



Fractus 2D Versão 2.0

**Cálculo Numérico de Parâmetros de Mecânica da Fratura
em Componentes Estruturais 2D Contendo Trincas**

Claudio Ruggieri, Ph.D., Professor Titular
Depto. de Engenharia Naval e Oceânica, Universidade de São Paulo

Projeto de Pesquisa Patrocinado pelo

FAPESP
Fundação de Amparo à Pesquisa do Estado de São Paulo

Dezembro de 2011

Fractus 2D

Versão 2.0

**Cálculo Numérico de Parâmetros de Mecânica da Fratura
em Componentes Estruturais 2D Contendo Trincas**

Claudio Ruggieri

Depto. de Engenharia Naval e Oceânica - PNV

Escola Politécnica

Universidade de São Paulo

São Paulo
Dezembro de 2011

.... The search for truth is one way hard and another easy, for no one can grasp it fully nor miss it wholly. Each adds to our store of knowledge.
And from it arises a certain grandeur.

Aristóteles (355 A. C.)

AGRADECIMENTOS

Esta investigação foi patrocinada pela *Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP* por intermédio do Auxílio à Pesquisa 2009-54229-3 e pelo *Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq* por meio de Bolsa de Produtividade em Pesquisa PQ-304132/2009-8.

Índice

Secção No.	Página
Agradecimentos	ii
1. Introdução	1
2. A Metodologia $J-Q$	2
2.1 Limitações da Mecânica da Fratura Monoparamétrica	2
2.2 Mecânica da Fratura Biparamétrica: Teoria $J-Q$	4
2.3 Soluções de Referência Utilizando o Modelo MBL	7
2.4 Extensão da Teoria $J-Q$ para Propagação de Trincas e Curvas $J-R$	9
2.5 Procedimentos Computacionais	13
2.6 Trajetórias $J-Q$ para Componentes Estruturais Pré-Trincados	16
3. A Metodologia ETA para Determinação de J e CTOD	18
3.1 Fatores Eta para Determinação da Integral J em Trincas Estacionárias	18
3.2 Fatores Eta para Determinação do CTOD em Trincas Estacionárias ..	20
3.3 Procedimentos Computacionais	22
3.4 Fatores Eta para Corpos-de-Prova Convencionais SE(B)	29
4. Sintaxes de Comandos	30
4.1 Utilização de <i>FRACTUS2D</i>	30
4.2 Notação Utilizada na Sintaxe de Comandos	31
4.3 Definição da Análise	34
4.4 Blocos de Subcomandos	35
5. Determinação do Parâmetro Q	36
5.1 Definição da Análise	36
5.2 Blocos de Subcomandos	36
5.3 Placa Infinita de Referência	36
5.3.1 Coordenadas Nodais e Incidência de Elementos	37
5.3.2 Formato de Resultados de Elementos Finitos	39
5.3.3 Definição da Trinca	39
5.3.4 Elementos e Nós da Região da Trinca	41
5.4 Componente Estrutural	43
5.4.1 Identificação do Componente Estrutural	43
5.4.2 Coordenadas Nodais e Incidência de Elementos	44
5.4.3 Formato de Resultados de Elementos Finitos	44
5.4.4 Definição da Trinca	44

5.4.5	Elementos e Nós da Região da Trinca	44
5.4.6	Parâmetro de Carga: Integral J	44
5.4.7	Curva de Resistência $J-R$	45
5.4.8	Células Computacionais	46
5.5	Parâmetros de Análise	47
5.5.1	Raio Adimensional	47
5.5.2	Tolerância de Busca Nodal	47
5.5.3	Fator de Adaptação	47
5.5.4	Tensão de Escoamento do Material	48
5.5.5	Integral J de Referência	48
5.5.6	Construção de Trajetórias $J-Q$	48
5.5.7	Distribuição de Tensões e Deformações	49
5.5.8	Uso de Resultados Nodais	49
5.5.9	Versão do Programa FEA <i>WARP3D</i>	49
5.5.10	Trajetórias $J-Q$ para Propagação de Trincas	50
6.	Determinação dos Fatores ETA	51
6.1	Definição da Análise	51
6.2	Blocos de Subcomandos	51
6.3	Configuração do Componente ou Corpo-de-Prova	51
6.3.1	Identificação do Componente Estrutural ou Corpo-de-Prova ...	52
6.3.2	Configuração (Geometria) do Corpo-de-Prova	52
6.3.3	Dimensões Principais do Corpo-de-Prova	52
6.3.4	Coordenadas Nodais e Incidência de Elementos	53
6.3.5	Formato de Resultados de Elementos Finitos	56
6.3.6	Parâmetro de Carga: Integral J	57
6.4	Parâmetros da Malha de Elementos Finitos	57
6.4.1	Definição da Trinca	57
6.4.2	Definição da Curva de Carga <i>vs.</i> Deslocamento	59
6.4.3	Fator de Simetria ou Escala	61
6.5	Parâmetros de Análise	62
6.5.1	Tolerância de Busca Nodal	62
6.5.2	Fração da Área Plástica	63
6.5.3	Coeficiente Angular Linear da Curva Carga <i>vs.</i> Deslocamento .	63
6.5.4	Propriedades Mecânicas do Material	63
6.5.5	Fator de Restrição Plástica para CTOD Elástico	64
6.5.6	Construção de Fatores Eta	65
6.5.7	Curva de Carga <i>vs.</i> Deslocamento	65
6.5.8	Procedimentos para Determinação do CTOD	65
6.5.9	Determinação do Fator Rotacional Plástico	66
6.5.10	Versão do Programa FEA <i>WARP3D</i>	66
7.	Exemplos Ilustrativos	68
7.1	Trajetória $J-Q$ para um Corpo-de-Prova C(T)	68
7.2	Fatores Eta para um Corpo-de-Prova SE(T)	76
8.	Referências	83
	Anexo A: Geometria de Corpos-de-Prova Convencionais .	86
	Anexo B: Funções de Flexibilidade Elástica	88

Introdução

1

Este documento descreve os aspectos teóricos e comandos necessários para a utilização do programa *Fractus2D*, um código computacional escrito em FORTRAN F77 e F90 para o cálculo e determinação de diversos parâmetros utilizados em problemas típicos de mecânica da fratura aplicada a componentes estruturais bidimensionais (2D) contendo trincas estacionárias e não estacionárias (crescimento ou propagação de trincas sob carregamento monotônico). A presente versão de *Fractus2D* implementa as seguintes funcionalidades:

- Determinação numérica das trajetórias J - Q para qualquer configuração geométrica pré-trincada utilizando uma solução de referência baseada no modelo MBL para uma placa infinita contendo uma trinca;
- Determinação numérica dos fatores η para obtenção dos parâmetros elasto-plásticos J e CTOD em corpos-de-prova convencionais C(T), SE(B) e SE(T).

Os resultados numéricos de forças, deslocamentos, tensões e deformações são obtidos por intermédio de análises não lineares utilizando o método de elementos finitos. A presente versão de *Fractus2D* é construída a partir do programa de elementos finitos WARP3D [1]. Algumas características específicas do programa *Fractus2D* incluem uma linguagem conveniente de comandos em formato livre e uma interface (transparente ao usuário) para tratamento de resultados de elementos finitos armazenados em formato binário ou ASCII compatível com o programa MSC PATRAN [2].

Este manual é composto da seguinte forma. A Seção 2 sumariza os fundamentos teóricos da metodologia biparamétrica J - Q para componentes estruturais contendo trincas estacionárias e não estacionárias. A Seção 3 apresenta a metodologia η a partir da interpretação energética da Integral J para sólidos elasto-plásticos contendo trincas e defeitos. A Seção 4 introduz os comandos básicos para utilização do programa *Fractus2D* a qual é seguida pela descrição dos comandos e informações necessárias à execução do programa para análises de trajetórias J - Q e fatores η apresentadas nas Seções 5 e 6. A Seção 7 apresenta exemplos ilustrativos para determinação da trajetória J - Q de um corpo-de-prova convencional compacto C(T) e dos fatores η para um corpo-de-prova SE(T) fixado por pinos.

A Metodologia J-Q

2

2.1 Limitações da Mecânica da Fratura Monoparamétrica

A mecânica da fratura clássica (convencional) emprega um parâmetro único para descrever as condições de propagação instável da fratura [3,4] o que permite, sob condições de plasticidade restrita nas vizinhanças da trinca ou defeito (i.e., escoamento em pequena escala ou, como é usualmente conhecido na literatura, *small scale yielding (SSY) conditions*), estabelecer uma correlação direta entre valores de tenacidade medidos experimentalmente (K_{Ic} , J_c ou CTOD) e o comportamento à fratura de um componente estrutural em serviço. O conceito fundamental que permite o emprego deste procedimento (*mecânica da fratura correlativa*) é a existência de *similaridade* ou *similitude* dos campos de tensões entre corpos-de-prova laboratoriais (i.e., com dimensões reduzidas) e o componente estrutural em serviço [4], como ilustrado esquematicamente na Fig. 1.

Tais métodos convencionais, não obstante a sua grande utilidade e emprego generalizado, apresentam deficiências potencialmente importantes como procedimentos robustos para avaliação de integridade estrutural. Para aços estruturais ferríticos, observações experimentais realizadas na região de transição frágil-dúctil (*DBT transition region*) consistentemente revelam um significativo efeito da geometria do corpo de prova, do tamanho relativo da trinca (relação tamanho da trinca sobre largura do corpo de prova – a/W) e do modo de carregamento (flexão *vs.* tensão). A Fig. 2 apresenta dados ilustrativos para aços estruturais típicos testados na região de transição frágil-dúctil [5,6] os quais mostram claramente uma elevação significativa nos valores experimentais de tenacidade (J_c) para corpos de prova SE(B) com entalhe raso (*shallow crack SE(B) specimen*). Este aumento *aparente* de tenacidade para aços estruturais possui enormes implicações práticas sobre procedimentos de avaliação de defeitos, particularmente programas de reparos e extensão da vida útil de estruturas em serviço.

O comportamento apresentado na Fig. 2 ocorre devido à extensão generalizada da plasticidade (i.e., escoamento em grande escala ou, como é usualmente conhecido na literatura, *large*

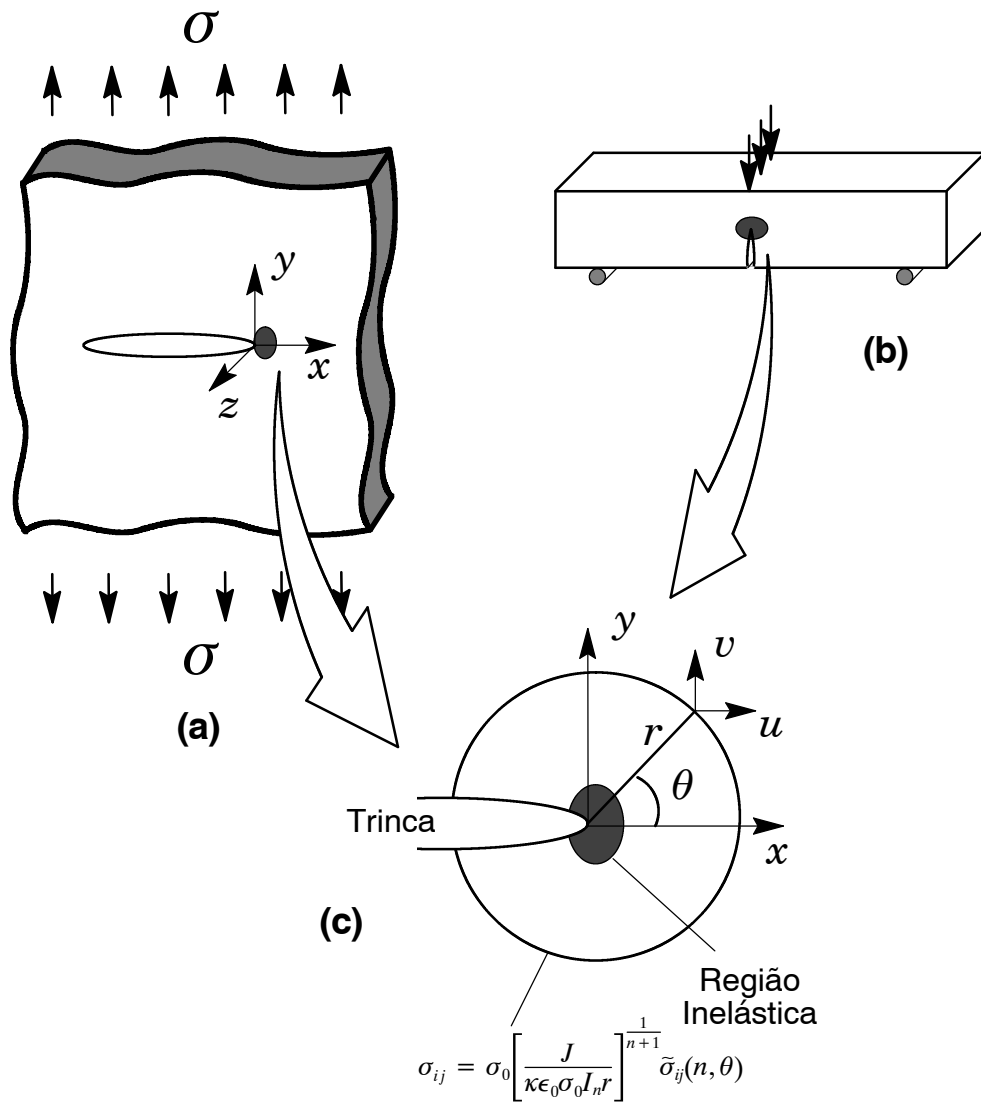


Figura 1 (a) Placa tensionada representativa de um componente estrutural constituída de material plasticamente deformável contendo uma trinca; (b) Corpo de prova convencional SE(B) de 3 Pontos à Flexão; (c) Zona de dominância J na região da ponta da trinca indicando a validade da mecânica da fratura monoparamétrica.

scale yielding (LSY conditions) para regiões suficientemente remotas das vizinhanças da trinca (violando, portanto, as condições *SSY* descritas anteriormente) o que provoca um relaxamento acentuado do campo de tensões nas proximidades da trinca. Este importante fenômeno, genericamente denominado *perda de restrição da ponta da trinca* (comumente denominado na literatura como *loss of crack-tip constraint* [7]) invalida o conceito de um parâmetro único como *descriptor* das condições de fratura uma vez que valores maiores da Integral J (ou, equivalentemente, CTOD) são necessários para provocar a fratura frágil (clivagem transgranular) em relação à condição de elevada triaxialidade caracterizada pelo regime *SSY*.

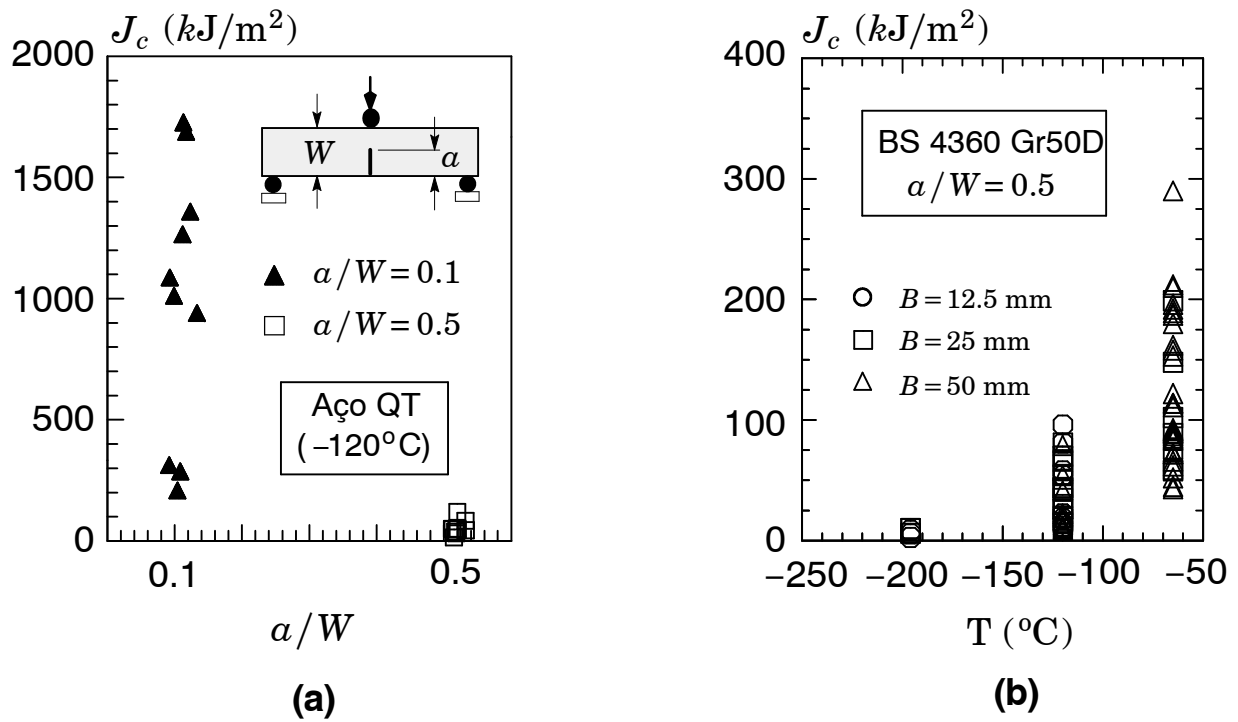


Figura 2 Valores experimentais de tenacidade (clivagem) para corpos de prova SE(B) testados na região de transição DBT: (a) aço estrutural ARBL temperado e revenido (QT) a -120°C [5]. (b) aço estrutural BS 4360 steel testado sob várias temperaturas [6].

2.2 Mecânica da Fratura Biparamétrica: Teoria J-Q

Os argumentos apresentados acima sobre as limitações de um parâmetro único para caracterizar o campo de tensões de sólidos contendo trincas sob condições LSY motivaram o desenvolvimento de teorias biparamétricas para a fratura. Sob carregamento fixo, tais metodologias assumem uma forma separável do campo local de tensões para sólidos contendo trincas em um campo de alta triaxialidade (tal como o campo SSY) e um campo *constante* o qual “quantifica” o nível de restrição da ponta da trinca. Tais desenvolvimentos centralizam essencialmente sobre três linhas metodológicas: (1) a teoria J - T desenvolvida por Hancock e colaboradores [8-9], e Parks [10], a qual é construída sobre a tensão elástica T , (2) a teoria J - Q desenvolvida por O’Dowd e Shih [11,12] utilizando o parâmetro hidrostático Q e (3) a teoria J - A_2 proposta por Chao e colaboradores [13-15].

A característica central de metodologias biparamétricas descritivas do campo elasto-plástico de tensões para trincas (estacionárias) em materiais homogêneos é a utilização de dois parâmetros escalares para quantificar a magnitude destes campos e o seu nível de triaxialidade (*crack-tip constraint*) em relação a uma solução referência. Desta forma, a Integral J de-

termina a *amplitude* dos campos elasto-plásticos de tensões de alta triaxialidade dentro de regiões suficientemente próximas da trinca (cerca de 5~10 CTODs) enquanto o segundo parâmetro (T ou Q) determina o *nível de triaxialidade* associado à perda de restrição da ponta da trinca (*constraint loss*). Em particular, a descrição J - Q dos campos de tensões para estado plano de deformação sob Modo I deriva da aplicação da formulação MBL (*modified boundary layer*), representado pela placa infinita mostrada na Fig. 3, onde as trações remotas são dadas pelos primeiros dois termos da solução linear elástica de William [16]:

$$\sigma_{ij} = \frac{K_I}{\sqrt{2\pi r}} f_{ij}(\theta) + T\delta_{1i}\delta_{1j} . \quad (1)$$

onde r e θ são coordenadas polares centradas na ponta da trinca ($\theta = 0$ corresponde ao plano de propagação da trinca sob Modo I de deformação), f_{ij} são funções angulares adimensionais, K_I é o fator de intensidade de tensões, $K_I = \sqrt{EJ/(1 - \nu^2)}$, E é o módulo de elasticidade longitudinal, ν é o coeficiente de Poisson, δ_{ij} denota o delta Kronecker ($\delta_{ij} = 0$ para qualquer $i \neq j$) e T representa uma tensão não singular aplicada paralelamente ao plano da trinca. A introdução de $T = 0$ na Eq. (1) recupera a solução elástica clássica baseada somente no parâmetro K_I . Campos de tensões à frente da ponta da trinca com diferentes níveis de triaxialidade são gerados a partir de tensões T variáveis aplicadas no modelo (a tensão T não afeta o valor da Integral J). A partir de considerações dimensionais, estes campos podem ser representados por famílias de campos de tensões parametrizadas pelo parâmetro de carga T/σ_0 ; tal parâmetro fornece uma medida conveniente do estado de triaxialidade (restrição da ponta da trinca) para diferentes configurações estruturais. Entretanto, a aplicabilidade da tensão T como parâmetro descritor do nível de triaxialidade para configurações estruturais sob condições generalizadas de escoamento e plasticidade (LSY) torna-se questionável uma vez que a solução *elástica* dada pela Eq. (1), sobre a qual a tensão T é definida, é uma solução assintótica a qual é fortemente violada quando a deformação plástica estende-se além da região de validade de escoamento limitado (SSY) na ponta da trinca.

As limitações da teoria J - T discutidas acima motivaram O'Dowd e Shih [11,12] a proporem uma descrição biparamétrica *aproximada* para os campos de tensões elasto-plásticos utilizando uma parâmetro de triaxialidade mais geral e aplicável a condições de escoamento generalizado (LSY). Utilizando análises numéricas detalhadas empregando o modelo MBL, O'Dowd e Shih identificaram famílias de campos elasto-plásticos de tensões parametrizáveis na forma:

$$\sigma_{ij} = \sigma_0 \hat{f}_{ij} \left(\frac{r}{J/\sigma_0}, \theta, Q \right) , \quad (2)$$

onde σ_0 é a tensão de referência ou tensão de escoamento e o parâmetro adimensional Q define a quantidade pela qual o campo de tensões σ_{ij} em configurações estruturais difere da solução

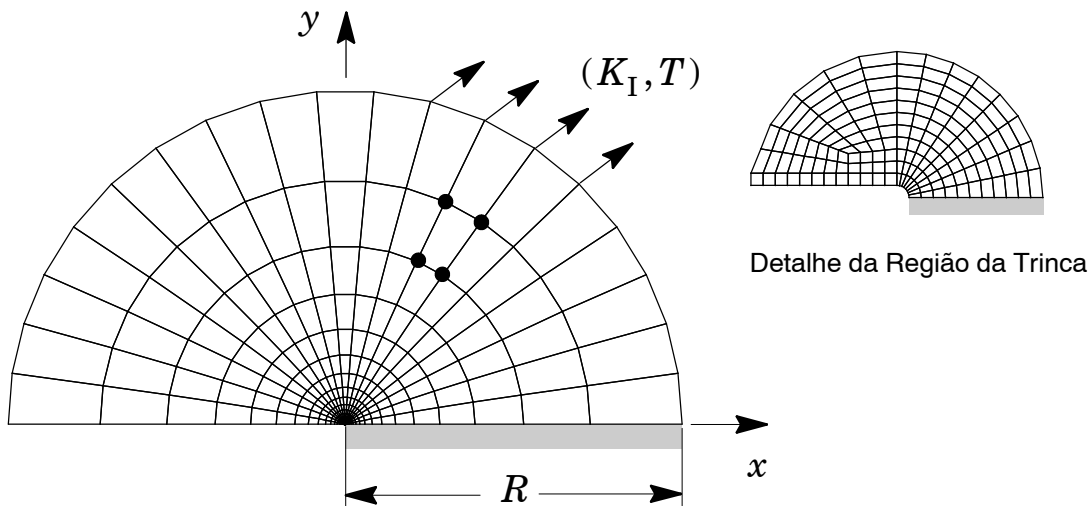


Figura 3 Modelo SSY com campos (K, T) aplicados no seu contorno.

referência SSY com $T = 0$. Desta forma, Q efetivamente *quantifica* os efeitos de geometria, tamanho da trinca, modo de carregamento, etc.

Considerando o campo de tensões no setor angular à frente da trinca para a solução referência SSY $_{T=0}$ (*i.e.*, a formulação MBL descrita pela Eq. (1) com $T = 0$) e a configuração estrutural, O'Dowd e Shih mostraram que $Q\sigma_0$ corresponde efetivamente a uma tensão de natureza hidrostática espacialmente uniforme. Conseqüentemente, Q representa a diferença (escalar) de tensões relativa a um campo de elevada triaxialidade dentro de uma região microestruturalmente significativa

$$\sigma_{ij} = (\sigma_{ij})_{\text{SSY}; T=0} + Q\sigma_0\delta_{1i}\delta_{1j}; \quad |\theta| < \frac{\pi}{2}, \quad J/\sigma_0 < r < 5J/\sigma_0. \quad (3)$$

Operacionalmente, Q é definido como

$$Q \equiv \frac{\sigma_{\theta\theta} - (\sigma_{\theta\theta})_{\text{SSY}; T=0}}{\sigma_0}, \quad \text{para } \theta = 0, \quad r = 2J/\sigma_0 \quad (4)$$

onde análises numéricas utilizando o método de elementos finitos sobre modelos suficientemente refinados para a região microestruturalmente significativa fornecem os campos de tensões elasto-plásticas necessárias (modelo MBL e a configuração estrutural) para determinação da diferença de tensões. Na Eq. (3), Q é convenientemente avaliado em $r = 2J/\sigma_0$; entretanto, O'Dowd e Shih também mostraram que Q é relativamente independente de r para valores $J/\sigma_0 < r < 5J/\sigma_0$.

A construção das *trajetórias* J - Q deriva diretamente da resolução da Eq. (3) para cada estágio (nível) de carregamento no corpo finito. Tal procedimento não impõe, em princípio, nen-

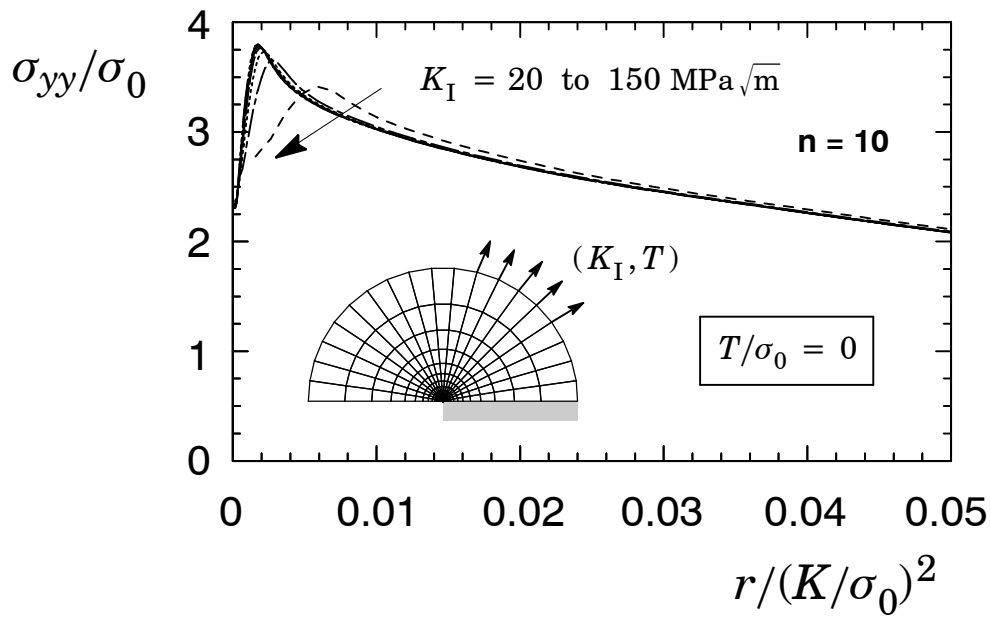
hum tipo de restrição sobre a descrição constitutiva do material ou formulação numérica de plasticidade e deformações. Desta forma, formulações incrementais de plasticidade incorporando grandes deformações podem ser utilizadas para descrever com maior precisão o campo elasto-plástico de tensões na região da trinca. Entretanto, é computacionalmente mais simples empregar análises baseadas sobre a teoria de pequenas deformações como apresentado nas referências [17-19].

2.3 Soluções de Referência Utilizando o Modelo MBL

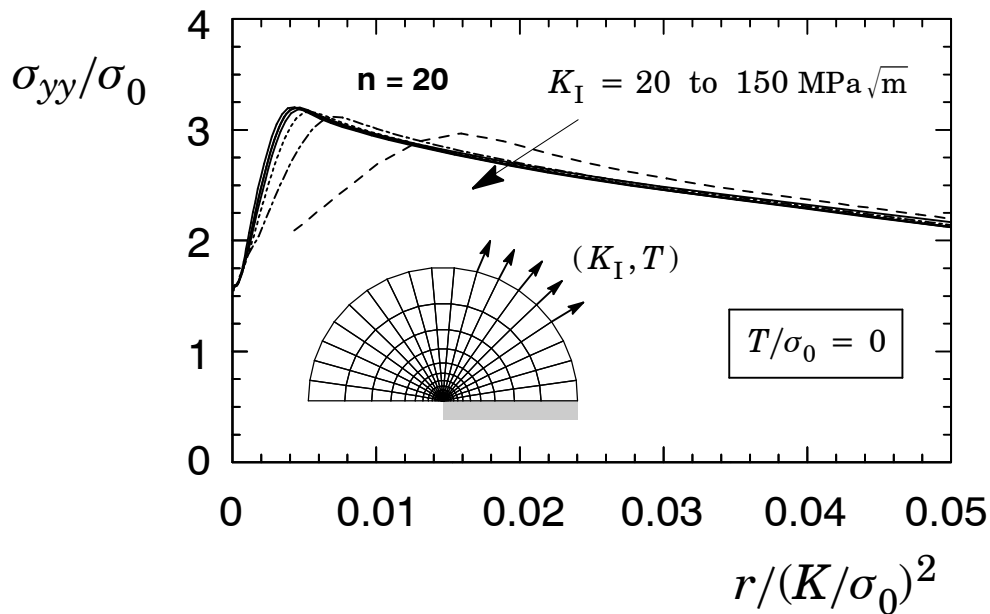
A solução de referência utilizada para determinar o parâmetro Q utiliza o campo de tensões obtido a partir de análises numéricas para o modelo MBL apresentado na Fig. 3. Este modelo computacional representativo de uma placa/trinca *infinita* simplifica a geração de soluções numéricas para trincas estacionárias sob condições de escoamento limitado (*SSY*) completamente definidas. A simetria do carregamento (Modo I) e das condições de contorno permite a análise de somente metade da placa infinita como mostrado na Fig. 3. Estudos numéricos conduzidos por Trovato e Ruggieri [20,21] demonstram que estes campos de tensões são campos estacionários independentes do nível de carregamento (medido convenientemente por K_I ou J) quando predominarem as condições de escoamento limitado (condições *SSY*) em uma pequena região ao redor da ponta da trinca.

As Figs. 4(a-b) apresentam a evolução da tensão de abertura (σ_{yy}) normalizada pela tensão de referência (σ_0) com a distância da ponta da trinca normalizada por $(K/\sigma_0)^2$ para dois materiais elasto-plásticos descritos por um modelo de potência (derivado do modelo de Ramberg-Osgood [4,22]) da forma $(\epsilon/\epsilon_0) = (\sigma/\sigma_0)^n$ com $n = 10$ e $n = 20$. O modelo de elementos finitos para estas análises utiliza uma malha convencional com anéis focais na região da trinca, como mostrado na Fig. 3. A limitação da plasticidade na ponta da trinca impõe a construção do modelo com raio externo, R , suficientemente grande tal que $R_p < R/20$, onde R_p é o raio da zona plástica (ver Fig. 3). A ponta da trinca é modelada através de uma configuração adequada à acomodação do arredondamento inicial (*blunted tip*) com raio $\rho_0 = 2.5\mu\text{m}$ (0.0025 mm); esta dimensão permite obter os campos de tensões e deformações com o elevado grau de precisão necessário às análises e, ao mesmo tempo, acelera a convergência dos algoritmos de plasticidade utilizados nas soluções numéricas. Este modelo *SSY* possui uma camada de ~ 2000 elementos isoparamétricos (8 nós) com condições correspondentes ao estado plano de deformação ($w = 0$) impostos sobre cada nó.

Como pode ser observado nas Figs 4(a-b), após um regime transitório inicial ($K \approx 20 \sim 40 \text{ MPa}\sqrt{\text{m}}$), os campos de tensões *independem* do nível de carregamento aplicado e, conseqüentemente, podem ser associados a um estado de tensões de *referência* correspondentes à uma placa



(a)



(b)

Figura 4 Campos de tensões para o modelo MBL em dois materiais elasto-plásticos [20,21].

infinita (representada pelo modelo MBL no presente contexto). Estas soluções de referência dependem das propriedades de encruamento do material: para o material com $n = 10$, o valor de tensão máxima é cerca de $3.8 \times \sigma_0$, enquanto para o material com $n = 20$ o valor de tensão máxima é cerca de $3.2 \times \sigma_0$.

2.4 Extensão da Teoria J-Q para Propagação de Trincas e Curvas J-R

Aços estruturais e aços para vasos de pressão geralmente exibem substancial aumento da tenacidade à fratura, caracterizada pela Integral J [3,23], sobre os milímetros iniciais ($\Delta a \approx 3 \sim 5$ mm) de propagação estável (extensão dúctil) de uma trinca. Este crescimento de trinca é freqüentemente acompanhado por grande aumento da plasticidade do material em relação a uma trinca estacionária, como ilustrado esquematicamente na Fig. 5. Para materiais possuindo maior tenacidade e elevada resistência à fratura frágil (ver discussão sobre o mecanismo de clivagem transgranular em Ruggieri [25]), o desenvolvimento de deformações intensas e tensões elevadas sobre regiões diretamente à frente da trinca geralmente produzem extensão e rasgamento dúcteis antes da sua propagação instável por clivagem. Tal extensão dúctil da trinca altera a evolução dos campos de tensões (em relação a uma trinca estacionária) e o volume da zona de processo de fratura [24, 26] com potencial impacto sobre o comportamento à fratura frágil do material. Consequentemente, inclusão dos efeitos do crescimento estável da trinca sobre a descrição da evolução dos campos de tensões com o aumento do carregamento (caracterizado pela Integral J) torna-se altamente relevante dentro do presente contexto, particularmente em relação à evolução do parâmetro hidrostático Q como descritor do nível de restrição plástica para trincas não estacionárias.

2.4.1 Modelagem de Crescimento de Trincas Utilizando Células Computacionais

A descrição mecanística da fratura dúctil em metais (ver, *e.g.*, Ruggieri [27]) permite identificá-la como um modo de falha que incorpora a combinação de vários mecanismos simultâneos e operativos em escala microestrutural convenientemente divididos em: a) nucleação de microcavidades resultantes da fratura ou separação de inclusões; b) crescimento subsequente de microcavidades suficientemente grandes e separadas uma das outras (microcavidades pequenas e próximas coalescem formando uma cavidade maior); c) localização de deformações plásticas e d) coalescência final das microcavidades.

Observações experimentais e estudos numéricos [28-34] suportam a simplificação deste processo múltiplo objetivando o desenvolvimento de um modelo micromecânico tratável para utilização em métodos preditivos de integridade estrutural. Em aços estruturais convencionais (por exemplo, aços para vasos de pressão), a nucleação de microcavidades (devido à fratura ou decoesão de partículas e inclusões maiores) ocorre sob níveis de tensões relativamente baixos em comparação às máximas tensões que desenvolvem-se na região material à frente da trinca. Tal comportamento fenomenológico permite identificar a população de inclusões simplesmente como uma distribuição de cavidades pré-existentes sem introduzir grandes efeitos sobre a re-

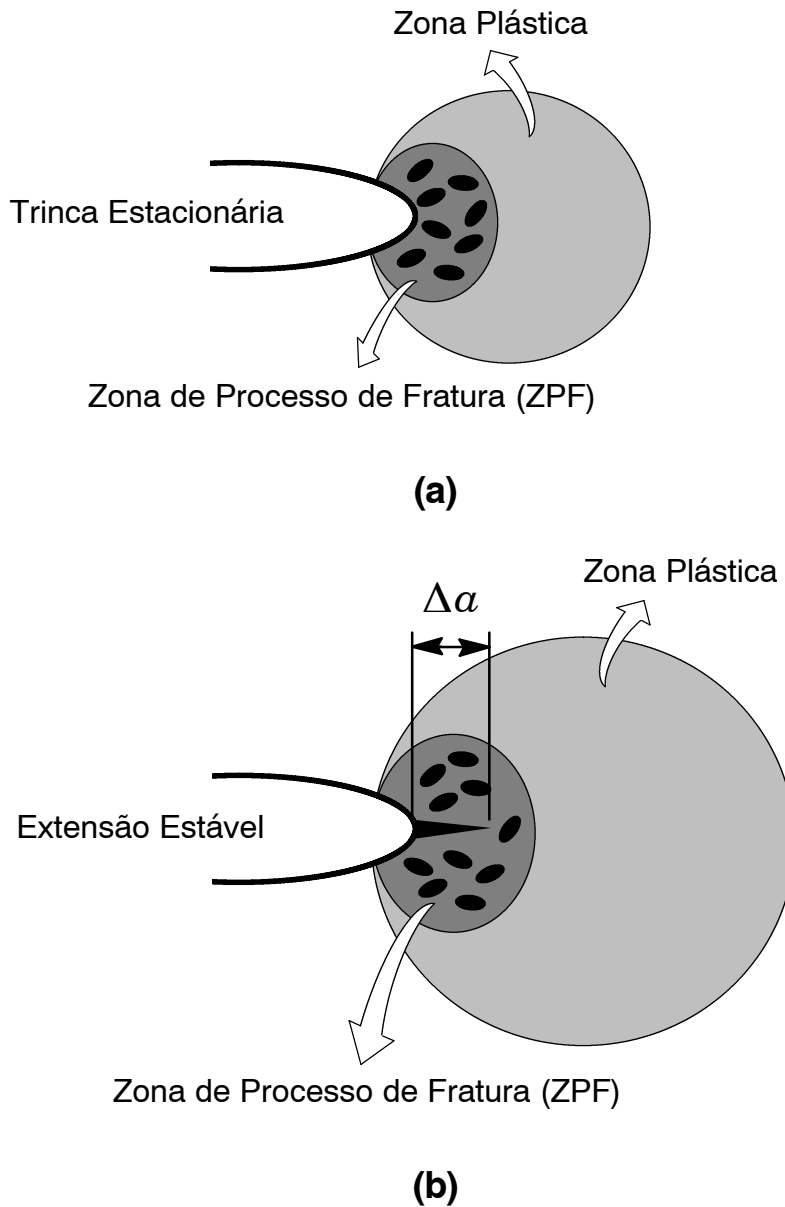
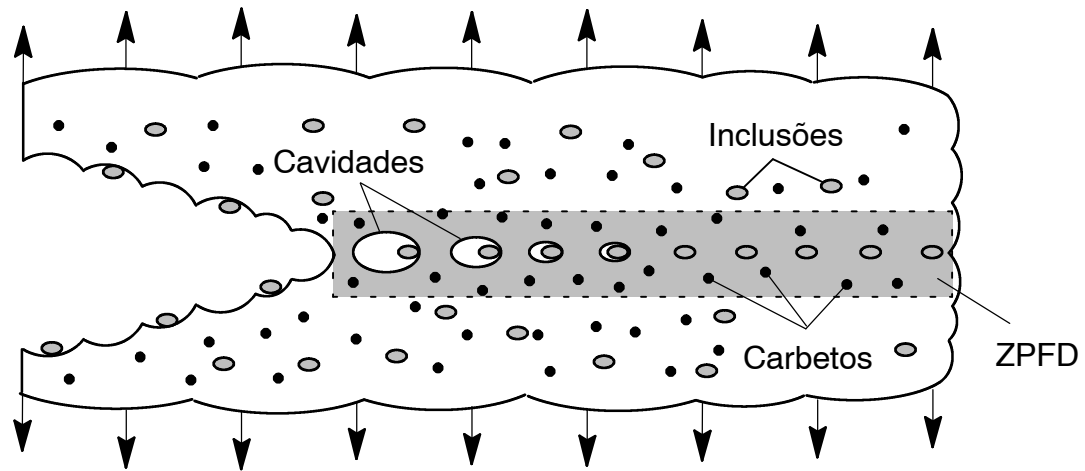


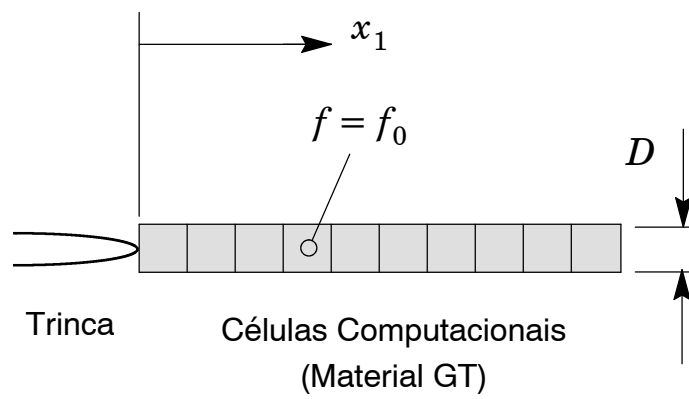
Figura 5 Ilustração esquemática do efeito do aumento da plasticidade na região da trinca decorrente do aumento de J e da extensão dútil em relação a uma trinca estacionária: (a) Trinca estacionária; (b) Extensão estável da trinca.

sistência à fratura dúctil do material. Adicionalmente, os mecanismos de localização e coalescência de cavidades (por exemplo, nucleação e crescimento de partículas menores as quais geralmente intervêm no processo de fratura dúctil somente nos estágios finais de coalescência ocorrem somente sob elevados níveis de plasticidade, normalmente na região imediatamente próxima à ponta trinca (região de arredondamento da ponta da trinca ou *blunting region*) onde o dano material já é bastante elevado.

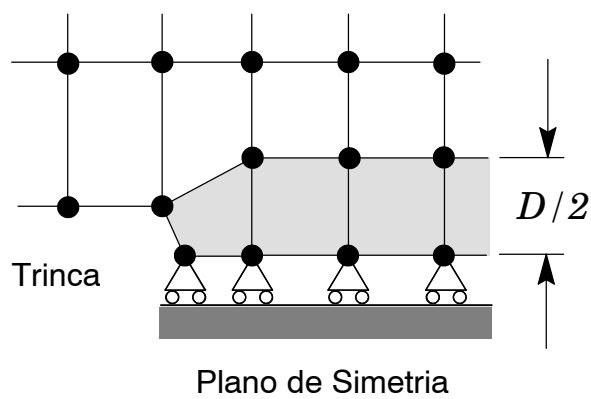
Motivados pelos argumentos mecanísticos apresentados acima, Xia and Shih [30-32] (X&S) propuseram um modelo incorporando um mecanismo realista de crescimento de microcavi-



(a)



(b)



(c)

Figura 6 Modelagem da extensão dúctil de trincas utilizando células computacionais.

dades através do conceito de *células computacionais*, o qual foi implementado em um contexto 3D por Ruggieri e Dodds [35]. A Figura 6(a) ilustra o processo de crescimento de uma trinca em um material dúctil tal como um aço ferrítico operando na região superior da curva de transição frágil-dúctil. A camada de material diretamente associada à propagação da trinca macroscópica define a *Zona de Processo da Fratura Dútil* (ZPFD). Esta zona material deve possuir suficiente espessura para incluir pelo menos uma microcavidade; tal dimensão também possibilita introduzir a *dimensão linear* necessária ao modelo. O crescimento e a coalescência da cavidade dentro da ZPFD (eventos que representam o dano material dentro do presente contexto) provocam a redução das forças que esta camada exerce sobre o material adjacente. Sob a ação de esforços externos (veja Fig. 6(a)), o aumento do dano na ZPFD provoca a nulificação destas forças com conseqüente avanço da trinca. A modelagem deste processo utilizando o conceito de *células computacionais* proposto por X&S considera o mecanismo de crescimento de cavidades confinado dentro de uma camada material simetricamente localizada no plano da trinca macroscópica como ilustrado na Fig. 6(b). A camada material possui espessura D , onde D está associado com o espaçamento médio das (grandes) inclusões responsáveis pela nucleação das cavidades. O presente modelo considera ainda a discretização desta camada em células cúbicas possuindo fração volumétrica inicial f_0 , onde f_0 é definido como a relação entre o volume inicial da cavidade e o volume da célula.

Numericamente, o crescimento progressivo da cavidade e o subseqüente dano material em cada célula são descritos pelo modelo constitutivo de Gurson [36] e modificado por Tvergaard [37] (denominado simplifcamente de modelo GT) na forma

$$g(\sigma_e, \sigma_m, \bar{\sigma}, f) = \left(\frac{\sigma_e}{\bar{\sigma}}\right)^2 + 2q_1 f \cosh\left(\frac{3q_2 \sigma_m}{2\bar{\sigma}}\right) - (1 + q_3 f^2) = 0 \quad (5)$$

onde σ_e representa a tensão (macroscópica) efetiva de Von Mises, σ_m é a tensão (macroscópica) média, $\bar{\sigma}$ é a tensão (plástica) do material da célula e f define a fração volumétrica corrente. Os fatores q_1 , q_2 e q_3 introduzidos por Tvergaard [37] servem como parâmetros de ajuste do modelo e são geralmente adotados como: $q_1 = 1.25$, $q_2 = 1.0$ e $q_3 = q_1^2$. Trabalhos recentes de Faleskog e Shih [33, 34] fornecem valores adicionais dos fatores q_1 , q_2 e q_3 em função das propriedades elasto-plásticas do material. O avanço e propagação da trinca ocorre quando f na célula imediatamente adjacente à ponta da trinca atinge um valor crítico, f_E .

A Fig. 6(c) mostra a representação típica 2D do modelo de células computacionais utilizando elementos finitos. O material externo à ZPFD obedece um modelo elasto-plástico convencional (modelo constitutivo de von Mises) e permanece sem dano durante toda a história de carregamento. Os parâmetros-chave D e f_0 são *calibrados* utilizando-se medições experimentais da

fratura em materiais dúteis por intermédio de corpos-de-prova laboratoriais (geralmente configurações pré-trincadas C(T) ou SE(B), ou ainda barras de seção cilíndrica submetidas a esforços uniaxiais de tração contendo um entalhe circular em sua seção central). Por intermédio de uma série de análises numéricas não lineares (Método de Elementos Finitos), os parâmetros D e f_0 são ajustados de forma que os resultados computacionais descrevam as medições experimentais. Uma vez determinados desta maneira, os parâmetros D e f_0 são assumidos como propriedades constantes do material para utilização em análises de integridade estrutural de componentes com diferentes geometria e submetidos a diferentes condições de carregamento. Ruggieri [27] e Ruggieri e Dodds [35] apresentam detalhes adicionais sobre a calibração dos parâmetros D e f_0 para aços estruturais.

2.4.2 Extensão do Parâmetro Q à Análise de Trincas Não Estacionárias

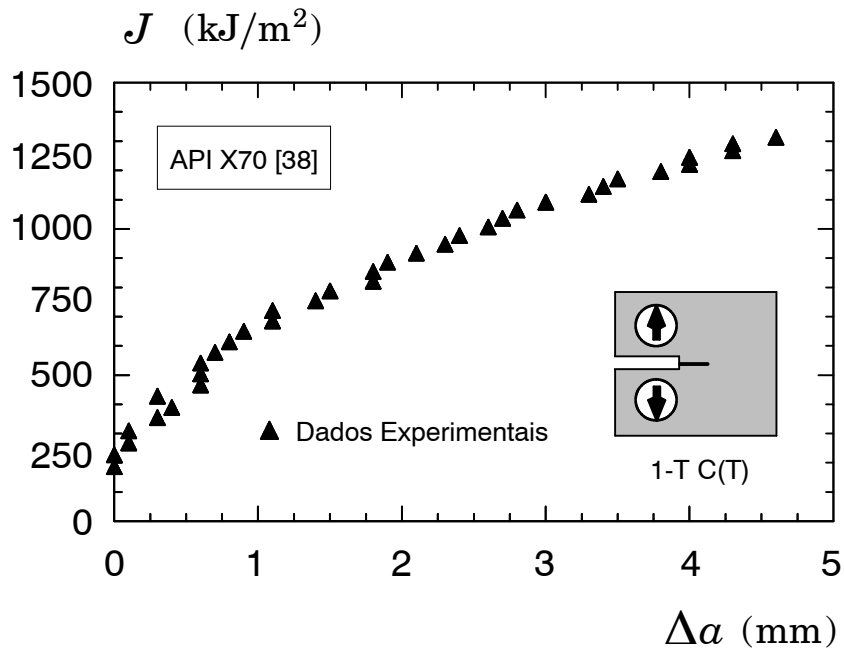
O conceito do parâmetro hidrostático Q introduzido anteriormente pode ser estendido de forma simples e direta à descrição da evolução dos campos de tensões à frente de trincas não estacionárias (*i.e.*, crescimento estável de trincas). A Fig. 7(a) ilustra uma curva de *resistência* à propagação da fratura J vs. Δa , também denominada curva R ou curva $J-R$ [4], típica para um aço estrutural API X70 testado por Hippert [38]. Com o aumento da propagação estável, Δa , a distribuição (*instantânea* ou *corrente*) de tensões de abertura, σ_{yy} , também avança à frente da trinca como apresentado na Fig. 7(b); aqui, a tensão de abertura é normalizada pela tensão de referência (ou tensão de escoamento, σ_0), enquanto a distância à ponta da trinca, x , é normalizada pelo tamanho da célula computacional, D , utilizada na análise do crescimento da trinca por intermédio do modelo de Gurson-Tvergaard descrito anteriormente. Desta forma, Q pode ser redefinido como a diferença (escalar) de tensões correspondente a uma trinca não estacionária relativa a um campo de elevada triaxialidade dentro de uma região microestruturalmente significativa na forma

$$Q \equiv \frac{\sigma_{\theta\theta}^{\Delta a} - (\sigma_{\theta\theta})_{SSY; T=0; \Delta a=0}}{\sigma_0}, \text{ para } \theta = 0, \hat{r} = 2J/\sigma_0 \quad (6)$$

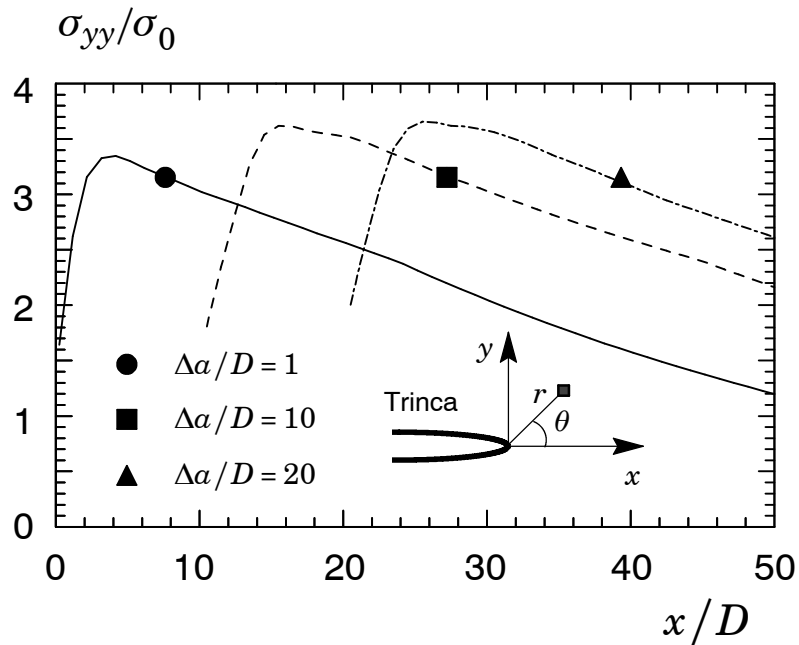
onde \hat{r} agora define a distância *instantânea* (ou *corrente*) normalizada em relação à ponta da trinca em propagação.

2.5 Procedimentos Computacionais

A geração das trajetórias $J-Q$ para o componente estrutural sob análise é definida simplesmente por intermédio da diferença entre os valores das tensões de abertura (Modo I de carrega-



(a)



(b)

Figura 7 (a) Curva típica de resistência J - R para um aço estrutural API X70 obtida experimentalmente por Hippert [38]; (b) Distribuição de tensões de abertura com o avanço da trinca correspondente ao corpo-de-prova C(T) para o aço API X70 testado por Hippert [38].

mento) obtidas sobre uma distância pré-especificada da ponta da trinca para o componente estrutural e a placa infinita de referência (modelo MBL). A Fig. 8 apresenta uma distribuição de tensões de abertura (σ_{yy} no presente exemplo) característica de um componente estrutural pré-trincado. As curvas sólidas do gráfico representam a variação da tensão σ_{yy} normalizada pela

tensão de referência ou de escoamento (σ_0) do material em função da distância da ponta da trinca normalizada por J/σ_0 para valores crescentes de carga medida pela Integral J . A curva tracejada corresponde à distribuição de tensões para uma placa *infinita* obtida através da análise de um modelo MBL (*modified boundary layer model*); como introduzido anteriormente, este modelo representa a condição de um campo de *alta triaxialidade* adotado como solução *referência*.

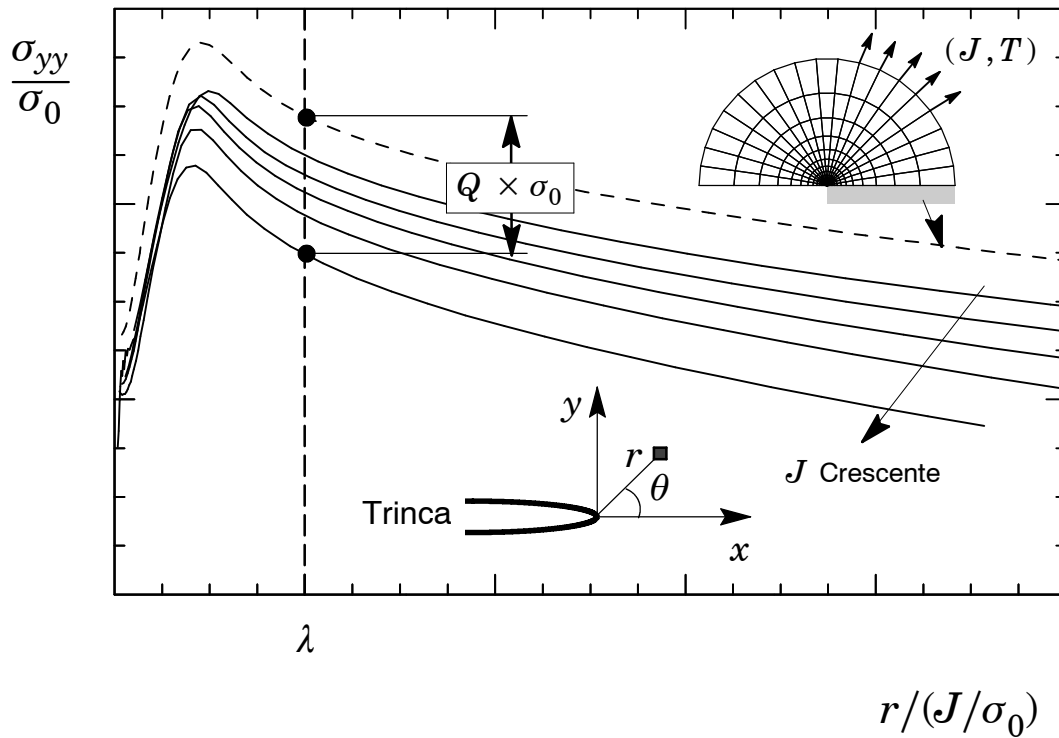


Figura 8 Distribuição de tensões na região da ponta da trinca para diferentes níveis de carregamento (expresso pela integral J)

Dentro do contexto metodológico definido pela teoria J - Q , o parâmetro Q é operacionalmente definido no programa *Fractus2D* para a coordenada $\theta = 0$ como

$$Q \equiv \frac{(\sigma_{yy})_{FB} - (\sigma_{yy})_{SSY;T=0}}{\sigma_0}, \quad \text{para } \theta = 0, r/(J/\sigma_0) = \lambda \quad (7)$$

onde o subscrito *FB* denota o componente estrutural e *SSY-T=0* denota a placa infinita representada pelo modelo MBL com tensão T nula. Em geral, o parâmetro Q é avaliado convencionalmente para $\lambda = 2$, embora outros valores de λ possam ser especificados em *Fractus2D* como será detalhado adiante.

2.6 Trajetória J-Q para Componentes Estruturais Pré-Trincados

Para ilustrar o procedimento, a Fig. 9 mostra diferentes corpos-de-prova convencionais e a Fig. 10 apresenta as correspondentes trajetórias J - Q para um material elasto-plástico descrito por um modelo de potência (Ramberg-Osgood) da forma $(\epsilon/\epsilon_0) = (\sigma/\sigma_0)^n$ com $n = 10$ e 20 [19]. O parâmetro Q é obtido utilizando a Eq. (3) para o ponto à frente da trinca definido por $r/(J/\sigma_0) = 2$. Para o corpo-de-prova C(T) com $a/W = 0.6$, os valores de Q são positivos para praticamente toda a faixa de carregamento (caracterizado pela Integral J) correspondendo a tensões elásticas T positivas para esta geometria. Em contraste, o espécime M(T) apresenta valores negativos do parâmetro Q quase que imediatamente após o início do carregamento, caracterizando uma severa perda de restrição na ponta da trinca (*constraint loss*) para esta configuração.

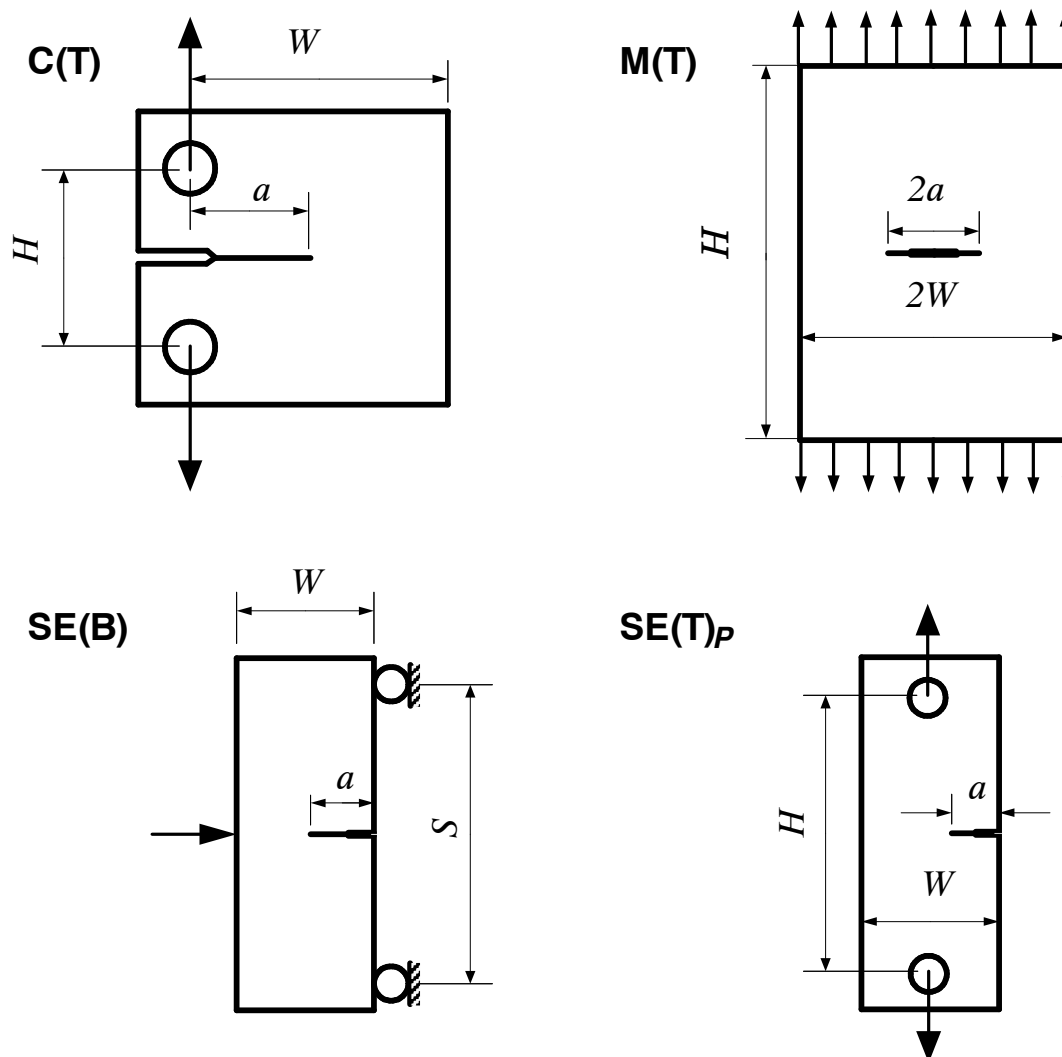
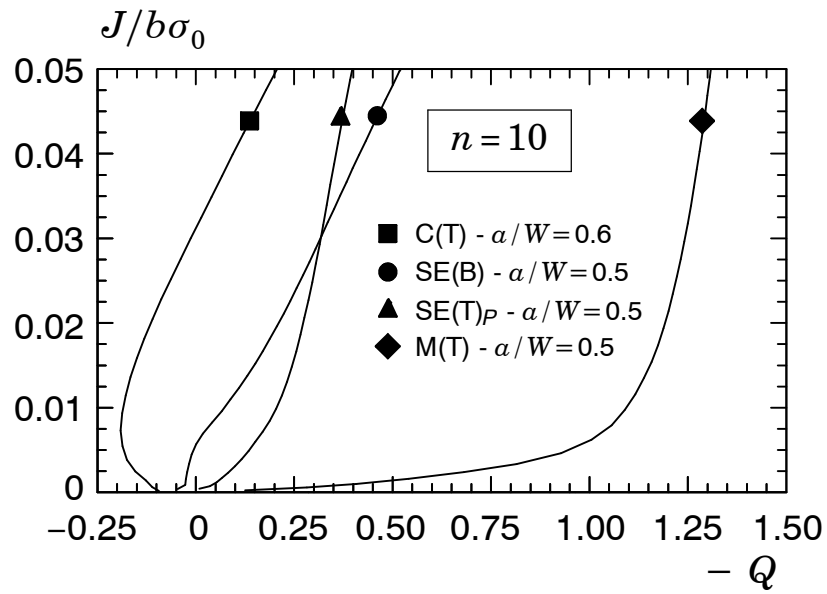
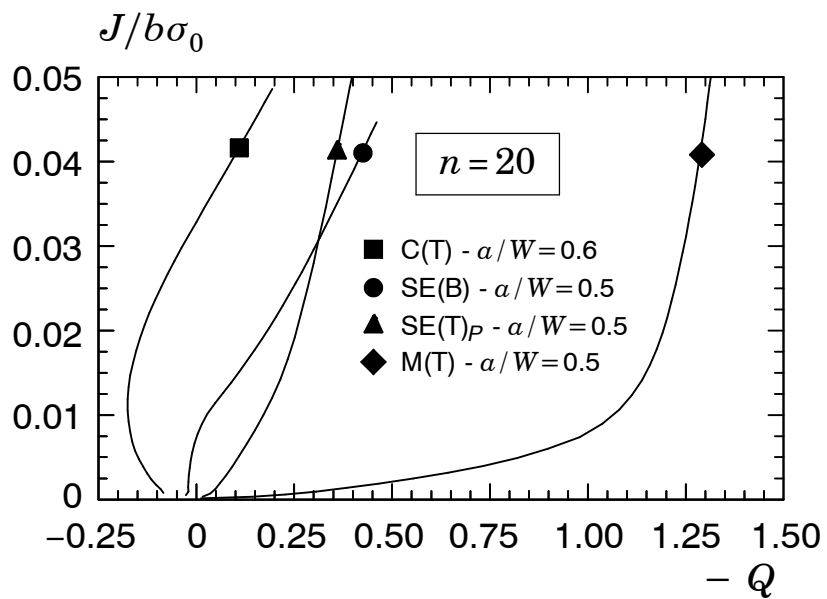


Figura 9 Espécimes convencionais C(T), M(T), SE(B) e SE(T)_P [19].



(a)



(b)

Figura 10 Trajetórias J - Q para espécimes convencionais e material com $n = 10$ e 20 [19].

A Metodologia *ETA* para Determinação dos Parâmetros J e CTOD

3

3.1 Fatores *Eta* para Determinação da Integral J em Trincas Estacionárias

A interpretação energética da Integral J como a taxa de liberação de energia para um sólido com espessura unitária contendo uma trinca ilustrado na Fig. 11(a) fornece a base para avaliação de J utilizando medições experimentais de carga (força) *vs.* deslocamento da linha de carga. Assumindo teoria de deformação plástica, a Integral J é expressa por [39,40]

$$J = \int_0^P \left(\frac{\partial \Delta}{\partial a} \right)_P dP = - \int_0^\Delta \left(\frac{\partial P}{\partial a} \right)_\Delta d\Delta \quad (8)$$

onde P é a carga aplicada, Δ é o deslocamento da linha de carga (ou LLD – *load line displacement*) e a é a dimensão da trinca. As integrais da equação acima correspondem a condições de controle de carga e controle de deslocamento.

Considerando as componentes elástica, Δ_e , e plástica, Δ_p , do deslocamento da linha de carga na forma

$$\Delta = \Delta_e + \Delta_p \quad (9)$$

e manipulando a primeira integral da Eq. (8), J pode ser expresso como

$$J = \int_0^P \left(\frac{\partial \Delta_e}{\partial a} \right) dP + \int_0^P \left(\frac{\partial \Delta_p}{\partial a} \right) dP = J_e + J_p \quad (10)$$

onde J_e e J_p são as suas componentes elástica e plástica. No presente contexto, J_e é convenientemente definido pela taxa de liberação de energia para um sólido elástico linear contendo uma trinca sob Modo I de deformação na forma convencional

$$J_e = \frac{K_I^2}{E'} \quad (11)$$

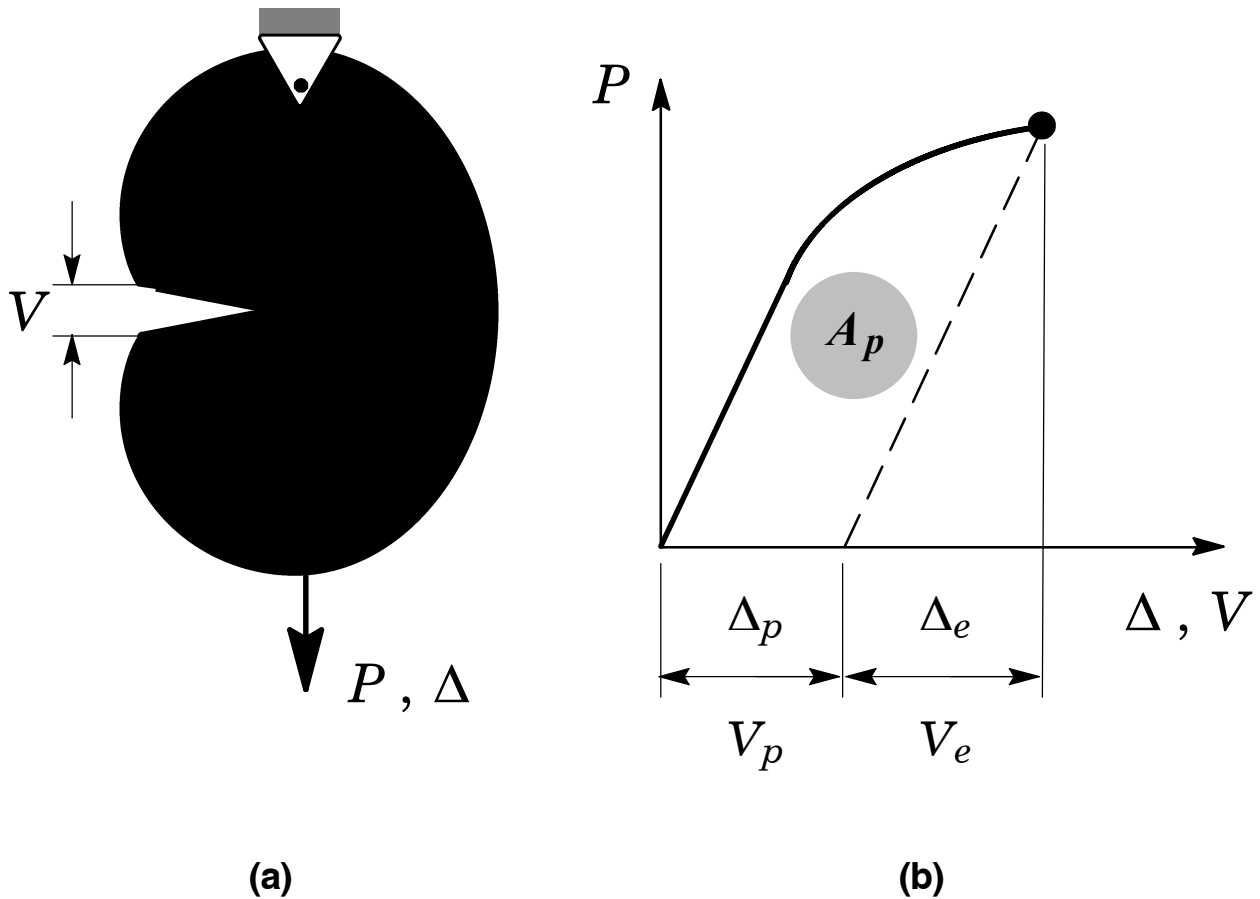


Figura 11 (a) Sólido com espessura unitária contendo uma trinca submetido a carregamento remoto; (b) Definição da componente plástica da área sob a curva carga vs. deslocamento.

onde K_I denota o fator elástico de intensidade de tensões (Modo I) para a configuração contendo a trinca, $E' = E/(1 - \nu^2)$ com E e ν representando o módulo de elasticidade longitudinal (módulo de Young) e o coeficiente de Poisson.

A componente plástica, J_p , é derivada a partir da abordagem proposta por Sumpter and Turner [41] utilizando trabalho anterior de Rice [42] a qual fornece a conexão entre a Integral J e a área sob a curva carga vs. deslocamento da linha de carga. A Fig. 11(b) ilustra os aspectos essenciais da metodologia para a estimativa de J_p . O procedimento simplesmente relaciona a contribuição plástica à energia de deformação do sólido (devido à presença da trinca) e J na forma

$$J_p = \int_0^P \left(\frac{\partial \Delta_p}{\partial \alpha} \right) dP = \frac{\eta_J^{\text{LLD}}}{b} \int_0^{\Delta_p} P d\Delta_p = \frac{\eta_J^{\text{LLD}} A_p^{\text{LLD}}}{b} \quad (12)$$

onde A_p^{LLD} é a área plástica sob a curva carga *vs.* deslocamento da linha de carga e b é o ligamento remanescente da trinca definido por $(W - a)$. No desenvolvimento da Eq. (12), a igualdade $da = -db$ foi usada. O factor η_J^{LLD} introduzido por Sumpter and Turner [41] representa um parâmetro adimensional que relaciona a contribuição plástica à energia de deformação do sólido (devido à presença da trinca) e J sendo considerado uma função da configuração e geometria do sólido contendo a trinca e independente do carregamento [39].

A análise apresentada acima também pode ser estendida quando o comportamento mecânico do sólido (e, conseqüentemente, a área plástica sob a curva de carga *vs.* deslocamento) ilustrado na Fig. 11(b) for definido em termos da abertura da boca da trinca, V (ou CMOD – *crack mouth opening displacement*). A componente plástica J_p resulta de forma análoga à Eq. (12) como

$$J_p = \frac{\eta_J^{CMOD} A_p^{CMOD}}{b} \quad (13)$$

onde A_p^{CMOD} é a área plástica sob a curva carga *vs.* abertura da boca da trinca e o fator η_J^{CMOD} relaciona agora a contribuição plástica à energia de deformação do sólido (devido à presença da trinca) e J em termos de medições experimentais de CMOD .

As Eqs. (12) e (13) anteriores são igualmente aplicáveis a corpos-de-prova convencionais (e também não convencionais) com espessura B . As expressões para a componente plástica, J_p , resultam portanto

$$J_p = \frac{\eta_J^{LLD} A_p^{LLD}}{Bb} \quad (14)$$

e

$$J_p = \frac{\eta_J^{CMOD} A_p^{CMOD}}{Bb} \quad (15)$$

Embora ambas as definições para os fatores *eta* descrevam de forma essencialmente similar o efeito da energia de deformação plástica sobre a Integral J , o fator η_J baseado sobre medições de LLD possui um carácter diferente do fator η_J correspondente baseado em medições experimentais de CMOD. Por exemplo, Cravero and Ruggieri [43] e Ruggieri [44] discutem este aspecto com maiores detalhes e fornecem um extensivo conjunto de resultados para fatores *eta* obtidos a partir de valores de LLD e CMOD para corpos-de-prova SE(T).

3.2 Fatores *Eta* para Determinação do CTOD em Trincas Estacionárias

A definição do fator η correspondente ao parâmetro elasto-plástico CTOD, denotado η_δ no presente contexto, deriva de desenvolvimento similar àquele apresentado na seção anterior. Se-

guindo a análise prévia para a Integral J e utilizando a conexão entre J e o deslocamento da ponta da trinca (δ) expresso por [45,46]

$$\delta = \frac{J}{m\sigma_{ys}} \quad (16)$$

na qual m é uma constante adimensional, uma expressão formalmente análoga à Eq. (9) caracteriza a força motriz da trinca em termos de CTOD na forma

$$\delta = \delta_e + \delta_p \quad (17)$$

onde δ_e e δ_p representam as componentes elástica e plástica de CTOD.

Similarmente à análise da seção precedente, δ_e é convenientemente relacionado com o fator de intensidade de tensões sob Modo I de deformação como

$$\delta_e = \frac{K_I^2}{m_{ssy}\sigma_{ys}E'} \quad (18)$$

onde m_{ssy} é um fator de restrição plástica sob condições de escoamento de pequena monta o qual comumente assume valores entre 1 e 2 [4].

Conseqüentemente, a componente plástica, δ_p é definida por intermédio do fator η na forma

$$\delta_p = \frac{\eta_{\delta}^{LLD} A_p^{LLD}}{Bb\sigma_{ys}} \quad (19)$$

e

$$\delta_p = \frac{\eta_{\delta}^{CMOD} A_p^{CMOD}}{Bb\sigma_{ys}} \quad (20)$$

onde novamente A_p^{CMOD} (A_p^{LLD}) é a área plástica sob a curva carga *vs.* abertura da boca da trinca (deslocamento da linha de carga) e o fator η_{δ}^{CMOD} (η_{δ}^{LLD}) relaciona a contribuição plástica à energia de deformação do sólido (devido à presença da trinca) e CTOD em termos de medições experimentais de CMOD (LLD) como ilustra a Fig. 12. Frequentemente, a tensão de escoamento, σ_{ys} , nas expressões (19) e (20) é substituída pela tensão de fluxo (*flow stress*), σ_f , definida por $\sigma_f = (\sigma_{ys} + \sigma_{uts})/2$, onde σ_{uts} é o limite de resistência do material. Tal procedimento objetiva essencialmente introduzir de forma simplificada o efeito do encruamento do material sobre o valor do CTOD.

Analogamente à seção anterior, ambas as definições para os fatores *eta*, embora possuam caráter diferente, descrevam de forma essencialmente similar o efeito da energia de defor-

mação plástica sobre o CTOD. Entretanto, o parâmetro elasto-plástico CTOD é normalmente obtido por intermédio de medições experimentais de CMOD devido à sua maior simplicidade operacional (por exemplo, medições experimentais de CTOD utilizando espécimes flexão 3 pontos SE(B) ilustrado na Fig. 12).

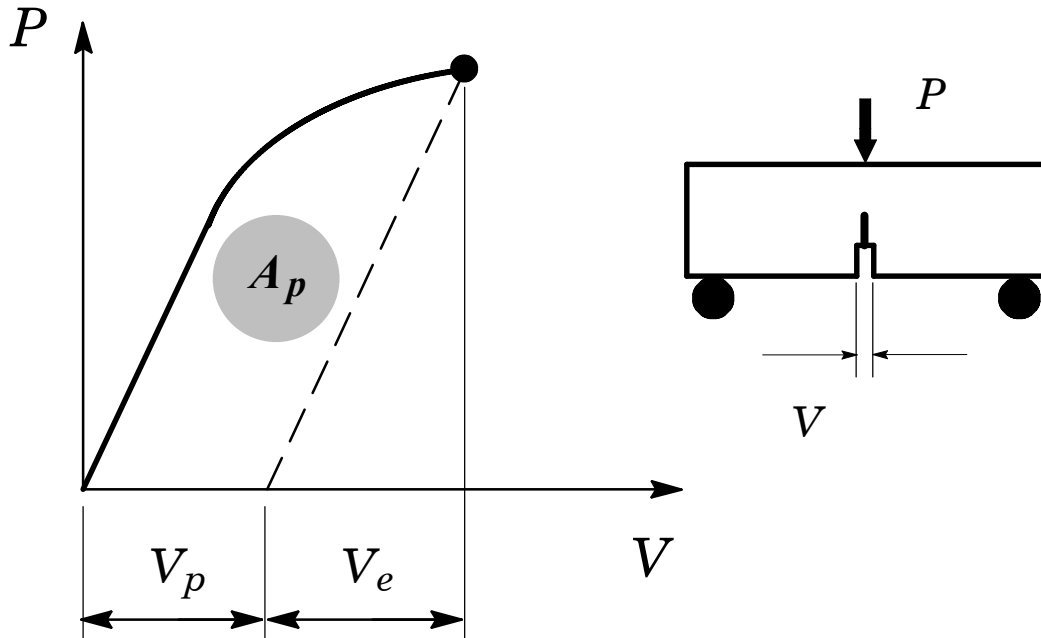


Figura 12 Esquema para determinação do CTOD baseado na metodologia η e medições experimentais de CMOD.

3.3 Procedimentos Computacionais

A implementação computacional para a determinação do fator η envolve essencialmente o tratamento adequado de resultados numéricos de forças e deslocamentos obtidos a partir das análises de elementos finitos (não são necessários os resultados de tensões e deformações para o componente ou corpo-de-prova analisado). Entretanto, uma vez que estes resultados são disponibilizados em incrementos discretos de carregamento (*load steps*), o procedimento numérico incorpora alguns aspectos adicionais para garantir uma precisão mais elevada do fator η , particularmente em relação à determinação da componente plástica da área sob a curva carga *vs.* deslocamento, cálculo do valor de η e determinação do CTOD para o modelo numérico. As seções seguintes discutem estes procedimentos de forma mais detalhada.

3.3.1 Área Plástica sob a Curva Carga vs. Deslocamento

Um ingrediente-chave à correta e acurada determinação do fator η para o componente ou espécime sob análise é a avaliação da área total, A_t , sob a curva P vs. CMOD ou LLD para um nível de carregamento especificado (*load step*) e subsequente separação em suas componentes elástica, A_e , e plástica, A_p . A Fig. 13 ilustra o procedimento adotado o qual é sumarizado a seguir:

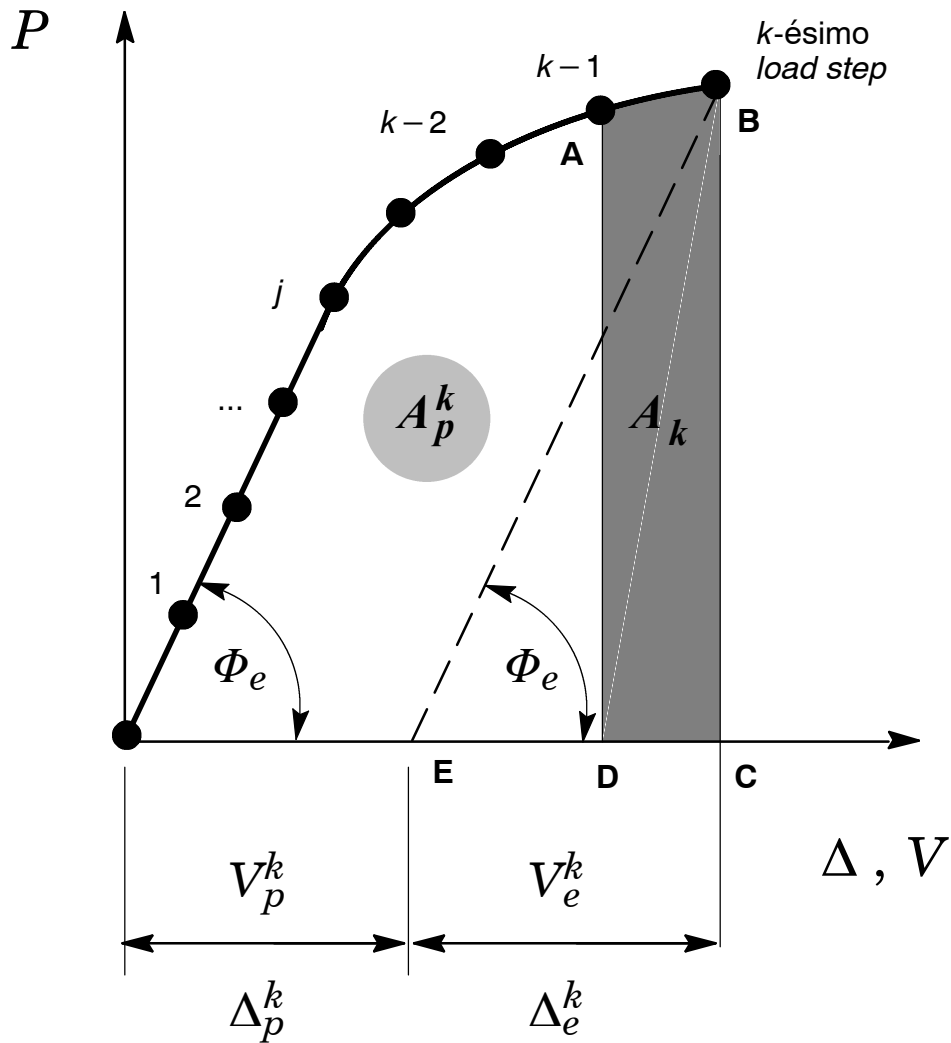


Figura 13 Esquema adotado para determinação da área plástica, A_p , sob a curva carga vs. deslocamento.

- i) Determinação do coeficiente angular, Φ_e , correspondente à região elástica da curva carga vs. deslocamento. Idealmente, uma análise elástica para o componente ou espécime deveria ser utilizada para determinar a relação (elástica) entre P e CMOD (LLD). De forma a simplificar o procedimento para determinação de A_p e utilizar os resultados numéricos (elasto-plásticos) de forças e deslocamentos, *Fractus2D* adota uma estratégia para identificação de Φ_e baseada em sucessivas regressões lineares até que o j -ésimo coeficiente angular estimado, $\Phi_{e,j}^*$, satisfaça à condição

$$|\Phi_{e,j}^* - \Phi_{e,j-1}^*| \leq \chi \quad (21)$$

onde χ é o nível de tolerância adotado (10^{-2} na presente versão). Adicionalmente, o coeficiente angular, Φ_e , pode ser determinado por intermédio da flexibilidade elástica para o espécime sob análise conforme descrito na Seção 6.5.3 e Anexo B.

- ii) Para o k -ésimo nível de carga (*load step*), a área total, A_t , é determinada pelo somatório de cada área parcial correspondente ao intervalo $[k-1, k]$ indicado pela área **ABCD** na Fig. 13 na forma

$$A_t = \sum A_k \quad (22)$$

onde A_k é avaliada numericamente por intermédio da regra de Simpson [47] expressa por

$$A_k = \frac{(b-a)}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (23)$$

- iii) Uma vez determinada a área total, A_t^k , para o k -ésimo nível de carga (*load step*), a área plástica correspondente, A_p^k , é calculada subtraindo-se a área elástica, A_e^k , obtida simplesmente a partir da área do triângulo **EBC** na Fig. 13.

3.3.2 Fatores Eta

Para um nível de força (local) motriz sobre a trinca (J ou CTOD) correspondente ao k -ésimo nível de carga (global) remota (*load step*), o fator η pode ser determinado por meio da solução das Eqs. (14) e (15) em termos da Integral J ou, equivalentemente, das Eqs. (19) e (20) em termos de CTOD. Suprimindo os superescritos LLD e CMOD para maior clareza, os fatores η correspondentes ao k -ésimo nível de carregamento sobre o componente ou espécime sob análise são expressos por

$$\eta_J^k = \frac{J_p^k B b}{A_p^k} \quad (24)$$

e

$$\eta_\delta^k = \frac{\delta_p^k B b \sigma_{ys}}{A_p^k} \quad , \quad \eta_\delta^k = \frac{\delta_p^k B b \sigma_f}{A_p^k} \quad (25)$$

As definições acima permitem obter valores *instantâneos* para o fator η (veja Fig. 13) sendo, portanto, relativamente dependentes do nível de carga particularmente para pequenos valores de J (ou, equivalentemente, CTOD). De fato, sob baixos níveis de carga, os valores das áreas elástica e plástica apresentam magnitudes similares o que potencialmente afeta os valores correspondentes dos fatores η_J^k e η_δ^k .

Valores acurados do fator η e, ao mesmo tempo, representativos da contribuição plástica à energia de deformação do sólido (devido à presença da trinca) podem ser determinados por intermédio de uma estratégia simples para obtenção da *média* dos valores instantâneos η_J^k e η_δ^k . A manipulação das Eqs. (14) e (15) em termos de J , e Eqs. (19) e (20) em termos de CTOD (novamente os superescritos LLD e CMOD são suprimidos para maior clareza) resulta em

$$\frac{J_p}{b\sigma_{ys}} = \frac{\eta_J A_p}{Bb^2\sigma_{ys}} \quad (26)$$

e

$$\frac{\delta_p}{b} = \frac{\eta_\delta A_p}{Bb^2\sigma_{ys}} \quad (27)$$

ou

$$\frac{\delta_p}{b} = \frac{\eta_\delta A_p}{Bb^2\sigma_f} \quad (28)$$

nas quais os fatores η_J e η_δ representam a inclinação das curvas (retas) $J_p/(b\sigma_{ys})$ vs. $A_p/(Bb^2\sigma_{ys})$ e δ_p/b vs. $A_p/(Bb^2\sigma_{ys})$ ou $A_p/(Bb^2\sigma_f)$ como ilustrado na Fig. 14.

O procedimento implementado em *Fractus2D* utiliza uma regressão linear sobre todos os valores numéricos da Integral J (CTOD) normalizada vs. área plástica normalizada desde o nível de carga inicial, j_0 , até o nível de carga corrente, k (veja Fig. 14). A estratégia numérica define o nível de carga inicial, j_0 , como aquele no qual o critério $A_p \leq \beta(A_e + A_p) = \beta A_t$, onde o valor $\beta = 0.1$ é normalmente adotado.

3.3.3 Determinação do CTOD

O programa de elementos finitos não linear WARP3D [1] sobre o qual *Fractus2D* é construído (assim como outros programas de elementos finitos disponíveis) não calcula o valor do parâmetro elasto-plástico CTOD (WARP3D fornece somente os valores da Integral J). Portanto, *Fractus2D* implementa rotinas específicas para determinação do CTOD utilizando os deslocamentos nodais obtidos a partir da análise de elementos finitos para o componente ou espécime sob consideração.

A determinação numérica do CTOD convencionalmente utilizada em *Fractus2D* é obtida por intermédio do procedimento largamente utilizado baseado no método de intersecção 90° [4] (também conhecido como método da intersecção 45°) ilustrado na Fig. 15. O procedimento simplesmente define o CTOD (δ) como a distância (vertical) da abertura da boca da trinca correspondente à intersecção de duas retas ortogonais com origem na ponta da trinca sobre o seu flanco deformado.

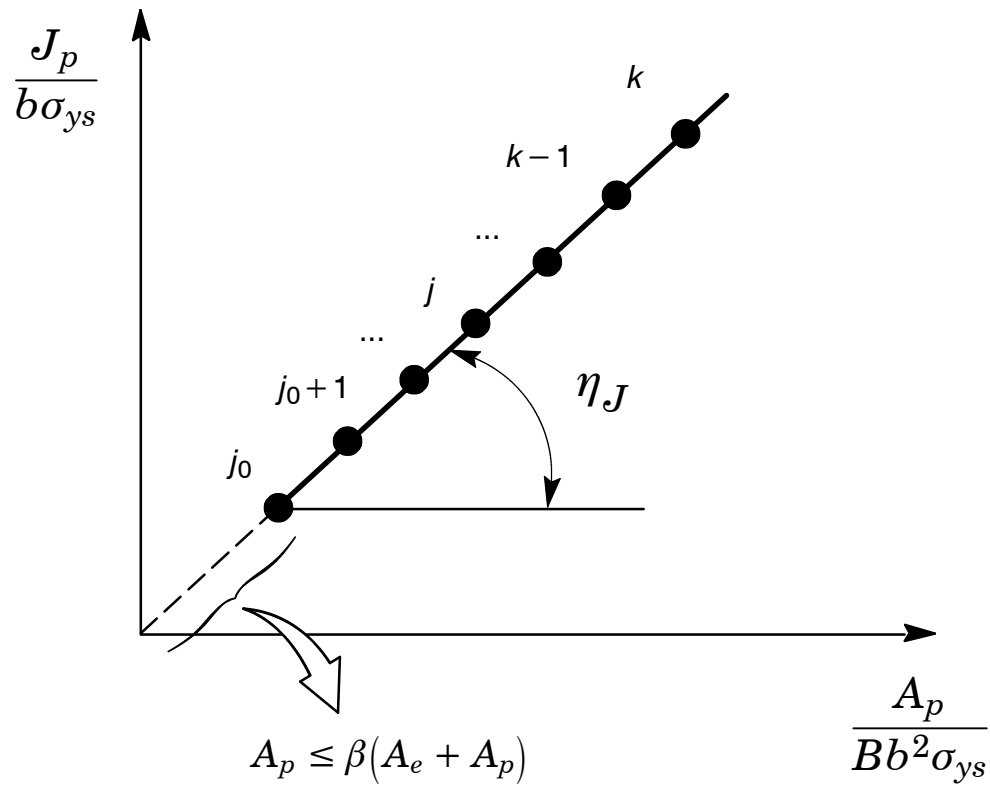


Figura 14 Esquema utilizado para determinação numérica por intermédio de uma reta de ajuste (mínimos quadrados) sobre os valores normalizados de J_p e A_p .

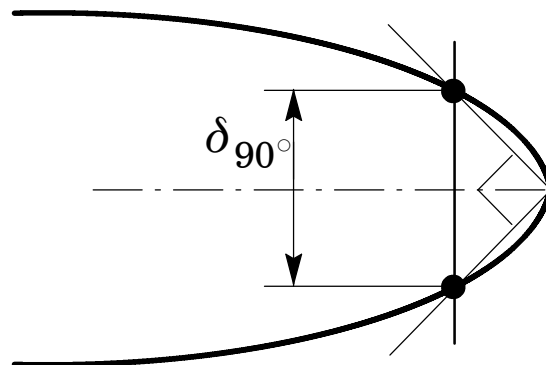


Figura 15 Determinação do CTOD por intermédio do método da intersecção 90° [4].

Para minimizar potenciais problemas numéricos devido à dificuldade de definição do “correto” ponto de intersecção sobre a região intensamente deformada do *blunting* da trinca, *Frac-*

tus2D utiliza um procedimento de regressão linear (método dos mínimos quadrados) sobre as coordenadas nodais *deformadas* para o flanco da trinca. Tal procedimento permite construir uma reta *média* representativa do perfil deformado da trinca e, conseqüentemente, definir adequadamente o ponto de intersecção. A Seção 6.4.1 descreve detalhes adicionais do procedimento.

Outra alternativa para determinação numérica do CTOD implementada em *Fractus2D* é utilização do procedimento derivado do método da tangente [4] o qual forma a base para a determinação experimental do CTOD especificado na norma BS 7448 [48]. O método da tangente define o CTOD (δ) como a distância (vertical) da abertura da boca da trinca correspondente à intersecção de uma reta tangente aos flancos da trinca com o eixo vertical com origem na ponta da trinca como ilustrado na Fig. 16. A Seção 6.4.1 também descreve detalhes adicionais deste procedimento.

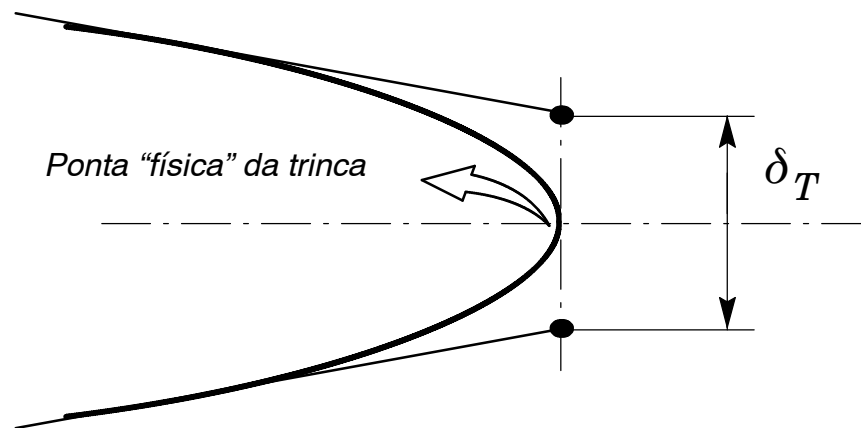
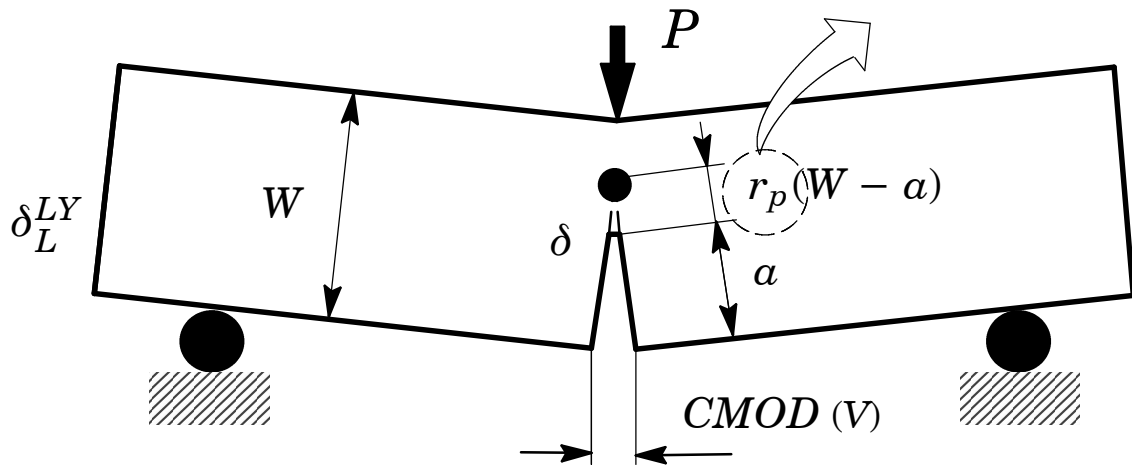


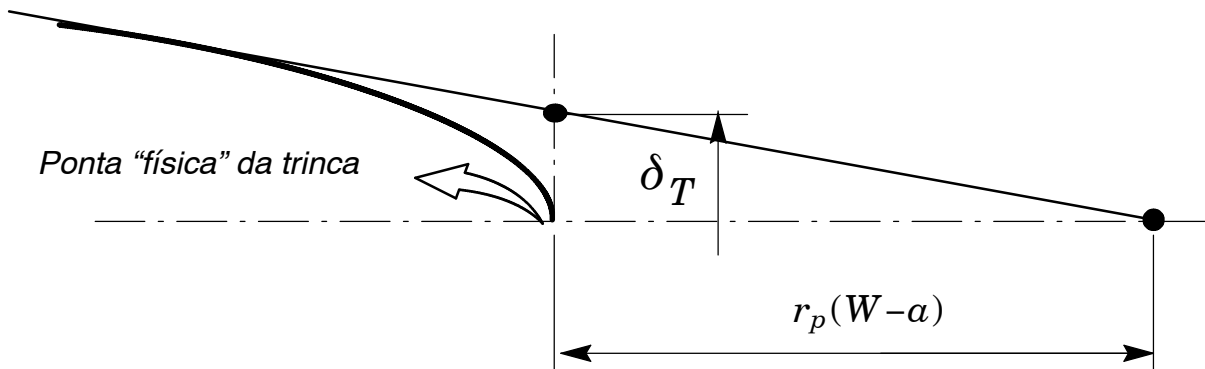
Figura 16 Determinação do CTOD por intermédio do método tangente [4].

3.3.4 Determinação do Fator Rotacional Plástico

Alguns procedimentos para medição experimental do CTOD utilizando curvas de carga *vs.* CMOD, tais como a norma BS 7448 [48], adotam o modelo da rótula plástica [4] para determinação da sua componente plástica, δ_p . Tal método deriva da consideração de que o corpo-de-prova rotacional em torno de um ponto fixo localizado no ligamento remanescente da trinca, denominado rótula plástica, como ilustrado na Fig. 17(a).



(a)



(b)

Figura 17 Modelo da rótula plástica e definição do fator rotacional plástico, r_p [4].

Assumindo que os flancos da trinca permanecem retos durante a deformação do corpo-de-prova e utilizando uma simples semelhança de triângulos, a componente plástica do CTOD, δ_p , é relacionada com a componente plástica do CMOD, V_p , por meio da expressão

$$\delta_p = \frac{r_p(W - a)V_p}{r_p(W - a) + a + z} \quad (29)$$

onde z é a altura do suporte de fixação (*knife edge*) do extensômetro (*clip gage*) utilizado para medição do CMOD ($z=0$ se o *clip gage* for fixado diretamente no espécimes – veja [48]) e o parâmetro r_p representa o fator rotacional plástico o qual define a posição relativa (aparente)

do ponto de rotação (rótula plástica) sobre o ligamento remanescente da trinca como indicado na Fig. 17(b). Para corpos-de-prova SE(B) com trincas profundas, r_p assume o valor de 0.4 na norma BS 7448 [48]. Cravero and Ruggieri [43] apresentam também valores do parâmetro r_p para corpos-de-prova SE(T) fixados por pinos necessários à correção da flexibilidade em procedimentos para determinação de curvas J - R utilizando estes espécimes.

A presente versão de Fractus2D somente permite determinar o fator r_p para os espécimes analisados pelo programa quando o modelo tangente for especificado para o cálculo do CTOD.

3.4 Fatores η para Corpos-de-Prova Convencionais SE(B)

Para ilustrar o procedimento, a Fig. 18 apresenta os fatores η_J^{CMOD} para um espécime SE(B) com diversas relações a/W e diferentes materiais elasto-plásticos descritos por um modelo de potência (Ramberg-Osgood) da forma $(\epsilon/\epsilon_0) = (\sigma/\sigma_0)^n$ com $n = 5, 10$ e 20 [49]. O fator η_J^{CMOD} é definido através da Eq. (15) adotando o critério $A_p \leq \beta(A_e + A_p) = \beta A_b$, onde $\beta = 0.1$. Os fatores η_J^{CMOD} obtidos praticamente independem das propriedades de encruamento do material (coeficiente de encruamento n) e apresentam uma variação essencialmente linear com a relação a/W .

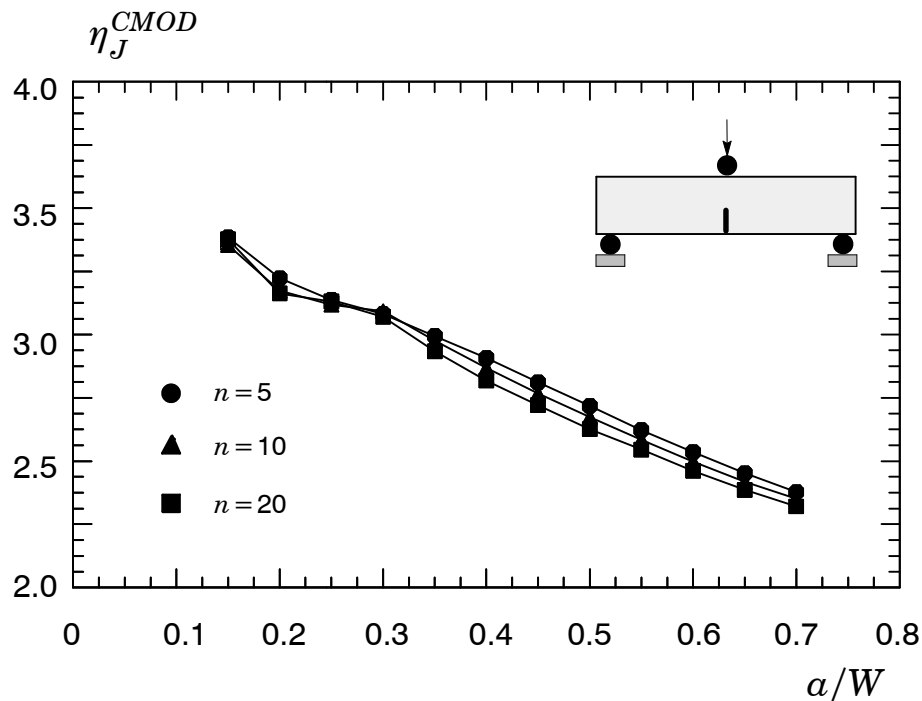


Figura 18 Variação do fator η_J baseado no CMOD para um corpo-de-prova SE(B) com diversas profundidades de trinca e diferentes propriedades de materiais. [49].

Sintaxes de Comando

4

4.1 Utilização de Fractus2D

Fractus2D utiliza comandos em formato livre fornecidos através de um arquivo para definir os modelos (espécime de mecânica da fratura ou componente estrutural e a placa infinita de referência quando necessária) e parâmetros de análise. O arquivo de comandos permite o amplo uso de comentários podendo, portanto, prover a análise com documentação própria. Os resultados de elementos finitos consistem de arquivos armazenados em formato binário ou ASCII compatíveis com o pós-processador PATRAN e os resultados de tensões e deformações, incluindo deslocamentos e forças nodais, obtidos a partir de análises de elementos finitos por meio do programa WARP3D [1].

Fractus2D pode ser executado em estações de trabalho UNIX e sistemas operacionais LINUX e DOS/Windows através do comando *shell*:

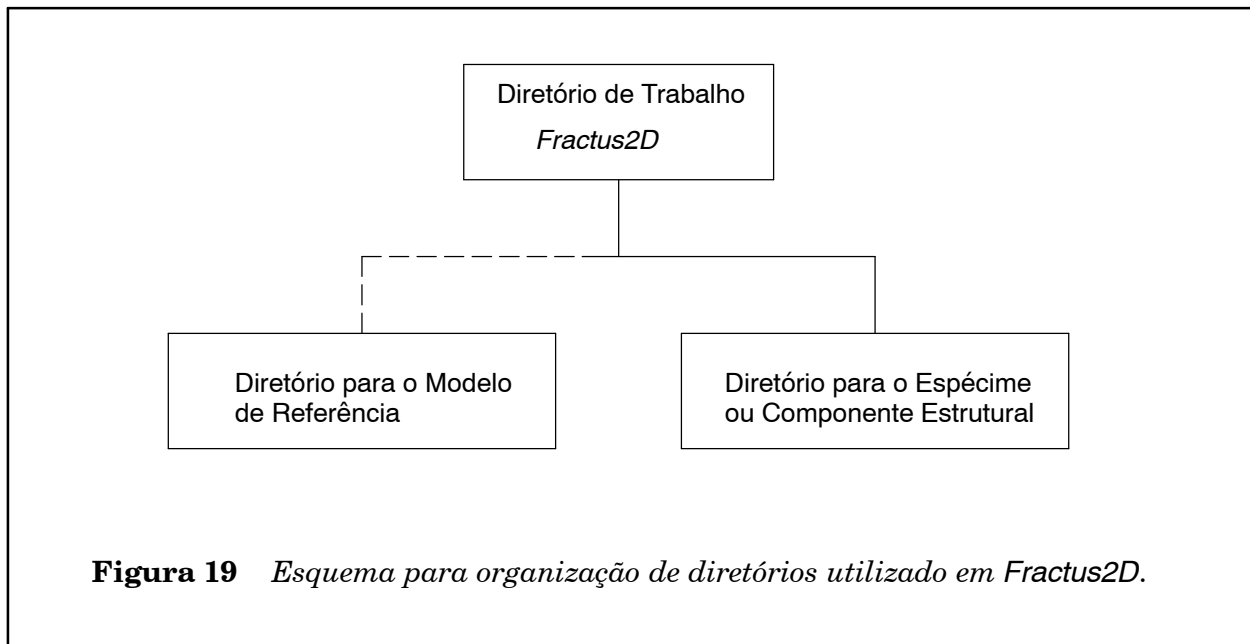
```
cracktool < arquivo_entrada > arquivo_saída
```

Em sistemas UNIX e LINUX, a execução do programa pode ser colocada em segundo plano (*background*) utilizando-se o comando

```
cracktool < arquivo_entrada > arquivo_saída &
```

Convencionalmente, o diretório de trabalho (a partir de onde *Fractus2D* deve ser executado) está um nível acima do diretório contendo os arquivos descritivos do modelo de elementos finitos e resultados. Adicionalmente, devido à necessidade de manipulação de dois modelos diferentes (placa infinita de referência e o componente estrutural) para a determinação das trajetórias *J-Q*, os arquivos utilizados (malha de elementos finitos, parâmetro de carga e

resultados de elementos finitos) para este tipo de análise estão localizados em diretórios distintos em um nível imediatamente abaixo do diretório de trabalho como ilustrado na Fig 19 a seguir.



4.2 Notação Utilizada na Sintaxe de Comandos

O processador de comandos (*input translator*) implementado em *Fractus2D* possibilita a utilização de comandos em linguagem estruturada orientada ao problema (*problem oriented language* – POL) de forma a simplificar a especificação dos modelos e parâmetros de análise. As notações utilizadas neste manual para a descrição dos comandos são apresentadas a seguir.

Dentro de um comando processado pelo programa *Fractus2D*, um descritor na forma

< inteiro >

implica que o usuário deve informar ou fornecer um dado naquela posição da linha de comando na classe representada pelo descritor (um número inteiro no exemplo acima). Portanto, a linha de comando

```
crack tip node < integer >
```

implica que a palavra *node* deve ser seguida por um número inteiro, tal como 10 ou 356, resultando em um comando fornecido pelo usuário na forma

```
crack tip node 356
```

Os descritores apresentados abaixo são comumente utilizados para a entrada de dados em *Fractus2D*. Outros descritores eventualmente não incluídos nas definições abaixo são explicados ao longo do texto.

- < inteiro > é uma série de dígitos opcionalmente precedidos por um sinal positivo ou negativo. Exemplos são 121, +300, -410.
- < real > é uma série de dígitos com um ponto decimal incluído ou uma série de dígitos seguidos por um símbolo exponencial (“e”) indicando uma potência de 10. Números reais também podem ser opcionalmente precedidos por um sinal positivo ou negativo. Exemplos são 1.0, -2.5, 4.3e-01.
- < número > representa tanto um número < real > como um número < inteiro >. O interpretador de comandos implementado em *Fractus2D* executa a conversão necessária para armazenamento interno.
- < label > é uma série de letras e dígitos. A seqüência deve necessariamente iniciar com uma letra. O interpretador de comandos também aceita o caractere “_” como uma letra válida. Um label pode ter a forma amostra_ct, por exemplo, para possibilitar maior clareza e documentação à análise.
- < lista > é a notação usada para indicar uma seqüência de valores inteiros positivos – usualmente seqüência de nós e elementos. Uma lista geralmente contém duas formas de dados que podem ser utilizadas simultaneamente (dentro da mesma lista). A primeira forma de dados é uma série de números inteiros opcionalmente separados por vírgula. Um exemplo é a lista 1, 3, 6, 10, 12. A segunda forma comum de uma lista representa uma seqüência consecutiva de inteiros e consiste de dois números inteiros separados por um hífen. Um exemplo é a representação 1-10, a qual implica todos os números inteiros da seqüência 1 até 10. Uma extensão desta forma de lista implica um incremento constante como, por exemplo, 1-10 by 2, a qual representa a seqüência 1, 3, 5, 7, 9. Uma terceira forma é utilizar a palavra *all* (quando permitida pelo comando) a qual implica que todos os números inteiros (aplicáveis à lista) devem ser associados à lista. As formas de lista previamente descritas podem ser combinadas em um mesmo comando como, por exemplo, ... *nodes 1-100 by 3, 200-300, 500-300 by -3*.

Os comandos utilizados pelo programa *Fractus2D* são definidos em língua inglesa para permitir a sua utilização e portabilidade em diferentes países. Para a definição adequada de cada comando, palavras (ou parte de palavras) sublinhadas são requeridas para a operação correta do interpretador de comandos. Os descritores, tais como <inteiro>, não são sublinhados mas devem sempre ser substituídos pelo dado correspondente à classe especificada pelo descritor. Por exemplo, a linha de comando definida por

nondimensional radius < real >

pode ser resumidamente definida por

nondim rad 2.0

caso o usuário assim preferir.

Em alguns comandos, mais de uma palavra é aceita pelo interpretador com o objetivo de especificar outras opções dentro do comando. Para estes casos, as opções são agrupadas verticalmente dentro de chaves como no exemplo a seguir

```
compute j-q curve { on
                    off }
```

o que indica que ambos os comandos são aceitos pelo interpretador

```
compute j-q curve on
compute j-q curve off
```

Palavras opcionais são inseridas em parêntesis, (), como no comando a seguir

format (of) fe-results patran type binary

De forma a tornar a definição do comando mais descritiva, os dados indicados por <> podem ser adicionalmente definidos pelo seu significado físico seguidos pelo tipo ou classe a que pertencem. Por exemplo, o comando

structure < name of structure: label >

implica que a informação que deve ser fornecida após a palavra *structure* é o nome ou identificação da estrutura a qual deve ser um descritor do tipo < label >. Exemplos de comandos são

```
structure duto_ext
struct senb
```

enquanto o comando

```
structure 1a
```

não é aceitável uma vez que o nome da estrutura não é do tipo label (*labels devem iniciar por uma letra*).

Continuação da Linha

Uma vírgula (,) colocada ao final de uma linha faz com que a linha subsequente seja considerada uma continuação lógica da linha corrente. Não há limitações para o número de continuações de linhas. A continuação, definida pela vírgula, pode ser posicionada em qualquer ponto dentro de qualquer comando.

Comentários

Comentários podem ser inseridos no arquivo de comandos de maneira similar à linguagem Fortran. A letra 'c' ou 'C' colocada na primeira coluna (física) da linha a torna uma linha de comentários. A linha é lida pelo interpretador de comandos e (possivelmente) repetida no terminal ou arquivo de saída. O seu conteúdo é ignorado pelo interpretador o qual avança para a leitura da próxima linha.

Final de Linha

A indicação de final de linha é feita através de três formas. Na primeira forma, o interpretador de comandos examina as colunas em cada linha somente até a coluna 72. Na segunda forma, o interpretador de comandos conta o número de caracteres brancos sucessivos após encontrar o primeiro dado de entrada. Caso o número de caracteres brancos seja igual a 40, o restante da linha é ignorado e assumido como branco. Finalmente, o término da linha pode ser indicado pelos caracteres "\$" ou "!". Um espaço após os caracteres "\$" ou "!" é ignorado pelo interpretador de comandos e pode ser usado para introdução de pequenos comentários.

4.3 Definição da Análise

Fractus2D foi concebido de forma estruturada para acomodar a possibilidade de determinação e cálculo de diversos parâmetros associados a problemas de mecânica da fratura. O primeiro comando do arquivo de entrada define o tipo de análise (o qual, por sua vez, define os blocos de subcomandos descritos a seguir) na forma

```
c
  crack analysis type < tipo de análise: label >
c
```

A versão corrente de *Fractus2D* implemente dois tipos de análises: *i*) Determinação das trajetórias *J-Q* e *ii*) Determinação dos fatores η para espécimes convencionais de mecânica da fratura. A definição destes tipos de análise será descrita nas seções correspondentes a seguir.

Versões futuras de *Fractus2D* implementarão outras capacidades e funcionalidades tais como a determinação dos fatores de intensidade de tensões (K_I) para geometrias diversas baseados em fatores de influência especificados no procedimento API 579 [NO TAG] e o procedimento EPRI [NO TAG,NO TAG] para determinação de J e CTOD utilizando fatores h_1 e h_2 .

4.4 Blocos de Subcomandos

A definição da análise descrita anteriormente é seguida pela especificação de blocos de comandos (dependentes do tipo de análise sendo conduzida) e pela palavra chave end como mostrado no exemplo a seguir. Cada bloco de comandos define um bloco de subcomandos utilizados para descrever os modelos e parâmetros de análise. A leitura de parâmetros e informações de entrada cessa quando o comando end é encontrado pelo interpretador de comandos. Cada bloco de subcomandos é delimitado por chaves “{” e “}” como apresentado a seguir:

```

c
  ssy model {
    <lista de sub-commands> }
c
  finite body {
    <lista de sub-commands> }
c
  analysis parameters {
    <lista de sub-commands> }
c
end

```

Determinação do Parâmetro Q

5

5.1 Definição da Análise

A execução de análise das trajetórias J - Q em *Fractus2D* é especificada pelo comando

```
c
  crack analysis type jq-curves
c
```

A linha de comando acima deve necessariamente ser o primeiro comando interpretado pelo processador de *Fractus2D*.

5.2 Blocos de Subcomandos

A execução da análise para determinação do parâmetro J - Q necessita da especificação dos seguintes blocos de comandos

```
c
  ssy model {
    <lista de sub-commandoss> }
c
  finite body {
    <lista de sub-commands> }
c
  analysis parameters {
    <lista de sub-commands> }
c
  end
```

para os quais os blocos de subcomandos utilizados para descrever os modelos e parâmetros de análise são apresentados a seguir.

5.3 Placa Infinita de Referência

A seção `ssy model` define o modelo de elementos finitos e informações específicas para a placa infinita de referência (modelo MBL). Um bloco típico de subcomandos possui a forma

```

ssy model {
  get files from directory ssy_x60
  input mesh from file ssy_x60_coor
  format fe-results patran type ascii
  crack tip node 1
  crack plane angle nx 1 ny 0
  near tip elements 145 128 111 94 77 13 1 27 43 60 }

```

5.3.1 Coordenadas Nodais e Incidência de Elementos

A incidência de nós dos elementos (conectividades) e as coordenadas nodais do modelo de elementos finitos (configuração indeformada) são necessárias para determinar a distribuição de tensões na região da trinca em função da distância (e, implicitamente, determinar o raio adimensional λ). Adicionalmente, o número de nós e número de elementos do modelo também devem ser especificados.

Inicialmente, o diretório que contém os arquivos de definição do modelo[†] é especificado através do comando

```

get files (from) directory < nome do diretório: label >

```

O comando seguinte especifica o arquivo contendo as informações do modelo de elementos finitos

```

input mesh (from) file < nome do arquivo: label >

```

O arquivo de informações do modelo possui formato livre cujas linhas de dados são apresentadas na Fig 20 onde as linhas de dados são da forma:

Linha 1

Número de nós < inteiro > Numero de elementos < inteiro >

Linhas de Coordenadas Nodais

Número do nó < inteiro >

coordenada x_1 < número >

[†]) O diretório de arquivos do modelo situa-se geralmente um nível abaixo do diretório de trabalho – ver Seção 4.1.

1260 575		} Informações da Malha							
1	-.145000000E+02	.700000000E+02	.250000000E+00						
2	-.145000000E+02	.700000000E+02	.000000000E+00						
3	-.108750000E+02	.700000000E+02	.250000000E+00						
:	:	:	:						} Coordenadas Nodais
:	:	:	:						
1257	-.440890379E-01	.829215795E-01	.250000000E+00						
1258	-.292893108E-01	.707106888E-01	.250000000E+00						
1259	-.292893108E-01	.707106888E-01	.000000000E+00						
1260	-.440890379E-01	.829215795E-01	.000000000E+00						
1	173	174	146	145	175	176	148	147	
2	175	176	148	147	177	178	150	149	
3	177	178	150	149	179	180	152	151	
:	:	:	:	:	:	:	:	:	} Incidência de Elementos
:	:	:	:	:	:	:	:	:	
573	929	930	928	927	897	898	896	895	
574	963	964	962	961	931	932	930	929	
575	931	932	930	929	899	900	898	897	

Figura 20 Arquivo de definição do modelo de elementos finitos utilizado pelo programa Fractus2D.

coordenada x_2 < número >

coordenada x_3 < número >

Linhas de Elementos (elemento sólido de 8 nós)

Número do elemento < inteiro >

Nó n_1 < inteiro >

.....

Nó n_8 < inteiro >

Em geral, é mais simples editar o próprio arquivo de entrada utilizado pelo programa de elementos finitos WARP3D [1] para gerar o arquivo de definição do modelo apresentado na Fig. 20.

5.3.2 Formato dos Resultados de Elementos Finitos

Este comando define o formato dos resultados de elementos finitos que serão utilizados pelo programa *Fractus2D* para a construção das trajetórias *J-Q*. A implementação-padrão adotada (veja Seção 3) emprega as tensões de Cauchy nas coordenadas paramétricas $\eta_1 = \eta_2 = \eta_3 = 0$ (*i.e.*, no centro do elemento); desta forma, os resultados numéricos consistem essencialmente das tensões nos elementos especificados para a região da trinca (conforme notado anteriormente, o raio adimensional λ é calculado sobre a configuração não deformada). Adicionalmente, *Fractus2D* também permite a manipulação de tensões nodais utilizando arquivos de resultados em formato binário ou ASCII compatível com o programa CAE MSc Patran [2]. A sintaxe do comando é da forma

```
format (of) fe-results patran type {  asci  }
                                   {  binary }
```

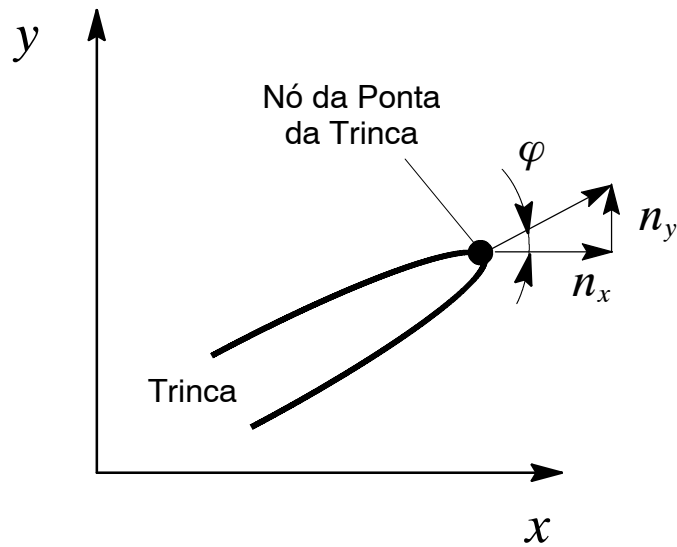
Fractus2D utiliza a mesma convenção de arquivos de resultados Patran adotada pelo programa de elementos finitos WARP3D [1]. Os arquivos binários Patran para as tensões dos elementos são denominados “websxxxxx”, onde “xxxxx” representa o número do incremento de carga (*load step*). Similarmente, os arquivos ASCII Patran para as tensões dos elementos são denominados “wefsxxxxx”. Recomenda-se aos usuários de *Fractus2D* uma leitura prévia do manual do usuário de WARP3D [1] para familiarizar-se com a forma de resultados adotada por aquele programa. A especificação do uso de resultados nodais ou no centro do elemento é feita pelo usuário na Seção *analysis parameters* descrita adiante.

5.3.3 Definição da Trinca

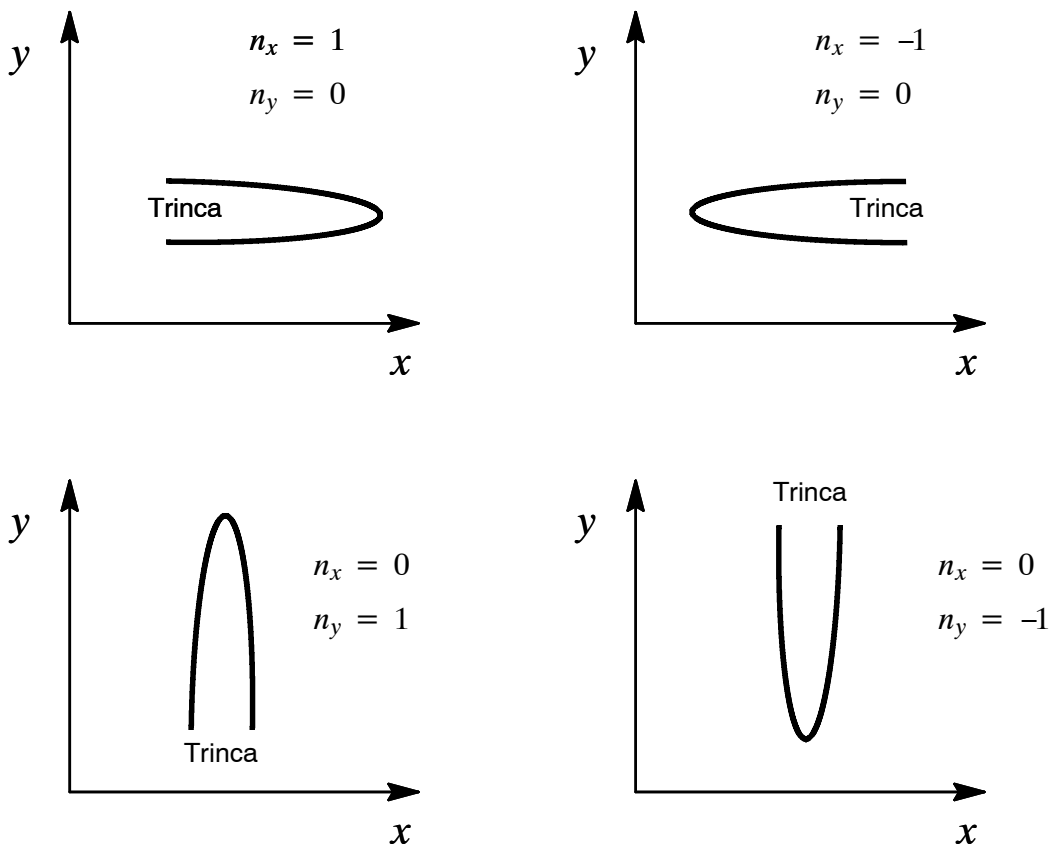
A definição da trinca no modelo requer a especificação do nó de sua ponta (nó localizado no plano de simetria) e de seu ângulo conforme ilustrado genericamente na Fig. 21(a). Os comandos são descritos a seguir

```
crack tip node < inteiro >
```

e



(a) Representação genérica de trinca com inclinação arbitrária.



(b) Definição de orientações possíveis do plano da trinca (implementação corrente).

Figura 21 Definição da trinca utilizada por Fractus2D através da especificação do nó da ponta da trinca e dos cossenos diretores n_x e n_y relativos ao ângulo da trinca (plano x - y).

`crack (plane) angle nx < número > ny < número >`

O ângulo da trinca (denotado φ) é definido pelos cossenos diretores n_x e n_y (ou simplesmente pelas componentes do vetor unitário orientado segundo o ângulo φ) conforme explicitado na Fig. 21(a). Entretanto, a versão corrente de *Fractus2D* não permite a construção de trajetórias J - Q para o caso de trincas inclinadas. O algoritmo de cálculo do parâmetro Q permite somente o tratamento de trincas coplanares ao eixo horizontal (x) ou vertical como indicado na Fig. 21(b). Em versões futuras do programa, o tratamento de trincas inclinadas será implementado.

5.3.4 Elementos e Nós da Região da Trinca

Os elementos da região da trinca devem ser especificados pelo usuário para o processamento da distribuição de tensões em função da distância à ponta da trinca; como notado anteriormente, a distribuição de tensões é utilizada para a construção da trajetória J - Q . Estes elementos localizam-se sobre o ligamento remanescente da trinca no seu plano de simetria ($\theta = 0$). Não é necessário especificar os elementos ordenados (i.e., do mais próximo da ponta da trinca ao mais distante); o programa automaticamente executa o ordenamento dos elementos. Entretanto, o usuário deve assegurar que um número suficiente de elementos seja especificado de forma a permitir o cálculo do raio adimensional λ durante toda a história de carregamento (i.e., até o máximo valor de J desejado); notar que embora o raio adimensional seja fixo nas análises, o raio efetivo $r = \lambda J / \sigma_0$ aumenta com valores crescentes de J . O comando para especificação dos elementos da região da trinca é da forma

`near tip elements < lista de elementos: inteiros >`

Alternativamente, *Fractus2D* permite a construção automática da lista de elementos na região da trinca dentro de um raio especificado pelo usuário. O comando para especificação automática dos elementos da região da trinca utiliza a sintaxe

`near tip elements automatic maximum radius < número >`

onde o subcomando `maximum radius` especifica o raio máximo (medido a partir do nó da ponta da trinca) dentro do qual a lista (automática) de elementos será construída.

No caso da especificação automática da lista de elementos, o usuário deverá atentar para o valor do raio máximo especificado uma vez que valores inadequados deste parâmetro podem

comprometer a determinação do parâmetro Q . De fato, como descrito nas Seções 2 e 3 anteriores, o parâmetro Q é a diferença entre as tensões correspondentes à solução referência e ao corpo finito calculadas para um mesmo valor do *raio adimensional*, $\lambda = r/(J/\sigma_0)$. Conseqüentemente, valores elevados de J implicam em valores elevados do *raio absoluto*, r , o qual pode potencialmente situar-se fora da faixa especificada pelo usuário. Para evitar tais potenciais dificuldades, *Fractus2D* possibilita a especificação de um raio (absoluto) adaptativo por intermédio do comando

```
near tip elements automatic maximum radius adaptive
```

onde o subcomando *adaptive* permite *Fractus2D* determinar o raio máximo adequado ao parâmetro $\lambda = r/(J/\sigma_0)$ especificado em conjunção com um fator de adaptação especificado pelo usuário (ver Seção 5.5.3).

Uma vez que *Fractus2D* também permite a utilização de resultados nodais (tensões e deformações nodais), os nós da região da trinca devem ser especificados pelo usuário para o processamento da distribuição de tensões em função da distância à ponta da trinca. Analogamente à descrição anterior, estes nós localizam-se sobre o ligamento remanescente da trinca no seu plano de simetria ($\theta = 0$). Não é necessário especificar os nós ordenados (i.e., do mais próximo da ponta da trinca ao mais distante); o programa automaticamente executa o ordenamento dos nós. Aqui também o usuário deve assegurar que um número suficiente de nós seja especificado de forma a permitir o cálculo do raio adimensional λ durante toda a história de carregamento (i.e., até o máximo valor de J desejado); notar que embora o raio adimensional seja fixo nas análises, o raio efetivo $r = \lambda J/\sigma_0$ aumenta com valores crescentes de J .

Os comandos para especificação dos nós da região da trinca seguem sintaxe similar à especificação de elementos e são da forma

```
near tip nodes < lista de nodes: inteiros >
```

quando o usuário desejar especificar uma lista definida de nós ou

```
near tip nodes automatic maximum radius adaptive
```

para construção automática da lista de nós.

5.4 Componente Estrutural

A seção `finite body` define o modelo de elementos finitos e informações específicas para o componente estrutural sob análise. Um bloco típico de subcomandos para análises de trinca *estacionária* possui a forma

```
finite body {
    structure duto
    get files from directory duto_x60
    input mesh from file duto_x60_coor
    input j-values from file duto_x60_jval
    format fe-results patran type ascii
    crack tip node 25
    crack plane angle nx 1 ny 0
    near tip elements 485 484 483 473 461 447 431 413 394 }
```

Para o caso de avaliação de trajetória J - Q utilizando análises de *propagação* de trinca, o bloco típico de subcomandos apresenta a seguinte estrutura

```
finite body {
    structure ct_specimen
    get files from directory ctaw6_gt
    input mesh from file ctaw6_coor
    input j-values from file ctaw6_jval
    input r-curve from file ctaw6_jrcurve
    format fe-results patran type ascii
    crack tip node 25
    crack plane angle nx 1 ny 0
    cell size 0.1
    number of cell elements 80
    near tip elements automatic maximum radius adaptive }
```

Diversos comandos possuem sintaxe idêntica aos comandos apresentados anteriormente para a seção `ssy model` e serão referidos à Seção 5.3. Comandos específicos deste bloco serão detalhados a seguir.

5.4.1 Identificação do Componente Estrutural

Este comando especifica a identificação ou nome do componente estrutural sob análise. Caso este comando for omitido, o programa adotará uma identificação padrão como *unnamed*. A sintaxe é simplesmente

```
structure < nome da estrutura: label >
```

5.4.2 Coordenadas Nodais e Incidência de Elementos

A especificação da incidência de nós dos elementos (conectividades) e as coordenadas nodais do modelo de elementos finitos (configuração indeformada) utiliza os mesmos comandos apresentados anteriormente na Seção 5.3.1.

5.4.3 Formato dos Resultados de Elementos Finitos

A especificação do formato dos resultados de elementos finitos que serão utilizados pelo programa *Fractus2D* para a construção das trajetórias *J-Q* utiliza os mesmos comandos apresentados anteriormente na Seção 5.3.2.

5.4.4 Definição da Trinca

A definição da trinca para o modelo do componente estrutural utiliza os mesmos comandos apresentados anteriormente na Seção 5.3.3.

5.4.5 Elementos e Nós da Região da Trinca

A especificação dos elementos e nós da região da trinca para o processamento da distribuição de tensões em função da distância à ponta da trinca utiliza os mesmos comandos apresentados anteriormente na Seção 5.3.4.

5.4.6 Parâmetro de Carga: Integral *J*

Os valores calculados da Integral *J* para passos de carga (*load steps*) especificados devem ser fornecidos ao programa por intermédio de um arquivo de dados, o qual situa-se no mesmo diretório contendo os arquivos de definição do modelo (ver Seção 5.3.1). *Estes valores devem ser consistentes com os resultados de elementos finitos gerados pela análise de tensões do componente estrutural.* O comando para especificação do arquivo é da forma

```
input j-values (from) file < nome do arquivo: label >
```

O arquivo de valores de Integral *J* possui formato livre cujas linhas de dados são apresentadas na Fig 22 a seguir:

10	0.555
20	1.145
30	2.008
.	.
.	.
.	.

Figura 22 Arquivo de valores calculados de Integral J para os passos de carga (load steps) especificados utilizado pelo programa Fractus2D.

onde as linhas de dados são da forma:

Linha 1 ao número de passos de carga

Valor do passo de carga < inteiro > Valor da Integral J < number >

5.4.7 Curva de Resistência J - R

Os valores de Integral J versus crescimento de trinca, Δa , (também denominada curva de resistência J - R) correspondentes ao componente sob análise devem ser fornecidos ao programa por intermédio de um arquivo de dados, o qual situa-se no mesmo diretório contendo os arquivos de definição do modelo (ver Seção 5.3.1). *Estes valores devem ser consistentes com os resultados de elementos finitos gerados pela análise de crescimento de trincas do componente estrutural.* O comando para especificação do arquivo é da forma

```
input r-curve (from) file < nome do arquivo: label >
```

O arquivo de valores de J versus Δa possui formato livre cujas linhas de dados são apresentadas na Fig 23 a seguir:

1	103	2	87.38	0.1
2	250	3	314.4	0.2
3	259	4	319.1	0.3
4	269	5	342.90	0.4
.
.
.

Figura 23 Arquivo de valores calculados de Integral J para os passos de carga (load steps) especificados utilizado pelo programa Fractus2D.

onde as linhas de dados são da forma:

Linha 1 ao número de passos

Número do passo < inteiro > Valor do passo de carga < inteiro > ,

Número do elemento da ponta da trinca < inteiro > ,

Valor da Integral J < number > Valor do crescimento de trinca Δa < number >

5.4.8 Células Computacionais

As análises de crescimento de trinca para determinação da curva de resistência J - R para o componente sob análise utilizam o conceito de células computacionais baseado em dois critérios de propagação (critério de *Gurson-Tvergaard* e critério do *crack tip opening angle* (CTOA) - ver Seção 5.5.# adiante). De forma a manipular corretamente o arquivo de valores J versus Δa fornecido ao programa, *Fractus2D* necessita da definição do tamanho da célula computacional e do número de células computacionais dispostas ao longo do ligamento remanescente da trinca. Estes valores são especificados pelos comandos

`cell size < number >`

`number (of) cell elements < integer >`

Não há valor padrão adotado pelo programa para o tamanho da célula e número de células.

5.5 Parâmetros de Análise

A seção `analysis parameter` define os parâmetros de análise e de controle utilizados para a execução do programa *Fractus2D*. Um bloco típico de subcomandos possui a forma

```
analysis parameters {
  nondimensional radius 2.0
  yield stress 400
  reference j-integral 44.5 at load step 100
  compute j-q curves on steps 100-500 by 50
  plot stress on steps 100 200 300 400 500 }
```

5.5.1 Raio Adimensional

Como explicitado na Eq. (7) apresentada na Seção 2, o parâmetro Q é determinado a partir da diferença de tensões medidas sobre valores fixos do raio adimensional $\lambda = r/(J/\sigma_0)$. O valor de λ utilizado nas análises é especificado por intermédio do comando

```
nondimensional radius < número >
```

Quando o valor do raio adimensional não for especificado pelo usuário, o valor padrão $\lambda = 2$ é assumido pelo programa.

5.5.2 Tolerância de Busca Nodal

A construção automática da lista de elementos na região da trinca utiliza uma rotina de “busca” nodal onde as coordenadas nodais dos elementos incluídos na lista de elementos são comparadas com as coordenadas do nó da ponta da trinca dentro de uma tolerância adequada. O valor da tolerância de equivalência nodal utilizado nas análises é especificado através do comando

```
node tolerance < número >
```

Quando o valor da tolerância de equivalência nodal não for especificado pelo usuário, o valor padrão $\xi = 1e^{-4}$ é assumido pelo programa.

5.5.3 Fator de Adaptação

A especificação de um raio (absoluto) adaptativo descrito anteriormente permite *Fractus2D* determinar um raio máximo consistente com o parâmetro $\lambda = r/(J/\sigma_0)$. Entretanto, devido ao

refinamento e zonas de transição da malha de elementos finitos em regiões suficientemente distantes da ponta da trinca, é conveniente especificar uma margem adicional sobre o raio (absoluto) máximo determinado a partir do parâmetro $\lambda = r/(J/\sigma_0)$. Tal margem adicional é especificada pelo comando

```
adaptive radius factor < número >
```

onde o descritor < número > representa efetivamente a margem adicional sobre o valor do raio (absoluto) máximo inicialmente determinado por *Fractus2D* (por exemplo, um fator de adaptação igual a 1.30 implica em um raio máximo efetivo 30% maior que o raio máximo inicialmente calculado). Quando o valor do fator de adaptação não for especificado pelo usuário, um fator padrão igual a 1.25 é assumido pelo programa.

5.5.4 Tensão de escoamento do Material

O valor da tensão de escoamento (ou referência) do material é necessário para o cálculo do raio adimensional e do parâmetro Q (ver Eq. (7) apresentada na Seção 3). O comando é simplesmente

```
yield stress < número >
```

5.5.5 Integral J de Referência

O valor de referência da Integral J , associado à análise de tensões da placa infinita de referência (modelo MBL), é necessário para o cálculo do raio efetivo (com o valor fixo de λ) onde as tensões de referência serão obtidas. O usuário deve fornecer o valor calculado de Integral J e o passo de carga (*load step*) correspondente para posterior processamento de tensões pelo programa. O comando possui a forma

```
reference j-integral < número > (at) load step < inteiro >
```

5.5.6 Construção de Trajetórias J-Q

O controle de execução das rotinas para a construção de trajetórias $J-Q$ e a especificação dos passos de carga (*load steps*) sobre os quais o parâmetro Q é calculado utilizam o seguinte comando

```
compute j-q curves { on } steps < lista de passos: inteiros>
                   { off }
```

O valor padrão adotado pelo programa para o controle de execução é `on`. Caso os passos de carga não sejam especificados pelo usuário, o programa utiliza todos os passos de carga especificados no arquivo de valores de Integral J (ver Seção 5.5.5).

5.5.7 Distribuição de Tensões e Deformações

O controle de execução das rotinas para a distribuição de tensões e deformações assim como a especificação dos passos de carga (*load steps*) sobre os quais as tensões são obtidas utilizam o seguinte comando

```
plot { stress
      strain
      stress strain } { on } steps < lista de passos: inteiros>
      { off }
```

O valor-padrão adotado pelo programa para o controle de execução é `off`. Caso os passos de carga não sejam especificados pelo usuário, o programa utiliza todos os passos de carga especificados no arquivo de valores de Integral J (ver Seção 5.5.5).

5.5.8 Uso de Resultados Nodais

A utilização de resultados nodais (tensões nodais e, quando necessário, deformações nodais) sobre os quais o parâmetro Q é calculado é especificada pelo usuário por intermédio do seguinte comando

```
nodal results { on }
              { off }
```

onde o valor-padrão adotado pelo programa para o controle de execução é `off`.

5.5.9 Versão do Programa FEA WARP3D

Embora não existam diferenças significativas entre os resultados de deslocamentos e forças nodais (incluindo tensões e deformações) para as diferentes versões do programa de elementos finitos WARP3D [1], há algumas pequenas diferenças de formato dos arquivos “wnfdxxxx”

(“wnbdxxxx”) e “wnfrxxxx” (“wnbrxxxx”) – ver Seção 5.3.2. As versões correntemente suportadas por *Fractus2D* são 14.x, 15.x, 16.x e 17.x. A especificação da versão utilizada do programa WARP3D é feita pelo comando

$$\text{warp3d release } \left\{ \begin{array}{l} \underline{\text{v14}} \\ \underline{\text{v15}} \\ \underline{\text{v16}} \\ \underline{\text{v17}} \end{array} \right\}$$

onde o valor padrão adotado para a versão do programa WARP3D é v15.

5.5.10 Trajetórias *J-Q* para Propagação de Trincas

Conforme descrito anteriormente na Seção \$\$, as trajetórias *J-Q* para análises não estacionárias de trinca (crescimento de trincas em regime dúctil sob carregamento monotônico) utilizam a determinação do parâmetro *Q* para a posição *corrente* da ponta da trinca em propagação com incrementos correspondentes da Integral *J*. A definição de trajetórias *J-Q* para trincas em propagação é especificada em *Fractus2D* por meio do comando

$$\text{crack growth } \left\{ \begin{array}{l} \underline{\text{on}} \\ \underline{\text{off}} \end{array} \right\} \quad \text{fracture criterion } \left\{ \begin{array}{l} \underline{\text{gurson}} \\ \underline{\text{ctoa}} \end{array} \right\}$$

onde as opções para o critério de propagação definidas pelos descritores < gurson > e < ctoa > representam os critérios de *Gurson-Tvergaard* e *crack tip opening angle (CTOA)* para extensão dúctil de trincas (ver discussão detalhada sobre estes critérios assim como os modelos associados de propagação de trincas em Ruggieri e Dodds [35], Gullerud et al. [50,51] e Ruggieri e Dotta [52]). Os valores-padrão definidos em *Fractus2D* correspondem a < crack growth off > (trajetórias *J-Q* para trincas estacionárias) e < fracture criterion gurson > para o caso de construção de trajetórias *J-Q* para trincas não estacionárias.

Determinação dos Fatores *ETA*

6

6.1 Definição da Análise

A execução de análise de fatores η em *Fractus2D* é especificada pelo comando

```
c
  crack analysis type eta-factor
c
```

A linha de comando acima deve necessariamente ser o primeiro comando interpretado pelo processador de *Fractus2D*.

6.2 Blocos de Subcomandos

A execução da análise para determinação dos fatores η necessita da especificação dos seguintes blocos de comandos

```
c
  crack configuration {
    <lista de sub-commandoss> }
c
  mesh based parameters {
    <lista de sub-commands> }
c
  analysis parameters {
    <lista de sub-commands> }
c
  end
```

para os quais os blocos de subcomandos utilizados para descrever os modelos e parâmetros de análise são apresentados a seguir.

6.3 Configuração do Componente ou Espécime

A seção `crack configuration` define a geometria do espécime sob análise e informações associadas ao diretório e arquivos de resultados. Um bloco típico de subcomandos possui a forma

```

crack configuration {
  structure sebaw5
  crack geometry seb
  specimen thickness 1.0
  specimen width 50.8
  crack size 25.4
  get files from directory seb_aw5_n10
  input mesh from file sebaw5_coor
  input j-values from file sebaw5_jvalues
  format fe-results patran type ascii }

```

6.3.1 Identificação do Componente Estrutural ou Corpo-de-Prova

Este comando especifica a identificação ou nome do componente estrutural sob análise. Caso este comando seja omitido, o programa adotará uma identificação padrão denominada *unnamed*. A sintaxe é simplesmente

```
structure < nome da estrutura: label >
```

6.3.2 Configuração (Geometria) do Corpo-de-Prova

A definição da configuração ou geometria do corpo-de-prova permite a execução de subrotinas específicas para determinação dos fatores de intensidade de tensão, curvas de carga *vs.* deslocamento, entre outras. A versão corrente de *Fractus2D* implementa a determinação de fatores η para três configurações básicas (conforme Anderson [4]): *i*) corpo-de-prova compacto C(T) – *compact tension specimen*; *ii*) corpo-de-prova flexão 3 pontos SE(B) – *single edge notched bend specimen*; *iii*) corpo-de-prova submetido a tração (fixado por pinos ou garras) SE(T) – *single edge notched tension specimen*. A sintaxe do comando é da forma

```

crack geometry {
  ct
  seb
  pinned set
  clamped set
}

```

6.3.3 Dimensões (Principais) do Corpo-de-Prova

Alguns parâmetros e variáveis os quais entram diretamente na determinação dos fatores η , tais como o fator de intensidade de tensão e a curva de carga *vs.* deslocamento, necessitam da

especificação de dimensões principais do corpo-de-prova como a sua espessura, largura e tamanho (profundidade) da trinca. Estas dimensões são especificadas por intermédio dos comandos

```
specimen thickness < número >
```

```
specimen width < número >
```

```
crack size < número >
```

Notar que a espessura do corpo-de-prova definida pelo comando *specimen thickness* refere-se à espessura do modelo de elementos finitos não a sua espessura real. Tal observação é importante dentro do procedimento implementado uma vez que *Fractus2D* determina a carga aplicada diretamente a partir do somatório das reações nodais do modelo de elementos finitos. Somente o fator de simetria planar (veja Seção 6.4.3) é aplicado somente o somatório das reações nodais do modelo de elementos finitos. *Fractus2D* não aplica nenhum fator de correção para a espessura real do corpo-de-prova.

Para corpos-de-prova com configuração SE(B) (*single edge notch bend specimens*) e SE(T) (*single edge notch tension specimens*), a distância entre apoios (*span*) e a distância entre os pontos de fixação ou aplicação da carga (*load point*) também são necessárias ao cálculo dos respectivos fatores de intensidade de tensões. Estas dimensões são especificadas por intermédio dos comandos

```
specimen span < número >
```

```
load point distance < número >
```

6.3.4 Coordenadas Nodais e Incidência de Elementos

A incidência de nós dos elementos (conectividades) e as coordenadas nodais do modelo de elementos finitos (configuração indeformada) são necessárias para a manipulação de algumas in-

formações-chave tais como a construção da lista de nós que definem o flanco da trinca. Adicionalmente, o número de nós e número de elementos do modelo também devem ser especificados.

Inicialmente, o diretório que contém os arquivos de definição do modelo[†] é especificado através do comando

```
get_files (from) directory < nome do diretório: label >
```

O comando seguinte especifica o arquivo contendo as informações do modelo de elementos finitos

```
input_mesh (from) file < nome do arquivo: label >
```

O arquivo de informações do modelo possui formato livre cujas linhas de dados são apresentadas na Fig 24 onde as linhas de dados são da forma:

Linha 1

Número de nós < inteiro > Numero de elementos < inteiro >

Linhas de Coordenadas Nodais

Número do nó < inteiro >

coordenada x_1 < número >

coordenada x_2 < número >

coordenada x_3 < número >

Linhas de Elementos (elemento sólido de 8 nós)

Número do elemento < inteiro >

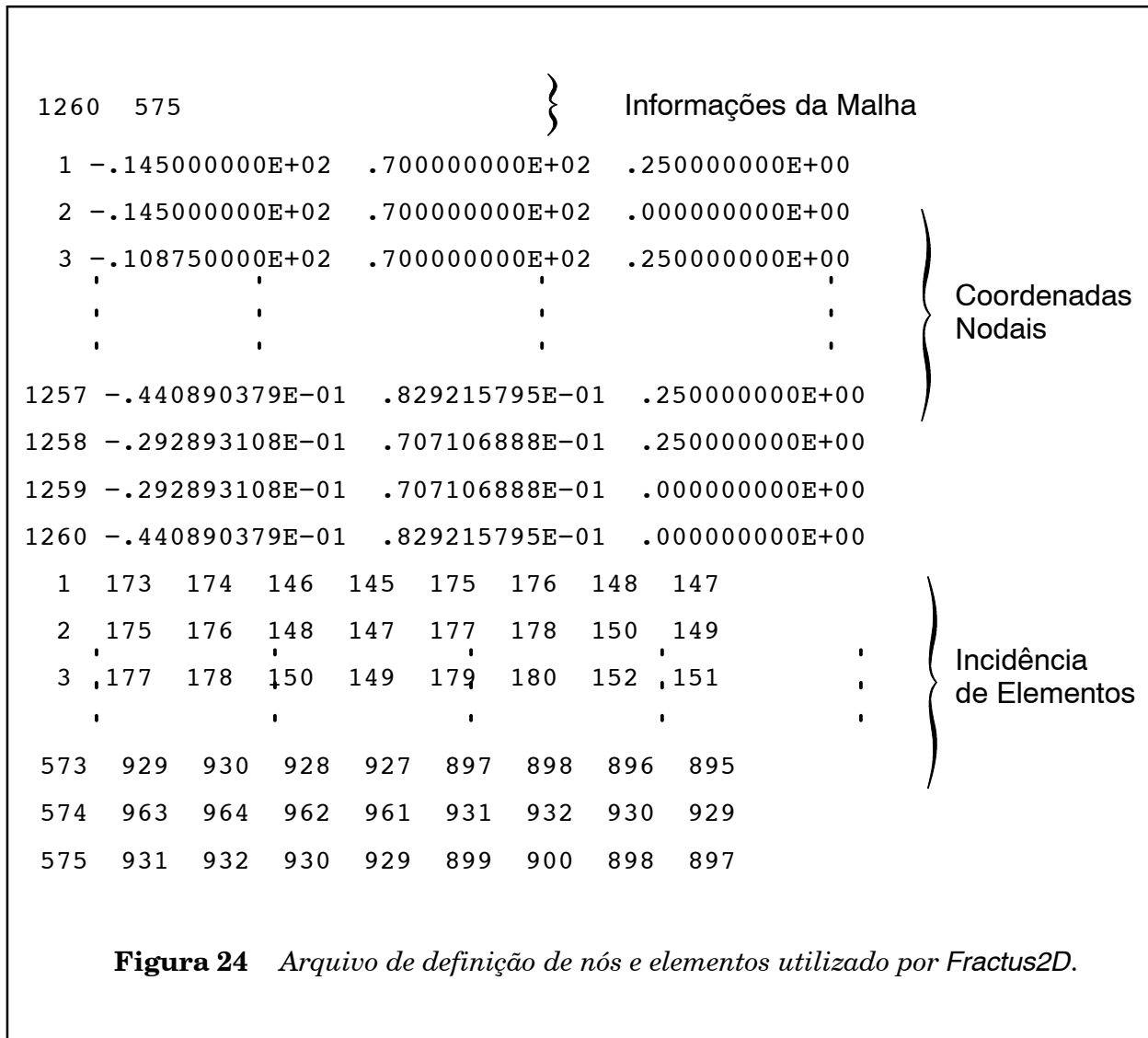
Nó n_1 < inteiro >

.....

Nó n_8 < inteiro >

Em geral, é mais simples editar o próprio arquivo de entrada utilizado pelo programa de elementos finitos WARP3D [1] para gerar o arquivo de definição do modelo na Fig. 24.

[†] O diretório de arquivos do modelo situa-se geralmente um nível abaixo do diretório de trabalho – ver Seção 4.1.



6.3.5 Formato dos Resultados de Elementos Finitos

Este comando define o formato dos resultados de elementos finitos que serão utilizados pelo programa *Fractus2D* para a determinação dos fatores η . A presente implementação de *Fractus2D* utiliza somente arquivos de resultados (deslocamentos e reações nodais) em formato binário ou ASCII compatível com o programa CAE MSc Patran [2]. A sintaxe do comando é da forma

```
format (of) fe-results patran type { ascii }
                                     { binary }
```

Fractus2D utiliza a mesma convenção de arquivos de resultados Patran adotada pelo programa de elementos finitos WARP3D [1]. Os arquivos binários Patran para os deslocamentos nodais são denominados “wnbdxxxxx”, onde “xxxxx” representa o número do incremento de carga (*load step*). Similarmente, os arquivos ASCII Patran para os deslocamentos nodais são denominados “wnfdxxxxx”. Recomenda-se aos usuários de *Fractus2D* uma leitura prévia do manual do usuário de WARP3D [1] para familiarizar-se com a forma de resultados adotada por aquele programa.

6.3.6 Parâmetro de Carga: Integral J

Os valores calculados da Integral J para passos de carga (*load steps*) especificados devem ser fornecidos ao programa através de um arquivo de dados, o qual situa-se no mesmo diretório contendo os arquivos de definição do modelo (ver Seção 4.4.1). *Estes valores devem ser consistentes com os resultados de elementos finitos gerados pela análise de tensões do componente estrutural*. O comando para especificação do arquivo é da forma

```
input j-values (from) file < nome do arquivo: label >
```

O arquivo de valores de Integral J possui formato livre cujas linhas de dados são apresentadas na Fig 25 a seguir:

```

10  0.555
20  1.145
30  , 2.008 ,
   ,      ,
   ,      ,

```

Figura 25 Arquivo de valores calculados de Integral J para os passos de carga (*load steps*) especificados utilizado pelo programa *Fractus2D*.

onde as linhas de dados são da forma:

Linha 1 ao número de passos de carga

Número do passo < inteiro > Valor da Integral J < number >

6.4 Parâmetros da Malha de Elementos Finitos

A seção `mesh based parameters` define os diversos parâmetros e informações especificamente associados à malha de elementos finitos e necessários à determinação da curva de carga *vs.* deformação, CTOD, etc.. Um bloco típico de subcomandos possui a forma

```
mesh based parameters {
  crack tip node 3020
  crack flank node set 1702 1918 1990 2026 2062 2134 ,
                      2206 2278 2350 2426 2500

  lld node 102
  lld reference node 28
  cmod node 626
  reaction forces node set 101 102
  crack plane angle nx 1 ny 0 normal ny 1
  symmetry factor displacement 2 load 1}
```

6.4.1 Definição da Trinca

A definição da trinca no modelo requer a especificação do nó de sua ponta (nó localizado no plano de simetria) e dos nós que definem o seu flanco para determinação do CTOD conforme ilustrado esquematicamente na Fig. 26. Os comandos são descritos a seguir

```
crack tip node < inteiro >
```

e

```
crack flank node set < lista de nós: inteiros >
```

Alternativamente, *Fractus2D* permite a construção automática de uma lista de nós os quais definem o flanco da trinca externamente a um raio especificado pelo usuário. O comando para especificação automática dos nós para o flanco da trinca utiliza a sintaxe

```
crack flank node set automatic ,
blunting radius < número > exclusion radius < número >
```

onde ρ_b define o raio de arredondamento inicial da ponta da trinca (*blunting radius*) e R_x é o raio de exclusão nodal (veja Fig. 26). Uma vez definidos os nós correspondentes ao flanco da trinca, *Fractus2D* utiliza um procedimento de regressão linear (método dos mínimos quadra-

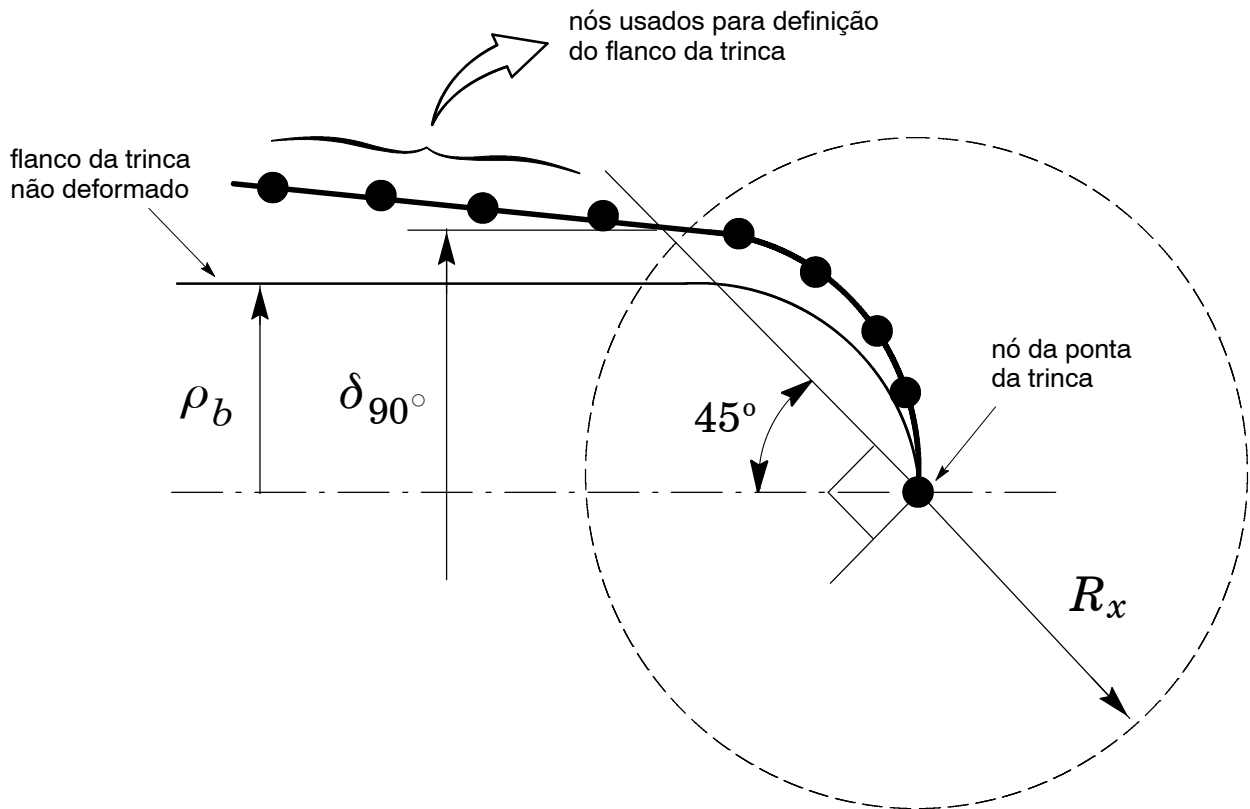


Figura 26 Determinação numérica do CTOD utilizando o método da intersecção 90°.

dos) sobre as coordenadas nodais *deformadas* para construir uma reta *média* representativa do perfil deformado da trinca. Tal funcionalidade é particularmente interessante para espécimes com trincas relativamente rasas (curtas) submetidos a cargas elevadas para os quais o perfil da trinca pode exibir algum curvamento.

Adicionalmente à especificação dos nós que descrevem o perfil da trinca, torna-se necessário definir também a orientação do seu plano e de sua normal em relação a um sistema Cartesiano conveniente. O ângulo da trinca (denotado φ) é definido pelos cossenos diretores n_x e n_y (ou simplesmente pelas componentes do vetor unitário orientado segundo o ângulo φ) conforme mostrado na Fig. 27(a). Como já enfatizado na Seção 5 anterior, a versão corrente de *Fractus2D* não permite o tratamento de espécimes com trincas inclinadas. O algoritmo de cálculo implementado em *Fractus2D* permite somente o tratamento de trincas coplanares ao eixo horizontal (x) ou vertical como indicado na Fig. 27(b). A definição da orientação do plano da trinca no plano xy utiliza o comando

onde a normal ao plano da trinca define na prática a direção de abertura (positiva ou negativa) em relação ao sistema de referência adotado.

`crack plane angle nx < número > ny < número > normal` $\left\{ \begin{array}{l} \underline{nx} < \text{número} > \\ \underline{ny} < \text{número} > \end{array} \right\}$

6.4.2 Definição da Curva de Carga vs. Deslocamento

A determinação dos fatores η deriva diretamente da construção das curvas de carga *vs.* deslocamento (CMOD e LLD) para o espécime sob análise e subsequente avaliação da área plástica sob a curva *P*-CMOD ou *P*-LLD. As rotinas numéricas implementadas em *Fractus2D* requerem a especificação de alguns parâmetros associados à malha de elementos finitos como: *i*) nós para obtenção das reações nodais correspondentes à carga aplicada (*P*); *ii*) nó correspondente ao deslocamento da linha de carga (LLD); *iii*) nó correspondente ao deslocamento da abertura da boca da trinca (CMOD) e *iv*) nó de referência correspondente ao deslocamento da linha de carga (LLD). A Fig. 28 ilustra esquematicamente a definição destes nós sobre um modelo de elementos finitos para um espécime SE(B).

A especificação do nó de referência é recomendada em corpos-de-prova SE(B) para eliminar deslocamentos espúrios associados à compressão do corpo do espécime e que não devem ser computados para a determinação do deslocamento da linha de carga (LLD). Para espécimes C(T) e SE(T), a especificação do nó de referência não é necessária e é simplesmente ignorada por *Fractus2D*. Os comandos para especificação dos nós utilizados para a construção da curva de carga *vs.* deslocamento utilizam as sintaxes

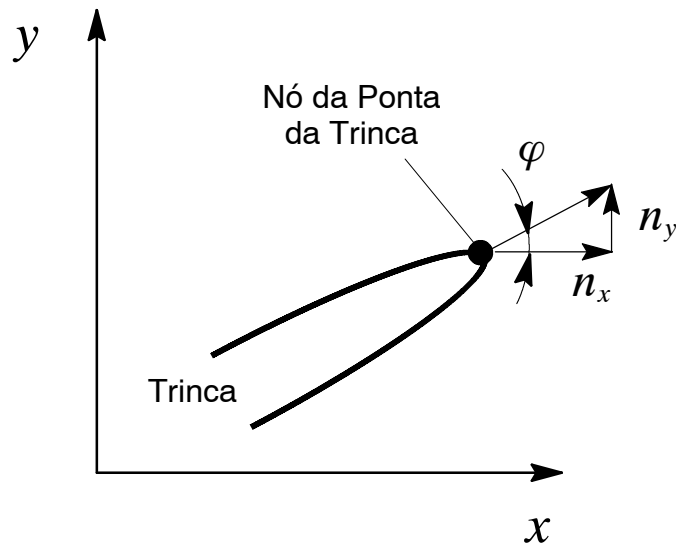
`lld node < inteiro >`

`lld reference node < inteiro >`

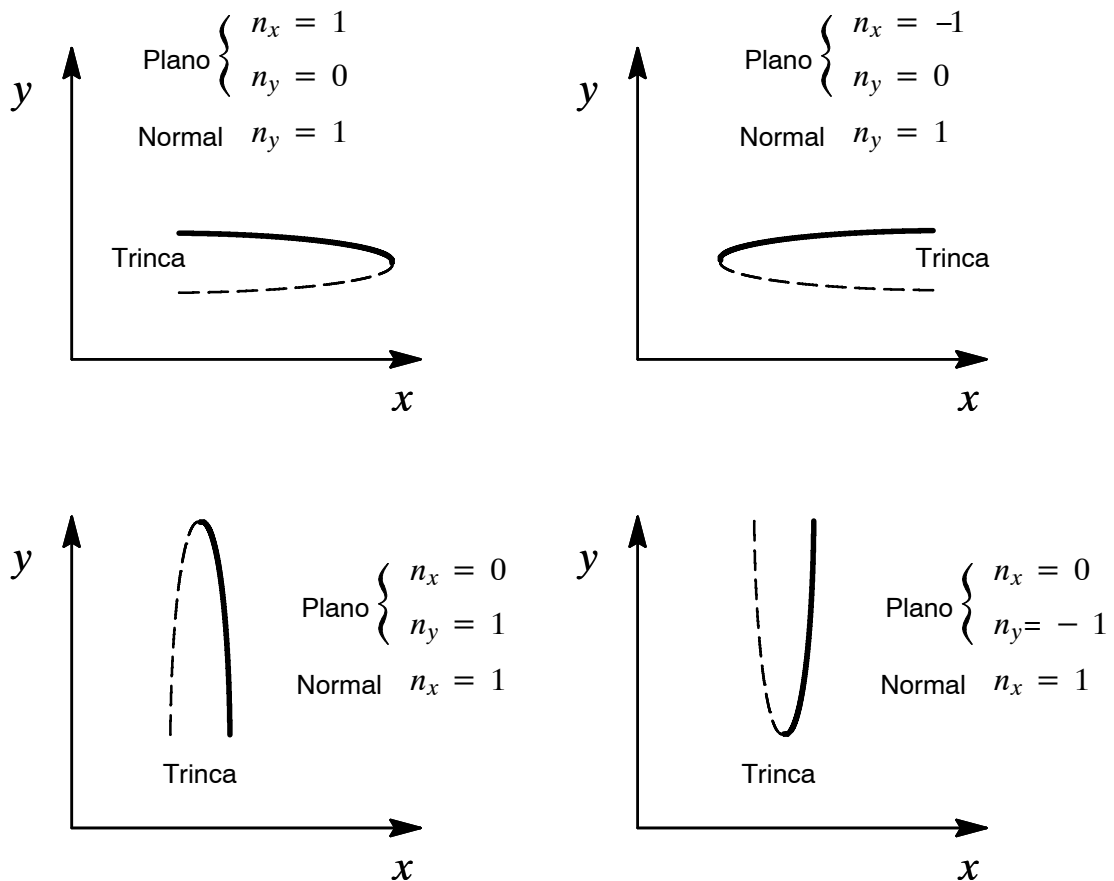
`cmod node < inteiro >`

`reaction forces node set < lista de nós: inteiros >`

Alternativamente, *Fractus2D* permite a construção automática de uma lista de nós sobre os quais as reações nodais equivalentes à carga (remota) aplicada serão calculadas. O comando para especificação automática dos nós para as reações nodais utiliza a sintaxe



(a) Representação genérica de trinca com inclinação arbitrária.



(b) Definição de orientações possíveis do plano da trinca (implementação corrente).

Figura 27 Definição da trinca utilizada por Fractus2D através da especificação do nó da ponta da trinca e dos cossenos diretores n_x e n_y relativos ao ângulo da trinca (plano x - y).

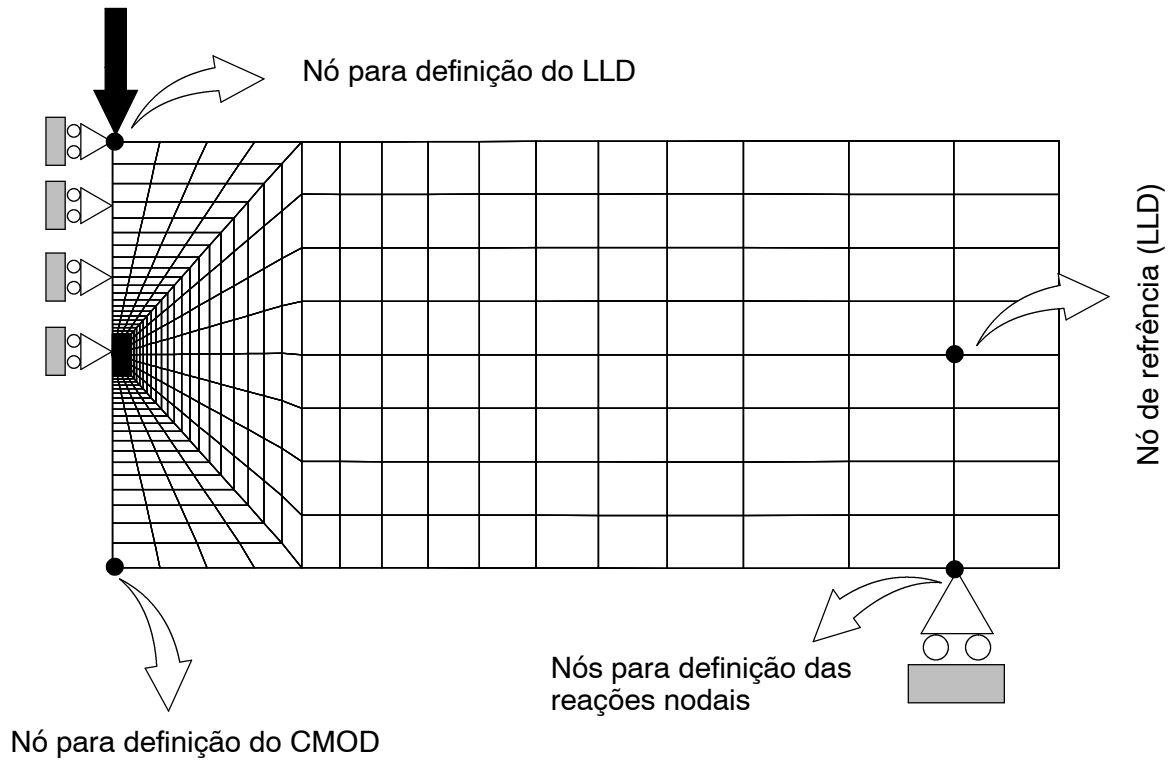


Figura 28 Determinação numérica do CTOD utilizando o método da intersecção 90°.

`reaction forces node set automatic`

6.4.3 Fator de Simetria ou Escala

Espécimes convencionais possuem planos de simetria associados à sua geometria e condições de carregamento. Quando as propriedades de simetria forem usadas para a geração do modelo de elementos finitos (veja, por exemplo, o modelo SE(B) apresentado na Fig. 28 anterior), uma fator de escala apropriado deve ser aplicado ao cálculo da carga aplicada. O comando de especificação do fator de simetria ou escala é definido por

`symmetry factor displacement < integer > load < integer >`

onde o valor *default* é 1.

Atenção especial deve ser tomada na definição do fator de simetria para os espécimes implementados na versão corrente de *Fractus2D*. Uma vez que o programa constrói a curva de carga vs. deslocamento diretamente a partir do modelo de elementos finitos analisado (utilizando, portanto, a espessura do modelo numérico – veja descrição de comandos em 6.3.3), o fator de simetria para a determinação das cargas não reflete a simetria e, conseqüentemente, o fator de escala em relação ao espécime real mas sim em relação ao modelo utilizado. Portanto, se as condições geométricas e de carregamento permitirem o modelamento de somente metade de um espécime *SE(B)*, o seu fator de simetria será 2 para o deslocamento e forças. Entretanto, o fator de simetria referente ao modelamento de somente metade de espécimes *C(T)* e *SE(T)* será 2 para deslocamentos e 1 para forças.

6.5 Parâmetros de Análise

A seção `analysis parameter` define os parâmetros de análise e de controle utilizados para a execução do programa *Fractus2D*. Um bloco típico de subcomandos possui a forma

```
analysis parameters {
  node tolerance 0.0001
  plastic area ratio 0.1
  ctod constraint factor 2
  ctod model node flank regression
  yield stress 400
  tensile stress 580
  use elastic compliance on
  plot load-displacement on format short
  save load-displacement on
  compute eta factors on steps 100-500 by 50 }
```

6.5.1 Tolerância de Busca Nodal

A construção automática da lista de nós para definição das reações nodais ou do flaco da trinca utiliza uma rotina de “busca” nodal onde as coordenadas dos nós incluídos na lista satisfazem a condições (coordenadas) especificadas pelo usuário. O valor da tolerância de equivalência nodal utilizado nas análises é especificado através do comando

```
node tolerance < número >
```

Quando o valor da tolerância de equivalência nodal não for especificado pelo usuário, o valor padrão $\xi = 1e^{-4}$ é assumido pelo programa.

6.5.2 Fração da Área Plástica

Para baixos níveis de carregamento (J ou CTOD), os valores das áreas elástica e plástica apresentam magnitudes similares o que potencialmente afeta os valores correspondentes dos fatores η_J^k e η_δ^k . O procedimento implementado em *Fractus2D* considera somente valores numéricos da Integral J (CTOD) normalizada *vs.* área plástica normalizada para os quais o critério $A_p \leq \beta(A_e + A_p) = \beta A_t$, onde o valor β é a fração da área plástica, é satisfeito (ver Seção 3.3). A especificação do valor adotado para β utiliza o comando

```
plastic area ratio < número >
```

onde valores de $\beta = 0.1 \sim 0.2$ são freqüentemente usados. O valor padrão adotado por *Fractus2D* é $\beta = 0.1$.

6.5.3 Coeficiente Angular Linear da Curva de Carga *vs.* Deslocamento

Conforme descrito na Seção 3.3, a separação da área total sob a curva de carga *vs.* deslocamento em uma fração elástica e plástica depende da determinação do coeficiente angular da região elástica, Φ_e , obtido diretamente a partir dos resultados numéricos P *vs.* V (ou, equivalentemente, P *vs.* Δ). Alternativamente, *Fractus2D* implementa funções de flexibilidade elástica correspondentes aos espécimes analisados pelo programa. Estas funções de flexibilidade permitem ao programa *Fractus2D* determinar o valor do coeficiente angular da região elástica, Φ_e , diretamente a partir de polinômios de ajuste obtidos para análises elásticas dos espécimes (ver ref. [43] para maiores detalhes). A especificação do procedimento para determinação do coeficiente Φ_e é feita por meio do comando

```
use elastic compliance { on }
                        { off }
```

onde o valor padrão adotado por *Fractus2D* é *on*.

A presente versão de *Fractus2D* possibilita a utilização de funções de flexibilidade elástica somente para o deslocamento caracterizado pelo CMOD e LLD de espécimes convencionais 1-T C(T) e SE(T), e caracterizado somente pelo CMOD para espécimes 1-T SE(B), com as seguintes

dimensões geométricas: *i*) Espécimes SE(B) : $0.1 \leq a/W \leq 0.5$ e $S/W = 4$; *ii*) Espécimes SE(T) fixados por garras: $4 \leq H/W \leq 20$; *iii*) Espécimes SE(T) fixados por pinos: $H/W \geq 4$; *iv*) Espécimes C(T) : $0.4 \leq a/W \leq 0.7$ e $W/B = 2$. Nestas expressões, a é o comprimento da trinca, B é a espessura, W é a largura do espécime, S é a distância entre apoios para o espécime SE(B) e H é a distância entre pontos de aplicação da carga (distância entre as garras de fixação ou centro dos pinos) para o espécime SE(T) – ver Fig. 9.

6.5.4 Propriedades Mecânicas do Material

A especificação das propriedades mecânicas do material sob análise é necessária para a construção das curvas normalizadas de J (CTOD) vs. área plástica e, principalmente, para a determinação do fator η_δ . As propriedades mecânicas são especificadas simplesmente por meio dos comandos

`young modulus < número >`

`poison ratio < número >`

`yield stress < número >`

`tensile stress < número >`

A especificação do limite de resistência, σ_{uts} , é necessária somente quando o valor do CTOD e, conseqüentemente, do fator η_δ utilizar a tensão de fluxo, σ_f . Em análises convencionais nas quais somente a tensão de escoamento for utilizada, o usuário poderá especificar $\sigma_{uts} = \sigma_{ys}$ ou simplesmente omitir o comando `tensile stress` (mais recomendável).

6.5.5 Fator de Restrição Plástica para o CTOD Elástico

O fator de restrição plástica, m , é necessário para determinação da componente elástica do CTOD (ver Seção 3.3). Este parâmetro é especificado por intermédio do comando

`ctod constraint factor < número >`

O valor padrão para o fator de restrição plástica é $m = 2$.

6.5.6 Construção de Fatores Eta

A especificação dos passos de carga (*load steps*) sobre os quais os fatores η são determinados utiliza o seguinte comando

```
compute eta factors steps < lista de passos: inteiros >
```

Quando os passos de carga não forem especificados pelo usuário, o programa utiliza todos os passos de carga especificados no arquivo de valores de Integral J (ver Seção 6.3.6).

6.5.7 Curva de Carga vs. Deslocamento

A exibição dos valores de carga e deslocamento (CMOD e LLD) determinados para cada passo de carga (*load step*) no arquivo de saída do programa é freqüentemente interessante e necessária à interpretação adequada da análise. *Fractus2D* fornece estes valores simplesmente por meio da especificação do comando

```
plot load-displacement { on } format { short }
                        { off }      { long }
```

onde o valor padrão adotado é `on` com formato `short`. O formato `long` exhibe as componentes elástica e plástica do CMOD e LLD.

Adicionalmente, *Fractus2D* permite salvar as informações essenciais da curva de carga vs. deslocamento assim como os parâmetros J e CTOD associados a cada passo de carga (*load step*) em um arquivo texto no diretório de trabalho. A sintaxe de comando é simplesmente

```
save load-displacement { on }
                        { off }
```

6.5.8 Procedimento para Determinação do CTOD

Conforme descrito na Seção 3.3.3, *Fractus2D* implementa duas alternativas para a determinação numérica do CTOD: *i*) método da intersecção 90° (δ_{90°) e *ii*) método da tangente (δ_T) utilizando o conceito de rótula plástica. Como refinamento adicional à obtenção do valor de δ_{90° , o programa permite especificar dois subprocedimentos distintos para determinação deste

parâmetro: *i*) intersecção da reta de inclinação 45° com a reta “média” definindo o flanco da trinca (reta de regressão linear sobre o flanco deformado – veja Seção 3.3.3 e 6.4.1) e *ii*) intersecção da reta de inclinação 45° com uma reta definida por dois nós consecutivos sobre o flanco da trinca. A especificação do procedimento para determinação do CTOD utiliza os comandos

```
ctod model { node flank regression }
            { flank intercept   }
            { tangent           }
```

onde o valor padrão adotado é `node flank regression` o qual corresponde método da intersecção 90° utilizando uma reta de regressão para definir o flanco deformado da trinca.

6.5.9 Determinação do Fator Rotacional Plástico

Embora não seja um parâmetro específico associado à análise de fatores *eta* para corpos-de-prova convencionais, *Fractus2D* permite uma facilidade adicional diretamente derivada da obtenção do parâmetro CTOD para determinação do fator rotacional plástico, r_p , conforme descrito na Seção 3.3.4. O usuário pode especificar a determinação do fator r_p utilizando o comando

```
compute rotational factor { on }
                          { off }
```

onde o valor-padrão adotado é `off`.

Como mencionado anteriormente, a presente versão de Fractus2D somente permite determinar o fator r_p para os espécimes analisados pelo programa quando o modelo tangente for especificado para o cálculo do CTOD.

6.5.10 Versão do Programa FEA WARP3D

Embora não existam diferenças significativas entre os resultados de deslocamentos e forças nodais (incluindo tensões e deformações) para as diferentes versões do programa de elementos finitos WARP3D [1], há algumas pequenas diferenças de formato dos arquivos “wnfdxxxx” (“wnbdxxxx”) e “wnfrxxxx” (“wnbrxxxx”) – ver Seção 6.3.5. As versões correntemente suportadas por *Fractus2D* são 14.x, 15.x, 16.x e 17.x. A especificação da versão utilizada do programa WARP3D é feita pelo comando

`warp3d release` { V14 }
 { V15 }
 { V16 }
 { V17 }

onde o valor-padrão adotado para a versão do programa WARP3D é v15.

Exemplos Ilustrativos

7.1 Trajetórias J-Q para um Corpo-de-Prova C(T)

Esta seção ilustra a aplicação do programa *Fractus2D* para a construção da trajetória *J-Q* para um corpo de prova convencional (sem entalhe lateral) *compact tension* 1-T C(T) com trinca profunda caracterizada pela relação $a/W = 0.6$, onde a é o comprimento (nominal) da trinca e W é a largura do espécime. A Fig. 29 apresenta o modelo de elementos finitos utilizado e as dimensões principais do espécime.

A análise numérica 2D sob estado plano de deformação é conduzida para o modelo C(T) apresentado na Fig. 29 utilizando-se o programa de elementos finitos não linear WARP3D [1]. Condições de simetria permitem a construção de somente metade da amostra com vínculos apropriados impostos sobre o ligamento ($W - a$) da trinca. A metade simétrica destes modelos possui ~ 2000 elementos isoparamétricos 3D de 8 nós constituindo, portanto, uma única camada sobre o plano *X-Y* da amostra, conforme indicado na Fig. 29. As condições correspondentes ao estado plano de deformações são obtidas através da imposição de deslocamentos (vínculos) $w = 0$ sobre cada nó do malha. A ponta da trinca é modelada através de uma configuração adequada à acomodação do arredondamento inicial (*blunted tip*) com raio $\rho_0 = 2.5\mu\text{m}$ (0.0025 mm); esta dimensão permite obter os campos de tensões e deformações com o elevado grau de precisão necessário às análises e, ao mesmo tempo, acelera a convergência dos algoritmos de plasticidade utilizados nas soluções numéricas.

O comportamento constitutivo adotado para o material utiliza o modelo de *von Mises* incorporando teoria de pequenas deformações (*small geometry changes - SGC*). A resposta tensão ($\bar{\sigma}$) vs. deformação (ϵ) *uniaxial* do material segue um modelo não linear incorporando o encruamento plástico na forma

$$\frac{\epsilon}{\epsilon_0} = \frac{\bar{\sigma}}{\bar{\sigma}_0} \quad \epsilon \leq \epsilon_0 ; \quad (30)$$

$$\frac{\epsilon}{\epsilon_0} = \left(\frac{\bar{\sigma}}{\bar{\sigma}_0} \right)^n \quad \epsilon > \epsilon_0 \quad (31)$$

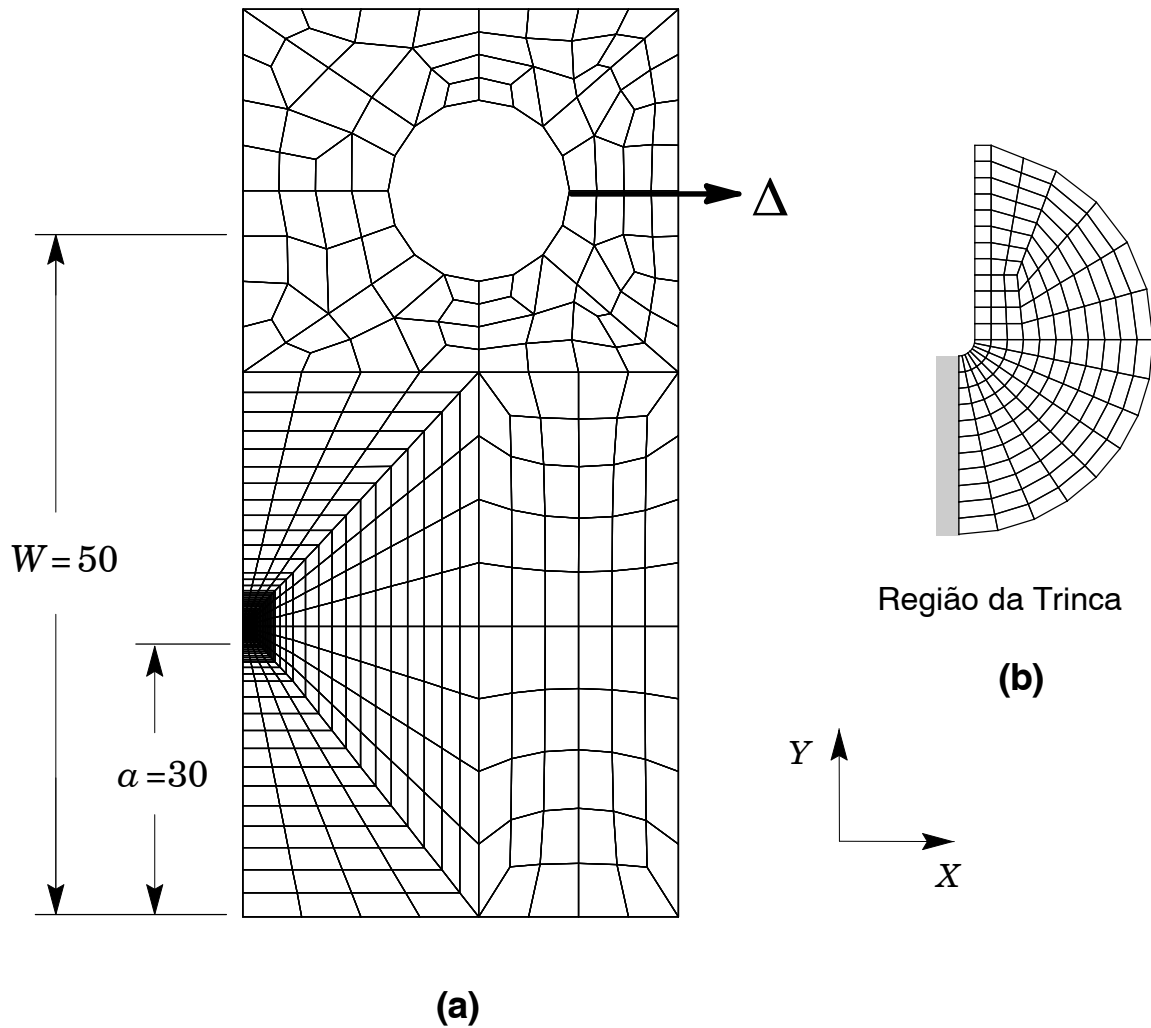


Figura 29 (a) Modelo de elementos finitos 2D para o espécime C(T) com $aW=0.6$; (b) Detalhe da região da trinca (blunted notch).

onde σ_0 e ϵ_0 são a tensão e deformação de referência (as quais representam as tensões e deformações de escoamento no presente contexto), n é o coeficiente de encruamento do material. A Eq. (31) é também conhecida como modelo de *Ramberg-Osgood* para materiais elasto-plásticos com encruamento e representa uma das descrições mais utilizadas para o regime pós-escoamento de aços ferríticos. Ainda nesta equação, o parâmetro n define o coeficiente de *Ramberg-Osgood*, o qual quantifica o nível de encruamento do material durante o processo de deformação plástica. As soluções numéricas para as análises do modelo C(T) são obtidas para um material com $n = 10$ (encruamento moderado) e $E/\sigma_0 = 500$ com $E = 206$ GPa e $\nu = 0.3$. Estas propriedades tipificam um aço API X60, o qual é bastante empregado em dutos terrestres e submarinos.

As soluções numéricas para os campos de tensões e deformações em trincas estacionárias obtidas através do código não linear 3D de elementos finitos WARP3D [1] incluem: (1) modelo constitutivo de von Mises implementado em um contexto de pequenas deformações e grandes deformações (não linearidades geométricas); (2) dimensionamento automático do incremento de carga (*load step*); (3) cálculo da Integral J utilizando um procedimento de integração numérica. WARP3D emprega uma formulação Lagrangiana (atualização contínua da deformação do sólido) adequada para elementos isoparamétricos sólidos possuindo somente deslocamentos nodais translacionais.

Os modelos (sólidos) de fratura são construídos através de elementos isoparamétricos 3D com 8 nós. A utilização de uma forma modificada da matriz deformação-deslocamento, chamada matriz $\bar{\mathbf{B}}$ [1], em sua formulação evita problemas com “travamento” (*mesh locking*) e modos espúrios quando as deformações evoluem para regimes completamente plásticos e incompressíveis. Os termos dilatacionais da matriz deformação-deslocamento original, \mathbf{B}^{dil} , são substituídos por termos dilatacionais *médios* (em relação ao volume do elemento), $\bar{\mathbf{B}}^{dil}$, os quais produzem tensão uniforme sobre o elemento e minimizam o “travamento”. A matriz \mathbf{B} portanto possui a forma $\bar{\mathbf{B}} = \mathbf{B}^{dev} + \bar{\mathbf{B}}^{dil}$, onde \mathbf{B}^{dev} denota a contribuição deviatórica original (não modificada). Adicionalmente, a forma $\bar{\mathbf{B}} = \mathbf{B}^{dev} + \bar{\mathbf{B}}^{dil} + \zeta [\mathbf{B}^{dil} - \bar{\mathbf{B}}^{dil}]$, com ζ tipicamente igual a 0.05, é freqüentemente usada para aumentar a estabilização numérica da formulação.

O valor *local* da taxa de liberação de energia mecânica, expressa pela Integral J , em pontos ao longo da frente de trinca é dada por [NO TAG]

$$J = \lim_{\Gamma \rightarrow 0} \int_{\Gamma} \left[\mathcal{W} n_1 - P_{ji} \frac{\partial u_i}{\partial X_1} n_j \right] d\Gamma \quad (32)$$

onde Γ denota um contorno fechado definido sobre um plano normal à frente da trinca (configuração indeformada), iniciando na face inferior da trinca e terminando na sua face superior, n_j é o vetor norma à superfície Γ (apontado para fora), \mathcal{W} denota a densidade de energia de deformação por unidade de volume indeformado, P_{ij} e u_i são as componentes Cartesianas do tensor de tensões (assimétricas) de Piola-Kirchoff e dos deslocamentos no sistema de coordenadas localizado na frente da trinca. O procedimento numérico para determinação da integral expressa pela Eq. (32) utiliza a estratégia de integral de domínio proposta por Moran e Shih [NO TAG]. Para modelos sólidos 3D, o valor médio da Integral J sobre a espessura é determinado para domínios de integração definidos fora da região próxima à ponta da trinca, onde intensa deformação plástica e carregamento não proporcional invalidam a sua definição; os valores calculados, portanto, retém uma forte independência do caminho de integração. Estes valores de J concordam, em geral, com os valores obtidos por esquemas de avaliação experimental (*e.g.*, método *eta da* ASTM E1820 [NO TAG]). Conseqüentemente, a Integral J numérica fornece um

parâmetro conveniente e robusto para caracterizar (localmente) a intensidade do carregamento remoto sobre a frente da trinca.

O arquivo de saída do programa *Fractus2D* apresentado a seguir especifica os parâmetros de entrada e blocos de comandos, assim como os resultados principais e controles de execução. A Fig. 30 a trajetória J - Q para o modelo C-T analisado obtido através do programa *Fractus2D*.

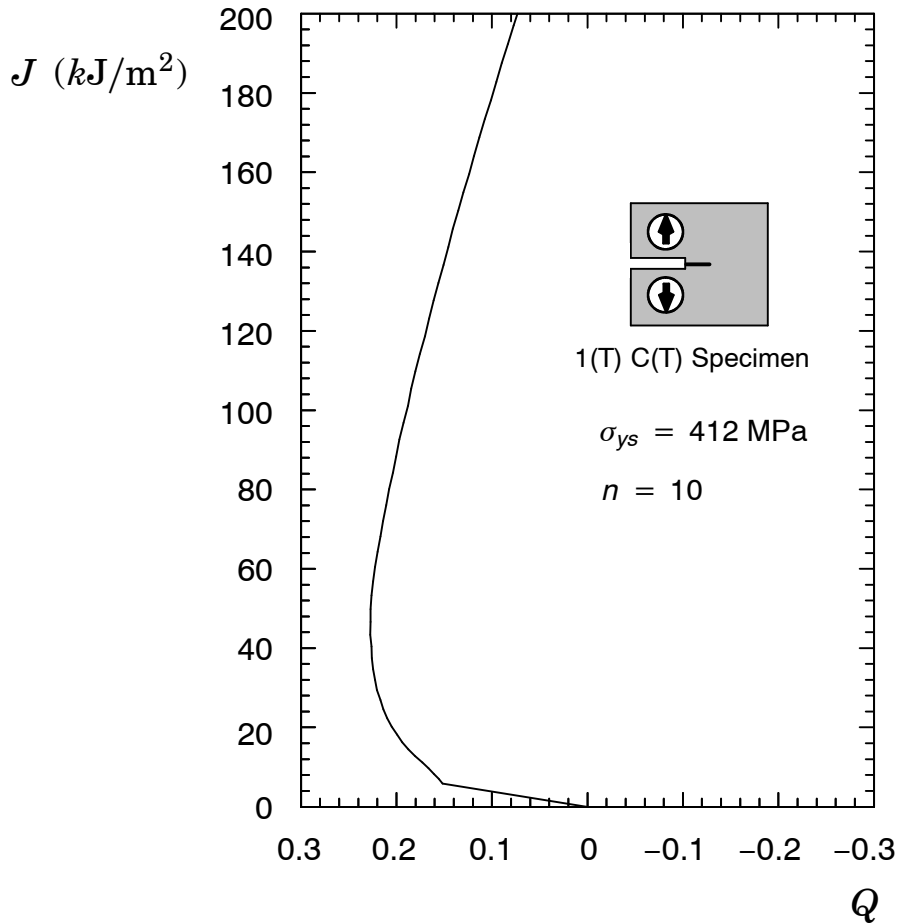


Figura 30 Trajetórias J - Q para amostras $C(T)$ com $a/W=0.6$ para o material com $n=10$.

>>> Initializing global variables and setting default values ...

>>> Scanning input parameters ...

```

c
c   FRACTUS Example
c
c   Computation of J-Q curves for a C(T) specimen
c   with a/W = 0.6 and n=10 material.
c
c   Elastic-plastic analysis with SGC and power hardening law
c
c
c   crack analysis type jq-curves
c
c
c   ssy model {
      get files from directory ssy_small_n10_nx1ny0
      input mesh from file ssy_small_coor
      format fe-results patran type ascii
      crack tip node 1
      crack plane angle nx 1 ny 0
      near tip elements automatic maximum radius adaptive }
c
c
c   finite body {
      structure CT_Specimen
      get files from directory ct_aw6_n10_nx1ny0
      input mesh from file ct_aw6_coor
      input j-values from file ct_aw6_jvalues
      format fe-results patran type ascii
      crack tip node 2510
      crack plane angle nx 1 ny 0
      near tip elements automatic maximum radius adaptive }
c
c
c   analysis parameters {
      nondimensional radius 2.0
      adaptive radius factor 1.25
      yield stress 412
      node tolerance 0.0001
      reference j-int 44.53 at load step 100
      compute j-q curves on steps 100-900 by 10
      plot stress on steps 100-500 by 100 }
c
end

```

>>> Checking input parameters ...

>>> Opening files ...

>>> Processing crack tip elements ...

>>> Computing J-Q trajectories ...

```

>> processing step 1 .... load step = 100
>> processing step 2 .... load step = 110
>> processing step 3 .... load step = 120
>> processing step 4 .... load step = 130
>> processing step 5 .... load step = 140
>> processing step 6 .... load step = 150
>> processing step 7 .... load step = 160
>> processing step 8 .... load step = 170
>> processing step 9 .... load step = 180

```

```

>> processing step 10 .... load step = 190
>> processing step 11 .... load step = 200
>> processing step 12 .... load step = 210
>> processing step 13 .... load step = 220
>> processing step 14 .... load step = 230
>> processing step 15 .... load step = 240
>> processing step 16 .... load step = 250
>> processing step 17 .... load step = 260
>> processing step 18 .... load step = 270
>> processing step 19 .... load step = 280
>> processing step 20 .... load step = 290
>> processing step 21 .... load step = 300
>> processing step 22 .... load step = 310
>> processing step 23 .... load step = 320
>> processing step 24 .... load step = 330
>> processing step 25 .... load step = 340
>> processing step 26 .... load step = 350
>> processing step 27 .... load step = 360
>> processing step 28 .... load step = 370
>> processing step 29 .... load step = 380
>> processing step 30 .... load step = 390
>> processing step 31 .... load step = 400
>> processing step 32 .... load step = 410
>> processing step 33 .... load step = 420
>> processing step 34 .... load step = 430
>> processing step 35 .... load step = 440
>> processing step 36 .... load step = 450
>> processing step 37 .... load step = 460
>> processing step 38 .... load step = 470
>> processing step 39 .... load step = 480
>> processing step 40 .... load step = 490
>> processing step 41 .... load step = 500
>> processing step 42 .... load step = 510
>> processing step 43 .... load step = 520
>> processing step 44 .... load step = 530
>> processing step 45 .... load step = 540
>> processing step 46 .... load step = 550
>> processing step 47 .... load step = 560
>> processing step 48 .... load step = 570
>> processing step 49 .... load step = 580
>> processing step 50 .... load step = 590
>> processing step 51 .... load step = 600
>> processing step 52 .... load step = 610
>> processing step 53 .... load step = 620
>> processing step 54 .... load step = 630
>> processing step 55 .... load step = 640
>> processing step 56 .... load step = 650
>> processing step 57 .... load step = 660
>> processing step 58 .... load step = 670
>> processing step 59 .... load step = 680
>> processing step 60 .... load step = 690
>> processing step 61 .... load step = 700
>> processing step 62 .... load step = 710
>> processing step 63 .... load step = 720
>> processing step 64 .... load step = 730
>> processing step 65 .... load step = 740
>> processing step 66 .... load step = 750
>> processing step 67 .... load step = 760
>> processing step 68 .... load step = 770
>> processing step 69 .... load step = 780
>> processing step 70 .... load step = 790
>> processing step 71 .... load step = 800
>> processing step 72 .... load step = 810
>> processing step 73 .... load step = 820

```

```

>> processing step 74 .... load step = 830
>> processing step 75 .... load step = 840
>> processing step 76 .... load step = 850
>> processing step 77 .... load step = 860
>> processing step 78 .... load step = 870
>> processing step 79 .... load step = 880
>> processing step 80 .... load step = 890
>> processing step 81 .... load step = 900

```

```
>>> Plotting Near-Tip Stress Distribution ...
```

```

>> plotting load step ... 100
>> plotting load step ... 200
>> plotting load step ... 300
>> plotting load step ... 400
>> plotting load step ... 500

```

```
*****
```

J - Q T R A J E C T O R I E S

```
*****
```

```
>>> Structure : CT_Specimen
```

Step	Load Step	J Value	Q Parameter
1	100	5.8260	0.1519
2	110	7.0070	0.1561
3	120	8.2940	0.1611
4	130	9.6900	0.1667
5	140	11.1900	0.1731
6	150	12.7900	0.1808
7	160	14.4900	0.1879
8	170	16.2900	0.1943
9	180	18.2000	0.1994
10	190	20.2100	0.2053
11	200	22.3400	0.2101
12	210	24.5800	0.2138
13	220	26.9300	0.2173
14	230	29.3900	0.2206
15	240	31.9700	0.2226
16	250	34.6600	0.2246
17	260	37.4700	0.2262
18	270	40.3900	0.2266
19	280	43.4200	0.2278
20	290	46.5800	0.2275
21	300	49.8600	0.2274
22	310	53.2700	0.2263
23	320	56.8300	0.2248
24	330	60.5100	0.2226
25	340	64.2800	0.2201
26	350	68.1400	0.2171
27	360	72.0600	0.2143
28	370	76.0500	0.2108
29	380	80.0900	0.2079
30	390	84.2000	0.2041
31	400	88.3700	0.2007
32	410	92.5900	0.1971
33	420	96.8400	0.1928

34	430	101.1000	0.1883
35	440	105.4000	0.1848
36	450	109.8000	0.1805
37	460	114.2000	0.1758
38	470	118.6000	0.1706
39	480	123.1000	0.1661
40	490	127.6000	0.1614
41	500	132.1000	0.1564
42	510	136.6000	0.1510
43	520	141.1000	0.1460
44	530	145.7000	0.1410
45	540	150.3000	0.1356
46	550	154.9000	0.1300
47	560	159.5000	0.1243
48	570	164.1000	0.1191
49	580	168.8000	0.1133
50	590	173.4000	0.1077
51	600	178.1000	0.1015
52	610	182.8000	0.0957
53	620	187.5000	0.0899
54	630	192.3000	0.0837
55	640	197.0000	0.0775
56	650	201.7000	0.0714
57	660	206.5000	0.0653
58	670	211.3000	0.0591
59	680	216.1000	0.0528
60	690	220.9000	0.0463
61	700	225.7000	0.0400
62	710	230.5000	0.0339
63	720	235.3000	0.0276
64	730	240.2000	0.0210
65	740	245.0000	0.0146
66	750	249.9000	0.0081
67	760	254.8000	0.0017
68	770	259.6000	-0.0046
69	780	264.5000	-0.0111
70	790	269.4000	-0.0178
71	800	274.3000	-0.0242
72	810	279.3000	-0.0309
73	820	284.2000	-0.0374
74	830	289.1000	-0.0439
75	840	294.1000	-0.0508
76	850	299.0000	-0.0572
77	860	304.0000	-0.0639
78	870	309.0000	-0.0706
79	880	313.9000	-0.0771
80	890	318.9000	-0.0839
81	900	323.9000	-0.0906

>>>> Fractus2D: normal termination.

7.2 Fatores E_t para um Corpo-de-Prova SE(T)

Esta seção ilustra a aplicação do programa *Fractus2D* para a determinação dos fatores η_J^{CMOD} e η_δ^{CMOD} para um corpo-de-prova *single edge notch under tension* SE(T) carregado por pino com trinca profunda caracterizada pela relação $a/W = 0.5$ e $H/W = 6$, onde a é o comprimento (nominal) da trinca, W é a largura do corpo-de-prova e H representa a distância entre pontos de aplicação da carga (pinos). A Fig. 31 apresenta o modelo de elementos finitos utilizado e as dimensões principais do corpo-de-prova. Os detalhes referentes às análises de elementos finitos são essencialmente similares ao descrito no exemplo anterior para o espécime C(T). O arquivo de saída do programa *Fractus2D* apresentado a seguir especifica os parâmetros de entrada e blocos de comandos, assim como os resultados principais e controles de execução.

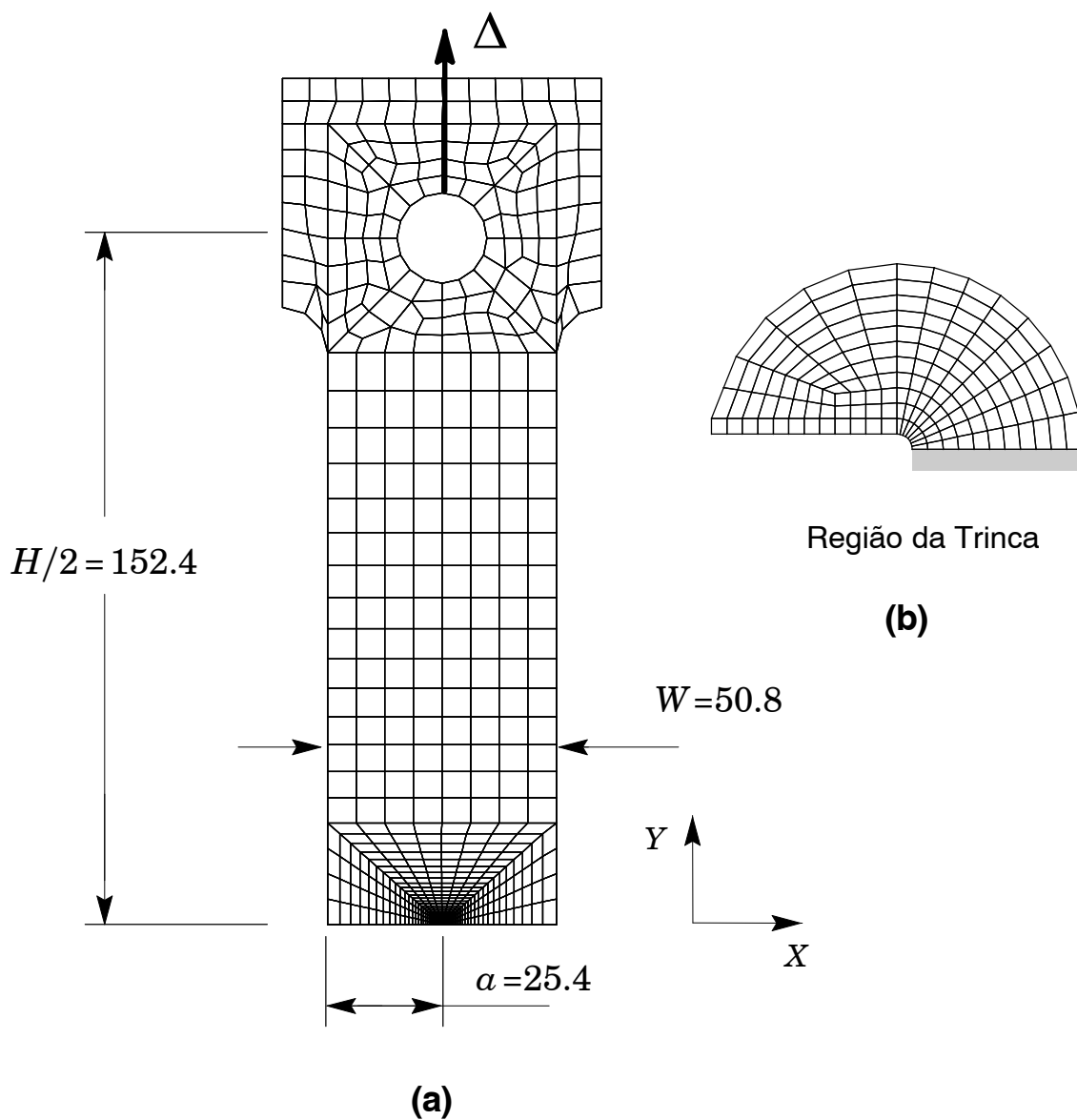


Figura 31 (a) Modelo de elementos finitos 2D para o corpo-de-prova SE(T) com $a/W = 0.5$ e $H/W = 0.5$; (b) Detalhe da região da trinca (blunted notch).

```

>>> Initializing global variables and setting default values ...
>>> Scanning input parameters ...

c
c   ETA Factor Analysis
c
c   Computation of eta factor for a Pin-Loaded SE(T) specimen
c   with a/W = 0.5 and n=10 material.
c
c   Elastic-plastic analysis with SGC and power hardening law
c
c
c   crack analysis type eta-factor
c
c
c   crack configuration {
      structure set_aw5_pin
      crack geometry pinned set ! Pin-loaded SE(T)
      specimen thickness 1.0 ! Espessura do modelo FE
      specimen width 50.8
      load point distance 304.8 ! h/W=6
      crack size 25.4
      get files from directory setaw05_nx1ny0
      input mesh from file setaw5_coor
      input j-values from file setaw5_jvalues
      format fe-results patran type ascii }
c
c   mesh based parameters {
      crack tip node 3020
      crack flank node set automatic blunting radius 0.0025 ,
                                exclusion radius 0.3
      lld node 102
      cmod node 626
      reaction forces node set 101 102
      crack plane angle nx 1 ny 0 normal ny 1
      symmetry factor displacement 2 load 1 }
c
c   analysis parameters {
      yield stress 412
      poisson ratio 0.3
      young modulus 206000
      ctod constraint factor 2
      ctod model node flank regression ! 90 degree CTOD
      node tolerance 0.0001
      plastic area ratio 0.10
      use elastic compliance on
      plot load-disp on format short
      compute eta factors steps 10-1000 by 10 }
c
end

>>> Checking input parameters ...

>>> Opening files ...

>>> Building load displacement curve and determining CTOD ...

>> processing step 1 .... load step = 10
>> processing step 2 .... load step = 20
>> processing step 3 .... load step = 30
>> processing step 4 .... load step = 40
>> processing step 5 .... load step = 50
>> processing step 6 .... load step = 60

```

```
>> processing step 7 .... load step = 70
>> processing step 8 .... load step = 80
>> processing step 9 .... load step = 90
>> processing step 10 .... load step = 100
>> processing step 11 .... load step = 110
>> processing step 12 .... load step = 120
>> processing step 13 .... load step = 130
>> processing step 14 .... load step = 140
>> processing step 15 .... load step = 150
>> processing step 16 .... load step = 160
>> processing step 17 .... load step = 170
>> processing step 18 .... load step = 180
>> processing step 19 .... load step = 190
>> processing step 20 .... load step = 200
>> processing step 21 .... load step = 210
>> processing step 22 .... load step = 220
>> processing step 23 .... load step = 230
>> processing step 24 .... load step = 240
>> processing step 25 .... load step = 250
>> processing step 26 .... load step = 260
>> processing step 27 .... load step = 270
>> processing step 28 .... load step = 280
>> processing step 29 .... load step = 290
>> processing step 30 .... load step = 300
>> processing step 31 .... load step = 310
>> processing step 32 .... load step = 320
>> processing step 33 .... load step = 330
>> processing step 34 .... load step = 340
>> processing step 35 .... load step = 350
>> processing step 36 .... load step = 360
>> processing step 37 .... load step = 370
>> processing step 38 .... load step = 380
>> processing step 39 .... load step = 390
>> processing step 40 .... load step = 400
>> processing step 41 .... load step = 410
>> processing step 42 .... load step = 420
>> processing step 43 .... load step = 430
>> processing step 44 .... load step = 440
>> processing step 45 .... load step = 450
>> processing step 46 .... load step = 460
>> processing step 47 .... load step = 470
>> processing step 48 .... load step = 480
>> processing step 49 .... load step = 490
>> processing step 50 .... load step = 500
>> processing step 51 .... load step = 510
>> processing step 52 .... load step = 520
>> processing step 53 .... load step = 530
>> processing step 54 .... load step = 540
>> processing step 55 .... load step = 550
>> processing step 56 .... load step = 560
>> processing step 57 .... load step = 570
>> processing step 58 .... load step = 580
>> processing step 59 .... load step = 590
>> processing step 60 .... load step = 600
>> processing step 61 .... load step = 610
>> processing step 62 .... load step = 620
>> processing step 63 .... load step = 630
>> processing step 64 .... load step = 640
>> processing step 65 .... load step = 650
>> processing step 66 .... load step = 660
>> processing step 67 .... load step = 670
>> processing step 68 .... load step = 680
>> processing step 69 .... load step = 690
>> processing step 70 .... load step = 700
```

```

>> processing step 71 .... load step = 710
>> processing step 72 .... load step = 720
>> processing step 73 .... load step = 730
>> processing step 74 .... load step = 740
>> processing step 75 .... load step = 750
>> processing step 76 .... load step = 760
>> processing step 77 .... load step = 770
>> processing step 78 .... load step = 780
>> processing step 79 .... load step = 790
>> processing step 80 .... load step = 800
>> processing step 81 .... load step = 810
>> processing step 82 .... load step = 820
>> processing step 83 .... load step = 830
>> processing step 84 .... load step = 840
>> processing step 85 .... load step = 850
>> processing step 86 .... load step = 860
>> processing step 87 .... load step = 870
>> processing step 88 .... load step = 880
>> processing step 89 .... load step = 890
>> processing step 90 .... load step = 900
>> processing step 91 .... load step = 910
>> processing step 92 .... load step = 920
>> processing step 93 .... load step = 930
>> processing step 94 .... load step = 940
>> processing step 95 .... load step = 950
>> processing step 96 .... load step = 960
>> processing step 97 .... load step = 970
>> processing step 98 .... load step = 980
>> processing step 99 .... load step = 990
>> processing step 100 .... load step = 1000

```

```
>>> Computing eta factors ...
```

```
*****
```

E T A F A C T O R S

```
*****
```

```
>>> STRUCTURE : set_aw5_pin
```

```
>>> Load Displacement Curve:
```

Step	Load Step	J Int	CTOD	Load	CMOD	LLD
1	10	0.0933	0.0060	294.6920	0.0127	0.0100
2	20	0.3734	0.0071	589.2360	0.0254	0.0200
3	30	0.8411	0.0082	883.4840	0.0382	0.0300
4	40	1.4980	0.0093	1177.2780	0.0510	0.0400
5	50	2.3440	0.0107	1470.4520	0.0640	0.0500
6	60	3.3820	0.0122	1762.8400	0.0770	0.0600
7	70	4.6090	0.0138	2054.2600	0.0901	0.0700
8	80	6.0170	0.0158	2344.5000	0.1034	0.0800
9	90	7.6050	0.0180	2633.3601	0.1168	0.0900
10	100	9.3700	0.0204	2920.6001	0.1304	0.1000
11	110	11.3100	0.0232	3206.0000	0.1443	0.1100
12	120	13.4400	0.0262	3489.2600	0.1584	0.1200

13	130	15.7500	0.0295	3770.0601	0.1727	0.1300
14	140	18.2500	0.0331	4048.0801	0.1874	0.1400
15	150	20.9600	0.0371	4322.9399	0.2024	0.1500
16	160	23.8900	0.0413	4594.2402	0.2177	0.1600
17	170	27.0500	0.0459	4861.4800	0.2335	0.1700
18	180	30.4500	0.0508	5124.1602	0.2498	0.1800
19	190	34.1100	0.0561	5381.7598	0.2666	0.1900
20	200	38.0400	0.0618	5633.7002	0.2839	0.2000
21	210	42.2600	0.0679	5879.3799	0.3019	0.2100
22	220	46.8000	0.0743	6118.2598	0.3205	0.2200
23	230	51.6500	0.0813	6349.8198	0.3399	0.2300
24	240	56.8500	0.0886	6573.6001	0.3600	0.2400
25	250	62.4100	0.0964	6788.8198	0.3810	0.2500
26	260	68.4300	0.1048	6993.8999	0.4031	0.2600
27	270	74.8900	0.1137	7189.2202	0.4262	0.2700
28	280	81.8100	0.1231	7374.5801	0.4504	0.2800
29	290	89.3100	0.1332	7548.2402	0.4761	0.2900
30	300	97.3000	0.1438	7712.4800	0.5028	0.3000
31	310	105.8000	0.1548	7866.7402	0.5306	0.3100
32	320	114.8000	0.1665	8011.1001	0.5595	0.3200
33	330	124.4000	0.1787	8145.1602	0.5898	0.3300
34	340	134.6000	0.1914	8269.4199	0.6213	0.3400
35	350	145.3000	0.2046	8384.6396	0.6538	0.3500
36	360	156.6000	0.2183	8491.1201	0.6875	0.3600
37	370	168.4000	0.2324	8590.5801	0.7221	0.3700
38	380	180.7000	0.2469	8684.4404	0.7575	0.3800
39	390	193.3000	0.2617	8772.7598	0.7935	0.3900
40	400	206.4000	0.2769	8854.8799	0.8304	0.4000
41	410	219.9000	0.2924	8931.5996	0.8680	0.4100
42	420	233.6000	0.3081	9004.0996	0.9061	0.4200
43	430	247.7000	0.3241	9072.9199	0.9447	0.4300
44	440	262.1000	0.3403	9138.4404	0.9837	0.4400
45	450	276.9000	0.3568	9200.7197	1.0232	0.4500
46	460	292.0000	0.3735	9259.8799	1.0631	0.4600
47	470	307.4000	0.3904	9316.0195	1.1035	0.4700
48	480	323.1000	0.4075	9369.2402	1.1443	0.4800
49	490	338.9000	0.4248	9420.0195	1.1854	0.4900
50	500	355.0000	0.4422	9468.8799	1.2268	0.5000
51	510	371.3000	0.4597	9516.0801	1.2685	0.5100
52	520	387.7000	0.4774	9561.6602	1.3104	0.5200
53	530	404.2000	0.4952	9605.6602	1.3525	0.5300
54	540	420.8000	0.5131	9648.2002	1.3949	0.5400
55	550	437.5000	0.5310	9689.3398	1.4374	0.5500
56	560	454.4000	0.5491	9729.2197	1.4801	0.5600
57	570	471.3000	0.5673	9767.7998	1.5229	0.5700
58	580	488.4000	0.5856	9805.0801	1.5659	0.5800
59	590	505.6000	0.6040	9841.2197	1.6090	0.5900
60	600	522.9000	0.6224	9876.2998	1.6523	0.6000
61	610	540.3000	0.6409	9910.4004	1.6957	0.6100
62	620	557.8000	0.6595	9943.5400	1.7392	0.6200
63	630	575.3000	0.6781	9975.7197	1.7828	0.6300
64	640	593.0000	0.6968	10007.0400	1.8265	0.6400
65	650	610.8000	0.7155	10037.5996	1.8703	0.6500
66	660	628.6000	0.7344	10067.4404	1.9143	0.6600
67	670	646.5000	0.7533	10096.5400	1.9583	0.6700
68	680	664.5000	0.7723	10124.9600	2.0024	0.6800
69	690	682.6000	0.7914	10152.7402	2.0466	0.6900
70	700	700.8000	0.8105	10179.9805	2.0909	0.7000
71	710	719.1000	0.8297	10206.6797	2.1353	0.7100
72	720	737.5000	0.8490	10232.8799	2.1798	0.7200
73	730	756.0000	0.8683	10258.5996	2.2243	0.7300
74	740	774.5000	0.8877	10283.7998	2.2689	0.7400
75	750	793.2000	0.9071	10308.5195	2.3135	0.7500
76	760	811.9000	0.9266	10332.7803	2.3583	0.7600

77	770	830.7000	0.9461	10356.5801	2.4030	0.7700
78	780	849.5000	0.9657	10379.9600	2.4479	0.7800
79	790	868.4000	0.9853	10402.9199	2.4927	0.7900
80	800	887.4000	1.0050	10425.4805	2.5377	0.8000
81	810	906.4000	1.0247	10447.6602	2.5827	0.8100
82	820	925.5000	1.0445	10469.4805	2.6277	0.8200
83	830	944.6000	1.0643	10490.9404	2.6728	0.8300
84	840	963.9000	1.0842	10512.0801	2.7180	0.8400
85	850	983.1000	1.1042	10532.9004	2.7631	0.8500
86	860	1002.0000	1.1242	10553.3604	2.8084	0.8600
87	870	1022.0000	1.1442	10573.4805	2.8537	0.8700
88	880	1041.0000	1.1643	10593.2598	2.8990	0.8800
89	890	1061.0000	1.1845	10612.7197	2.9443	0.8900
90	900	1080.0000	1.2047	10631.8799	2.9897	0.9000
91	910	1100.0000	1.2249	10650.7402	3.0351	0.9100
92	920	1119.0000	1.2452	10669.3203	3.0806	0.9200
93	930	1139.0000	1.2655	10687.6396	3.1261	0.9300
94	940	1159.0000	1.2859	10705.7002	3.1716	0.9400
95	950	1178.0000	1.3064	10723.5000	3.2171	0.9500
96	960	1198.0000	1.3268	10741.0801	3.2627	0.9600
97	970	1218.0000	1.3474	10758.4404	3.3083	0.9700
98	980	1238.0000	1.3679	10775.5596	3.3540	0.9800
99	990	1258.0000	1.3885	10792.5000	3.3996	0.9900
100	1000	1278.0000	1.4092	10809.2197	3.4453	1.0000

>>> CMOD Based Eta Factors:

Step	J Int	CTOD	Eta J	Eta J Avg	Eta CTOD	Eta CTOD Avg
20	38.0400	0.0618	0.5911	0.5911	1.2200	1.2200
21	42.2600	0.0679	0.6156	0.7176	1.0990	0.5936
22	46.8000	0.0743	0.6407	0.7362	1.0027	0.5862
23	51.6500	0.0813	0.6635	0.7494	0.9255	0.5793
24	56.8500	0.0886	0.6857	0.7632	0.8628	0.5727
25	62.4100	0.0964	0.7061	0.7758	0.8110	0.5664
26	68.4300	0.1048	0.7267	0.7892	0.7671	0.5600
27	74.8900	0.1137	0.7462	0.8025	0.7304	0.5539
28	81.8100	0.1231	0.7643	0.8150	0.6995	0.5482
29	89.3100	0.1332	0.7822	0.8276	0.6726	0.5425
30	97.3000	0.1438	0.7988	0.8398	0.6500	0.5372
31	105.8000	0.1548	0.8143	0.8513	0.6309	0.5323
32	114.8000	0.1665	0.8281	0.8620	0.6145	0.5278
33	124.4000	0.1787	0.8414	0.8721	0.6001	0.5235
34	134.6000	0.1914	0.8541	0.8818	0.5874	0.5195
35	145.3000	0.2046	0.8656	0.8910	0.5764	0.5157
36	156.6000	0.2183	0.8768	0.8998	0.5667	0.5122
37	168.4000	0.2324	0.8871	0.9082	0.5581	0.5089
38	180.7000	0.2469	0.8972	0.9163	0.5505	0.5058
39	193.3000	0.2617	0.9054	0.9236	0.5438	0.5029
40	206.4000	0.2769	0.9135	0.9305	0.5377	0.5002
41	219.9000	0.2924	0.9208	0.9369	0.5323	0.4977
42	233.6000	0.3081	0.9267	0.9425	0.5273	0.4954
43	247.7000	0.3241	0.9327	0.9478	0.5229	0.4932
44	262.1000	0.3403	0.9383	0.9527	0.5189	0.4911
45	276.9000	0.3568	0.9440	0.9574	0.5152	0.4892
46	292.0000	0.3735	0.9493	0.9619	0.5119	0.4874
47	307.4000	0.3904	0.9542	0.9662	0.5088	0.4857
48	323.1000	0.4075	0.9589	0.9702	0.5059	0.4841
49	338.9000	0.4248	0.9627	0.9740	0.5033	0.4826
50	355.0000	0.4422	0.9666	0.9775	0.5008	0.4812
51	371.3000	0.4597	0.9702	0.9808	0.4984	0.4798

52	387.7000	0.4774	0.9733	0.9839	0.4962	0.4785
53	404.2000	0.4952	0.9760	0.9867	0.4941	0.4772
54	420.8000	0.5131	0.9782	0.9893	0.4921	0.4760
55	437.5000	0.5310	0.9802	0.9915	0.4902	0.4748
56	454.4000	0.5491	0.9822	0.9936	0.4884	0.4737
57	471.3000	0.5673	0.9838	0.9955	0.4868	0.4726
58	488.4000	0.5856	0.9854	0.9972	0.4852	0.4715
59	505.6000	0.6040	0.9868	0.9987	0.4837	0.4705
60	522.9000	0.6224	0.9882	1.0001	0.4822	0.4696
61	540.3000	0.6409	0.9894	1.0014	0.4808	0.4686
62	557.8000	0.6595	0.9906	1.0025	0.4795	0.4677
63	575.3000	0.6781	0.9914	1.0036	0.4782	0.4668
64	593.0000	0.6968	0.9925	1.0045	0.4770	0.4660
65	610.8000	0.7155	0.9935	1.0054	0.4758	0.4651
66	628.6000	0.7344	0.9942	1.0062	0.4747	0.4643
67	646.5000	0.7533	0.9949	1.0069	0.4736	0.4635
68	664.5000	0.7723	0.9957	1.0076	0.4726	0.4627
69	682.6000	0.7914	0.9964	1.0082	0.4716	0.4620
70	700.8000	0.8105	0.9970	1.0087	0.4707	0.4613
71	719.1000	0.8297	0.9978	1.0093	0.4698	0.4606
72	737.5000	0.8490	0.9984	1.0098	0.4689	0.4599
73	756.0000	0.8683	0.9991	1.0102	0.4681	0.4593
74	774.5000	0.8877	0.9997	1.0107	0.4673	0.4587
75	793.2000	0.9071	1.0004	1.0111	0.4665	0.4581
76	811.9000	0.9266	1.0010	1.0116	0.4658	0.4575
77	830.7000	0.9461	1.0016	1.0120	0.4651	0.4569
78	849.5000	0.9657	1.0021	1.0123	0.4644	0.4563
79	868.4000	0.9853	1.0027	1.0127	0.4637	0.4558
80	887.4000	1.0050	1.0032	1.0131	0.4630	0.4553
81	906.4000	1.0247	1.0037	1.0134	0.4624	0.4548
82	925.5000	1.0445	1.0041	1.0137	0.4618	0.4543
83	944.6000	1.0643	1.0045	1.0140	0.4612	0.4538
84	963.9000	1.0842	1.0051	1.0143	0.4607	0.4533
85	983.1000	1.1042	1.0054	1.0146	0.4601	0.4529
86	1002.0000	1.1242	1.0053	1.0148	0.4596	0.4524
87	1022.0000	1.1442	1.0064	1.0151	0.4591	0.4520
88	1041.0000	1.1643	1.0063	1.0154	0.4586	0.4516
89	1061.0000	1.1845	1.0072	1.0156	0.4581	0.4512
90	1080.0000	1.2047	1.0070	1.0159	0.4577	0.4508
91	1100.0000	1.2249	1.0078	1.0161	0.4572	0.4504
92	1119.0000	1.2452	1.0075	1.0163	0.4568	0.4500
93	1139.0000	1.2655	1.0081	1.0165	0.4564	0.4497
94	1159.0000	1.2859	1.0087	1.0167	0.4560	0.4493
95	1178.0000	1.3064	1.0083	1.0169	0.4556	0.4490
96	1198.0000	1.3268	1.0087	1.0170	0.4552	0.4487
97	1218.0000	1.3474	1.0091	1.0172	0.4549	0.4484
98	1238.0000	1.3679	1.0095	1.0174	0.4545	0.4481
99	1258.0000	1.3885	1.0098	1.0175	0.4542	0.4478
100	1278.0000	1.4092	1.0101	1.0177	0.4539	0.4475

>>>>> FRACTUS2D: normal termination.

Referências

- [1] Koppenhoefer, K., Gullerud, A., Ruggieri, C., Dodds, R. and Healy, B., “WARP3D: Dynamic Non-linear Analysis of Solids Using a Preconditioned Conjugate Gradient Software Architecture”, *Structural Research Series (SRS) 596*, UILU-ENG-94-2017, University of Illinois at Urbana-Champaign, 1994.
- [2] MSC Patran Reference Manual, <http://mscsoftware.com>.
- [3] Hutchinson, J.W., “Fundamentals of the Phenomenological Theory of Nonlinear Fracture Mechanics,” *Journal of Applied Mechanics*, Vol. 50, pp. 1042–1051, 1983.
- [4] Anderson, T. L., *Fracture Mechanics: Fundamentals and Applications - 3rd Edition*, CRC Press, New York, 2005.
- [5] Toyoda, M., Minami, F., Matsuo, T., Hagiwara, Y and Inoue, T., “Effect of Work Hardening Properties of High Strength Steels on Cleavage/Ductile Fracture Resistance”, Prep. National Meeting of the Japan Welding Society, Vol. 49, pp. 112–113, 1991 (in Japanese).
- [6] Wiesner, C. S. and Goldthorpe, M. R., “The Effect of Temperature and Specimen Geometry on the Parameters of the Local Approach to Cleavage Fracture” in *International Conference on Local Approach to Fracture (MECAMAT 96)*, Fontainebleau, France, 1996, pp. C6-295–304.
- [7] R. H. Dodds, Jr., C. Ruggieri and K. Koppenhoefer, *3-D Constraint Effects on Models for Transferability of Cleavage Fracture Toughness in Fatigue and Fracture Mechanics, 28th Volume, ASTM STP 1321* (Underwood and MacDonald, Ed.), American Society for Testing and Materials, pp. 179-196, 1997.
- [8] Betegon, C., and Hancock, J.W., “Two-Parameter Characterization of Elastic-Plastic Crack Tip Fields,” *Journal of Applied Mechanics*, Vol. 58, pp. 104–113, 1991.
- [9] Du, Z.-Z., and Hancock, J.W., “The Effect of Non-Singular Stresses on Crack-Tip Constraint,” *Journal of Mechanics and Physics of Solids*, Vol. 39, pp. 555–567, 1991.
- [10] Parks, D.M., “Advances in Characterization of Elastic-Plastic Crack-Tip Fields,” in *Topics in Fracture and Fatigue*, A. S. Argon, Ed., Springer Verlag, pp. 59–98, 1992.
- [11] O’Dowd, N.P., and Shih, C.F., “Family of Crack-Tip Fields Characterized by a Triaxiality Parameter: Part I – Structure of Fields,” *Journal of the Mechanics and Physics of Solids*, Vol. 39., No. 8, pp. 989–1015, 1991.
- [12] O’Dowd, N.P., and Shih, C.F., “Family of Crack-Tip Fields Characterized by a Triaxiality Parameter: Part II – Fracture Applications,” *Journal of the Mechanics and Physics of Solids*, Vol. 40, pp. 939–963, 1992.
- [13] Yang, S., Chao, Y. J. and Sutton, M. A., “Higher Order Asymptotic Crack Tip Fields in a Power-Law hardening Materials,” *Engineering Fracture Mechanics*, Vol. 45, pp. 1–20, 1993.
- [14] Chao, Y. J. and X. K. Zhu, “ $J-A_2$ Characterization of Crack-Tip Fields: Extent of $J-A_2$ Dominance and Size Requirements,” *International Journal of Fracture*, Vol. 89, pp. 285–307, 1998.
- [15] Kim, Y., X. K. Zhu and Chao, Y. J., “Quantification of Constraint on Elastic-Plastic 3D Crack Front by the $J-A_2$ Three-Term Solution,” *Engineering Fracture Mechanics*, Vol. 68, pp. 895–914, 2001.
- [16] Williams, M.L., “On the Stress Distribution at the Base of a Stationary Crack”, *Journal of Applied Mechanics*, Vol. 24, pp. 109–114, 1957.
- [17] Dodds, R. H., Shih, C. F., and Anderson, T. L. “Continuum and Micro-Mechanics Treatment of Constraint in Fracture,” *International Journal of Fracture*, Vol. 64, pp. 101–133, 1993.
- [18] Nevalainen, M. and Dodds, R. H., “Numerical Investigation of 3-D Constraint Effects on Brittle Fracture in SE(B) and C(T) Specimens,” *International Journal of Fracture*, Vol. 74, pp. 131–161, 1995.
- [19] Cravero, S. and Ruggieri, C., “Correlation of Fracture Behavior in High Pressure Pipelines with Axial Flaws Using Constraint Designed Test Specimens - Part I: Plane-Strain Analyses,” *Engineering Fracture Mechanics*, Vol. 72, pp. 1344-1360, 2005.

- [20] E. Trovato and C. Ruggieri, “Constraint Effects on Brittle Fracture Under Small Scale Yielding Conditions – Part I: Reference Fields for Elastic-Plastic Materials”, *Sixth Panamerican Congress of Applied Mechanics – (PACAM IV)*, Vol. 7, pp. 943-946, Rio de Janeiro, 1999.
- [21] E. Trovato and C. Ruggieri, “Micromechanics Characterization of Constraint and Ductile Tearing Effects in Small Scale Yielding Fracture”, *International Journal of Solids and Structures*, Vol. 38, pp. 2171-2187, 2001.
- [22] Dowling, N. E., *Mechanical Behavior of Materials*, 2nd Edition, Prentice Hall, NJ, 1999.
- [23] Rice, J. R., “A Path Independent Integral and the Approximate Analysis of Strain Concentrations by Notches and Cracks”, *Journal of Applied Mechanics*, Vol. 35, pp. 379-386, 1968.
- [24] Varias, A. G. and Shih, C. F., “Quase-Static Crack Advance Under a Range of Constraints – Steady State Fields Based on a Characteristic Length”, *Journal of the Mechanics and Physics of Solids*, Vol. 41, pp. 835-861, 1993.
- [25] Ruggieri, C., “WSTRESS Release 3.0: Numerical Computation of Probabilistic Fracture Parameters for 3-D Cracked Solids,” EPUSP, University of São Paulo, 2009.
- [26] Dodds, R. H., Tang, M. and Anderson, T. L., “Effects of Prior Tearing on Cleavage Fracture Toughness in the Transition Region” in *Constraint Effects in Fracture, Theory and Application*, ASTM STP 1244, M. Kirk and A. Bakker, Eds., American Society for Testing and Materials, Philadelphia, Pennsylvania, 1993.
- [27] Ruggieri, C., “Modelagem Micromecânica da Fratura Dúctil e Aplicações à Integridade Estrutural,” *Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Professor Livre Docente junto ao Departamento de Engenharia Naval e Oceânica*, EPUSP, 2001.
- [28] Garrison, W. M, Jr. and Moody, N. R., “Ductile Fracture,” *Journal of the Physics and Chemistry of Solids*, Vol. 48, pp. 1035-1074, 1987.
- [29] Van Stone, R., Cox, T., Low, J., and Psioda, J., “Microstructural Aspects of Fracture by Dimpled Rupture,” *International Metals Reviews*, Vol. 30, No. 4, pp. 157-179, 1985.
- [30] Xia, L. and Shih, C. F., “Ductile Crack Growth – I. A Numerical Study Using Computational Cells with Microstructurally-Based Length Scales”, *Journal of the Mechanics and Physics of Solids*, Vol. 43, pp. 233-259, 1995.
- [31] Xia, L. and Shih, C. F., “Ductile Crack Growth – II. Void Nucleation and Geometry Effects on Macroscopic Fracture Behavior”. to appear in *Journal of the Mechanics and Physics of Solids*.
- [32] Xia, L., Shih, C. F. and Hutchinson, J. W., “A Computational Approach to Ductile Crack Growth Under Large Scale Yielding Conditions”, *Journal of the Mechanics and Physics of Solids*, Vol. 43, pp. 398-413, 1995.
- [33] Faleskog, J., and Shih, C.F., “Micromechanics of Coalescence - I. Synergistic Effects of Elasticity, Plastic Yielding and Multi-Size-Scale Voids,” to appear in *Journal of the Mechanics and Physics of Solids*, 1997. >>>>>>>>
- [34] Faleskog, J., and Shih, C.F., “Micromechanics of Coalescence - I. Synergistic Effects of Elasticity, Plastic Yielding and Multi-Size-Scale Voids,” to appear in *Journal of the Mechanics and Physics of Solids*, 1997. >>>>>>>>
- [35] Ruggieri, C. and Dodds, R. H., “Numerical Modeling of Ductile Crack Growth in 3-D Using Computational Cell Elements,” *International Journal of Fracture*, Vol. 82, pp. 67-95, 1996.
- [36] Gurson, A. L., “Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I - Yield Criteria and Flow Rules for Porous Ductile Media,” *Journal of Engineering Materials and Technology*, Vol. 99, pp. 2-15, 1977.
- [37] Tvergaard, V., “Material Failure by Void Growth to Coalescence,” *Advances in Applied Mechanics*, Vol. 27, pp. 83-151, 1990.
- [38] Hippert, E., “Investigação Experimental do Comportamento Dúctil de Aços API-X70 e Aplicação de Curvas de Resistência $J-\Delta\alpha$ para Previsão de Colapso em Dutos”. *Tese de Doutorado*, Escola Politécnica, Universidade de São Paulo, 167 p., 2004.
- [39] Kanninen, M. F and Popelar, C. H., “Advanced Fracture Mechanis”, Oxford University Press, New york, 1985.

- [40] Hutchinson, J. W. and Paris, P. C., “Stability Analysis of J -Controlled Crack Growth”, *Elastic-Plastic Fracture*, ASTM STP 668, American Society for Testing and Materials, Philadelphia, pp. 37-64, 1979.
- [41] Sumpter, J. D. G. and Turner, C. E., “Method for Laboratory Determination of J_c ,” *Cracks and Fracture*, ASTM STP 601, American Society for Testing and Materials, pp 3-18, 1976.
- [42] Rice, J. R., Paris, P. C. and Merkle, J. G., “Some Further Results of J -Integral Analysis and Estimates,” *Progress in Flaws Growth and Fracture Toughness Testing*, ASTM STP 536, American Society for Testing and Materials, pp 231-245, 1973.
- [43] Cravero, S. and Ruggieri, C., “Estimation Procedure of J -Resistance Curves for SE(T) Fracture Specimens Using Unloading Compliance,” *Engineering Fracture Mechanics*, Vol. 74, pp. 2735-2757, 2007.
- [44] Ruggieri, C., “Further Results in J and CTOD Estimation Procedures for SE(T) Fracture Specimens - Part I: Homogeneous Materials,” *Engineering Fracture Mechanics*, (No Prelo), 2011.
- [45] Kirk, M. T. and Dodds, R. H., “ J and CTOD Estimation Equations for Shallow Cracks in Single Edge Notch Bend Specimens,” *Journal of Testing and Evaluation*, Vol. 21, pp. 228-238, 1993.
- [46] Shih, C. F., “Relationship Between the J -integral and the Crack Opening Displacement for Stationary and Extending Cracks,” *Journal of the Mechanics and Physics of Solids*, Vol. 29, pp. 305-326, 1981.
- [47] Heath, M. T., *Scientific Computing: An Introductory Survey*, 2nd Edition, McGraw-Hill, 2002.
- [48] British Standard, “Fracture Mechanics Toughness Tests”, BS 7448, 1991.
- [49] Donato, G. H. B. and Ruggieri, C., “Estimation Procedure for J and CTOD Fracture Parameters Using Three-Point Bend Specimens”. *6th International Pipeline Conference - IPC 2006*, Calgary, Canadá, 2006.
- [50] Gullerud, A. S., Dodds, R. H., Hampton, R. W. and Dawicke, D. S. “Three-Dimensional Modeling of Ductile Crack Growth in Thin Sheet Metals: Computational Aspects and Validation” *Engineering Fracture Mechanics*, Vo. 63, pp. 347-374, 1999.
- [51] Gullerud, A. S., Gao, X., Dodds, R. H. and Haj-Ali, R. “Simulation of Ductile Crack Growth Using Computational Cells: Numerical Aspects.” *Engineering Fracture Mechanics*, Vol. 66, pp. 65-92, 2000.
- [52] Ruggieri, C. and Dotta, F., “Numerical Modeling of Ductile Crack Extension in Pipelines with Longitudinal Flaws and Applications to Predict Burst Pressure”, *Engineering Structures*, Vol. 33, pp. 1423-1438, 2011.
- [53] American Society for Testing and Materials, “Standard Test Method for Measurement of Fracture Toughness”, ASTM 1820-2008, Philadelphia, 2008.
- [54] American Society for Testing and Materials, “Standard Test Method for Crack-Tip Opening Displacement (CTOD) Fracture Toughness Measurement”, ASTM 1290-2008, Philadelphia, 2008.
- [55] Saxena, A., “Nonlinear Fracture Mechanics for Engineers”, CRC Press, Boca Raton, 1998.
- [56] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, 1992.
- [57] Tada, H, Paris, P. C. and Irwin, G. R., “The Stress Analysis of Cracks Handbook”, 3rd Edition, American Society of Mechanical Engineers, NY, 2000.
- [58] International Organization for Standardization, “Metallic Materials – Unified Method of Test for the Determination of Quasistatic Fracture Toughness”, ISO 12135, 2002.
- [59] Cravero, S., “Desenvolvimento de Procedimentos para Avaliação de Curvas J - R em Espécimes à Fratura SE(T) Utilizando o Método de Flexibilidade”. *Tese de Doutorado*, Escola Politécnica, Universidade de São Paulo, 162 p, 2007.
- [60] Dodds, R. H., Tang, M. and Anderson, T. L., “Effects of Prior Tearing on Cleavage Fracture Toughness in the Transition Region” in *Constraint Effects in Fracture, Theory and Application*, ASTM STP 1244, M. Kirk and A. Bakker, Eds., American Society for Testing and Materials, Philadelphia, Pennsylvania, 1993.

Anexo A

Geometria de Corpos-de-Prova Convencionais de Fratura

As Figs. A1-A3 apresentam a geometria e dimensões principais dos corpos-de-prova de fratura implementados no programa *FRACTUS2D* incluindo as configurações compacta C(T), flexão 3 pontos SE(B) e corpos-de-prova de tração com entalhe lateral SE(T) carregados por pino e garra. A configuração geométrica dos corpos-de-prova C(T) e SE(B) segue as geometrias especificadas pelos procedimentos de teste ASTM1820 [53] e ASTM1290 [54]. Os corpos-de-prova SE(T) ainda não são padronizados e, desta forma, a geometria adotada neste manual é consistente com os procedimentos ASTM referenciados anteriormente e, também, com trabalhos recentes publicados na literatura internacional. Nas figuras apresentadas a seguir, a define o tamanho ou profundidade da trinca, W representa a largura do corpo-de-prova, S é a distância entre apoios para corpos-de-prova SE(B) e H é a distância entre os pontos de aplicação da carga (pino ou garra) para corpos-de-prova SE(T). Detalhes específicos sobre a geometria e configuração destes corpos-de-prova podem ser vistos na ASTM1820 [53] e ASTM1290 [54].

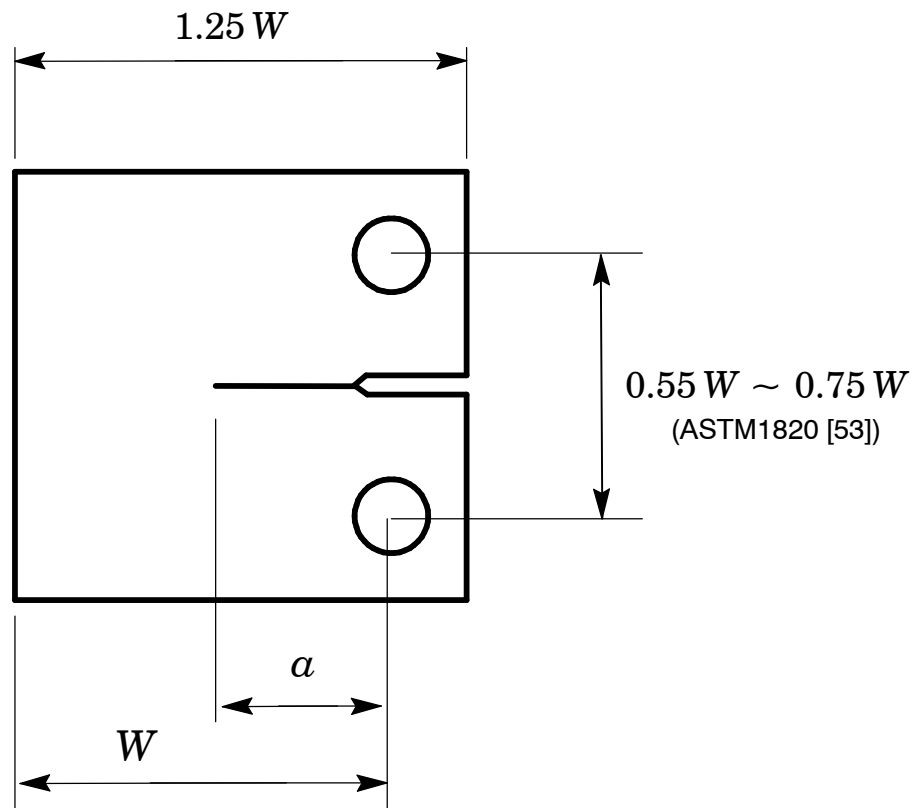


Figura A1 Geometria do corpo-de-prova compacto C(T).

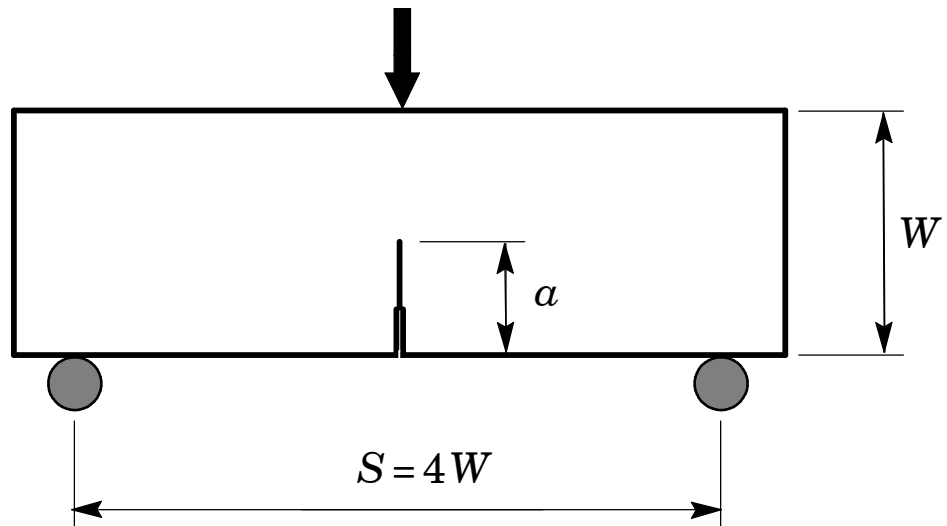


Figura A2 Geometria do corpo-de-prova flexão 3 pontos SE(B).

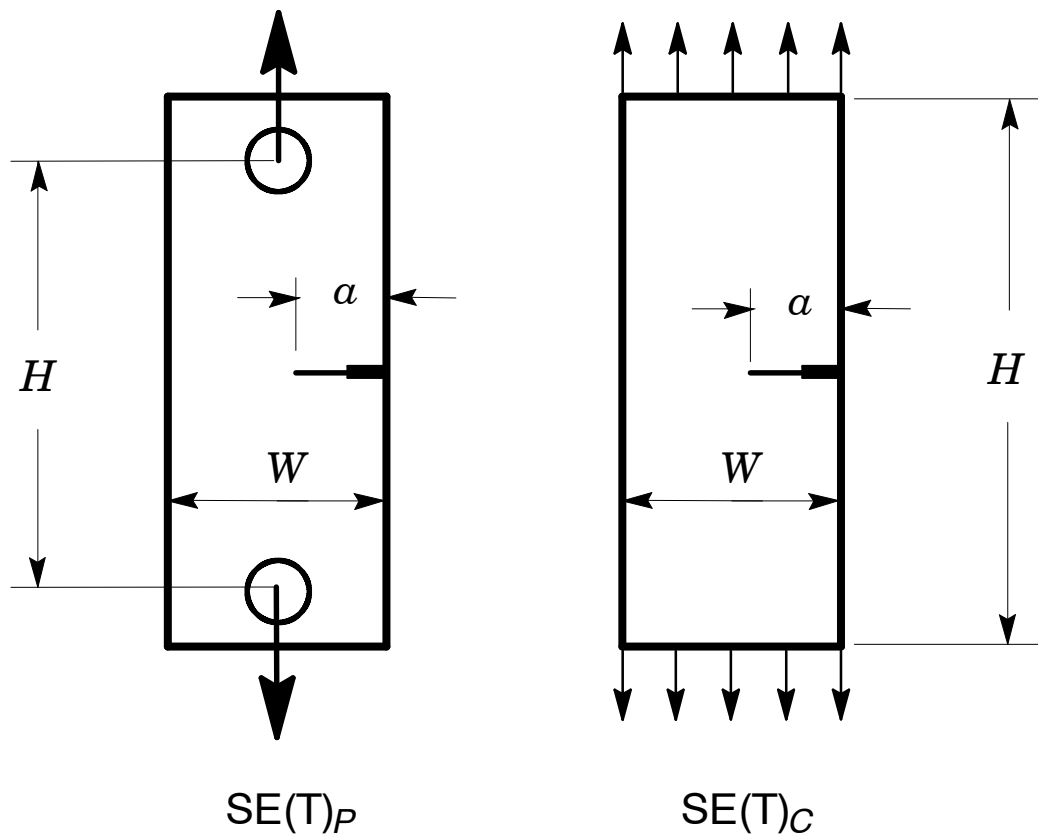


Figura A3 Geometria do corpo-de-prova de tração com entalhe lateral SE(T) carregado por pino e garra.

Anexo B

Funções de Flexibilidade Elástica

B.1 Flexibilidade Elástica em Corpos-de-Prova de Fratura

A relação linear (elástica) entre o carregamento aplicado e o deslocamento correspondente, definido pelo CMOD (*crack mouth opening displacement*) ou LLD (*load line displacement*), em um corpo-de-prova de fratura ilustrado na Fig. B1 define a sua flexibilidade elástica, $C_{a/W}$, na forma

$$C_{a/W}^{\text{CMOD}} = \frac{V}{P} \quad (\text{B1})$$

e

$$C_{a/W}^{\text{LLD}} = \frac{\Delta}{P} \quad (\text{B2})$$

onde P representa a força e V e Δ denotam os valores correspondentes de CMOD e LLD respectivamente. A flexibilidade elástica é simplesmente o inverso da rigidez elástica do corpo-de-prova e, conseqüentemente, varia com a relação a/W , onde a é o comprimento ou profundidade da trinca e W é a largura do corpo-de-prova (mantendo-se fixas as dimensões principais do corpo-de-prova como sua espessura, largura e comprimento). O aumento da relação a/W causa um decréscimo da rigidez a um aumento da flexibilidade do corpo-de-prova como indicado na Fig. B1.

FRACTUS2D utiliza a flexibilidade elástica, C , essencialmente em dois procedimentos implementados no programa: *i*) determinação da área elástica sob a curva P vs. CMOD ou P vs. LLD para o cálculo dos fatores η (veja Seção 3) e *ii*) estimativa do crescimento de trinca, Δa , por intermédio da variação da flexibilidade elástica durante a determinação de curvas de resistência J - R (veja Seção 4). As seções seguintes apresentam as funções de flexibilidade elástica para diferentes corpos-de-prova de fratura implementadas em *FRACTUS2D* obtidas por meio de análises numéricas refinadas utilizando o método de elementos finitos.

B.2 Modelos Numéricos e Procedimentos Computacionais

Análises numéricas refinadas em estado plano de deformações (2D) utilizando o método de elementos finitos foram conduzidas para corpos-de-prova compactos C(T), flexão 3 pontos SE(B) e de tração SE(T) carregados por pinos e garras. A geometria dos corpos-de-prova (veja Fig. A1-A3 anteriores) analisados típicas configurações 1-T (espessura $B = 25.4$ mm) com $W = 2B$ e diferentes relações a/W conforme descrito a seguir

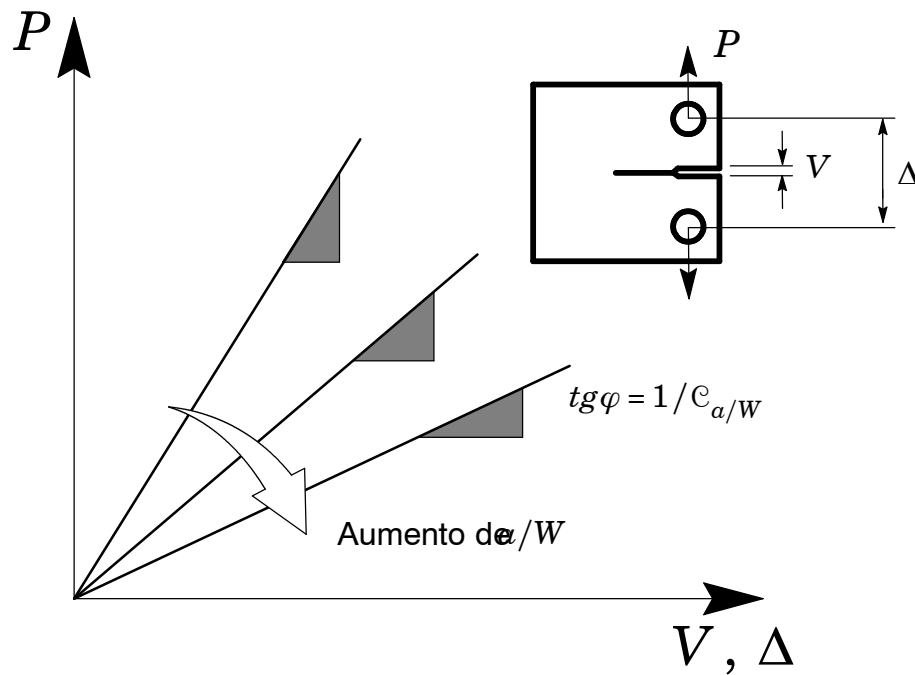


Figura B1 Ilustração esquemática da relação linear entre carga aplicada e deslocamento (CMOD ou LLD) para um corpo-de-prova C(T) com diferentes relações entre a profundidade de trinca (a) e largura (W).

- **Corpos-de-Prova Compactos C(T)**
 $0.45 \leq a/W \leq 0.70$ em incrementos de 0.05

- **Corpos-de-Prova Flexão 3 Pontos SE(B)**
 $0.10 \leq a/W \leq 0.80$ em incrementos de 0.05
 $S/W = 4$

- **Corpos-de-Prova SE(T) Carregados por Pino e Garra**
 $0.10 \leq a/W \leq 0.70$ em incrementos de 0.05
 $H/W = 6$ e 10

A Fig. B2 apresenta o modelo de elementos finitos 2D em estado plano de deformação utilizado para as análises elásticas do corpo-de-prova SE(B) com trinca profunda ($a/W = 0.5$). Os modelos numéricos empregados para as análises elásticas das demais configurações geométricas são similares. Condições de simetria permitem a modelagem de somente metade dos cor-

pos-de-prova com condições de contorno (vínculos) apropriadas impostas nos plano de simetria dos modelos. A utilização de uma malha refinada em todos os modelos, constituída por anéis focais centrados na ponta da trinca, e um pequeno raio ($\rho_0=0.0025$ mm) simulando o arredondamento inicial da trinca (*crack tip blunting*) permite modelar mais adequadamente a região da trinca. Os modelos de elementos finitos utilizados nas análises elásticas tipicamente possuem aproximadamente 1500 elementos e 3000 nós. O código de elementos finitos WARP3D [1] é utilizado para a solução das análises numéricas e determinação da evolução linear entre a carga e deslocamento em todos os modelos. A relação constitutiva elástica utilizada adota módulo de elasticidade, $E=206$ GPa e coeficiente de Poisson, $\nu=0.3$.

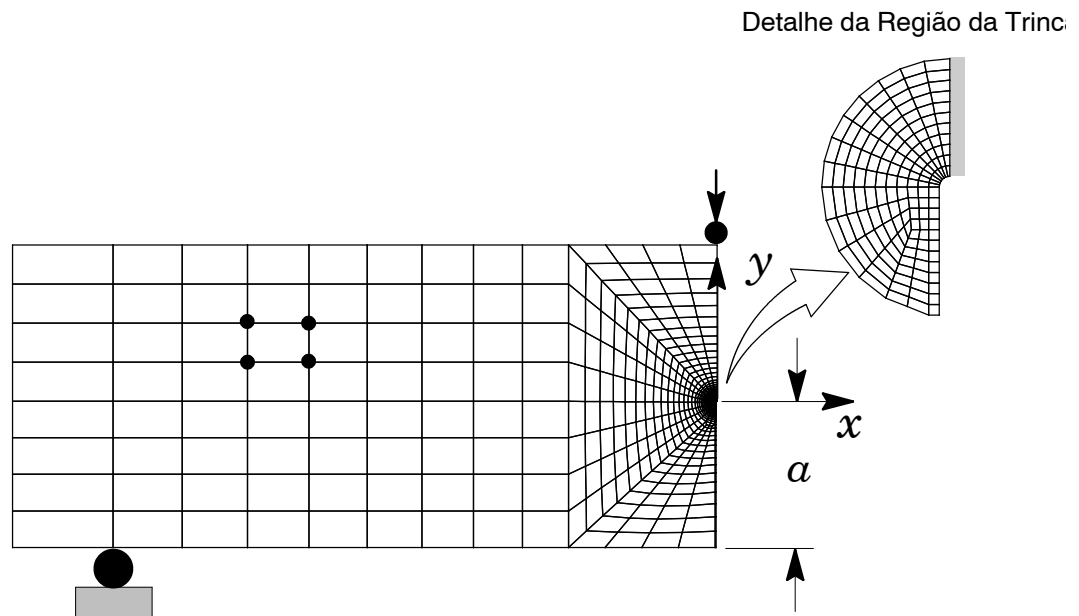


Figura B2 Modelo de elementos finitos para o corpo-de-prova 1-T SE(B) com $a/W=0.5$.

A Fig. B3 mostra a relação elástica entre a carga aplicada (P) e o deslocamento correspondente caracterizado pelo CMOD. A determinação da flexibilidade, C , para cada relação a/W permite construir uma relação funcional entre a flexibilidade normalizada pelas propriedades elásticas e dimensões do corpo-de-prova com a relação a/W para cada configuração geométrica como descrito a seguir.

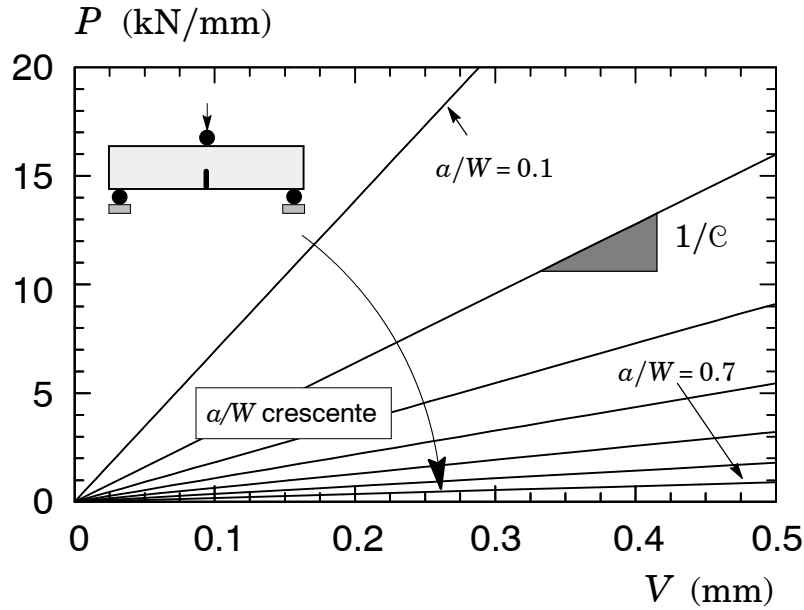


Figura B3 Relação linear entre carga e deslocamento (CMOD) para o corpo-de-prova 1-T SE(B) com diferentes relações a/W .

B.3 Corpos-de-Prova Compactos C(T)

Para corpos-de-prova compactos C(T) (veja Fig. A1), a flexibilidade normalizada, μ , é expressa por [53,55][†]

$$\mu = \left[\sqrt{B_e E C} + 1 \right]^{-1} \quad (\text{B3})$$

onde E é o módulo de elasticidade do material, C expressa a flexibilidade elástica do corpo-de-prova e B_e é a espessura efetiva do corpo-de-prova definida por

$$B_e = B - \frac{(B - B_N)^2}{B} \quad (\text{B4})$$

na qual B é a espessura total do corpo-de-prova e B_N denota a sua espessura líquida expressa por

[†]) Há certa divergência na literatura em relação ao valor do módulo de elasticidade utilizado para a normalização da flexibilidade elástica. Saxena [55] adota $E = E'$ (i.e., $E/(1 - \nu^2)$) correspondendo a estado plano de deformações), enquanto versões recentes da ASTM 1820 [53] adotam o valor de E (i.e., estado plano de tensões). FRACTUS2D adota o valor de E em estado plano de tensões para definir a flexibilidade normalizada, μ .

$$B_N = B - 2d_{SG} \quad (\text{B5})$$

onde d_{SG} é a espessura do entalhe lateral (*side groove*). Quando o corpo-de-prova não possuir entalhe lateral (*i.e.*, $d_{SG} = 0$), a espessura efetiva é equivalente á espessura total, $B = B_e$.

As Figs. B4(a-b) apresentam a variação da flexibilidade normalizada, μ , com a relação a/W obtida por meio das análises descritas anteriormente e implementada em *FRACTUS2D*. Para efeitos comparativos, as Figs. B4(a-b) também incluem as funções de flexibilidade normalizadas obtida em Tada et al [57] e no procedimento ASTM 1820 [53]. Observa-se boa concordância entre os valores de μ implementados em *FRACTUS2D* e outras referências, particularmente para o caso de CMOD e tamanhos de trinca correspondentes a $a/W \geq 0.5$.

Para facilitar a manipulação das funções de flexibilidade obtida por intermédio das análises numéricas descritas anteriormente, um ajuste polinomial utilizando o método de mínimos quadrados [56] fornece as seguintes expressões para μ_{LLD} e μ_{CMOD} :

$$\mu_{LLD} = \zeta_0 + \zeta_1(a/W) + \zeta_2(a/W)^2 + \zeta_3(a/W)^3 + \zeta_4(a/W)^4 + \zeta_5(a/W)^5 \quad (\text{B6})$$

onde

$$\zeta_0 = +0.7105$$

$$\zeta_1 = -4.5404$$

$$\zeta_2 = +15.6964$$

$$\zeta_3 = -28.1402$$

$$\zeta_4 = +24.9198$$

$$\zeta_5 = -8.7684$$

e

$$\mu_{CMOD} = \xi_0 + \xi_1(a/W) + \xi_2(a/W)^2 + \xi_3(a/W)^3 + \xi_4(a/W)^4 + \xi_5(a/W)^5 \quad (\text{B7})$$

onde

$$\xi_0 = +0.8024$$

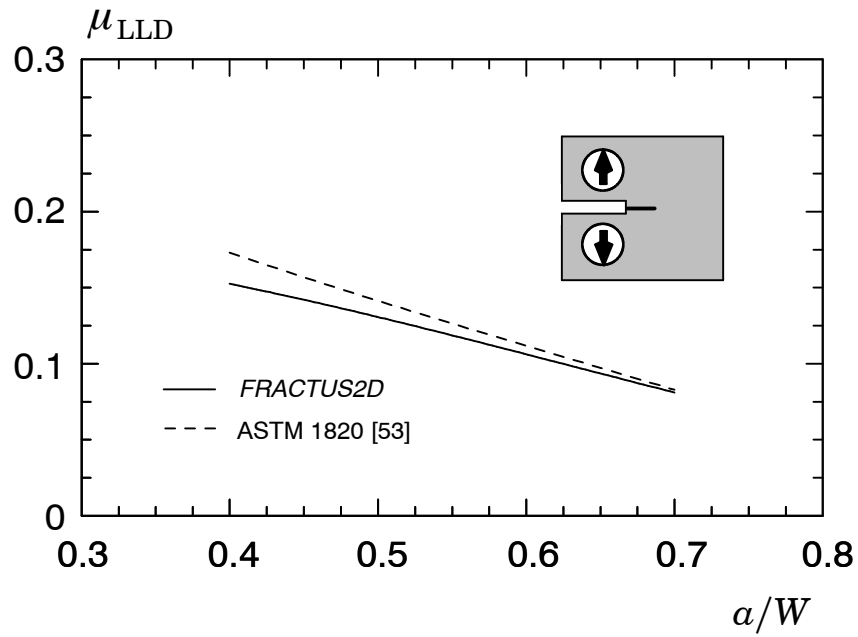
$$\xi_1 = -5.3430$$

$$\xi_2 = +18.2762$$

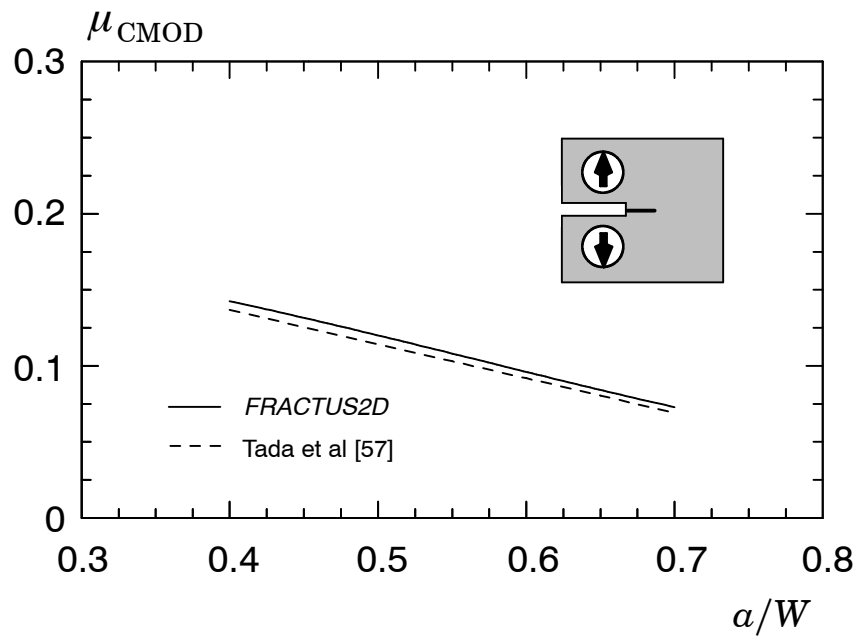
$$\xi_3 = -32.4300$$

$$\xi_4 = +28.5928$$

$$\xi_5 = -10.0353$$



(a)



(b)

Figura B4 Variação da flexibilidade normalizada, μ , correspondente ao LLD e CMOD para o corpo-de-prova $C(T)$ com diferentes relações α/W .

B.4 Corpos-de-Prova Flexão 3 Pontos SE(B)

Para corpos-de-prova flexão 3 pontos SE(B), a flexibilidade normalizada, μ , é expressa por [53,55]

$$\mu = \left[\sqrt{\frac{B_e W E C}{S/4}} + 1 \right]^{-1} \quad (\text{B8})$$

onde E é o módulo de elasticidade do material, C expressa a flexibilidade elástica do corpo-de-prova, W e S representam a sua largura e distância entre apoios (veja Fig. A2) e B_e é a espessura efetiva do corpo-de-prova definida por

$$B_e = B - \frac{(B - B_N)^2}{B} \quad (\text{B9})$$

na qual B é a espessura total do corpo-de-prova e B_N denota a sua espessura líquida expressa por

$$B_N = B - 2d_{SG} \quad (\text{B10})$$

onde d_{SG} é a espessura do entalhe lateral (*side groove*). Quando o corpo-de-prova não possuir entalhe lateral (*i.e.*, $d_{SG} = 0$), a espessura efetiva é equivalente á espessura total, $B = B_e$.

A Fig. B5 apresenta a variação da flexibilidade normalizada, μ , com a relação a/W obtida por meio das análises descritas anteriormente e implementada em *FRACTUS2D*. Para efeitos comparativos, a Fig. B5 também inclui a função de flexibilidade normalizada obtida em Tada et al [57] e fornecida nos procedimentos ASTM 1820 [53] e ISO 12135 [58]. Observa-se excelente concordância entre os valores de μ implementados em *FRACTUS2D* e outras referências, particularmente para tamanhos de trinca correspondentes a $a/W \geq 0.2$.

Para facilitar a manipulação das funções de flexibilidade obtida por intermédio das análises numéricas descritas anteriormente, um ajuste polinomial utilizando o método de mínimos quadrados [56] fornece a seguinte expressão para μ_{CMOD}

$$\mu_{\text{CMOD}} = \xi_0 + \xi_1(a/W) + \xi_2(a/W)^2 + \xi_3(a/W)^3 + \xi_4(a/W)^4 + \xi_5(a/W)^5 \quad (\text{B11})$$

onde

$$\xi_0 = +0.5142$$

†) Há certa divergência na literatura em relação ao valor do módulo de elasticidade utilizado para a normalização da flexibilidade elástica. Saxena [55] adota $E = E'$ (*i.e.*, $E/(1 - \nu^2)$ correspondendo a estado plano de deformações), enquanto versões recentes da ASTM 1820 [53] adotam o valor de E (*i.e.*, estado plano de tensões). *FRACTUS2D* adota o valor de E em estado plano de tensões para definir a flexibilidade normalizada, μ .

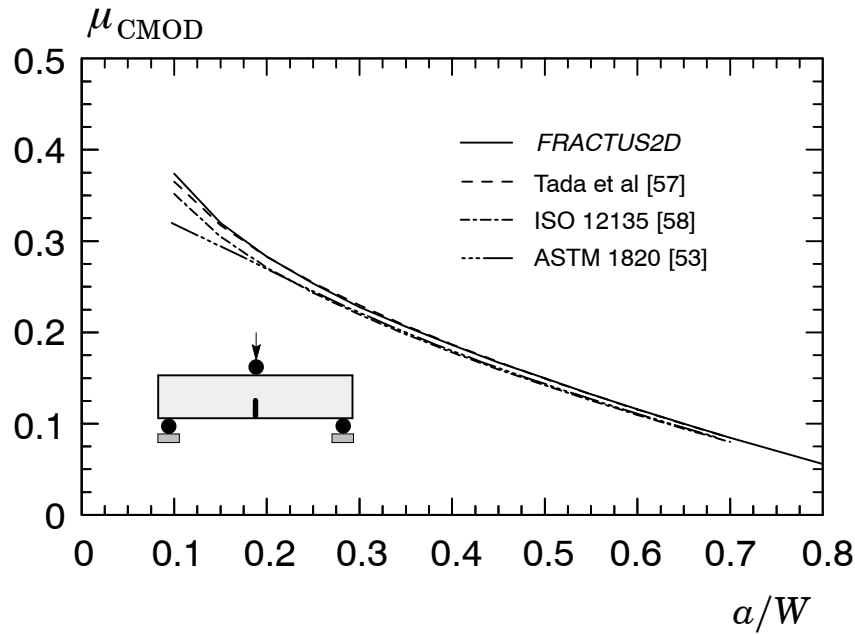


Figura B5 Variação da flexibilidade normalizada, μ , correspondente ao CMOD para o corpo-de-prova SE(B) com diferentes relações a/W .

$$\xi_1 = -1.9379$$

$$\xi_2 = +5.6002$$

$$\xi_3 = -10.2410$$

$$\xi_4 = +9.5040$$

$$\xi_5 = -3.4847$$

B.5 Corpos-de-Prova SE(T)

Para corpos-de-prova SE(T) carregados por pinos (*pin-loaded ends*) ou garras (*clamped ends*) (veja Fig. A3), a flexibilidade normalizada, μ , é expressa por [55]

$$\mu = \left[\sqrt{B_e E C} + 1 \right]^{-1} \quad (\text{B12})$$

[†]) Há certa divergência na literatura em relação ao valor do módulo de elasticidade utilizado para a normalização da flexibilidade elástica. Saxena [55] adota $E = E'$ (i.e., $E/(1-\nu^2)$ correspondendo a estado plano de deformações), enquanto versões recentes da ASTM 1820 [53] adotam o valor de E (i.e., estado plano de tensões). FRACTUS2D adota o valor de E em estado plano de tensões para definir a flexibilidade normalizada, μ .

onde E é o módulo de elasticidade do material, C expressa a flexibilidade elástica do corpo-de-prova e B_e é a espessura *efetiva* do corpo-de-prova definida por

$$B_e = B - \frac{(B - B_N)^2}{B} \quad (\text{B13})$$

na qual B é a espessura *total* do corpo-de-prova e B_N denota a sua espessura *líquida* expressa por

$$B_N = B - 2d_{SG} \quad (\text{B14})$$

onde d_{SG} é a espessura do entalhe lateral (*side groove*). Quando o corpo-de-prova não possuir entalhe lateral (*i.e.*, $d_{SG} = 0$), a espessura efetiva é equivalente á espessura total, $B = B_e$.

A Fig. B6(a-b) apresenta a variação da flexibilidade normalizada, μ , com a relação a/W obtida por meio das análises descritas anteriormente e implementada em *FRACTUS2D*. Para efeitos comparativos, a Fig. B6 também inclui a função de flexibilidade normalizada obtida em Tada et al. [57] e Cravero [59] para corpos-de-prova SE(T) carregados por pinos (*pin-loaded ends*) e aquela fornecida por Cravero [59] para corpos-de-prova SE(T) carregados por garras (*clamped ends*) com $H/W = 10$. Observa-se excelente concordância entre os valores de μ implementados em *FRACTUS2D* e demais referências para todos os tamanhos de trinca caracterizados pela relação a/W . É importante observar que a flexibilidade elástica não depende da relação H/W para corpos-de-prova SE(T) carregados por pinos; conseqüentemente, os resultados apresentados na Fig. B6(a) representam todos os valores de μ para diversas relações H/W para esta configuração.

Para facilitar a manipulação das funções de flexibilidade obtida por intermédio das análises numéricas descritas anteriormente, um ajuste polinomial utilizando o método de mínimos quadrados [56] fornece a seguinte expressão para μ_{CMOD}

$$\mu_{\text{CMOD}} = \xi_0 + \xi_1(a/W) + \xi_2(a/W)^2 + \xi_3(a/W)^3 + \xi_4(a/W)^4 + \xi_5(a/W)^5 \quad (\text{B15})$$

onde os valores dos coeficientes polinomiais, ξ_k , dependem da configuração geométrica e condições de carregamento como descrito a seguir.

- **Corpos-de-Prova SE(T) Carregados por Pino (Todos H/W)**

$$\xi_0 = +0.7334$$

$$\xi_1 = -2.0350$$

$$\xi_2 = +5.0122$$

$$\xi_3 = -9.3635$$

$$\xi_4 = +8.8877$$

$$\xi_5 = -3.2771$$

- **Corpos-de-Prova SE(T) Carregados por Garra (H/W = 6)**

$$\xi_0 = + 0.7342$$

$$\xi_1 = - 2.0519$$

$$\xi_2 = + 5.3531$$

$$\xi_3 = - 9.7498$$

$$\xi_4 = + 9.2488$$

$$\xi_5 = - 3.4388$$

- **Corpos-de-Prova SE(T) Carregados por Garra (H/W = 10)**

$$\xi_0 = + 0.7296$$

$$\xi_1 = - 1.9469$$

$$\xi_2 = + 4.5249$$

$$\xi_3 = - 7.6064$$

$$\xi_4 = + 6.6378$$

$$\xi_5 = - 2.2371$$

- **Corpos-de-Prova SE(T) Carregados por Garra (H/W = 20)**

$$\xi_0 = + 0.7325$$

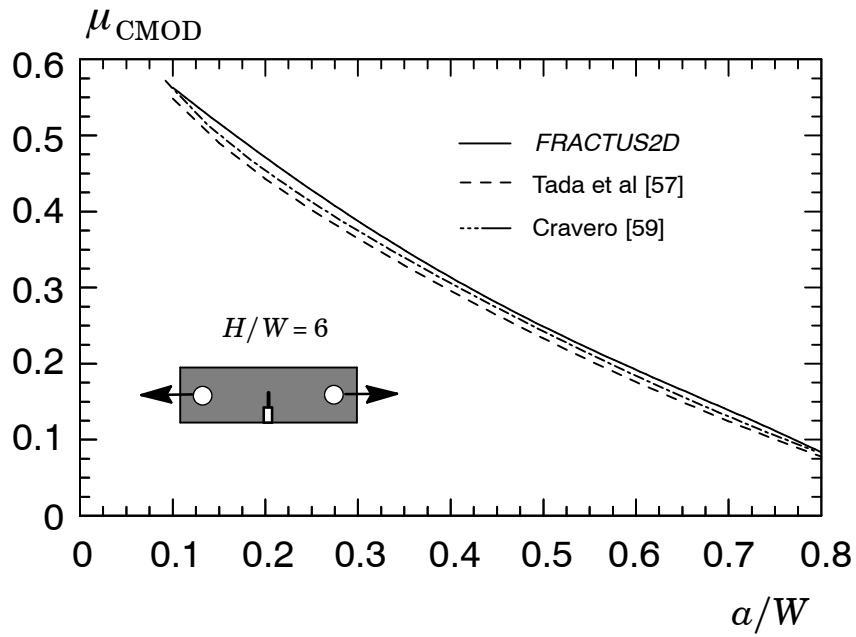
$$\xi_1 = - 2.0119$$

$$\xi_2 = + 4.9311$$

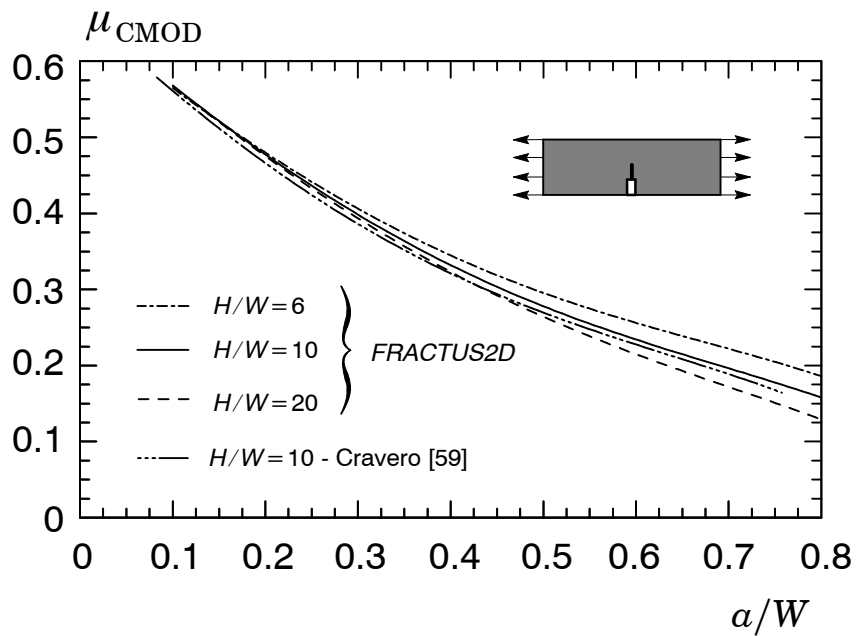
$$\xi_3 = - 8.9689$$

$$\xi_4 = + 8.3689$$

$$\xi_5 = - 3.0035$$



(a)



(b)

Figura B6 Variação da flexibilidade normalizada, μ , correspondente ao CMOD para o corpo-de-prova SE(T) com diferentes relações a/W . (a) Carregamento por pino ; (b) Carregamento por garra.