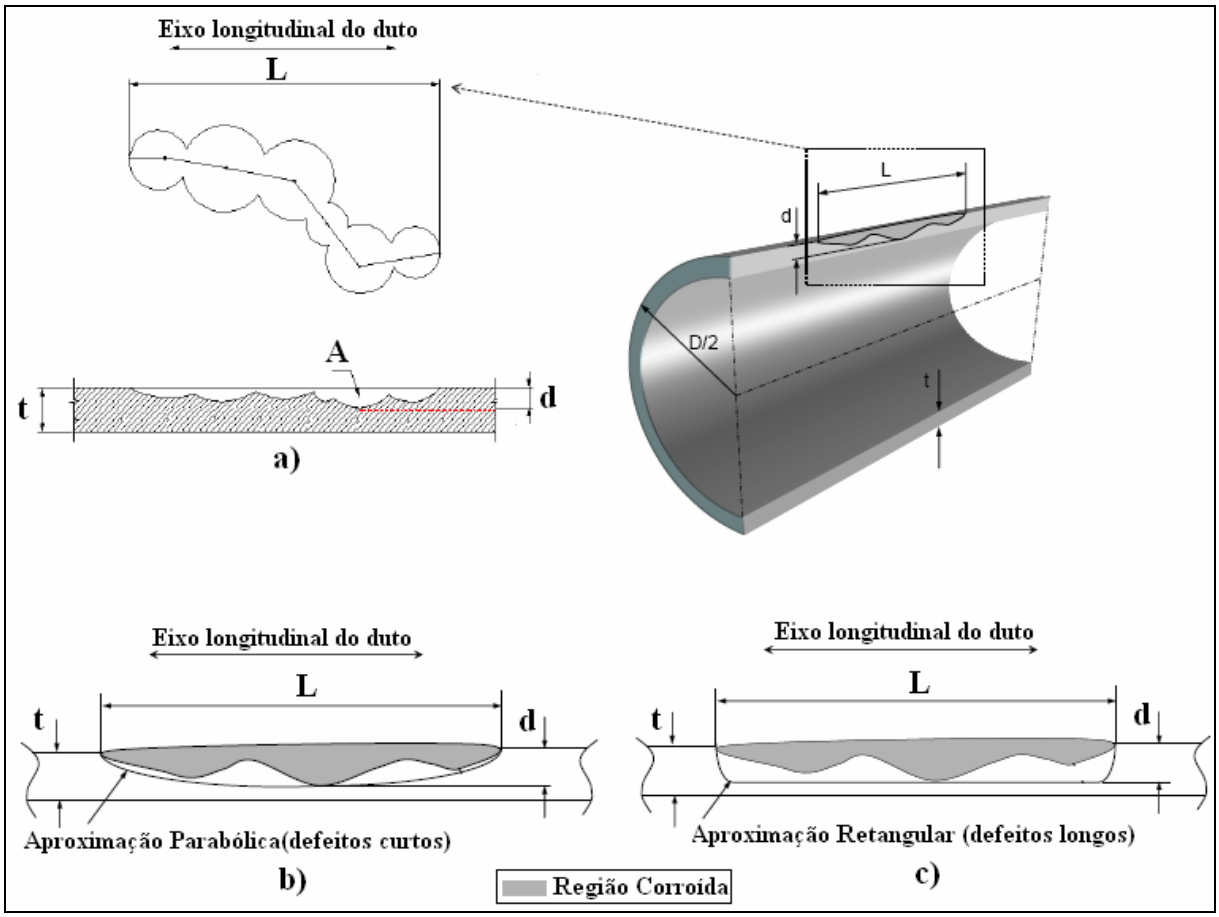


Figura 2.15 – Ilustração típica de formas de aproximação de defeitos de corrosão em dutos: a) Técnica da projeção, b) Aproximação para defeitos curtos (forma parabólica), c) Aproximação para defeitos longos (forma retangular).[13]

A área "A" do material perdido também pode ser aproximada segundo um perfil parabólico (para defeitos curtos) ou segundo um perfil retangular (para defeitos longos), conforme ilustrado na Figura 2.15b e Figura 2.15c, respectivamente.

[13] fonte: Adaptado de Adib-Ramezani et al (2006).

A tensão circunferencial numa região fora do defeito, submetido à pressão interna, pode ser facilmente calculada pela fórmula de Barlow, dada por:

$$\sigma_{circ} = P \cdot \left(\frac{D}{2 \cdot t} \right) \tag{2.3}$$

onde:

σ_{circ} - Tensão Circunferencial ("hoop stress").

P - Pressão interna atuante no duto.

Considerando-se o estado limite, em que "P" é a pressão de ruptura (P_{rup}), então, conseqüentemente, $\sigma_{circ} = \sigma_{rup}$. Daí, temos a seguinte igualdade:

$$P_{rup} = \sigma_{flow} \cdot \left(\frac{2 \cdot t}{D} \right) \cdot \left[\frac{1 - \frac{A}{A_0}}{1 - \frac{A}{A_0} M^{-1}} \right] \tag{2.4}$$

A equação (2.4), mostra que a pressão de ruptura depende de três parcelas. A primeira, relacionada com as características do material (σ_{flow}); a segunda, relacionada com as características geométricas do duto ($\frac{2 \cdot t}{D}$) e a terceira, relacionada com as características do defeito (f_R).

Onde: $f_R = \left[\frac{1 - \frac{A}{A_0}}{1 - \frac{A}{A_0} M^{-1}} \right]$

Esta última parcela, denominada de fator de redução (f_R), é bastante influenciada pelo valor de M. Quando M assume valores próximos à unidade, o fator de redução tende também para um. Quando M tende para infinito, o fator de redução é dado por: $f_R = 1 - \frac{A}{A_0}$.

A equação (2.4) forma a base para os métodos tais como ASME B31G, RSTRENG 0,85dL, RPA(Benjamin & Andrade, 2003a), DNV RP-F101 (Parte B) e BS 7910 (este dois últimos, para defeitos isolados). De acordo com o procedimento de análise por níveis de complexidade proposto por Cosham & Hopkins (2001), todos estes métodos são classificados como Nível 2. Dentre os métodos classificados como Nível 3, podemos citar o método DNV RP-F101 para defeitos interagentes ou de geometria complexa (Parte B), RSTRENG "Effective Area", o método WDD "Weighted Depth Difference" (Cronin & Pick, 2000b) e a norma BS 7910 para defeitos interagentes. A Tabela 2.1 apresenta uma lista com alguns dos principais métodos disponíveis para avaliação de defeitos de corrosão em dutos. Os métodos estão agrupados verticalmente de acordo com o tipo de defeito abordado (isolado ou interagente), e horizontalmente de acordo com o tipo de carregamento aplicado (pressão interna, carregamento combinado, etc.).

Tabela 2.1 – Principais métodos semi-empíricos para avaliação de defeitos de corrosão em dutos.[14]

NORMAS/ MÉTODOS	Apenas Pressão Interna		Carregamento Combinado
	Comprimento Longitudinal profundidade	**Área Longitudinal e profundidade**	**Pressão e compressão axial**
Defeitos Isolados	ASME B31G		
	DNV RP-F101	DNV RP-F101	DNV RP-F101
	BS 7910		
	RSTRENG 0,85dL	RSTRENG "Effective Area"	
	Método RPA	WDD Method	
	Apenas Pressão Interna		
Defeitos Interagentes[15]	DNV RP-F101	DNV RP-F101	
	BS 7910		

A maioria desses métodos não leva em consideração o comprimento circunferencial do defeito. Assim, defeitos alinhados circunferencialmente irão se superpor após aplicação de alguma técnica de projeção no plano longitudinal. Nestas técnicas, apenas tensões circunferenciais devido à pressão (normal ao plano de projeção) são consideradas, sendo impossível avaliar o efeito de tensões longitudinais devido ao carregamento de extremidade e flexão.

O método ASME B31G, foi concebido a partir da equação semi-empírica NG-18 "Surface Flaw Equation"(Eq. 2.1). Ele fornece, para alguns casos, resultados com um elevado grau de conservadorismo, isto é, os valores estimados para a pressão de ruptura são excessivamente baixos, o que leva à remoção de vários dutos ainda em condições de serem mantidos em operação.

Algumas modificações nas equações do método B31G foram feitas com o objetivo de diminuir o seu grau de conservadorismo. Dentre as modificações, podemos citar a definição de novos fatores de "Folias" e a tensão de escoamento, além de uma consideração mais detalhada da geometria da corrosão. Estas modificações foram implementadas no programa RSTRENG de onde surgiram os métodos RSTRENG 0,85dL e RSTRENG "Effective Area"(Kiefner & Vieth, 1989).

Estudos experimentais realizados pela PETROBRAS (Benjamin et al, 2000) mostraram que o método RSTRENG 0,85dL fornece resultados não-conservadores para defeitos longos de profundidade uniforme. Benjamin & Andrade (2003a), propuseram uma versão modificada para o método 0,85dL que fornece resultados adequadamente conservadores para o caso de defeitos longos. Este novo método, denominado Método RPA ("Rectangular Parabolic Area") ou Método 0,85dL Modificado, usa duas diferentes equações para estimar a pressão de falha de duto com defeito de corrosão. Uma equação, que é a mesma usada no método RSTRENG 0,85dL, é utilizada para defeitos curtos (defeitos nos quais $L \le \sqrt{20 \cdot D \cdot t}$).

[14] Fonte: Adaptado de MMS (2000).

[15] Nota: Para esses tipos de defeitos, é necessário também conhecer o espaçamento axial/angular entre os defeitos e a largura dos mesmos.

A outra equação, utilizada para defeitos longos (defeitos nos quais $L > \sqrt{20 \cdot D \cdot t}$), é uma versão modificada da primeira equação e considera a área longitudinal de material perdido como sendo uma composição das formas retangular e parabólica.

Novos métodos de avaliação da resistência residual de dutos com defeitos de corrosão, submetidos a carregamento de pressão interna, foram desenvolvidos através de um projeto denominado "Line Pipe Corrosion Group Project" elaborado pela British Gas Technology (atualmente Advantica). Este projeto consistiu na execução de mais de 70 ensaios de pressão, em escala real, em dutos contendo defeitos de corrosão usinados (para simular a corrosão), incluindo defeitos isolados, defeitos interagentes com outros e defeitos de forma complexa. Durante este projeto, foram também realizadas extensivas análises tridimensionais, não-lineares de elementos finitos (via software comercial ABAQUS), considerando carregamento de pressão interna e material elasto-plástico. Estes estudos resultaram no desenvolvimento de um método de avaliação de defeitos de corrosão em dutos, que foi, posteriormente, incorporado no Anexo G[16] da norma britânica BS-7910 (BS 7910, 1999).

Além do projeto acima mencionado, surgiu também outro projeto denominado "Joint Industry Project" elaborado pela DNV (Det Norske Veritas) em cooperação com a British Gas Technology e o patrocínio de onze empresas/organizações internacionais, dentre elas a PETROBRAS. Neste projeto, foi elaborada uma metodologia para avaliação de defeitos de corrosão (isolados, interagentes e de geometria complexa) em dutos considerando além da pressão interna, carregamento axial e flexão. O estudo foi realizado a partir de vários ensaios de pressão e análises não-lineares por elementos finitos. O resultado deste outro projeto gerou o documento publicado em 1999, chamado de "Recommended Practice RP-F101 Corroded Pipelines" (DNV,1999).

Em seguida, faz-se uma breve descrição de alguns destes métodos, ressaltando, sempre quando possível, as condições de aplicação e as principais diferenças entre eles.

2.6.2 ASME B31G

Este método foi concebido a partir da equação semi-empírica NG-18 "Surface Flaw Equation". O emprego deste método está limitado às várias condições, entre elas, podemos destacar:

- Apenas carregamento de pressão interna;
- Defeitos de corrosão com profundidade compreendida entre 10% e 80% da espessura de parede do duto;

  As considerações feitas para a sua aplicação são as seguintes:
- A tensão circunferencial no defeito no instante da ruptura é igual à tensão de escoamento média(σ_{flow}), que é dada por:

$$\sigma_{flow} = 1{,}1 \cdot \sigma_{esc} \tag{2.5}$$

  onde: σ_{esc} - tensão de escoamento do material.
- O defeito é considerado curto se $L \leq \sqrt{20 \cdot D \cdot t}$ e longo se $L > \sqrt{20 \cdot D \cdot t}$, onde L é o comprimento do defeito;
- A área original (A_0) da região corroída é calculada como:

$$A_0 = L \cdot t \tag{2.6}$$

- A área corroída (A), pode ser expressa por:

[16] Nota: Como é necessário a posterior validação do método, a parte que aborda defeitos de geometria complexa não foi inclusa no Anexo G da norma BS 7910 (Wiesner et al, 2000).

$$A = \alpha \cdot L \cdot d \tag{2.7}$$

onde, "α" é a constante que define a forma geométrica adotada para representar a área de material perdido e "d" é a profundidade máxima do defeito.

A área de material perdido é representada de duas formas no método B31G: forma de parábola para defeitos curtos ($\alpha = 2/3$) ou forma retangular para defeitos longos($\alpha = 1$), conforme ilustrado na Figura 2.16.

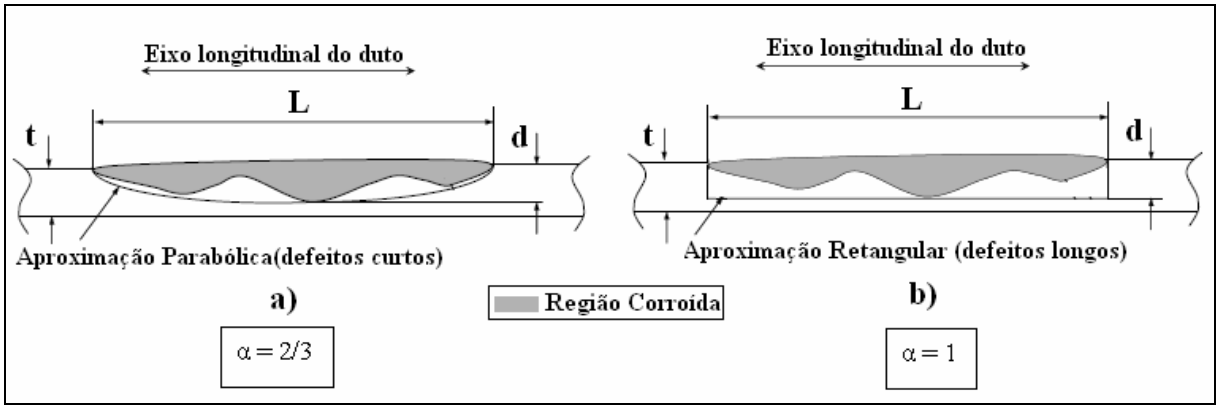


Figura 2.16 – Representação da área longitudinal de material perdido: a) forma parabólica e b) forma retangular.[17]

Substituindo as equações (2.5), (2.6) e (2.7) na Eq.(2.4), tem-se, as expressões da pressão de ruptura para o caso de defeitos curtos e longos com os respectivos fatores de dilatação (M), conforme expressões abaixo:

- Pressão de ruptura para defeitos curtos ($L \le \sqrt{20 \cdot D \cdot t}$ e $\alpha = 2/3$):

$$P_{rup} = 1{,}1 \cdot \sigma_{esc} \cdot \left(\frac{2 \cdot t}{D}\right) \cdot \left[\frac{1 - \left(\frac{2}{3}\right)\frac{d}{t}}{1 - \left(\frac{2}{3}\right)\frac{d}{t} M^{-1}}\right] \tag{2.8}$$

onde,

$$M = \sqrt{1 + 0{,}8 \cdot \left(\frac{L^2}{D \cdot t}\right)}$$

- Pressão de ruptura para defeitos longos ($L > \sqrt{20 \cdot D \cdot t}$, $\alpha = 1$ e $M \to \infty$):

$$P_{rup} = 1{,}1 \cdot \sigma_{esc} \cdot \left(\frac{2 \cdot t}{D}\right) \cdot \left[1 - \frac{d}{t}\right] \tag{2.9}$$

2.6.3 RSTRENG

Kiefner & Vieth(1989) propuseram dois novos métodos (0,85dL e "Effective Area") através do Projeto PR-3-805 do Pipeline Research Commitee da AGA (American Gas Assotiation) em parceria com o Instituto Battelle.

[17] Fonte: Adaptado de Adib-Ramezani et al (2006).

Fez parte também deste projeto, o desenvolvimento do programa computacional RSTRENG que calcula a pressão de ruptura de dutos com defeitos pelos métodos ASME B31G, RSTRENG 0,85dL e RSTRENG "Effective Area".

a) RSTRENG 0,85dL

Este método é uma versão modificada do método ASME B31G cujas principais diferenças entre estes dois métodos são mostradas na Tabela 2.2.

Tabela 2.2 – Principais diferenças entre os métodos ASME B31G e RSTRENG 0,85dL.

	ASME B31G	**RSTRENG 0,85dL**
Faixa de Aplicação	$0,1 \cdot t < d < 0,8 \cdot t$	$0,2 \cdot t \le d \le 0,8 \cdot t$
σ_{flow}	$\sigma_{flow} = 1,1 \cdot \sigma_{esc}$	$\sigma_{flow} = \sigma_{esc} + 69MPa$
	Defeitos Curtos	
L	$L \le \sqrt{20 \cdot D \cdot t}$	$L \le \sqrt{50 \cdot D \cdot t}$
α	$\alpha = 2/3$	$\alpha = 0,85$
M	$M = \sqrt{1 + 0,8 \cdot \left(\frac{L^2}{D \cdot t}\right)}$	$M = \sqrt{1 + 0,6275 \cdot \left(\frac{L^2}{D \cdot t}\right) - 0,003375 \cdot \left(\frac{L^2}{D \cdot t}\right)^2}$
	Defeitos Longos	
L	$L > \sqrt{20 \cdot D \cdot t}$	$L > \sqrt{50 \cdot D \cdot t}$
α	$\alpha = 1$	$\alpha = 0,85$
M	$M \to \infty$	$M = 3,3 + 0,032 \cdot \left(\frac{L^2}{D \cdot t}\right)$

A pressão de ruptura para este método é determinada pela seguinte expressão:

$$P_{rup} = \left(\sigma_{esc} + 69MPa\right) \cdot \left(\frac{2 \cdot t}{D}\right) \cdot \left[\frac{1 - 0,85\frac{d}{t}}{1 - 0,85\frac{d}{t}M^{-1}}\right] \tag{2.10}$$

b) RSTRENG "Effective Area"

Este método se baseia em definir diversos defeitos de comprimentos variados (L1, L2, ..., Ln), contidos dentro do comprimento total do defeito (L), conforme ilustrado na Figura 2.17, e calcular a pressão de ruptura para cada um deles. Cada um dos comprimentos de defeito (L1, L2,..., Ln) é denominado de $L_{efetivo}$ e sua respectiva área corroída de $A_{efetiva}$.

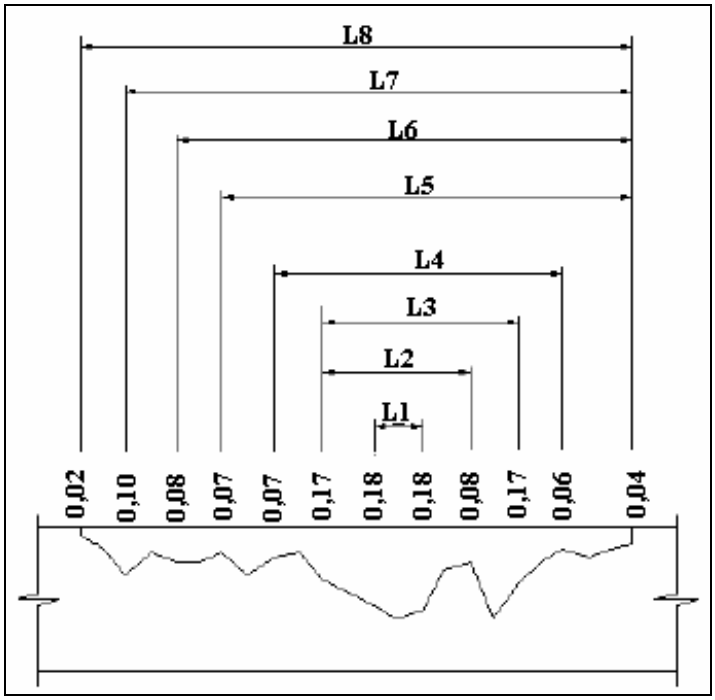


Figura 2.17 – Ilustração dos comprimentos para o cálculo pelo método RSTRENG "Effective Area".[18]

A área original da região corroída (A_0) para cada $L_{efetivo}$ é dada por: $A_0 = \left(L_{efetivo}\right) \cdot t$, onde t é a espessura de parede do duto.

A Tabela 2.3 mostra os principais parâmetros para o cálculo da pressão de ruptura para o método "Effective Area" através do software RSTRENG.

Tabela 2.3 – Parâmetros para o cálculo da pressão de ruptura via método "Effective Area".

	RSTRENG "Effective Area"
σ_{flow}	$\sigma_{flow} = \sigma_{esc} + 69MPa$
α	$\alpha = 0,85$
	Defeitos Curtos
$L_{efetivo}$	$L_{efetivo} \leq \sqrt{50 \cdot D \cdot t}$
M	$M = \sqrt{1 + 0,6275 \cdot \left(\frac{L_{efetivo}^{\ 2}}{D \cdot t}\right) - 0,003375 \cdot \left(\frac{L_{efetivo}^{\ 2}}{D \cdot t}\right)^2}$
	Defeitos Longos
$L_{efetivo}$	$L_{efetivo} > \sqrt{50 \cdot D \cdot t}$
M	$M = 3,3 + 0,032 \cdot \left(\frac{L_{efetivo}^{\ 2}}{D \cdot t}\right)$

[18] Fonte: Adaptado de Souza (2003).

A pressão de ruptura para cada defeito de comprimento $L_{efetivo}$ é determinada então pela seguinte expressão:

$$P_{rup} = \left(\sigma_{esc} + 69MPa\right) \cdot \left(\frac{2 \cdot t}{D}\right) \cdot \left[\frac{1 - \frac{A_{efetiva}}{A_0}}{1 - \frac{A_{efetiva}}{A_0} M^{-1}}\right] \tag{2.11}$$

onde:

$A_{efetiva} \Rightarrow$ Área efetiva corroída do defeito.
$L_{efetivo} \Rightarrow$ Comprimento efetivo do defeito.
$A_0 \Rightarrow$ Área original do defeito.

A pressão de ruptura do defeito de comprimento total L é a menor das pressões calculadas.

2.6.4 BS-7910

Esta norma fornece métodos para avaliação de defeitos de corrosão em dutos, tanto para o caso de defeitos simples quanto para o caso de múltiplos defeitos interagentes. As equações apresentadas no Anexo G da norma BS 7910, consideram apenas carregamento de pressão interna e podem ser aplicadas para os seguintes tipos de defeitos de corrosão:

- Corrosão interna ou externa no metal base;
- Corrosão em soldas longitudinais e circunferenciais (ou em regiões adjacentes);
- Colônias de defeitos de corrosão interagentes;

Para o caso de avaliação de áreas corroídas em solda, deve-se ter certeza de que: os defeitos provenientes do processo de soldagem do duto não estejam interagindo com os defeitos de corrosão, o material de solda deve ter resistência superior ao do metal base e que fratura frágil é improvável de ocorrer.

Os métodos inclusos na norma BS 7910 não podem ser aplicados para algumas condições, dentre as quais, podemos citar:

- Dutos de materiais diferentes de aço carbono;
- Defeitos não suaves, do tipo "trinca" ou defeitos devido a danos mecânicos;
- Defeitos com profundidade maior que 85% da espessura de parede do duto;
- Corrosão em regiões de concentração de tensões;

A seguir, descrevemos a terminologia adotada pela norma britânica BS 7910 assim como suas regras que definem se um defeito se comporta isoladamente ou se há interação com outros defeitos.

Defeito Isolado (ou simples) – é aquele o qual não interage com outros defeitos vizinhos (adjacentes). A pressão de falha de um defeito isolado é independente de outros defeitos existentes no duto.

Defeitos Interagentes – são aqueles os quais interagem com outros defeitos vizinhos nas direções longitudinais e circunferenciais.

Defeitos Complexos – são defeitos que resultam da combinação de colônias de defeitos interagentes ou pode ser um defeito simples (isolado) no qual o seu perfil da profundidade é conhecido.

Os defeitos de corrosão adjacentes podem interagir entre si ocasionando uma pressão de falha menor que aquela causada por um defeito isolado.

Um defeito é considerado como isolado se:

1) Sua profundidade for menor que 20% da espessura de parede do duto e;
2) O espaçamento circunferencial entre defeitos adjacentes, ϕ, exceder o ângulo:

$$\phi > 360 \cdot \frac{3}{\pi}\sqrt{\frac{t}{D}} \quad [em \quad graus] \quad \text{e;} \tag{2.12}$$

3) O espaçamento axial (longitudinal) entre defeitos adjacentes, s, exceder o valor dado por:

$$s > 2\sqrt{D \cdot t} \tag{2.13}$$

Dois defeitos adjacentes, com espaçamento axial (s) menor que o valor $2 \cdot \sqrt{D \cdot t}$, irão interagir entre si se uma das condições abaixo for satisfeita:

$$1)\ \left(\frac{1-\frac{d_1}{t}}{1-\frac{d_1}{t \cdot M_1}} \right) > \left\{ \frac{1-\left(\frac{1}{t}\right)\left(\frac{d_1 \cdot L_1 + d_2 \cdot L_2}{L_1 + L_2 + s}\right)}{1-\left(\frac{1}{t \cdot M_{12}}\right)\left(\frac{d_1 \cdot L_1 + d_2 \cdot L_2}{L_1 + L_2 + s}\right)} \right\} \tag{2.14}$$

$$2)\ \left(\frac{1-\frac{d_2}{t}}{1-\frac{d_2}{t \cdot M_2}} \right) > \left\{ \frac{1-\left(\frac{1}{t}\right)\left(\frac{d_1 \cdot L_1 + d_2 \cdot L_2}{L_1 + L_2 + s}\right)}{1-\left(\frac{1}{t \cdot M_{12}}\right)\left(\frac{d_1 \cdot L_1 + d_2 \cdot L_2}{L_1 + L_2 + s}\right)} \right\} \tag{2.15}$$

onde:

$$M_1 = \sqrt{1+0{,}31 \cdot \left(\frac{L_1}{\sqrt{D \cdot t}}\right)^2} \quad , \quad M_2 = \sqrt{1+0{,}31 \cdot \left(\frac{L_2}{\sqrt{D \cdot t}}\right)^2} \quad e \quad M_{12} = \sqrt{1+0{,}31 \cdot \left(\frac{L_1 + L_2 + s}{\sqrt{D \cdot t}}\right)^2}$$

Nas equações acima, d_1 e d_2 são as profundidades, L_1 e L_2 são os comprimentos longitudinais, M_1 e M_2 são os fatores de correção (dilatação) dos defeitos 1 e 2, e M_{12} é o fator de dilatação para a combinação dos defeitos 1 e 2, conforme ilustrado na Figura 2.18 que apresenta as principais dimensões para o caso de defeitos interagentes.

a) Cálculo da Pressão de Ruptura para Defeito Isolado

Caso não haja interação entre defeitos, calcula-se a pressão de ruptura (falha) considerando o defeito como sendo isolado por meio da seguinte equação:

$$P_{rup} = \left(\sigma_u\right) \cdot \left(\frac{2 \cdot t}{D - t}\right) \cdot \left[\frac{1-\frac{d}{t}}{1-\frac{d}{t}M^{-1}} \right] \tag{2.16}$$

onde:

$M = \sqrt{1+0{,}31\cdot\left(\frac{L}{\sqrt{D\cdot t}}\right)^2}$ e σ_u é o limite de resistência à tração do material (tensão de engenharia).

Note que neste caso, a área corroída é considerada como sendo retangular, e, portanto $\alpha = 1$. Para se definir a pressão máxima de operação, deve-se multiplicar a pressão de ruptura (P_{rup}) pelo fator de segurança (f_c) que é dado por:

$$f_c = f_{c1} \cdot f_{c2}$$

onde:

$f_{c1} \Rightarrow$ é o fator de modelagem[19] (baseado na precisão das equações em comparação com os dados experimentais)

$f_{c2} \Rightarrow$ é o fator de projeto (margem de segurança entre a pressão de operação e a pressão de falha).

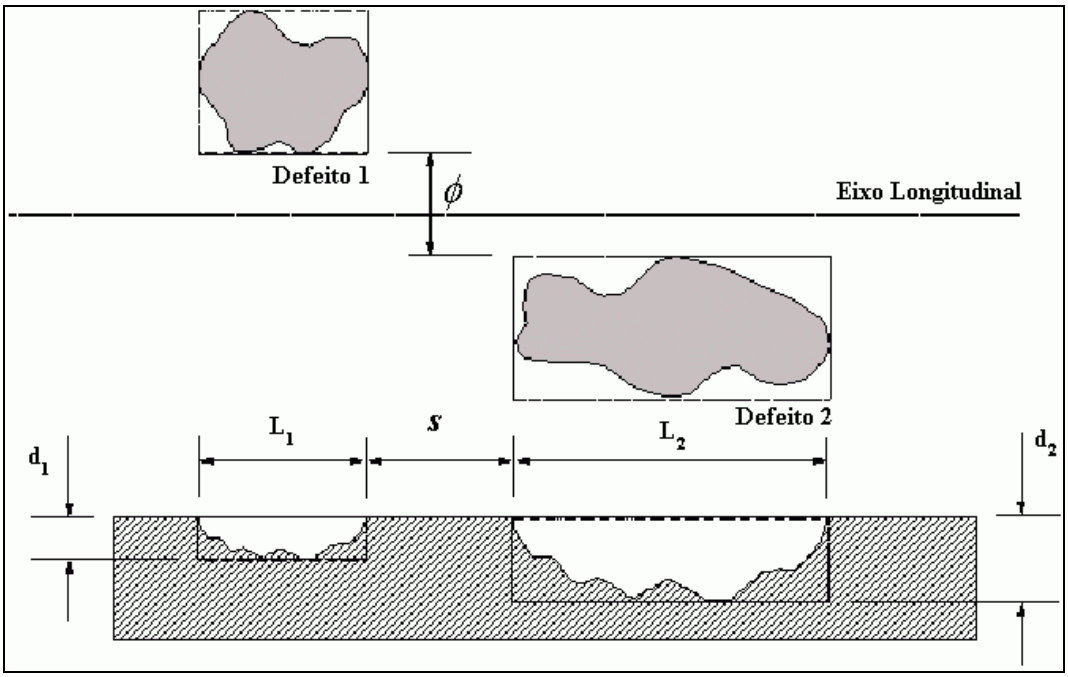


Figura 2.18 – Principais dimensões para o caso de defeitos interagentes.[20]

b) Cálculo da Pressão de Ruptura para Defeitos Interagentes

A seguir, é descrito o procedimento da norma BS 7910 para avaliação da resistência residual de dutos na presença de defeitos múltiplos interagentes. Este procedimento considera apenas carregamento de pressão interna.

Passo 1) Para regiões onde há perda generalizada de material (menor que 10% da espessura de parede do duto), a espessura de parede local e a profundidade do defeito devem ser utilizadas, conforme ilustrado na Figura 2.19.

[19] f_{c1} = 0,9 caso esteja utilizando o limite máximo de resistência à tração (tensão última) medido experimentalmente.
f_{c1} = 1,0 caso esteja utilizando o limite mínimo de resistência à tração medido experimentalmente.
[20] Fonte: Adaptado de http://exchange.dnv.com/OGPI/OffshorePubs/Members/rp-f101.pdf.

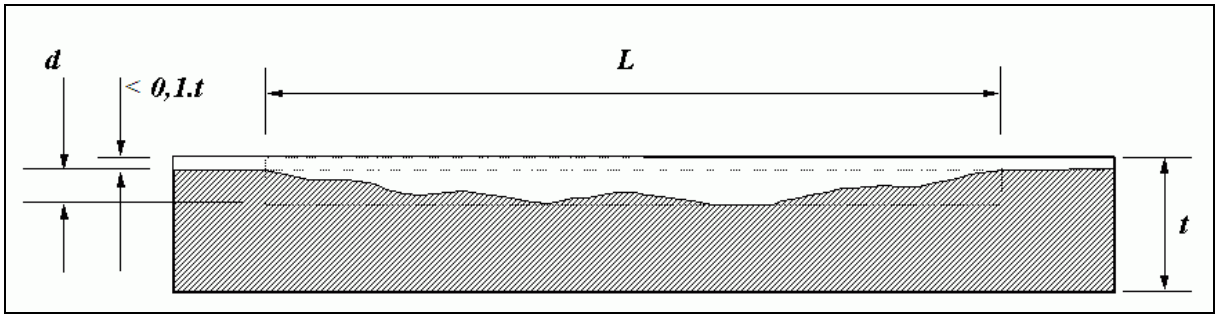


Figura 2.19 – Ajuste na espessura e profundidade do defeito com corrosão generalizada.[21]

Passo 2) A região corroída do duto deve ser dividida em seções com comprimento mínimo de $5 \cdot \sqrt{D \cdot t}$, com uma superposição mínima de $2{,}5 \cdot \sqrt{D \cdot t}$. Em seguida, os passos 3 a 12 devem ser repetidos para cada comprimento seccionado de forma a avaliar todas as possíveis interações.

Passo 3) Construir um série de linhas longitudinais de projeção espaçadas circunferencialmente do ângulo dado por:

$$\phi = 360 \cdot \frac{3}{\pi} \sqrt{\frac{t}{D}} \quad [em \quad graus]$$

Passo 4) Considerar uma linha de projeção de cada vez. Se os defeitos estiverem dentro de $\pm\phi$, eles devem ser projetados na linha de projeção atual, conforme ilustrado na Figura 2.20.

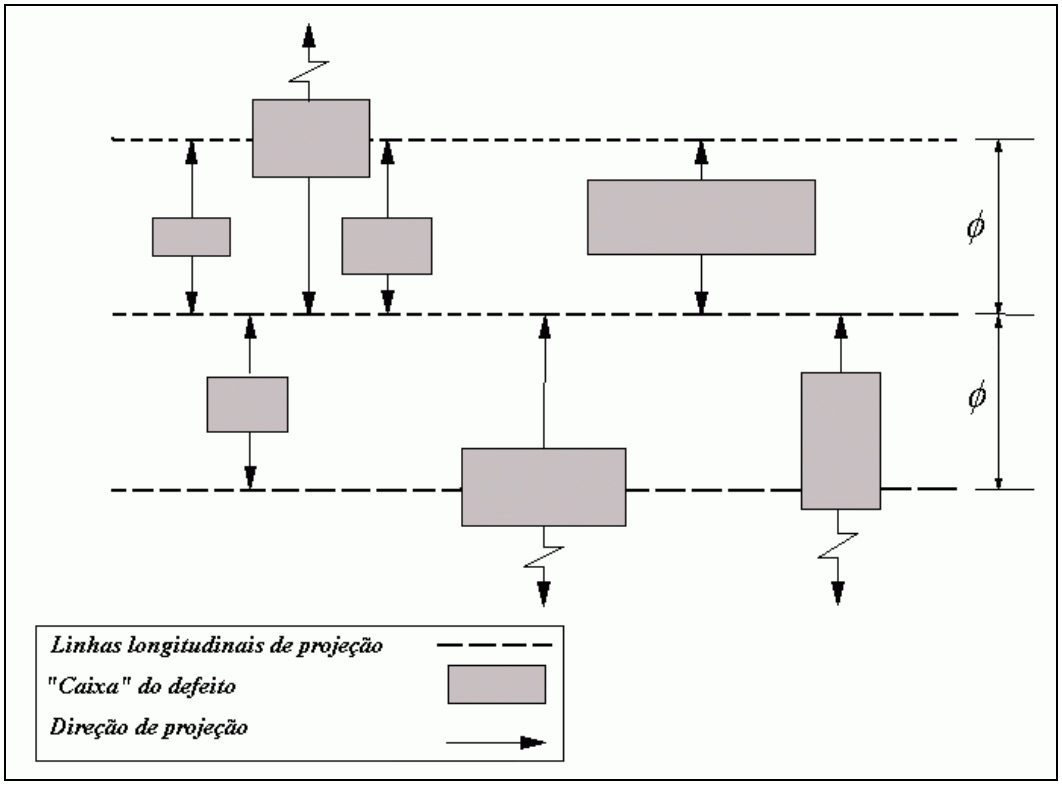


Figura 2.20 – Projeção de defeitos interagentes na direção circunferencial.

[21] Fonte: Adaptado de http://exchange.dnv.com/OGPI/OffshorePubs/Members/rp-f101.pdf.

Passo 5) Se houver superposição de defeitos, eles devem ser combinados para formar um defeito "composto". Isto é feito combinando-se o comprimento dos defeitos e considerando a maior profundidade entre os mesmos, conforme ilustrado na Figura 2.21.

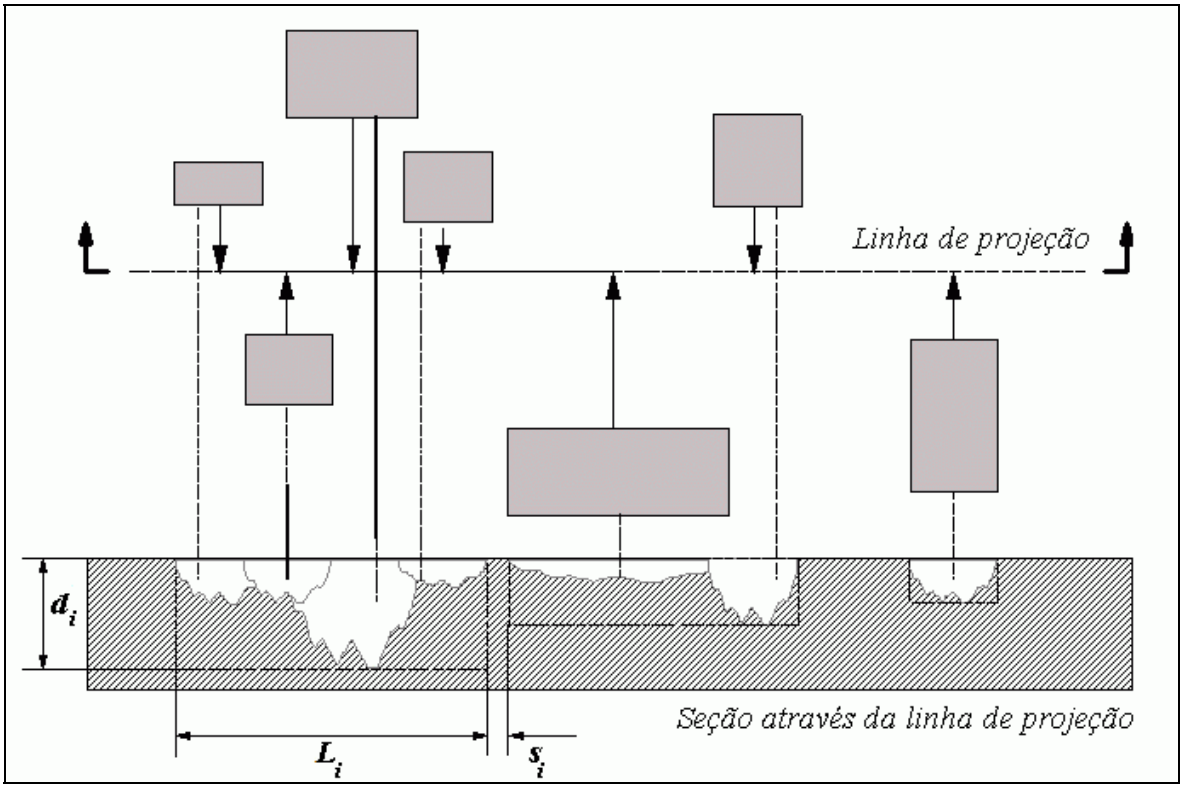


Figura 2.21 – Projeção de defeitos superpostos na linha de projeção.[22]

Passo 6) Calcular as pressões de ruptura (P_1, P_2,..., P_N) para cada defeito (ou defeito composto), tratando cada um, como um defeito isolado (ver Figura 2.19), usando as seguintes equações:

$$P_{rup_{(i)}} = (\sigma_u) \cdot \left(\frac{2 \cdot t}{D - t} \right) \cdot \left[\frac{1 - \frac{d_i}{t}}{1 - \frac{d_i}{t} M_i^{-1}} \right] \tag{2.17}$$

onde:

$$M_i = \sqrt{1 + 0{,}31 \cdot \left(\frac{L_i}{\sqrt{D \cdot t}} \right)^2}$$

Passo 7) Calcular o comprimento total de todas as combinações de defeitos interagentes (ver Figura 2.22). Para os defeitos "n" até "m", o comprimento total (L_{nm}) é dado por:

$$L_{nm} = L_m + \sum_{i=n}^{i=m-1} (L_i + s_i) \tag{2.18}$$

[22] Fonte: Adaptado de http://exchange.dnv.com/OGPI/OffshorePubs/Members/rp-f101.pdf.

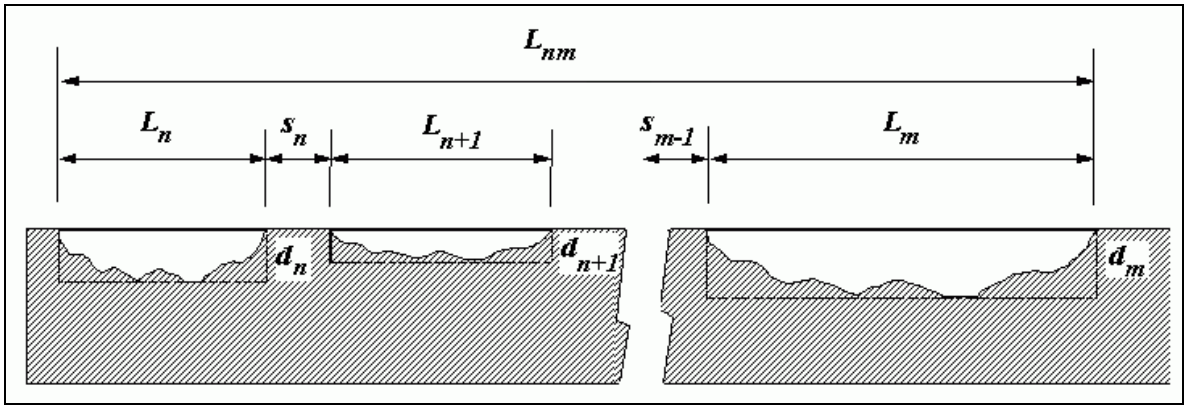


Figura 2.22 – Combinação de defeitos interagentes.[23]

Passo 8) Calcular a profundidade efetiva do defeito combinado (formado pelos defeitos interagentes "n" até "m") dada por (ver Figura 2.22):

$$d_{nm} = \frac{\sum_{i=n}^{i=m} (d_i \cdot L_i)}{L_{nm}} \tag{2.19}$$

Possíveis grupos de defeitos interagentes também são considerados nas avaliações para encontrar o grupo que resulta na menor estimativa da pressão de ruptura conforme ilustrado na Figura 2.23.

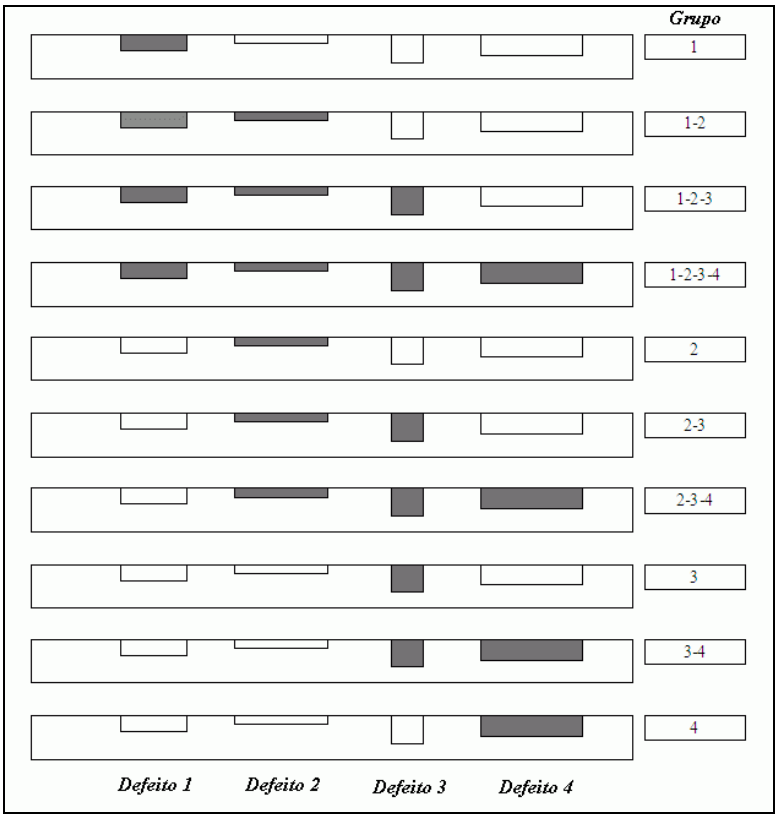


Figura 2.23 – Exemplo de grupos de defeitos interagentes.

[23] Fonte: Adaptado de http://exchange.dnv.com/OGPI/OffshorePubs/Members/rp-f101.pdf.

Passo 9) Calcular a pressão de ruptura para cada combinação de defeitos (P_{nm}) usando L_{nm} e d_{nm} na equação para defeito isolado, dada por:

$$P_{nm} = (\sigma_u) \cdot \left(\frac{2 \cdot t}{D - t} \right) \cdot \left[\frac{1 - \frac{d_{nm}}{t}}{1 - \frac{d_{nm}}{t} M_{nm}^{-1}} \right] \tag{2.20}$$

onde:

$$M_{nm} = \sqrt{1 + 0,31 \cdot \left(\frac{L_{nm}}{\sqrt{D \cdot t}} \right)^2}$$

Passo 10) A pressão de ruptura, para a linha de projeção atual, é dada como sendo o menor valor entre as pressões de ruptura dos defeitos individuais (P_1 até P_N) e as pressões de ruptura dos defeitos combinados (P_{nm}). Ou seja[24]:

$$P_{rup} = \min\left(P_1, P_2, ..., P_N, P_{nm}\right) \tag{2.21}$$

Passo 11) A pressão de ruptura, para a seção corroída do duto, é dada como sendo a menor das pressões calculadas para cada linha de projeção ao longo da circunferência.

Passo 12) Repetir os passos 4 a 11 para a próxima seção corroída do duto.

[24] Nota: Para determinar a máxima pressão admissível de operação do duto corroído, deve-se multiplicar a pressão de ruptura (P_{rup}) pelo fator de segurança (f_c).

2.7 Simulações Numéricas e Ensaios Experimentais

Os trabalhos mais recentes encontrados na literatura mostram que os métodos semi-empíricos, descritos na seção anterior, ainda são bastante utilizados como instrumentos de avaliação da resistência residual de dutos corroídos. Atualmente há também uma forte tendência de cada vez mais realizar ensaios experimentais (em escala real) e simulações numéricas, por meio de softwares comerciais que implementam o método dos elementos finitos, para estimar a pressão de ruptura de dutos corroídos com um bom grau de precisão (Alves, 2002; Hopkins, 2002a).

Vários ensaios experimentais em escala real têm sido feitos para validar novos métodos semi-empíricos que estão sendo desenvolvidos ou para comprovar o elevado grau de conservadorismo de métodos já existentes para determinadas configurações de defeito de corrosão.

O desenvolvimento e validação de novos métodos semi-empíricos baseiam-se não somente nos resultados experimentais feitos em laboratórios, mas também em análises 3-D via método dos elementos finitos gerando assim grandes bancos de dados com diferentes dimensões de defeitos de corrosão em dutos (Benjamin et al, 2000; Noronha Jr. et al, 2002).

2.7.1 Defeitos Artificiais de Corrosão

Em vários trabalhos experimentais desenvolvidos nesta área, os ensaios são realizados em dutos com defeitos artificiais de corrosão de forma a facilitar a análise por elementos finitos. Ou seja, os defeitos são usinados (por eletro-erosão, por exemplo) assumindo determinadas formas geométricas na superfície do duto (retangular, elíptica, etc.) com o objetivo de simular a corrosão para a posterior investigação experimental e comparação com modelos numéricos.

Diniz (2002) mostra os principais resultados do Projeto Produt[25] 317900 (Programa Tecnológico de Dutos da Petrobrás) e que foram publicados por Benjamin et al (2000). Neste projeto, foram realizadas uma série de testes experimentais e simulações numéricas 3-D em nove espécimes tubulares de aço API 5L X60, com comprimento nominal de 2,0m, diâmetro de 323,85mm e espessura nominal de 9,53mm. Os defeitos contidos nos espécimes foram usinados por eletro-erosão assumindo a forma retangular na superfície externa do duto com uma profundidade de aproximadamente 6,67mm (70% da espessura) e largura de 95,3mm (10 vezes a espessura). O comprimento do defeito foi o único parâmetro que variou (entre 250mm e 525mm) e as arestas do defeito retangular foram suavizadas por meio de raio de adoçamento.

Nas análises numéricas tridimensionais realizadas, utilizando-se elementos hexaédricos (8 nós) e 3 elementos ao longo da espessura remanescente, Diniz obteve valores numéricos bem próximos aos valores experimentais. Verificou também que esses modelos numéricos são bastante sensíveis em relação às dimensões geométricas do duto e do defeito, indicando que pequenas variações no valor da espessura resultam em grandes diferenças nas previsões das pressões de ruptura, podendo assim, obter valores não-conservadores para a pressão de ruptura. Num trabalho posterior, Noronha Jr. et al (2002) compara as pressões de falha experimentais (de quatro dos nove espécimes descritos acima) com as pressões de falha estimadas por meio de modelagens computacionais tridimensionais usando elementos finitos sólidos e elementos finitos de casca. Os resultados mostraram que ambos os modelos foram capazes de representar bem os dutos com defeitos analisados. Os modelos de elementos finitos de casca obtiveram resultados mais conservadores que os modelos de elementos finitos sólidos, este último fornecendo resultados mais precisos.

---

[25] Benjamin, A.C., 2000. "*Avaliação Estrutural de Dutos Corroídos com Defeitos Longos*"- Relatório Final Projeto Produt 317900.

O desempenho desses dois tipos de elementos finitos (de casca e sólido) também foi investigado em trabalhos posteriores sobre o comportamento de dutos com defeitos longos de profundidade variável, por meio de experimentos (Benjamin et al, 2002) e análises via MEF (Benjamin & Andrade, 2003b).

Os resultados numéricos obtidos por Benjamin & Andrade (2003b), foram similares aos resultados obtidos por Noronha Jr. et al (2002) mostrando que tanto os elementos finitos tridimensionais de casca quanto os elementos finitos sólidos foram capazes de representar satisfatoriamente os defeitos longos de profundidade variável.

Poucos trabalhos experimentais e numéricos, abordando o comportamento de dutos na presença de múltiplos defeitos de corrosão, têm sido publicados na literatura. Conforme mostrado na Tabela 2.1, a maior parte dos métodos semi-empíricos ou não tratam este tipo de configuração de defeitos ou assumem procedimentos bastante conservadores no cálculo da pressão de falha. Este assunto foi estudado de forma mais abrangente por Chouchaoui & Pick que analisaram a resistência residual de dutos na presença de múltiplos defeitos alinhados circunferencialmente (Chouchaoui & Pick, 1994) e longitudinalmente (Chouchaoui & Pick, 1996) por meio de uma série de testes experimentais em escala real e análises tridimensionais de elementos finitos. Os ensaios experimentais foram realizados em amostras de dutos contendo defeitos artificiais de corrosão usinados por meio de eletro-erosão e tinham o formato elíptico com profundidade em torno de 60% da espessura de parede do duto. Cada duto continha dois grupos de defeitos alinhados (circunferencialmente ou longitudinalmente) e cada grupo possuía um determinado espaçamento entre os defeitos. Chouchaoui & Pick investigaram diferentes configurações de espaçamento entre defeitos com o objetivo de avaliar o efeito da interação dos mesmos no valor da pressão de ruptura medida experimentalmente.

Para o caso de múltiplos defeitos alinhados circunferencialmente, e com o duto submetido apenas à pressão interna, os resultados numéricos foram bastante satisfatórios em comparação com os resultados experimentais. Verificou-se também que quase não houve interação entre os defeitos, mesmo em situações onde os defeitos adjacentes estavam juntos um do outro (espaçamento nulo entre defeitos). No entanto, para esta mesma configuração de defeitos, porém com o duto submetido a carregamento de pressão interna mais carregamento axial de extremidade, os resultados numéricos não foram tão precisos quanto os obtidos com carregamento apenas de pressão interna. Isto levou Chouchaoui & Pick a concluir que na presença de carregamento combinado pode ser necessário considerar a anisotropia do material nas análises numéricas.

Para o caso de múltiplos defeitos alinhados longitudinalmente, os resultados numéricos também foram bastante satisfatórios em comparação com os resultados experimentais novamente com o duto submetido apenas à pressão interna. Para esta configuração de defeitos, verificou-se a interação entre defeitos adjacentes próximos uns dos outros. Os resultados mostraram que, em um grupo de defeitos alinhados longitudinalmente, o defeito central terá a tendência de falhar primeiro, diferentemente do que foi verificado para o caso de defeitos alinhados circunferencialmente onde o defeito mais externo do grupo é que esteve sujeito aos maiores níveis de tensões.

Recentes estudos do comportamento da pressão de falha de dutos na presença de múltiplos defeitos interagentes foram realizados por Benjamin et al (2005), Andrade et al (2006) e Benjamin et al (2006). Nestes estudos (experimentais e numéricos) os autores investigaram dois dos três tipos básicos de interação entre defeitos (Tipo 1 e Tipo 2) definidos por Kiefner & Vieth (1990) da seguinte forma:

Interação Tipo 1: Encontrada em grupos de defeitos nos quais os defeitos são separados circunferencialmente mas os perfis individuais de cada um se sobrepõem quando projetados no plano longitudinal através da espessura de parede do duto (Figura 2.24).

Um caso particular de interação do Tipo 1 ocorre quando os defeitos estão alinhados circunferencialmente.

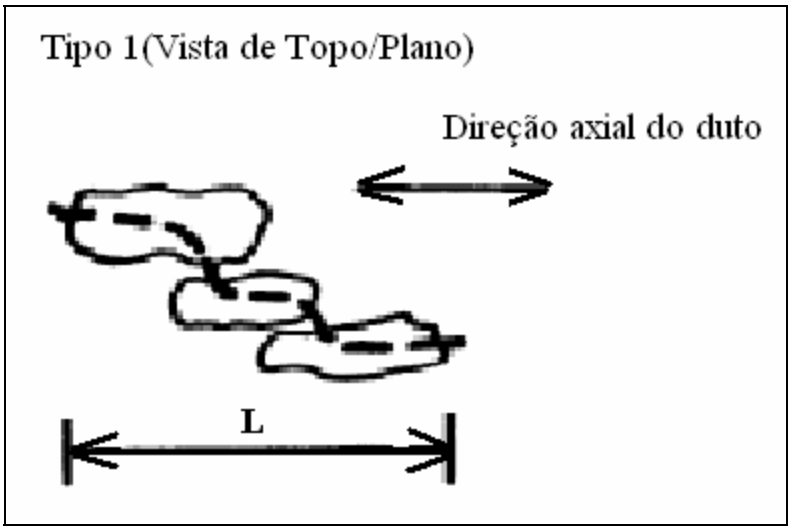


Figura 2.24 – Interação Tipo 1 e Tipo A.[26]

Interação Tipo 2: Encontrada em grupos de defeitos nos quais os defeitos são alinhados longitudinalmente e os perfis individuais de cada um são separados por uma distância "L_3" equivalente à dimensão da espessura "t" de parede íntegra do duto (Figura 2.25).

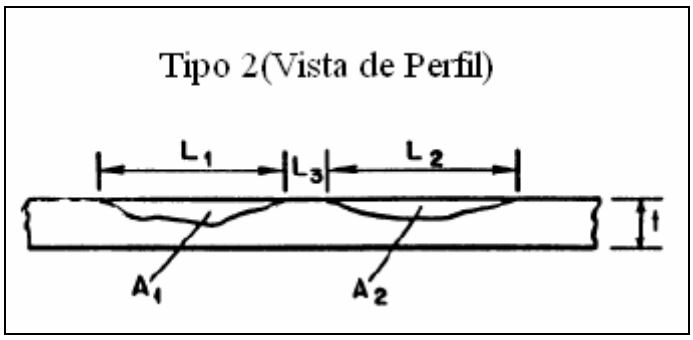


Figura 2.25 – Interação Tipo 2 e Tipo B.

Interação Tipo 3: Encontrada em grupos constituídos por um defeito de corrosão raso com defeitos profundos dentro dele (defeitos complexos) (Figura 2.26).

Benjamin et al (2006) redefiniram os tipos básicos de interação 1 e 2 dando ênfase na forma geométrica assumida pelo grupo de defeitos na superfície do duto ao invés de dar ênfase na projeção dos perfis individuais dos defeitos. As novas definições para os tipos de interação 1 e 2, denominadas de Tipo A e Tipo B, respectivamente, são descritas a seguir.

Interação Tipo A: Encontrada em um grupo de defeitos composto por poucos defeitos dispostos na superfície do duto (interna ou externa) ao longo de um caminho curvilíneo (Figura 2.24).

[26] Fonte: Adaptado de Kiefner & Vieth (1990) apud Benjamin et al (2006).

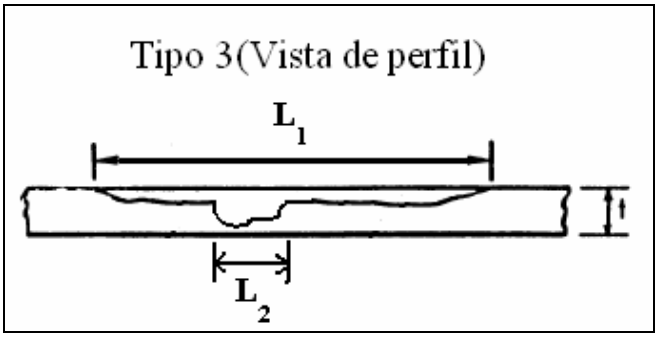


Figura 2.26 – Interação Tipo 3.[27]

> "*Na realidade, uma interação do Tipo 2 pode ocorrer mesmo nos casos em que os defeitos não são alinhados longitudinalmente. Dessa forma, uma definição mais geral seria: interação do Tipo 2 é encontrada em grupos de defeitos nos quais as projeções dos perfis individuais de cada defeito são separadas por uma distância equivalente à espessura de parede íntegra do duto*."
> (*Benjamin et al, 2006, p.3*)

Interação Tipo B: Encontrada em um grupo de defeitos composto por poucos defeitos dispostos na superfície do duto (interna ou externa) ao longo de uma linha reta que pertence ao plano longitudinal através da espessura de parede do duto (Figura 2.25).

Numa primeira etapa, Benjamin et al (2005) realizaram ensaios experimentais em seis espécimes tubulares contendo defeitos interagentes usinados por meio de eletro-erosão no formato retangular na superfície externa do duto. Os defeitos foram usinados assumindo configurações de interações de defeitos do Tipo 1, Tipo 2 e do Tipo 1 e 2 (combinação dos Tipos 1 e 2). Os espécimes tubulares de 1,7m de comprimento cada, 457,2mm de diâmetro externo nominal e espessura de parede de 7,93 foram submetidos à pressão interna até a ruptura (teste hidrostático), instante no qual foi medida a pressão de ruptura.

Numa segunda etapa, Andrade et al (2006) realizaram análises por elementos finitos dos seis espécimes tubulares ensaiados por Benjamin et al (2005) com o objetivo de comparar a pressão de ruptura medida experimentalmente com os valores de pressão de ruptura estimados numericamente.

A Tabela 2.4 apresenta um resumo dos principais resultados experimentais e numéricos obtidos por Benjamin et al (2005) e Andrade et al (2006) para os seis espécimes analisados.

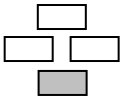
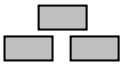
Para cada espécime, a Tabela 2.4 apresenta a configuração de defeitos, o tipo de interação (segundo Kiefner & Vieth, 1990), o tipo de falha ocorrida nos ensaios experimentais e as respectivas pressões de ruptura medidas experimentalmente $(P_f)_{exp}$ e as obtidas numericamente via elementos finitos $(P_f)_{MEF}$.

Segundo Benjamin et al (2006), os tipos de falhas são definidos da seguinte forma:

Falha Total: Um grupo de defeitos composto por poucos defeitos dispostos na configuração de um dos tipos básicos de interação (Tipo 1 ou Tipo 2) geralmente sofre uma falha que engloba todos os defeitos do grupo e a espessura íntegra de parede do duto entre eles.

[27] Fonte: Adaptado de Kiefner & Vieth (1990).

Tabela 2.4 – Tipos de falha dos espécimes e resultados experimentais *versus* numéricos (MEF) da pressão de ruptura.[28]

Espécime tubular	Configuração do grupo de defeitos	Tipo de Interação	Tipo de Falha	$(P_f)_{exp}$ [MPa]	$(P_f)_{MEF}$ [MPa]
IDTS 2		-	individual	22,679	22,710
IDTS 3		2	individual	20,314	19,535
IDTS 4		1	individual	21,138	22,298
IDTS 5		1	individual	20,873	20,083
IDTS 6		1 e 2	individual	18,656	19,190
IDTS 7		1 e 2	total	18,773	19,090

Nota: defeito que falhou

Falha Parcial: Um grupo de defeitos composto por vários defeitos dispostos numa configuração que combina os tipos básicos de interação (Tipo 1 e Tipo 2) geralmente sofre uma falha que engloba parte dos defeitos do grupo e a espessura íntegra de parede do duto entre eles.

Falha Individual: Um grupo de defeitos composto por defeitos nos quais há uma grande diferença entre a espessura remanescente e a espessura íntegra do duto (defeitos profundos), os defeitos podem sofrer uma falha que engloba apenas um defeito.

Conforme mostrado na Tabela 2.4, a falha individual ocorreu em cinco dos seis espécimes ensaiados que tinham defeitos cuja relação (profundidade do defeito/espessura do duto) era de aproximadamente 0,66.

Os resultados numéricos foram bastante satisfatórios e os erros na estimativa da pressão de ruptura ficaram dentro da faixa de -3,78% e +5,49% utilizando elementos finitos sólidos com quatro elementos ao longo da espessura remanescente do duto.

Os modelos de elementos finitos apresentaram uma configuração de falha bastante semelhante às obtidas experimentalmente e foram capazes de prever, na maior parte dos espécimes, a região onde a fratura ocorreu. A modelagem e análise por elementos finitos foram feitas por meio do software ANSYS e levou em consideração grandes deformações e deslocamentos, endurecimento e não-linearidade do material (API 5L-X80).

Os espécimes IDTS2, IDTS3 e IDTS4 serão aqui também simulados numericamente para validar as ferramentas computacionais propostas neste trabalho. Os resultados serão apresentados na seção 5.2 desta dissertação.

[28] Fonte: Adaptado de Andrade et al (2006).

2.7.2 Defeitos Reais de Corrosão

A maior parte dos trabalhos publicados na área de defeitos reais de corrosão (também chamados de defeitos naturais ou defeitos complexos) são experimentais. Infelizmente, poucos trabalhos têm sido publicados com o objetivo de estimar a pressão de ruptura de dutos com defeitos de corrosão reais por meio do método dos elementos finitos.

Embora o uso de simulações numéricas via MEF requeira informações detalhadas sobre a geometria da corrosão, o uso desta técnica pode ser bastante viável para casos em que se necessita de resultados precisos para a estimativa da pressão de ruptura de dutos com defeitos de corrosão reais (perfil complexo). Souza (2003) apresenta parte dos resultados experimentais realizados em dutos com defeitos de corrosão reais que foram removidos do oleoduto Orbel I, pertencente à Petrobrás, durante uma obra de reabilitação.

Souza utilizou alguns dos ensaios experimentais, realizados em outro projeto denominado Produt[29] 600536, para comparar os resultados das pressões de falha obtidos experimentalmente com os valores das pressões de falha estimados pelos métodos semi-empíricos: ASME B31G, RSTRENG 0,85dL, RSTRENG "Effective Area", DNV RP-F101 (defeitos isolados) e DNV RP-F101 (defeitos complexos). Foram utilizados cinco espécimes tubulares de aço API 5L X46, com 3,0m de comprimento, 457,2mm de diâmetro nominal e 6,35mm de espessura nominal de parede. Estes espécimes continham defeitos reais de corrosão interna, do tipo longo, localizados na geratriz inferior do duto e foram mapeados por meio de medição manual e mecanizada (ultra-som) para a obtenção de vários perfis de corrosão.

Os resultados apresentados por Souza confirmaram o conservadorismo embutido no método ASME B31G e comprovaram que os métodos "Effective Area" e DNV RP-F101 (defeito complexo), que utilizam o perfil de corrosão, apresentaram melhores resultados que os outros métodos para os casos analisados.

O uso de análises numéricas via o método dos elementos finitos para estimar a pressão de ruptura de dutos com defeitos de corrosão reais foi investigado por Cronin (2002) em um dos poucos trabalhos publicados nesta área. Talvez, os principais motivos pela escassez de trabalhos nesta área sejam devidos à necessidade de mapeamento preciso do perfil real da corrosão e à dificuldade na geração de malha na região corroída uma vez que esta região não é facilmente definida através de ferramentas existentes nos softwares comerciais. Cronin (2002) utilizou um programa personalizado que foi especialmente desenvolvido para ler os dados geométricos do perfil de corrosão e gerar a malha de elementos finitos sólidos (de 20 nós) na região corroída. A malha gerada neste programa foi então utilizada como entrada no software comercial MSC.Patran, onde o restante do modelo foi gerado em torno do defeito. Após estudos de refinamento de malha, Cronin (2002) utilizou dois elementos finitos sólidos ao longo da espessura remanescente do duto. Cronin (2002) comparou os resultados das análises numéricas (realizadas via software comercial Abaqus) com os resultados experimentais. Os resultados numéricos apresentaram um erro médio de -0,18% e um desvio padrão médio de 8,45% em relação aos obtidos experimentalmente. No entanto, quando os defeitos foram medidos por um aparelho mais preciso, os resultados apresentaram um erro médio de 0,1% e um desvio padrão médio de 4,1%, indicando que, quando comparado com os métodos semi-empíricos, o método dos elementos finitos fornece resultados mais precisos na estimativa da pressão de ruptura de defeitos de corrosão reais nas situações onde se dispõe de informações precisas do perfil de corrosão.

[29] Benjamin, A.C., 2003. "*Avaliação Estrutural com Defeitos de Profundidade Variável*"- Relatório Final Projeto Produt 600536.

2.8 Considerações Finais

Os métodos semi-empíricos apesar de seu comprovado conservadorismo, é ainda a ferramenta mais utilizada na tomada de decisões sobre a resistência residual de dutos corroídos. Além disso, é uma ferramenta que possibilita que o engenheiro em campo possa avaliar rapidamente a situação do duto corroído através de uma simples calculadora.

O uso de avaliações por níveis de complexidade para avaliar a resistência residual de dutos corroídos é uma metodologia bastante interessante como forma de tratar a grande quantidade de dados obtidas dos métodos automáticos de medição da corrosão. O objetivo desta metodologia é aplicar procedimentos simples de avaliação para identificar possíveis defeitos críticos. Estes defeitos críticos podem ser analisados utilizando técnicas de avaliação mais precisas e complexas. Quanto mais preciso for o método de avaliação, maior será o número de informações necessárias e mais complexa e detalhada se tornará a análise elevando assim o custo.

Outros métodos alternativos para avaliação da resistência residual de dutos corroídos também foram desenvolvidos recentemente. Como exemplo, podemos citar os métodos que utilizam análise limite e probabilidade. Procedimentos como o SINTAP ("Structural Integrity Assessment Procedures for European Industry") baseiam-se no fato de que a falha é evitada até que o duto não seja submetido a carregamentos além da sua capacidade máxima definida por critérios baseados na mecânica da fratura e análise limite. Na aplicação dos métodos probabilísticos, a probabilidade de falha pode ser calculada levando em consideração as incertezas do modelo, variação natural dos parâmetros, capacidade de dimensionamento dos defeitos e estimativa da evolução do defeito. Os métodos probabilísticos ainda são pouco utilizados na avaliação estrutural de defeitos de corrosão em dutos, mas, com o tempo, ele será mais usado em função de um maior rigor das autoridades com relação ao controle do risco envolvido no transporte por dutos (Bjornoy & Marley, 2001).

A análise computacional por meio do método dos elementos finitos tem se mostrado uma das ferramentas mais eficientes para a avaliação precisa da integridade estrutural de dutos corroídos, principalmente para casos de múltiplos defeitos ou defeitos de geometria complexa, fornecendo resultados bastante precisos. No entanto, a modelagem e análise pelo MEF requer mão de obra especializada e um treinamento específico por parte dos engenheiros de tubulação. Além disso, o processo de geração de bons modelos é bastante demorado, repetitivo e muito propenso a erros.

O próximo capítulo desta dissertação irá propor o uso de ferramentas computacionais desenvolvidas com o objetivo de produzir automaticamente modelos de elementos finitos de dutos com defeitos (simples ou múltiplos alinhados), prontos para serem analisados nos principais programas comerciais que implementam o MEF. A utilização destas ferramentas implica na redução do tempo de criação dos modelos e redução dos erros de modelagem o que se traduz em economia de recursos e no aumento da segurança operacional do duto.

3 AUTOMATIZAÇÃO DA MODELAGEM DE DEFEITOS DE CORROSÃO EM DUTOS

3.1 Introdução

A modelagem e análise computacional de dutos com defeitos causados por corrosão por meio do Método dos Elementos Finitos (MEF) tem se mostrado uma das ferramentas mais eficientes para a avaliação correta da integridade estrutural de dutos corroídos (Costa, 2004). A avaliação da segurança operacional de dutos com defeitos de corrosão é normalmente feita com o uso de normas, como a britânica BS 7910. O uso de normas, no entanto, sempre implica em uma grave simplificação na geometria dos defeitos reais, o que pode levar a resultados imprecisos. A simulação computacional pelo MEF permite uma representação muito mais fiel destes defeitos, e além disto, por considerar diretamente os fenômenos físicos envolvidos no processo de falha do duto, produz resultados mais precisos, como tem sido comprovado em experiências recentes(Chouchaoui & Pick,1996; Fu & Kirkwood, 1995; Diniz, 2002; Cronin, 2002; Benjamin & Andrade, 2003b, Costa, 2004; Andrade et al, 2006).

A modelagem e análise via MEF, no entanto, requer uma grande especialização e um treinamento específico que não são característicos de todos os engenheiros de tubulações. O processo para a criação de bons modelos computacionais para um defeito, que inclui a modelagem fiel da geometria deste defeito e a geração de uma malha apropriada, demanda uma interação manual constante do engenheiro, é demorado e muito repetitivo, e por estas razões muito propenso a erros. Normalmente, este procedimento é repetido desde o início para cada novo problema analisado, em um patente desperdício de recursos humanos qualificados.

Neste capítulo, é apresentado um conjunto de ferramentas computacionais, baseadas no programa comercial de pré e pós-processamento MSC.PATRAN (PATRAN, 2005), para geração automática de modelos de elementos finitos de defeitos causados por corrosão em dutos metálicos para posterior análise via programa de elementos finitos (“solver”). Estas ferramentas foram produzidas pelo grupo de pesquisa PADMEC por indução do CENPES/PETROBRÁS no projeto “Geração Automática de Defeitos de Corrosão em Dutos - MOSINEIP” da Rede Norte-Nordeste de Pesquisa Cooperativa em Modelagem Computacional (RPCMOD-Rede9) com o apoio financeiro do FINEP/CTPETRO e CENPES/PETROBRÁS.

Entendendo que a análise de defeitos via o MEF já é uma das principais ferramentas para a avaliação da integridade estrutural de dutos, os benefícios obtidos pela companhia operadora de dutos ao utilizar essas ferramentas automáticas são vários:

Redução no tempo de criação do modelo – Atualmente, a geração de um bom modelo de defeito pode levar muitos dias. Claramente, isto dificulta que simulações computacionais sejam usadas na tomada de decisão sobre a segurança de um duto específico, já que raramente o engenheiro pode dar-se ao luxo de exigir que um duto suspeito seja retirado de operação por vários dias, enquanto elabora modelos computacionais.

Com a aceleração deste processo, visualizamos um futuro no qual o engenheiro pode apelar para ferramentas computacionais e tomar decisões baseadas nos seus resultados.

Redução de erros de modelagem – A geração automática, pode requisitar muito menor intervenção manual, é muito menos propensa a erros do que a geração manual, sendo possível também implementar alguns mecanismos de verificação automática que diminuem a probabilidade de geração de modelos inadequados mesmo que o usuário assim o requisite. As conseqüências de um modelo computacional incorreto, que está sendo usado para avaliar a segurança de um duto, podem ser catastróficas.

Uso eficiente de mão de obra especializada – Os engenheiros de tubulações são, em geral, extremamente competentes em sua área de atuação.

Tê-los sentados na frente de um computador, simplesmente repetindo um procedimento mecânico que pode ser automatizado, é um desvio de função.

Os engenheiros devem concentrar-se em controlar a simulação computacional, avaliando a confiabilidade dos seus resultados e verificando sua validade, e amparados por estes resultados, mas baseados também em sua experiência e conhecimentos, propor soluções para os problemas encontrados. Além disto, um processo de geração automática de modelos permite que outros engenheiros, que não tenham treinamento específico no programa de modelagem, possam realizar análises computacionais.

Economia e segurança – Claramente, as vantagens acima se traduzem em economia de recursos e no aumento da segurança operacional. Com o aumento da velocidade de tomada de decisão sobre a condição de um duto pode-se diminuir o tempo de parada de um sistema comprometido. Além disto, uma avaliação mais precisa da gravidade de um defeito permite que, por exemplo, se opere o duto à pressão reduzida, com segurança, até que uma parada pré-programada seja atingida, sendo preservada pelo menos parte da produção.

As ferramentas computacionais apresentadas neste capítulo foram produzidas por meio da linguagem de programação PCL - "Patran Command Language" (PATRAN, 2005) e têm interfaces gráficas simplificadas e personalizadas, de forma que um engenheiro, com noções básicas de simulação computacional com elementos finitos, possa gerar rapidamente modelos que resultem em simulações precisas e confiáveis. Este conjunto de ferramentas para geração automática de modelos de dutos com defeitos constitui o programa computacional denominado de PIPEFLAW. Até o momento, o programa PIPEFLAW gera automaticamente modelos de dutos com defeitos simples ou múltiplos alinhados (longitudinalmente ou circunferencialmente), com geometria retangular, localizados na superfície interna ou externa do duto. Até o final deste projeto, o programa PIPEFLAW deverá comportar também outras funcionalidades como geração de múltiplos defeitos em posição arbitrária, geração de defeitos superpostos (malhas não-estruturadas) e deverá incluir defeitos com geometria retangular e elíptica.

3.2 Linguagem PCL

3.2.1 Introdução

O software MSC.PATRAN, utilizado na modelagem dos modelos de defeitos de corrosão, possui uma linguagem própria de programação PCL que é parte integral do mesmo. PCL é uma linguagem de alto nível estruturada por blocos e que fornece várias características encontradas em uma linguagem de programação tradicional. Ela permite também automatizar o processo de modelagem por meio de comandos e funções específicos para o determinado tipo de problema. Além disso, pode-se configurar o ambiente do PATRAN introduzindo novas ferramentas de interface gráfica tornando o procedimento de modelagem mais fácil e rápido (Cabral et al, 2006b; PATRAN, 2005).

Dentre as várias características da linguagem PCL podemos citar:

- Operadores para expressões aritméticas, expressões com "strings", expressões relacionais e lógicas;
- Funções intrínsecas para operações matemáticas, operações com "strings" entre outras;
- Variáveis com atribuição de tipo, escopo e dimensão;
- "Arrays" de "strings" alocados dinamicamente;
- Estrutura de controle de "loops" tais como: *while*, *for*, *list* e *repeat*;
- Controle condicional tais como: *if-then-else* e *switch-case*;
- Chamadas de funções e sub-rotinas de dentro de outras funções;
- Agrupamento de funções através de classes;

- Acesso a arquivos externos para leitura e escrita;
- Funções para geração de geometria, malha, condições de contorno, visualização e definição de elementos gráficos (interface gráfica).

Uma explanação completa da linguagem PCL pode ser encontrada na documentação disponível em (PATRAN, 2005). Portanto, mostraremos aqui apenas alguns dos principais conceitos, comandos e funções utilizadas de forma que o leitor possa ter uma noção básica da linguagem PCL, tornando assim mais fácil a compreensão dos exemplos e partes dos códigos fontes que foram implementados e que serão mostrados durante este capítulo.

3.2.2 Conceitos Básicos

Os comandos e declarações em PCL podem ser digitados na janela de linha de comandos do próprio PATRAN, processados por meio de arquivos do tipo "session file" (extensão ".ses") ou ainda podem ser executados por meio de arquivos externos do tipo PCL (extensão ".pcl"). Toda a execução do programa PIPEFLAW é feita por meio de acesso às funções e sub-rotinas implementadas em arquivos do tipo PCL uma vez que este tipo de implementação reduz muito o tempo de execução dos comandos.

Em PCL, um comentário pode ser especificado de três maneiras: iniciando com a seqüência "/*" e finalizando com a seqüência "*/", introduzindo um comentário simples usando o símbolo "$" ou "#" no início da linha, ou definindo um bloco de comentário conforme indicado na Figura 3.1.

```
$ Declaracao de Variaveis Globais do tipo real.


GLOBAL REAL RE,LD,T /*--------> Dimensoes do Duto */
GLOBAL REAL H,LC,LL,RA,RC /*--> Dimensoes do Defeito */

# Isto tambem eh um comentario simples.

/*
*
* Um bloco de comentario
* deve ser estruturado dessa forma.
*
*/
```

Figura 3.1 – Exemplos de declarações de variáveis e comentários em PCL.

Uma forma de usar o PATRAN como uma calculadora é usar a função *write* conforme exemplo da Figura 3.2 indicando o cálculo da raiz quadrada de 9 (através da função *mth_sqrt*).

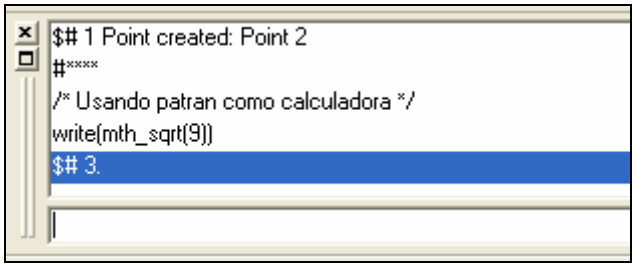


Figura 3.2 – Usando a janela de comandos do PATRAN como calculadora por meio da função *write()*.

Conforme dito anteriormente, as variáveis têm atributos de tipo, escopo e dimensão. Todas as variáveis devem ser definidas antes de serem usadas. Os principais tipos de variáveis são: *integer*, *real*, *logical*, *string* e *widget*. Uma variável do tipo *widget* é usada apenas para definição de variáveis de elementos gráficos (por ex: botões, menus, etc.) criados para interface gráfica com o usuário.

O escopo define a visibilidade da variável e seu tempo de vida. Variáveis que não apresentam definição de escopo se comportam como variáveis locais se definidas dentro de funções. A tabela Tabela 3.1 mostra os principais escopos da linguagem PCL e sua correspondente descrição.

Tabela 3.1 – Principais escopos da linguagem PCL.

Escopo	Descrição
GLOBAL	Definição da variável é visível para todas as funções que a contenham. Variáveis globais tornam-se indefinidas quando o PATRAN é finalizado.
LOCAL	Definição da variável é local dentro de uma função. Uma definição local substitui temporariamente qualquer definição global. Variáveis locais tornam-se indefinidas quando a função é finalizada.
STATIC	Definição da variável é local dentro de uma função. Uma variável de escopo *static* mantém seu valor entre chamadas de funções. Variáveis *static* tornam-se indefinidas quando o PATRAN é finalizado.
CLASSWIDE	Definição da variável é local dentro de uma classe (grupos de funções). Uma variável de escopo *classwide* mantém seu valor entre chamadas de funções. Variáveis *classwide* tornam-se indefinidas quando o PATRAN é finalizado.

A Figura 3.3 mostra alguns exemplos de definição de variáveis com escopo, tipo e dimensão. Conforme pode ser visto, qualquer variável pode ser definida como um "array" ou vetor, independente do seu tipo. Para isso, basta fornecer um número qualquer entre parênteses indicando a quantidade de posições do vetor ao lado da declaração da variável. Na Figura 3.3 pode-se verificar a definição de quatro "arrays" ou vetores. Os dois primeiros "arrays" (um de inteiro e outro de "*string*") têm uma quantidade pré-definida de posições (2 e 11 respectivamente). Ao invés de alocar diretamente um "array" com tamanho e quantidade pré-definidos, pode-se também definir "arrays" virtuais que não têm tamanho e quantidade de posições declarados no início do programa. Para isto, basta usar a palavra "*virtual*" no lugar da quantidade entre parênteses conforme as outras duas declarações de "arrays" (um de "string" e outro de "real" com escopo global). Para alocar quantidade de armazenamento no "array" e especificar os limites inferiores e superiores do "array", usa-se a função sys_allocate_array(). O exemplo da Figura 3.3 indica a alocação do "array" unidimensional Row_Labels informando o limite inferior (posição 1) e o limite superior (posição 4, que equivale ao valor da variável Num_Rows). Existe a possibilidade de realocar ou liberar espaço de um "array" por meio das funções *sys_reallocate_array()* e *sys_free_array()*. Uma "string" também pode ser definida como virtual ao invés de declarar o seu tamanho conforme definição da variável CoordDefect.

```
/*********************************************************/
/* Janela para Entrada de dados do Defeito Retangular */
/*********************************************************/

CLASS IntRetangular_dat
   /* Definicao de variaveis para Interface Grafica (botoes,janelas, etc..)*/
   CLASSWIDE WIDGET REC_WINDOW, W_RESET_BUTTON, W_BREAK_BUTTON
   .
   .
$#*************************************************************
   FUNCTION init()

   GLOBAL INTEGER MULTIDEFECTSTATUS,Num_Rows, NUMBER_OF_DEFECTS
   GLOBAL REAL LD, RE, T, LL, H, LC, RA, RC

   /* Exemplos de Definicao de 4 Arrays */
   INTEGER List_Solid(2)          /* Array de inteiros com 2 posicoes */
   STRING ListSolids[256](11)     /* Array de 11 strings com 256 caracteres cada */
   STRING Row_Labels[10](VIRTUAL)/* Array virtual de strings de 10 caracteres */
   GLOBAL REAL Z_DEFECT(VIRTUAL) /* Array virtual de numeros reais */

   /* Definicao de Strings*/
   STRING DEFECT_LOCATION[32]     /* String de 32 caracteres */
   STRING CoordDefect[VIRTUAL]    /* String virtual*/

   /* Inicializacao de variavel */
   DEFECT_LOCATION = "internal"
   Num_Rows = 4
   .
   .
   /* Alocando memoria de array virtual */
   sys_allocate_array(Row_Labels,1,Num_Rows)
   .
   .
   END FUNCTION
.
.
END CLASS
```

Figura 3.3 – Exemplos de definições de classes, funções e variáveis.

As funções PCL são definidas em arquivos que podem ser criados e modificados utilizando qualquer editor de texto. O editor utilizado foi o TextPad uma vez que já é bastante conhecido e pode ser facilmente configurável para uma determinada classe de linguagem de programação.

As funções são compiladas durante cada sessão iniciada do PATRAN por meio do comando "*!!input*" seguido do nome do arquivo da função conforme Figura 3.4.

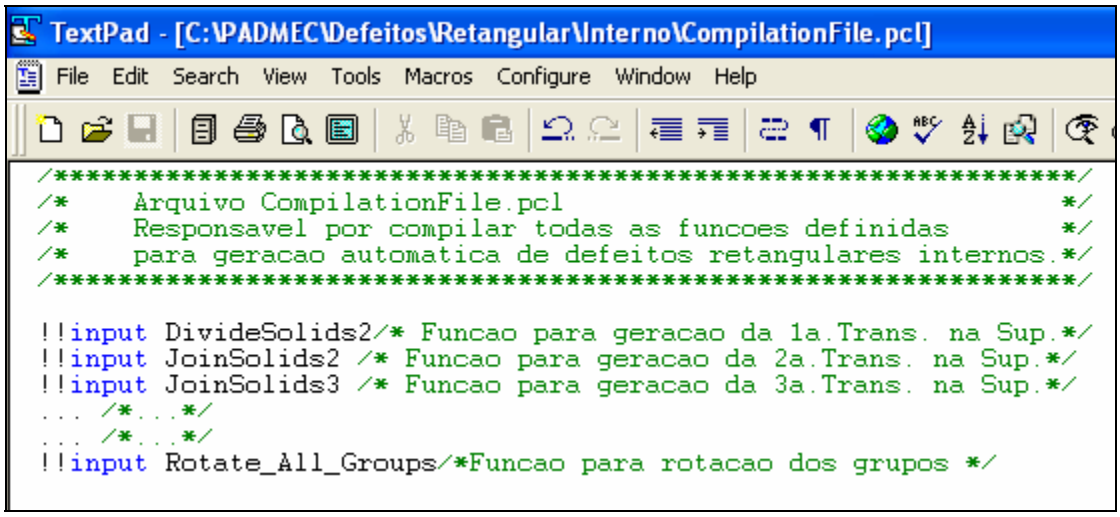


Figura 3.4 – Comando "*!!input*" para compilação e acesso às funções.

As funções podem chamar outras funções e podem assumir chamadas recursivas. As funções são iniciadas com a palavra "*function*" seguida do nome da função e finalizada com a declaração "*end function*" (Figura 3.3). A linguagem PCL fornece também a possibilidade de poder agrupar funções, que dividem variáveis em comum, por meio de classes. As classes são usadas principalmente na implementação de funções para interface gráfica personalizada. Conforme visto na Figura 3.3, a definição de uma classe inicia-se com a palavra "*class*" seguida do nome da classe e finaliza-se com a declaração "*end class*".

A depuração de erros na linguagem PCL ainda é um pouco limitada. Porém, existem basicamente duas formas de depurar erros na linguagem PCL: por meios dos comandos "*trace*" e "*debug*" que rastreiam o programa enquanto é executado. Outro comando bastante útil na depuração de erros em programas é o comando "*dump*" seguido do nome da variável. Quando o comando "*dump*" é encontrado durante a execução de qualquer função, o conteúdo da variável é exibido na janela de comandos e impresso no arquivo de "jornal" do PATRAN.

É válido lembrar que uma possível causa de erro fatal no PATRAN, encontrado durante o desenvolvimento do programa PIPEFLAW, foi devido ao mau uso de comprimento de "strings" e má definição do tamanho de "arrays". Por exemplo, suponha que no programa principal exista a seguinte declaração de uma variável global:

```
GLOBAL STRING solid[30] (2) /* Array unidimensional de strings com 2 componentes de comprimento igual a 30 caracteres cada */
```

Se em outro local do programa, geralmente dentro de uma outra função, houver a redeclaração do mesmo "array", porém com tamanho diferente (por exemplo: *GLOBAL STRING solid[30] (3)*), o PATRAN dá um erro fatal ("crash") e fecha imediatamente sem deixar nenhum rastro de onde foi o problema. Portanto, deve-se ter muita atenção quando estiver trabalhando com "arrays" e "strings" em PCL.

3.2.3 Funções Intrínsecas

Como qualquer outra linguagem de programação, a linguagem PCL dispõe de uma extensa biblioteca de funções intrínsecas identificadas com um prefixo de acordo com a categoria, por exemplo, "*mth_*" e "*str_*" que identificam as funções matemáticas e de manipulação de "strings", respectivamente. Estas funções são mais flexíveis que as outras funções de comando do PATRAN e o uso delas permite um controle sofisticado da base de dados e interface gráfica do PATRAN.

a) Funções Matemáticas

A linguagem PCL possui funções matemáticas que permitem, por exemplo, o cálculo de funções trigonométricas, realização de operações como ordenação e procura de elementos em vetores, cálculo de valores mínimos e máximos em uma dada lista, arredondamentos de números e muitas outras funcionalidades. Abaixo, mencionamos as principais funções matemáticas utilizadas durante a implementação do programa PIPEFLAW.

mth_cosd(angle)	*Retorna o valor do cosseno do ângulo especificado em graus.*
mth_sinr(angle)	*Retorna o valor do seno do ângulo especificado em radianos.*
mth_asind(value)	*Retorna o ângulo (em graus) equivalente ao valor do seno informado como argumento de entrada.*
mth_atan2d(y,x)	*Retorna o ângulo (em graus) correspondente à tangente representada pelas componentes x e y.*

mth_abs(value)	*Retorna o valor absoluto do argumento.*
mth_nint(value)	*Retorna o inteiro mais próximo do valor do argumento.*
mth_mod(value,divisor)	*Retorna o resto da divisão de um valor por um divisor.*

b) Manipulação de "strings"

Uma "string" é definida entre aspas ("") conforme exemplo de inicialização da variável *DEFECT_LOCATION* mostrado na Figura 3.3. A linguagem PCL disponibiliza várias funções para manipulação de "strings" que incluem, entre outras funcionalidades, funções para teste, comparação, conversão, procura e substituição de caracteres de "strings". As principais funções utilizadas para manipulação de "strings" foram as seguintes.

str_length(string)	*Retorna o comprimento atual da "string" especificada.*
str_substr(string, position,length)	*Retorna parte de uma string extraída a partir de uma dada posição e comprimento fornecidos.*
str_index(string1, string2)	*Retorna a posição onde uma dada string (string2) é encontrada dentro de outra (string1).*
str_find_match(string,chars)	*Retorna a posição onde qualquer um dos caracteres ("chars") for encontrado dentro de uma string.*
str_equal(string1,string2)	*Retorna verdadeiro se as duas strings são idênticas.*
str_to_integer(string)	*Converte uma string em número inteiro.*
str_to_real(string)	*Converte uma string em número real.*
str_from_integer(value)	*Converte um número inteiro em string.*
str_from_real(value)	*Converte um número real em string.*

3.2.4 Funções Geométricas

O software de pré e pós-processamento MSC.PATRAN dispõe de várias ferramentas e funções para a geração de modelos de elementos finitos. A primeira etapa da modelagem é a geração da geometria do problema (neste caso, duto com defeitos). Para isto, o PATRAN fornece uma grande variedade de funções geométricas para criar, modificar e parametrizar a geração de modelos. A seguir, serão descritas as principais funções geométricas utilizadas na geração dos elementos geométricos do modelo juntamente com os seus principais parâmetros.

a) Geração de Sistema de Coordenadas

asm_const_coord_3point(id, coord, type, Point1, Point2, Point3, Coord_id)

Descrição: Cria um sistema de coordenadas a partir de três pontos especificados.

Parâmetros de Entrada

STRING *id*[]	Especifica o identificador (id) do sistema de coordenadas.
STRING *coord*[]	Especifica o sistema de coordenadas de referência dos pontos.
INTEGER *type*[]	Especifica o tipo de sistema de coordenadas a ser criado: 1(retangular), 2(Cilíndrico) e 3(Esférico).
STRING *Point1*[]	Especifica a origem do novo sistema de coordenadas.
STRING *Point2*[]	Especifica um ponto no eixo Z do novo sistema de coordenadas.
STRING *Point3*[]	Especifica um ponto no plano X-Z do novo sistema de coordenadas

Parâmetros de Saída

STRING *Coord_id*[]	String que armazena o identificador do sistema de coordenada criado.
INTEGER *status*	Retorna o valor 0, quando a função é executada com sucesso.

A Figura 3.5 ilustra exemplo do comando para criação de um sistema de coordenadas cilíndricos.

```
GLOBAL STRING Sist_Coord_Cilindrical[VIRTUAL]
status = asm_const_coord_3point( "1",@
                                 "Coord 0",@
                                 2,@
                                 "[0 0 0]",@
                                 "[0 0 1]",@
                                 "[1 0 0]",@
                                 Sist_Coord_Cilindrical)
$# Sistema de Coordenada Cilindrico criado
```

Figura 3.5 – Exemplo de geração de um sistema de coordenadas cilíndrico.

b) Geração de Pontos

asm_const_grid_xyz(id, coordinates, sist_coord, created_id)

Descrição: Esta função constrói pontos a partir das coordenadas fornecidas num dado sistema de coordenada.

Parâmetros de Entrada	
STRING *id*[]	Especifica o identificador (id) do ponto a ser criado.
STRING *coordinates*[]	Especifica as coordenadas do ponto a ser criado.
STRING *sist_coord*[]	Especifica o sistema de coordenadas no qual o ponto será criado.
Parâmetros de Saída	
STRING *created*_id[]	String que armazena o identificador do ponto criado.
INTEGER *status*	Retorna o valor 0, quando a função é executada com sucesso.

Como ilustração do comando acima descrito, tem-se o exemplo da Figura 3.6.

```
GLOBAL STRING Point4[VIRTUAL]
asm_const_grid_xyz( "#",@
      "["//str_from_real(RE)//" "//str_from_real(Dt0)//" 0]",@
      Sist_Coord_Cilindrical,@
      Point4 )
$# 1 Ponto criado no sistema de coordenadas cilindrico
```

Figura 3.6 – Exemplo de geração de ponto no sistema de coordenadas cilíndrico.

Algumas observações podem ser feitas no exemplo acima. Note que no lugar do parâmetro do id do ponto existe a "string" "#". Toda entidade criada no PATRAN, seja ela geométrica ou relacionada à malha de elementos finitos, recebe um identificador como parâmetro na respectiva função de criação. O identificador na realidade é uma string que representa o número daquela entidade a ser criada. Num primeiro momento, poderíamos entrar o valor "4" para "string" que representa o "id" do ponto no exemplo acima. Porém, ao entrar com o valor "#" estamos permitindo que o PATRAN identifique automaticamente o "id" do ponto a ser criado de maneira seqüencial, evitando assim o conflito de pontos com mesmo "id". Isto evita também que futuras operações com o ponto criado não sejam afetadas quando a numeração de "id" for alterada devido à adição de novos comandos no programa. Ou seja, efetua-se as futuras operações com aquele ponto por meio da "string" de saída da função (*Point4*) ao invés do valor constante "*Point 4*". Observe também que se utiliza o símbolo "@" para quebrar comandos em múltiplas linhas e as duas barras "//" para concatenar "strings".

c) Geração de Curvas

Dentre as inúmeras funções de criação de curvas, basicamente as principais funções utilizadas neste trabalho foram as seguintes:

asm_const_line_2point(id, point1, point2,..., created_id)
sgm_const_curve_revolve(id, axis, angle, offset, sist_coord, point, created_id)
asm_const_line_normal(id, point, curve, created_id)
sgm_const_curve_2d_arc2point_v2(id, method, radius, ..., center_p, start_p, end_p,...)
sgm_const_curve_project_v1(id, curve, surface, type, FALSE, method, vector,..., created_id)

A primeira função (*asm_const_line_2point*) cria uma linha reta a partir de dois pontos fornecidos (*point1* e *point2*). Os outros parâmetros desta função foram omitidos por simples questão de simplificação.

A função *sgm_const_curve_revolve* cria uma curva por meio de revolução de um ponto (*point*) com um determinado ângulo (*angle*) em torno de um eixo de rotação (*axis*).

A terceira função (*asm_const_line_normal*) cria uma linha reta perpendicular à uma dada curva (*curve*) a partir de um ponto fornecido (*point*) (Figura 3.7).

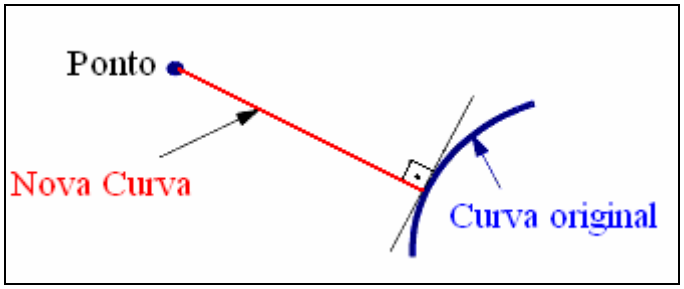


Figura 3.7 – Geração de curva via método "*normal*".

A quarta função (*sgm_const_curve_2d_arc2point_v2*) cria um arco de curva no espaço bidimensional definido por um plano de construção, um ponto central (*method = 1*) ou raio (*method = 2*), o ponto inicial e o ponto final do arco de curva. A Figura 3.8 ilustra um exemplo de geração de curva utilizando esta função a partir de um ponto central (*method = 1*).

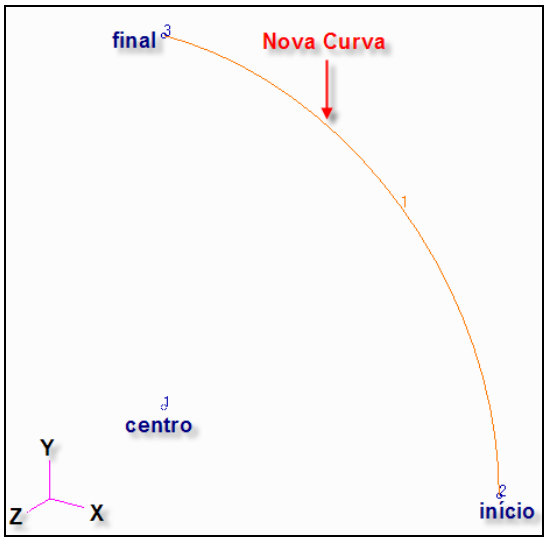


Figura 3.8 – Geração de curva via método "*2d_arc2point*" a partir de um ponto central.

A última função (*sgm_const_curve_project_v1*) cria uma curva projetada numa superfície (*type = 1*) ou num plano (*type = 2*) em uma determinada direção (*vector*). A direção de projeção é referenciada em função do método (*method*) utilizado que define se a curva será projetada na direção normal ao plano (*method = 1*), normal à superfície (*method = 2*) ou a direção é especificada por meio de um vetor (*method = 3*). A Figura 3.9 mostra um exemplo de construção deste método de projeção.

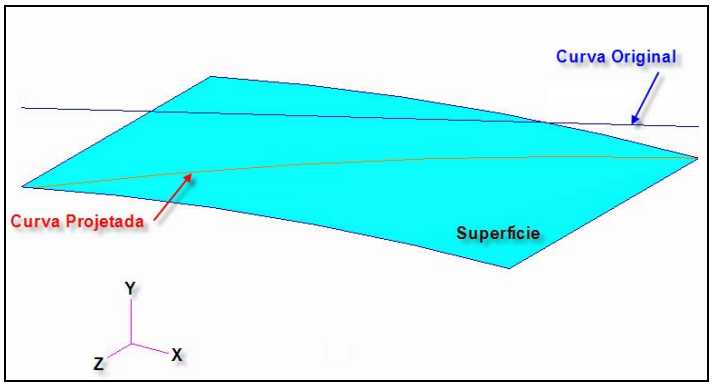


Figura 3.9 – Geração de curva via método "*project*".

d) Geração de Superfícies

Existem vários métodos para geração de superfícies disponíveis no PATRAN. A escolha do método adequado para realização das superfícies baseou-se no fato de que as malhas de elementos finitos teriam que ser do tipo estruturadas por blocos (composta por elementos hexaédricos). Este tipo de malha já é tradicionalmente utilizada pela comunidade de análise de tensões em dutos e essa foi uma das exigências do CENPES para a realização deste projeto. Além disso, a geometria do duto fica melhor modelada utilizando malhas estruturadas.

A geração deste tipo de malha só é possível no PATRAN quando a geometria gerada é paramétrica (sólido tri-paramétrico[30] congruente). Conforme será mostrado mais adiante, um dos métodos de geração de sólido tri-paramétrico utilizado neste trabalho foi o que cria um sólido a partir de duas superfícies. No caso em que se deseja construir o defeito de corrosão a partir de apenas duas superfícies, é necessário que as mesmas representem fielmente a geometria desejada. O método que se mostrou mais adequado para a geração automática de superfícies bi-paramétricas foi o método "g*lide*". Este método cria superfícies bi-paramétricas por meio de uma curva base que é "varrida" ao longo de um caminho definido por uma (*sgm_const_surface_glide*) ou duas (*sgm_const_surface_glide_2curve*) curvas diretrizes (Figura 3.10).

e) Geração de Sólidos

Conforme mencionado anteriormente, os sólidos gerados para representação da geometria do duto com defeito devem ser sólidos do tipo tri-paramétricos a fim de possibilitar a geração de malhas com elementos hexagonais. Uma forma alternativa de se obter sólidos tri-paramétricos é gerar sólidos primitivos (cilindros, esferas, elipsóides, etc.) e em seguida realizar operações de decomposição do domínio que nada mais é do que a subdivisão da geometria principal de forma a se obter sólidos menores tri-paramétricos.

[30] Nota: Um sólido é considerado tri-paramétrico quando qualquer ponto no seu interior pode ser descrito por meio de três parâmetros. Um sólido tri-paramétrico só pode ter cinco ou seis faces congruentes assumindo uma forma topológica de um cubo ou cunha.

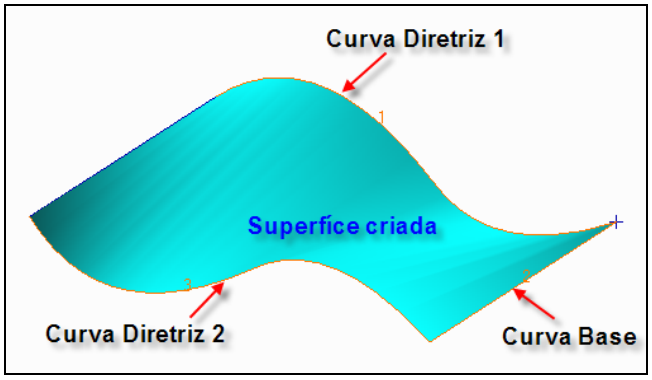


Figura 3.10 – Geração de superfícies bi-paramétricas via método "*glide*".

No entanto, as operações de decomposição do domínio, tais como as operações "booleanas"[31], são muito suscetíveis a erros e nem sempre funcionam mesmo em situações onde as geometrias são bem definidas. Experiências anteriores (Costa, 2004) mostraram que a utilização destas operações de decomposição não são muito confiáveis para o uso em procedimentos automáticos de geração de geometria. Neste contexto, a utilização do método "*glide*" de geração de superfícies foi fundamental, o que evitou que fossem feitas operações de decomposição de domínio por meio de operações do tipo interseção curva-superfície, superfície-superfície e sólido-superfície. As únicas operações de interseção utilizadas neste trabalho envolveram apenas entidades geométricas do tipo ponto-curva ou curva-curva. Estas operações, sendo bem definidas, não apresentam riscos de falha durante a geração automática.

Basicamente, as funções utilizadas na geração de sólidos foram as seguintes:

sgm_const_solid_extrude(id, dist,..., surface, created_solid)
sgm_const_solid_revolve(id, axis, angle, offset, sist_coord, surface, created_solid)
sgm_const_solid_2surface_v1(id, auto_align, surface1, surface2, created_solid)

A função (*sgm_const_solid_extrude*) cria um sólido tri-paramétrico por meio de extrusão de uma dada superfície (*surface*) ao longo de um vetor (*dist*) que define a direção e a magnitude da extrusão. Os sólidos distantes da região do defeito que tinham orientação longitudinal foram criados através desta função.

A função (*sgm_const_solid_revolve*) cria um sólido tri-paramétrico de revolução por meio de rotação de uma dada superfície (*surface*) em torno de um eixo (*axis*). Os sólidos distantes da região do defeito que tinham orientação circunferencial foram criados através desta função.

A última função (*sgm_const_solid_2surface_v1*) cria um sólido tri-paramétrico entre duas superfícies (*surface1* e *surface2*). O parâmetro lógico *auto_align*, quando assume o valor "*TRUE*", significa que as superfícies usadas na construção do sólido devem ser automaticamente alinhadas. Os sólidos que definem o perfil do defeito foram gerados através desta função utilizando as superfícies criadas por meio do método "*glide*".

f) Funções Geométricas Auxiliares

Ao longo do desenvolvimento do programa, várias outras funções geométricas disponíveis no PATRAN foram utilizadas para transladar, rotacionar e espelhar entidades geométricas. Além disso, as operações de modificação da geometria também foram utilizadas.

[31] Nota: Operações "booleanas" de sólidos são operações do tipo adição (união), subtração ou interseção entre sólidos.

Entre elas, podemos mencionar: quebra de curvas por meio de pontos (*sgm_edit_curve_break_point*), escalonamento de curvas (*sgm_transform_scale*) e extensão de curvas (*sgm_edit_curve_extend_1curve_v1*) e superfícies (*sgm_edit_surface_extend_uvl*).

3.2.5 Funções de Malha

A geração de malha é o processo de criação de uma malha de elementos finitos a partir de uma determinada geometria. No presente trabalho, todas as malhas foram geradas a partir de sólidos. O PATRAN dispõe de dois tipos de geradores de malha para sólidos: *IsoMesh* e *TetMesh*. O gerador IsoMesh foi o utilizado pois é o único que possibilita a geração de elementos hexaédricos a partir de um sólido tri-paramétrico. As várias ferramentas de geração de malha do PATRAN possibilitam que o modelo seja criado de forma rápida e eficiente por meio de ferramentas como: execução de transições de malha (*mesh_seed*), renumeração de nós e elementos, testes de verificação de malha (elementos distorcidos), associação de malha com geometria, eliminação de nós duplicados (equivalência) e várias outras. Como exemplo, segue abaixo a descrição de algumas funções de malha utilizadas.

mesh_seed_create(edge, option, number_seeds, ...)
mesh_seed_create_tabular_points(edge, point_list, tolerance)

As duas funções acima servem para criar as chamadas "mesh seeds" ao longo de uma linha (aresta do sólido). A operação de "mesh seed" é feita antes de criar a malha com o objetivo de controlar a densidade de elementos ao longo das arestas de um sólido. A primeira função (*mesh_seed_create*) cria uma determinada quantidade de "sementes" (*number_seeds*) em uma aresta (*edge*) espaçadas uniformemente ou não (dependendo da opção utilizada). Os outros parâmetros desta função servem para especificar a forma de distribuição de "sementes" quando a opção escolhida for não-uniforme.

Outra forma de definir a distribuição de elementos ao longo da aresta de um sólido é por meio da função *mesh_seed_create_tabular_points*. Esta função cria as "sementes" na aresta do sólido usando a localização dos pontos, nós ou coordenadas fornecidas no parâmetro *point_list*. A Figura 3.11 ilustra o uso destas funções em uma pequena parte do código fonte retirada do programa.

```
/*...*/
/*...*/
mesh_seed_create( Solid18//".2.1", 1, ne_esp_rem , 0., 0., 0.)
mesh_seed_create( Solid18//".3.3", 1, ne_long    , 0., 0., 0.)
/*...*/
/*...*/
mesh_seed_create_tabular_points( NewSolid//".1.4",@
                                 TabularPoints,@
                                 TOL_TABULAR)
```

Figura 3.11 – Exemplo de funções para controle de densidade de malha.

No exemplo acima, ocorre a criação de "sementes" uniformemente espaçadas (*option=1*) em duas arestas de um sólido (representado pela variável Solid18). A primeira aresta contém o número de elementos ao longo da espessura remanescente do duto (*ne_esp_rem*) e a segunda o número de elementos ao longo do comprimento longitudinal do defeito (*ne_long*). A criação de "sementes" por meio de pontos (*TabularPoints*) em uma aresta do sólido *NewSolid* também é mostrada na mesma figura. A numeração de referência dos vértices, arestas e faces de um sólido no PATRAN é dada de forma hierárquica. Por exemplo, as seis faces de um sólido X qualquer são representadas pela numeração "Solid X.1" até "Solid X.6".

Da mesma forma são as arestas pertencentes a uma determinada face ou os vértices pertencente a uma determinada aresta de um sólido (por exemplo, "Solid 18.2.1" representa uma das arestas pertencentes à face 2 do sólido 18).

Depois de definida a densidade de elementos no sólido através de operações de "mesh seed", é então gerada a malha por meio da função do PATRAN *fem_create_mesh_sol_5*.

A Figura 3.12 mostra exemplo da função *fem_create_mesh_sol_5* utilizada para geração de malha nos sólidos. A função contém 16 parâmetros de entrada e 4 parâmetros de saída. Os principais parâmetros de entrada são a "string" que representa o sólido a ser gerado (*NewSolid*), a "string" com o nome do gerador a ser utilizado ("*IsoMesh*") e a topologia do elemento a ser gerado ("*Hex8*", elemento hexaédrico com oito nós). Outros parâmetros adicionais de entrada se referem à montagem e controle de malha automático ("assembly meshing"). Neste caso, não nos interessa utilizar estes tipos de controle uma vez que já utilizamos operações de "*mesh seed*" para definir a densidade de malha em torno do sólido. Além disso, posteriormente são feitas operações automáticas de verificação de malha como a equivalência (*fem_equiv_all_group3*) dos nós e a renumeração dos nós (*fem_renum_node_1*) e elementos (*fem_renum_elem_1*).

```
/* Gerando Malha no Solido representado pela variavel NewSolid */
fem_create_mesh_sol_5( NewSolid,@
             "IsoMesh", @        /* Tipo de gerador utilizado */
             "Hex8",@            /* Topologia do elemento */
             1,@
             ["2.55432"],@       /* parametros adicionais */
             49152,@
             0,1,0,1,0.,@
             "", "#", "#",@
             "Coord 0","Coord 0",@ /*sist. coordenada de analise e referencia*/
             num_nodes, num_elems,@ /*numero de nós e elementos criados*/
             nodes_created,@       /* Ids dos nós criados: Ex: "1:100" */
             elems_created)        /* Ids dos elementos criados: Ex: "1:60" */

/* Armazenando Malha no vetor NewMesh */
NewMesh(2) = "Node "//nodes_created//" "//"Elm "//elems_created

/* NewMesh(2) = "Node 1:100 Elm 1:60"*/
```

Figura 3.12 – Função para geração de malha nos sólidos.

3.2.6 Funções Auxiliares

Além das ferramentas de geração de geometria e malha, citadas anteriormente, o PATRAN oferece várias outras funções adicionais que foram bastante úteis para o desenvolvimento do programa PIPEFLAW. Por exemplo, o PATRAN possibilita a transformação de entidades geométricas e de malha de uma só vez por meio de funções de grupo. Ao criar um grupo (*ga_group_create*), atribui-se um nome de identificação para o mesmo e a partir daí, várias entidades podem ser adicionadas ao grupo (*ga_group_entity_add*), se tornando assim, membros do grupo. As operações de transformações em grupos são movimentos de corpo rígido executados tratando um grupo como uma única entidade. Estas operações de transformações permitem transladar (*ga_group_move_translate*), rotacionar (*ga_group_transform_rotate_1*), espelhar (*ga_group_transform_mirror_1*), escalonar (*ga_group_transform_scale*), pivotar (*ga_group_transform_pivot*) ou reorientar (*ga_group_transform_position*) uma grande quantidade de entidades (membros) de uma única vez. Conforme será mostrado mais adiante neste capítulo, estas operações com grupos foram bastante úteis na geração automática de múltiplos defeitos alinhados (longitudinalmente ou circunferencialmente). Uma vez gerada a geometria e malha na região do defeito e adjacências (regiões de transição de malha), criaram-se vários grupos para cada região e em seguida, os grupos foram posicionados um a um de acordo com a localização dos defeitos múltiplos (idênticos), bastando gerar nova geometria e malha apenas nas regiões remanescentes entre

defeitos adjacentes. O acesso às informações dos membros contidos nos grupos é feito por meio da função *uil_entity_group_members_get*.

É válido mencionar que algumas funções de grupos não permitem passar uma variável no lugar do parâmetro equivalente ao nome do grupo. Por exemplo, a Figura 3.13 mostra o exemplo da função *ga_group_transform_mirror_1* utilizada para espelhar os grupos frontais da região dos defeitos gerando assim a parte traseira[32] ou espelhada dos mesmos. Os principais parâmetros de entrada da função também estão indicados.

```
/* Este exemplo nao funciona! */

ga_group_transform_mirror_1( 1,@ /* quantidade total de grupos */

         ["Current_Defect"//str_from_integer(k)],@ /* nome dos grupos */

         CurrentCoordDefect//".1",@ /* Eixo Normal ao Plano de Espelhagem */

         0., 28,@ /* parametros adicionais */

         1,@ /* Indica criacao de novo grupo apos transformacao */

         "Current_Defect"//str_from_integer(k)//"_b" )/* Novo grupo criado */
```

Figura 3.13 – Exemplo de limitação das funções de grupo: variáveis embutidas no nome do defeito não são aceitas.

A princípio, essa seria a forma na qual um determinado grupo seria espelhado. Durante a execução do programa PIPEFLAW (para o caso de múltiplos defeitos alinhados) vários grupos são criados a cada iteração "*k*" dentro de um "*loop for*". Para fazer referência aos grupos de cada defeito do modelo, utiliza-se o próprio índice "*k*". Ou seja, numa seqüência de seis defeitos alinhados longitudinalmente (caso simétrico), os grupos representando a região frontal de cada defeito são criados a cada iteração com os seguintes nomes: "Current_Defect4", "Current_Defet5" e "Current_Defect6". O problema é que a função *ga_group_transform_mirror_1* não aceita que variáveis sejam passadas no lugar do parâmetro do nome do grupo (neste exemplo, a variável é o inteiro "*k*"). Uma alternativa para resolver esta limitação foi utilizar a função *sys_eval()* que executa uma expressão ou comando do PCL contidos em uma "string". Assim, para o utilizar a função *sys_eval*, o comando apresentado na Figura 3.13 deve que ser transformado em uma string conforme Figura 3.14. Note que até o símbolo de aspas (") teve de ser definido como um caractere por meio de quatro aspas consecutivas ("""").

```
/* Agora sim. Este exemplo funciona!*/

STRING groupname[31]
STRING PCL_Command[VIRTUAL]
groupname = "Current_Defect"//str_from_integer(k)

/* Representacao do comando em forma de string */
PCL_Command = "ga_group_transform_mirror_1( 1,["//""""//groupname//""""//"],"@
//""""//CurrentCoordDefect//".1"//""""//", 0.,28, 1,"@
//""""//"Current_Defect"//str_from_integer(k)//"_b"//""""//")"

/* Executando o comando atraves da funcao sys_eval */
sys_eval(PCL_Command)
```

Figura 3.14 – Utilização da função *sys_eval()* para executar as funções de grupo.

[32] Nota: A definição e ilustração das partes frontal e traseira de cada defeito serão abordadas em mais detalhes na seção 3.6.

Outras funções especiais também foram muito úteis para a geração automática. Como exemplo, podemos citar as funções *pref_geo_set_v1* e *pref_global_set_v3* que servem para definir as tolerâncias geométricas e globais do modelo.

Dependendo da operação geométrica a ser realizada e das dimensões do modelo, essas tolerâncias são alteradas automaticamente.

As operações realizadas com as ferramentas disponíveis no PATRAN também emitem mensagens de "warning", erros ou simplesmente mensagens de informações intimando o usuário tomar alguma decisão. Por exemplo, suponha que o usuário queira criar um ponto numa posição onde já existe um ponto anteriormente criado neste mesmo local. Caso tente fazer isso, o usuário receberá uma mensagem informando que já existe um ponto naquele local e perguntando se o usuário deseja criar ou não um ponto duplicado (Figura 3.15).

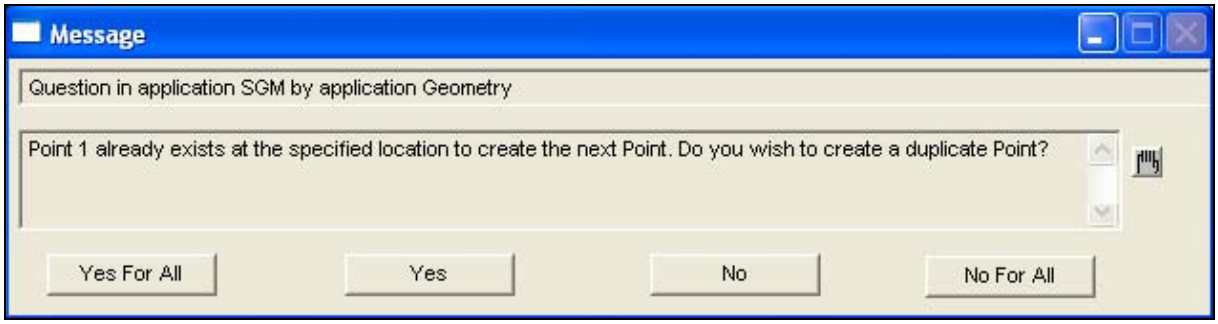


Figura 3.15 – Janela de mensagem perguntando ao usuário se deseja ou não criar um ponto duplicado.

Nem sempre é desejável que tais janelas apareçam. Além disso, a interpretação destas mensagens depende da experiência de cada usuário com o programa. Sendo assim, durante o desenvolvimento do programa PIPEFLAW procuramos ao máximo omitir mensagens desnecessárias que eram emitidas para o usuário em determinadas etapas da construção do modelo. Isto foi feito por meio das funções *ui_answer_message* e *ui_override_message* que possibilita que o programa responda automaticamente a uma determinada pergunta referenciada por um código que identificada a mensagem emitida. Este código pode ser facilmente encontrado no arquivo jornal[33] do PATRAN logo após o usuário ter respondido a mensagem gerada no processo interativo. A Figura 3.16 mostra parte do arquivo jornal gerado ao tentar criar um ponto duplicado e a Figura 3.17 ilustra parte do código fonte do programa PIPEFLAW onde necessita-se criar pontos duplicados para usar como "sementes" na geração de malha. Note que a resposta deve vir antes do comando.

```
asm_const_grid_xyz( "2", "[0 0 0]", "Coord 0", asm_create_grid_xyz_created_ids )
$# Question in application SGM by application Geometry
$#     Point 1 already exists at the specified location to create the next
$# Point. Do you wish to create a duplicate Point?
$? YES 1000034
$# 1 Point created: Point 2
$# Journal file stopped recording at 08-Jan-07 16:40:26
```

Figura 3.16 – Acessando o arquivo jornal do PATRAN para descobrir código de identificação da mensagem.

Por último, citamos as funções de baixo nível para acesso à algumas informações geométricas ou de malha que não são disponíveis na forma de funções de alto nível. Nem todas

[33] Nota: A maioria dos comandos executados no PATRAN é registrada no arquivo de jornal (jobname.db.jou) e no arquivo de sessão (patran.ses).

as informações obtidas interativamente no PATRAN são disponíveis na forma de funções mais amigáveis (alto-nível). Para isso, existe uma série de funções de baixo nível ("low level") que compõem o processador chamado "*list processor*".

```
INTEGER msgcode
INTEGER status_value
STRING answer[50]
msgcode = 1000034
/* Respondendo a mensagem (pontos coincidentes)*/
answer = "YESFORALL"

/* Resposta deve vir antes do comando!*/
ui_answer_message(msgcode,answer)

/* Criando Pontos coincidentes para usar como mesh seed tabular */
IF (Edges(1) == ".3.3") THEN
   asm_const_grid_xyz("#", "Solid "//str_from_integer(Solid_i)//".1.1.2"//@
   " "//str_from_integer(Solid_i)//".2.1.2"//@
   " "//str_from_integer(Solid_i+1)//".2.1.2", "Coord 0", TabularPoints )
ELSE
   asm_const_grid_xyz("#", "Solid "//str_from_integer(Solid_i)//".1.1.2"//@
   " "//str_from_integer(Solid_i)//".6.1.2"//@
   " "//str_from_integer(Solid_i+1)//".6.1.2", "Coord 0", TabularPoints )
END IF
```

Figura 3.17 – Utilizando o código gerado (1000034) para responder à mensagem evitando que a janela apareça para o usuário.

Estas funções recebem o prefixo "*lp_*" e a principal função que estabelece o acesso à várias informações é a função *lp_eval*.

3.2.7 Funções Gráficas

Praticamente todos os objetos de interface gráfica (*widgets*) contidos no PATRAN foram criados usando a linguagem PCL. Pode-se também configurar o ambiente do PATRAN adicionando um novo "menu" na janela principal do PATRAN. A partir daí, vários objetos ("menus", janelas, botões, caixa de diálogo para captura de dados, etc.) podem ser criados de forma a personalizar a interface gráfica do PATRAN tornando o ambiente mais amigável para o usuário. Basicamente a estrutura de uma interface gráfica é mais bem organizada através do uso de classes, onde, cada nova janela criada representada uma classe implementada. Cada classe contém geralmente um determinado grupo de funções padrões denominadas *init()*, *display()* e *refresh()* que são utilizadas para definir, apresentar e atualizar as janelas e todos os itens contidos nela.

A seguir são descritas algumas funções PCL disponíveis para criação dos principais elementos de interface gráfica utilizados no programa PIPEFLAW.

a) "Menus"

ui_menu_create(parent, callback, label, options)

Descrição: Cria um menu.

Parâmetros de Entrada

WIDGET *parent*	Identificador (*id*) do *widget* pai.
STRING *callback*	Nome da função que irá gerenciar os eventos ocorridos no *widget*. A função deve estar dentro da classe na qual o *widget* foi criado.
STRING *label*	Nome do menu.
INTEGER *options*	Argumentos opcionais.

Parâmetro de Saída

WIDGET *new_menu*	*"Id"* do menu criado. Retorna NULL se menu não for criado.

Os objetos gráficos (*widget*) são construídos de forma hierárquica. Todo objeto, com exceção das janelas (*forms*), tem um "pai" (*parent*) que é na realidade o objeto *widget* hierarquicamente superior.

A Figura 3.18 mostra parte do código fonte utilizado para criar o menu adicional "PipeFlaw" por meio da função *ui_menu_create()*. Neste exemplo, o "*widget"* pai é a barra de menus principal do PATRAN cujo identificador ("*id*") foi recuperado por meio da função *uil_primary.get_menubar_id()* e armazenado na variável MENUBAR. Ao criar o menu "PipeFlaw", seu "*id*" é armazenado na variável MENU que é então utilizada para criar itens no menu através da função *ui_item_create()*, mantendo assim a hierarquia entre os objetos criados.

```
/* Criando Menu PIPEFLAW na janela principal do Patran */
CLASS pulldown

   FUNCTION init()
      WIDGET   MENUBAR, MENU
      MENUBAR=uil_primary.get_menubar_id()
      /* Criando menu PipeFlaw */
      MENU=ui_menu_create(MENUBAR,"modell","PipeFlaw")
      /* Adicionando itens ao menu PipeFlaw */
      ui_item_create (MENU,"geo",    " Geometry ",FALSE)
      /*...*/
   END FUNCTION
      /*...*/
END CLASS
```

Figura 3.18 – Hierarquia na criação de objetos gráficos (*widgets*) em PCL.

Cada objeto *widget* está associado à determinada função conhecida como "*callback function*". A execução destas funções é dada a partir de eventos gerados. Se o usuário clicar com o "mouse" em algum componente da interface gráfica, um evento é gerado. Dessa forma, a execução do programa é controlada por eventos externos gerados pelo usuário e manipulados pelas funções "*callback*". Por exemplo, o usuário ao clicar no menu "PipeFlaw" aparecerá todos os itens contido nele (Figura 3.19). A partir daí, se o usuário clicar em um dos itens deste menu, a função *model( )* é executada direcionando o usuário para a ação desejada.

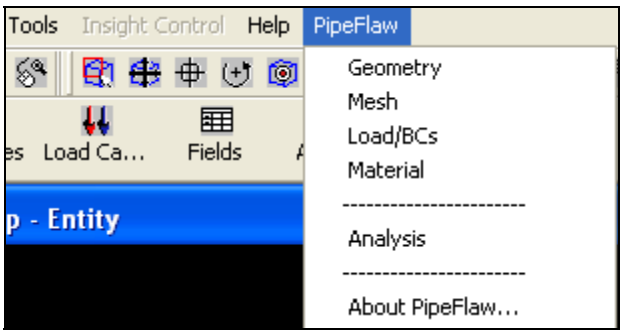


Figura 3.19 – Menu "PipeFlaw" com os seus respectivos itens.

b) Janelas ("*forms*"), "Switchs", Caixa de dados, Tabelas e Botões

As janelas ou "*forms*" servem como "background" para os objetos widgets e podem ser criadas por meio de duas funções: *ui_form_create()* e *ui_modalform_create()*. Uma janela do tipo *modalform* difere da janela normal pelo fato de que a primeira não pode conter ícones e não pode ser movida. A Figura 3.20 ilustra exemplo do uso da função *ui_form_create()* juntamente com os seus respectivos parâmetros. O primeiro parâmetro é o nome da função callback. Como as janelas não registram nenhum evento, então é passada a string vazia ("") como

parâmetro. O segundo e terceiro parâmetros são as localizações x e y (em polegadas) da janela na tela. Os valores x e y são dados em relação a um ponto de referência da tela.

```
FORM_ID = ui_form_create("",@ /* Nome da função callback. */
      form_x_location,form_y_location,@ /*Posição (x,y) da janela na tela*/
      "UL",@ /* Indica ponto de referência da tela e da janela */
      uil_form_sizes.form_wid(2),@ /* largura da janela */
      uil_form_sizes.form_hgt(2),@ /* altura da janela */
      "Rectangular Defect Parameters","")/*título da janela */
```

Figura 3.20 – Função para criação de janelas em PCL.

Esse ponto de referência é informado no quarto parâmetro ("UL" – *upper left*) significando que o ponto está localizado no canto superior esquerdo da tela. Outros valores são: "UR" (canto superior direito), "LL" (canto inferior esquerdo), "LR" (canto inferior esquerdo). A largura e altura da janela são calculadas por meio de funções de baixo nível que são bastante úteis no dimensionamento e localização de objetos *widget*[34].

Os principais objetos gráficos utilizados neste trabalho foram os "switchs", caixa para captura de dados, tabelas com células e botões. A Figura 3.21 mostra exemplo de parte do código fonte do programa PIPEFLAW onde é criado o "switch" para escolha do tipo de defeito. Os objetos criados estão indicados também na figura. O segundo argumento da função *ui_switch_create()* é o nome da função "callback" ("Choose_Defect_Shape") que irá gerenciar os eventos no objeto "switch".

```
/* Criando "frame" com titulo "Defect Shape" */
W_SHAPE_FRAME = ui_frame_create (FORM_ID,@
                "", 0.0, y_loc,@
                uil_form_sizes.form_wid(1),@
                0.75,@
                "Defect Shape",0)
/* Switch para escolha do tipo de defeito */
W_TD_SW = ui_switch_create(FORM_ID,@
            "Choose_Defect_Shape",@
            0.1,y_loc,@
            1,"",TRUE)
/* Criando os itens do switch */
W_R_IT = ui_item_create (W_TD_SW,"rec",@
          " Rectangular",TRUE)
W_E_IT = ui_item_create (W_TD_SW,"elip",@
            " Eliptic",FALSE)
```

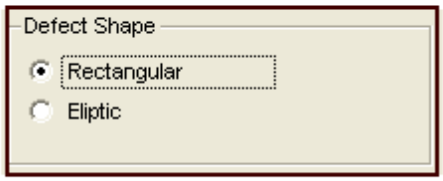


Figura 3.21 – Funções para criação de "frames", "switchs" e seus respectivos itens.

As funções "callback" têm quantidades e tipos de parâmetros pré-definidos. Em particular, as funções "callback" de objetos do tipo "switch" têm como parâmetros de entrada duas "strings": uma representando o nome do item do "switch" selecionado (*ITEM_NOME*) e a outra "string" indicando se o "switch" está ativado ("ON") ou não ("OFF"). O código fonte da função "callback" *Choose_Defect_Shape()* pode ser visto na Figura 3.22.

A Figura 3.23 mostra outro exemplo de funções para criação de objetos do tipo caixa de dados (*ui_data_box_create*) e tabela com células (*ui_spread_create*). Neste exemplo, o usuário digita os dados através da caixa de entrada "Input Data" e após teclar "enter" é acionada a função "callback" *Get_Input_from_Data_b()*. Esta função tem como parâmetro a string representando o evento ocorrido da caixa de dados.

[34] Nota: As descrições destas funções são encontradas no capítulo 5 do manual "*PCL and Customization*" ou ainda nos arquivos *appforms.p* e *uiforms.p* inclusos no pacote MSC.PATRAN.

```
FUNCTION Choose_Defect_Shape(ITEM_NOME,ONOFF)
   string ITEM_NOME[],ONOFF[]
   GLOBAL STRING DEFECT_SHAPE[VIRTUAL]
   IF (ONOFF == "ON") THEN
      IF (ITEM_NOME == "rec") THEN
         DEFECT_SHAPE = "rec"
      ELSE IF(ITEM_NOME == "elip") THEN
         DEFECT_SHAPE = "elip"
      ELSE
         write_line("Chose one defect type")
      END IF
   ELSE
      DEFECT_SHAPE = ""
   END IF
END FUNCTION
```

Figura 3.22 – Função "callback" do "switch" para escolha do tipo de defeito.

```
W_INPUT_DB = ui_databox_create(FORM_ID,@
             "Get_Input_from_Data_b",@
             0.05,y_loc,@
             0.0,0.3*uil_form_sizes.dbox_wid(1),@
             "Input Data",@
             str_from_real(POS_DATA),
             TRUE,"REAL",1)

W_DPOS_SHEET = ui_spread_create(FORM_ID,@
             "Set_Cell_Sheet",@
             0.05, y_loc,@
             uil_form_sizes.form_wid(1),@
             1.6,0.5, 1,@
             1, Num_Rows, 1,@
             ["Z [m]"], Row_Labels,@
             "Defect","", "", "SINGLE")
```

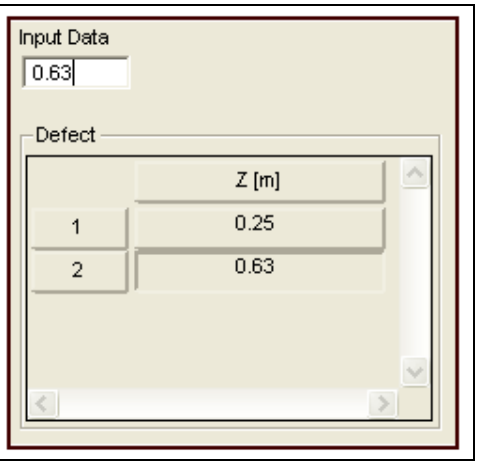


Figura 3.23 – Funções para criação de caixa de dados e tabela com células.

As "strings" que representam os eventos que podem ocorrer na caixa de dados podem ser: "GAIN" (indicando que o usuário clicou dentro da caixa de dados), "CR" (indicando que o usuário digitou "enter"), "LOSE" (indicando que o usuário clicou fora da caixa de dados) e "WIDSET".

A Figura 3.24 mostra o código fonte da função "callback" *Get_Input_from_Data_b*. Após o usuário digitar os dados e clicar "enter" (EVENT = "CR"), a função então recupera o valor digitado pelo usuário por meio da função *ui_wid_get()* e limpa o conteúdo contido na caixa de dados por meio da função *ui_wid_set()*. O valor digitado pelo usuário na caixa de dados irá aparecer na célula que estiver selecionada dentro da tabela (na Figura 3.23 a segunda célula está selecionada). Por meio da função *ui_spread_get_selected()* sabe-se qual célula está selecionada, e em seguida, é exibido o valor na célula por meio da função *ui_spread_set_cell()*. Para agilizar o processo de entrada de dados e evitar que o usuário tenha que selecionar manualmente com o "mouse" cada célula, utilizou-se a função *ui_spread_set_selected()* que seleciona automaticamente a próxima célula para que o usuário digite o próximo valor, seguindo assim até que a tabela fique totalmente preenchida.

Outro objeto também bastante utilizado foi o botão. Este objeto além de ser muito útil para o direcionamento do usuário dentro do programa é também muito simples de implementar. A Figura 3.25 mostra exemplo de criação do botão "Apply" da janela principal do PIPEFLAW por meio da função *ui_button_create*.

```
FUNCTION Get_Input_from_Data_b(EVENT)
   STRING EVENT[]
   /* EVENT = "GAIN", "CR", "LOSE" or "WIDSET"*/
   INTEGER START_COL,START_ROW,END_COL,END_ROW,LAYER
   GLOBAL STRING DEFECT_SHAPE[VIRTUAL],DEFECT_LOCATION[VIRTUAL]
   GLOBAL INTEGER MULTIDEFECTSTATUS,SINGLE,LONGALIGNED
   GLOBAL INTEGER CIRCALIGNED,ARBITRARY,SIMETRICO,NAOSIMETRICO
   REAL POS_DATA

   IF(EVENT == "CR")THEN
      ui_wid_get (W_INPUT_DB,"VALUE",POS_DATA)
      ui_wid_set (W_INPUT_DB,"VALUE","")
      ui_spread_get_selected (W_DPOS_SHEET,@
                     START_COL, START_ROW,@
                     END_COL, END_ROW, LAYER)
      ui_spread_set_cell(W_DPOS_SHEET,@
                     START_COL, START_ROW,@
                     LAYER, str_from_real(POS_DATA))
      ui_spread_set_selected (W_DPOS_SHEET,@
                     START_COL, START_ROW+1,@
                     END_COL, END_ROW+1)
      IF ( (START_ROW+1)> 3 )THEN
         ui_spread_set_top_row (W_DPOS_SHEET, START_ROW)
      END IF
   ELSE
      write_line(".....")
   END IF
END FUNCTION
```

Figura 3.24 – Função “callback” da caixa de dados “Input Data”.

```
W_APLLY_B = ui_button_create(FORM_ID,@
                  "Apply_b",@
                  5*uil_form_sizes.button_x_loc1(1),@
                  y_loc, 0.5*uil_form_sizes.button_wid(1),@
                  0.0, "- Apply -",TRUE,TRUE)
```

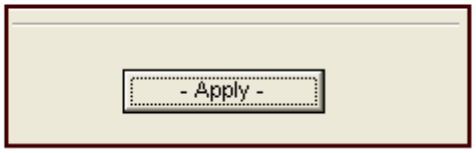


Figura 3.25 – Função para criação de botões.

Após o usuário clicar no botão “Apply”, é acionada a função “callback” *Apply_b()* que por sua vez aciona a função *Model_Generation()* que dará início ao processo de modelagem automático. Observe na Figura 3.26 que a função “callback” *Apply_b()* não possui nenhum parâmetro de entrada uma vez que o objeto do tipo botão tem a função apenas de direcionar o usuário para outras etapas do programa, que neste caso, é o início da geração da modelagem automática.

```
FUNCTION Apply_b()
   Model_Generation()
END FUNCTION
```

Figura 3.26 – Função “callback” do botão “Apply”.

3.3 Estrutura Geral do Programa

O programa PIPEFLAW compõe todo o conjunto de funções e classes de interface gráfica implementados na linguagem PCL para a geração automática de modelos de elementos finitos de defeitos de corrosão em dutos. Com as ferramentas desenvolvidas até o momento, é possível gerar automaticamente modelos de dutos com defeitos de corrosão com geometria retangular, situados na superfície interna ou externa do duto e podendo assumir configurações de defeito isolado ou múltiplos defeitos alinhados (longitudinalmente ou circunferencialmente). A estrutura do programa foi construída de forma a possibilitar a adição de novas configurações de defeito (por exemplo, múltiplos defeitos em posição arbitrária) e novas geometrias (por exemplo, defeito elíptico).

A integração da interface gráfica do programa PIPEFLAW com o sistema PATRAN é feita por meio da leitura do arquivo *init.pcl* durante a inicialização do PATRAN. Geralmente, este arquivo não é modificado. Ao invés disso, o arquivo *init.pcl* (fornecido pelo pacote MSC.PATRAN) executa os arquivos adicionais de inicialização *p3prolog.pcl* e *p3epilog.pcl*. Estes arquivos, se existirem, são executados no início (*p3prolog.pcl*) e no final (*p3epilog.pcl*) do processamento do arquivo *init.pcl*. O uso do arquivo *p3epilog.pcl* permite o usuário adicionar ao ambiente do PATRAN a interface gráfica personalizada que foi implementada em PCL. Este arquivo deve ser adicionado no diretório padrão de instalação do PATRAN (*C:\MSC.Software\MSC.Patran\2005*) ou em qualquer outro diretório de onde o PATRAN é executado. Conforme mostrado na Figura 3.29, o arquivo *p3epilog.pcl* é lido durante a inicialização do PATRAN pelo arquivo *init.pcl*. É justamente no arquivo *p3epilog.pcl* onde são executados os comandos para "carregar" as funções e classes implementadas para o funcionamento do programa PIPEFLAW. O conteúdo inteiro do arquivo *p3epilog.pcl* é mostrado na Figura 3.27. O primeiro comando do arquivo *p3epilog.pcl* é uma diretiva para adicionar diretórios que serão acessados pelo PATRAN para leitura de arquivos utilizados no programa PIPEFLAW. Os comandos de *!!input* "carregam" as classes e funções implementadas. Por último, é executada a função *init()* da classe *pulldown*. Como resultado disso, aparecerá o menu adicional "PipeFlaw" da interface gráfica personalizada na barra de menus da janela principal do PATRAN (Figura 3.28).

```
/* Adicionando diretorios...*/
!!PATH "C:\PADMEC\Defeitos\Retangular"

/* Carregando Interface Grafica e Funcoes...*/
!!input PipeFlaw_UI.pcl
!!input CompilationFile

/* Inicializando interface grafica... */
ui_exec_function("pulldown","init")
```

Figura 3.27 – Conteúdo do arquivo *p3epilog.pcl*.

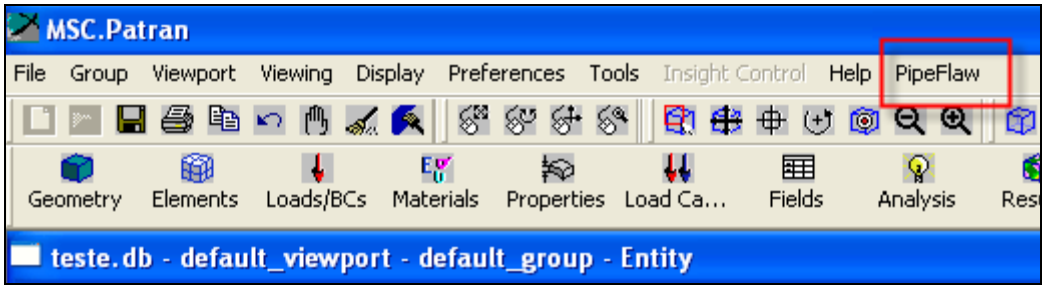


Figura 3.28 – Janela principal do PATRAN com o novo menu "PipeFlaw" adicionado.

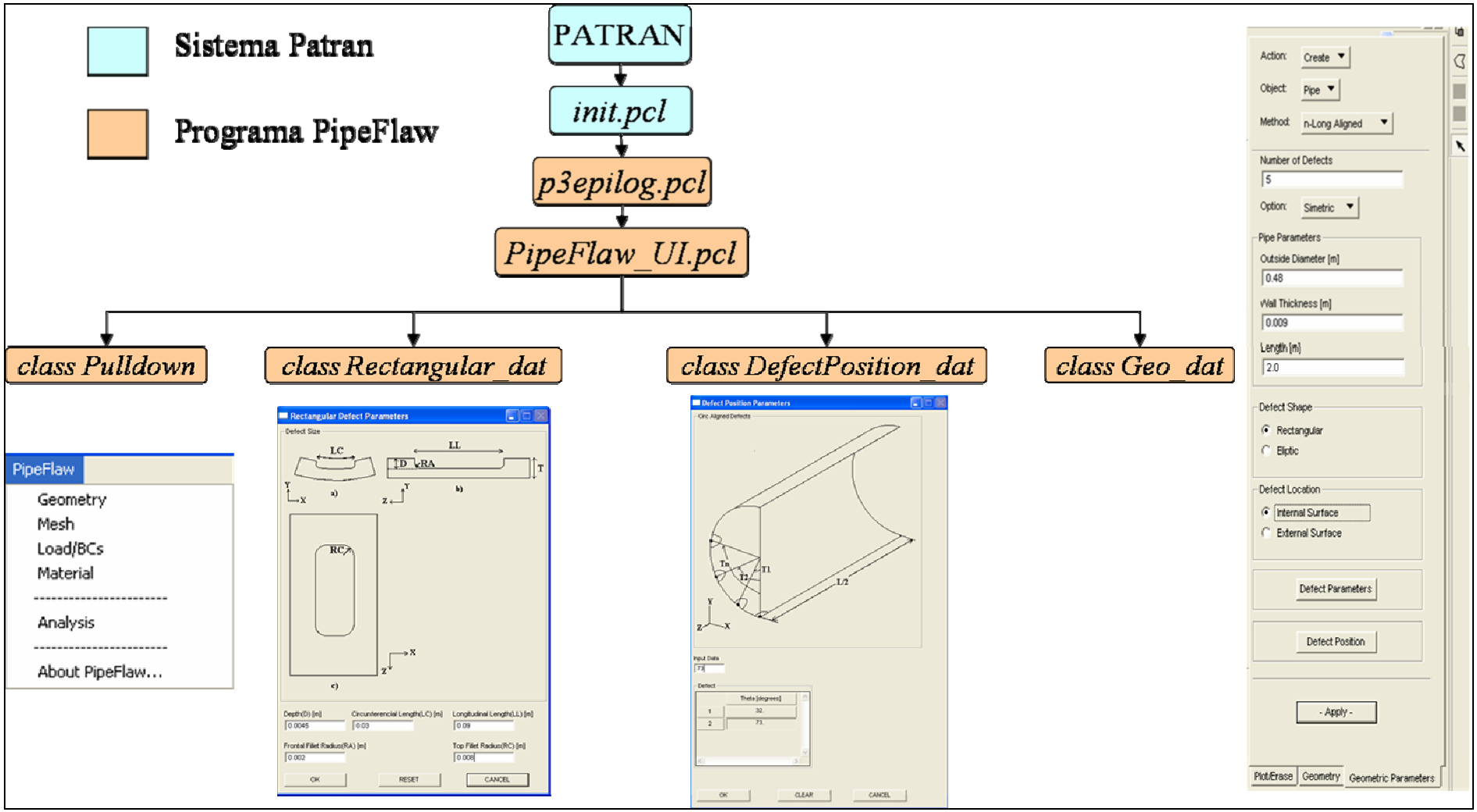


Figura 3.29 – Estrutura geral do programa PIPEFLAW com suas classes principais.

Conforme foi visto na seção 3.2.7, o usuário ao clicar no menu "PipeFlaw", aparecerá os itens contidos nele conforme Figura 3.19. Esses itens representam as principais etapas do processo de modelagem e podem ser estruturados em forma de classes definidas dentro do arquivo *PipeFlaw_UI.pcl*. A Figura 3.30 mostra as principais classes implementadas até o momento para o gerenciamento dos eventos ocorridos na interface gráfica personalizada.

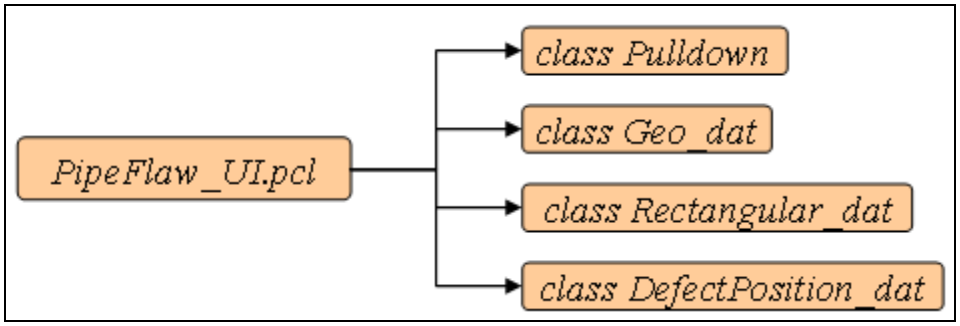


Figura 3.30 – Principais classes implementadas para o gerenciamento dos eventos da interface gráfica.

3.3.1 Classe *Pulldown*

Esta classe é responsável por criar e exibir o menu principal "*PipeFlaw*" (Figura 3.19) de onde as principais etapas de modelagens são solicitadas pelo usuário. Os atributos desta classe são:

- *function init()* : cria o menu principal da interface com os seus respectivos itens.

-*function modell()*: função "callback" que serve para manipular os eventos que ocorrem no menu principal. É a partir desta função que as outras janelas irão aparecer de acordo com o item escolhido pelo usuário.

3.3.2 Classe *Geo_dat*

Esta classe é responsável por criar uma janela com vários objetos gráficos (menus, caixas de diálogo para captura de dados, "switchs" e botões) para que o usuário possa informar os principais parâmetros geométricos para a modelagem do duto com defeito (Figura 3.31).

Os atributos desta classe são:

- *function init()* : cria a janela de modelagem e todos os objetos gráficos contidos nela.
- *function Choose_Defect_Shape()*: função "callback" que serve para manipular os eventos que ocorrem no "switch" criado para escolha do tipo de defeito a ser modelado.
- *function Choose_Defect_Location()*: função "callback" que serve para manipular os eventos que ocorrem no "switch" criado para definição da superfície onde está localizado o defeito.
- *function Defect_Parameters_b()*: função "callback" do botão "Defect Parameters" cujas principais tarefas são recuperar os valores digitados pelo usuário na caixa de dados relacionados com os parâmetros do duto "Pipe Parameters" (ilustrado na Figura 3.31) e acionar a função que exibirá a janela para entrada de dados dos parâmetros do defeito.
- *function Defect_Position_b()*: função "callback" do botão "Defect Position" responsável por acionar a função que irá exibir a janela para entrada de dados das posições dos defeitos. Note na Figura 3.31 que este botão e a caixa de dados (relacionada com o número de defeitos) estão desabilitados uma vez que o método selecionado foi "Single Defect".

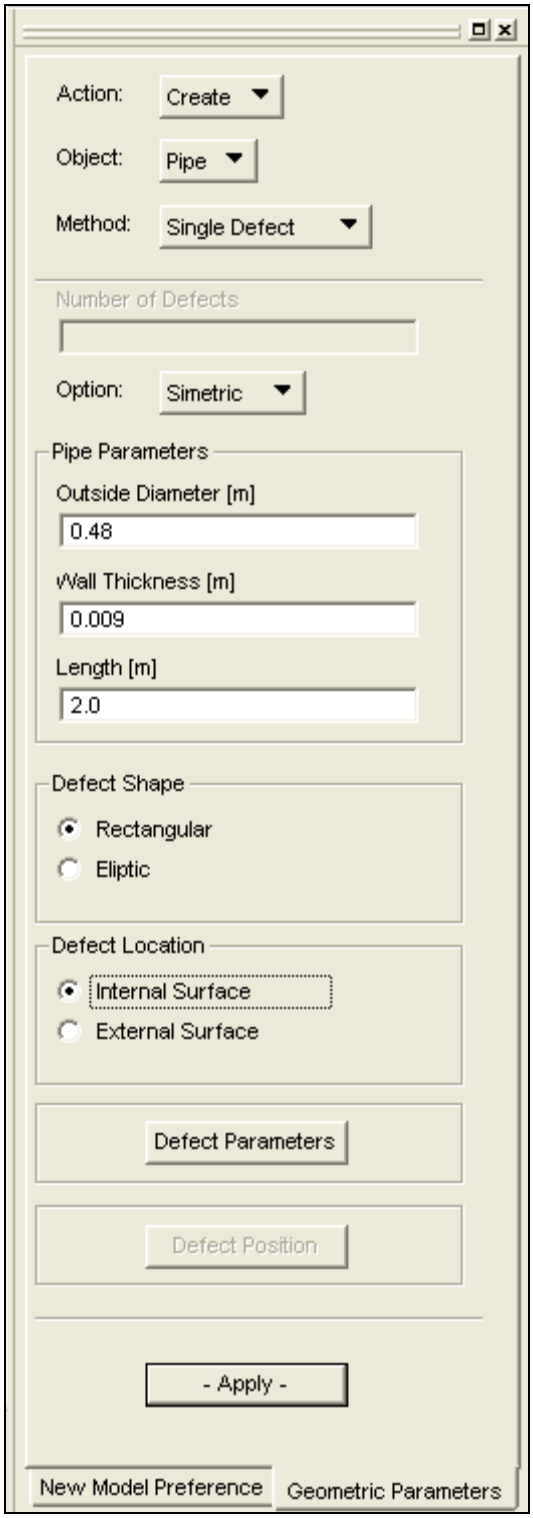


Figura 3.31 – Janela para entrada dos principais dados para geração do modelo.

Estes objetos só ficam ativados quando o usuário selecionar um método de modelagem de múltiplos defeitos, situação na qual são necessárias informações adicionais sobre o número de defeitos e a posição de cada um, ou quando o usuário optar em gerar o modelo sem considerar simetria (menu “Option” = “Non-Simetric”).

- *function Get_Multi_Defect_Status()*: função "callback" do menu de opções "Method". É justamente essa função que verifica qual método foi escolhido pelo usuário (Figura 3.32) ativando ou não a caixa de dados "Number of Defects" para entrada do número de defeitos e o botão "Defect Position" para entrada de dados da posição de cada defeito (Figura 3.33).

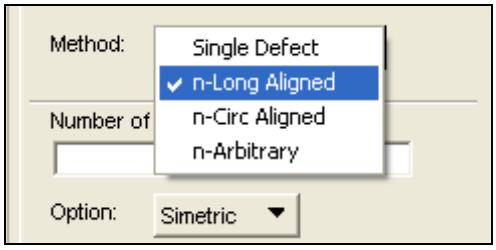


Figura 3.32 – Menu de opções para modelagem com diferentes tipos de configurações de defeitos.

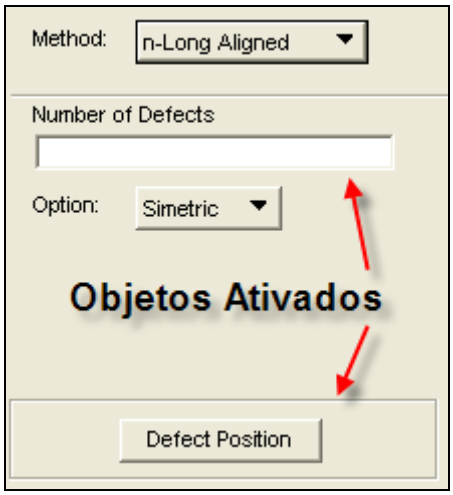


Figura 3.33 – Situação onde os objetos (caixa de dados e botão) estão ativados.

- *function Get_Simetric_Option()*: função "callback" do menu de opções "Option". Essa função ativa o botão "Defect Position" quando a opção escolhida for "Non-Simetric". Ou seja, será gerado todo o duto e não apenas ¼ do duto.

- *function display()*: função responsável apenas por exibir a janela da classe *Geo_dat* com todos os objetos atualizados. Ela não cria nenhum objeto gráfico.

- *function Apply_b()*: função "callback" do botão "Apply". Neste momento, o usuário ao clicar neste botão deve ter informado todos os parâmetros corretamente. Esta função filtra todos os dados informados pelo usuário e verifica se há algum dado faltando ou informado de forma errada. Só após essa verificação é que é acionada a função *Model_Generation()* que dará início ao processo automático de geração do modelo.

3.3.3 Classe *Rectangular_dat*

Esta classe é responsável por criar a janela com caixas de diálogo para captura dos dados relacionados com as dimensões geométricas do defeito retangular tal como ilustrado na Figura 3.34. Os atributos desta classe são:

- *function init()*: cria a janela "Rectangular Defect Parameters" (Figura 3.34) e todos os objetos gráficos contidos nela (por exemplo: ícones, caixas de dados e botões).

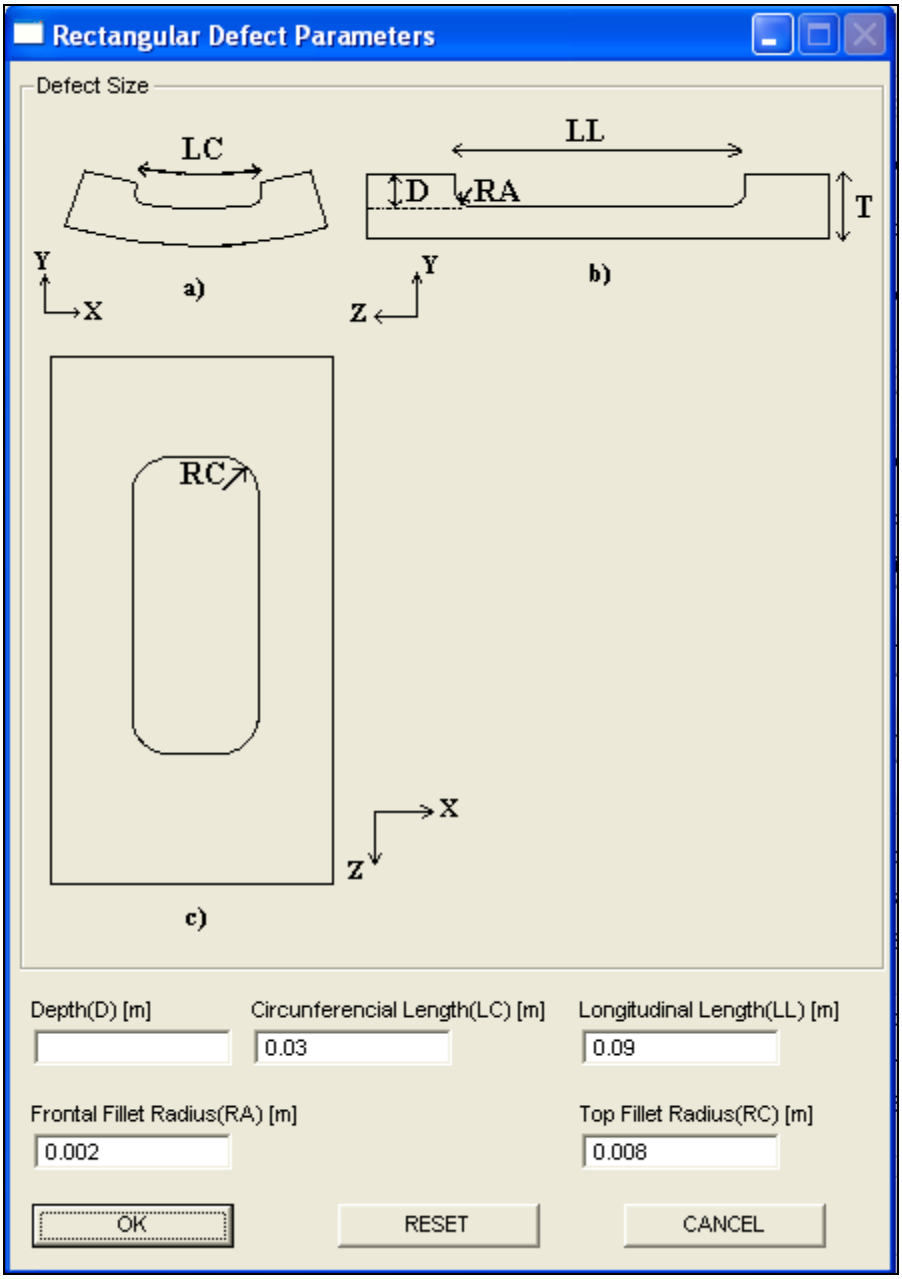


Figura 3.34 – Janela para captura dos dados relacionados às dimensões do defeito retangular.

- *function display()*: exibe a janela ao usuário.

- *funtion ok_b()*: função "callback" do botão "OK". Recupera os valores digitados pelo usuário após verificar se todos os dados foram informados corretamente. Caso haja algum problema, é emitida mensagem indicando o motivo do erro. Por exemplo, na Figura 3.34 a caixa de dados relacionada com a profundidade do defeito não foi preenchida. Após clicar no botão "OK", o usuário recebe a mensagem de que estão faltando dados necessários à modelagem e que devem ser informados (Figura 3.35).

- *function reset_b()*: função "callback" do botão "Reset". Exibe os últimos valores informados pelo usuário salvos na função ok_b().

- *function cancel_b()*: função "callback" do botão "Cancel". Fecha a janela sem salvar os dados.

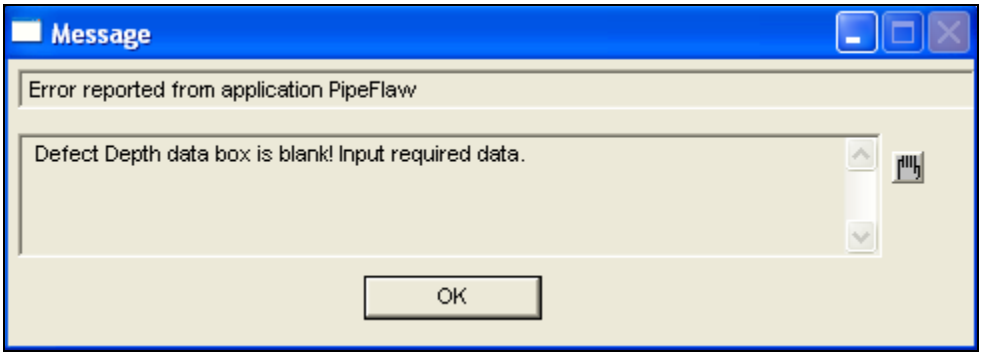


Figura 3.35 – Mensagem indicativa de erro na entrada de dados.

3.3.4 Classe *DefectPosition_dat*

Esta classe é responsável por criar a janela "Defect Position Parameters" com objetos gráficos para capturar as posições dos múltiplos defeitos alinhados longitudinalmente (Figura 3.37) ou circunferencialmente (Figura 3.38).

Os atributos desta classe são:

- *function init()*: cria a janela e todos os objetos gráficos contidos nela (ícones, caixa de dados e tabela).
- *function display()*: exibe a janela ao usuário.
- *function Get_Input_from_Data_b()*: função "callback" da caixa de dados "Input Data". Recupera os valores digitados pelo usuário e transfere-os para a tabela de dados.
- *function ok_b()*: função "callback" do botão "OK". Armazena as posições dos defeitos em um vetor global e fecha a janela após salvar os dados.
- *function clear_b()*: função "calback" do botão "Clear". Apaga todos os valores da tabela.
- *function cancel_b()*: função "callback" do botão "Cancel". Fecha a janela sem salvar os dados.

A janela para captura de dados relacionados com as posições dos defeitos possui uma caixa de dados para que o usuário entre com as posições individuais de cada defeito e uma tabela onde os valores são exibidos para o usuário. Para o caso de múltiplos defeitos alinhados longitudinalmente, o usuário deve fornecer a distância (em metros) do centro de cada defeito a partir do plano XY de simetria (Figura 3.37). De forma similar, para o caso múltiplos defeitos alinhados na direção circunferencial, o usuário deve fornecer o ângulo em graus (a partir do plano YZ) que define o posição central do defeito conforme ilustrado no ícone da Figura 3.38.

Vale lembrar que os ícones com ilustrações foram adicionados à interface gráfica por meio da função *ui_labelicon_create()*. A utilização de ícones facilita muito a comunicação com o usuário permitindo que o mesmo identifique de forma clara e rápida como utilizar os objetos contidos na interface gráfica. A Figura 3.36 mostra parte do código fonte onde é criado o ícone para o caso de "n" defeitos alinhados longitudinalmente mostrado na Figura 3.37.

```
ui_labelicon_create (FORM_ID,"",@
                     0.1,y_loc,@
                     "Long_Aligned_odd.bmp" )
```

Figura 3.36 – Função para criação de ícones.

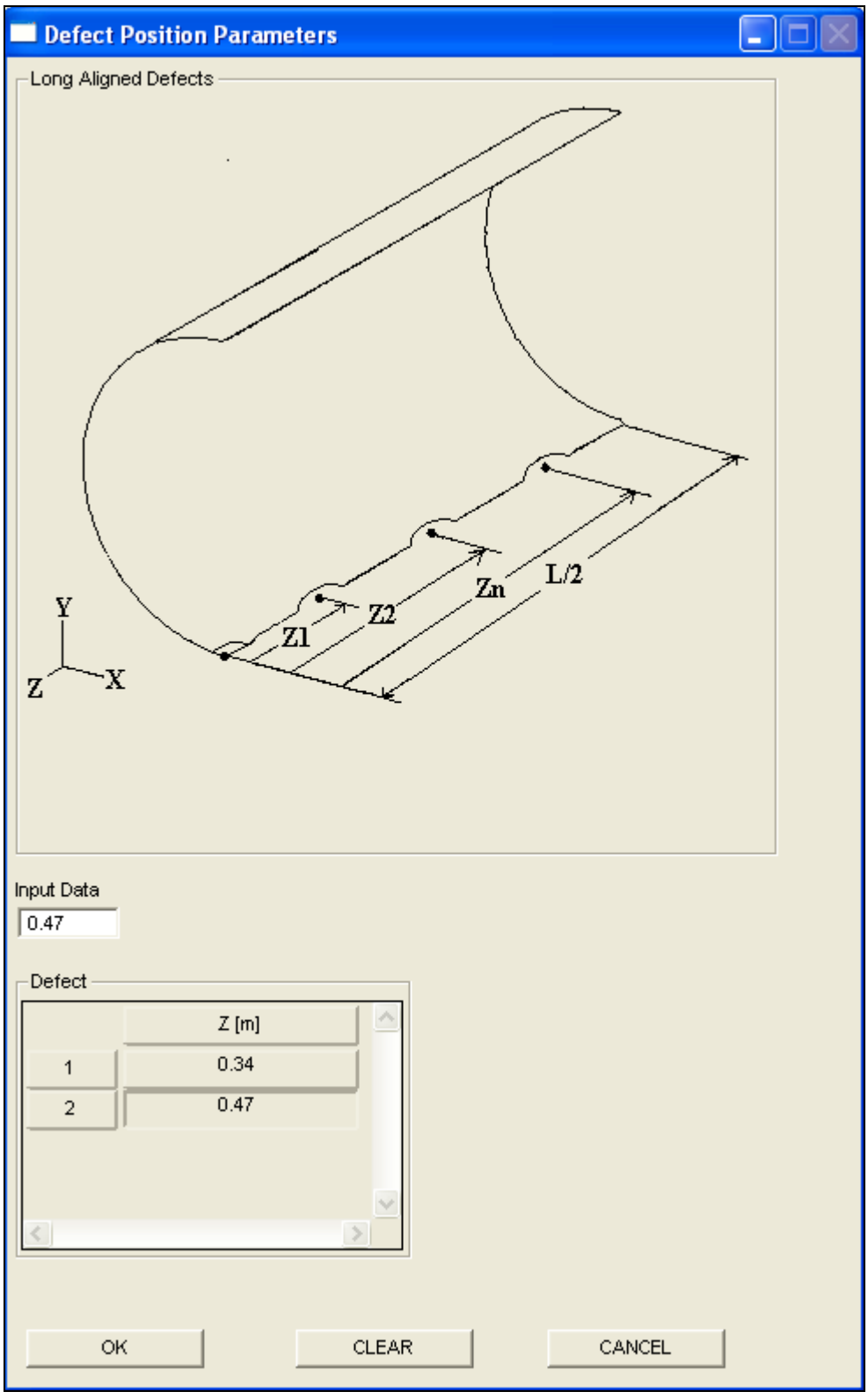


Figura 3.37 – Janela para entrada das posições dos defeitos alinhados longitudinalmente.

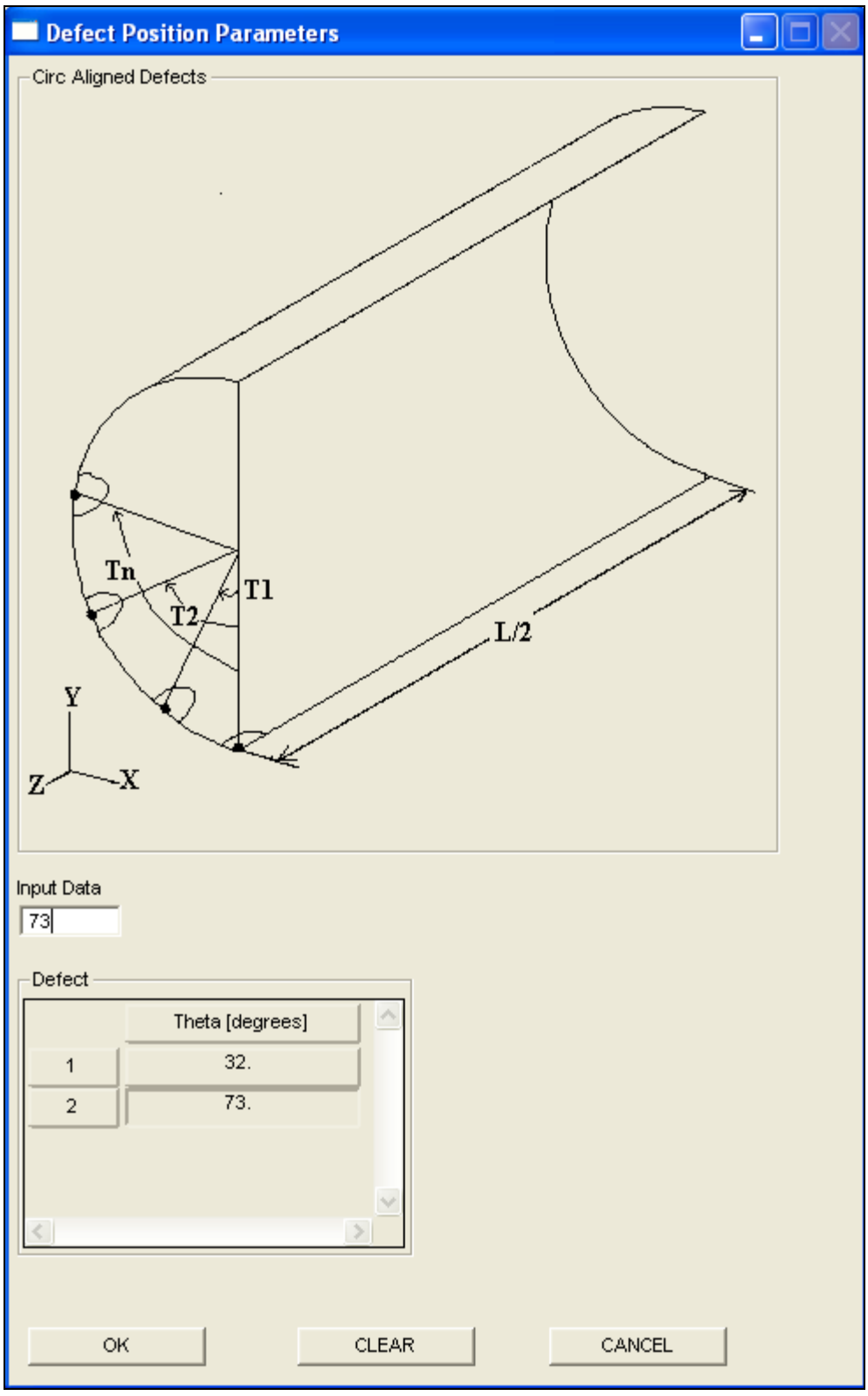


Figura 3.38 – Janela para entrada das posições dos defeitos alinhados circunferencialmente.

A seguir, é apresentado um resumo das principais funções implementadas para o processo automático de geração de modelos do programa PIPEFLAW.

3.3.5 Funções Implementadas

Todo o processo de geração automática dos modelos de dutos com defeitos é executado por meio de várias funções implementadas em PCL que, junto com o arquivo de interface gráfica, formam o programa PIPEFLAW.

A Figura 3.39 mostra um fluxograma simplificado das principais tarefas executadas durante a geração automática dos modelos.

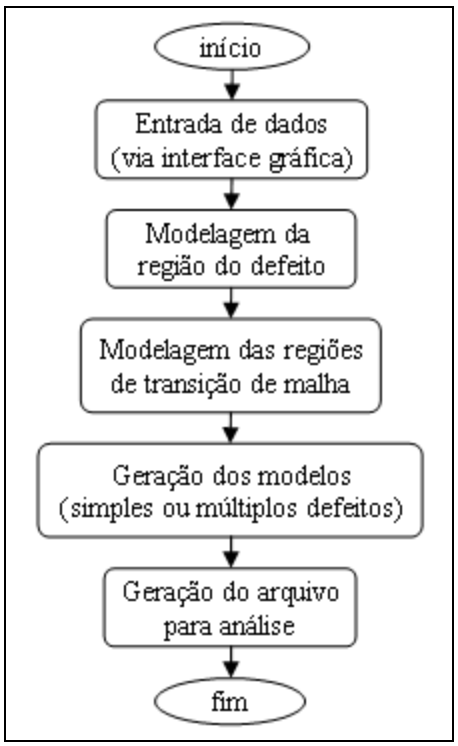


Figura 3.39 – Fluxograma simplificado do programa PIPEFLAW.

Após a obtenção dos dados de entrada por meio da interface gráfica descrita na seção anterior, é então iniciado o processo de geração automático do modelo. Até o momento foram implementadas 62 funções em PCL para a geração dos modelos de defeitos (simples ou múltiplos alinhados) com geometria retangular localizados na superfície interna ou externa do duto.

A primeira etapa executada pelo programa é a modelagem da região do defeito. Esta etapa inclui a geração da geometria e malha do defeito. A Tabela 3.2 mostra um resumo das principais funções implementadas para execução desta etapa. As cinco primeiras funções são responsáveis pela geração da geometria do defeito enquanto que a última (*Mesh_Defect*) é responsável pela geração de malha nos sólidos do defeito. O número de linhas de cada função também está indicado na Tabela 3.2.

A segunda etapa corresponde à modelagem das regiões de transição de malha adjacentes ao defeito. Essas regiões representam diferentes níveis de refinamento de malha de forma que à medida que vão se aproximando do defeito elas vão assumindo um maior nível de refinamento para melhor capturar os gradientes de tensão naquelas regiões. Esse assunto será abordado mais detalhadamente ainda durante este capítulo. As principais funções implementadas para executar esta etapa podem ser vistas na Tabela 3.3.

Tabela 3.2 – Principais funções implementadas para a modelagem da região do defeito.

Objetivo	**Funções Implementadas**	
	Nome da função	**Número de Linhas**
Geração dos sólidos na direção longitudinal	*Generate_LongSolids()*	152
Geração dos sólidos na direção circunferencial	*Generate_CircSolids()*	101
Geração dos sólidos na região de concordância	*Generate_CentralSolids()*	825
Geração dos sólidos em torno do defeito	*Generate_SolidsAroundDef()*	213
Geração dos sólidos retangulares do defeito	*Generate_RecSolids()*	267
Geração de malha nos sólidos do defeito	*Mesh_Defect()*	251

Tabela 3.3 – Principais funções implementadas para a modelagem das regiões de transição de malha.

Objetivo	**Funções Implementadas**	
	Nome da função	**Número de Linhas**
Geração da 1ª. Região de Expansão de Malha, logo após o Defeito.	*FirstExpansionRegion()*	280
Geração da Região de Transição ao longo da Espessura do Duto.	*ThicknessTransition()*	648
Geração da 1ª. Transição na Superfície	*FirstSurfaceTransition()*	512
	DivideSolids2()	569
Geração da 2ª. Região de Expansão de Malha.	*SecondExpansionRegion()*	266
Geração da 2ª. Transição na Superfície	*SecondSurfaceTransition()*	427
	JoinSolids2()	767
Geração da 3ª. Transição na Superfície	*ThirdSurfaceTransition()*	629
	JoinSolids3()	606

A Tabela 3.4 apresenta a lista das principais funções implementadas para a geração dos modelos de dutos com defeito simples, múltiplos defeitos alinhados longitudinalmente e múltiplos defeitos alinhados circunferencialmente. Para o caso de defeito simples, é gerada a geometria (sólidos) e malha nas regiões complementares do duto por meio das funções *Generate_Single_Defect()* e *Generate_Complem_Solids()*. Todas as outras funções foram implementadas para o caso de múltiplos defeitos alinhados. Uma parte destas funções trabalha com operações de transformação em grupos individuais que representam cada região do modelo discretizado. A definição destes grupos será vista mais na frente neste capítulo. O restante das funções foram implementadas para gerar a geometria e malha complementares nas regiões remanescentes entre defeitos adjacentes.

Tabela 3.4 – Principais funções para modelagem de defeitos simples ou múltiplos alinhados.

Objetivo	**Funções Implementadas**	
	Nome da função	**Número de Linhas**
Geração de Defeito Simples.	*Generate_Single_Defect()*	275
Geração de "n" Defeitos Alinhados Longitudinalmente.	*Generate_n_LongAlignedDef()*	1757
Geração de "n" Defeitos Alinhados Circunferencialmente.	*Generate_n_CircAlignedDef()*	1748
Geração dos Sólidos Complementares do Duto.	*Generate_Complem_Solids()*	498
Translação/Rotação das Regiões Frontais de Transição	*Translate_Front()*	162
	Temp_Translate_Front()	808
	Rotate_Front()	159
	Temp_Rotate_Front()	819
Espelhar Regiões de Transição	*Mirror_Back()*	133
Geração das Regiões Remanescentes (frontal e traseira) entre defeitos Adjacentes	*Generate_RemFront_Region()*	1041
	Generate_RemFront_Region_Circ()	1145
	Generate_RemBack_Region()	1031
	Generate_RemBack_Region_Circ()	1101
Geração da 1ª. Transição na Superfície	*DivideSolids_Circ()*	321
Geração da 2ª. Região de Expansão de Malha na região remanescente.	*SER_Rem()*	120
Geração da 2ª. Transição na Superfície	*JoinSolids_Circ()*	556
Geração da 3ª. Transição na Superfície	*JoinSolids_Circ3()*	413

Por último, na Tabela 3.5 são apresentadas algumas funções auxiliares bastante úteis que foram implementadas durante o desenvolvimento deste trabalho. Os principais objetivos na implementação destas funções foram: facilitar o processo de extração de "strings" dos nomes dos grupos (para realização das operações de transformações dos grupos), recuperar as faces frontais (para geração dos sólidos na região remanescente entre defeitos via extrusão ou revolução), armazenar as superfícies internas do modelo (para aplicação automática de carregamento de pressão interna) e realizar cálculos de pré-processamento. Além disso, a implementação destas funções tornou o enorme e complexo código fonte do programa mais legível.

Tabela 3.5 – Funções auxiliares implementadas.

Objetivo	Funções Implementadas	
	Nome da função	**Número de Linhas**
Manipulação de S-trings	*Extract_First_Last_ID()*	108
	Extract_Front_Faces()	170
	Get_Internal_Surfaces()	1061
Cálculo do Número de Elementos na Região Remanescente	*Compute_nelm_Remain()*	401
Cálculo dos Limites de Interação entre Defeitos Adjacentes	*Compute_InteractingParameters()*	130
Rotação de Grupos	*Rotate_All_Groups()*	40

3.4 Principais Etapas da Modelagem Automática

Após o usuário fornecer os dados para a modelagem via interface gráfica personalizada, inicia-se a geração automática da geometria do defeito. A seguir, são descritas, de forma resumida, as principais etapas envolvidas no processo de geração da geometria e malha de um defeito retangular interno. Essas mesmas etapas são executadas para o caso de defeito retangular externo.

a) Geração dos sólidos nas direções longitudinais e circunferenciais

As funções implementadas para a geração dos sólidos da região do defeito nas direções longitudinais e circunferências foram *Generate_LongSolids()* e *Generate_CircSolids()* respectivamente. Conforme pode ser visto na Figura 3.40, primeiro gera-se as superfícies bases via o método "*glide*".

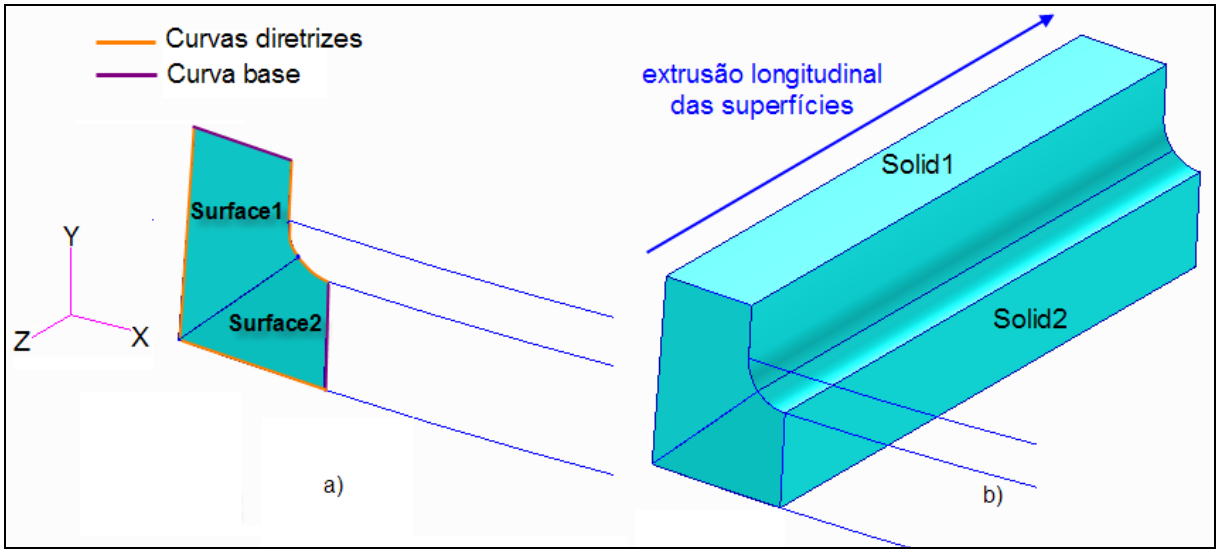


Figura 3.40 – Geração dos sólidos na direção longitudinal via extrusão.

As respectivas curvas diretrizes e base utilizadas na geração das superfícies estão indicadas também na Figura 3.40.

A geração dos sólidos longitudinais é então executada a partir destas superfícies criadas por meio de uma operação de extrusão (*sgm_const_solid_extrude*). A distância de extrusão equivale ao comprimento longitudinal do defeito menos o raio de concordância (RC) (ver Figura 3.34).

Em seguida, geram-se as superfícies que irão compor os sólidos da região do defeito na direção circunferencial (Figura 3.41).

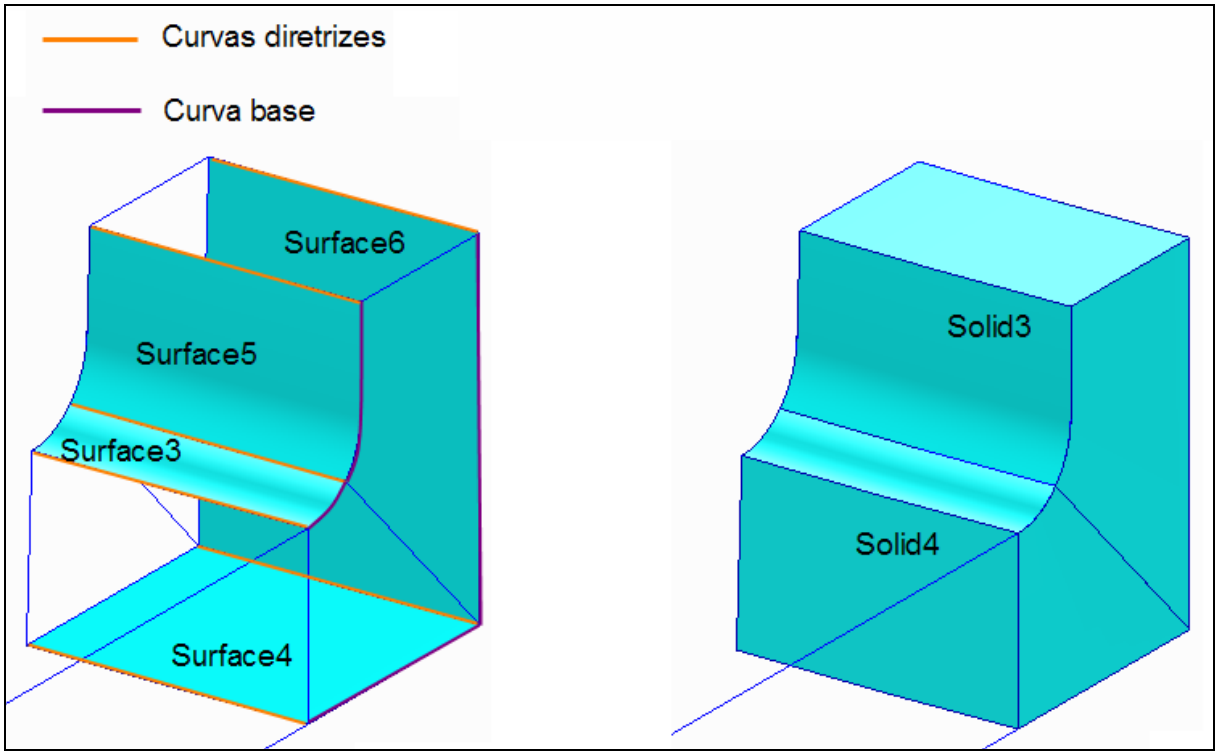


Figura 3.41 – Geração dos sólidos do defeito na direção circunferencial.

Mais uma vez é utilizado o método "*glide*" com suas respectivas curvas diretrizes e bases indicadas. No caso de sólidos orientados na direção circunferencial o método mais simples para geração seria via revolução de uma superfície base de maneira semelhante aos sólidos gerados nas direções longitudinais (via extrusão). No entanto, optou-se por gerar os sólidos por meio de duas superfícies (*sgm_const_solid_2surface_v1*), pois a geração da superfície base lateral apresentou problemas.

b) Geração dos sólidos centrais

O próximo passo é a geração dos sólidos centrais na região do raio de concordância (RC) conforme ilustra a Figura 3.42.

As curvas que definem o raio de concordância formam um arco de elipse na superfície do duto. Para gerar esses arcos de elipses, primeiro gera-se um arco de curva equivalente ao raio específico da região de concordância (com centro nos sistemas de coordenadas locais criados). Em seguida, faz-se um escalonamento destas curvas no espaço na direção circunferencial gerando assim um arco de elipse no espaço. Por último, esses arcos são projetados nas superfícies cilíndricas auxiliares formando assim o perfil da região de concordância do defeito.

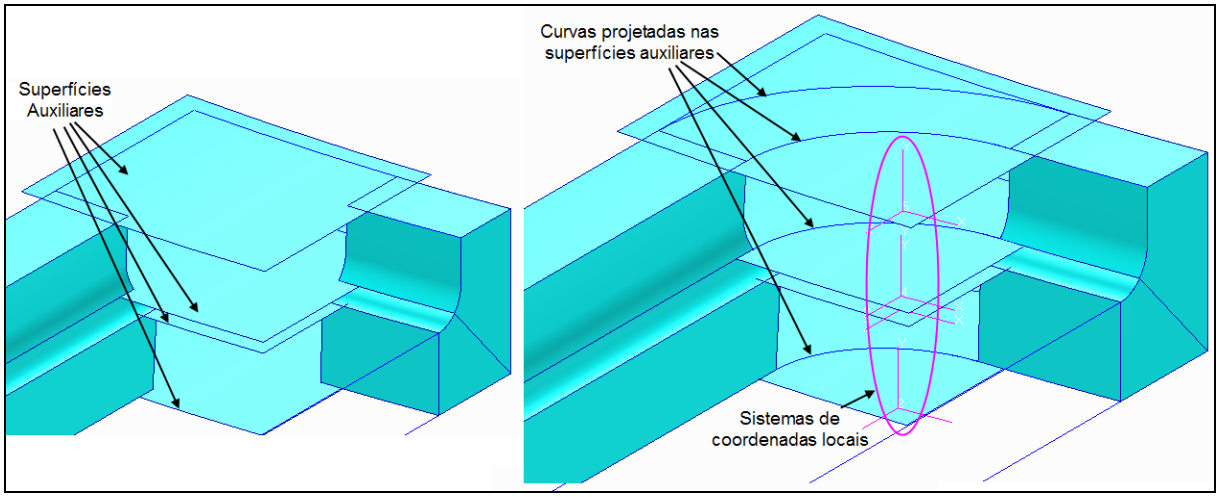


Figura 3.42 – Geração das curvas projetadas nas superfícies auxiliares.

As curvas projetadas são "quebradas" para depois serem utilizadas como curvas bases e diretrizes na geração das superfícies que irão compor os sólidos centrais na região do raio de concordância do defeito. A Figura 3.43 mostra as curvas diretrizes (em amarelo) e as curvas base (na cor roxa). As superfícies bi-paramétricas são então geradas novamente via o método "*glide*". As setas na cor vermelha indicam a direção em que a curva base é "varrida" ao longo do caminho definido por duas curvas diretrizes que definem a região de concordância.

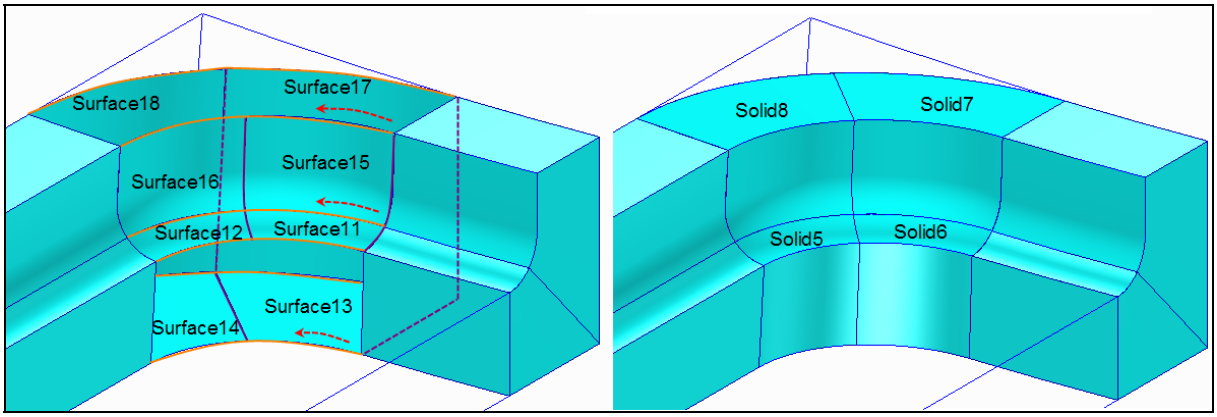


Figura 3.43 – Geração das superfícies e sólidos centrais na região de concordância.

c) Geração dos sólidos em torno do defeito

Os sólidos 9 e 10 (Figura 3.44) são gerados por meio de revolução e extrusão de suas respectivas superfícies bases (faces do sólido vizinho). Os restantes dos sólidos que compõem a região em torno do defeito (Figura 3.45) não possuem uma superfície base orientada longitudinalmente ou circunferencialmente como no caso dos sólidos 9 e 10 e por isso não puderam ser gerados via extrusão e revolução. Neste caso, os sólidos restantes (sólido 11, 12 e 13) foram gerados a partir de duas superfícies. Estas superfícies foram obtidas novamente via o método "glide" que utiliza as curvas bases e diretrizes indicadas na Figura 3.45. Vale lembrar que para a construção de algumas destas superfícies, foi necessário a criação de curvas projetadas nas superfícies cilíndricas auxiliares que definem a curvatura do duto naquela região de concordância.

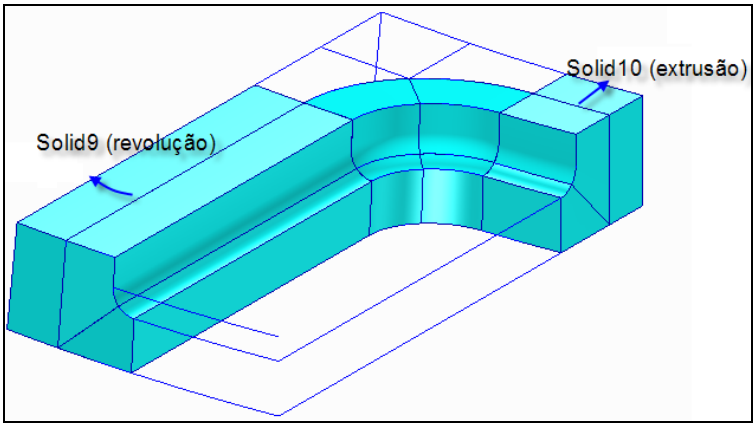


Figura 3.44 – Geração dos sólidos 9 e 10 via revolução e extrusão.

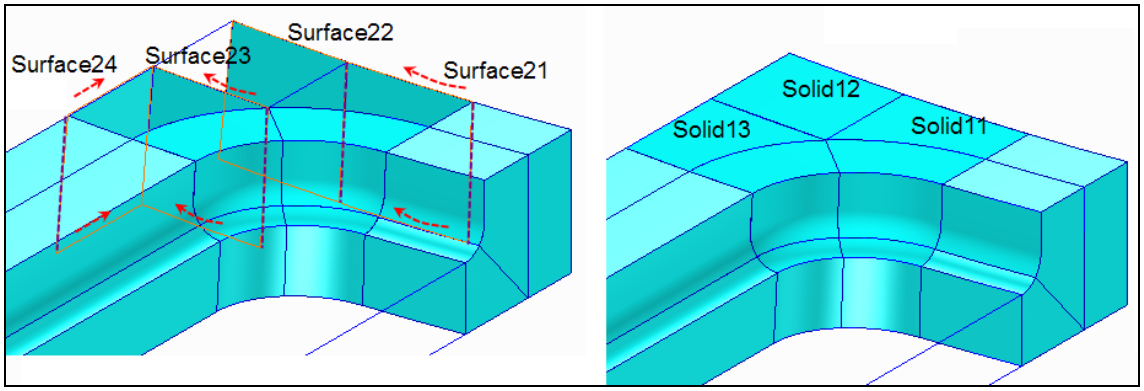


Figura 3.45 – Geração dos sólidos em torno do defeito via método de duas superfícies.

d) Geração dos sólidos retangulares

Esta é a última etapa de geração dos sólidos do defeito. Nesta etapa, são gerados os sólidos retangulares cujo perfil de profundidade do defeito é constante nesta região.

Primeiro são gerados os sólidos 14 e 15 (Figura 3.46). Estes sólidos se localizam na região equivalente à diferença entre o raio de concordância (RC) e o raio de adoçamento (RA).

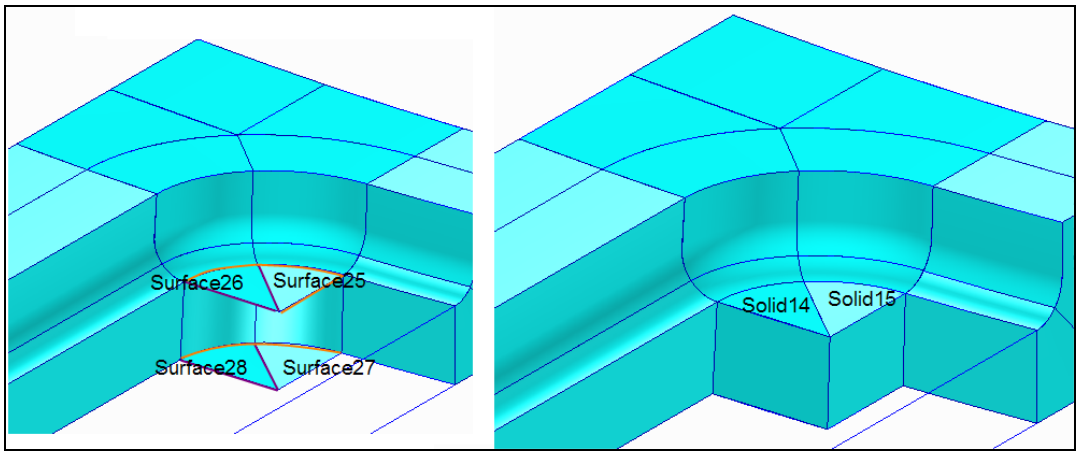


Figura 3.46 – Geração dos sólidos retangulares 14 e 15.

É válido lembrar que RC deve ser sempre maior ou igual ao valor de RA. À medida que o valor de RC se aproxima do valor de RA, os sólidos 14 e 15 vão diminuindo de tamanho. Numa situação limite, onde RC = RA, estes sólidos deixam de existir.

O programa PIPEFLAW não permite que o usuário forneça valores de RC muito próximos de RA e, portanto, não considera a geração de modelos próximos desta situação limite. Naturalmente, novas funcionalidades podem ser melhoradas no programa de forma a tornar possível a geração de modelos nestas situações.

A Figura 3.47 apresenta a geração dos sólidos 16, 17 e 18 cujas superfícies bases foram geradas por meio do método "*glide*".

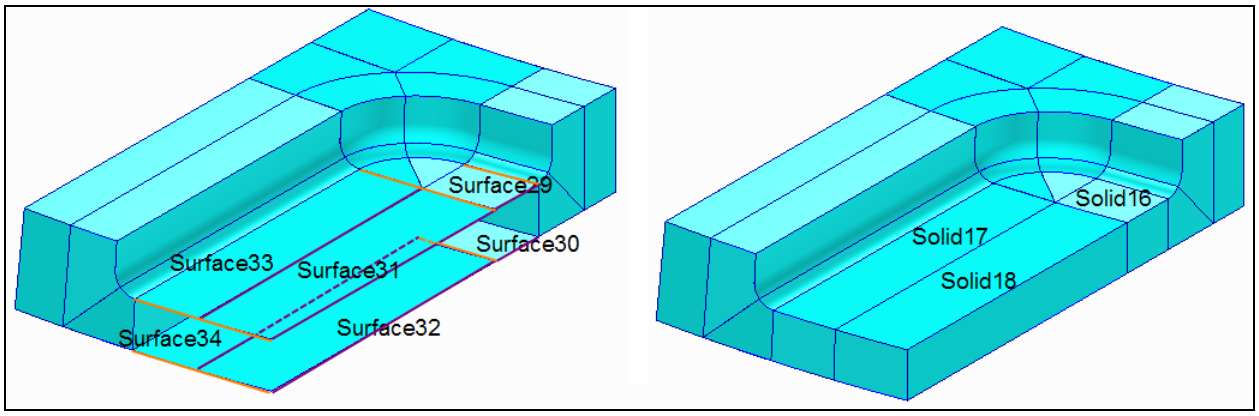


Figura 3.47 – Geração dos sólidos retangulares 16, 17 e 18.

e) Geração da malha

Após a geração automática da geometria do defeito, dá-se início ao procedimento automático de geração de malha. O programa calcula a espessura dos elementos na região do defeito dividindo o comprimento da espessura remanescente (T-D) por quatro (modelos de defeitos contêm sempre quatro elementos ao longo da espessura). O comprimento médio global dos elementos (*glb_length*) ao longo da superfície da região do defeito é então obtido multiplicando por 1,3 o comprimento dos elementos ao longo da espessura. Isto dá aos elementos uma forma parecida com a de um cubo de dimensões iguais (Figura 3.48).

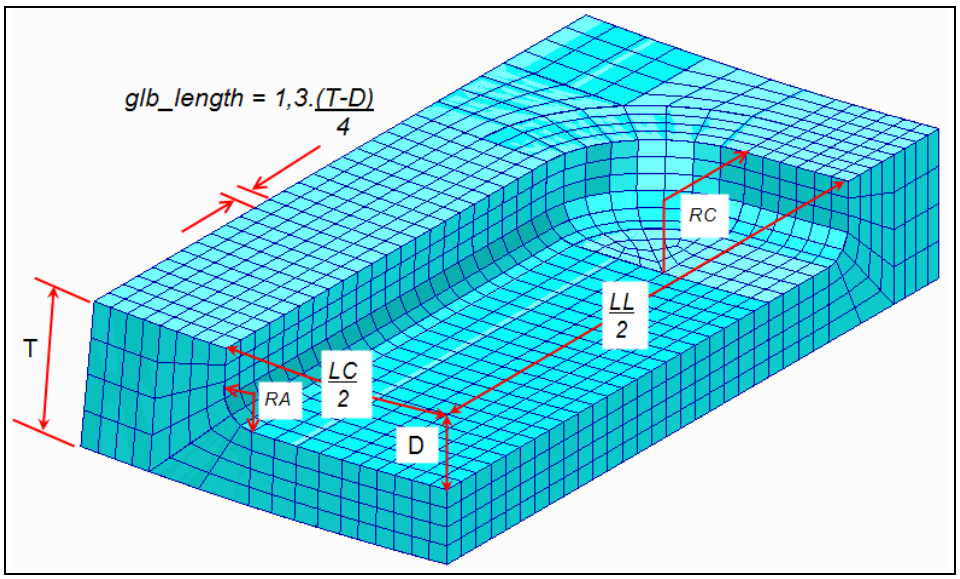


Figura 3.48 – Geração da malha na região do defeito.

A partir do cálculo do parâmetro *glb_length*, é feita uma estimativa do número de elementos no restante do modelo em função das medidas geométricas das arestas de cada sólido. Por exemplo, para estimar o número de elementos nos sólido longitudinais (*ne_long*), divide-se o valor do comprimento desses sólidos (*LL-RC*) pelo parâmetro *glb_length*. O algoritmo desenvolvido para geração automática de malha requer que o número de elementos ao longo das arestas dos sólidos seja par. Isto garante que apenas transições do tipo 2 para 1 sejam utilizadas na primeira transição na superfície. Assim, após fazer a estimativa do número de elementos, o programa verifica se este número é par, e, caso contrário, arredonda para o próximo número inteiro par superior.

f) Primeira Região de Expansão de Malha

A Figura 3.49 ilustra a geração da primeira região de expansão de malha logo após a região do defeito. O limite entre os sólidos estão destacados por linhas na cor azul escuro na superfície. Percebe-se que não há transições de malha nesta região. Há apenas um pequeno alongamento das dimensões dos elementos ao longo da superfície equivalente ao fator multiplicativo igual a 1,7, ao invés de 1,3 como feito anteriormente.

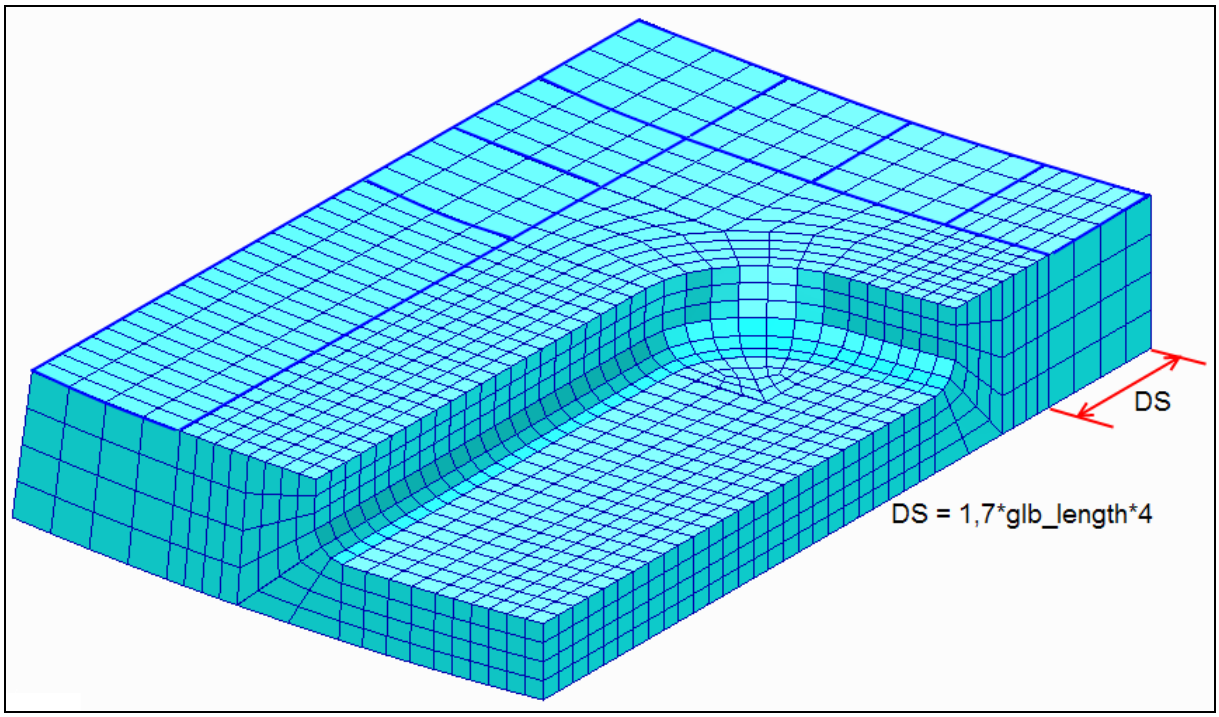


Figura 3.49 – Modelagem da primeira região de expansão

g) Transição na Espessura

A Figura 3.50 mostra o detalhe da transição de quatro para dois elementos ao longo da espessura com as linhas delimitando os sólidos desta região. A realização desta transição é feita basicamente por meio de "*mesh seeds*". No entanto, a maior dificuldade nesta etapa foi gerar a malha no sólido central desta região, pois as funções disponíveis no PATRAN não permitem gerar este tipo de transição no sólido central. Neste caso, teve-se que implementar uma função que cria automaticamente os elementos a partir dos vértices do sólido central. Em seguida, esses elementos recém criados são associados à geometria do sólido central, completando assim a geração da transição ao longo da espessura.

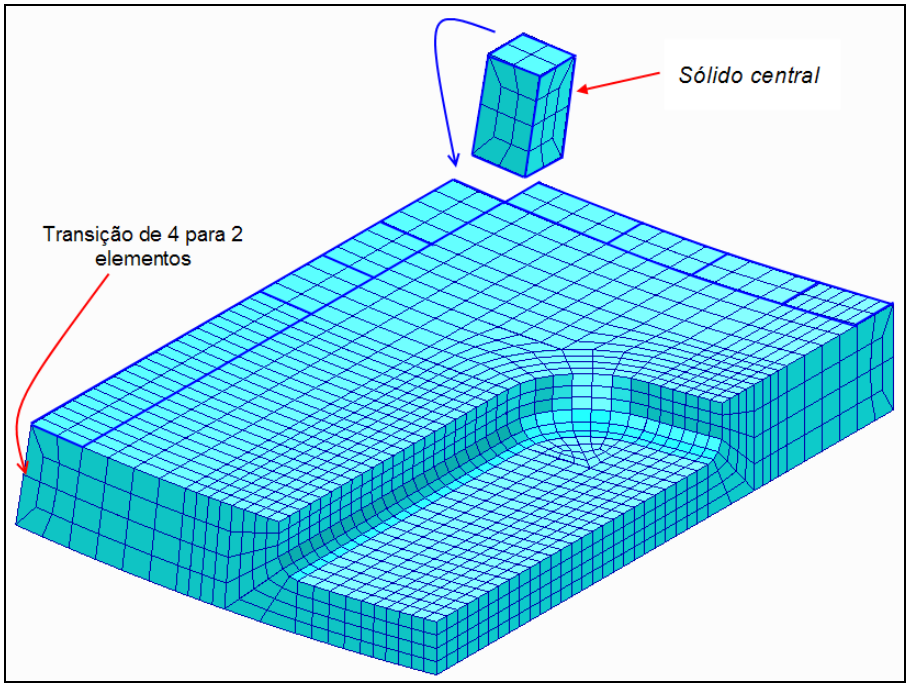


Figura 3.50 – Transição ao longo da espessura.

h) Primeira Transição na Superfície

Uma das maiores dificuldades na implementação das funções de transição ao longo da superfície foi gerar modelos de acordo com o padrão utilizado pelo CENPES. Num primeiro trabalho (Cabral et al, 2006a), as transições ao longo da superfícies foram feitas simplesmente por meio de "*mesh seeds*" num único sólido. No entanto, o aspecto da malha torna-se bastante irregular à medida que as dimensões dos defeitos aumentam. A Figura 3.51 mostra região de transição ao longo da superfície realizada num único sólido destacado na cor azul.

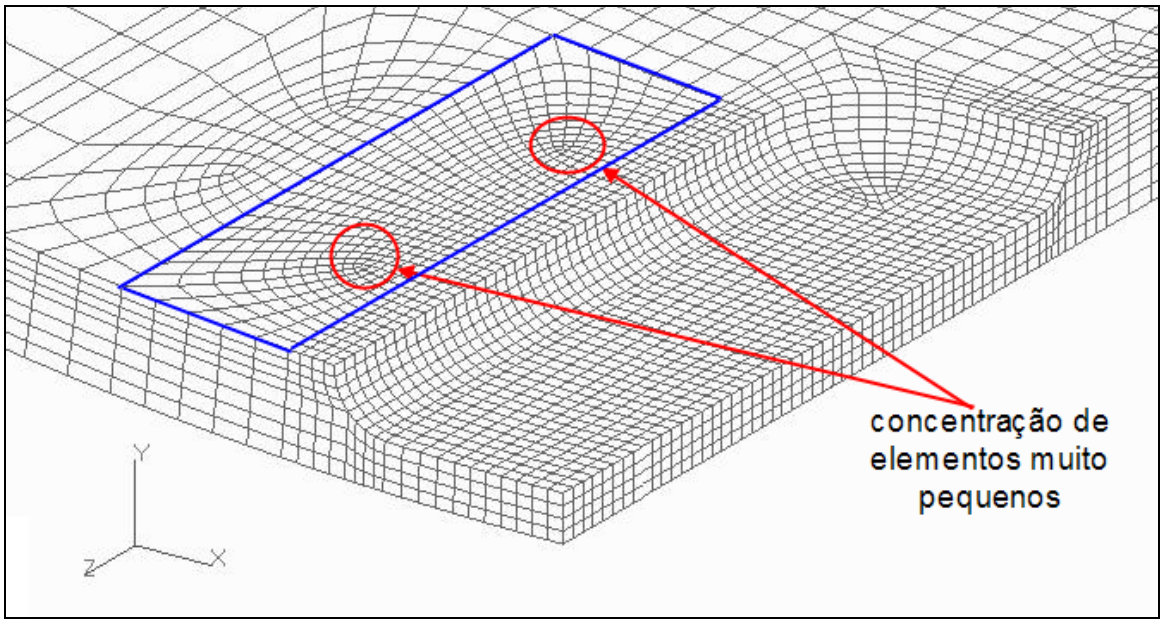


Figura 3.51 – Modo alternativo para geração da transição de malha na superfície.

Pode-se observar a existência de elementos muito pequenos, formados pelo gerador de malhas do PATRAN, em algumas regiões do sólido e outros elementos maiores nas extremidades. Isto ocorre, pois a largura do sólido na direção circunferencial, utilizada para realizar a transição, é menor que o comprimento na direção longitudinal. Quanto maior for a diferença entre a largura e o comprimento do sólido, mais irregular se torna a distribuição de elementos no sólido deixando a malha exterior ao defeito bem mais refinada do que na região do defeito propriamente dito, o que é extremamente desnecessário e só faz aumentar o custo computacional durante a análise.

A geração de transição na superfície utilizando o padrão do CENPES implicou numa maior dificuldade de implementação computacional. Ao invés de gerar um único sólido a partir da superfície do sólido vizinho (via extrusão ou revolução), são gerados vários sólidos menores (com comprimento equivalente a dois elementos) cujas malhas são geradas individualmente. A principal função implementada para executar a primeira transição na superfície é a função *DivideSolids2()*. A Figura 3.52 mostra exemplo do funcionamento desta função. A partir de um "*loop for*", para cada sólido vizinho da região de transição na espessura (linha na cor preta) são gerados novos sólidos menores (cor azul) cujas malhas são geradas individualmente. Esta primeira transição na superfície sempre ocorrerá utilizando transições do tipo de 2 elementos para 1 conforme descrito no item "e" desta seção.

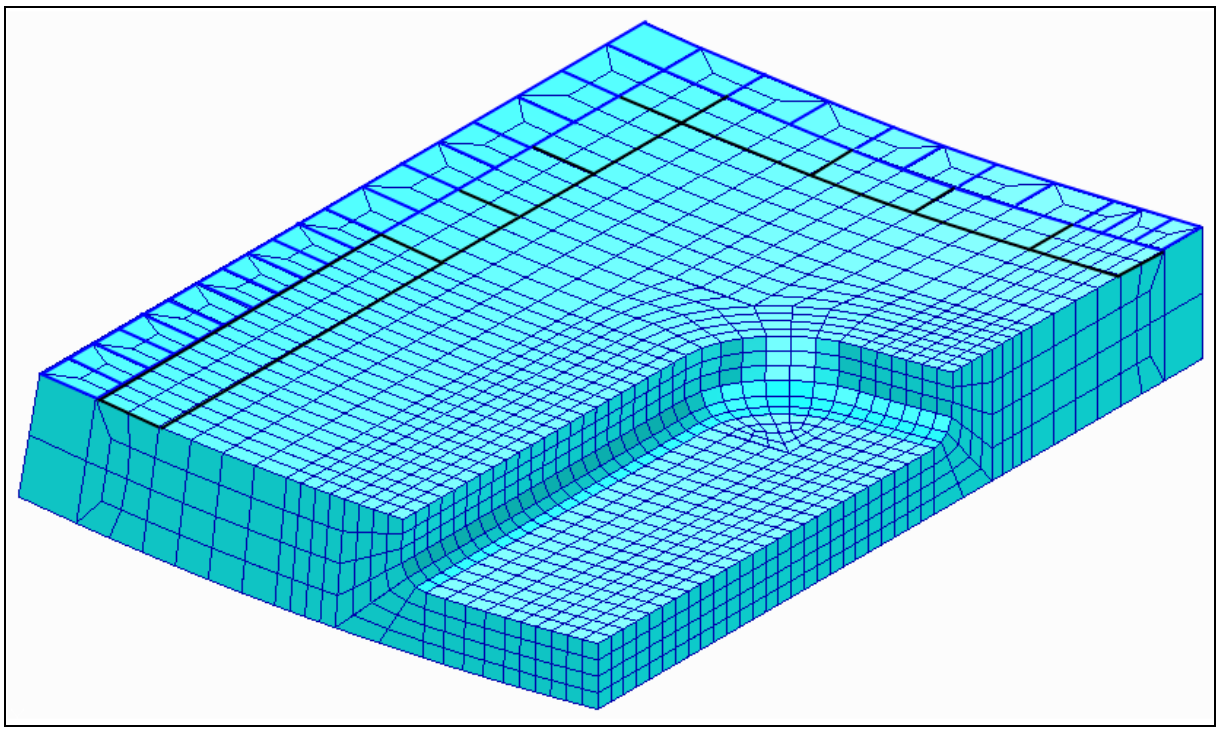


Figura 3.52 – Transição de malha ao longo da superfície utilizando o padrão adotado pelo CENPES.

As malhas são geradas na direção longitudinal e circunferencial iniciando sempre pelo sólido central, conforme indicado na Figura 3.53. A malha é gerada individualmente nos pequenos sólidos (sólidos na cor azul) a cada iteração. Estes pequenos sólidos são gerados a partir do seu respectivo sólido "pai" pertencente à região imediatamente anterior (região de transição ao longo da espessura) (ver sólidos na cor preta na Figura 3.52 e Figura 3.53).

O procedimento automático de geração de malha nestes sólidos requer um controle do número de elementos ao longo das arestas laterais uma vez que os novos sólidos gerados possuem sempre um elemento nas arestas frontais e dois elementos nas arestas traseiras (ver definição das arestas na Figura 3.53). A exceção é o sólido central que sempre terá apenas um

elemento ao longo de suas arestas na superfície. Isto garante que a próxima seqüência de malha (que se inicia com o sólido 1) tenha sempre um elemento na lateral esquerda (caso a 1ª seqüência seja na direção longitudinal) ou tenha sempre um elemento na lateral direita (caso a 1ª seqüência seja na direção circunferencial).

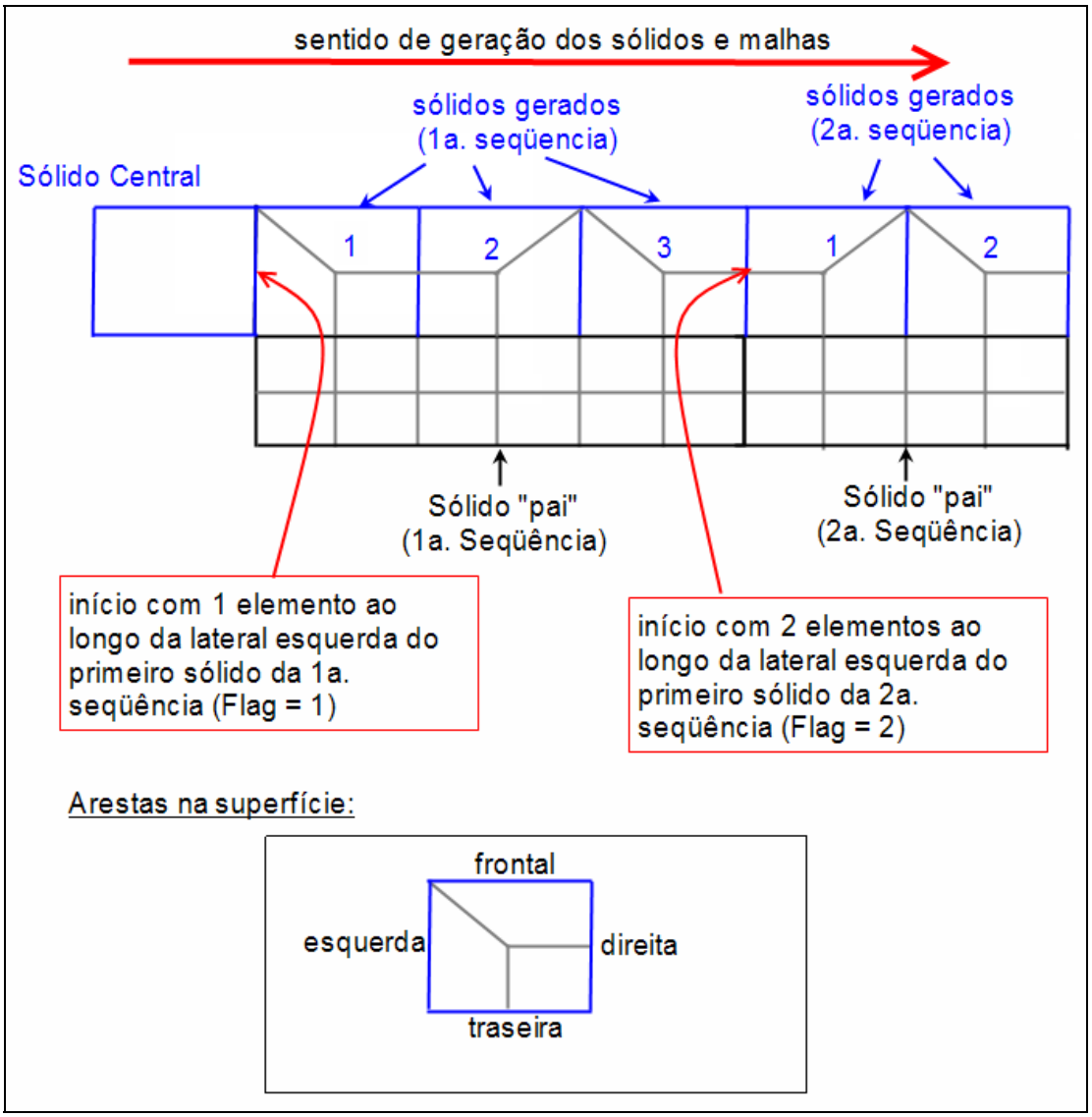


Figura 3.53 – Ilustração do procedimento automático de geração da primeira transição de malha ao longo da superfície.

Observe que dentro de cada seqüência de sólidos gerados (Figura 3.53) o número de elementos ao longo das arestas laterais (esquerda e direita) só pode assumir o valor 1 ou 2. No exemplo acima, os sólidos ímpares da 1ª seqüência (sólidos 1 e 3) têm sempre um elemento na lateral esquerda e dois elementos na lateral direita, enquanto que os sólidos pares (sólido 2) é o inverso (dois elementos na lateral esquerda e um elemento na lateral direita).

Cada iteração do "*loop for*" dentro da função *DivideSolids2*() equivale à geração de uma seqüência de sólidos com suas respectivas malhas. Dentro de cada seqüência de sólidos, o programa verifica se o sólido é ímpar ou par para em seguida gerar os "*mesh seeds*" nas arestas laterais (esquerda e direita). Este procedimento é feito até que todos os sólidos daquela seqüência sejam malhados. A próxima iteração do "*loop for*" irá repetir esse mesmo procedimento de forma semelhante para a próxima seqüência de sólidos, respeitando apenas os números de elementos contidos na lateral direita da última seqüência de sólidos que foi gerada.

No exemplo da Figura 3.53, a malha do primeiro sólido da 1ª seqüência continha um elemento ao longo da aresta lateral esquerda enquanto que a malha do primeiro sólido da 2ª seqüência contém dois elementos ao longo da aresta lateral esquerda. O programa PIPEFLAW, baseado no número de sólidos que foram gerados, salva automaticamente estas informações à medida que as seqüências de sólidos e malhas são geradas. Isto é feito por meio de "Flags" que indicam se a próxima seqüência a ser gerada deve iniciar com um (*Flag = 1*) ou dois (*Flag = 2*) elementos ao longo das arestas laterais.

i) Segunda Região de Expansão

Depois de gerada a primeira transição na superfície, segue-se então a geração da segunda região de expansão de malha que contém sempre seis elementos ao longo do comprimento dos sólidos destacados na cor azul, conforme Figura 3.54. Os elementos nesta região possuem um comprimento médio igual a três vezes o comprimento dos elementos na região do defeito.

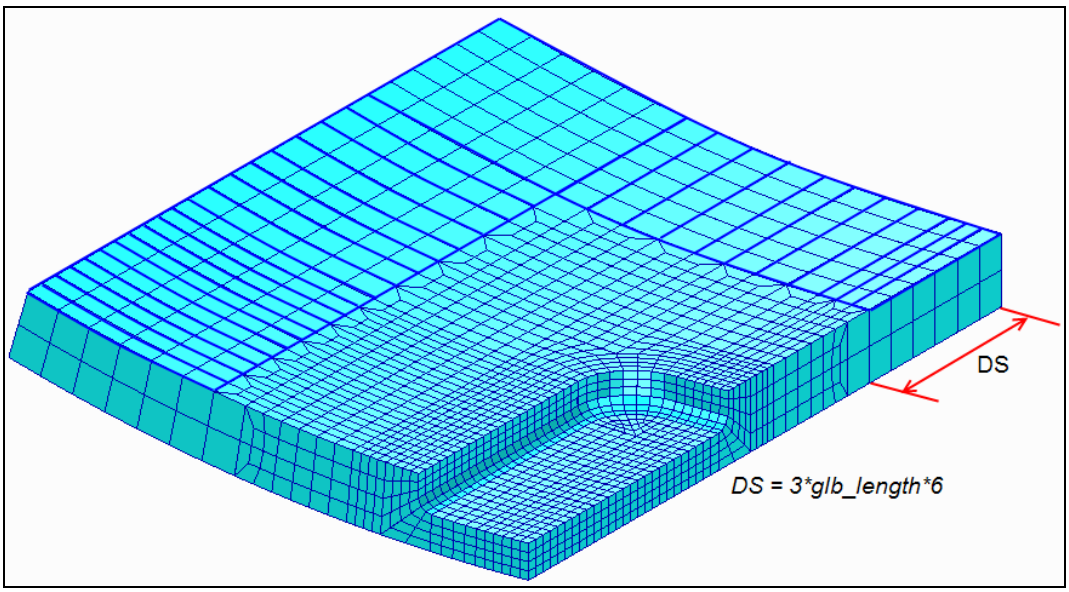


Figura 3.54 – Geração da segunda região de expansão de malha.

j) Segunda e Terceira Transições na Superfície

As principais funções implementadas para a geração da segunda transição na superfície (Figura 3.55) e terceira transição na superfície (Figura 3.56) são as funções *JoinSolids2()* e *JoinSolids3()*, respectivamente. A idéia destas funções é justamente o contrário da função *DivideSolids2()*. Ou seja, para cada dois ou três sólidos vizinhos pertencentes à região imediatamente anterior (sólidos na cor preta) é gerado um novo sólido (cor azul) e malha usando as transições de 2 para 1 e de 3 para 1, dando preferência sempre ao tipo de transição de 2 para 1.

A exceção da segunda transição na superfície é o sólido central da segunda região de expansão (Figura 3.55) que é subdividido da mesma maneira que os sólidos da transição na espessura durante a realização da primeira transição na superfície. Essa operação de subdivisão do sólido é feita utilizando a mesma função *DivideSolids2()* citada anteriormente.

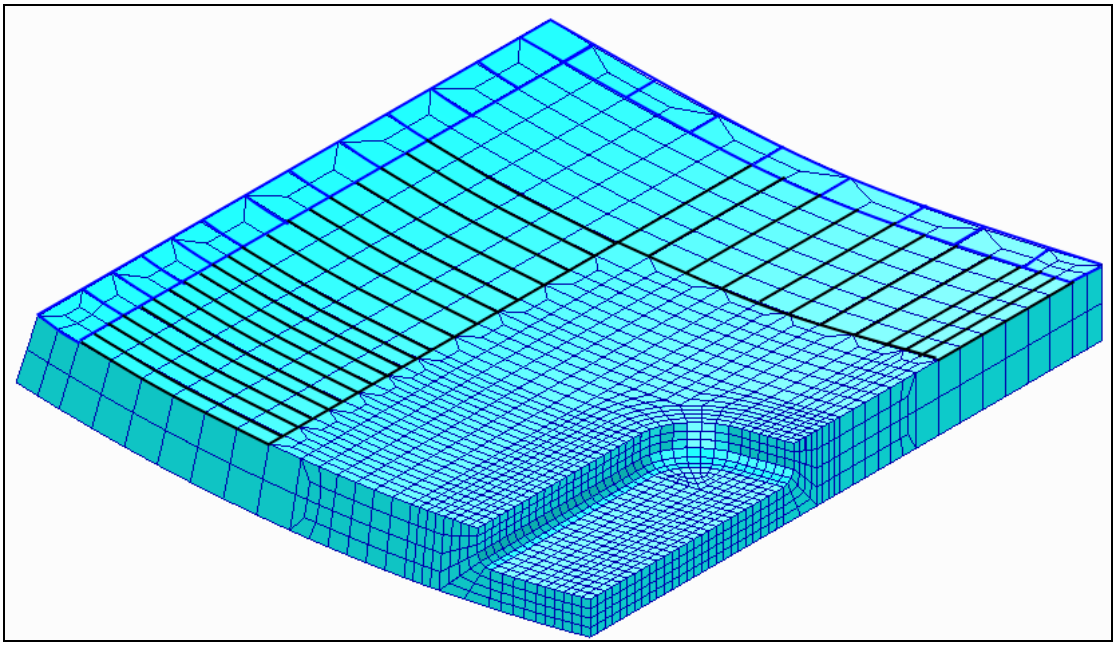


Figura 3.55 – Segunda transição na superfície.

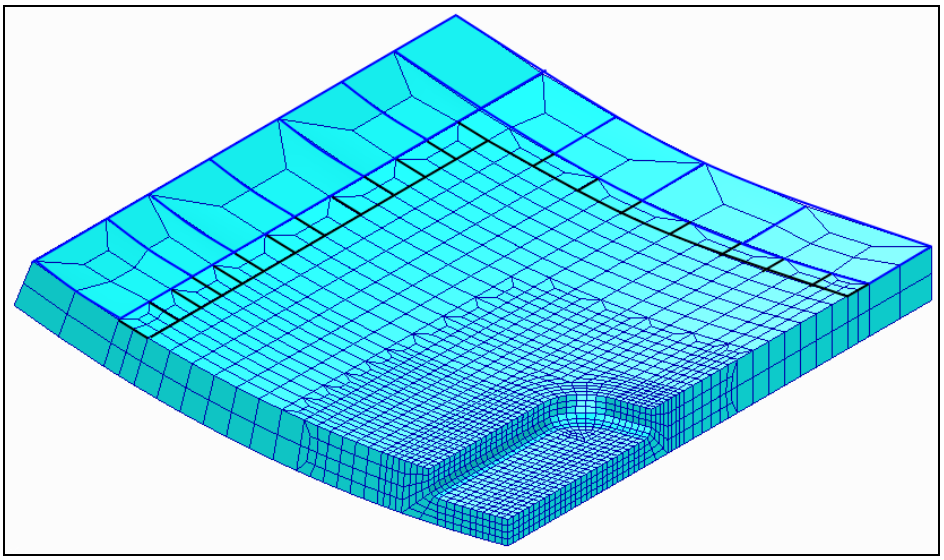


Figura 3.56 – Terceira transição na superfície.

k) Geração de defeito simples

A terceira transição na superfície é a última transição de malha realizada pelo programa e a densidade de elementos desta última transição será a utilizada no restante dos sólidos distantes do defeito. Para finalizar a geração do modelo de duto com defeito simples, são gerados apenas os sólidos e malha complementares de ¼ do duto via revolução e extrusão (Figura 3.57).

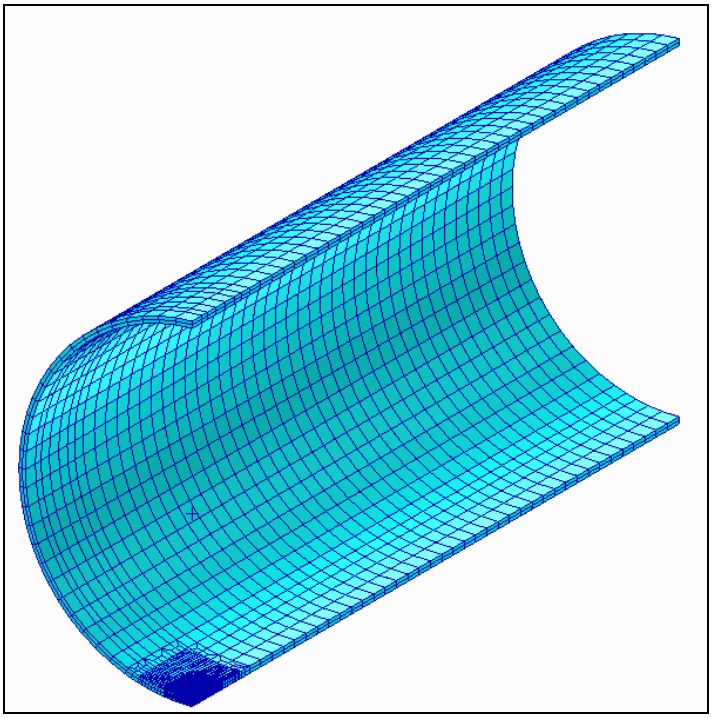


Figura 3.57 – Geração dos sólidos complementares de ¼ do duto via extrusão e revolução.

l) Geração de múltiplos defeitos alinhados

A geração de múltiplos defeitos alinhados é executada a partir dos grupos (sólido e malha) de cada região já gerada anteriormente. Assim, em função da localização de cada defeito fornecida pelo usuário via interface gráfica, os grupos são então transladados (múltiplos defeitos alinhados longitudinalmente) ou rotacionados (múltiplos defeitos alinhados circunferencialmente) para as suas devidas posições. Isto significa que as geometrias e malhas para cada região do defeito e adjacências são simplesmente uma cópia dos grupos que definem estas regiões. As únicas regiões onde são geradas novas geometrias e malhas são as regiões remanescentes entre defeitos adjacentes conforme limites de transição definidos na seção 3.5.4. A Figura 3.58 apresenta detalhe de um modelo com dois defeitos retangulares alinhados longitudinalmente situados na superfície externa do duto. A Figura 3.59 mostra detalhe de um modelo com três defeitos alinhados circunferencialmente localizados na superfície interna do duto.

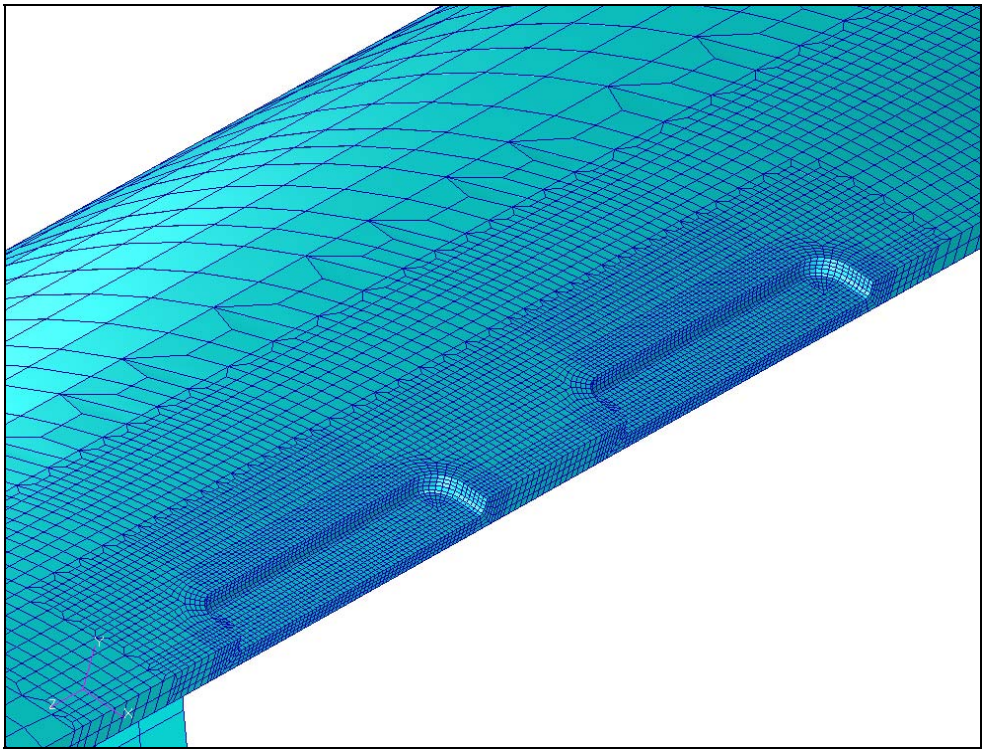


Figura 3.58 – Exemplo de múltiplos defeitos externos alinhados longitudinalmente.

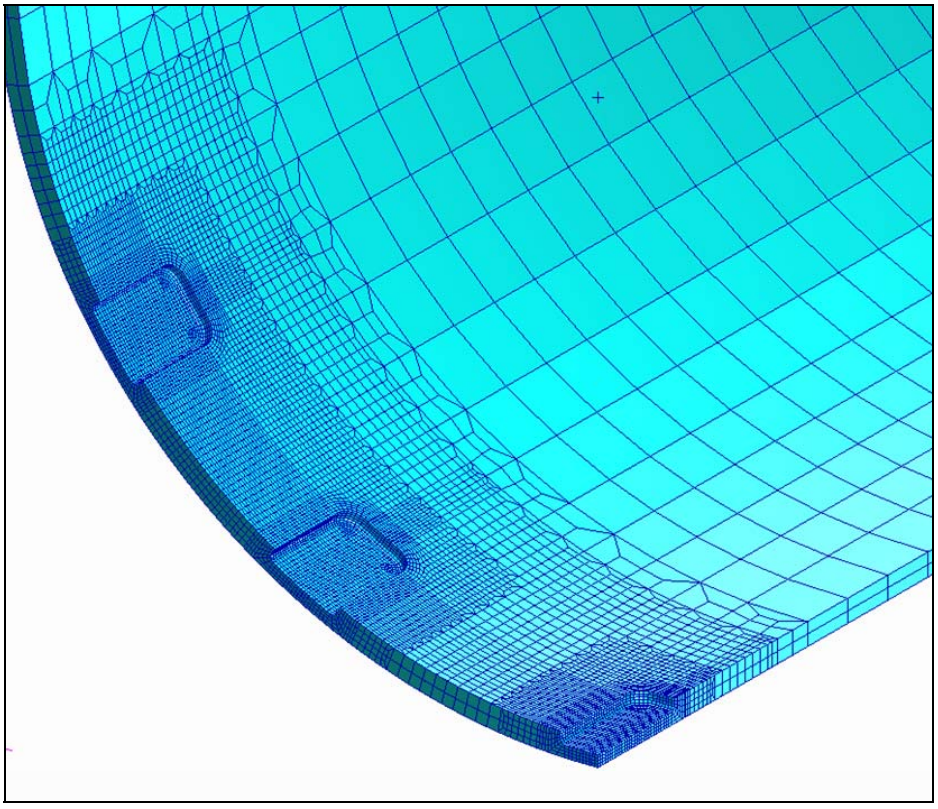


Figura 3.59 – Exemplo de modelo com múltiplos defeitos internos alinhados circunferencialmente.

3.5 Modelo de Elementos Finitos

3.5.1 Introdução

A maioria dos problemas de engenharia são governados por sistemas de equações diferenciais (ordinárias ou parciais) que descrevem matematicamente o comportamento físico de um determinado fenômeno. Estas equações são válidas em uma certa região (ou domínio) onde são impostas condições de contorno ou iniciais apropriadas formando assim o modelo matemático para o problema físico.

No entanto, na prática, a solução analítica da maioria desses problemas de engenharia são desconhecidas ou de difícil obtenção. Neste contexto, os métodos numéricos de resolução de equações diferenciais têm sido utilizados amplamente com o objetivo de obter soluções aproximadas para o problema físico.

O método dos elementos finitos (MEF) é um dos métodos numéricos mais utilizados na modelagem do comportamento de estruturas, principalmente devido a sua grande confiabilidade e flexibilidade. A idéia básica do MEF consiste em discretizar o domínio físico (geometria) por meio de uma quantidade finita de elementos com forma e tamanho arbitrários, formando assim o domínio computacional. O domínio discretizado, denominado malha de elementos finitos, consiste em um conjunto de regiões, ou elementos, interconectados entre si através de um número discreto de nós (pontos nodais ou conectividade do elemento) resultando assim num sistema de equações lineares ou não-lineares do tipo [K].[u] = [f], onde [K] é a matriz de rigidez, [u] é o vetor de deslocamentos nodais e [f] é o vetor de forças nodais. Esses sistemas podem ser resolvidos numericamente por meio de métodos de otimização e de algoritmos matriciais onde as incógnitas são os deslocamentos nodais que são aproximados por funções contínuas (chamadas de funções de forma ou funções de interpolação) expressas em termos de variáveis nodais.

Embora o MEF seja uma ferramenta bastante poderosa e versátil na simulação de problemas de engenharia, a geração de modelos inadequados pode resultar em péssimas interpretações da análise. A geração de bons modelos implica na geração de malhas com boa qualidade (reduzindo os erros de aproximação), escolha adequada das propriedades do material e aplicação de adequadas condições iniciais e de contorno. As próximas seções irão tratar com mais detalhe sobre esses assuntos.

3.5.2 Elemento Finito Utilizado

A maior parte dos trabalhos envolvendo análise por elementos finitos de defeitos de corrosão em dutos tem sido realizada utilizando elementos finitos sólidos. Os elementos a serem adotados podem diferir em diversos aspectos, mas para a simulação numérica, os itens mais significantes são a forma do elemento e sua ordem de interpolação, que se refere ao grau do polinômio que aparece nas funções de forma.

Vários trabalhos foram publicados onde os pesquisadores investigaram o desempenho dos elementos de casca tridimensional em comparação com os elementos sólidos (Noronha Jr et al, 2002; Benjamin et al, 2002; Benjamin & Andrade, 2003b). Os resultados mostrados nestes trabalhos indicaram que ambos os tipos de elementos foram capazes de representar bem os defeitos analisados. Os modelos de elementos finitos de casca obtiveram resultados mais conservadores que os modelos de elementos finitos sólidos, este último fornecendo portanto resultados mais precisos.

No entanto, quando se trata de modelagem de múltiplos defeitos ou defeitos de geometria complexa a tendência é utilizar elementos sólidos, pois com estes tipos de elementos é possível obter uma melhor idéia do perfil de tensões e deformações ao longo da espessura remanescente do duto e capturar melhor o efeito de interação entre defeitos (Chouchaoui & Pick, 1994; Chouchaoui & Pick, 1996; Cronin, 2002).

Uma vez que o procedimento automático de geração da geometria do programa PIPE-FLAW é baseado na criação de sólidos para representação do perfil do defeito, então a possibilidade de utilização de elementos tridimensionais de casca foi descartada. Além disso, o projeto original desta pesquisa previu também a geração de malhas estruturadas por blocos compostas por elementos sólidos hexaédricos. Dentro desta gama de opções, o PATRAN dispõe de uma família de elementos hexaédricos. Os dois tipos de elementos hexaédricos mais utilizados são: Hex8 e Hex20 (Figura 3.60). As principais características destes elementos estão mostradas na Tabela 3.6.

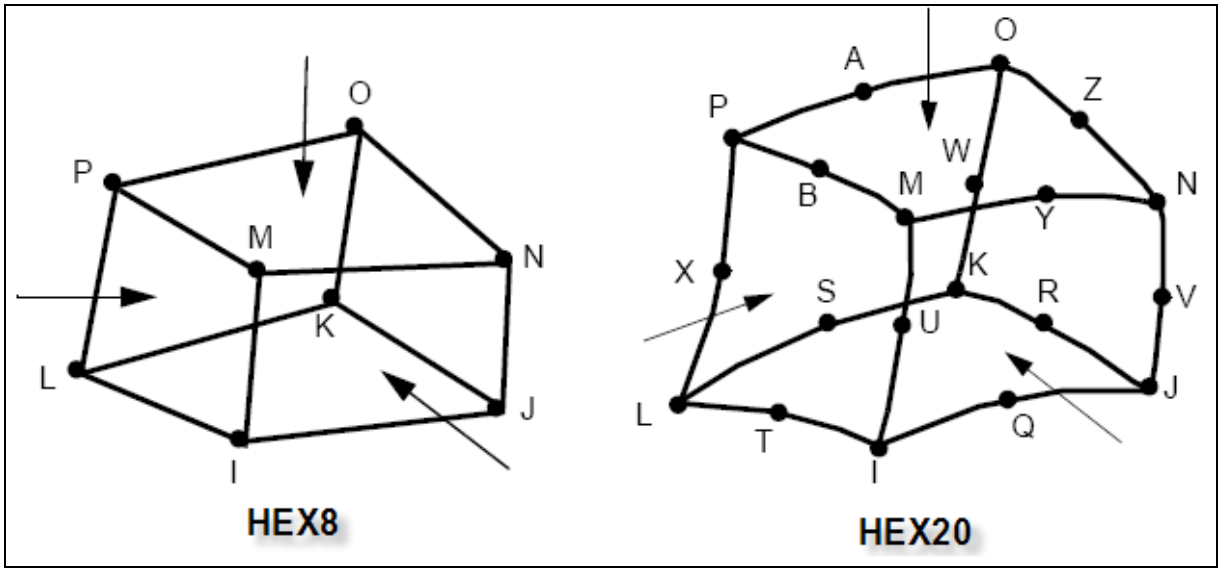


Figura 3.60 – Elementos finitos hexaédricos disponíveis no PATRAN.

Tabela 3.6 – Principais características dos elementos Hex8 e Hex20 do PATRAN.

	HEX8	HEX20
Tipo	3D-Isoparamétrico	3D-Isoparamétrico
Número de nós	8	20
Graus de liberdade por nó	3(Ux, Uy, Uz)	3(Ux, Uy, Uz)
Função de forma (interpolação)	tri-linear (1ª.ordem)	tri-quadrática (2ª.ordem)
Referência MSC.MARC	Element 7	Element 21
Referência ANSYS	Solid45	Solid95
Referência MSC.NASTRAN	CHEXA	CHEXA

Como o elemento Hex8 usa funções de interpolação tri-lineares, então as deformações tendem a serem constantes ao longo do elemento. Isto pode resultar em uma pobre representação do comportamento de cisalhamento que pode ser melhorado utilizando funções de forma de ordem mais alta. Os três "solvers" acima mencionados (MARC, ANSYS e NASTRAN) utilizam uma malha de oito (2x2x2) pontos de integração de Gauss para este elemento.

O elemento Hex20 usa funções de interpolação tri-quadráticas para representar as coordenadas e os deslocamentos; assim, as deformações têm um comportamento linear dentro do elemento o que permite uma representação mais precisa do campo de deformações. Os "solvers" MARC e NASTRAN utilizam uma malha de 27 pontos (3x3x3) de integração para este elemento enquanto que o ANSYS utiliza uma malha de 14 pontos ou 8 pontos (2x2x2).

Normalmente as tensões e deformações são calculadas nos pontos de Gauss e extrapoladas para os pontos nodais usando as funções de forma.

Em geral, necessita-se de uma maior quantidade de elementos de ordem mais baixa (Hex8) do que os de ordem mais alta (Hex20). Trabalhos como os de Diniz (2002) e Costa,(2004) mostraram que o uso de elementos Hex8 produzem resultados bastante satisfatórios.

Uma alternativa, adotada neste trabalho, foi utilizar uma malha mais refinada de elementos hexaédricos (8 nós) na região do defeito e à medida que vai se distanciando do mesmo, diminui-se a densidade de elementos para economizar tempo computacional. Esta é uma metodologia já bastante utilizada pelo CENPES/PETROBRÁS e que aqui foi adotada conforme será mostrado na seção seguinte. É válido lembrar que o programa PIPEFLAW permite que o usuário escolha o tipo de elemento hexaédrico (Hex8 ou Hex20) que será utilizado no modelo.

3.5.3 Discretização Utilizada

A discretização utilizada para representar a geometria de um determinado problema deve ser bem dimensionada para que represente de forma satisfatória e fiel o comportamento real da estrutura diante das condições de contorno aplicadas. Em geral, o uso de uma malha de elementos finitos mais grosseira reduz o custo computacional e facilita o processo de modelagem, mas conduz a resultados menos precisos e de menor magnitude. Conseqüentemente, o uso de uma malha grosseira nas análises de defeitos de corrosão irá causar possivelmente resultados incorretos na estimativa da pressão de ruptura dos dutos corroídos. Por outro lado, o uso de uma malha extremamente refinada pode ser desnecessário para determinados tipos de problemas onde os resultados serão tão precisos quanto se fosse utilizada uma malha menos refinada. Além disso, todo esse esforço desnecessário causaria um enorme custo de tempo computacional durante uma análise não-linear. Assim, é aconselhável realizar análises de sensibilidade e estudos de convergência para demonstrar que a discretização utilizada é adequada ao modelo a ser analisado.

A discretização utilizada no programa PIPEFLAW é baseada no procedimento padrão utilizado pelo CENPES/PETROBRÁS que realizaram vários estudos de convergência e refinamento de malha. Este procedimento consiste na utilização de uma malha refinada na região do defeito e de malhas intermediárias menos refinadas nas regiões mais distantes do defeito. A transição entre malhas de níveis de refinamento diferentes é feita de forma gradativa e suave e o número de elementos na região do defeito é sempre igual a quatro elementos.

Os defeitos aqui modelados possuem dois planos de simetria: plano longitudinal (YZ) e transversal (XY). A Figura 3.61 apresenta exemplo de discretização da geometria de ¼ de um defeito retangular interno já considerando os dois planos de simetria. Conforme pode ser visto, todo defeito gerado pelo programa possui quatro elementos ao longo da espessura remanescente do duto. Em geral, utiliza-se de dois a quatro elementos nesta região.

À medida que as regiões do defeito são geradas, são criados também os respectivos grupos que contém as entidades geométricas e de malha de cada região. Estes grupos são utilizados na geração de múltiplos defeitos alinhados. Cada grupo recebe um nome específico no momento da sua criação, pois é a partir deste nome que as operações de transformação poderão ser feitas em um determinado grupo. O grupo correspondente à região de ¼ do defeito é denominado "Aux_Box_Defect" (Figura 3.61). Este grupo corresponde à região na qual o defeito está contido e se estende até uma distância equivalente a duas vezes a espessura remanescente do duto (*T-D*), definindo assim o comprimento da "caixa" fixa de cada defeito ("*Box_Defect*"). Este é o limite de interação entre defeitos. Isso significa que a distância mínima entre centros de defeitos adjacentes equivale a duas vezes a distância "*Box_Defect"* (em metros) ou duas vezes o ângulo "*Ang_Box_Defect"* (em graus, para o caso de defeitos alinhados circunferencialmente).

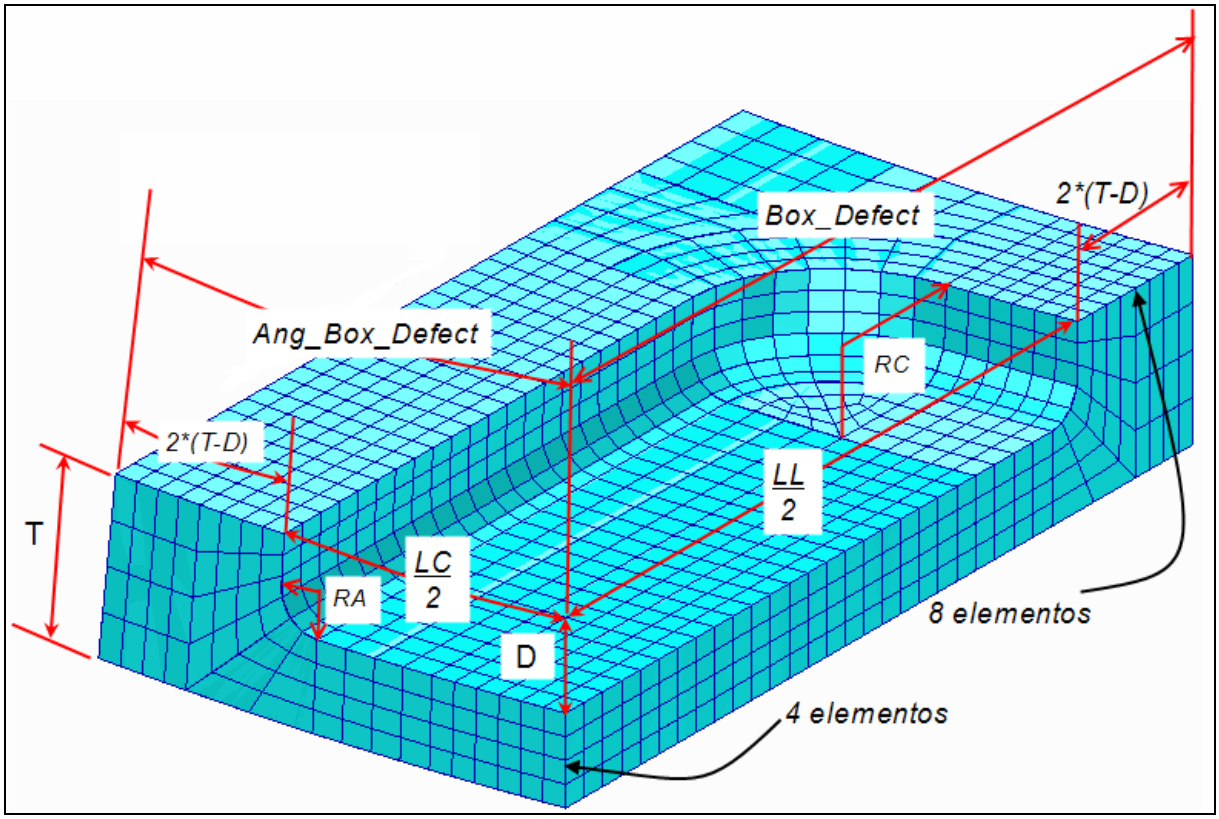


Figura 3.61 – Exemplo de discretização de um defeito retangular interno.

A discretização completa do modelo inclui a região do defeito e seis regiões adjacentes para transição de malha e que estão indicadas na Figura 3.62 e Figura 3.63. Dentre as seis regiões tem-se: uma região de transição ao longo da espessura; três regiões de transição ao longo da superfície e duas regiões de expansão de malha (sem transições). O nível de refinamento das regiões de transição vai diminuindo à medida que vão se distanciando da região do defeito.

Logo após a região do defeito segue-se a primeira região de expansão de malha (Figura 3.62). Esta região não apresenta transições de malha em nenhuma direção servindo apenas para aumentar a região adjacente ao defeito que possui quatro elementos ao longo da espessura. Desta forma, o nível de refinamento desta região é o mesmo da região do defeito o que permite capturar melhor os elevados gradientes de tensão que surgem nestas regiões.

Após a primeira região de expansão de malha segue-se a única transição de malha ao longo da espessura. Conforme pode ser visto na Figura 3.62, esta transição de malha ocorre de quatro elementos para dois elementos. Com esta transição de malha, todas as outras regiões a partir daí terão apenas dois elementos ao longo da espessura.

Imediatamente após a transição na espessura, segue-se a primeira transição de malha ao longo da superfície. Esta transição de malha ocorre sempre de dois elementos para um e nunca de três elementos para um conforme procedimento descrito na seção 3.4.

Em seguida, é gerada a malha na segunda região de expansão de malha que, da mesma forma que a primeira, não possui transições de malha em nenhuma direção (Figura 3.63).

Finalmente são geradas a segunda e terceira transições de malha ao longo da superfície (Figura 3.63). Estas regiões utilizam o maior número possível de combinações de transição do tipo dois elementos para um. Caso seja necessário, são utilizadas transições do tipo de três elementos para um (Figura 3.64).

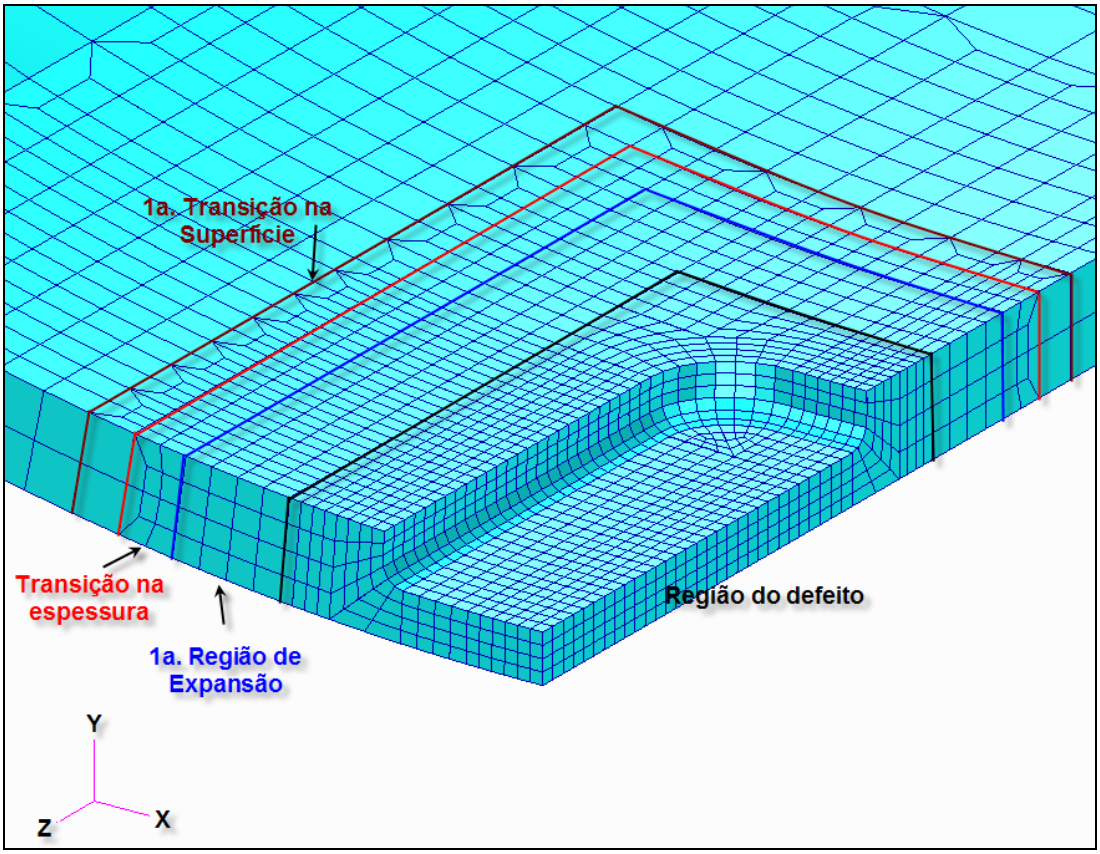


Figura 3.62 – Transição de malha nas regiões próximas ao defeito.

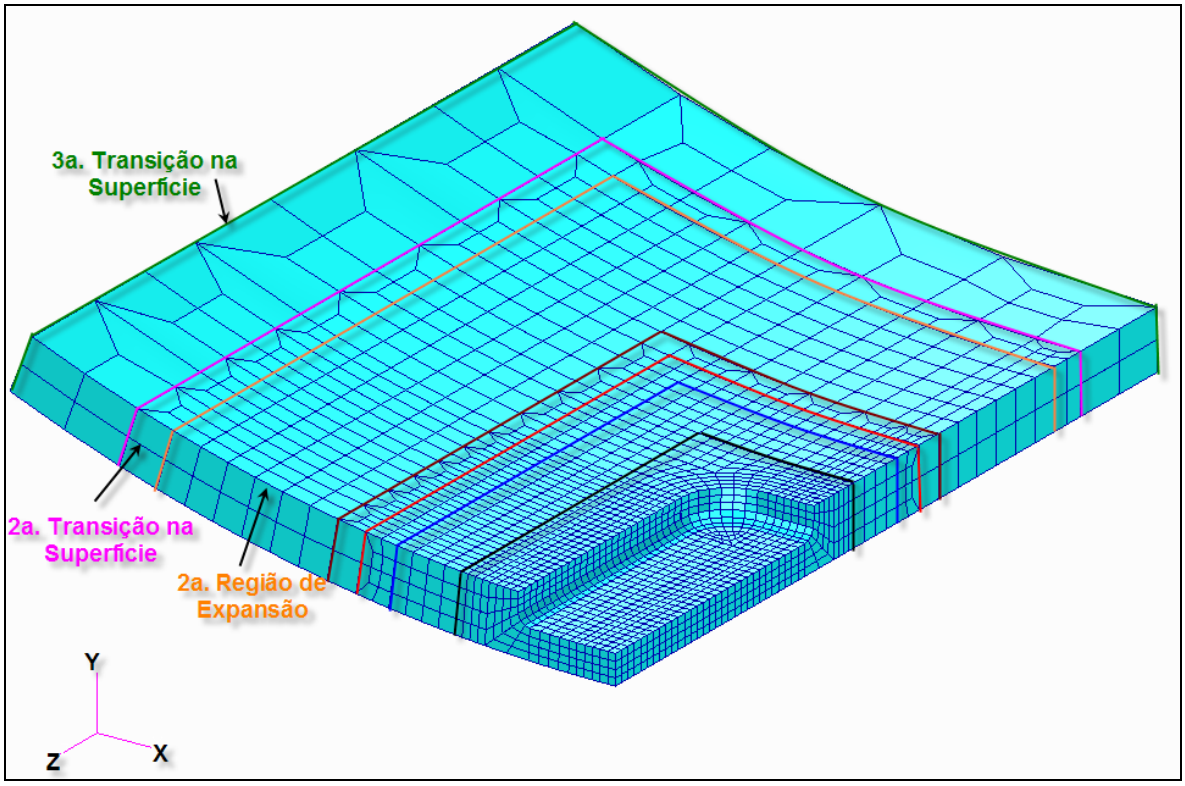


Figura 3.63 – Transições de malha nas regiões mais distantes do defeito.

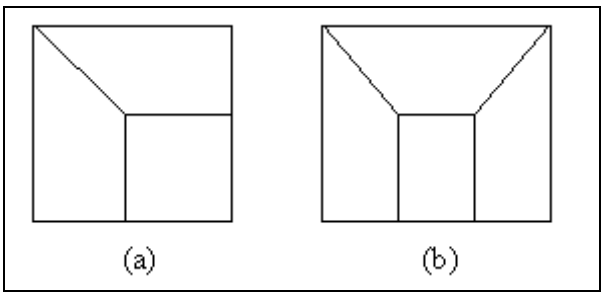


Figura 3.64 – Tipos de transição de elementos utilizadas: a) dois para um; b) três para um.

3.5.4 Definição dos Limites de Transição de Malha entre Defeitos Adjacentes

A seção anterior descreveu a discretização padrão utilizada nas várias regiões do defeito e adjacências. Em se tratando de geração de múltiplos defeitos alinhados longitudinalmente ou circunferencialmente, o programa é capaz de verificar a priori o nível de proximidade de um defeito em relação ao outro adjacente por meio da função implementada *Compute_InteractionParameters()*. Esta função calcula todas as distâncias de cada região de transição, descritas na seção anterior, com o objetivo de definir as distâncias limites entre defeitos adjacentes para possibilitar ou não a realização de determinada transição.

O programa permite três níveis de proximidade entre defeitos adjacentes conforme Figura 3.65. O nível 0, mostrado em detalhe na Figura 3.66, é o limite máximo de proximidade entre defeitos. Isto significa que o programa não permite a geração de defeitos cujas distâncias entre centros sejam menores que duas vezes o comprimento da caixa fixa (*2*Box_Defect*).

O nível 1 (em detalhe na Figura 3.67) e o nível 3 são os limites relacionados à primeira e terceira transição na superfície, respectivamente. De posse desses valores, o algoritmo de geração de múltiplos defeitos verifica a distância entre centros dos defeitos adjacentes e compara com as distâncias limites de transição. A decisão de realizar determinadas transições será dada de acordo com o nível de proximidade entre os defeitos. A Figura 3.65 mostra exemplo de oito defeitos alinhados longitudinalmente onde ocorre os três níveis de proximidade entre os defeitos. Devido à condição de simetria, apenas ¼ do duto foi modelado e portanto, apenas quatro dos oito defeitos estão sendo mostrados na Figura 3.65.

O primeiro par de defeitos (defeitos 5 e 6) está bem próximo da situação limite permitida pelo programa. Conforme pode ser visto no detalhe da Figura 3.66, a região remanescente entre os defeitos recebe um tratamento especial onde uma nova geometria e malha são gerados acompanhando sempre que possível as mesmas densidades de elementos das regiões adjacentes dos defeitos.

O segundo par de defeitos (defeito 6 e 7) já está um pouco afastado da região do defeito de forma que já é possível a realização das transições até a primeira transição na superfície (Figura 3.67). Novamente temos a geração de nova geometria e malha na região remanescente entre esses defeitos sempre utilizando transições do tipo de 2 para 1 ou de 3 para 1.

O último par de defeitos (defeito 7 e 8) está distante o suficiente para tornar possível a realização de todas as transições de malha das regiões adjacentes dos defeitos. Esta é a densidade de elementos que é mantida nas regiões distantes do defeito.

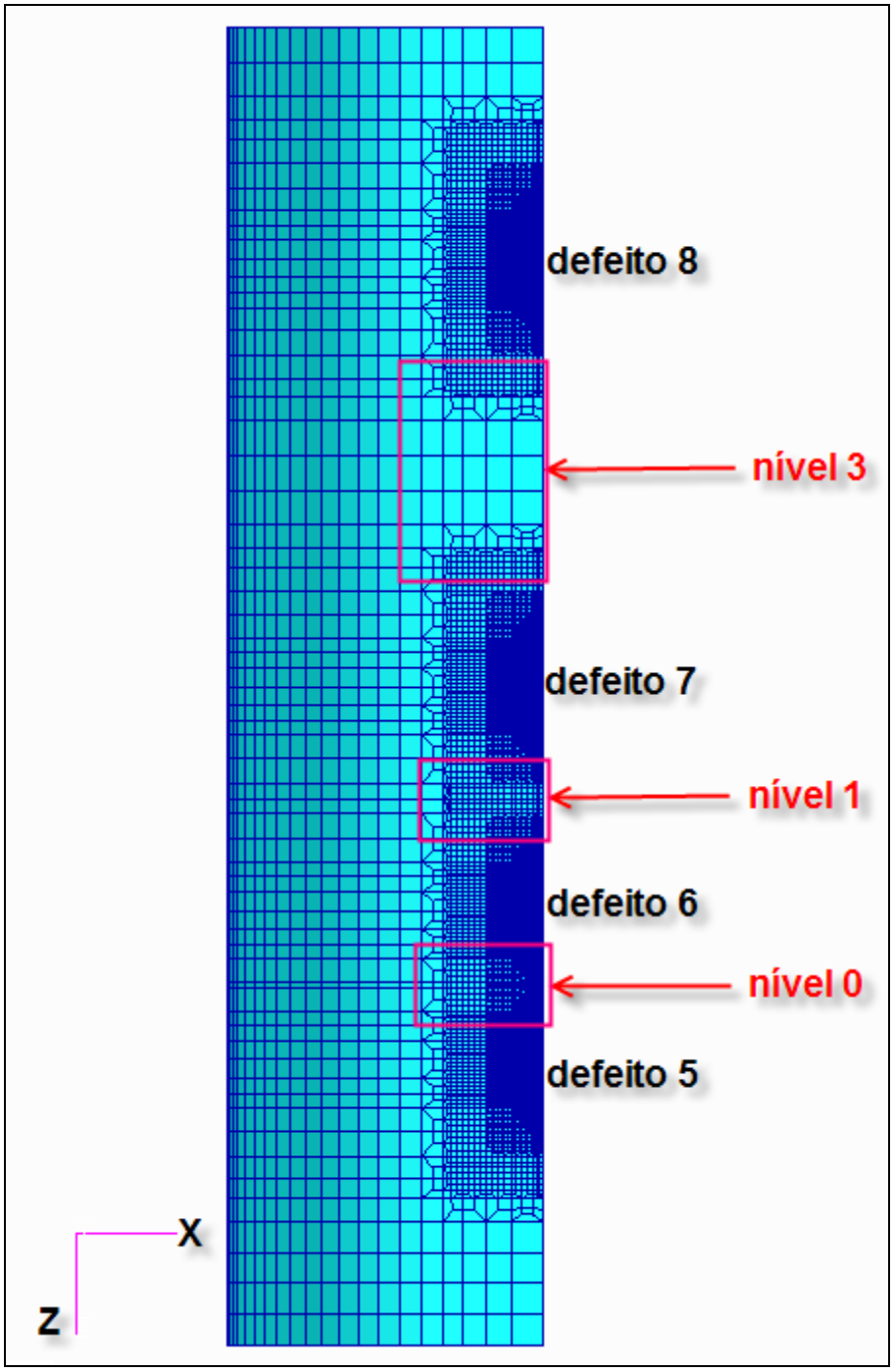


Figura 3.65 – Exemplo de modelo com três níveis de proximidades entre defeitos.

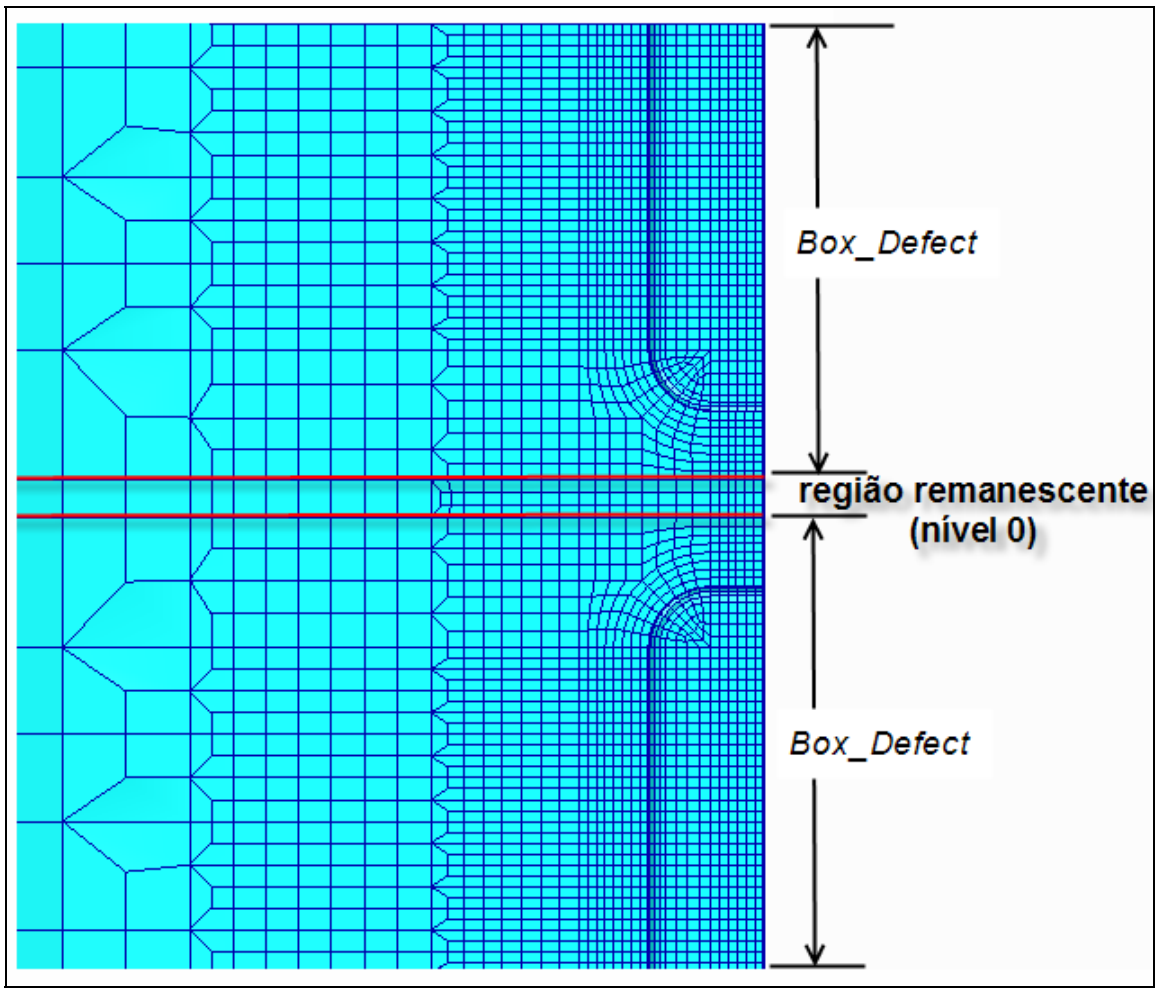


Figura 3.66 – Detalhe da região entre os defeitos 5 e 6 (nível 0).

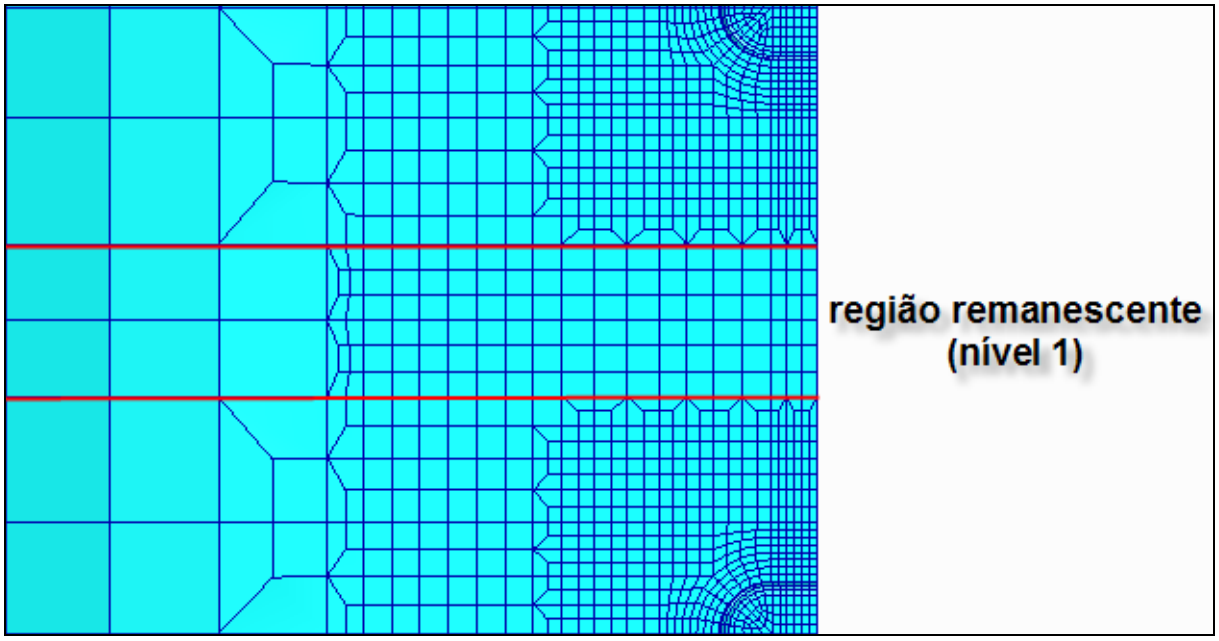


Figura 3.67 – Detalhe da região entre os defeitos 6 e 7 (nível 1)

3.5.5 Condições de Contorno e Carregamento

A aplicação de condições de contorno e carregamento adequadas no duto a ser analisado requer um conhecimento do comportamento das deformações do modelo real sob determinadas condições de operação além de exigir uma boa experiência na idealização de condições de apoio ou suportes.

Os dutos, como qualquer outra estrutura, podem estar submetidos a diversos tipos de carregamentos e condições de contorno. O software PATRAN permite facilmente introduzir condições de contorno que podem ser impostas sobre as entidades geométricas ou diretamente nas entidades de malha. Quando aplicadas à geometria, elas são transferidas às entidades de malha em um processo automático. A vantagem de aplicar as condições de contorno sobre a geometria é que caso seja necessário gerar uma nova malha sobre a geometria (por exemplo, refinar a malha) então não será necessário redefinir as condições de contorno novamente uma vez que a geometria é a mesma.

Os exemplos aqui tratados estão submetidos a um carregamento de pressão interna (P_{INT}) e tensão longitudinal (σ_{AXIAL}) de tração na borda livre do modelo para simular a situação existente num ensaio de laboratório (duto fechado com pressão interna) (Figura 3.68). As pressões são aplicadas sempre perpendicularmente à superfície interna do duto.

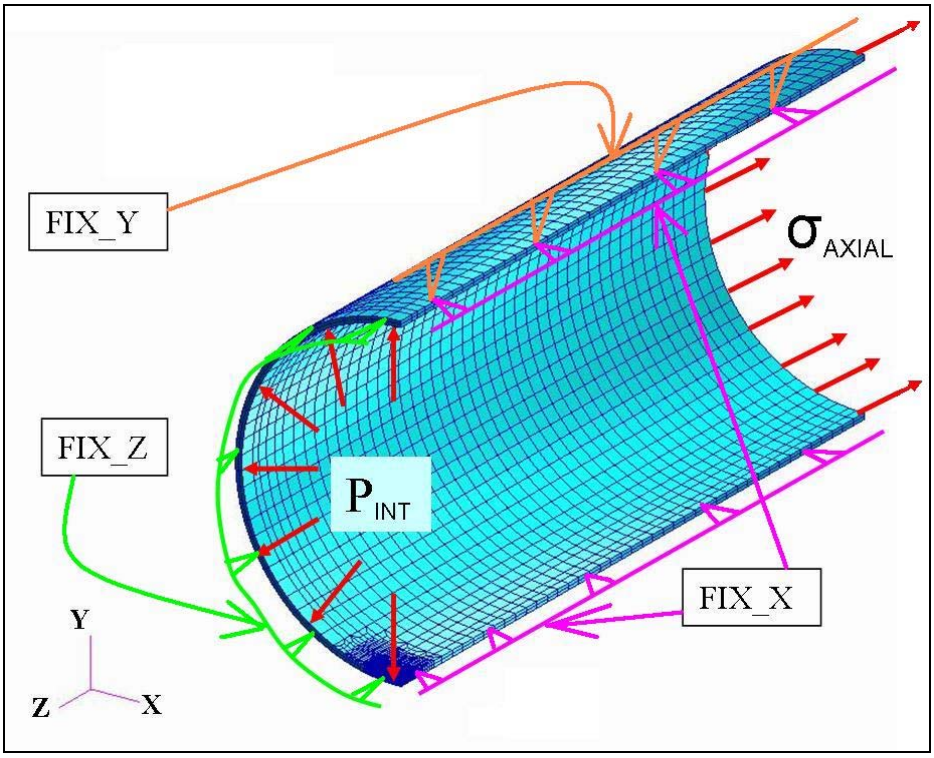


Figura 3.68 – Condições de contorno e carregamentos aplicados no duto considerando-se dois planos de simetria.

Em princípio, o usuário pode utilizar as ferramentas interativas do PATRAN para aplicar manualmente as condições de contorno desejadas. No entanto, em se tratando de dutos submetidos à pressão interna, o trabalho manual por parte do usuário seria enorme e extremamente desgastante uma vez que ele teria que selecionar todas as superfícies internas do modelo para então aplicar o carregamento de pressão interna.

Esta é uma tarefa que pode parecer um pouco trivial, mas quando se trata de modelos com múltiplos defeitos, onde o número de superfícies é muito grande, o trabalho manual torna-se praticamente inviável. Para se ter uma noção, o simples exemplo mostrado na Figura 3.65, com apenas quatro defeitos modelados usando as condições de simetria, possui 706 superfícies internas que devem ser selecionadas do modelo que possui um total de 5.800 superfícies.

Para contornar esse problema, foi implementada a função *Get_Internal_Surfaces()* que armazena em uma variável global (*Internal_Surfaces*) a identificação das superfícies internas em cada região do modelo à medida em que os defeitos são gerados. Assim, depois de gerado todo o modelo, o programa aplica automaticamente as condições de carregamento de pressão interna por meio da função do PATRAN *loadsbcs_create2()*.

Devido à simetria do defeito e carregamentos, pode-se modelar apenas ¼ do duto (Figura 3.68). Dessa forma, considera-se como condição de contorno a simetria na seção longitudinal do duto (FIX_X – restrição de deslocamentos nulos na direção x) e a simetria na seção transversal em uma das extremidades do duto (FIX_Z – restrição de deslocamentos nulos na direção z). Para evitar movimento de corpo rígido, restringem-se os deslocamentos na direção y (FIX_Y) ao longo de uma linha geratriz do duto.

3.5.6 Propriedades do Material

A modelagem da lei constitutiva dos materiais é um dos aspectos mais importantes na modelagem numérica, pois ela influi significativamente nos valores dos resultados.

Muitos materiais possuem comportamento linear-elástico quando estão sujeitos a baixos níveis de tensão e deformação, apresentando assim propriedades aproximadamente constantes. No entanto, a grande maioria dos materiais, como por exemplo, os aços, quando submetidos a altos níveis de tensão, apresentam uma mudança do comportamento elástico para o comportamento plástico. Isto significa que nestas situações, quando todas as forças atuantes na estrutura são removidas, a mesma não retorna para sua configuração (ou forma) original apresentando assim, algumas deformações plásticas permanentes. Esta mudança do comportamento elástico para o plástico caracteriza um comportamento não-linear do material.

Em uma análise não-linear por elementos finitos, onde estão envolvidas grandes deformações, é necessária a utilização de dados da relação tensão-deformação verdadeiras que caracterizam esse comportamento não-linear do material. A curva tensão-deformação verdadeira do material pode ser construída a partir dos dados experimentais obtidos no ensaio de tração em corpos de prova retirados dos espécimes tubulares. Geralmente, os testes de tração são executados em corpos de prova retirados na direção longitudinal do duto. Dependendo do processo de fabricação, é necessário também retirar corpos de prova na direção transversal (Diniz, 2002).

O material considerado nas análises foi o aço API 5L-X80, o mesmo utilizado nos trabalhos experimentais e numéricos realizados por Benjamin et al (2005) e Andrade et al (2006), respectivamente. Este material possui comportamento elasto-plástico com endurecimento isotrópico e foi adotado como critério de escoamento o critério de von Mises. A Figura 3.69 apresenta a curva tensão verdadeira *versus* deformação verdadeira do material construída a partir da equação de Ramberg-Osgood determinada nos experimentos realizados por Benjamin et al (2005) conforme expressão abaixo:

$$\varepsilon^* = \frac{\sigma^*}{E} + 0,0788174 \cdot \left(\frac{\sigma^*}{\sigma_u^*} \right)^{12,642026} \tag{3.1}$$

onde: ε^*, σ^* e σ_u^* correspondem à deformação verdadeira, tensão verdadeira e tensão última verdadeira do material, respectivamente.

A Tabela 3.7 mostra os principais dados do material bem como os pares de tensão verdadeira *versus* deformação verdadeira utilizados na modelagem numérica.

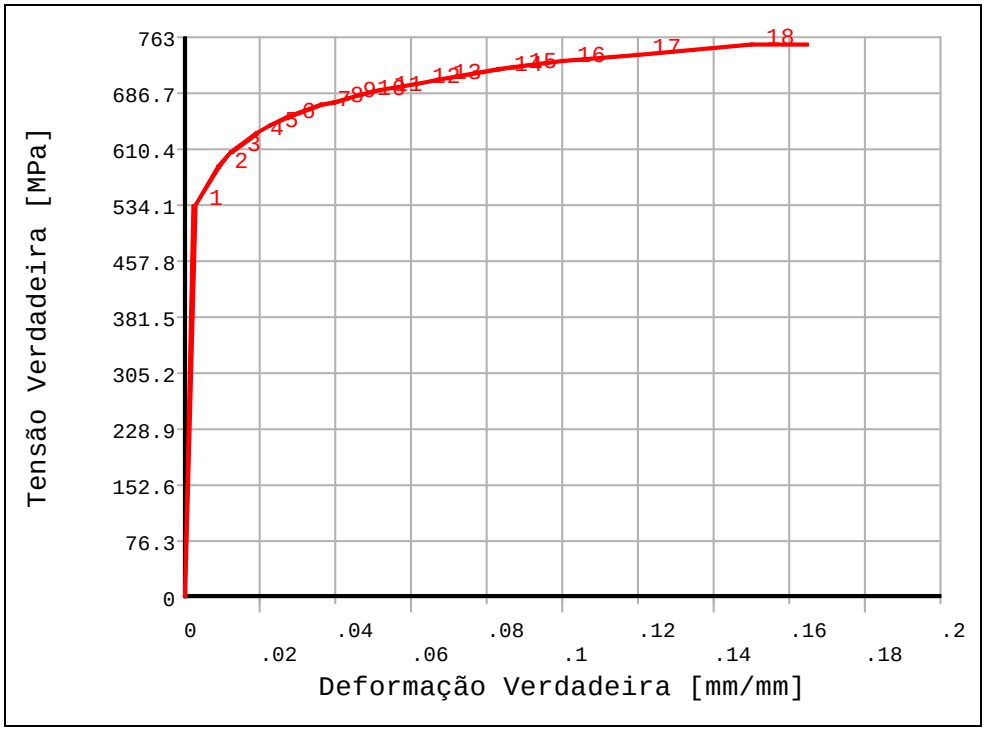


Figura 3.69 – Curva tensão verdadeira *versus* deformação verdadeira (material API 5L-X80).

Tabela 3.7 – Dados do Material API 5L-X80.

Módulo de elasticidade longitudinal	$E = 200.000MPa$
Coeficiente de Poisson	$\nu = 0,3$
Tensão de escoamento	$\sigma_{esc} = 534,1MPa$
Deformação correspondente à tensão σ_{esc}	$\varepsilon_{esc} = 0,002671$
Tensão última verdadeira	$\sigma^{*}_{ult} = 718,2MPa$
Deformação verdadeira (mm/mm)	***Tensão verdadeira(MPa)***
0,0000000	0,0000
0,0026710	534,1000
0,0090000	586,3560
0,0125000	607,3325
0,0187197	631,7030
0,0225671	642,6945
0,0272592	653,7068
0,0362591	670,2555
0,0395930	675,3539
0,0432324	680,4538
0,0472031	685,5568
0,0515325	690,6619
0,0613880	700,8750
0,0669801	705,9840
0,0830373	718,6498
0,0868612	721,3181
0,1000000	729,7026
0,1200000	740,6435
0,1500000	754,1836

4 AUTOMATIZAÇÃO DA ANÁLISE NÃO-LINEAR

4.1 Introdução

A pressão de ruptura de dutos corroídos, submetidos a diversos tipos de carregamentos, pode ser estimada por meio de simulações numéricas não-lineares usando o método dos elementos finitos e um adequado critério de falha previamente validado. Neste trabalho as análises não-lineares foram realizadas utilizando o programa comercial ANSYS (2004).

Conforme foi visto anteriormente, em uma análise linear, o método dos elementos finitos (MEF) resulta em um sistema de equações do tipo [K][u] =[F], que é resolvido facilmente por métodos diretos ou iterativos. Para o caso de problemas onde se pretende estimar a pressão de ruptura de dutos, grandes deformações estão envolvidas o que conseqüentemente obriga a realização de análises não-lineares. Nestes tipos de análises o sistema de equações passa a ser resolvido por incrementos, ou seja, [K][Δu] =[ΔF]. Isso se deve ao fato de [K] não mais ser constante, passando a depender do estado atual e da história de deformação do material (Diniz, 2002).

O comportamento não-linear de dutos corroídos pode ser ocasionado devido basicamente a dois fatores principais: não-linearidade geométrica e a não-linearidade física (ou do material).

A não-linearidade geométrica é resultante geralmente da influência de grandes deformações sofridas pela estrutura. Por exemplo, a determinação dos esforços solicitantes em uma dada estrutura é feita, geralmente, supondo a mesma na sua configuração original (posição não-deformada). Fazendo isso, despreza-se a deformação da estrutura. Considerando, porém, a mudança na configuração geométrica (posição deformada) pode causar uma resposta não-linear da estrutura geralmente caracterizada por grandes deslocamentos e/ou rotações. Para todos os modelos aqui analisados, é considerado o efeito da não-linearidade geométrica através do comando "*NLGEOM, ON*" do ANSYS. Através deste comando, o procedimento de grandes deformações é ativado fazendo com que não haja nenhum limite teórico na rotação ou deformação sofrida por um elemento. No entanto, este procedimento requer que o incremento de deformação seja limitado a fim de manter a precisão dos resultados da análise.

A não-linearidade geométrica indica que pode haver um comportamento não-linear da estrutura mesmo quando o material é elástico-linear. O problema se agrava quando o próprio material não é linear, o que caracteriza a não-linearidade física. Ao contrário da não-linearidade geométrica, a não-linearidade física é uma propriedade intrínseca do material que resulta em uma relação não-linear entre tensão e deformação. Esta relação não-linear entre tensão e deformação é expressa na curva tensão verdadeira *versus* deformação verdadeira do material conforme visto na Figura 3.69 da seção 3.5.6. Neste tipo de não-linearidade várias considerações podem ser feitas como, por exemplo, o uso de teorias constitutivas não-lineares (teoria da plasticidade, viscoplasticidade, etc.) que podem levar em conta os efeitos da temperatura e do tempo.

Estas não-linearidades fazem a matriz de rigidez da estrutura modificar-se à medida que a estrutura se deforma. Desta forma, quando estes efeitos estão presentes, a solução do problema deve ser obtida por métodos incrementais e iterativos como o método de Newton-Raphson. Também, tornam-se necessárias técnicas de controle como: controle de carga, controle de deslocamento, controle do comprimento de arco. Cada um destes métodos e técnicas têm suas particularidades e devem ser escolhidos de maneira adequada. Devido às suas características de convergência, o Método "Full Newton-Raphson" foi utilizado, no qual a matriz de rigidez é atualizada a cada iteração de equilíbrio. Para maiores detalhes, consultar Bathe(1996), Crisfield (1991) e ANSYS(2004).

4.2 Ferramentas Existentes para Análise Não-Linear

Os softwares comerciais de elementos finitos atualmente existentes no mercado possuem ferramentas poderosas para este tipo de análise, possibilitando que o usuário estabeleça parâmetros para a execução e controle da análise, entre eles, pode-se citar: tipo de análise (estática ou transiente, linear ou não-linear); controle do tempo (número de "substeps", tamanho do incremento de carga); tipo de solver (Direto com Matriz Esparsa, Iterativo, Pré-Condicionado, Direto Frontal, Gradiente Conjugado, entre outros); e critérios de parada e convergência (número máximo de iterações, incremento de carga automático, etc.) (ANSYS, 2004).

Basicamente uma análise não-linear consiste em uma seqüência de passos de incremento de carga ("load step"), que podem ser divididos em incrementos menores ("substeps"). Dentro de cada "substep", são realizadas iterações ("equilibrium iterations") até que algum determinado critério de convergência seja atingido. A Figura 4.1 ilustra um histórico típico de carregamento em uma análise não-linear com os incrementos de carga ("load steps") e seus respectivos "substeps" indicados ao longo dos passos de incremento.

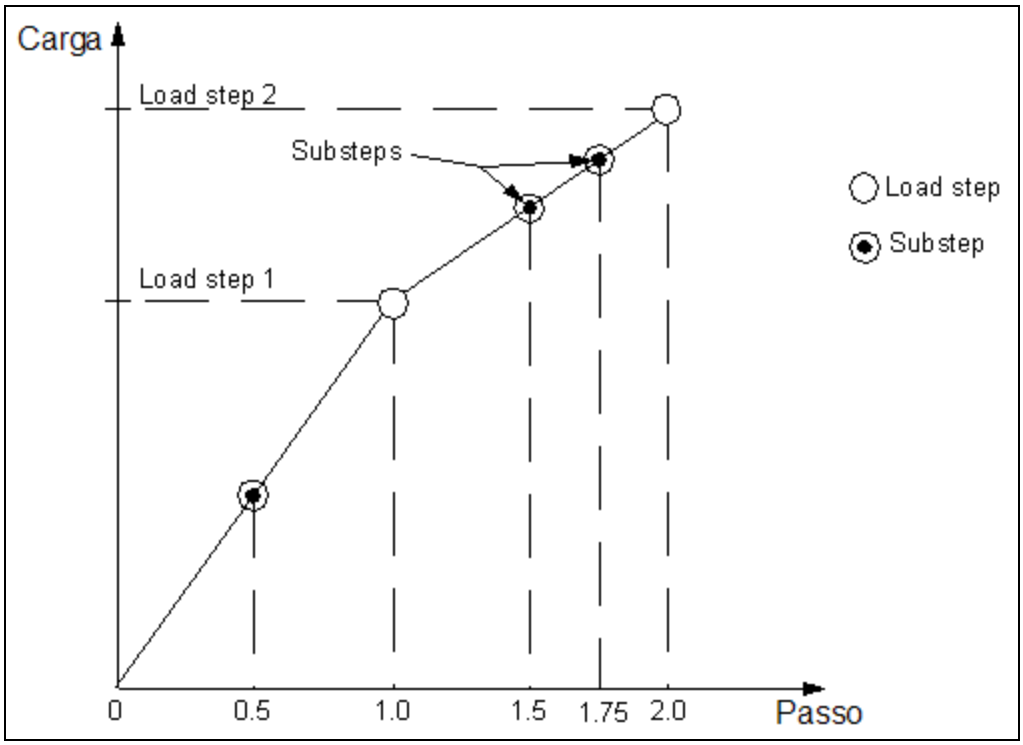


Figura 4.1 – Passo de incremento de carga e seus respectivos "substeps".

Particularmente, o ANSYS utiliza o método Newton-Raphson para resolver problemas não-lineares. Antes de cada solução, o método Newton-Raphson avalia o vetor de resíduos ou forças não-equilibradas ("out-of-balance load vector") que é a diferença entre as forças internas (cargas correspondentes às tensões nos elementos) e as cargas externas aplicadas. O método obtém uma solução linear, usando o vetor de forças não-equilibradas e em seguida verifica o critério de convergência. Caso o critério não seja atingido, o vetor de forças não-equilibradas é avaliado novamente, a matriz de rigidez é atualizada e uma nova solução é obtida. Este processo iterativo continua até que o problema atinja algum critério de convergência.

Outras funcionalidades como "line search", "automatic load step", bisseção, Newton-Raphson modificado, entre outras, também podem ser ativadas para auxiliar no problema de convergência e incremento de carga.

Em algumas análises não-lineares se for usado apenas o método Newton-Raphson a matriz de rigidez tangente ("tangent stiffness matrix") pode tornar-se singular, causando problemas de convergência. Para estas situações, pode-se ativar o método do comprimento de arco ("Arc-Length") como um esquema alternativo de iteração para evitar pontos de singularidades. O método "Arc-Length" faz com que as iterações de equilíbrio de Newton_Raphson convirjam ao longo de um arco, impedindo desse modo a divergência mesmo quando a inclinação da curva carga *versus* deslocamento torna-se zero ou negativa.

No ANSYS, os vários parâmetros disponíveis permitem que o usuário controle a análise por meio de bisseções no incremento de carga. Ou seja, o usuário informa limites para alguma variável do problema físico (por exemplo: incremento máximo de deformação), de forma que, se este limite for excedido, ocorre então a re-análise do "substep".

4.2.1 Recursos "Automatic Time Step" e "Save-Restart"

Em particular, no ANSYS, o usuário pode optar pelo recurso "Automatic Time Step". Este recurso ativa um esquema automático de forma a garantir que a variação do incremento de carga não seja nem muito grande (o que resulta em muitas bisseções e re-análises) e nem também muito conservador (o que resulta em um incremento de carga muito pequeno). No final de um incremento de carga, o tamanho do próximo incremento é calculado baseado em fatores como: número de iterações usadas no último incremento de carga (muitas iterações ocasionam uma redução do tamanho do próximo incremento); mudança no "status" de elementos não-lineares prestes a acontecer (o tamanho do incremento de carga é reduzido quando uma mudança no status é iminente); e tamanho do incremento de deformação plástica (ANSYS, 2004).

No entanto, algumas situações particulares requerem um controle mais determinístico da aplicação dos incrementos de carga. Uma alternativa viável no ANSYS é o uso do recurso "save/restart" que possibilita que o usuário reinicialize a análise não-linear a cada incremento de carga a partir de um passo imediatamente anterior. Apesar de ser um procedimento manual, este tipo de recurso possibilita que o usuário controle a cada incremento de carga os resultados gerados durante a análise e estabeleça critérios próprios de convergência baseados nos valores de variáveis físicas quaisquer, o que não ocorre no recurso "Automatic Time Step" no qual o controle da análise é limitado somente a alguns parâmetros pré-definidos pelo solver.

Além das análises não-lineares já serem bastante demoradas, o uso do procedimento "save/restart" padrão fornecido pelo ANSYS exige que o engenheiro detenha boa parte do seu tempo coletando e interpretando os resultados obtidos a cada passo de incremento de carga, o que torna o trabalho bastante repetitivo e demorado, e por estas razões, muito propenso a erros.

A fim de tornar este processo mais rápido e mais confiável, foi criado um procedimento automático, usando o recurso "save/restart", para controlar análises não-lineares no ANSYS baseado no procedimento padrão pré-estabelecido pelo CENPES/PETROBRÁS (Benjamin & Andrade, 2005). Para isso, foi implementado um programa interpretado ("script"), na linguagem de programação Python (PYTHON, 2005), a partir do qual toda a análise é gerenciada por meio da execução automática de tarefas pré-determinadas. Assim, a cada iteração da análise não-linear, o solver (ANSYS) é acionado pelo "script" e em seguida, os resultados gerados durante aquela iteração, são extraídos por meio de manipulação de arquivos (Cabral et al, 2006a).

A interpretação desses resultados pelo "script", possibilita que os critérios de convergência e de incremento de carga, determinados pelo usuário, sejam verificados e calculados automaticamente. No final da análise, é gerado também automaticamente um resumo do histórico da análise não-linear em um arquivo no formato Excel.

4.3 Linguagem Python

4.3.1 Introdução

PYTHON é uma linguagem de programação de altíssimo nível (VHLL – "Very High Level Language"), de sintaxe simples e moderna, orientada a objetos, interpretada, de tipagem dinâmica (não há pré-declaração de variáveis), altamente modular, multiplataforma, de fácil aprendizado e implementação livre (PYTHON, 2005). Apesar de ser uma linguagem de uso geral, que pode ser empregada em vários tipos de problema, PYTHON é bastante utilizado como linguagem de "scripting", isto é, programas escritos em linguagens interpretadas que automatizam tarefas repetitivas (como manipulação de texto, cópia de arquivos, etc.) ou que conectam programas distintos (por exemplo, acionar o solver ANSYS a cada passo da análise não-linear).

4.3.2 Manipulação de Arquivos

PYTHON possui uma série de funções de manipulação de arquivos pré-definidas, que já estão disponíveis quando se executa o interpretador, sem ter que recorrer a bibliotecas externas. Por meio das funções *file()* e *open()*, obtém-se uma referência a um objeto do tipo arquivo que possui um conjunto de métodos úteis para abrir, ler, editar e fechar arquivos (PYTHON, 2005).

4.3.3 Módulos Padrões e Pacotes Utilizados

Conforme dito anteriormente, PYTHON é uma linguagem altamente modular. Esta modularização pode ter diversas motivações: o programa pode ser grande demais (pode ser desejável dividi-lo em vários módulos para facilitar a manutenção); ter sub-partes reusáveis que devem ser separadas; ou ainda necessitar de módulos escritos por terceiros (isto significa que provavelmente alguém já fez todo ou boa parte do programa que precisa ser desenvolvido, economizando tempo de programação e depuração) (Labaki; Reis, 2004). Os principais módulos utilizados neste trabalho foram os módulos *RE*, *OS*, *SHUTIL*.

O módulo *RE* ("regular expressions") dispõe de vários métodos usados para a pesquisa de padrões em arquivos de texto. Expressões regulares são combinações de caracteres especiais que descrevem um conjunto de padrões de texto permitindo localizar e reconhecer um trecho de texto dentro de um arquivo ou uma cadeia de caracteres. Dessa forma, as ações de substituição de "strings" nos arquivos utilizados na análise não-linear a cada incremento de carga, podem ser feitas em um texto se a expressão regular utilizada coincidir com alguma parte do texto onde a procura está sendo feita. Esta pesquisa por um determinado padrão, dentro dos arquivos de saída do solver ANSYS, pode ser feita por meio das funções, *re.search* e *re.match*, enquanto que podemos extrair os resultados contidos no arquivo por meio da função *re.group.*

O módulo *OS* ("operating system") oferece funções relacionadas ao ambiente de execução do sistema incluindo as funções de criar e remover diretórios, *os.mkdir()* e *os.rmdir()*; alterar nomes de arquivos e diretórios, *os.rename()*; remover arquivos, *os.remove()*; e execução de comandos do sistema, *os.system()*. Esta última função, permite que o "solver" ANSYS seja acionado e executado como se estivesse sendo feita sua chamada no prompt do MsDOS, ou na linha de comando do UNIX.

O módulo *SHUTIL* ("high level file operations") oferece uma grande quantidade de operações de alto nível em arquivos. Em especial, foi utilizada a função *shutil.copy*, para copiar os arquivos gerados pelo ANSYS durante um determinado passo para um diretório de Backup.

Embora a biblioteca padrão do PYTHON possua vários módulos que cobrem uma grande variedade de necessidades de programação, também é necessário adicionar novas funcionalidades à instalação padrão do PYTHON.

Por exemplo, neste trabalho, houve a necessidade de gerar automaticamente planilhas no formato Excel contendo o histórico da análise não-linear. Isto foi possível por meio da instalação adicional do pacote PyExcelerator (2005). PyExcelerator é uma biblioteca de módulos e funções destinados à geração de planilhas compatíveis com os formatos Excel e OpenOffice Calc.

4.4 Estratégia de Aplicação dos Incrementos de Carga

O único carregamento aplicado nos modelos aqui analisados é a pressão interna. Nos modelos de elementos finitos este carregamento é representado por dois tipos de cargas. O primeiro tipo é a pressão interna "P_{INT}" propriamente dita, aplicada perpendicularmente à superfície dos elementos. O segundo tipo é a carga longitudinal de tração, chamada de carga de extremidade, transmitida para as extremidades do tubo pelos tampos planos usados na vedação dos dutos durante um eventual ensaio experimental. Nos modelos constituídos por elementos sólidos esta carga é uma pressão P_L, aplicada na extremidade do modelo na direção longitudinal. Esta pressão, resultado do equilíbrio de forças na direção longitudinal no duto, é dada por:

$$P_L = P_{INT} \cdot \left(\frac{D_i^2}{D_e^2 - D_i^2} \right) \tag{4.1}$$

onde D_e e D_i são o diâmetro externo e o diâmetro interno do duto, respectivamente.

O carregamento, constituído pela pressão interna e pela carga de extremidade, é aplicado de forma incremental, de acordo com a estratégia de carregamento apresentada nos parágrafos a seguir, até o duto atingir a ruptura. O valor do carregamento dentro de cada passo de carga ("load step") é especificado no ANSYS para variar linearmente (comando KBC,0).

Inicialmente é realizada uma análise linear elástica do modelo, submetido a uma pressão P_0 de valor unitário, para determinação do valor máximo da tensão de von Mises correspondente a esta pressão $(\sigma_{\max})_0$. Considerando-se que o modelo usado nesta análise é linear, determina-se o valor aproximado da pressão de início de escoamento (P_{esc}) da seguinte forma:

$$P_{esc} = \sigma_{esc} \frac{P_0}{(\sigma_{\max})_0} \tag{4.2}$$

onde:

σ_{esc} - tensão de escoamento do material.

A pressão a cada i-ésimo passo (P_i) é calculada baseada na soma da pressão do passo imediatamente anterior (P_{i-1}) mais o valor do incremento de pressão atual (ΔP_i), conforme equação abaixo:

$$P_i = P_{i-1} + \Delta P_i \tag{4.3}$$

Os incrementos de pressão interna (ΔP_i), aplicados ao duto na análise de elementos finitos, devem variar. Até o ponto imediatamente antes do início do escoamento, o material apresenta um comportamento linear.

Nesta situação o incremento pode ser grande (ou mesmo único) até que o primeiro elemento escoe. No decorrer da análise elasto-plástica, o incremento de pressão ótimo tende a diminuir, porque a cada incremento mais elementos se plastificam.

O trabalho de Fu & Kirkwood (1995) recomenda incrementos da ordem de 10^{-5} da pressão total aplicada. Este incremento, por ser pequeno, é recomendado para valores de pressão mais próximos da ruptura, sendo desnecessário no começo da análise.

Neste trabalho, os valores iniciais da pressão e do incremento de carga aqui adotados para o primeiro passo da análise são, respectivamente:

$$P_1 = 0,8 \cdot P_{esc} \qquad \Delta P_1 = \frac{P_{esc}}{3} \tag{4.4}$$

O coeficiente 0,8 garante que no primeiro passo de carga inicial todos os pontos do modelo permanecem no regime elástico. No segundo passo de carga pode ocorrer, ou não, o início da plastificação, dependendo do grau de precisão da estimativa feita para o valor de P_{esc} . Cada passo de carga é dividido em 4 sub-passos ("*substeps"*).

O valor de ΔP_i para os passos subseqüentes será o mesmo, a menos que a análise seja interrompida por falta de convergência em um dos "*substeps*" ou porque o valor do incremento máximo de deformação plástica ultrapassou o limite estabelecido pelo usuário. Caso ocorra uma dessas duas situações, o valor de ΔP_i deverá ser reduzido de 50% do valor anteriormente utilizado e, em seguida, corrige-se automaticamente o valor da pressão com este novo incremento e faz-se novamente a execução da análise daquele passo onde não houve a convergência. O valor do incremento máximo de deformação plástica utilizado foi de 0,0025 baseado no procedimento padrão do CENPES (Benjamin & Andrade, 2005).

4.5 Critério de Convergência

O programa ANSYS possibilita que o usuário defina os critérios de convergência baseados nos valores de força, momentos, deslocamentos, rotações ou qualquer combinação destes itens podendo assumir diferentes valores cada um. O critério aqui adotado foi o baseado nos valores de forças (*F*) por meio do seguinte comando do ANSYS:

```
CNVTOL,F,,0.001,2,1
```

Neste caso, o "substep" irá atingir o critério de convergência se as forças não-equilibradas "out-of-balance force" (verificadas para cada grau de liberdade separadamente) forem menores ou iguais que o valor de 0,001.

O número máximo de iterações a serem executadas dentro de cada "*substep*" foi limitado a 50 iterações (comando no ANSYS: *NEQUIT, 50*). Caso o problema não convirja dentro do número máximo de iterações pré-definido, ocorre então uma correção (redução) no valor do incremento de carga ΔP_i de 50% conforme visto anteriormente.

4.6 Critério de Ruptura

A ruptura de um duto corroído pode ser definida como o instante em que a sua estanqueidade é perdida, ou seja, o instante em que começa a ocorrer vazamento (de líquido ou gás) através de uma descontinuidade (rasgo, fissura ou trinca) surgida na região do defeito.

A formulação do método dos elementos finitos, usada neste trabalho, é baseada na Mecânica do Contínuo. Por esta razão, é necessário definir um critério para detectar ao longo da análise a proximidade de um estado de ruptura do duto corroído (Chouchaoui et al, 1992; Fu & Kirkwood, 1995; Batte et al, 1997; apud Andrade et al, 2006).

O critério de ruptura aqui adotado estabelece que a análise é interrompida quando a tensão de von Mises em qualquer ponto da região do defeito atinge um valor igual à tensão última verdadeira (σ_{ult}^{*}) do material. No entanto, para os modelos usados na validação das ferramentas deste trabalho, o critério de ruptura foi o mesmo adotado por Andrade et al (2006) que diz que a ruptura ocorrerá quando as tensões de von Mises ao longo da direção radial (todos os pontos situados ao longo da espessura do duto), atingirem o valor igual à tensão última verdadeira do material.

Além dos critérios acima mencionados, a análise deve ser interrompida (critério de parada) caso o valor do incremento de pressão ΔP_i fique menor que 0.01MPa, pois do ponto de vista de engenharia, incrementos menores que este valor são desprezíveis (Benjamin & Andrade, 2005).

4.7 Integração dos Sistemas PATRAN/ANSYS

O programa MSC.PATRAN, utilizado para o desenvolvimento das ferramentas de modelagem automática do programa PIPEFLAW (apresentadas no capítulo 1), é um sistema aberto que permite que códigos de análises desenvolvidos por terceiros sejam integrados e acessados no sistema PATRAN. Além disso, o próprio PATRAN já fornece interface para alguns dos principais códigos comerciais de análise de elementos finitos tais como ABAQUS, ANSYS, MARC, NASTRAN, DYTRAN, LS-DYNA3D, PAMCRASH, SAMCEF, entre outros (PATRAN, 2005).

Estas interfaces convertem os dados contidos no arquivo base de dados do PATRAN em dados de entrada para análise na plataforma escolhida. Após a execução do código da análise, a interface permite que os resultados obtidos da análise sejam convertidos novamente para a base de dados do PATRAN para posterior pós-processamento e visualização dos resultados. O usuário executa todas essas tarefas sem precisar sair de dentro do ambiente do PATRAN.

Neste trabalho, utilizou-se o código ANSYS (ANSYS, 2004) para realizar as análises não-lineares dos modelos de duto com defeitos.

A Figura 4.2 ilustra de forma resumida a integração do PATRAN com o ANSYS durante o processo de geração do arquivo de entrada do ANSYS para posterior análise via procedimento automático.

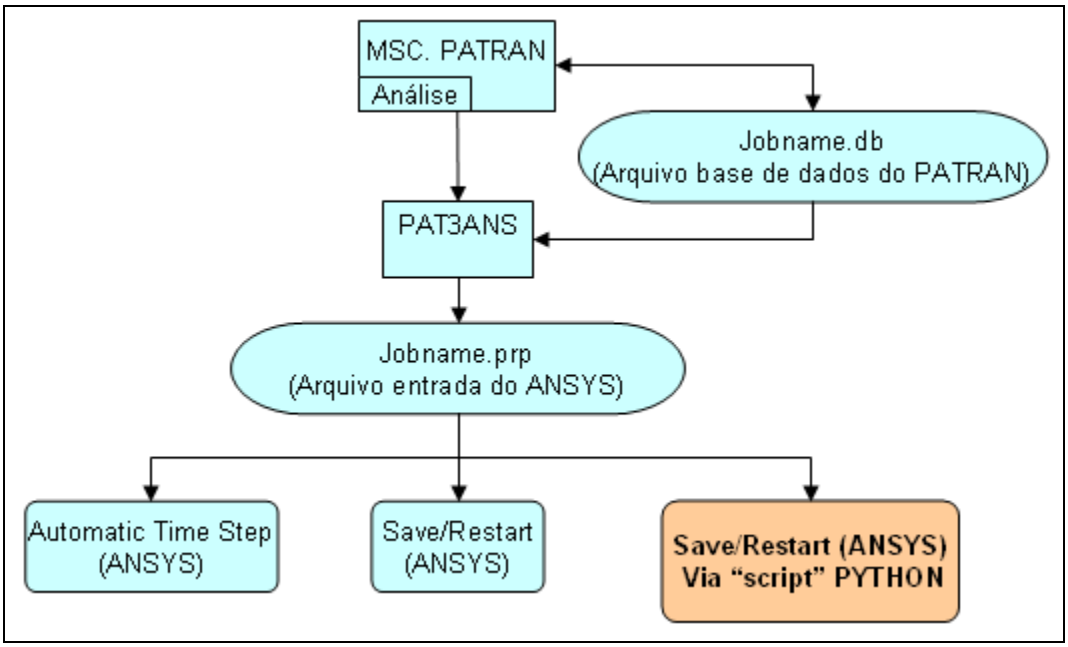


Figura 4.2 – Processo de integração do PATRAN com o solver ANSYS.

Após definir todos os parâmetros da análise, o usuário aciona a análise no ambiente do PATRAN dando início ao processo de integração. Durante o processo, o programa PAT3ANS é executado através do PATRAN. É este executável que realiza a conversão dos dados contidos no arquivo do PATRAN ("Jobname.db") gerando assim, o arquivo "Jobname.prp" pronto para análise no ANSYS[35].

4.8 Descrição do Procedimento Automático

O procedimento de automatização de análise não-linear aqui descrito foi implementado a partir de um "script" na linguagem PYTHON de onde toda a análise é gerenciada por meio da execução automática de tarefas pré-determinadas.

A Figura 4.3 mostra de forma simplificada os principais blocos de tarefas executadas pelo "script" durante o processo de automatização da análise não-linear e que serão descritos mais detalhadamente nos parágrafos seguintes.

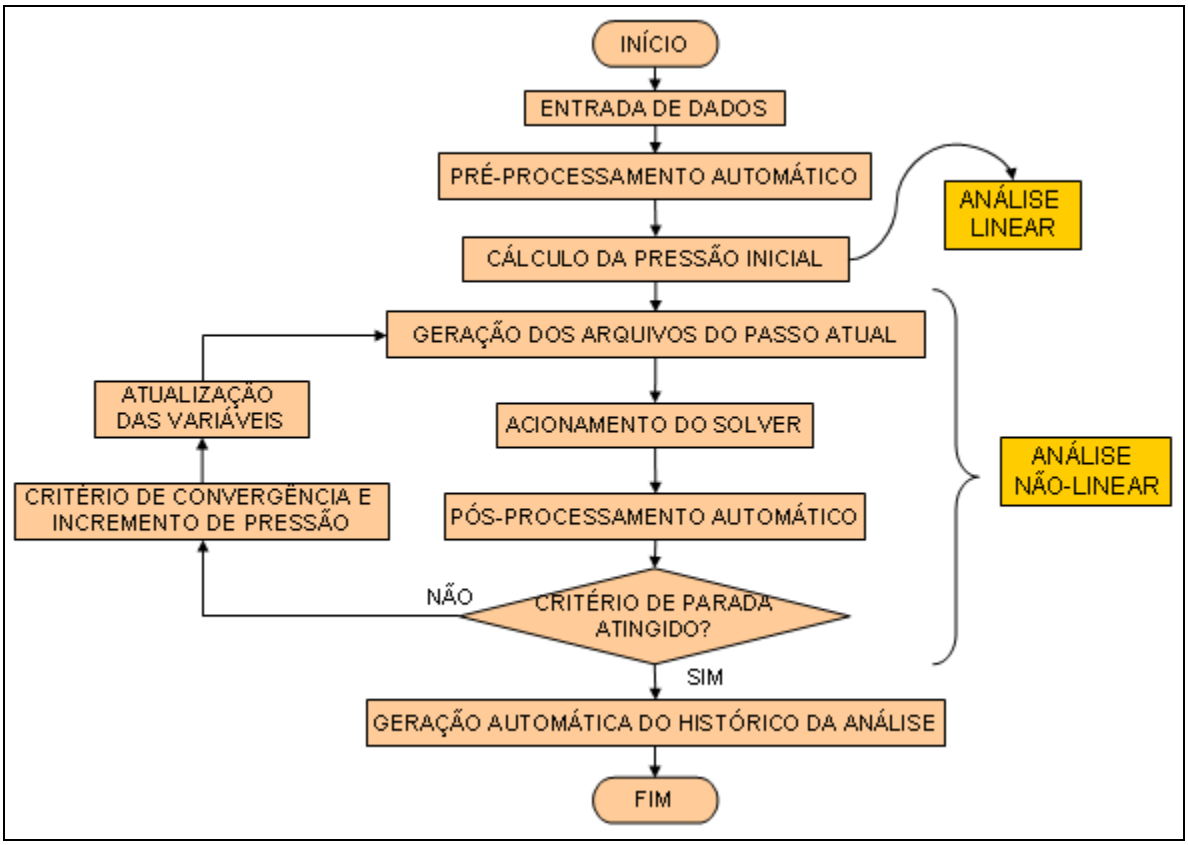


Figura 4.3 – Fluxograma das principais tarefas executadas pelo "script" de automatização.

4.8.1 Entrada de Dados

Antes de utilizar o "script" de automatização, é necessário fornecer todos os dados para a realização da análise não-linear. Primeiro, o usuário deverá copiar o arquivo *Jobname.prp* para o diretório de entrada ("*C:\Test\Input*") e informar o nome (*Jobname*) a partir do qual, serão gerados automaticamente todos os arquivos e diretórios durante a análise.

[35] Nota: Na realidade, o arquivo "Jobname.prp" está pronto para a análise tradicional usando os recursos padrão disponíveis no ANSYS. Como estamos utilizando um procedimento automático personalizado, este arquivo teve que ser editado automaticamente por meio de um pré-processador implementado na linguagem PYTHON, conforme será visto no item 4.8.2 deste capítulo.

Depois, o usuário deve fornecer os seguintes dados:

Dados relacionados à geometria:
- diâmetro e espessura do duto;

Dados relacionados ao material:
- módulo de elasticidade, coeficiente de Poisson, pares de pontos da curva do material (tensão verdadeira x deformação verdadeira);

Dados relacionados aos critérios de convergência e parada:
- tolerância para o incremento máximo de deformação plástica, tensão de ruptura do material, tolerância para o incremento de pressão mínimo, número de "*substeps*".

Essa entrada de dados ainda não possui interface gráfica e o usuário deve abrir o arquivo *Pre-Processor.py* para editá-lo manualmente. Uma futura implementação de uma interface gráfica para entrada desses dados pode ser feita, por exemplo, através do módulo *Tkinter* do PYTHON. Este módulo fornece ferramentas baseadas na linguagem TCL/Tk para construção dos principais elementos gráficos. Como sugestão, pode-se também optar por outras linguagens alternativas tais como a linguagem Delphi.

4.8.2 Pré-Processamento Automático e Análise Linear

O processo de integração entre o sistema PATRAN e o solver ANSYS permite que todo o arquivo base de dados do PATRAN seja convertido em comandos do ANSYS para a posterior análise. O resultado dessa conversão automática feita pelo PATRAN gerou o arquivo de entrada do ANSYS aqui denominado *Jobname.prp* (ver Figura 4.2).

Este arquivo possui todos os comandos do ANSYS necessários para a realização da análise não-linear utilizando as ferramentas padrões existentes. No entanto, o procedimento automático (utilizando o recurso "save/restart" do ANSYS via script PYTHON, proposto neste trabalho) requer a criação de alguns arquivos padrões que são utilizados para definir o modelo, aplicar os incrementos de carga e recuperar os resultados de tensão e deformações a cada passo durante a análise não-linear. Para isso, teve-se que implementar o arquivo *Pre-Processor.py* cujo principal objetivo é criar os arquivos necessários para a análise não-linear já no formato padrão utilizado pelo procedimento automático.

A partir do arquivo "*Jobname.prp*" e dos dados de entrada fornecidos pelo usuário, são gerados automaticamente os seguintes arquivos:

Jobname_Lin.dat e *Jobname_Non.dat* – Arquivos do modelo linear e não-linear respectivamente, que contêm comandos de geração do modelo (coordenadas dos nós e conectividades dos elementos), propriedades do material e condições de contorno entre outros (Figura 4.4).

Jobname_job_i.dat – Arquivo de entrada padrão usado para chamada do arquivo de carregamento e para posterior extração dos resultados.

A Figura 4.5 mostra os comandos contidos no arquivo *Jobname_job_i.dat* onde "i" representa o número do passo atual. Assim, antes de acionar o solver, o "script" lê o conteúdo deste arquivo, atualiza o número do passo atual do comando do ANSYS, e gera, em cada passo, um novo arquivo denominado *Jobname_job_i.dat*. É este novo arquivo que será usado como arquivo de entrada durante o acionamento do solver pelo "script".

```
/PREP7
...
/TITLE,IDTS3
!Coordenadas dos Nós
N, 1, -0.010969, 0.223811, -2.545021
...
N, 28547, 1.350055E-5, -0.2213, -3.75
! Definindo tipos de Elementos
ET, 1,45, 0, 0, 0, 0, 0, 0
! Propriedades do Material
E = 200000
MI = 0.3
...
TBPT,DEFI,0.15,809.5852
TBPLOT,MISO,1
!Propriedades e Conectividades dos Elementos
MAT,   1
TYPE,   1
EN, 1, 1, 2, 11, 14, 38, 39, 48, 50
...
EN, 19964,28384,28385,28388,28387,28543,28544,28547,28546
! Condicoes de contorno(FIX_X,FIX_Y,FIX_Z)
D,336,UX,0.,,342,1
...
D,13648,UZ,0.,,13697,1
FINISH
```

Figura 4.4 – Estrutura geral do arquivo *Jobname_non.dat*.

```
RESUME
!Realizando Análise do passo atual
/INPUT,C:\Test\Input\IDTS3_car_Noni,DAT
/POST1
/GRAFICS,FULL
!Recuperando Tensões
/OUTPUT,C:\IDTS3\Output\VM_Stress,DAT
PRESOL,S,PRIN
/OUTPUT
/EXIT,NOSAVE
```

expressão a ser alterada a cada passo

Figura 4.5 – Conteúdo dos arquivos *Jobname_job_i.dat*.

Jobname_car_Lin.dat e *Jobname_car_Noni.dat* – Arquivos de carregamento padrão usados para inserir o valor da pressão interna em cada passo de acordo com o incremento de carga determinado automaticamente pelo critério de convergência pré-definido pelo usuário. Assim, a cada i-ésimo passo, o "script" cria automaticamente um novo arquivo denominado *Jobname_car_Noni.dat* cujo valor da pressão interna estará atualizado para o passo atual a ser analisado. A Figura 4.6 mostra a estrutura geral dos comandos contidos no arquivo *Jobname_car_Noni.dat*.

```
/SOLUTION
ANTYPE,,REST
NEQIT,50
KBC,0
...
! Valor da Pressão Interna no passo atual
P_INT=6.06
TIME,P_INT
NSUBST,4
...
P_LONG=(P_INT*(PIPE_DI**2))/((PIPE_DE**2)-(PIPE_DI**2))
!Condicoes de Contorno (FIX_X,FIX_Y,FIX_Z)
D,336,UX,0.,,342,1
...
!Carregamento Pressão Interna (P_INT)
SFE,1353,2,PRES,0,P_INT
...
!Carregamento Pressão Longitudinal (P_LONG)
SFE,10683,6,PRES,0,-P_LONG
...
SAVE
! Executando Análise do passo atual
SOLVE
SAVE
FINISH
```

expressão a ser atualizada a cada passo

Figura 4.6 – Estrutura geral do arquivo *Jobname_car_Noni.dat*.

Ao final da etapa de pré-processamento, devem ter sido gerados os arquivos acima citados dentro do diretório de entrada conforme Figura 4.7.

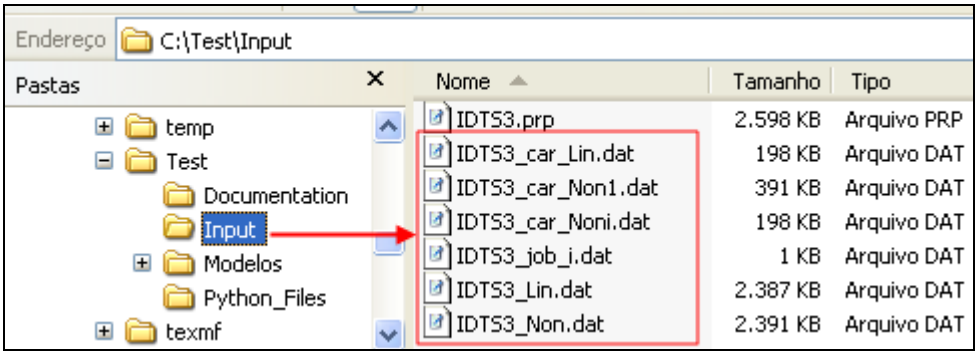


Figura 4.7 – Diretório *Input* onde são criados os arquivos utilizados no procedimento automático de análise não-linear.

Observe que o arquivo adicional *Jobname_car_Non1.dat* também foi gerado durante o pré-processamento. Ele possui basicamente o mesmo conteúdo do arquivo *Jobname_car_Noni.dat* exceto pelo fato de que nele são executadas sempre duas análises: a primeira análise é executada sempre com valor de pressão interna igual a zero. A segunda análise ("*Step1*") é executada com o valor da pressão interna dada pela equação 4.4 após a obtenção das tensões máximas durante a análise linear preliminar.

4.8.3 Análise Não-Linear

Toda análise é executada dentro de um diretório, por exemplo, "*<drive>:\Jobname*". A cada passo *i*, é gerado um novo subdiretório "*C:\Jobname\Step_i*" onde são gerados os

arquivos de entrada (*Jobname_job_i.dat*) e de carregamento (*Jobname_car_Noni.dat*), com os valores do passo e da pressão atualizados.

Após gerar os arquivos do passo atual, o "script" acessa o solver ANSYS por meio da função do PYTHON *"os.system(AnsysCommand)"*, o qual executa o comando em uma sub-shell. Este comando é expresso como "string" na seguinte forma:

```
AnsysCommand = Ansysdir+"-i"+Jobname_job_i_file+"-o" + Output_File_Name_i + "-j "+Jobname
```

Onde: `Ansysdir` (especifica o caminho do arquivo de inicialização do solver ANSYS); `Jobname_job_i_file` (especifica o caminho do arquivo a ser lido pelo ANSYS para o início da execução da análise); `Output_File_Name_i` (especifica o nome do arquivo onde serão armazenados os resultados da análise do ANSYS); `Jobname` (especifica o nome "prefixo" de todos os arquivos que serão gerados pelo ANSYS para o modelo específico).

A Figura 4.8 ilustra de forma simplificada o acesso do "script" ao solver e o processo de gerenciamento dos arquivos utilizados durante a execução da análise não-linear automática. Para cada passo da análise, o "script" acessa os arquivos padrão de modelagem e carregamento gerados na etapa de pré-processamento e em seguida gera novos arquivos para o passo atual com o valor de pressão interna devidamente atualizado em função dos critérios de incremento de carga pré-definidos.

Em seguida, o "script" aciona o solver ANSYS para execução do próximo passo da análise não-linear com os arquivos atualizados. Ao final da análise, o "script" lê o arquivo de resultados gerado pelo ANSYS durante a análise daquele passo e faz as operações de pós-processamento recuperando e armazenando os valores das tensões, deformações e número de iterações em cada passo. Estes valores são posteriormente avaliados para verificar os critérios de convergência e parada.

A cada passo, o "script" salva automaticamente as informações principais da análise em um arquivo "Resume.dat".

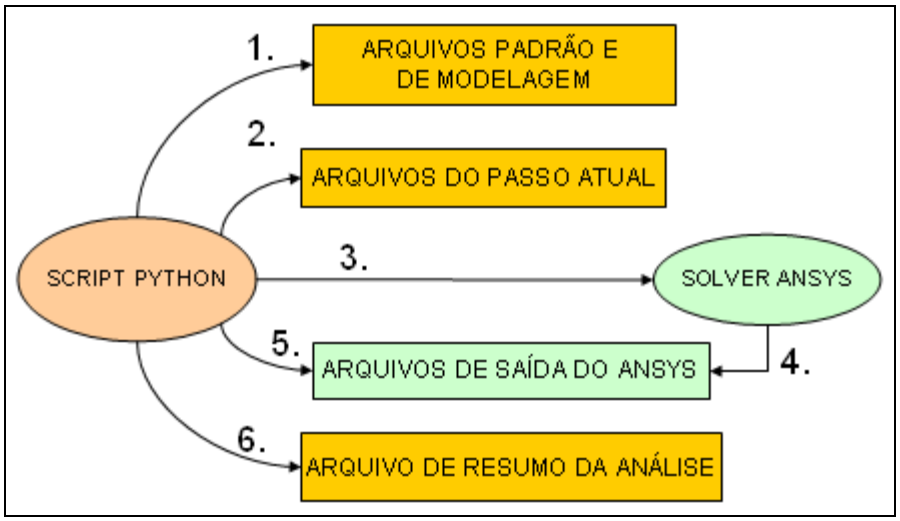


Figura 4.8 – Manipulação dos arquivos e acesso ao "solver" pelo "script".

4.8.4 Geração Automática do Histórico da Análise

No final da análise, o "script" acessa as informações contidas no arquivo "Resume.dat" e gera automaticamente um histórico da análise não-linear em um novo arquivo no formato Excel em cujas colunas são armazenados os seguintes dados: número do passo, pressão no passo, incremento de pressão, pressão no "substep", número de iterações no "substep", máxima tensão de Von Mises no passo e máximo incremento de deformação plástica no "substep". A Figura 4.9 mostra exemplo de parte de uma planilha Excel gerada automaticamente (via módulo PyExcelerator) com o resumo do histórico da análise. Em destaque o passo onde se iniciou o escoamento (passo 2), o passo onde o limite de deformação plástica (0,0025) foi ultrapassado (passo 17) e o passo onde o critério de ruptura foi atingido (passo 17a).

PASSO	PRESSÃO	ΔP	p	ITER.	σeqv.(MPa)	Max.Def.Plástica
0	0.00	0.00	0.0000	1	0.0000	0.0
			1.81060	8		0.0
			3.62120	8		0.0
			5.43180	8		0.0
1	7.2424	2.6823	7.24240	8	476.35	0.0
			7.91297	7		0.0
			8.58355	7		0.0
			9.25412	7		0.2580E-03
2	9.9247	2.6823	9.92470	7	578.53	0.4105E-03
			10.5953	7		0.4755E-03
			11.2658	7		0.5803E-03
			11.9364	6		0.6958E-03
3	12.6070	2.6823	12.6070	6	601.17	0.8845E-03
			24.7603	10		0.2530E-02
			24.8441	10		0.2620E-02
			24.9279	10		0.2687E-02
17	25.0117	0.3352	25.0117	10	777.17	0.2812E-02
			24.7184	9		0.1256E-02
			24.7603	9		0.1281E-02
			24.8022	10		0.1307E-02
17a	24.8441	0.1676	24.8441	10	774.07	0.1320E-02

Figura 4.9 – Exemplo de planilha Excel com o histórico da análise não-linear gerada automaticamente.

5 RESULTADOS

Este capítulo apresenta alguns exemplos de modelos gerados automaticamente por meio do programa PIPEFLAW desenvolvido durante este trabalho (ver capítulo 1). Tais exemplos buscam mostrar não somente a versatilidade e robustez do programa – em relação à variação dos diversos parâmetros do defeito - como também identificar situações geométricas extremas nas quais a geração automática começa a falhar. O capítulo apresenta ainda resultados provenientes das análises numéricas realizadas por meio do procedimento automático de análises não-lineares via "script" desenvolvido na linguagem PYTHON descrito no capítulo 4. Esses resultados numéricos servem como validação dos modelos analisados utilizando as ferramentas de automatização apresentadas neste trabalho.

5.1 Exemplos de modelos gerados automaticamente via PIPEFLAW

Usando-se as ferramentas de modelagem automática descritas no capítulo 1, pode-se gerar rapidamente vários modelos que podem ser utilizados em um eventual estudo paramétrico. A seguir são apresentados alguns modelos gerados automaticamente pelo programa PIPEFLAW.

A Figura 5.1 apresenta as principais dimensões de um defeito retangular.

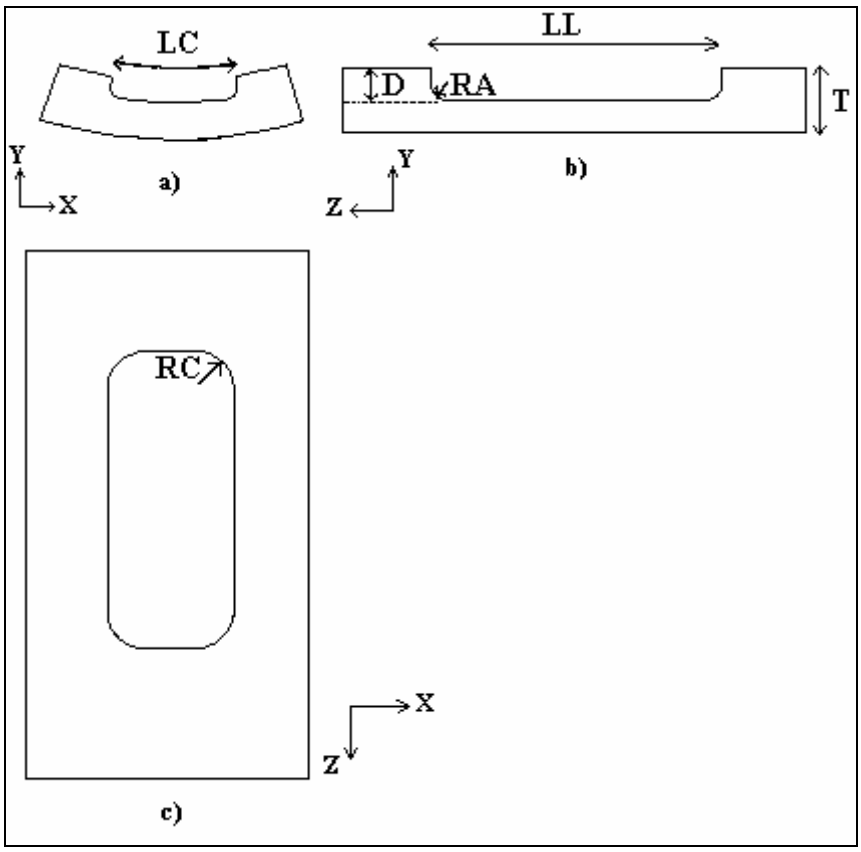


Figura 5.1 – Principais dimensões do defeito retangular.

Os parâmetros apresentados na Figura 5.1 são:
T = espessura íntegra do duto
D = profundidade do defeito
LL = comprimento na direção longitudinal do defeito
LC = comprimento na direção circunferencial do defeito
RA = raio de adoçamento
RC = raio de concordância

Pela Figura 5.1 pode-se constatar duas condições de restrição geométrica para variação da profundidade (D) do defeito. A primeira condição (e mais importante do ponto de vista estrutural) é a que relaciona o valor da profundidade do defeito com o valor da espessura do duto revelando assim o nível percentual de evolução da corrosão no duto. A segunda condição de restrição geométrica é que o valor do raio de adoçamento é sempre menor ou igual à profundidade do defeito ($RA \leq D$). Na situação limite, a parte reta da profundidade do defeito (D-RA) é igual a zero.

Nos exemplos mostrados a seguir, as dimensões do duto utilizadas foram: diâmetro externo (DE) igual a 480mm e espessura (T) igual a 9mm. O defeito mais raso que o programa conseguiu gerar sem apresentar problemas apresentava profundidade equivalente à 7% da espessura de parede do duto (D = 0,63mm) e raio de adoçamento (RA) igual a 0,27mm, conforme exemplo mostrado na Figura 5.2.

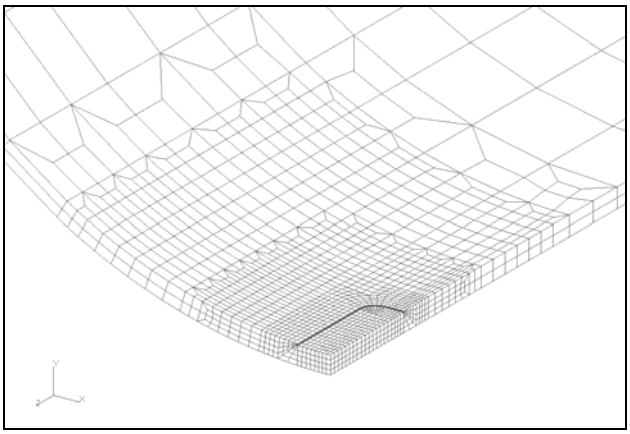


Figura 5.2 – Modelo de defeito muito raso (profundidade igual a 7% da espessura do duto).

O detalhe mais próximo do modelo pode ser visto na Figura 5.3 (região do defeito propriamente dito) e na Figura 5.4 (detalhe da região do raio de adoçamento).

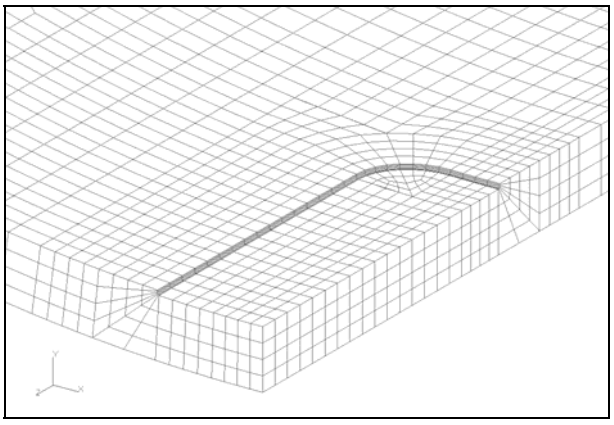


Figura 5.3 – Detalhe da região do defeito raso.

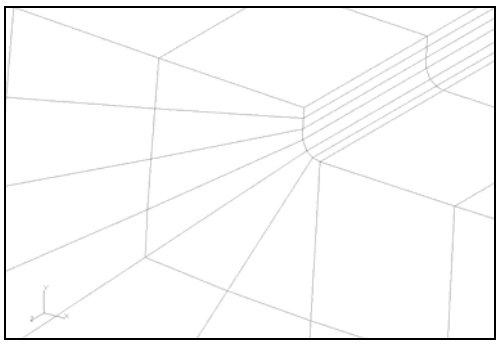


Figura 5.4 – Detalhe da região do raio de adoçamento do defeito raso.

Na situação limite oposta, o programa gerou modelos com profundidade de até 80% da espessura de parede do duto (D = 7,2mm) utilizando um raio de adoçamento de 3,5mm conforme mostrado na Figura 5.5.

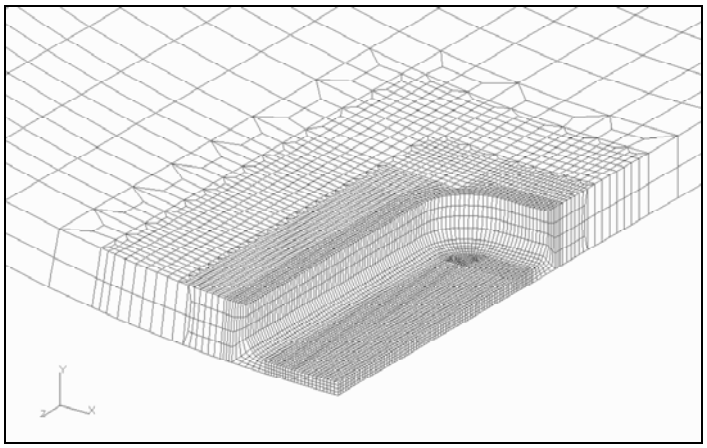


Figura 5.5 – Detalhe da malha de um defeito muito profundo (80% da espessura do duto).

A Figura 5.6 mostra detalhe da região próxima ao defeito e a Figura 5.7 mostra o detalhe da região dentro do defeito englobando o raio de adoçamento e o raio de concordância.

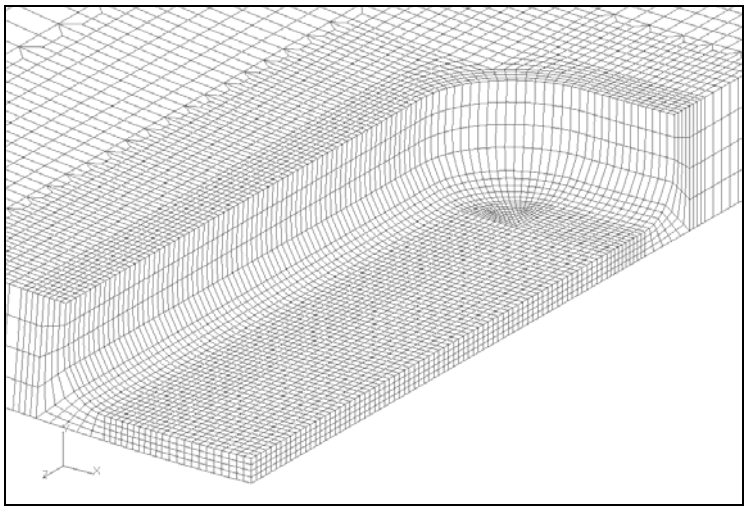


Figura 5.6 – Detalhe da região do defeito profundo.

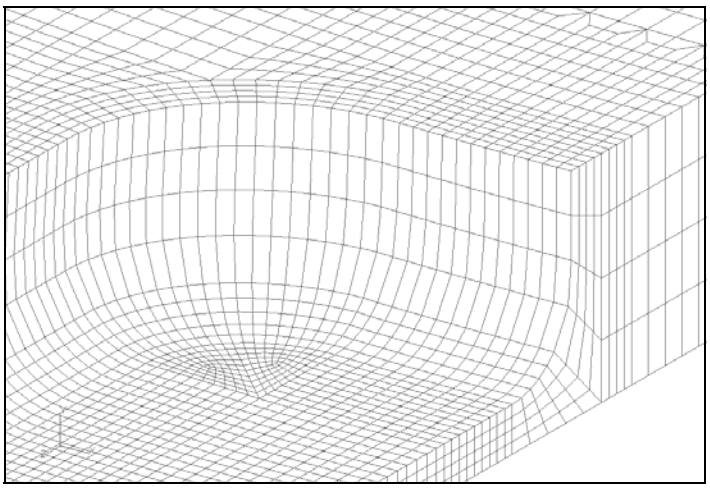


Figura 5.7 Detalhe da região englobando o raio de adoçamento e o raio de concordância.

Outra restrição geométrica é o fato de que o raio de concordância (RC) deve ser sempre maior ou igual ao valor do raio de adoçamento (RA). À medida que o valor de RC se aproxima do valor de RA, os sólidos na região central ("canto" do defeito) vão diminuindo de tamanho (ver sólidos 14 e 15 da Figura 3.46). Numa situação limite, onde RC = RA, estes sólidos deixam de existir. A Figura 5.8 mostra exemplo de defeito semelhante ao da Figura 5.6 com profundidade equivalente a 80% da espessura de parede do duto e raio de concordância igual a 8mm. No entanto, o exemplo mostrado na Figura 5.8 possui um raio de adoçamento de 6,8mm ao invés dos 3,5mm.

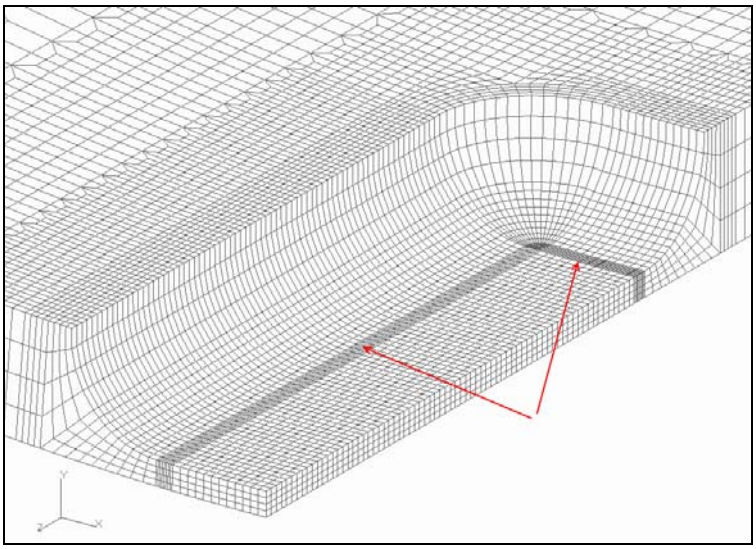


Figura 5.8 – Detalhe da região do defeito profundo com raio de adoçamento igual a 6,8mm.

Este último exemplo apresenta uma pequena região com grande concentração de elementos que se estende ao longo da direção longitudinal e circunferencial dentro do defeito conforme indicado nas setas em vermelho (Figura 5.8).

Uma visão mais próxima da malha e geometria na região central pode ser visto na Figura 5.9 com o contorno dos sólidos indicados na cor azul. Apesar de apresentar uma malha bastante densa, os elementos dessa região não apresentam problemas de distorção conforme detalhe da Figura 5.10.

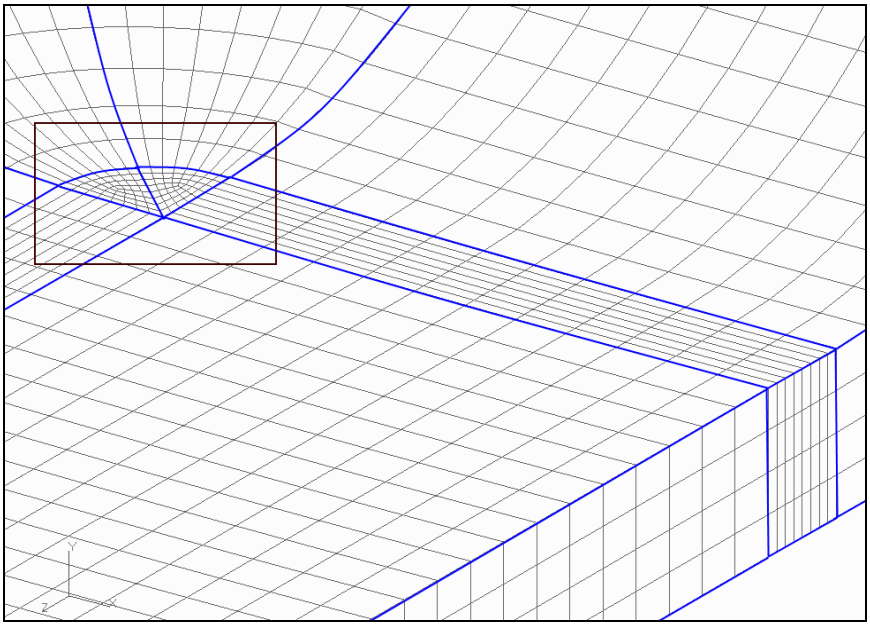


Figura 5.9 Malha e geometria nas regiões com grande densidade de elementos.

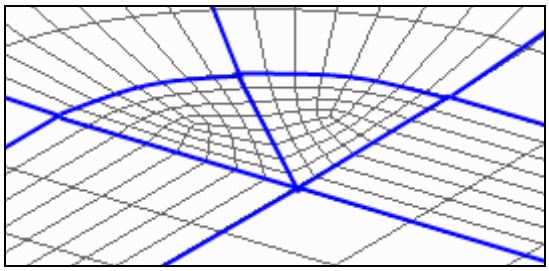


Figura 5.10 – Detalhe indicado na Figura 5.9.

Respeitadas as restrições geométricas, o programa PIPEFLAW gera defeitos de qualquer dimensão. Benjamin et al (2000) e Noronha Jr et al (2002) analisaram experimentalmente e numericamente, respectivamente, o comportamento de dutos na presença de defeitos longos de corrosão. A Figura 5.11 apresenta vista global de um dos espécimes tubulares analisados (espécime tubular "ET 4.2") com diâmetro externo de 323,85mm, espessura de parede de 9,74mm e comprimento total igual a 2,5m. O defeito contido neste modelo possui uma profundidade de 7,06mm (72,48% da espessura do duto), comprimento longitudinal de 527,8mm e comprimento circunferencial de 95,3mm. Utilizou-se um raio de adoçamento de 2,0mm e um raio de concordância de 8,0mm.

Na Figura 5.12 tem-se uma visão mais próxima do modelo na região do defeito englobando todas as transições de malha realizadas automaticamente pelo programa PIPEFLAW. O tempo computacional necessário para gerar um modelo varia conforme as dimensões do defeito. Quanto maior o comprimento e/ou a profundidade, maior será o tempo gasto para gerar o modelo. As etapas que mais consomem tempo computacional são as transições ao longo da superfície, pois elas exigem a criação de vários sólidos pequenos a cada dois ou três elementos (transições de 2 para 1 ou de 3 para 1) e o processo de geração de malha para cada sólido individual. Neste caso particular, utilizando-se uma máquina AMD Sempron 2600 (1,8GHz e 448MB RAM), foram necessários vinte e nove minutos para que o modelo "ET 4.2" fosse gerado completamente.

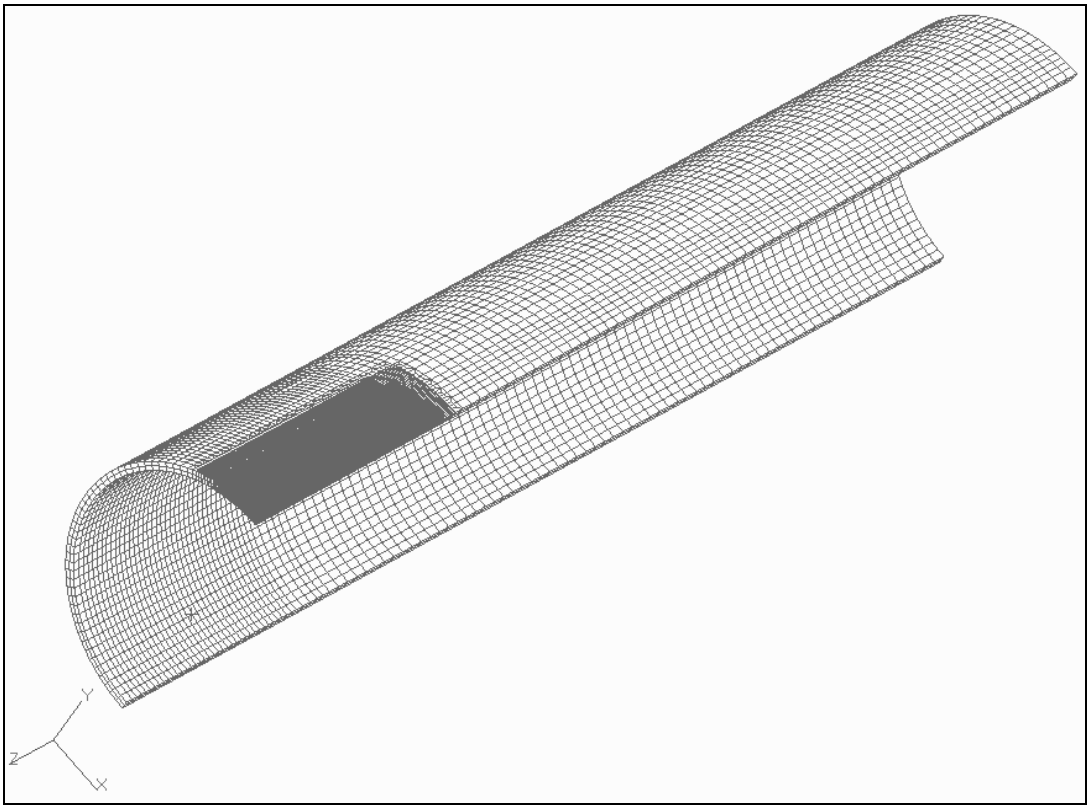


Figura 5.11 – Vista global do modelo “ET 4.2” gerado automaticamente pelo PIPEFLAW.

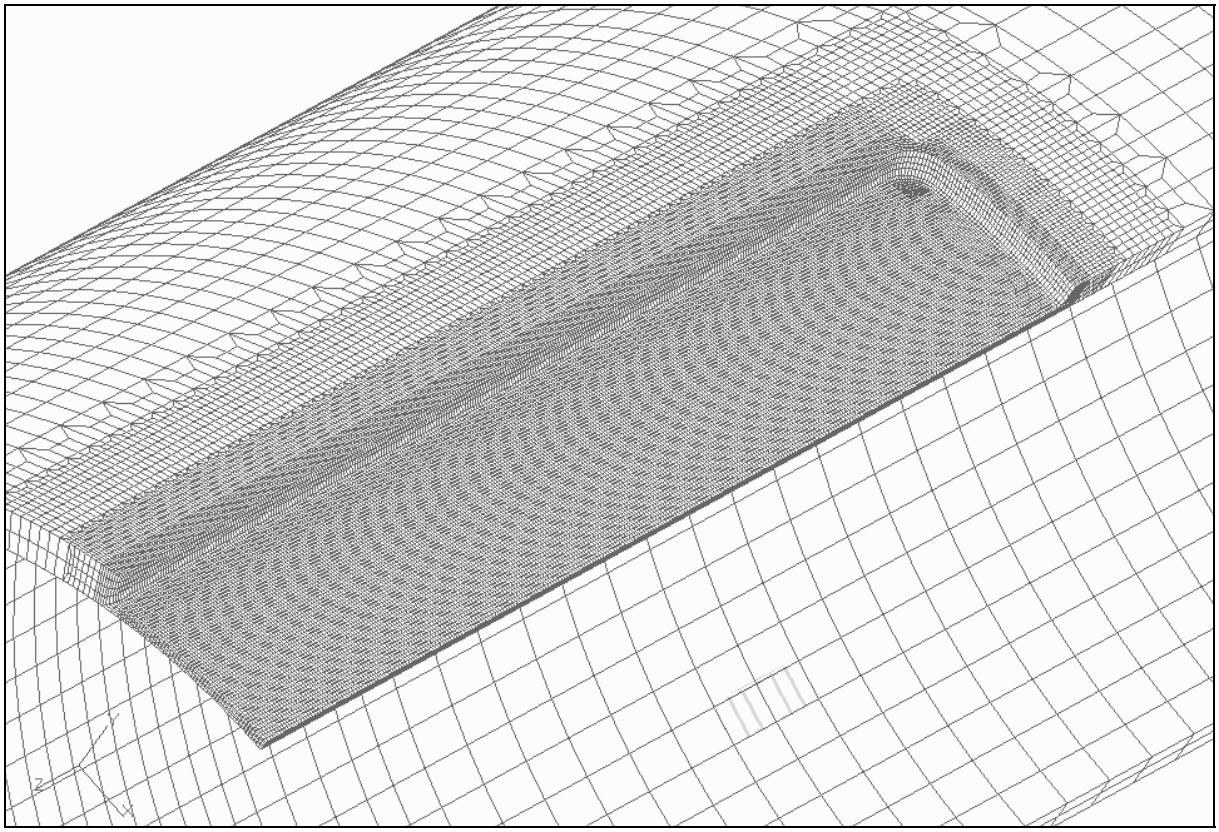


Figura 5.12 – Vista aproximada da região do defeito do modelo “ET 4.2”.

Um outro exemplo de defeito longo, porém com orientação circunferencial, é mostrado na Figura 5.13.

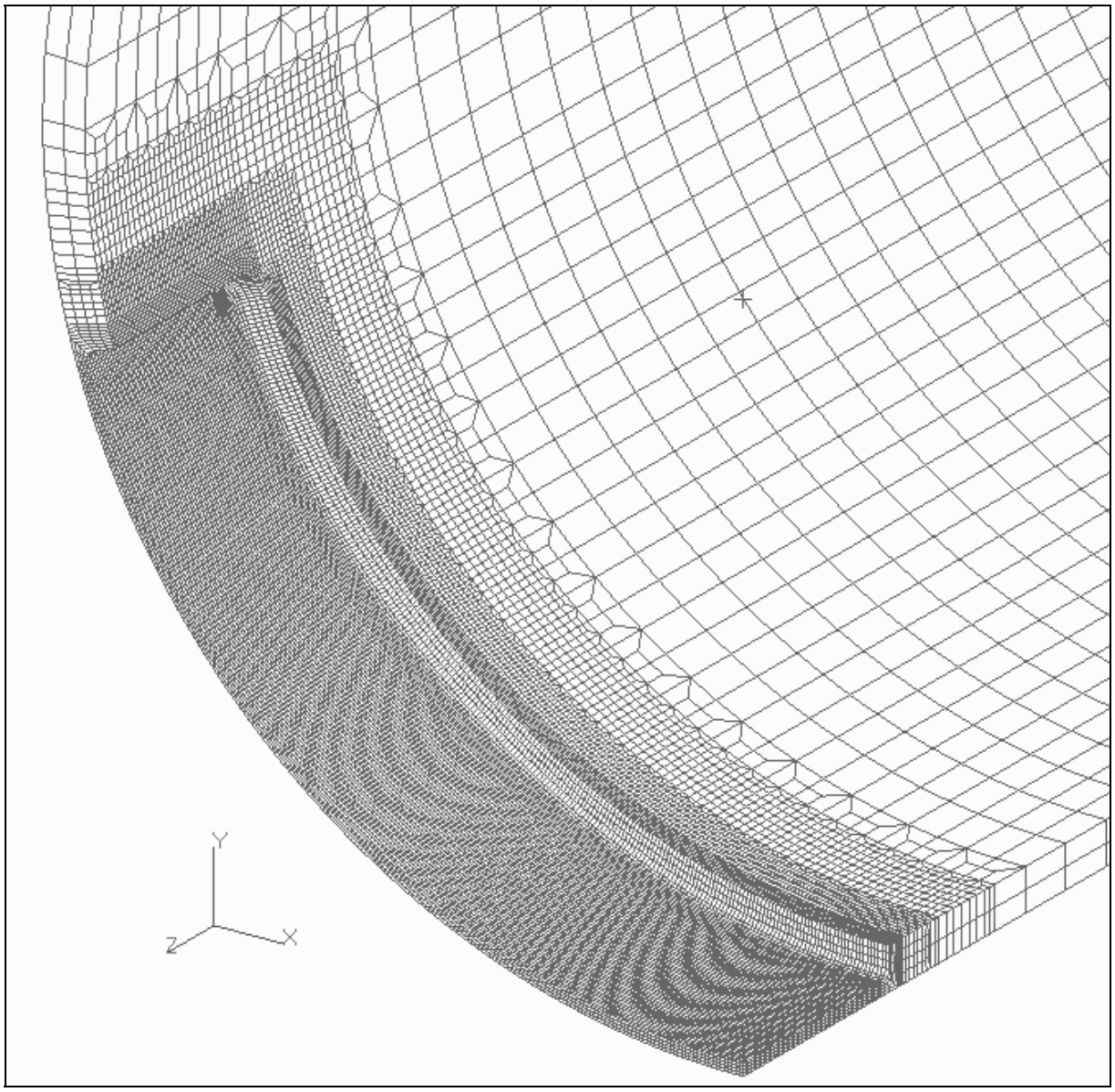


Figura 5.13 – Exemplo de defeito longo orientado na direção circunferencial.

5.2 Validação das Ferramentas Desenvolvidas

Nesta seção são apresentados alguns resultados de simulações numéricas que foram realizadas utilizando as ferramentas propostas neste trabalho. Realizou-se aqui a simulação numérica em três dos seis espécimes tubulares que foram ensaiados experimentalmente por Benjamin et al (2005) e simulados numericamente por Andrade et al (2006). Uma breve descrição desses trabalhos foi feita anteriormente na seção 2.7.1 (Defeitos Artificiais de Corrosão). A Figura 5.14 apresenta a configuração dos defeitos retangulares contidos nos três espécimes tubulares (IDTS2, IDTS3 e IDTS4) que serão aqui simulados. O modelo IDTS2 apresenta um defeito simples, o modelo IDTS3 contém dois defeitos alinhados longitudinalmente e o modelo IDTS4 contém dois defeitos alinhados circunferencialmente.

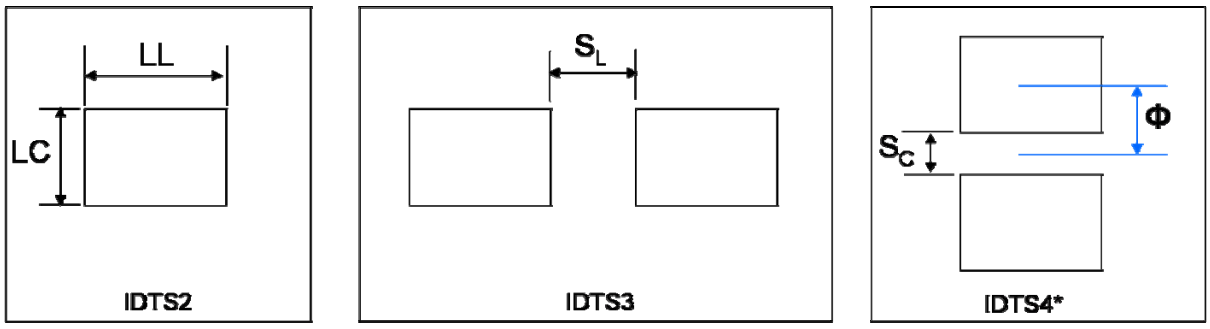


Figura 5.14 – Configuração dos defeitos analisados experimentalmente e numericamente por Benjamin et al (2005) e Andrade et al (2006).

Os defeitos foram usinados por eletroerosão assumindo a forma retangular na superfície externa do duto e as suas principais dimensões são apresentadas na Tabela 5.1 (ver Figura 5.1 e Figura 5.14).

Tabela 5.1 – Dimensões reais dos defeitos usinados nos três espécimes tubulares.[36]

Modelo	LL [mm]	LC [mm]	D [mm]	RA [mm]	RC** [mm]	S_L [mm]	S_C [mm]	S_C^* [mm]	$\frac{D}{T}\cdot 100[\%]$
IDTS2	39,6	31,9	5,39	3,5	8,0	-	-	-	66,5%
IDTS3	39,6	31,9	5,32	3,5	8,0	20,5	-	-	65,7%
IDTS4*	39,6	32,0	5,62	3,5	8,0	-	9,9	11,65	69,4%
DE = 458,8 mm (diâmetro externo do duto)									
T = 8,1 mm (espessura íntegra do duto)									

Nota: * → Espaçamento mínimo utilizado no PIPEFLAW.
** → Valor real não conhecido.
S_L e S_C = espaçamento longitudinal e circunferencial, respectivamente.

O programa PIPEFLAW conseguiu gerar automaticamente os modelos IDTS2 e IDTS3. No entanto, o modelo IDTS4 não pode ser gerado com as dimensões reais, pois os defeitos contidos neste modelo estavam muito próximos (S_C = 9,9 mm ou Φ = 5,23º), onde Φ é o valor de entrada utilizado no PIPEFLAW e representa o ângulo que define o centro do defeito, tal como ilustrado na Figura 3.38. Conforme visto na seção 3.5.4, a distância mínima entre centros de defeitos adjacentes que o programa PIPEFLAW permite gerar (nível 0) equivale à distância de duas vezes o comprimento da caixa fixa do defeito (defeitos alinhados longitudinalmente) (ver Figura 3.61 e Figura 3.66) ou duas vezes o ângulo da caixa fixa "Ang_Box_Defect" (defeitos alinhados circunferencialmente). No programa PIPEFLAW, a distância angular Φ equivalente à distância real de 9,9mm é de 5,23º enquanto que o ângulo usado para o modelo IDTS4 foi de 5,45º (equivalente à distância de S_C^* = 11,65mm). Portanto, o modelo aqui simulado difere desses valores e por isso foi denominado de IDTS4*.

As fotos dos defeitos dos espécimes IDTS2, IDTS3 e IDTS4 obtidas nos ensaios experimentais realizados por Benjamin et al (2005) logo após a ruptura, são apresentadas na Figura 5.15. As linhas na cor vermelha indicam a região onde a falha ocorreu. Nos modelos com múltiplos defeitos (IDTS3 e IDTS4) a falha foi individual (falha englobando apenas um defeito).

[36] Fonte: Adaptado de Benjamin et al (2005) apud Andrade et al (2006).

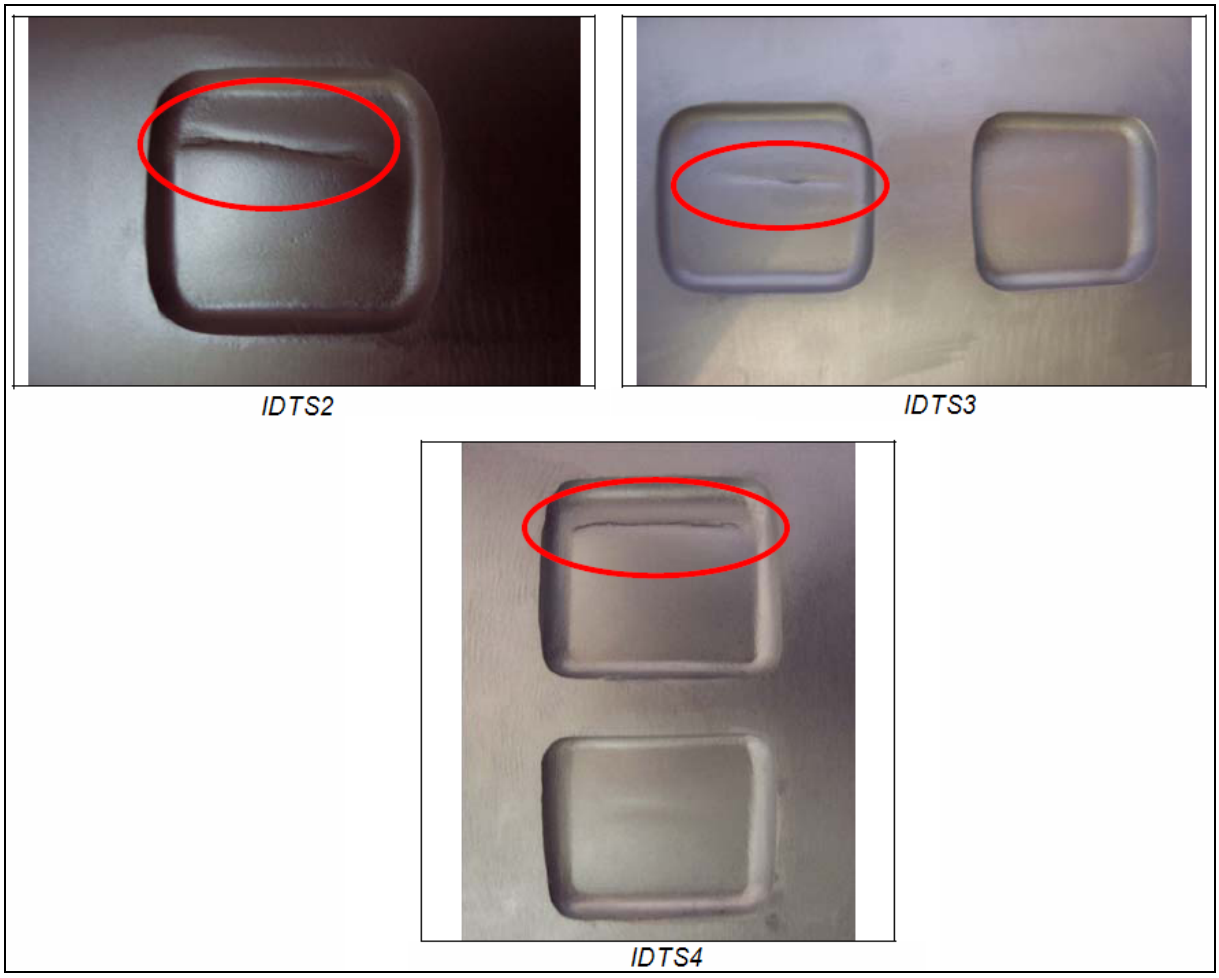


Figura 5.15 – Fotos dos defeitos dos espécimes IDTS2, IDTS3 e IDTS4 após a ruptura.[37]

A Tabela 5.2 apresenta as pressões de falha medidas experimentalmente em laboratório (Benjamin et al, 2005), as obtidas numericamente via MEF (Andrade et al, 2006), as obtidas numericamente via MEF (utilizando as ferramentas desenvolvidas neste trabalho) e as pressões de falha obtidas via método semi-empírico (BS-7910). Os erros das pressões falhas estimadas (numericamente e via método semi-empírico) em relação à pressão de falha experimental (Pf_{EXP}) também estão indicados na Tabela 5.2.

Tabela 5.2 – Pressão de falha experimental *versus* pressões de falha estimadas.

	Pressões de Falha [MPa]				Erro[1] (%)		
	Experimental	Numérico (MEF)		Empírico	Numérico		Empírico
Espécime	$Pf_{(EXP)}$	Pf_1 (Andrade et al, 2006)	Pf_2 (este trabalho)	Pf_3 ($_{BS7910}$)	(Andrade et al, 2006)	(este trabalho)	(BS7910)
IDTS2	22,679	22,710	22,0894	21,253	+0,14	-2,60	-6,29
IDTS3	20,314	19,535	19,5095	18,511	-3,83	-3,96	-8,88
IDTS4	21,138	22,298	21,5374*	20,944	+5,49	+1,89*	-0,92
Erro médio[2]	-	-	-	-	3,15	2,82	5,36

Notas: * – Valores associados ao modelo IDTS4* utilizado neste trabalho.
1 – Erro(%) = [(Pf_i - Pf_{EXP}) / (Pf_{EXP})] . 100% (i = 1, 2 e 3)
2 – Erro médio = ∑ | Erro | /3

[37] Fonte: Adaptado de Benjamin et al (2005) apud Andrade et al (2006).

Pode-se observar que os erros numéricos obtidos por Andrade et al (2006) na estimativa da pressão de ruptura ficaram dentro da faixa de -3,83% e +5,49%. Resultados semelhantes foram obtidos neste trabalho onde os erros numéricos ficaram dentro da faixa de -3,96% e +1,89%. A diferença entre os erros obtidos por Andrade et al (2006) e os obtidos neste trabalho deve-se principalmente ao fato de que, para os três modelos analisados neste trabalho (IDTS2, IDTS3 e IDTS4*), a análise foi interrompida devido ao critério de parada ter sido atingido pois o incremento de pressão ΔP_i ficou menor que o valor mínimo pré-estabelecido ($\Delta P_i < 0{,}01\text{MPa}$). Isto significa que para todos os modelos, a análise foi interrompida quando os maiores níveis de tensão (von Mises) ainda estavam abaixo do valor da tensão de ruptura do material ($\sigma_{rup} = 718{,}2MPa$) (ver Tabela 5.3). Os valores da pressão de falha, estimados via método semi-empírico (norma BS7910), apresentaram valores mais conservadores em relação aos obtidos via MEF para todos os espécimes. Para os espécimes IDTS2 (defeito simples) e IDTS3 (dois defeitos alinhados longitudinalmente), os erros ficaram entre -6,29% e -8,88%, respectivamente, comprovando assim o conservadorismo embutido nos métodos semi-empíricos. Para o espécime IDTS4 (dois defeitos alinhados circunferencialmente) a norma apresentou um erro igual a -0,92%. O menor erro apresentado pela norma para o modelo IDTS4 se deve ao fato de que defeitos alinhados circunferencialmente são avaliados como um único defeito, pois eles se superpõem após a aplicação da técnica de projeção no plano longitudinal, resultando assim em uma avaliação menos conservadora.

Tabela 5.3 – Valores de pressão e tensão no instante em que as análises foram interrompidas.

Modelo	Ruptura Numérica		
	Passo de carga	Pressão [MPa]	Máxima Tensão von Mises [MPa]
IDTS2	21	22,0894	717,48
IDTS3	21c	19,5095	717,37
IDTS4*	21b	21,5374	714,58

A Tabela 5.4 mostra em detalhe as principais partes do resumo do histórico da análise não-linear do modelo IDTS3, gerado automaticamente pelo "script" de automatização da análise. Em destaque o passo onde se iniciou o escoamento (passo 2), o passo onde o limite de deformação plástica (0,0025) foi ultrapassado (passo 8), o passo onde não houve convergência (passo 21b) e o passo onde o critério de parada foi atingido (passo 21c). Os históricos completos das análises não-lineares, gerados automaticamente pelo "script" para os modelos IDTS2, IDTS3 e IDTS4*, podem ser vistos nos apêndices **A**, **B** e **C** respectivamente.

A Tabela 5.5 apresenta o tempo computacional despendido para realização das análises não-lineares via MEF dos modelos IDTS2, IDTS3 e IDTS4* utilizando diferentes configurações de computadores. Para cada modelo é apresentado o número total de nós e elementos da malha, as principais características do computador utilizado, o número total de passo executados na análise (incluso o número de re-análises), o tempo total aproximado para execução da análise e o tempo médio para execução de cada passo.

Conforme pode se verificar, quanto maior o tamanho da malha do modelo, maior é o tempo computacional gasto para executar as análises. Se estas análises fossem realizadas manualmente, utilizando o procedimento padrão do CENPES (via recurso "Save/Restart" do ANSYS), o tempo despendido seria maior pois seria necessário o monitoramento manual do usuário durante toda a análise. Por exemplo, para o modelo IDTS3, o usuário teria que monitorar toda a análise, e, a cada 1,48 horas (aproximadamente), teria que coletar e interpretar manualmente os resultados obtidos durante aquele passo para depois aplicar os critérios de incremento de carga e reinicializar o próximo passo da análise.

Tabela 5.4 – Parte do resumo do histórico da análise não-linear do modelo IDTS3.

LOAD STEP	P	ΔP	p	ITER.	σeqv.(MPa)	Max.Def.Plástica
0	0.00	0.00	0.0000	1	**0.0000**	**0.0**
			4.42450	6		**0.0**
			4.84190	6		**0.0**
			5.25930	6		**0.0**
2	5.6767	1.6696	5.67670	6	**534.52**	**0.1095E-03**
			14.4421	6		**0.1811E-02**
			14.8595	7		**0.2055E-02**
			15.2769	7		**0.2351E-02**
8	15.6943	1.6696	15.6943	8	**647.32**	**0.2711E-02**
			14.2334	6		**0.8994E-03**
			14.4421	6		**0.9316E-03**
			14.6508	7		**0.9930E-03**
8a	14.8595	0.8348	14.8595	7	**633.67**	**0.1060E-02**
			19.5062	6		**0.1200E-03**
			19.5095	6		**0.1202E-03**
			19.51275	50		
21b	19.5160	0.0130	19.5160		**Non Conv.**	
			19.5046	5		**0.5999E-04**
			19.5062	5		**0.6002E-04**
			19.5079	5		**0.6007E-04**
21c	19.5095	0.0065	19.5095	9	**717.37**	**0.6013E-04**

Tabela 5.5 – Tempo computacional para realização das análises não-lineares.

Modelo	Malha	Computador	Número total De passos da análise	Tempo total [horas]	Tempo médio por passo [horas]
IDTS2	23.462 nós 15.948 elementos	AMD Athlon (tm)64 2.0 GHz 960 MB RAM	30 (8 re-análises)	16	0,53
IDTS3	28.547 nós 19.964 elementos	Intel Pentium 4 2.4GHz 1022MB RAM	29 (8 re-análises)	43	1,48
IDTS4*	31.601 nós 22.222 elementos	Intel Pentium 4 3.06 GHz (2CPUs) 1022MB RAM	29 (8 re-análises)	53	1,83

O ganho de tempo e eficiência obtido utilizando-se as ferramentas de automatização das análises não-lineares eliminou a necessidade de se ter um engenheiro repetindo um procedimento mecânico a cada passo, permitindo que o mesmo se concentre em avaliar a confiabilidade e validade dos resultados. No entanto, observa-se ainda assim um elevado tempo computacional, da ordem de dias, para a realização das análises dos modelos utilizando o recurso "Save/Restart" do ANSYS.

A limitação do tempo computacional necessário para realizar as análises não-lineares usando o recurso "Save/Restart" do ANSYS, serviu de motivação para investigação de outras técnicas de análises que pudessem reduzir o tempo computacional das análises.

Para investigar outras técnicas, os modelos IDTS2, IDTS3 e IDTS4* foram reanalisados utilizando-se o recurso "Automatic Time Step" disponível no ANSYS. Conforme visto anteriormente, este recurso ativa um esquema automático de forma a garantir que a variação do incremento de carga não seja nem muito grande (o que resulta em muitas bisseções e reanálises) e nem também muito conservador (o que resulta em um incremento de carga muito pequeno). Além disso, utilizou-se também o comando "SOLCONTROL,ON" que ativa as configurações automáticas do solver que fornecem algoritmos confiáveis e eficientes para resolução de problemas estruturais não-lineares. Em outras palavras, este comando permite que o ANSYS escolha as configurações padrões ótimas e algoritmos internos avançados para resolução de problemas não-lineares. A escolha dos algoritmos ótimos é baseada em vários fatores, entre eles, as características físicas (não-lineares) que envolvem o problema.

A Tabela 5.6 apresenta os resultados das simulações dos modelos IDTS2, IDTS3 e IDTS4* utilizando os recursos de incremento automático de carga ("Automatic Time Step") e os recursos automáticos para resolução de problemas não-lineares ("Solcontrol"). Conforme esperado, a diferença entre os erros obtidos por Andrade et al (2006) e os obtidos neste trabalho diminuíram pois as análises prosseguiram sem ser interrompidas até o momento em que as tensões de von Mises ao longo da direção radial (todos os pontos situados ao longo da espessura do duto) atingiram o valor igual ou superior à tensão última verdadeira do material. Conseqüentemente, os modelos aqui reanalisados (Tabela 5.6) atingiram pressões de falha um pouco acima dos valores anteriores (Tabela 5.3).

A Tabela 5.7 apresenta os tempos computacionais despendidos para realização das análises não-lineares via MEF dos modelos IDTS2, IDTS3 e IDTS4* utilizando o recurso "Save/Restart" e os recursos automáticos "Automatic Time Step" e "Solcontrol". As simulações não-lineares utilizando as ferramentas automáticas do ANSYS resultaram em uma considerável redução no tempo computacional despendido para realização das análises utilizando os mesmos computadores. As reduções dos tempos de análises ficaram entre 87,5% (modelo IDTS2) e 93,4% (modelo IDTS4*).

Tabela 5.6 – Pressão de falha experimental *versus* pressões de falha estimadas: Simulações realizadas utilizando os recursos "Automatic Time Step" e "Solcontrol".

	Pressões de Falha [MPa]				Erro[1] (%)		
	Experimental	Numérico (MEF)		Empírico	Numérico		Empírico
Espécime	$Pf_{(EXP)}$	Pf_1 (Andrade et al, 2006)	Pf_2 (este trabalho)	Pf_3 $(_{BS7910})$	(Andrade et al, 2006)	(este trabalho)	(BS7910)
IDTS2	22,679	22,710	22,791	21,253	+0,14	+0,49	-6,29
IDTS3	20,314	19,535	19,810	18,511	-3,83	-2,48	-8,88
IDTS4	21,138	22,298	22,403*	20,944	+5,49	+5,98*	-0,92
Erro médio[2]	-	-	-	-	3,15	2,98	5,36

Notas: * – Valores associados ao modelo IDTS4* utilizado neste trabalho.
1 – $Erro(\%) = [(Pf_i - Pf_{EXP}) / (Pf_{EXP})] . 100\%$ (i = 1, 2 e 3)
2 – Erro médio = $\sum | Erro | /3$

Tabela 5.7 – Comparação entre o tempo computacional para realização das análises não-lineares usando o recurso "Save/Restart" e os recursos automáticos disponíveis no ANSYS.

Modelo	Tempo total [horas]	
	"Save/Restart"	"Automatic Time Step"
IDTS2	16	2
IDTS3	43	2,7
IDTS4*	53	3,5

As figuras 5.16, 5.17 e 5.18 apresentam a distribuição de tensões de von Mises na região dos defeitos dos modelos IDTS2, IDTS3 e IDTS4*, respectivamente. Estas figuras apresentam a distribuição de tensão no instante da pressão final de cada modelo juntamente com a configuração deformada (utilizou-se um fator de escala igual a 5 para a configuração deformada). Comparando as figuras 5.16, 5.17 e 5.18 com a figura contendo as fotos do ensaio experimental (Figura 5.15) pode-se observar que a configuração de falha dos espécimes IDTS2 e IDTS4*, determinada por meio da simulação numérica via MEF, foi muito próxima da configuração de falha real observada nos experimentos em laboratório, permitindo representar de forma bastante satisfatória a configuração deformada dos defeitos e detectar de forma precisa o local onde a falha ocorreu.

O modelo IDTS3 (com dois defeitos alinhados longitudinalmente) apresentou os maiores níveis de tensão justamente na região entre os defeitos (Figura 5.17) o que caracteriza uma configuração de falha um pouco diferente do que foi observado nos experimentos em laboratório onde a falha ocorreu dentro de um defeito (Figura 5.15). No entanto, o mapa de tensões do modelo IDTS3 indica que a região onde ocorreu a falha também está sujeita a altos níveis de tensão (segundo nível mais alto de tensão, representado pela cor amarela na escala de tensões).

O modelo IDTS4* (com dois defeitos alinhados circunferencialmente) apresentou os maiores níveis de tensão na região mais externa dos defeitos (Figura 5.18), confirmando assim os resultados observados em laboratório (Figura 5.15).

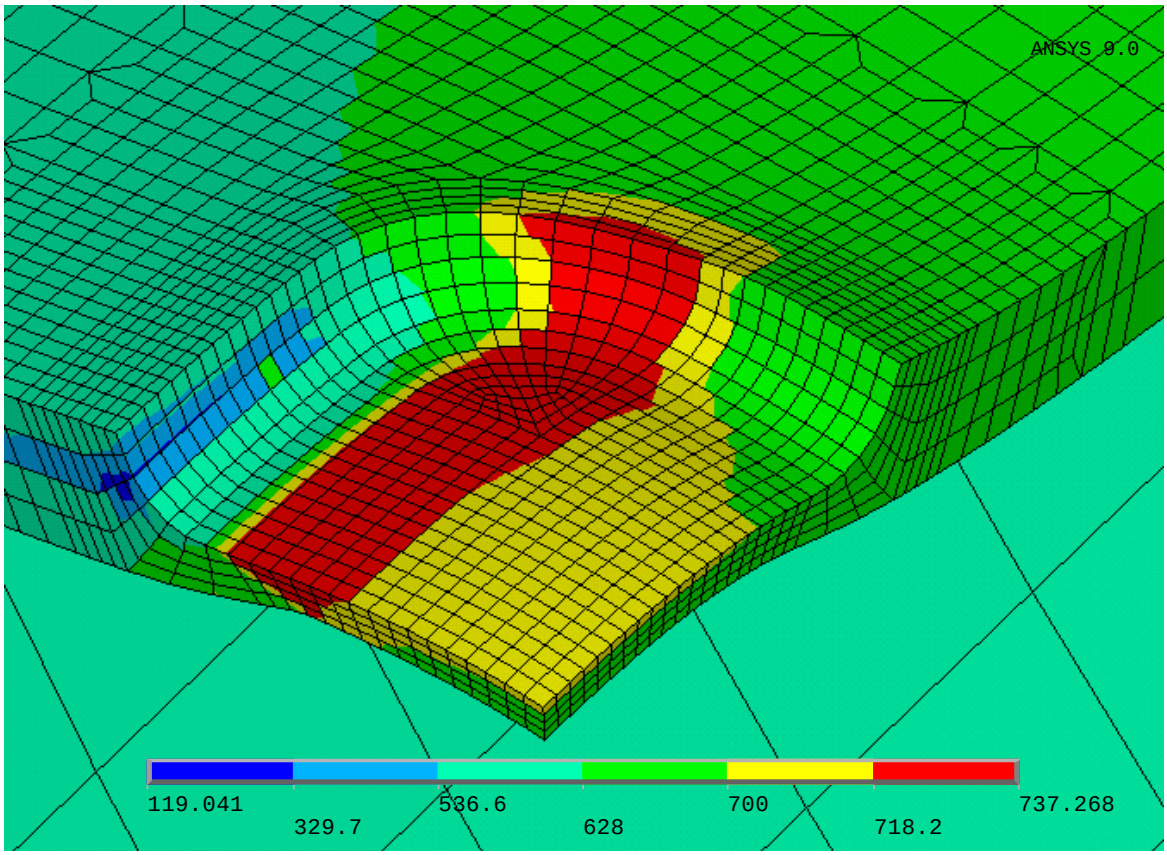


Figura 5.16 – Detalhe da distribuição de tensão na região do defeito (modelo IDST2).

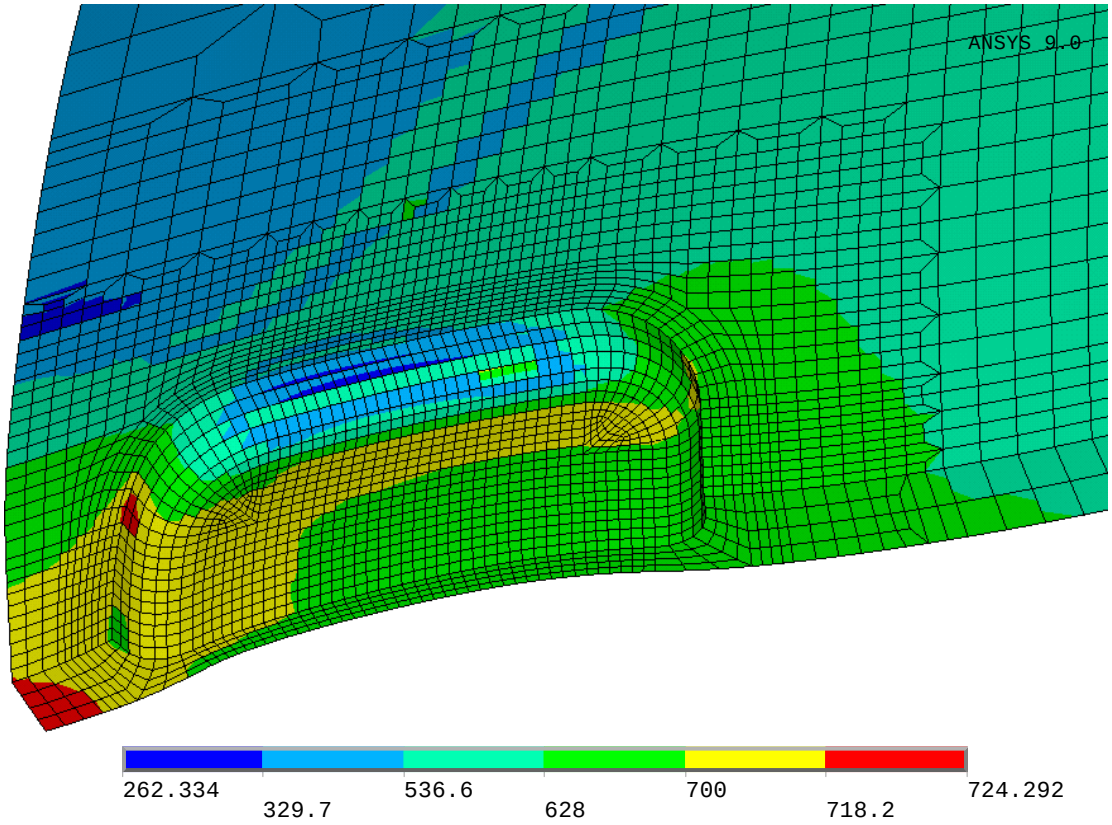


Figura 5.17 – Detalhe da distribuição de tensão na região dos defeitos (modelo IDTS3).

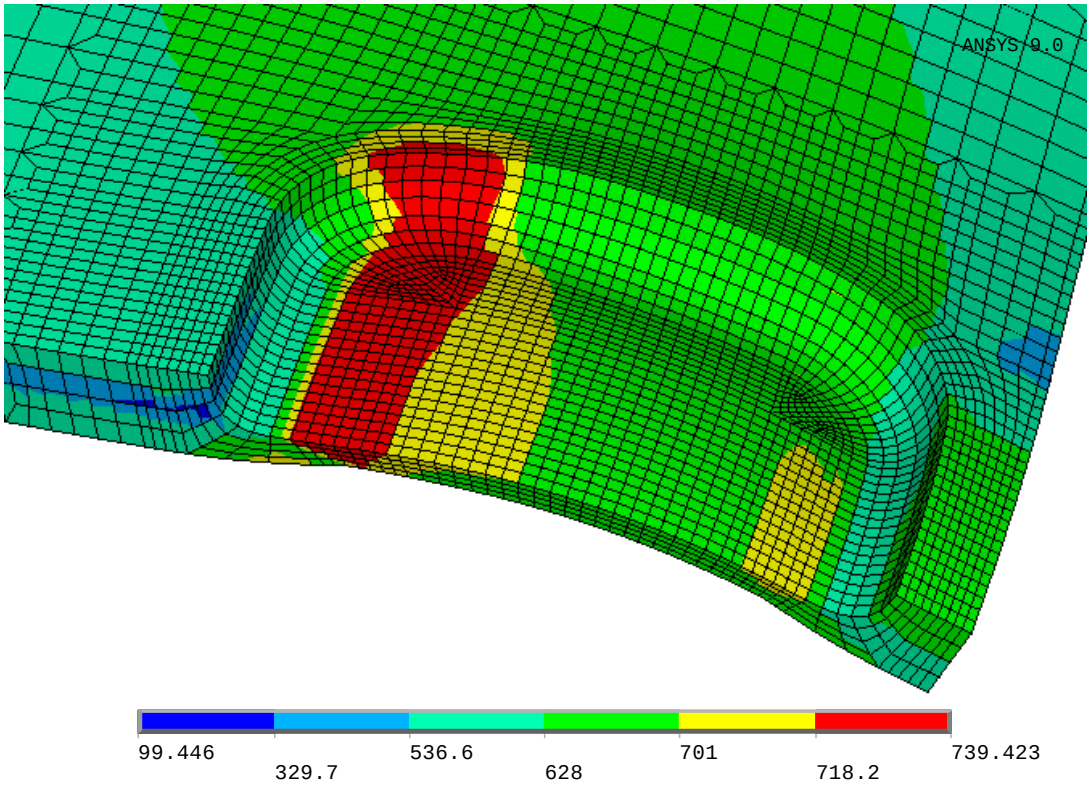


Figura 5.18 – Detalhe da distribuição de tensão na região dos defeitos (modelo IDTS4*).

A Figura 5.19 apresenta os resultados das variações das tensões de von Mises em função dos valores de incremento de pressão interna para os três modelos simulados (IDTS2, IDTS3 e IDTS4*). Estes resultados foram extraídos do ponto localizado na superfície externa do duto na região do defeito sujeita aos maiores níveis de tensão. Conforme sugerido no apêndice G da norma BS7910, a curva representando as variações de tensões com o aumento da pressão interna (para os três modelos aqui apresentados) indica três estágios distintos. O primeiro estágio é uma relação linear entre tensão e pressão interna aplicada. Este estágio segue até o ponto onde o limite elástico é atingido (σ_y = 534,1MPa). No segundo estágio, as máximas tensões de von Mises permanecem constantes ou aumentam ligeiramente até que toda a espessura remanescente do duto plastifique. O terceiro estágio é dominado pelo fenômeno do endurecimento do material e segue até o ponto onde ocorre a falha.

A curva do modelo com defeito isolado (IDTS2) praticamente se superpõe com a curva do modelo com dois defeitos alinhados circunferencialmente (IDTS4*). Apesar das dimensões dos defeitos contidos nestes dois modelos serem levemente diferentes, os resultados sugerem que não há interação entre defeitos alinhados circunferencialmente. A pequena diferença existente entre as curvas destes dois modelos deve-se ao fato de que o defeito contido no modelo IDTS4* é levemente mais profundo do que o do modelo IDTS2 e por isso, a pressão final de falha do modelo IDTS4* atinge um valor ligeiramente menor que o do modelo IDTS2.

A curva do modelo com dois defeitos alinhados longitudinalmente (IDTS3) está bastante deslocada para esquerda em relação às outras duas curvas indicando assim a predominância do efeito da interação entre defeitos adjacentes. Estes resultados sugerem que o efeito da interação entre defeitos alinhados na direção longitudinal resulta em uma considerável redução na resistência residual de dutos corroídos.

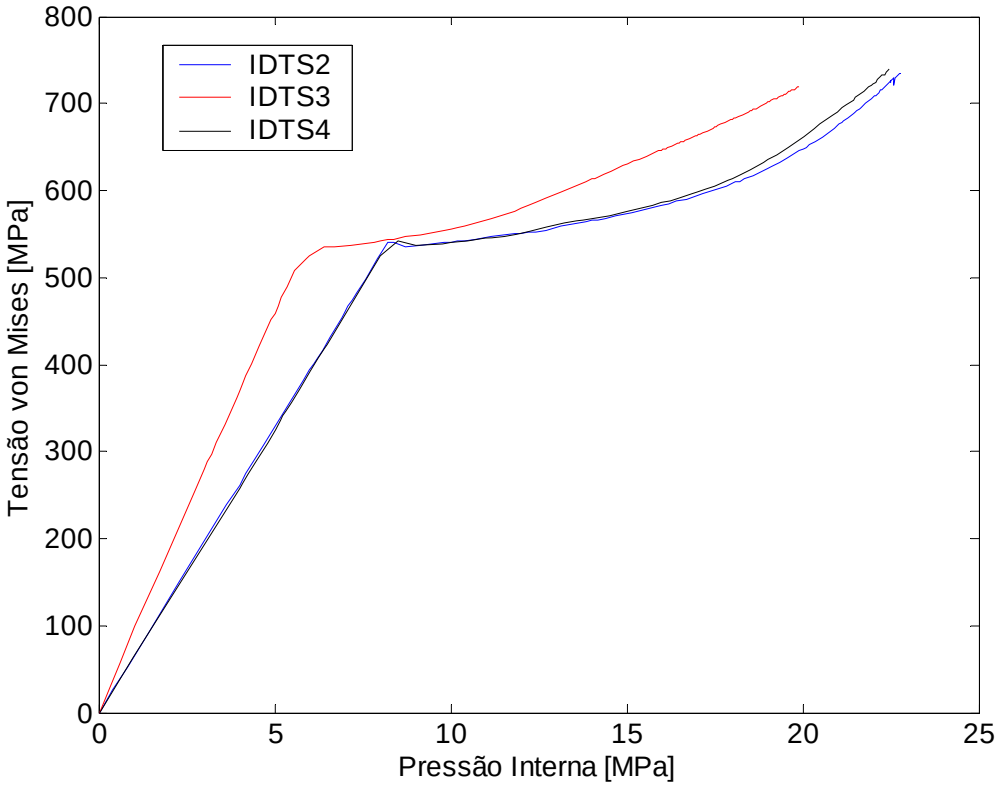


Figura 5.19 – Variação das tensões (von Mises) em função da pressão interna aplicada para os três modelos analisados.

Os resultados das variações dos deslocamentos na direção y em função da pressão interna aplicada também estão apresentados na Figura 5.20 para os três modelos simulados (IDTS2, IDTS3 e IDTS4*). Todas as três curvas foram geradas a partir de dados extraídos do ponto que obteve o maior valor de deslocamento y, localizado na região do defeito. Comparando a Figura 5.20 com a Figura 5.19, pode-se observar um comportamento linear dos deslocamentos em função da pressão interna aplicada. Esta resposta linear segue até aproximadamente o ponto onde o terceiro estágio da Figura 5.19 é iniciado. A partir deste ponto, os resultados dos deslocamentos apresentam um comportamento assintótico crescendo rapidamente com o aumento da pressão interna até o momento onde a pressão de falha é atingida.

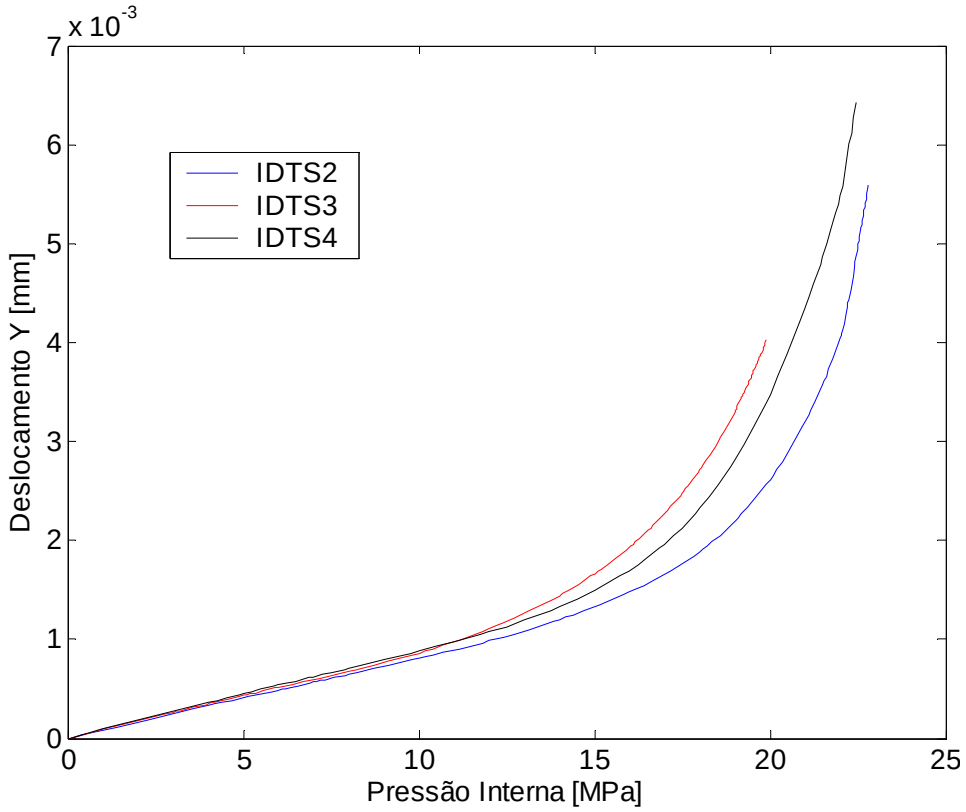


Figura 5.20 – Variação dos deslocamentos na direção y em função da pressão interna aplicada para os três modelos analisados.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Principais Contribuições

O principal objetivo deste trabalho foi desenvolver um conjunto de ferramentas computacionais confiáveis e robustas para a modelagem e análise automática de defeitos de corrosão em dutos através do método dos elementos finitos. As ferramentas para modelagem automática (programa PIPEFLAW) foram desenvolvidas por meio da linguagem PCL (Patran Command Language) e foram integradas no software comercial de pré e pós-processamento MSC.PATRAN por meio de interface gráfica personalizada. As ferramentas para automatização da análise foram desenvolvidas via linguagem Python. As principais contribuições de cada uma dessas ferramentas são descritas logo a seguir.

6.1.1 Ferramentas de Modelagem Automática

O programa PIPEFLAW possui um conjunto de funções implementadas na linguagem PCL para geração automática de modelos de dutos com defeitos de corrosão com geometria retangular localizados na superfície interna ou externa do duto. Os defeitos podem assumir a configuração de defeito simples (um defeito isolado no duto) ou de múltiplos defeitos alinhados longitudinalmente ou circunferencialmente ao longo do duto.

Os modelos gerados automaticamente pelo programa PIPEFLAW contêm malhas de elementos finitos discretizadas com elementos finitos sólidos hexaédricos e já possuem o padrão de discretização utilizado pelo CENPES/PETROBRÁS. As regiões em torno do defeito apresentam diferentes níveis de refinamento de malha de forma que à medida que vão se aproximando do defeito elas vão assumindo um maior nível de refinamento para melhor capturar os gradientes de tensão naquelas regiões.

6.1.2 Ferramentas de Interface Gráfica

As ferramentas de modelagem automática do programa PIPEFLAW possuem interface gráfica personalizada que permitem que o usuário forneça os dados necessários para a modelagem de forma amigável e intuitiva. A interface gráfica foi adicionada na janela principal do PATRAN a partir de um “menu” onde são criados vários elementos gráficos entre eles janelas, botões, “switchs”, caixas de diálogo para captura de dados, tabelas com células e ícones ilustrativos que direcionam o usuário ao longo do processo de entrada de dados.

6.1.3 Ferramentas de Automatização da Análise

O procedimento padrão de análises não-lineares adotado pelo CENPES/PETROBRÁS foi automatizado por meio de um “script”, implementado na linguagem Python, a partir do qual toda a análise é gerenciada por meio da execução automática de tarefas pré-determinadas. A utilização desse “script” possibilita que os critérios de convergência e incremento de carga, pré-definidos pelo usuário, sejam aplicados automaticamente, diferentemente do procedimento usual quando se ativa os critérios de convergência e de incremento de carga determinados pelo “solver”. O ganho de tempo e eficiência obtido utilizando-se as ferramentas de automatização das análises não-lineares eliminou a necessidade de se ter um engenheiro repetindo um procedimento mecânico a cada passo, permitindo que o mesmo se concentre em avaliar a confiabilidade e validade dos resultados.

6.2 Conclusões

6.2.1 Ferramentas de Modelagem Automática

As ferramentas de geração automática de modelos de dutos com defeitos mostraram-se bastante satisfatórias para a modelagem de defeitos retangulares localizados tanto na superfície interna quanto na superfície externa do duto.

A criação das superfícies via o método "glide" foi fundamental para a geração de sólidos triparamétricos sem precisar utilizar operações "booleanas" (do tipo sólido/sólido ou superfície/sólido) para decomposição do domínio. Isto garantiu uma maior confiabilidade das ferramentas de geração automática aqui desenvolvidas, pois estas operações "booleanas" são muito propensas a erros.

Os exemplos apresentados mostraram algumas situações críticas na modelagem de defeitos retangulares e provou o sucesso e versatilidade do programa PIPEFLAW.

As ferramentas foram aplicadas com sucesso nos modelos aqui analisados, com exceção do modelo IDTS4, onde o limite de proximidade entre defeitos do programa PIPEFLAW foi ultrapassado.

6.2.2 Ferramentas de Interface Gráfica

A interface gráfica personalizada, desenvolvida especialmente para este tipo de aplicação, facilitou bastante o trabalho de modelagem permitindo que usuários com o mínimo conhecimento em simulação via MEF pudesse realizar a modelagem de defeitos de corrosão em dutos de forma bastante intuitiva e amigável.

O uso de interface gráfica evita que o usuário precise alterar o valor de variáveis contidas nos códigos fonte das funções implementadas durante o processo de entrada de dados para início da modelagem.

6.2.3 Ferramentas de Automatização da Análise

As ferramentas de automatização das análises não-lineares, via "script" implementado na linguagem Python, foram aplicadas com sucesso nos modelos aqui apresentados.

O ganho de tempo e eficiência obtido utilizando-se as ferramentas de automatização das análises não-lineares eliminou a necessidade de se ter um engenheiro repetindo um procedimento mecânico a cada passo, permitindo que o mesmo se concentre em avaliar a confiabilidade e validade dos resultados.

No entanto, observou-se um grande tempo computacional para a realização de análises não-lineares para os modelos mais complexos envolvendo múltiplos defeitos. Em função disso, utilizou-se recursos automáticos padrões disponíveis no ANSYS e verificou-se uma considerável redução do custo computacional sugerindo assim a utilização deste tipo de procedimento para análise de modelos mais complexos.

6.3 Trabalhos em Andamento

O trabalho aqui apresentado representa parte dos resultados obtidos no projeto de pesquisa "Geração Automática de Defeitos de Corrosão em Dutos - MOSINEIP" da Rede Norte-Nordeste de Pesquisa Cooperativa em Modelagem Computacional (RPCMOD-Rede9) que está sendo desenvolvido pelo grupo de Processamento de Alto Desempenho na Mecância Computacional (PADMEC) cujos membros pertencem aos Departamentos de Engenharia Mecânica e Engenharia Civil da UFPE.

Este projeto ainda está em execução e possui vários alunos em nível de graduação (Iniciação Científica), mestrado e doutorado que estão contribuindo para a extensão das ferra-

mentas aqui apresentadas. Os principais trabalhos em andamento que estão sendo desenvolvidos neste projeto são:

- Geração automática de múltiplos defeitos em posição arbitrária no duto.

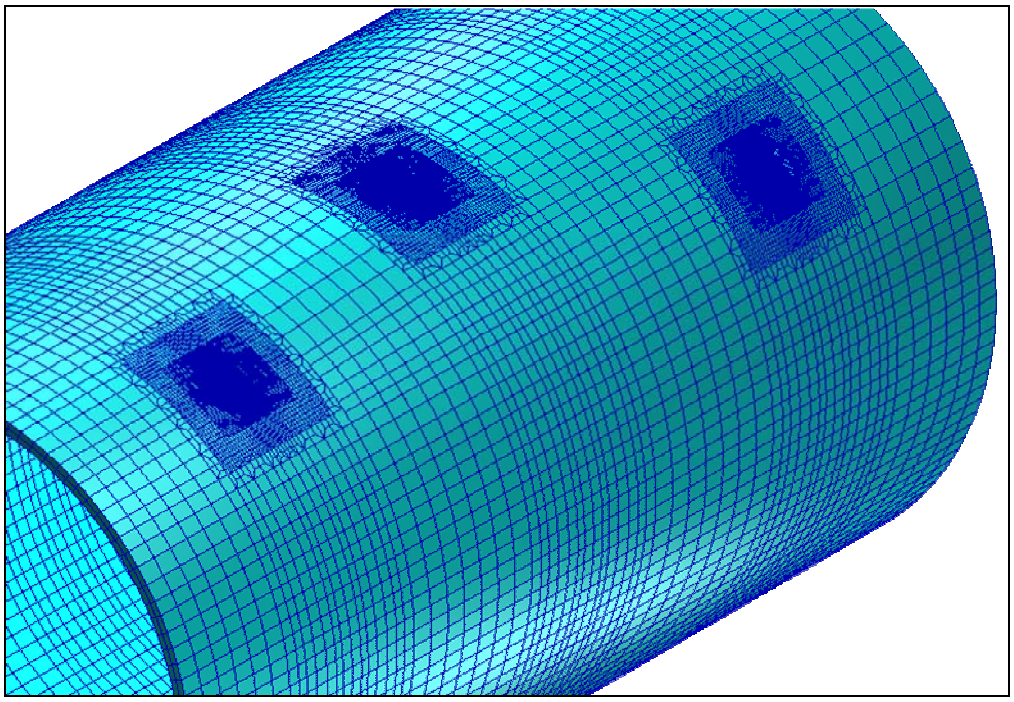


Figura 6.1 – Exemplo de defeitos em posição arbitrária na superfície do duto.

- Geração automática de defeitos com geometria elíptica.

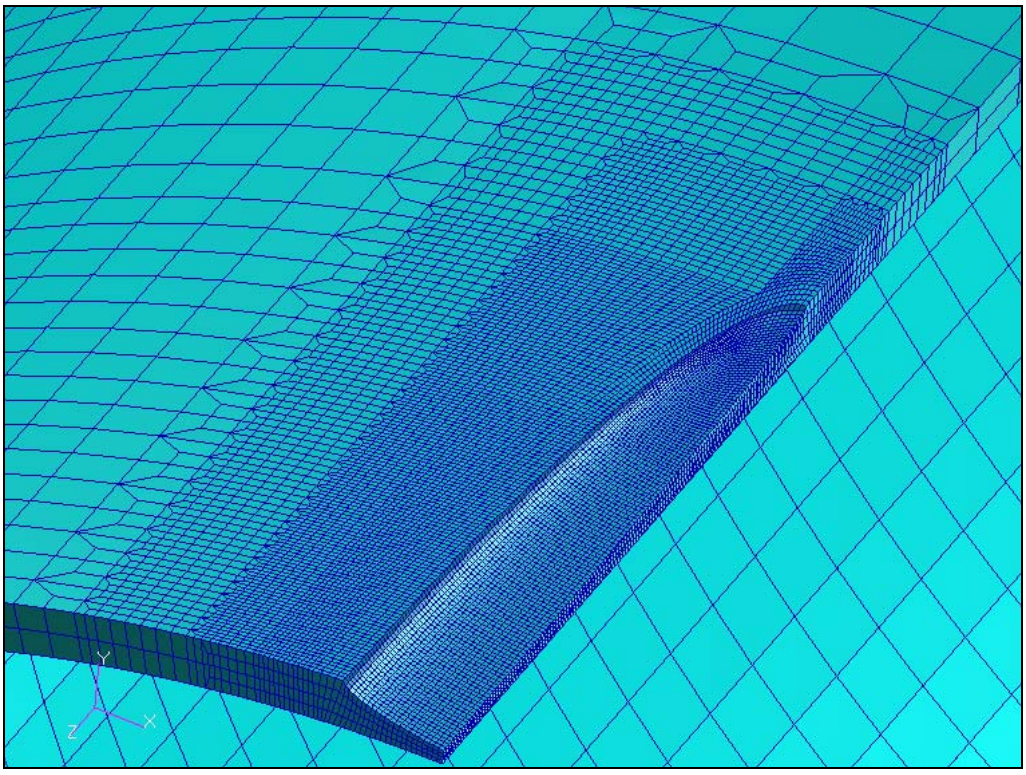


Figura 6.2 – Exemplo de defeito com geometria elíptica.

- Geração de defeitos superpostos (malha não-estruturada).
- Interface gráfica final integrando os módulos de geração de defeitos elípticos e retangulares.
- Automatização da visualização dos resultados.

6.4 Sugestões para Trabalhos Futuros

O trabalho aqui apresentado possui várias possibilidades para futuros melhoramentos e extensões. A seguir, segue um resumo de algumas sugestões para trabalhos posteriores.

6.4.1 Ferramentas de Modelagem Automática

- Considerar defeitos com orientação arbitrária.

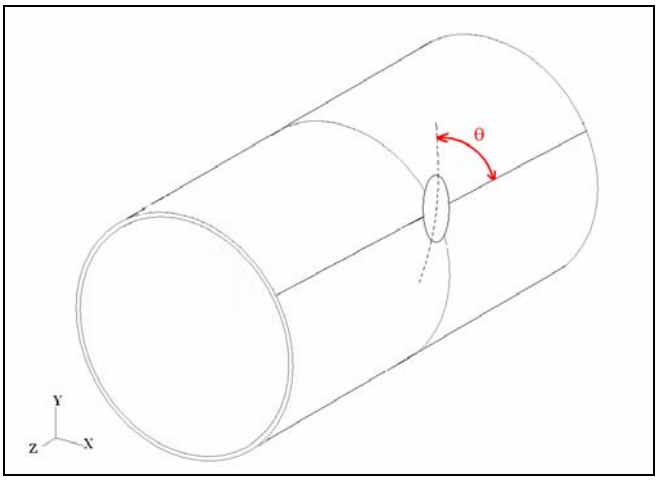


Figura 6.3 – Exemplo de defeito com orientação arbitrária.

- Considerar defeitos com profundidade variável.

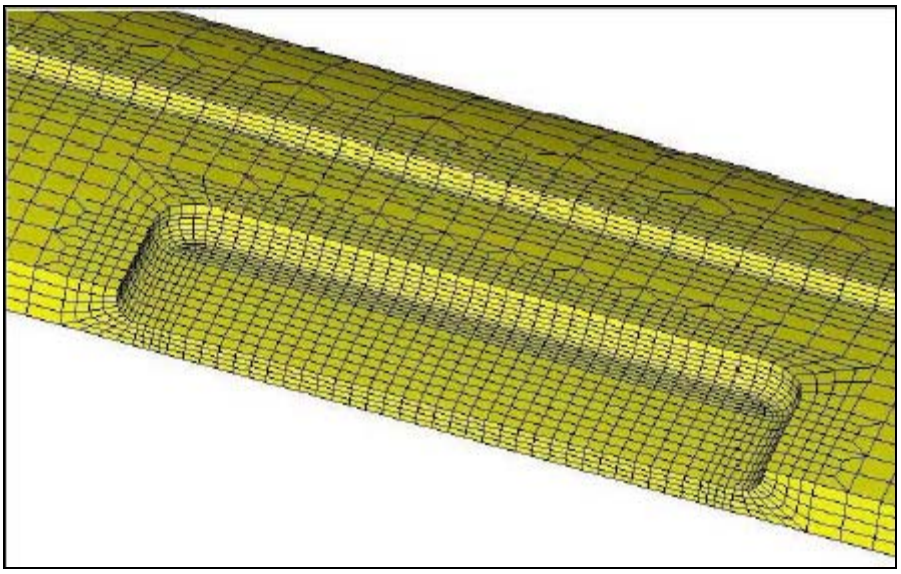


Figura 6.4 – Exemplo de defeito retangular com profundidade variável. Fonte: Benjamin & Andrade (2003b).

- Considerar defeitos com perfil complexo

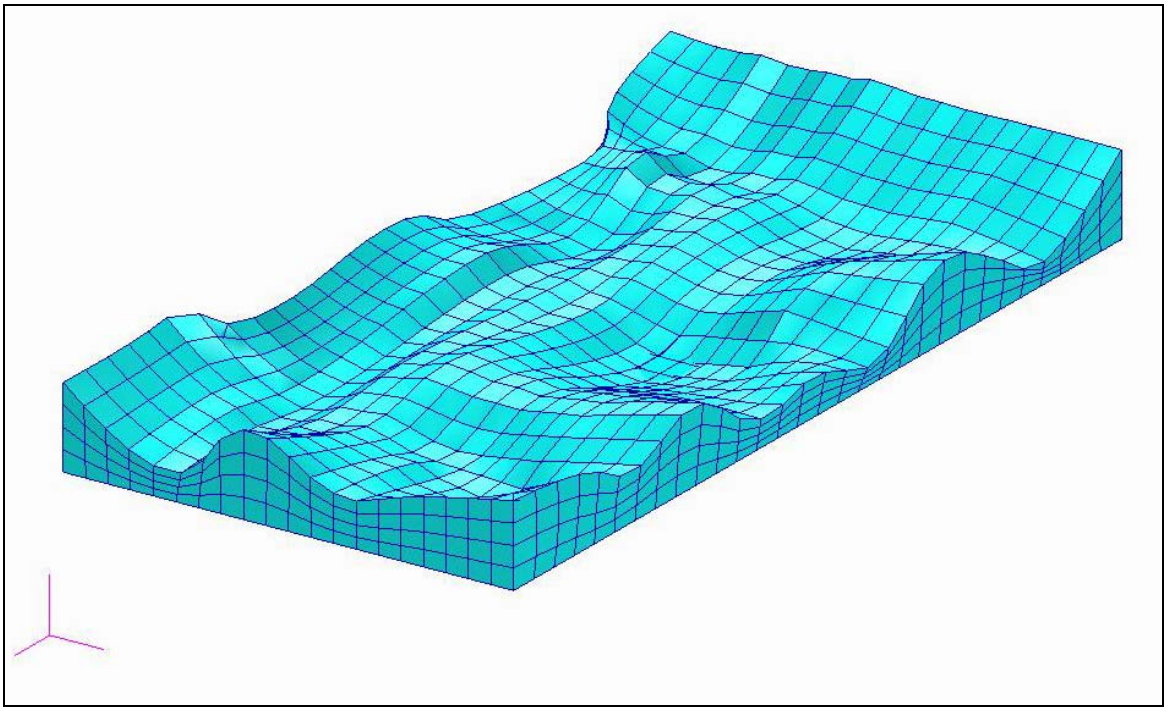


Figura 6.5 – Exemplo de defeito com perfil complexo.

6.4.2 Ferramentas de interface gráfica

- Implementar interface gráfica para o "script" de automatização das análises não-lineares via módulo Tkinter do Python ou qualquer outra linguagem como, por exemplo, Delphi.

6.4.3 Estudo paramétrico

- Realizar estudo paramétrico de defeitos simples e múltiplos. Este tipo de estudo é fundamental e se torna muito mais fácil por meio do uso das ferramentas automáticas aqui desenvolvidas.

6.4.4 Novas técnicas para redução do tempo computacional das análises.

- Utilizar malhas mistas com elementos mais simples nas regiões distantes do defeito.
- Utilizar malhas híbridas (estruturadas e não-estruturadas) com adaptação automática.
- Utilizar superelemento.
- Utilizar outros critérios automáticos de avanço do incremento de carga para acelerar a análise.
- Utilizar computação paralela para realizar as análises no ANSYS.

REFERÊNCIAS

Adib-Ramezani, H.; Jeong, J.; Pluvinage, G., 2006. *Structural Integrity Evaluation of X52 Gas Pipes Subjected to External Corrosion Defects Using SINTAP Procedure*. International Journal of Pressure Vessels and Piping, v.83, Issue 6, pp.420-432.

Alves, J.L., 2002. *Avaliação Numérica da Capacidade de Carga de Dutos Corroídos*. Pós-Graduação em Engenharia Civil, PUC-Rio, Rio de Janeiro, Dissertação de Mestrado, 89p.

Andrade, E.Q.; Benjamin, A.C.; Machado Jr., P.R.S.; Pereira, L.C.; Jacob, B.P.; Carneiro, E.G.; Guerreiro, J.N.C.; Silva, R.C.C.; Noronha Jr., D.B., 2006. *Finite Element Modeling of the Behavior of Pipelines Containing Interacting Corrosion Defects*. 25th International Conference on Offshore Mechanics and Arctic Engineering (OMAE2006-92600), Hamburg, Germany.

Anon, 1999. *Guide on Methods for Assessing the Acceptability of Flaws in Fusion Welded Structures*. BS 7910, Incorporating Amendment No.1, British Standards Institution, London,UK.

ANSYS, 2004. Ansys Release 9.0 Documentation: Operations Guide(Chapter 3) and Structural Guide (Chapter 8). http://www.ansys.com/products/multiphysics.asp

API, 2000. American Petroleum Institute. *API 5L: Specification for line Pipe*, 42ª ed., Washington, 153p.

Bathe, K.J., 1996. *Finite Elements Procedures*. Editora Prentice-Hall, pp.485-640.

Batte, A.D.; Fu, B.; Kirkwood, M.G.; Vu, D., 1997. "*Advanced Methods for Integrity Assessment of Corroded Pipelines*", Pipes & Pipelines International, pp. 5-11, January-February.

Benjamin, A.C. & Andrade, E.Q., 2003. IBP413-03 *Modified Method for the Assessment of the Remaining Strength of Corroded Pipelines*, Rio Pipeline Conference & Exposition, pp. 13.

Benjamin, A.C. & Andrade, E.Q., 2003. *Predicting the Failure Pressure of Pipelines Containing Nonuniform Depth Corrosion Defects Using the Finite Element Method*. 22nd International Conference on Offshore Mechanics and Arctic Engineering (OMAE2003-37072).

Benjamin, A.C. & Andrade, E.Q., 2005. *Projeto 601295* "*Avaliação de Dutos Corroídos com Defeitos Curtos*" Especificação 13 (Revisão 2): Procedimento para Definição da Estratégia de Aplicação do Carregamento em Análises de Ruptura de Espécimes Tubulares com Defeitos Curtos de Corrosão usando Elementos Finitos.

Benjamin, A.C.; Freire, J.L.F.; Vieira, R.D.; Castro, J.T.P., 2002. *Burst Tests on Pipeline with Nonuniform Depth Corrosion Defects.* 21st International Conference on Offshore Mechanics and Arctic Engineering (OMAE2002).

Benjamin, A.C.; Freire, J.L.F.; Vieira, R.D.; Diniz, J.L.C.; Andrade, E.Q., 2005. *Burst Tests on Pipeline Containing Interacting Corrosion Defects*. Proc. 24th International Conference on Offshore Mechanics and Arctic Engineering (OMAE2005).

Benjamin, A.C.; Vieira, R.D.; Freire, J.L.F.; Andrade, E.Q., 2006. *Burst Tests on Pipeline Containing Closely Spaced Corrosion Defects*. Proc. 25th International Conference on Offshore Mechanics and Arctic Engineering (OMAE2006-92131).

Benjamin, A.C.; Vieira, R.D.; Freire, J.L.F.; Castro, J.T.P., 2000. *Burst Tests on Pipeline with Long External Corrosion*. 3rd. International Pipeline Conference, ASME IPC2000, Vol.2, pp.793-799.

Bjornoy, O.H.; Marley, M.J., 2001. Assessment of Corroded Pipelines: Past, Present and Future. Proc. 11th International Offshore and Polar Engineering Conference (ISOPE01-138).

BS 7910, 1999. *Guide on Methods for Assessing the Acceptability of Flaws in Metallic Structures- Annex G: The Assessment of Corrosion in Pipes and Pressure Vessels*, Bristish Standard.

Cabral, H.L.D.; Willmersdorf, R.B.; Costa, F.A.; Lyra, P.R.M.; Afonso, S.M.B; Torres, J.V.S., 2006. *Automatização da Geração e Análise por Elementos Finitos de Defeitos de Corrosão em Dutos*. Congresso Nacional de Engenharia Mecânica (CONEM2006-03-536), Recife, Brasil.

Cabral, H.L.D.; Willmersdorf, R.B.; Lyra, P.R.M.; Afonso, S.M.B; Torres, J.V.S., 2006. *Desenvolvimento de Ambiente Computacional para Modelagem e Análise Automática de Dutos com Defeitos Causados por Corrosão*. Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE2006-CIL01-555), Belém, Brasil.

Caldwell, J.; Smith, G.; Vieth, P.;Williamson, G., 2001. *Pipeline Pigging Course*. Clarion Technical Conference and Pipes & Pipelines International.

Chouchaoui, B.A. & Pick, R.J., 1994. *Behaviour of circumferentially aligned corrosion pits*. International Journal of Pressure Vessels and Piping, v.57, pp.187-200.

Chouchaoui, B.A. & Pick, R.J., 1996. *Behaviour of longitudinally aligned corrosion pits*. International Journal of Pressure Vessels and Piping, v.67, pp.17-35.

Chouchaoui, B.A.; Pick, R.J.; Yost, D.B., 1992. *Burst Pressure Predictions of Line Pipe Containing Single Corrosion Pits using the Finite Element Method*, 11th International Conference on Offshore Mechanics and Arctic Engineering (OMAE 92), Vol. 5, Part A - Pipeline Technology, pp. 203-210.

Cosham, A.; Hopkins, P., 2001. *PDAM - The Pipeline Defect Assessment Manual*. A Report to the PDAM Joint Industry Project, Andrew Palmer and Associates, Draft Final Report.(www.penspenintegrity.com/library/library_frames.html)

Cosham, A.; Hopkins, P., 2002. *The Pipeline Defect Assessment Manual*. IPC02-27067, International Pipeline Conference, Calgary, Canada.
(www.penspenintegrity.com/library/library_frames.html)

Costa, F.A., 2004. *Geração Automática e Análise de Modelos de Dutos com Defeitos Causados por Corrosão*. Programa de Pós-Graduação em Engenharia Mecânica, UFPE, Recife, Dissertação de Mestrado, 56p.

Crisfield, M.N., 1991. *Non-Linear Finite Element Analysis of Solids and Structures*, Editora Jonh Wiley and Sons Ltd, London – England, pp.1-20.

Cronin, D.S., 2002. *Finite Element Analysis of Complex Corrosion Defects*. Proc. Pressure Vessel and Piping Conference (PVP2002-1288), PVP-Vol.441.

Cronin, D.S.; Pick, R.J., 2000. *Experimental Database for Corroded Pipe: Evaluation of RSTRENG and B31G*. 3rd International Pipeline Conference, ASME IPC, Vol. 2, pp. 757-767.

Cronin, D.S.; Pick, R.J., 2000. *A New Multi-level Assessment Procedure for Corroded Line Pipe*. 3rd International Pipeline Conference, ASME IPC, Vol. 2, pp. 801-808.

Diniz, J.L.C., 2002. *Resistência de Dutos com Defeitos Usinados*. Pós-Graduação em Engenharia Mecânica, PUC-Rio, Rio de Janeiro, Dissertação de Mestrado, 97p.

DNV, 1999. *Recomended Practice DNV RP-F101 Corroded Pipelines*, Det Norske Veritas, Norway.

Fedele, R.A., 2002. *Desafios da Soldagem em Tubulações. Metalurgia e Materiais.* ABM, v.58, n.521, p.322-326.

Fu, B. & Kirkwood, M.G., 1995. "*Predicting Failure Pressure of Internally Corroded Linepipe Using the Finite Element Method*", 14th International Conference on Offshore Mechanics and Arctic Engineering (OMAE'95), Vol. 5, Pipeline Technology, pp. 175-184.

Gentil, V., 2003. *Corrosão*. Editora LTC, 4ª ed., Rio de Janeiro, pp.275-276.

Hopkins, P., 2002. *The Challenge of Change in Engineering*. Journal of Pipeline Integrity, v.1, No.2, pp1-29.

Hopkins, P., 2002. *Training Engineers in Pipeline Integrity*. Western Regional Gas Conference, Arizona, EUA, pp 1-9.

Kennedy, J.L., 1993. *Oil and Gas Pipeline Fundamentals*. PennWell Publishing Company, 2ª edição, Tulsa, Oklahoma.

Kiefner, J.F.; Vieth, P.H., 1989. *A Modified Criterion for Evaluating the Remaining Strength of Corroded Pipe*. Contract PR-3-805, Pipeline Research Council International, Inc, American Gas Assotiation, Catalog No. L51688Hbe.

Kiefner, J.F.; Vieth, P.H., 1990. *Evaluating Pipe Conclusion: PC Program speeds new Criterion for Evaluating Corroded Pipe*. Oil & Gas Journal, v.88, no.34, pp.91-93.

Labaki, J.; *Introdução a Python: Tutorial Módulo A*, Universidade Estadual Paulista. http://labaki.tk/

Liu, H., 2003. *Pipeline Engineering*, Lewis Publisher, pp. 3-15, 319-330, 383-395.

MMS, 2000. *Appraisal and Development of Pipeline Defect Assessment Methodologies.* Minerals Management Service Contract No.1435-01-CT-99-50001, Final Report, Washington, 172p. (http://www.mms.gov/mmshome.htm)

Noronha Jr., D.B.; Benjamin, A.C.; Andrade, E.Q., 2002. *Finite Element Models for the prediction of failure of pipelines with long corrosion defects*, International Pipeline Conference (IPC02-27191),v.B, Calgary, Canada, pp. 8

Palmer-Jones, R.; Hopkins, P.; Pople, A.; Cosham, A., 2002. *Lessons Learnt from Fitness-for-Purpose Assessments of Defects Detected by Smart Pigs*. Onshore Pipelines Conference, Amsterdam.

PATRAN, 2005. Help system: *User's Guide, Reference Manual, PCL Manuals, PCL Reference*. http://www.mscsoftware.com/products/patran_support.cfm?Q=396&Z=402

PyExcelerator, 2005, http://sourceforge.net/projects/pyexcelerator

PYTHON, 2005. "Python Documentation Release 2.4.1: Tutorial and Library Reference Manual" http://www.python.org/doc/

Reis, C.R., 2004, "*Tutorial: Python na Prática*", http://www.async.com.br/projects/pnp/

Santos Neto, N. F., 2003. *Caracterização de Soldas em Aços API 5L com Diferentes Arames Tubulares e Temperaturas de Pré-Aquecimento.* Pós-Graduação em Engenharia Mecânica, Unicamp, Campinas, Dissertação de Mestrado, 87p.

Souza, R.D., 2003. *Avaliação Estrutural de Dutos com Defeitos de Corrosão Reais*. Pós-Graduação em Engenharia Mecânica, PUC-Rio, Rio de Janeiro, Dissertação de Mestrado, 112f.

Souza, R.D.; Benjamin, A.C.; Vieira, R.D.; Freire, J.L.F.; Castro, J.T.P, 2005. *Burst Tests of Corroded Pipe Segments Removed from Service*. IBP1220_05 - Rio Pipeline Conference and Exposition, pp.1-10.

Telles, P.C.S., 1997. *Tubulações Industriais: Materiais, Projeto, Montagem*. Editora LTC, 9ª edição, Rio de Janeiro, pp. 1-20, 239.

Tims, P.; Wilson, O., 2002. *When is Corrosion not Corrosion? A Decade of MFL Pipeline Inspection*. Pipeline Pigging, Integrity Assessment and Repair Conference, Amsterdam.

Tiratsoo, J.N.H., 1992. *Pipeline Pigging Technology*. Gulf Professional Publishing, 2nd Edition, pp.1-30.

Wiesner, C.S.; Maddox, S.J.; Xu, W.; Webster, G.A.; Burdekin, F.M.; Andrews, R.M.; Harrison, J.D., 2000. *Engineering critical analyses to BS 7910 – the UK guide on methods for assessing the acceptability of flaws in metallic structures*. International Journal of Pressure Vessels and Piping, v.77, pp.883-893.

APÊNDICE A – Histórico da Análise Não-Linear do Modelo IDTS2

LOAD STEP	P	ΔP	p	ITER.	σeqv (MPa)	Max.Def.Plástica
0	0.00	0.00	0.0000	1	**0.0000**	**0.0**
			1.13465	6		**0.0**
			2.26930	6		**0.0**
			3.40395	6		**0.0**
1	4.5386	1.8911	4.53860	6	**378.31**	**0.0**
			5.01138	6		**0.0**
			5.48415	6		**0.0**
			5.95693	6		**0.0**
2	6.4297	1.8911	6.42970	6	**534.74**	**0.1689E-03**
			6.90248	5		**0.2626E-03**
			7.37525	4		**0.3168E-03**
			7.84803	4		**0.3010E-03**
3	8.3208	1.8911	8.32080	4	**538.93**	**0.3366E-03**
			8.79358	5		**0.3383E-03**
			9.26635	5		**0.3504E-03**
			9.73912	5		**0.3573E-03**
4	10.2119	1.8911	10.2119	6	**545.89**	**0.3579E-03**
			10.6847	6		**0.3809E-03**
			11.1574	6		**0.4639E-03**
			11.6302	6		**0.5520E-03**
5	12.1030	1.8911	12.1030	6	**563.72**	**0.5711E-03**
			12.5758	6		**0.6137E-03**
			13.0485	6		**0.6241E-03**
			13.5213	6		**0.6154E-03**
6	13.9941	1.8911	13.9941	6	**587.26**	**0.6713E-03**
			14.4669	6		**0.7515E-03**
			14.9397	6		**0.8397E-03**
			15.4124	6		**0.9205E-03**
7	15.8852	1.8911	15.8852	6	**607.63**	**0.1020E-02**
			16.3580	6		**0.1233E-02**
			16.8307	6		**0.1481E-02**
			17.3035	7		**0.1721E-02**
8	17.7763	1.8911	17.7763	7	**632.78**	**0.2020E-02**
			18.2491	7		**0.2428E-02**
			18.7218	8		**0.2943E-02**
			19.1946	8		**0.3592E-02**
9	19.6674	1.8911	19.6674	8	**663.57**	**0.4426E-02**
			18.0127	7		**0.1161E-02**
			18.2491	7		**0.1263E-02**
			18.4854	7		**0.1393E-02**
9a	18.7218	0.9455	18.7218	8	**647.31**	**0.1547E-02**
			18.9582	8		**0.1708E-02**
			19.1945	8		**0.1880E-02**
			19.4309	8		**0.2087E-02**
10	19.6673	0.9455	19.6673	8	**663.54**	**0.2335E-02**
			19.9037	8		**0.2566E-02**
			20.1401	8		**0.2850E-02**
			20.3764	9		**0.3261E-02**
11	20.6128	0.9455	20.6128	9	**682.45**	**0.3782E-02**

			19.7855	8		**0.1251E-02**
			19.9036	8		**0.1313E-02**
			20.0218	8		**0.1386E-02**
11a	20.1400	0.4727	20.1400	8	**672.99**	**0.1463E-02**
			20.2582	8		**0.1553E-02**
			20.3764	8		**0.1694E-02**
			20.4945	8		**0.1804E-02**
12	20.6127	0.4727	20.6127	8	**682.44**	**0.1968E-02**
			20.7309	8		**0.2111E-02**
			20.8490	8		**0.2251E-02**
			20.9672	8		**0.2458E-02**
13	21.0854	0.4727	21.0854	8	**692.40**	**0.2683E-02**
			20.6718	8		**0.1035E-02**
			20.7309	8		**0.1069E-02**
			20.7899	8		**0.1105E-02**
13a	20.8490	0.2363	20.8490	8	**687.38**	**0.1141E-02**
			20.9081	8		**0.1201E-02**
			20.9672	8		**0.1252E-02**
			21.0262	8		**0.1306E-02**
14	21.0853	0.2363	21.0853	8	**692.40**	**0.1373E-02**
			21.1444	8		**0.1421E-02**
			21.2034	8		**0.1491E-02**
			21.2625	8		**0.1554E-02**
15	21.3216	0.2363	21.3216	8	**697.63**	**0.1606E-02**
			21.3807	8		**0.1664E-02**
			21.4397	8		**0.1734E-02**
			21.4988	8		**0.1840E-02**
16	21.5579	0.2363	21.5579	9	**703.28**	**0.1946E-02**
			21.6170	9		**0.2019E-02**
			21.6760	9		**0.2097E-02**
			21.7351	9		**0.2183E-02**
17	21.7942	0.2363	21.7942	9	**708.95**	**0.2272E-02**
			21.8533	10		**0.2410E-02**
			21.9123	10		**0.2557E-02**
			21.9714	10		**0.2746E-02**
18	22.0305	0.2363	22.0305	9	**715.43**	**0.3039E-02**
			21.8237	9		**0.1186E-02**
			21.8532	9		**0.1222E-02**
			21.8828	9		**0.1258E-02**
18a	21.9123	0.1181	21.9123	9	**711.97**	**0.1298E-02**
			21.9418	9		**0.1346E-02**
			21.9713	9		**0.1401E-02**
			22.0009	9		**0.1478E-02**
19	22.0304	0.1181	22.0304	10	**715.43**	**0.1561E-02**
			22.0599	10		**0.1643E-02**
			22.08945	50		
			22.11897			
20	22.1485	0.1181	22.1485		**Non Conv.**	
			22.0451	9		**0.8074E-03**
			22.0599	9		**0.8371E-03**
			22.0746	12		**0.8687E-03**
20a	22.0894	0.0590	22.0894	50	**Non Conv.**	
			22.0378	8		**0.4007E-03**
			22.0451	8		**0.4064E-03**
			22.0525	8		**0.4150E-03**
20b	22.0599	0.0295	22.0599	8	**716.40**	**0.4218E-03**

			22.0673	9		**0.4300E-03**
			22.0747	11		**0.4405E-03**
			22.0820	13		**0.4533E-03**
21	22.0894	0.0295	22.0894	13	**717.48**	**0.4797E-03**
			22.096775	50		
			22.10415			
			22.11153			
22	22.1189	0.0295	22.1189		**Non Conv.**	
			22.093075	50		
			22.09675			
			22.10042			
22a	22.1041	0.0147	22.1041		**Non Conv.**	
			22.091225	50		
			22.09305			
			22.09487			
22b	22.0967	0.0073	22.0967		**Non Conv.**	

APÊNDICE B – Histórico da Análise Não-Linear do Modelo IDTS3

LOAD STEP	P	ΔP	p	ITER.	σeqv.(MPa)	Max.Def.Plástica
0	0.00	0.00	0.0000	1	**0.0000**	**0.0**
			1.00178	6		**0.0**
			2.00355	7		**0.0**
			3.00533	7		**0.0**
1	4.0071	1.6696	4.00710	7	**391.69**	**0.0**
			4.42450	6		**0.0**
			4.84190	6		**0.0**
			5.25930	6		**0.0**
2	5.6767	1.6696	5.67670	6	**534.52**	**0.1095E-03**
			6.09410	6		**0.2794E-03**
			6.51150	6		**0.3034E-03**
			6.92890	5		**0.3487E-03**
3	7.3463	1.6696	7.34630	5	**538.89**	**0.3307E-03**
			7.76370	5		**0.3528E-03**
			8.18110	6		**0.3560E-03**
			8.59850	6		**0.3430E-03**
4	9.0159	1.6696	9.01590	7	**545.36**	**0.3606E-03**
			9.43330	7		**0.3933E-03**
			9.85070	7		**0.4152E-03**
			10.2681	7		**0.5156E-03**
5	10.6855	1.6696	10.6855	7	**564.81**	**0.6539E-03**
			11.1029	8		**0.6456E-03**
			11.5203	8		**0.7434E-03**
			11.9377	8		**0.8430E-03**
6	12.3551	1.6696	12.3551	8	**592.06**	**0.9617E-03**
			12.7725	8		**0.1349E-02**
			13.1899	7		**0.1492E-02**
			13.6073	6		**0.1625E-02**
7	14.0247	1.6696	14.0247	6	**619.30**	**0.1678E-02**
			14.4421	6		**0.1811E-02**
			14.8595	7		**0.2055E-02**
			15.2769	7		**0.2351E-02**
8	15.6943	1.6696	15.6943	8	**647.32**	**0.2711E-02**
			14.2334	6		**0.8994E-03**
			14.4421	6		**0.9316E-03**
			14.6508	7		**0.9930E-03**
8a	14.8595	0.8348	14.8595	7	**633.67**	**0.1060E-02**
			15.0682	7		**0.1136E-02**
			15.2769	7		**0.1213E-02**
			15.4856	7		**0.1305E-02**
9	15.6943	0.8348	15.6943	7	**647.28**	**0.1403E-02**
			15.9030	7		**0.1502E-02**
			16.1117	8		**0.1603E-02**
			16.3204	8		**0.1722E-02**
10	16.5291	0.8348	16.5291	8	**661.07**	**0.1856E-02**
			16.7378	8		**0.1991E-02**
			16.9465	8		**0.2142E-02**
			17.1552	8		**0.2356E-02**
11	17.3639	0.8348	17.3639	8	**676.28**	**0.2590E-02**

			16.6334	7		**0.9762E-03**
			16.7378	7		**0.1013E-02**
			16.8422	7		**0.1054E-02**
11a	16.9465	0.4174	16.9465	7	**668.71**	**0.1094E-02**
			17.0508	8		**0.1151E-02**
			17.1552	8		**0.1202E-02**
			17.2595	8		**0.1262E-02**
12	17.3639	0.4174	17.3639	8	**676.27**	**0.1324E-02**
			17.4682	8		**0.1373E-02**
			17.5726	8		**0.1422E-02**
			17.6769	8		**0.1489E-02**
13	17.7813	0.4174	17.7813	8	**683.83**	**0.1564E-02**
			17.8856	8		**0.1644E-02**
			17.9900	8		**0.1735E-02**
			18.0943	8		**0.1825E-02**
14	18.1987	0.4174	18.1987	8	**691.56**	**0.1912E-02**
			18.3030	8		**0.2002E-02**
			18.4074	8		**0.2105E-02**
			18.5117	8		**0.2204E-02**
15	18.6161	0.4174	18.6161	8	**699.38**	**0.2337E-02**
			18.7204	8		**0.2466E-02**
			18.8248	8		**0.2595E-02**
			18.9291	8		**0.2737E-02**
16	19.0335	0.4174	19.0335	8	**707.41**	**0.2908E-02**
			18.6683	8		**0.1215E-02**
			18.7205	8		**0.1246E-02**
			18.7726	8		**0.1278E-02**
16a	18.8248	0.2087	18.8248	8	**703.42**	**0.1313E-02**
			18.8770	8		**0.1346E-02**
			18.9291	8		**0.1388E-02**
			18.9813	8		**0.1432E-02**
17	19.0335	0.2087	19.0335	8	**707.41**	**0.1471E-02**
			19.0857	8		**0.1510E-02**
			19.1378	8		**0.1550E-02**
			19.1900	8		**0.1602E-02**
18	19.2422	0.2087	19.2422	8	**711.34**	**0.1648E-02**
			19.2944	8		**0.1699E-02**
			19.3465	8		**0.1751E-02**
			19.3987	8		**0.1791E-02**
19	19.4509	0.2087	19.4509	8	**715.69**	**0.1851E-02**
			19.5031	8		**0.1903E-02**
			19.5553	8		**0.1953E-02**
			19.6074	8		**0.2012E-02**
20	19.6596	0.2087	19.6596	50	**Non Conv.**	
			19.4770	7		**0.9436E-03**
			19.5030	7		**0.9566E-03**
			19.5291	7		**0.9687E-03**
20a	19.5552	0.1043	19.5552	50	**Non Conv.**	
			19.4639	7		**0.4700E-03**
			19.4769	7		**0.4722E-03**
			19.4900	7		**0.4757E-03**
20b	19.5030	0.0521	19.5030	7	**717.18**	**0.4794E-03**

			19.5160	7		**0.4822E-03**
			19.52905	50		
			19.54208			
21	19.5551	0.0521	19.5551		**Non Conv.**	
			19.5095	6		**0.2403E-03**
			19.5160	7		**0.2409E-03**
			19.5225	50		
21a	19.5290	0.0260	19.5290		**Non Conv.**	
			19.5062	6		**0.1200E-03**
			19.5095	6		**0.1202E-03**
			19.51275	50		
21b	19.5160	0.0130	19.5160		**Non Conv.**	
			19.5046	5		**0.5999E-04**
			19.5062	5		**0.6002E-04**
			19.5079	5		**0.6007E-04**
21c	19.5095	0.0065	19.5095	9	**717.37**	**0.6013E-04**

APÊNDICE C – Histórico da Análise Não-Linear do Modelo IDTS4*

LOAD STEP	P	ΔP	p	ITER.	σeqv.(MPa)	Max.Def.Plástica
0	0.00	0.00	0.0000	1	**0.0000**	**0.0**
			1.09565	6		**0.0**
			2.19130	7		**0.0**
			3.28695	7		**0.0**
1	4.3826	1.8261	4.38260	7	**367.45**	**0.0**
			4.83913	6		**0.0**
			5.29565	6		**0.0**
			5.75218	6		**0.0**
2	6.2087	1.8261	6.20870	6	**534.74**	**0.1688E-03**
			6.66522	6		**0.2618E-03**
			7.12175	5		**0.2870E-03**
			7.57827	5		**0.3052E-03**
3	8.0348	1.8261	8.03480	5	**538.92**	**0.3062E-03**
			8.49132	4		**0.3268E-03**
			8.94785	5		**0.3547E-03**
			9.40437	5		**0.3590E-03**
4	9.8609	1.8261	9.86090	6	**546.07**	**0.3811E-03**
			10.3174	6		**0.3953E-03**
			10.7739	6		**0.4899E-03**
			11.2305	6		**0.5941E-03**
5	11.6870	1.8261	11.6870	6	**564.90**	**0.6575E-03**
			12.1435	6		**0.6783E-03**
			12.6000	6		**0.7576E-03**
			13.0566	6		**0.7491E-03**
6	13.5131	1.8261	13.5131	6	**589.44**	**0.7909E-03**
			13.9696	6		**0.8225E-03**
			14.4261	6		**0.8826E-03**
			14.8827	5		**0.9687E-03**
7	15.3392	1.8261	15.3392	6	**610.39**	**0.1148E-02**
			15.7957	7		**0.1344E-02**
			16.2522	7		**0.1555E-02**
			16.7088	7		**0.1832E-02**
8	17.1653	1.8261	17.1653	8	**635.44**	**0.2101E-02**
			17.6218	8		**0.2493E-02**
			18.0783	8		**0.2980E-02**
			18.5349	9		**0.3511E-02**
9	18.9914	1.8261	18.9914	9	**664.62**	**0.4198E-02**
			17.3935	8		**0.1194E-02**
			17.6218	7		**0.1295E-02**
			17.8501	8		**0.1420E-02**
9a	18.0783	0.9130	18.0783	8	**649.37**	**0.1560E-02**
			18.3065	8		**0.1684E-02**
			18.5348	8		**0.1831E-02**
			18.7630	8		**0.2001E-02**
10	18.9913	0.9130	18.9913	8	**664.59**	**0.2190E-02**
			19.2195	8		**0.2391E-02**
			19.4478	9		**0.2626E-02**
			19.6761	9		**0.2890E-02**
11	19.9043	0.9130	19.9043	9	**681.76**	**0.3383E-02**

			19.1054	8		**0.1168E-02**
			19.2195	8		**0.1219E-02**
			19.3337	8		**0.1284E-02**
11a	19.4478	0.4565	19.4478	8	**673.25**	**0.1338E-02**
			19.5619	8		**0.1399E-02**
			19.6761	8		**0.1487E-02**
			19.7902	8		**0.1623E-02**
12	19.9043	0.4565	19.9043	8	**681.74**	**0.1745E-02**
			20.0184	8		**0.1885E-02**
			20.1326	8		**0.1996E-02**
			20.2467	8		**0.2114E-02**
13	20.3608	0.4565	20.3608	8	**690.52**	**0.2312E-02**
			20.4749	8		**0.2437E-02**
			20.5891	8		**0.2609E-02**
			20.7032	9		**0.2817E-02**
14	20.8173	0.4565	20.8173	9	**699.51**	**0.2971E-02**
			20.4179	8		**0.1194E-02**
			20.4749	8		**0.1234E-02**
			20.5320	8		**0.1277E-02**
14a	20.5890	0.2282	20.5890	8	**694.81**	**0.1323E-02**
			20.6460	8		**0.1377E-02**
			20.7031	8		**0.1431E-02**
			20.7601	8		**0.1463E-02**
15	20.8172	0.2282	20.8172	8	**699.50**	**0.1504E-02**
			20.8742	8		**0.1565E-02**
			20.9313	8		**0.1612E-02**
			20.9884	8		**0.1671E-02**
16	21.0454	0.2282	21.0454	8	**704.23**	**0.1741E-02**
			21.1024	8		**0.1800E-02**
			21.1595	8		**0.1850E-02**
			21.2165	8		**0.1918E-02**
17	21.2736	0.2282	21.2736	8	**708.85**	**0.1970E-02**
			21.3306	8		**0.2061E-02**
			21.3877	8		**0.2144E-02**
			21.44475	50		
18	21.5018	0.2282	21.5018		**Non Conv.**	
			21.3021	8		**0.1017E-02**
			21.3306	8		**0.1042E-02**
			21.3592	8		**0.1065E-02**
18a	21.3877	0.1141	21.3877	8	**711.23**	**0.1075E-02**
			21.4162	8		**0.1087E-02**
			21.4448	8		**0.1108E-02**
			21.4733	8		**0.1134E-02**
19	21.5018	0.1141	21.5018	8	**713.76**	**0.1160E-02**
			21.5303	8		**0.1181E-02**
			21.5589	9		**0.1197E-02**
			21.587375	50		
20	21.6159	0.1141	21.6159		**Non Conv.**	
			21.5161	7		**0.5882E-03**
			21.5303	8		**0.5915E-03**
			21.5446	8		**0.6025E-03**
20a	21.5588	0.0570	21.5588	50	**Non Conv.**	
			21.5089	7		**0.2928E-03**
			21.5160	7		**0.2946E-03**
			21.5232	7		**0.2957E-03**
20b	21.5303	0.0285	21.5303	7	**714.42**	**0.2954E-03**

			21.537425	50		
			21.54455			
			21.55168			
21	21.5588	0.0285	21.5588		**Non Conv.**	
			21.53385	50		
			21.53735			
			21.54092			
21a	21.5445	0.0143	21.5445		**Non Conv.**	
			21.5321	6		**0.8321E-04**
			21.5339	6		**0.7750E-04**
			21.5356	6		**0.7769E-04**
21b	21.5374	0.0071	21.5374	6	**714.58**	**0.7484E-04**