

An embedded interface finite element method for fluid-structure-fracture interaction

Sudhakar Yogaraj

Bericht Nr. 28 (2015)
Lehrstuhl für Numerische Mechanik
Professor Dr.-Ing. Wolfgang A. Wall
Technische Universität München

Berichte können bezogen werden über:



Lehrstuhl für Numerische Mechanik
Technische Universität München
Boltzmannstrasse 15
D-85747 Garching bei München
<http://www.lnm.mw.tum.de>

Alle Rechte, insbesondere das der Übersetzung in andere Sprachen, vorbehalten. Ohne Genehmigung des Autors ist es nicht gestattet, dieses Buch ganz oder teilweise auf photomechanischem, elektronischem oder sonstigem Wege zu kommerziellen Zwecken zu vervielfältigen.

All rights reserved. In particular the right to translate the text of this thesis into another language is reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any other information storage and retrieval system, without written permission of the author.

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Numerische Mechanik

An embedded interface finite element method for fluid-structure-fracture interaction

Sudhakar Yogaraj

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. dr. ir. Daniel J. Rixen

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Wolfgang A. Wall
2. Prof. Nicolas Moës, Ph.D.
École Centrale de Nantes, France

Die Dissertation wurde am 2. April 2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 25. Juni 2015 angenommen.

Abstract

The vast majority of the computational methodologies developed to model the fluid-structure interaction (FSI) assume that when the structure interacts with the fluid, it deforms completely within the elastic limit. They preclude not only any plastic deformation but also the failure by fracture. In a variety of natural phenomena and engineering applications, the fluid-induced loads on the structure are large enough to enable the initiation and propagation of cracks, which may ultimately result in catastrophic failure of the structure. This occurrence is termed as fluid-structure-fracture interaction (FSFI), and the present work is focused on devising an embedded interface finite element method (EIM) to model such coupled multiphysics phenomena.

The contributions of the current thesis are three-fold. First, on the structural part, a mesh refitting approach is devised to simulate the fracture of nonlinear hyperelastic materials. Then, on the fluid part, accurate and robust numerical integration methods are developed to enhance the robustness of EIMs. Finally, on the coupling part, the interaction of fluid flow with cracking structures is modeled using a strongly coupled partitioned approach.

The first step of extending FSI methods to handle FSFI is to equip the structural analysis with a fracture mechanics solver. In this work, a mesh refitting approach is developed to simulate brittle fracture of an isotropic homogeneous hyperelastic material. A two step mesh-modification algorithm is utilized to deal with the topology change introduced in the structural domain due to the crack propagation. First, the nodes are repositioned in such a way that the crack propagates along an existing edge in the new mesh. Then, if necessary, the element connectivity is locally modified in order to preserve the mesh quality. The excellent performance of this method is demonstrated by simulating crack propagation examples of varying complexity and comparing the results with those available in the literature.

One of the crucial steps that dictates the accuracy and robustness of EIMs, while solving the Navier-Stokes equations, is the accuracy of weak form integration, which gained least attention in the literature. The embedded interface cuts a few elements of the mesh, and on these cut elements the integration needs to be performed over the arbitrary polyhedral shaped cells resulting from the interface crossing. The most widely used volume decomposition based methods lack robustness i.e., they fail at certain critical cut configurations and the simulation crashes. To address this issue, two new numerical integration strategies, namely the generalized moment fitting method and the direct divergence method are developed in this work. Both methods are designed to work in situations, where the integrand is not explicitly prespecified. This is essential for their applications in FEM. Moreover, they can handle convex as well as concave shaped polyhedra. Numerical examples demonstrate that the direct divergence method is the most accurate and posses superior robustness characteristics when compared to all other available methods.

Having dealt with fracture and the robustness of EIM, in the final step, a coupling approach is developed to model the fluid-structure-fracture interaction. Since the objective is to use the existing methodologies for fracture and fluid field, and because this work considers highly flexible structures, a strongly coupled partitioned approach is utilized. The method involves Dirichlet-Neumann partitioning and the coupling is achieved using fixed-point iterative methods employing Aitken's Δ^2 relaxation as convergence accelerator. Numerical examples involving complete fracture of the structure due to fluid loads are solved using this method.

In short, this thesis presents a first step towards extending the numerical modeling of FSI to the next level by introducing two essential features: possibility of crack initiation and propagation within the structure due to fluid loads, and the mutual interaction of crack surfaces with the fluid entering the crack. Numerical example involving a structure breaking completely into two parts, of which one is carried away by the flow is presented in this thesis.



Acknowledgement

First and foremost, I express my heartfelt thanks to my advisor Prof. Dr.-Ing. Wolfgang A. Wall for giving me an opportunity to work in the ATCoMe project, for his constant support and guidance, for several discussions and encouragement that kept my motivation alive during the entire PhD time, and for the freedom he has provided to pursue our own ideas. It is a pleasure and a honour to work in his research group. Moreover I thank Prof. Daniel Rixen for chairing the examination committee and Prof. Nicolas Moës, École Centrale de Nantes, for taking time to read the thesis and constructing interesting discussions during the defense.

I gratefully acknowledge the financial support provided by the 7th framework programme of the European Union, in the form of Marie Curie early stage researcher fellowship. I also thank the International Graduate School of Science and Engineering (IGSSE) of TUM for providing additional support.

This work would not have been possible without the constant support from my colleagues working in XFEM: Ulrich Küttler for developing geometrical cut libraries, Benedikt Schott for patiently explaining many mathematical ideas, Shadan Shahmiri and Andy Wirtz for many helpful discussions and for their friendship. I express my sincere gratitude to all of them. Moreover, I extend my thanks to Christoph Ager, Raffaella Kruse and Magnus Winter for their recent improvements in the methods developed in this work.

I am indebted to Karl-Robert Wichmann, Georg Hammerl, and Martin Kronbichler for all their helps related to C++ implementation and also for spending time in maintaining our code & taking care of computer related issues. I extend my thanks to Tobias Wiesner and Andreas Rauch for their helps with the cluster facility.

I am grateful to Philipp Farah, Andy Wirtz, Andreas Rauch, Alexander Seitz, Benedikt Schott, and particularly to Raffaella Kruse for spending their time in proof-reading the thesis.

I owe my sincere gratitude to Ms. Renata Nagl and Ms. Annetta Reinecke for patiently and cheerfully handling all the paper works for more than four years. Special thanks to Mirella Coroneo, Mahmoud Ismail, and Kei Müller for their support & help in understanding and adapting to the German system.

I would like to thank Thomas Klöppel for explaining many finite element concepts in the earlier days. Special thanks to Philipp Farah, Rui Fang and Anh-Tu Vuong for keeping our office a very dynamic place. Furthermore, I wish you thank all the colleagues at the Institute for Computational Mechanics for their friendly association and fruitful collaboration.

Lastly but most importantly I thank my mother Mrs. Rajammal for everything.

Sudhakar Yogaraj



Contents

1	Introduction	1
1.1	Background and motivation	2
1.2	Problem definition	3
1.3	Overview of related research	4
1.3.1	Hydraulic fracture	4
1.3.2	Detonation and fragmentation studies	5
1.3.3	Inapplicability of existing methods to current problem class	5
1.4	Why embedded interface finite element methods?	6
1.5	Objectives and accomplishments	8
1.6	Structure of the thesis	9
2	Structure & Fracture	11
2.1	Microscopic fracture processes	11
2.1.1	Brittle fracture	12
2.1.2	Ductile fracture	12
2.1.3	Ductile–brittle transition	13
2.2	Macroscopic description of fracture process	13
2.3	Governing equations	17
2.3.1	Kinematics	17
2.3.2	Stress measure	18
2.3.3	Initial boundary value problem	18
2.3.4	Constitutive equations	19
2.4	Overview of the existing computational methods	20
2.5	The mesh refitting approach	22
2.5.1	Solve the governing equations	23
2.5.2	Perform computational crack propagation procedure	26
2.6	Numerical examples	33
2.6.1	Crack tip blunting	34
2.6.2	J -integral computation	34
2.6.3	Single edge notched beam with a hole	36
2.6.4	Single edge cracked plate under mixed-mode loading	37
2.6.5	Crack growth from a fillet	38
2.6.6	Crack deflection due to inclusion	39
2.6.7	Crack in a drilled plate	39
2.6.8	Four point beam with two notches	40
2.6.9	Nonlinear elastic plate with a hole	43
2.7	Closure	44

3	Fluid: Robust numerical integration in embedded interface methods	47
3.1	Navier-Stokes equations: strong form	47
3.2	Introduction to embedded interface methods	49
3.3	Navier-Stokes equations: weak form	50
3.3.1	Time discretization	50
3.3.2	Spatial discretization	51
3.4	A note on time integration	55
3.5	Numerical integration of weak forms	55
3.5.1	Review of existing integration methods	57
3.5.2	Generalized moment fitting method	62
3.5.3	Direct divergence method	70
3.6	Numerical examples	82
3.6.1	Fluid statics example: Significance of accurate weak form integration . . .	82
3.6.2	Integration over arbitrary polyhedra: Robustness	85
3.6.3	A simple test case : Accuracy	89
3.6.4	Beltrami flow : Order of convergence	91
3.6.5	Stationary interface simulations : Efficiency	92
3.6.6	Moving interface simulations: Efficiency	95
3.6.7	Comparison with other methods based on the divergence theorem	98
3.7	Closure	100
4	Fluid-structure-fracture interaction	103
4.1	Partitioned vs. Monolithic coupling	103
4.2	Problem definition	104
4.2.1	Structure	104
4.2.2	Fluid	105
4.2.3	Interface	106
4.3	The strongly coupled partitioned approach	106
4.4	Numerical examples	111
4.4.1	Failure of a structure supporting a static fluid	111
4.4.2	Breaking fluid channel due to external loads	112
4.4.3	A filament fixed at both ends subjected to fluid loading	114
4.4.4	Fracture of a bending structure	116
4.5	Unsolved issues	119
4.6	Closure	120
5	Summary and Outlook	121
5.1	Summary	121
5.2	Outlook	122
A	A simple crack propagation approach for linear elastic fracture mechanics	125
B	Implementation aspects of numerical integration techniques	129
B.1	Computation of normal of a polyhedron	129
B.2	Computation of normals and equation of plane for a simple polygon	129

Nomenclature

Domains

Ω_0^s	Structural domain in the initial configuration
Ω^f	Fluid domain in the current configuration
Ω_K^f	Physical fluid domain
Γ^s, Γ^f	Boundary of the structural and fluid domain
Γ_D^s, Γ_D^f	Dirichlet portion of the boundary
Γ_N^s, Γ_N^f	Neumann portion of the boundary
Γ^i	Embedded interface; FSFI interface
Γ^c	Crack surface
γ_j	Contour for J -integral in current configuration
Γ_j	Contour for J -integral in reference configuration
S	Domain used for J -integral in reference configuration

Mesh and other geometric quantities

\mathcal{T}	Finite element mesh
K	A finite element
$\Omega_{\mathcal{T}}$	Set of elements that are completely or partially in Ω^f
G_h	Set of all cut elements
V	Volume cell
$N_i^{(e)}$	Shape function at node i associated with element (e)
\mathbf{n}	Normal vector

Field quantities and related parameters

Fluid	
\mathbf{u}^f	Velocity
p^f	Pressure
μ^f	Dynamic viscosity
ρ^f	Density
\mathbf{b}^f	Body force per unit mass
$\boldsymbol{\sigma}^f$	Cauchy stress tensor
$\boldsymbol{\epsilon}^f$	Strain rate tensor
\mathbf{a}	Acceleration
C_L	Lift coefficient
C_D	Drag coefficient
τ	Non-dimensional time

Structure

\mathbf{d}^s	Displacement
$\dot{\mathbf{d}}^s$	Velocity
$\ddot{\mathbf{d}}^s$	Acceleration
ρ^s	Density
\mathbf{b}^s	Body force per unit mass
E^s	Young's modulus
ν^s	Poisson's ratio
$\boldsymbol{\sigma}$	Cauchy stress
\mathbf{F}	Deformation gradient
\mathbf{R}	Rigid body rotation
\mathbf{U}	Right stretch tensor
\mathbf{C}	Cauchy-Green tensor
\mathbf{E}	Green-Lagrange strain tensor
\mathbf{P}	First Piola-Kirchhoff stress tensor
\mathbf{S}	Second Piola-Kirchhoff stress tensor
\mathbf{t}	Traction vector
Ψ	Strain energy function
J	Jacobian
λ	Stretch ratio

Fracture

σ_f	Fracture stress
γ^s	Surface energy of the material
γ^p	Plastic work per unit crack surface area
\mathcal{G}	Energy release rate
\mathcal{G}_c	Critical energy release rate
Π	Potential energy
r	Distance of a point from crack tip
A	Surface area of crack
K	Stress intensity factor
K_c	Critical stress intensity factor
W	Strain energy density
J	Component of J -integral
\mathbf{J}, J_c	Vector J -integral and its critical value
\mathbf{x}	Coordinate in current configuration
\mathbf{X}	Coordinate in reference configuration
θ_p	Crack propagation angle
q	Support function for J -integral

Interface

\mathbf{d}^i	Displacement
\mathbf{u}^i	Velocity
\mathbf{f}^i	Nodal forces
\mathbf{h}_f^{-i}	Fluid traction applied on the structural side
$\bar{\mathbf{u}}_s^i$	Structural velocity applied on the fluid side

Function spaces and weak form

Structure

\mathcal{W}_d	Solution space for structure
\mathcal{V}_d	Test function space for structure
$\delta \mathbf{d}^s$	Test function; virtual displacements
\mathbf{M}^s	Mass matrix
\mathbf{F}_{int}^s	Nonlinear vector of internal forces
\mathbf{F}_{ext}^s	Vector of external forces
\mathbf{r}^s	Residual vector
γ, β	Parameters characterizing Generalized- α method
α_f, α_m	Parameters defining the midpoint of Generalized- α method

Fluid

\mathcal{S}_h^u	Trial function space for velocity
\mathcal{V}_h^u	Test function space for velocity
\mathcal{Q}_h^p	Trial and test function space for pressure
R_{NS}, f_{NS}	Terms arising from the discretization of Navier-Stokes equations
R_{STAB}^{resi}	Residual based stabilization terms
R_{STAB}^{edge}	Edge based stabilization terms
R_{emBC}	Terms associated with weak enforcement Dirichlet conditions on interface
$\mathbf{r}_{M,h}, \mathbf{r}_{C,h}$	Residual of momentum and continuity equation
τ_M, τ_C	Residual based stabilization parameters
C_I	Constant associated with τ_M
α, β	Parameter associated with ghost-penalty stabilization
$\gamma_u, \gamma_v, \gamma_{conv}, \gamma_t$	Stabilization parameters associated with Nitsche's method
h_K	Characteristic element length scale
h_F	Characteristic face length scale
j_{GP}	Ghost-penalty stabilization term
j_{stream}	Edge based stabilization version of SUPG
j_p	Edge based stabilization version of PSPG
γ_p	Edge based stabilization parameter for pressure
θ	Parameter characterizing one-step-theta method

Numerical integration

$\omega(\mathbf{x})$	Weight function
ϕ	Base functions
(\mathbf{x}, w)	Location and weights of quadrature rule
\mathcal{R}	Polyhedron
\mathcal{S}	Facets of a polyhedron
\mathcal{F}	Integrand of weak form
\mathcal{G}	Modified integrand of weak form
(\mathbf{X}, W)	Main Gauss rule
(\mathcal{X}, w)	Internal Gauss rule
$I_{\mathcal{R}}$	Integral over volume cell
e_Q	Error associated with quadrature rule

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ALE	Arbitrary Lagrangian Eulerian
CTOD	Crack tip opening displacement
DCT	Displacement correlation technique
DOF	Degrees of freedom
EIM	Embedded interface method
EPUM	Enriched partition of unity method
FE	Finite element
FEM	Finite element method
FSI	Fluid-structure interaction
FSFI	Fluid-structure-fracture interaction
Hex	Hexahedral
LBB	Ladyzhenskaya-Babuška-Brezzi
LEFM	Linear elastic fracture mechanics
LSIC	Least-squares incompressibility constraint
MCSC	Maximum circumferential stress criterion
MERR	Maximum energy release rate criterion
OST	One step theta
Quad	Quadrilateral
PSPG	Pressure-Stabilizing/Petrov-Galerkin
Re	Reynolds number
SIF	Stress intensity factor
SUPG	Streamline-Upwind/Petrov-Galerkin
Tri	Triangular
XEFM	Extended finite element method

1 Introduction

When a fluid flows past/within a structure, loads exerted by the fluid tend to change the configuration of the structure by inducing deformations and/or displacements. This change in structural configuration, in-turn, influences the dynamics of the fluid flow. The aforementioned intrinsic two-way communication between the structure and the fluid with which it is in contact with, is termed as fluid-structure interaction (FSI).

FSI is ubiquitous both in nature and in man-made engineering structures. Plant aerodynamics, animal locomotion, and flow of air, blood, lymph, and cerebrospinal fluid through the corresponding flexible biological structures are a few natural phenomena that are highly influenced by FSI; representative examples of man-made engineering structures where FSI analysis is inevitable are airplane wing flutter, energy extraction from rotary/oscillatory structures, wind induced vibration of tall buildings, and vibration of compressor and turbine blades of gas turbine engines. In short, in plethora of natural phenomena and in engineering applications, the presence of fluid-structure interaction is a rule rather than an exception.

The fluid induced loads on the structure can in many instances deflect the structure within the elastic limit, and in some cases introduce permanent plastic deformation. In other applications the fluid loads are so high that they subsequently lead to the initiation and propagation of crack, which may result in ultimate failure of the structure. This phenomenon in which the fluid loads result in crack propagation, in addition to deflection of the structure, is termed in this thesis as fluid-structure-fracture interaction (FSFI).

The coupled nonlinear governing partial differential equations, together with the associated complex geometry make it impossible to obtain analytical solutions in practically relevant FSI configurations. Moreover, conducting careful laboratory experiments are extremely expensive and also are limited in their scope. Therefore in order to understand the fundamental physical processes involved in nature and to design better engineering products, computational modeling of FSI in the aforementioned topics is an ideal choice because of its cost-effectiveness and the level of details one can obtain from simulations. This thesis takes the numerical modeling of FSI to the next step by introducing the possibility of fracture within the structure due to the fluid induced loads. In order to achieve this, the mere computation of the structural deflection in FSI analyses should be extended to include catastrophic failure by fracture. To be more precise, the aims here are to

- enable the possibility of crack initiation and propagation within the structure due to fluid loads
- model the mutual interaction of crack surfaces and the fluid entering the crack opening

These are achieved in this work by developing an embedded interface finite element methodology to handle fluid-structure-fracture interaction, which combines the knowledge from three different fields: structural mechanics, fluid dynamics and fracture mechanics.

1.1 Background and motivation

Owing to the fact that a variety of rich physical phenomena are governed by FSI, and due to their strong engineering relevance, FSI has gained great attention in the past decades. Several researchers have studied FSI using experimental methods as well as numerical approaches. Almost all of the existing studies assume that the structure is ideal without any imperfections. In addition, they invoke the assumption that due to the fluid loading, the structure deforms within the elastic limit; any permanent/plastic deformation and failure by fracture are precluded. Studies under these assumptions aim to analyze, in a coupled manner, the elastic deflections of the structure due to fluid flow and the corresponding changes in fluid flow owing to the deflection of the structure.

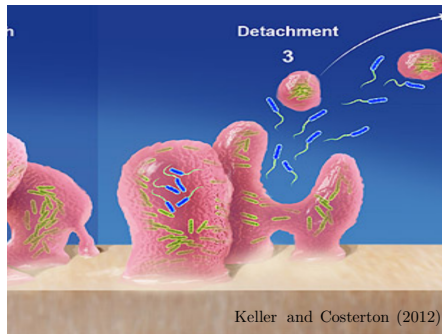
FSI analyses under the assumption that the structure always deflects within the elastic limit are very useful in a variety of circumstances. For instance, consider an example of fluid flow over an airplane wing. Due to the fluid loading the wing may bend, and this bending deflection affects the fluid flow characteristics. The coupled analysis in such cases is focused on confirming the following two: (a) structural stresses, owing to the deflection induced by the fluid loading, is within the design limits, and (b) even after the structural deformation, the fluid flow around the wing generates sufficient lift force to keep the airplane aloft.

The majority of the engineering structures and components contain preexisting flaws: internal voids, micro-cracks, inclusion, and second phase particles. These flaws are introduced either in the course of the manufacturing process, or they develop during the duration of their service period. The destructive influence of these preexisting flaws on the strength of the material is known even from the time of Leonardo da Vinci, who conducted experiments on iron wires, and concluded that the strength of wires varied inversely with its length. Longer wire possesses lower strength because due to larger volume, larger is the possibility of containing flaws than the shorter ones. These flaws elevated the stresses experienced by the structure which can be high enough to induce plastic deformation, enable crack initiation and ultimately lead to failure at lower applied stresses. In other words, in the presence of these flaws, the structure fails at much lower applied stress than the material strength, even if the loading is within the elastic regime. Owing to their considerable importance, it is now widely accepted that the crack propagation must be considered both during the design and the analysis phase of any engineering structure. Though fracture analysis has become a usual practice in a pure structural analysis, the presence of preexisting flaws and the possibility of fracture are completely ignored in a coupled fluid-structure interaction analysis.

A few natural phenomena and potential engineering applications in which FSFI analysis is inevitable are illustrated in figure 1.1. Detachment of biofilms [1–4] and breakage of wave-swept microalgae [5–8] are two important biological examples, wherein the life-time of the organisms involved are dictated by the fracture induced due to the fluid loads acting on them. Moreover, FSFI finds widespread applications in engineering structures. For example, the failure of fluid carrying pipes is one of the major safety threats to nuclear powerplants and natural gas transmission pipelines. Such scenarios in engineering are very important not only for their scientific relevance but also for the fact that these fracture failures can lead to loss of many human lives. Due to this fact, there have already been a few studies that aim to analyze FSFI in pipeline fracture [9–15] and to model warhead detonation [16, 17].

Despite the availability of several FSFI studies focused on nuclear and military purposes, the biological applications have not been addressed before, to the best of our knowledge. Moreover, the computational methods developed to address the aforementioned engineering applications cannot be used in biological problems because of the different physics involved. This issue is addressed clearly in section 1.3. The primary objective of this thesis is to develop an embedded

Applications in nature



Detachment of biofilms

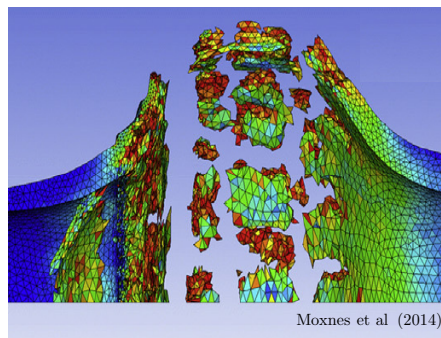


Breakage of wave-swept microalgae

Applications in engineering



Fracture of liquid carrying pipeline



Detonation of warheads

Figure 1.1: A few applications of fluid-structure-fracture interaction in nature and in man made engineering structures

interface finite element method to model FSFI in biological applications. The present work is motivated by our interest in characterizing fluid dynamic behavior of biofilm streamers [18–20]. This work can be considered as the first step towards the modeling of biofilm separation and other biological applications. It is worthwhile to mention that slow crack propagation within the structure is a common characteristic in such applications. This means that the fracture can be modeled as a quasi-static process. Some other physical processes where dynamic effects are important are listed in section 1.3.2.

1.2 Problem definition

The previous section explained how the preexisting flaws reduce the material strength, and it has been said that this study aims at introducing the influence of these flaws into FSI analysis. For brevity, the objective of this thesis is explained further in this section.

Consider a fluid domain Ω^f within which a structural obstacle Ω^s is embedded (figure 1.2a). Let the preexisting imperfection within the structure is denoted as Γ^c ; in figure 1.2a, the structure is shown to encompass a sharp notch. In a pure coupled FSI problem that does not consider the influence of any flaws, both the fluid and the structure exchange information through the common interface $\Gamma^i = (\Gamma^s \cup \Gamma^c)$, and the structure exhibits elastic deformation owing to the fluid loading.

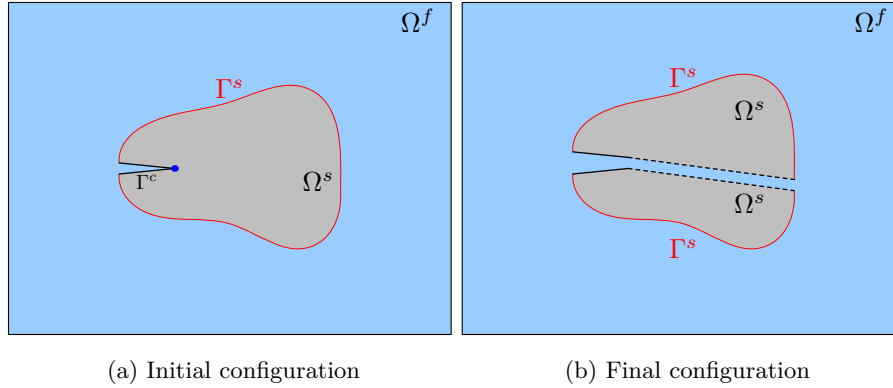


Figure 1.2: A schematic illustration of initial and final configurations of a fluid-structure-fracture interaction problem. (a) Initial configuration showing preexisting flaws in structure. (b) Final configuration with complete fracture. Crack path is shown as dotted lines

In contrast, FSFI analyses give rise to two possible scenarios.

No crack growth: The loading conditions are such that the structure experiences very low stress values, and it undergoes elastic deformation. This case can be modeled using the existing pure FSI solvers.

Fracture: In this case, in addition to elastic deformation, fracture can occur within the structure. Even when the stresses are completely within the elastic regime in other parts of the structure, the presence of the flaws enhances the stresses in their vicinity to a very high value. Peak stresses act at the blue point marked in figure 1.2a. Due to this high value, cracks can start growing from this notch, and the structure may even fracture i.e, completely break into two (figure 1.2b). In this configuration, both the structure and the fluid-structure interface are completely broken into two, and the fluid can flow through the crack. This work aims at simulating both the crack propagation, and the fluid entering the crack, in addition to simulating the usual fluid-structure interaction effects. This work though assumes clearly defined crack initiation points (blue dot in figure 1.2), no prior knowledge on the crack path through the structure is assumed.

1.3 Overview of related research

Research works that consider the influence of fluid-induced forces on the crack propagation within the structure are recalled here. This is mandatory to understand the current developments in this field, and more importantly to clearly identify how our work is different from others. The following text reviews the salient features of the existing methods, and explains why they are not suitable for modeling the biological applications listed in figure 1.1.

1.3.1 Hydraulic fracture

Hydraulic fracture is the process in which the hydraulic loading i.e., pressure induced by the fluid within the fracture is responsible for the initiation and subsequent propagation of fracture. In nature, this phenomenon is observed in propagation of water-containing crevasses through glaciers and the transport of magma through earth's crust in fissures. Such hydraulic fracture

is deliberately created in engineering, and finds applications in mining, petroleum engineering, waste disposal, and remediation of contaminated soils.

Traditionally, in this class of problems, the fluid flow within the crack opening is modeled as flow between two parallel plates [21, 22]. Under such assumptions, the Navier-Stokes equations are simplified into a “local cubic law” in which flow unsteadiness, inertial forces, and velocity components perpendicular to crack surfaces are neglected. It is called cubic law, owing to the fact that the volume flow rate is proportional to the cube of aperture. Recent studies [23–25] have questioned the validity of this simplified flow model, and concluded that inertial forces can significantly influence the internal flow field and flow rate within the crack. In order to include the influence of inertial effects, the complete Navier-Stokes equations must be solved.

1.3.2 Detonation and fragmentation studies

This class of studies addresses the interaction of the structure and the pressure & detonation waves traveling through the fluid. The fluid flow is assumed to be compressible and inviscid, hence Euler equations govern its behavior. The crack propagation within the material is modeled using cohesive elements. Importantly, since the structure is subjected to impact loads due to the traveling waves, dynamic effects must be included in the fracture modeling. Rupture of thin-walled tubes subjected to detonation waves traveling through the fluid contained in the pipe is the widely simulated example [9–11, 13–15]. Plastic and viscoplastic effects of the structure are also included in the modeling. This tube-rupture problem represents a motivation to study the accidents occurring in the cooling system of nuclear power plants, and in underwater implosions. The study of Wang [13] reports also the simulation of two-phase flow in this example. A novel meshless method is recently proposed to model such scenarios [12]. The speciality of this method is its ability to handle both FSI and FSI-crack interaction with minimal modifications.

Another interesting application for this class of problems is to model the pressure waves generated from the explosives, and the subsequent fragmentation of the surrounding structure. Specifically, detonation and fragmentation of rocket warheads [16] and blast environment due to explosives [26] are studied to understand the internal blast damage and fragment dispersion effects.

1.3.3 Inapplicability of existing methods to current problem class

Having reviewed the existing methods in the aforementioned text, the reasons why these methods cannot be used for biological applications presented in figure 1.1 are briefed here.

In hydraulic fracture theory, the fluid flow around the structure and the mutual interaction between fluid and structure are not considered. It is assumed that the structure is completely rigid, and the loads exerted by the fluid that fills the crack opening is the only source of stresses and fracture. This does not fit into our needs since the aim here is at simulating structures whose movement is governed by the fluid flow around them and FSI is important. In our problem class of interest, the crack propagates not only due to fluid loading, but also due to structural stresses induced by overall FSI.

In detonation and fragmentation studies, the fluid viscosity is neglected and the focus is on resolving the compressible waves traveling through the fluid. Moreover, since the objective is on modeling the influence of such waves or impact loading on the structure, the dynamic effects are very important. In this case, the crack propagates through the material at very high velocities and all the physically relevant events take place within micro-seconds. In order to model fluid forces induced material failure in biological applications, the present work focuses on coupling viscous

incompressible fluids with the structure, and to enable quasi-static brittle crack propagation within the structure.

It is clear that neither hydraulic fracture nor detonation and fragmentation studies are suitable for addressing the present problem class of interest, and the development of a new framework to model fluid-structure-fracture interaction is mandatory. This is accomplished in this work by devising an embedded interface finite element method to simulate FSFI, and the following section explains the reasons behind this choice.

1.4 Why embedded interface finite element methods?

The discretization in a mesh-based numerical approach, i.e., the process of converting the governing (usually partial) differential equations into a system of algebraic equations that can be easily handled by a computer, can be achieved using either finite difference, finite element or finite volume methods. Among these, the finite element method has evolved into one of the most powerful methods devised so far. The reasons for this are, among many others, the ability to handle complex geometries, consistent treatment of differential-type boundary conditions, and strong mathematical foundation.

Consider the problem of simulating fluid flow over an arbitrary body. In contrast to meshless methods, either a body-fitted grid or a non-body-fitted grid is used to arrive at an approximate solution of the governing equations in a mesh-based FEM. The present work interchangeably uses the terms “fixed grid method” and “embedded interface method” to denote the class of methods that utilize a non-body-fitted grid framework. In this section, the salient features of both body-fitted and embedded interface methods are described to provide an explanation for why embedded interface methods are opted in this work.

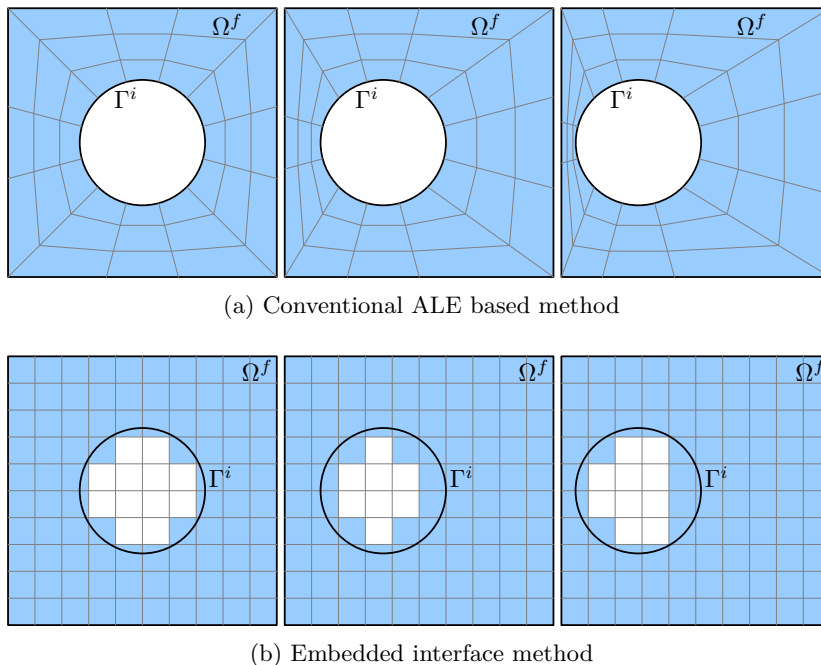


Figure 1.3: Illustration of how ALE based and embedded interface methods handle interface movement. The shaded region represents the domain (Ω^f) over which fluid equations are solved. Γ^i denotes the embedded interface.

In a conventional computational approach, a body-fitted structured or unstructured mesh is generated over the fluid part of the domain (figure 1.3a), and the governing equations are discretized and solved directly over this mesh. The major advantage of the method is the straightforward implementation of boundary conditions on the interface, and the fact that a solver of adequate accuracy can be designed for a quality mesh. However, generating a high quality mesh can be a daunting task, and sometimes even impossible if the geometry of the interface is highly complex. In certain cases, even after resorting to a multiblock grid generation approach, the appearance of poor quality elements is inevitable. The mesh generation process is highly time consuming, involves a lot of human intervention, and moreover the presence of low quality elements adversely affect the stability and convergence characteristics of the solver [27].

The problems experienced by these methods are elevated, when one attempts to perform simulations involving moving or deforming interfaces. Arbitrary Lagrangian Eulerian (ALE) methods are most widely employed to deal with moving interfaces in commercial and research codes. Such methods involve the following two steps:

- An automatic procedure to update the mesh points from one-step to another, in order to accommodate the movement of the interface and to maintain the “body fittedness” of the mesh in the new time step. Since the mesh is generated only on the fluid part of the domain, and the domain occupied by the fluid continuously changes with time, this step ensures that at any instant, the fluid domain is covered with a valid body-fitted mesh (compare the mesh at different interface positions in figure 1.3a).
- Projection of the solution from old mesh to the new one. The procedure should be accurate enough not to introduce large errors which deteriorate the solution.

The mesh movement is usually achieved by solving an auxiliary problem to decide the displacement of each node in the mesh. The major objective in this step is to preserve the quality of the updated mesh. These methods suffer to preserve the mesh quality unless the problem deals only with small interface deformations; the failure of such methods while handling large deformation FSI is very clearly documented in a recent work [28]. While handling large deformation problems, they introduce excessive element shearing and mesh tangling, and these degenerated elements negatively impact the stability and convergence of the solver. In order to address these issues, ALE based methods resort to remeshing strategies to avoid generating highly distorted elements. Remeshing strategies are extremely costly and introduce a lot of complications in the software design.

In short, the aforementioned two steps, especially when the interface exhibits large motions, negatively affect the accuracy, robustness as well as the computational cost of the method.

This work utilizes fixed grid methods in which the entire simulation is performed on a fixed (usually Cartesian) mesh irrespective of the shape or orientation of the interface. Unlike conventional methods, the mesh is generated both on the fluid domain and on the inner portion of the interface; the elements of the mesh that fall completely within the fluid domain, and that are cut by the interface are identified, and the governing equations are discretized and solved only on these elements (figure 1.3b). This greatly simplifies the mesh generation process. Moreover, handling the movement of the interface in fixed grid methods is relatively simple. From mesh modification perspective, no additional procedure is mandatory, as the interface simply moves within the fluid mesh without disturbing the mesh. In contrast to body-fitted methods, fixed grid methods do not require the projection of the whole solution field from one-step to another. Only the nodes that were located within the solid domain in the previous step, and are thrown into the

fluid domain at the current time step due to the movement of the interface need special treatment. Owing to these advantages, fixed grid methods are best suited for simulations involving either a complex or moving interface and also the only practical choice for simulating problems involving topology changes.

The above text highlighted the problems associated with ALE methods while handling “smooth” large displacements. Even in these cases, ALE methods require a few remeshing steps during the simulation. “smooth” deformation implies the instances where the interface does not exhibit abrupt movement or topological changes. The problem class of interest in this thesis, FSI with cracking structures, induces topology change of the fluid domain each time the crack propagates within the structure (figure 1.4). Moreover, even the state of the art ALE methods cannot handle the thin fluid opening introduced due to crack faces moving apart. When ALE methods are used to deal with such problems, remeshing must be performed at each crack propagation step, and hence not a practical approach to be used. Fixed grid methods, as already stated above, are therefore the ideal choice for the present work.

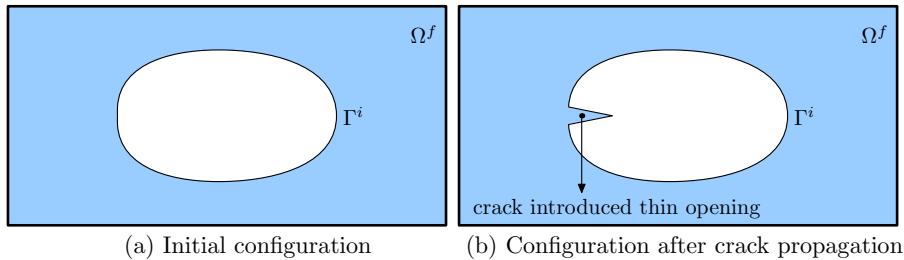


Figure 1.4: Topology change introduced by the crack propagation

All of the aforementioned advantages of fixed grid methods stem from the fact that they do not require a body-fitted grid. This feature, in addition to the advantages, brings several challenges to the solver part that are briefed here. The boundary condition enforcement is not straightforward since FE nodes are not available on the interface. The discretization in the vicinity of the interface must be modified to incorporate the boundary conditions. Since the interface cuts through the fluid mesh in an arbitrary manner, ensuring the robustness of the method is a daunting task. To be specific, convective stabilization and numerical integration of weak forms are challenging in peculiar interface cut situations. One focus of this thesis is to develop accurate weak form integration methods that are addressed in chapter 3. In the same chapter it is briefly pointed out how other challenges are tackled.

1.5 Objectives and accomplishments

The overall purpose of the present thesis is the development of a computational methodology to model fluid-structure interaction in the presence of cracking structures. This is very challenging, and in order to enable systematic developments, the overall purpose is broken into three separate objectives. These objectives and a brief note on how they are accomplished are stated as follows.

1. **Fracture:** The first step is to equip the structural analysis with a fracture mechanics solver. In this work, a mesh refitting approach is developed to simulate single and multiple crack propagation through an isotropic homogeneous hyperelastic material. Each time when the crack propagates, it induces topology change in the structural domain. In order to deal with this, the initial mesh covering the structure must be adjusted continuously. The majority of

the existing methods utilize adaptive remeshing strategies to deal with this issue. However, for practical reasons related to the computational costs associated with FSI simulations, adaptive methods are avoided in this work. Instead, a two step mesh-modification algorithm is proposed; in the first step, the computational nodes are repositioned in such a way that the crack in the new mesh propagates along an existing edge in the mesh, and in the next step, if necessary, the element connectivity is modified to enable arbitrary crack paths through the mesh. The excellent performance of this method is confirmed by simulating crack propagation examples of varying complexity and comparing the results with those available in literature.

- 2. Robust numerical integration in embedded interface methods to handle fluid formulations:** Fixed grid methods bring in several challenges, in addition to their advantages, as pointed out in the previous section. One of the crucial step that dictates the accuracy and robustness of such methods is the accuracy of weak form integration. In contrast to body-fitted methods (figure 1.3a), the interfaces in fixed grid methods are not aligned along the edges of the background mesh (figure 1.3b) i.e., the interface cuts the background mesh's elements in an arbitrary manner. Over these cut elements the weak form must be integrated in the region that is located within the fluid part. This requires accurate numerical integration strategies that work for arbitrary volumes. The most widely used volume decomposition based methods lack robustness i.e., they fail at certain cut configurations and the simulation crashes. To address this issue, two new numerical integration strategies, namely the generalized moment fitting method and the direct divergence method are constructed in this work. Numerical examples demonstrate that the direct divergence method is the most accurate of all the available methods, and possesses superior robustness characteristics.
- 3. Fluid-structure-fracture interaction:** Having dealt with fracture and the robustness of fluid formulations, a coupling approach is developed to model the fluid-structure-fracture interaction. As stated before, this enables the possibility of simulating fracture within the structure owing to the FSI loads. Since the objective is to use the existing methodologies for fracture and fluid field, and because this work considers highly flexible structures, a strongly coupled partitioned approach is utilized. Numerical examples involving complete fracture of the structure due to FSI loads are solved using this method.

All the methods devised to accomplish the aforementioned objectives are implemented in **BACI**, the FEM multiphysics solver developed at our institute. **BACI** is written in C++, and it makes use of the **Trilinos** open-source libraries [29] developed at Sandia National Laboratories. This work makes use of the already existing and well validated fluid- and structural-solvers, and implementation of the new methods devised are realized by implementing them in accordance with the **BACI** coding style.

1.6 Structure of the thesis

The organization of this thesis follows a similar structure to the objectives presented in the previous section. Each chapter addresses different physical fields involved in the development of fluid-structure-fracture interaction, and every chapter is designed to be self-content to the best of my efforts; this means that from fundamental concepts and the associated governing equations, to the complete numerical methodology and implementation aspects of a physical problem are completely addressed within the corresponding chapter. In few necessary places,

additional computational procedures necessary for implementation of the illustrated numerical method are pushed to the appendices.

In the presentation of the thesis, it is assumed that the reader is familiar with the fundamentals of linear and nonlinear FE methods, and the related continuum mechanics concepts. Only the necessary details that are highly relevant for the description of the proposed methodologies are briefly reviewed at the corresponding places. The remainder of this thesis is structured as follows.

Chapter 2 is devoted to the numerical modeling of fracture in homogeneous, isotropic, hyperelastic materials. It first starts with a review of microscopic physical events associated with fracture, and then moves on to briefly explain how macroscopic theory of fracture mechanics has evolved. This chapter, then, succinctly details the governing equations together with the relevant continuum concepts. After that, a short review of the existing computational methodologies to model a sharp crack is presented. Then, the mesh refitting method proposed in this work is discussed in detail together with the necessary implementation details. Finally, a range of numerical examples are presented to study the accuracy of crack propagation.

Accurate and robust numerical integration methods over arbitrary polyhedra developed in this thesis for embedded interface methods are presented in **chapter 3**. This chapter is concerned with the solution of fluid flow using embedded interface methods (EIM). It starts with the strong form of the Navier-Stokes equations, and then proceeds to present the associated weak form using stabilized EIM. The challenges and importance of accurate weak form integration in EIM are detailed, and the drawbacks of existing methods are reviewed. Then, the two numerical integration methods developed in this thesis, namely the moment fitting and direct divergence methods are discussed in detail. This chapter ends with several numerical examples that study the accuracy, robustness and computational efficiency of different integration schemes.

How the crack propagation method and the fluid solver described in the previous two chapters are coupled to devise a fluid-structure-fracture interaction methodology, is the focus of **chapter 4**. This chapter begins with the comparison of monolithic and partitioned coupling schemes, and explains why the partitioned approach is chosen for this work. Then, the governing equations for both fluid and structure, together with the coupling conditions are presented. After this, the strongly coupled partitioned coupling algorithm to simulate fluid-structure-fracture interaction is described. Finally, some numerical examples of increasing complexity are illustrated to demonstrate the applicability of the devised coupling scheme to solve complete fracture of structure due to the fluid loading.

The final part of this thesis, **chapter 5**, summarizes the salient points of the present work; the accomplishments and the major results are recalled. Moreover, several potential extensions are proposed to enhance the present formulations and algorithms to be directly used in biological fluid-structure-fracture interaction applications envisioned.

When dealing with crack propagation in linear elastic materials, a few computational steps involved in the crack propagation algorithm described in chapter 2 can be simplified. This alternative procedure which is applicable only to model fracture in linear elastic materials under small strain assumptions is presented in Appendix A.

For completeness, the procedure of finding the equation of plane for a simple polygon and the method of computing correct normal on polyhedra are presented in Appendix B. These are useful when implementing the numerical integration techniques presented in chapter 3.

2 Structure & Fracture

Fracture is, in general, an undesirable phenomenon in man-made engineering structures. When a crack is nucleated within a structure, it does not necessarily mean that the structure is fractured and that it is no more useful. A real-world structure, in practice, can accommodate thousands or even millions of cracks within its safe operating regime [30], and only those located at the highly strained regions are potential sources of fracture initiators. A structure is said to be fractured, only when the cracks are grown within the structure to an extent that it considerably affects the strength or stiffness of the material. Hence, as far as structural safety is concerned, assessing the onset of fracture is of paramount importance in wide range of engineering disciplines.

As in many fields of engineering, computational modeling became indispensable in fracture mechanics, owing to the cost effectiveness and the level of details one can obtain from the simulations. It has already been stated that this work is a first step towards the simulation of fluid-structure-fracture interaction in biological applications. It is assumed that the material involved in such applications is nonlinear elastic and brittle. This is a valid assumption because most of the biological materials exhibit large deformation, with little ductility [31]. This means that the dissipation takes place only in the crack tip vicinity and the bulk dissipation can be neglected.

This chapter describes a nonlinear FE methodology to model the propagation of through-thickness cracks within a homogeneous, isotropic, hyperelastic material. The current chapter first succinctly reviews the microscopic physical processes associated with the crack propagation. Then, it recalls how the macroscopic theory of fracture mechanics has evolved. It proceeds, then, to present the governing equations of structural mechanics, together with the associated continuum mechanics concepts. Thereafter, this chapter summarizes the existing computational strategies for fracture mechanics, before addressing in detail the computational methodology developed in this work. The last part of this chapter includes various numerical test cases that analyze the accuracy of the proposed method.

2.1 Microscopic fracture processes

The physically relevant microstructural events associated with the fracture process are explained succinctly in this section. Irrespective of the size or the nature of the material, a small region in the neighborhood of the crack tip, known as the process region, encompasses all the microscopic processes of crack propagation. This region is subjected to very high stresses. The constitutive law that is applicable in other parts of the material is not suitable for the process region. Moreover, it is not amenable for continuum description.

Nucleation and growth of micro-separations within the process region is the significant process of crack growth. Micro-separation implies decohesion at the micro-structural level. What kind of micro-separations drive the crack growth depends on several factors.

Depending on the amount of plastic flow in the process region and the energy required for the propagation, the process of crack growth can be classified into either brittle or ductile fracture. For a more detailed explanation, one can refer to the well-known literatures [30,32,33].

2.1.1 Brittle fracture

Brittle fracture is characterized by rapid propagation of a crack through the material, with negligible plastic deformation before the failure occurs. Metals at low temperature, ceramics and ice exhibit this type of fracture. The micromechanism of brittle fracture is the nucleation of micro-cracks through cleavage; the atomic bonds are gradually broken along the fracture plane. Cleavage fracture occurs when the energy required to separate atoms is lower than the energy required to emit dislocations from crack tip as in ductile fracture [34].

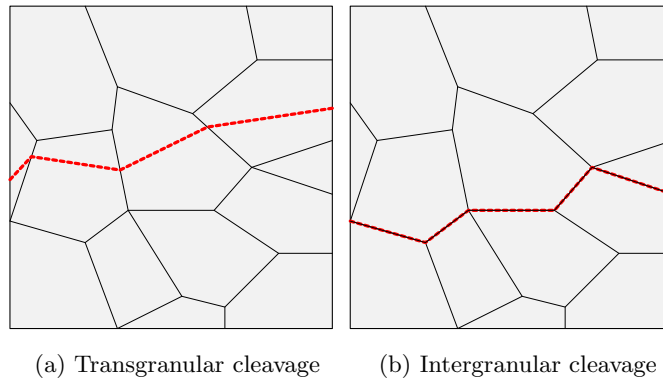


Figure 2.1: Brittle fracture

Micro-cracks can be formed at any point of high stress concentration. One possible mechanism is that due to the combination of high stresses together with elastic anisotropy mismatch at the grain vertices or junctions, micro-cracks are nucleated by rupturing the atomic bonds [30]. This type of fracture does not require the presence of any larger inhomogeneities of importance.

Cleavage propagation can either be transgranular or intergranular in nature. In the former, the fracture extends through the grain along the path of least resistance (figure 2.1a), and in the later, cracks travel along the grain boundaries (figure 2.1b).

2.1.2 Ductile fracture

Most materials at ambient temperatures exhibit ductile fracture behavior which is characterized by significant plastic flow in the crack tip vicinity. The formation and the growth of voids in the process region is the predominant physical mechanism associated with the crack growth in ductile materials [30, 35]. The events that lead to ductile fracture can be grouped into following 3 stages:

1. Nucleation of voids: When a ductile material is subjected to sufficiently large loads, due to a combination of stiffness differences and geometric factors, stress concentrations develop at inclusions and second-phase particles, after severe plastic flow. Such concentrations become high enough to nucleate micro-separations at particle-matrix interfaces (figure 2.2a). Once initiated, these micro-separations propagate along the interface and voids are eventually created by completely debonding the particles. Alternatively, voids may be formed due to particle cracking.
2. Growth of voids: During continued loading, the plastic strain and hydrostatic forces in their neighborhood enable the voids to grow. Simultaneously, the crack tip blunts as shown in figure 2.2b.

3. Coalescence: When voids grow further and further, they coalesce among themselves, and with the existing crack tip. Eventually, the crack starts growing (figure 2.2c).

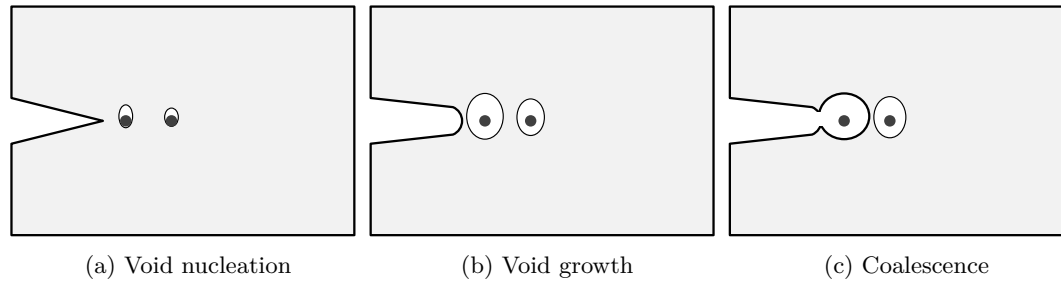


Figure 2.2: Ductile fracture

As long as these three processes take place within the process region, the crack grows continuously within the material.

2.1.3 Ductile–brittle transition

A material that exhibits ductile behavior may become brittle when the conditions influencing the material behavior such as temperature, particle shape and content, impurities at grain boundaries, and grain size are changed. Ferritic steels, one of the most technologically important materials, displays a large change in fracture toughness over a small temperature range. It behaves as a brittle material at low temperatures, and fails by ductile fracture when temperature is increased. In the transition region, both brittle and ductile micromechanisms co-exist. For example, in ferritic steels, the plastic flow by means of void growth and coalescence may be abruptly interrupted by unstable crack growth under cleavage as temperature is lowered [30]. This has serious practical implications; one of the well-known failure event is the sinking of Titanic. The steel used to construct the ship was ductile under ambient conditions, and became brittle when operated in the cold waters of Atlantic. The impact with the iceberg resulted in crack initiation, rapid brittle cleavage propagation, and subsequently lead to breakage.

The scales at which these microscopic fracture phenomena occur are not amenable for continuum description. For example, in ductile fracture, the size of the largest preexisting void is so small, about $1\mu\text{m}$ or even smaller. If one assumes that the average distance between the dislocations to be of the same order of magnitude, it is clearly evident that the continuum description is inappropriate in such regions.

Summary of the physical processes in the vicinity of crack tip is included in this thesis solely for the purpose of completeness, and to enable better understanding of the assumptions involved in the computational method. No other parts of the thesis address any physical mechanism of fracture process. Similar to the majority of other research works, the macroscopic crack tip parameters are used to study the evolution of crack within the material. The complete computational methodology is explained in detail in the forthcoming texts in this chapter.

2.2 Macroscopic description of fracture process

The previous section has highlighted the complex physical processes associated with the crack propagation. In general, in no field of engineering, such complex micro-structural details are directly taken into account: continuum assumption is invoked, and appropriate macroscopic theory

is developed. This section provides a brief account of how the macroscopic fracture theory has evolved. A more complete review of the historical development of fracture mechanics is described in [33,36].

No structure is perfect, and some initial defects/flaws are always present. In the presence of these flaws, the structure fails at much lower applied stress than the material strength, even if it is operating completely within the elastic regime. This is primarily due to the fact that these flaws act as stress concentration points, and elevate the stresses locally. **Inglis** analyzed the stress concentration effect due to an elliptical flaw within an infinite structure (figure 2.3), and derived the following expression for stress at the tip of the ellipse [37].

$$\sigma = \sigma_0 \left(1 + \frac{2a}{b} \right) \quad (2.1)$$

where σ_0 is the far-field applied stress, a and b are major and minor semi-axes, respectively. In the limit of this elliptical flaw representing a sharp crack ($a \gg b$), $\sigma \rightarrow \infty$. This is a paradox: owing to the infinite stress, the structure fails at the application of even infinitesimal load. However, real materials cannot develop infinite stresses, and yielding is expected to occur in a small zone, called as process region, close to crack tip.

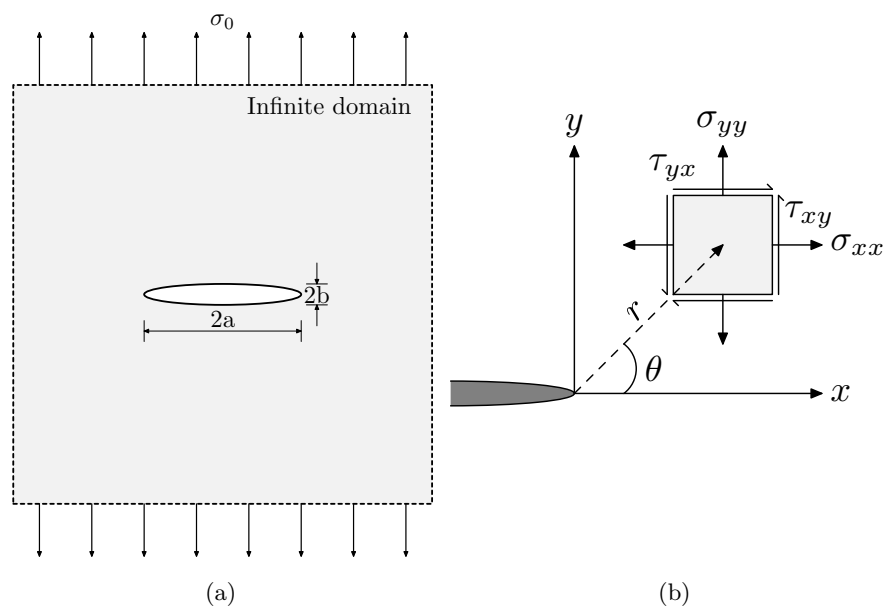


Figure 2.3: Infinite structural domain with an elliptical flaw: (a) Geometry & loading and (b) Local coordinate system at crack tip

In linear elastic fracture mechanics (LEFM), the following two important assumptions are made.

- **Infinitesimally small process region.** The process region around the crack tip is infinitesimally small in size when compared to the characteristic dimension of the structure (usually crack length). This helps us to proceed with continuum description of the material throughout the structure.
- **Small scale yielding.** Plastic flow is confined to a very small region near the crack tip. This permits us to use the linear elastic constitutive equations even in the close vicinity of crack tip.

The consequences of these assumptions make the analysis procedure simplified. However, when the process region is neglected, the Inglis' paradox prevails. **Griffith** eliminated this paradox by resorting to an energy based fracture theory [38], instead of stress based analysis of Inglis. His theory states that when strain energy change resulting from an incremental crack growth is adequate to overcome the surface energy of the material, then fracture gets initiated. For problem given in figure 2.3, the fracture stress is given by,

$$\sigma_f = \left(\frac{2E\gamma_s}{\pi a} \right)^{\frac{1}{2}} \quad (2.2)$$

where E and γ_s are the Young's modulus and surface energy of the material, respectively.

Griffith predicted accurately the relationship between strength of the material and flaw size, but only for ideally brittle materials like glass; the theory does not seem to work for metals. This limitation primarily arises from the assumption that the work of fracture is derived only from the surface energy of the material.

Irwin extended the energy theory of Griffith to metals by including the energy dissipation associated with the plastic flow [39]. A similar extension has also been provided by Orowan [40]. Equation (2.2) is modified as,

$$\sigma_f = \left(\frac{2E(\gamma_s + \gamma_p)}{\pi a} \right)^{\frac{1}{2}} \quad (2.3)$$

where γ_p is the plastic work per unit area of created crack surface. It should be kept in mind that though it considers plastic work, the global behavior of the material should be linear elastic.

In addition, Irwin has developed the following two concepts which are at the heart of fracture mechanics:

1. Energy release rate
2. Stress intensity factor

He has restated Griffith's energy theory in terms of a new quantity called energy release rate (\mathcal{G}) which quantifies energy available for an increment of crack extension [41].

$$\mathcal{G} = -\frac{\partial \Pi}{\partial A} \quad (2.4)$$

where $2A$ is the total surface area of crack, and Π is the potential energy supplied by internal strain energy and external forces. In other words, \mathcal{G} is a measure of fracture driving force. The fracture gets initiated when \mathcal{G} reaches its critical value \mathcal{G}_c . At the instant of fracture $\mathcal{G} = \mathcal{G}_c$; \mathcal{G}_c quantifies the measure of resistance to fracture, termed as fracture toughness.

Another most important contribution of Irwin is that he brought forth the concept of crack tip characterizing parameter [42], now known as stress intensity factor (K). He utilized the semi-inverse approach proposed by **Westergaard** [43] to analyze the stresses and displacements ahead of a sharp crack. A similar derivation has also been presented independently by **Williams** [44]. Stress field (σ_{ij}) in a cracked body exhibiting linear elastic behavior is given by,

$$\sigma_{ij} = \frac{K}{\sqrt{2\pi r}} f_{ij}(\theta) + \text{higher order terms} \quad (2.5)$$

where r and θ are defined in figure 2.3b, f_{ij} is a dimensionless function of θ . It can be seen that K is the single quantity that characterizes the stress field at the crack tip; it contains all

the influence of geometry and loading conditions. Hence it can be used as a crack propagation criterion; fracture occurs when K reaches its critical value, $K = K_c$. Similar to \mathcal{G}_c , K_c is also a measure of fracture toughness. One of the essential assumptions in LEFM is that the measures of fracture toughness (K_c and \mathcal{G}_c) are assumed to be size-independent geometry-independent material properties.

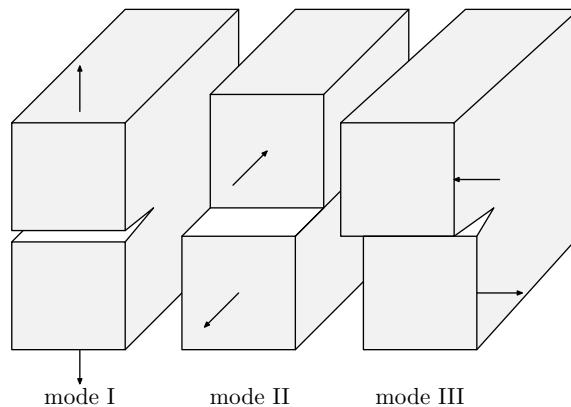


Figure 2.4: Three modes of loading a material that has a crack

The stress tensor can be decomposed into three modes of loading a crack can experience (figure 2.4): model I loading tends to open the crack faces, whereas modes II and III represents in-plane and anti-plane shearing. A cracked structure can experience any one of these loadings, or mixture of these modes. In mixed mode problems, due to linear superposition principle, the stress components are additive.

$$\sigma_{ij}^{total} = \sigma_{ij}^I + \sigma_{ij}^{II} + \sigma_{ij}^{III} \quad (2.6)$$

With the aforementioned contributions, one can conclude that the LEFM theory is well established. However, LEFM is applicable only for ideally brittle materials; any significant plastic deformation in the crack tip vicinity makes the LEFM based predictions invalid.

To alleviate the restrictions associated with LEFM, elastic-plastic fracture mechanics considers, among others, two important parameters: crack tip opening displacement (CTOD) [45] and J -integral [46]. By idealizing elasto-plastic deformation as nonlinear elastic behavior, **Rice** [46] has derived a contour integral that quantifies energy release rate for both linear and nonlinear elastic materials. This path independent J -integral characterizes the complete crack tip conditions in nonlinear materials [47, 48]. For the crack tip coordinate system given in figure 2.3b, it is expressed as,

$$J = \oint_{\Gamma_j} \left(W dy - \mathbf{T} \frac{\partial \mathbf{d}}{\partial x} ds \right) \quad (2.7)$$

where Γ_j represents a closed contour around the crack tip, \mathbf{T} and \mathbf{d} are traction vector and displacement vector on Γ_j respectively, and W is the strain energy density. J -integral quantifies energy release rate in nonlinear elastic materials. In case of elastic behavior, $J = \mathcal{G}$. It is an important quantity in nonlinear fracture mechanics, as the complete fracture behavior is characterized by this single quantity.

2.3 Governing equations

As already stated in section 2.1, from the physical perspective, the continuum model does not hold within the process region around the crack tip, irrespective of whether it is a brittle or a ductile material. However, as explained in the previous section, the assumptions involved in LEFM, which are also used in nonlinear fracture mechanics, enable us to treat the entire material as continuum, together with the possibility of using a single constitutive equation in all parts of the structure. This section reviews the essential continuum mechanics concepts before stating the governing equation of the structure. There exists a fairly large number of books on continuum mechanics for example [49–52], those can be consulted for further understanding of the concepts briefed here.

2.3.1 Kinematics

The present work employs the total Lagrangian approach for the structural field in which the kinematic equation of motion of a continuum is described by

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) \quad (2.8)$$

where \mathbf{x} is the position of a point in current configuration i.e., the configuration of continuum at current time t , and \mathbf{X} denotes the position of a point in reference configuration at time t_0 . The displacement of a material point from reference position to the position at time t , is described as

$$\mathbf{d}^s(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X} \quad (2.9)$$

A material element $d\mathbf{X}$ at t_0 is, by virtue of motion, transformed into material element $d\mathbf{x}$ at time t . The relation between $d\mathbf{X}$ and $d\mathbf{x}$ is given by

$$d\mathbf{x} = \mathbf{F}d\mathbf{X} \quad (2.10)$$

where \mathbf{F} is the deformation gradient at \mathbf{X} , which is given by

$$\mathbf{F} = \nabla_0 \mathbf{x} = \mathbf{I} + \nabla_0 \mathbf{d}^s \quad (2.11)$$

where \mathbf{I} is the second-order identity tensor and ∇_0 represents the gradient operator with respect to reference configuration \mathbf{X} . \mathbf{F} contains all the information about the deformation occurring at \mathbf{X} including the rigid body rotation. In order to quantify the stretch of material points, the following result from the polar decomposition theorem is used. Since \mathbf{F} is a real-valued tensor with non-zero determinant, it can be decomposed as

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} \quad (2.12)$$

where \mathbf{R} is the proper orthogonal tensor which contains information about rigid body rotation (volume-preserving), and \mathbf{U} is positive definite symmetric right-stretch tensor (volume-changing). Since a differential element is considered to define \mathbf{F} , rigid body displacements do not enter the decomposition. Based on these, the right Cauchy-Green tensor is defined as

$$\mathbf{C} = \mathbf{F}^\top \cdot \mathbf{F} = \mathbf{U}^\top \cdot \mathbf{U} = \mathbf{U}^2 \quad (2.13)$$

In the above, the orthogonality of \mathbf{R} is exploited. While \mathbf{F} describes mapping between infinitesimal line elements, \mathbf{C} describes mapping of their squares, i.e.,

$$d\mathbf{x} \cdot d\mathbf{x} = d\mathbf{X} \cdot \mathbf{C}d\mathbf{X} \quad (2.14)$$

The Green-Lagrange strain tensor (\mathbf{E}) which is defined from \mathbf{C} is the common choice of strain measure in nonlinear solid mechanics, and is defined as,

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^\top \cdot \mathbf{F} - \mathbf{I}) = \frac{1}{2} (\mathbf{C} - \mathbf{I}) \quad (2.15)$$

It is to be mentioned that when material element exhibits pure rigid body motion $\mathbf{E} = 0$.

2.3.2 Stress measure

The state of internal stresses within a continuum can be described by the traction vector field. For any internal point P located on surface S whose normal vector is \mathbf{n} , the traction vector is given by

$$\mathbf{t}(\mathbf{x}, t, \mathbf{n}) = \lim_{\Delta A \rightarrow 0} \frac{\Delta \mathbf{f}}{\Delta A} \quad (2.16)$$

where $\Delta \mathbf{f}$ is the resultant force acting a small area ΔA (around P) located on S . From Cauchy stress principle, the traction vector can be written as

$$\mathbf{t}(\mathbf{x}, t, \mathbf{n}) = \boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n} \quad (2.17)$$

In the above equation $\boldsymbol{\sigma}$ is the Cauchy stress tensor, which represents the physical state of internal stresses with diagonal and off-diagonal terms denoting normal and shear stresses respectively. However, it is defined with respect to the current configuration which is a priori unknown. Several alternative stress definitions exist in nonlinear continuum mechanics. However, since the Green-Lagrange strain is used in this work, the stress measure should be its energy conjugate pair [50], which is the second Piola-Kirchhoff stress tensor. It is defined completely in terms of the reference configuration, and is stated in terms of \mathbf{F} and $\boldsymbol{\sigma}$ as follows.

$$\mathbf{S} = (\det \mathbf{F}) \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-\top} \quad (2.18)$$

2.3.3 Initial boundary value problem

Let the structure occupies domain Ω_0^s bounded by Γ^s at reference time $t = t_0$ (figure 2.5). The boundary Γ^s is divided into three non-overlapping portions such that $\Gamma^s = \Gamma_D^s \cup \Gamma_N^s \cup \Gamma_c^s$ in which Γ_D^s and Γ_N^s are Dirichlet and Neumann portion of the boundary respectively, and Γ_c^s denotes the crack surfaces which contains always two physical crack faces $\Gamma_c^s = \Gamma_{c+}^s \cup \Gamma_{c-}^s$. The balance of linear momentum equation is written as,

$$\rho^s \ddot{\mathbf{d}}^s - \text{Div}(\mathbf{F}\mathbf{S}) = \rho^s \mathbf{b}^s \quad \text{in } \Omega_0^s \times (0, T) \quad (2.19)$$

where ρ^s is the density of structure, \mathbf{b}^s represents externally applied body force per unit mass, $\ddot{\mathbf{d}}^s = \frac{d^2 \mathbf{d}^s}{dt^2}$. T is the end time of the considered time interval, and $\text{Div}(\cdot)$ is the divergence operator defined with respect to the material reference frame.

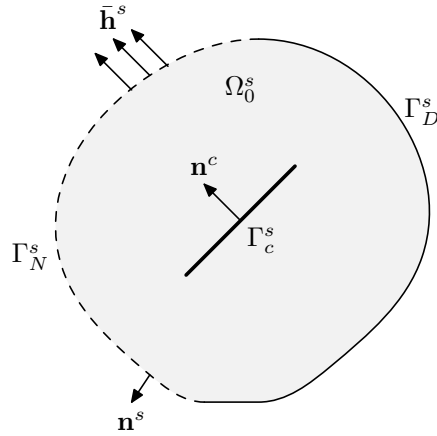


Figure 2.5: A schematic representation of structural domain containing a crack

Since this is an evolutionary problem involving second order time derivative, initial conditions must be specified on \mathbf{d}^s and its first derivative $\dot{\mathbf{d}}^s = \frac{d\mathbf{d}^s}{dt}$

$$\mathbf{d}^s|_{t=0} = \mathbf{d}_0^s \quad \text{on } \Omega_0^s \quad (2.20a)$$

$$\dot{\mathbf{d}}^s|_{t=0} = \dot{\mathbf{d}}_0^s \quad \text{on } \Omega_0^s \quad (2.20b)$$

On the boundary of the domain, Dirichlet conditions are specified on Γ_D^s , Neumann conditions are prescribed on Γ_N^s , and the crack surfaces are assumed to be traction-free.

$$\mathbf{d}^s = \bar{\mathbf{d}}^s \quad \text{on } \Gamma_D^s \times (0, T) \quad (2.21a)$$

$$(\mathbf{FS}) \cdot \mathbf{n}^s = \bar{\mathbf{h}}^s \quad \text{on } \Gamma_N^s \times (0, T) \quad (2.21b)$$

$$(\mathbf{FS}) \cdot \mathbf{n}^{c+} = 0 \quad \text{on } \Gamma_{c+}^s \times (0, T) \quad (2.21c)$$

$$(\mathbf{FS}) \cdot \mathbf{n}^{c-} = 0 \quad \text{on } \Gamma_{c-}^s \times (0, T) \quad (2.21d)$$

In order to close this system of equations, and to model the response of a specific material under loading, constitutive equations are required which is explained below.

2.3.4 Constitutive equations

Constitutive equations relate the state of stress and the corresponding strain experienced by the material. This work deals only with hyperelastic materials which is associated with the concept of existence of the strain energy function (Ψ). It is defined as follows

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} \quad (2.22)$$

In all the simulations presented in this thesis, the structure is modeled as Neo-Hookean material that defines a nonlinear stress-strain relationship. The strain energy function for such material is given as

$$\Psi_{\text{NH}} = \frac{\mu^s}{2} (\text{tr } \mathbf{C} - 3) - \mu^s \ln J + \frac{\lambda^s}{2} (\ln J)^2 \quad (2.23)$$

where J is the determinant of deformation gradient $J = \det \mathbf{F}$, λ^s and μ^s are Lamé's constants that are related to Young's modulus (E^s) and Poisson's ratio (ν^s) as

$$\lambda^s = \frac{E^s \nu^s}{(1 + \nu^s)(1 - 2\nu^s)} \quad (2.24)$$

$$\mu^s = \frac{E^s}{2(1 + \nu^s)} \quad (2.25)$$

Since the material is assumed to be isotropic and homogeneous, two material parameters are sufficient to completely describe its behavior and these parameters are the same constants at each point within the domain.

2.4 Overview of the existing computational methods

The present work focuses on the development of a mesh refitting approach to model crack propagation through the material. Before explaining the proposed method, the salient points of the widely used approaches in fracture mechanics are recalled briefly. The purpose of this section is to explain the motivation behind the development of a new method instead of directly adapting an existing approach. Hence the literature review presented here is illustrative rather than exhaustive.

The available approaches which sharply represent the cracks in an elastic structure are based on either linear elastic fracture mechanics (LEFM) principles [53–58] or cohesive zone concepts [59–63]. The focus of the present work is on the lines of LEFM, and majority of the existing computational methods in this category utilize either one of the following frameworks:

- Adaptive remeshing
- Enriched partition of unity

These methods are very powerful in their own right, and finds plethora of applications. However, implementing them in an existing (large scale) finite element (FE) package poses several challenges. Moreover, these methods offer enormous challenges and complexities when one attempts to couple them into fluid-structure interaction (FSI) modules.

Adaptive remeshing methods: As the name implies, these methods involve adaptively modifying the mesh, based on an error indicator. As one can expect, these methods highly refine the mesh in the crack tip vicinity where the solution dictates the dynamics of crack propagation, and coarsen the mesh away from the crack tip. These methods make use of special data structures [53, 54, 64], together with either a globally adaptive remeshing procedure or a local mesh modification algorithm to accommodate crack propagation at arbitrary directions within the computational domain. In globally adaptive remeshing procedure [55–57], each time when the crack is extended, the complete structure is remeshed. In contrary, the local mesh modification algorithms [58] modify the mesh locally near the crack tip region, and in locations that are far from the crack tip, the mesh remains unchanged. The computational tools that are developed based on both of the aforementioned methods are specifically designed to address fracture mechanics problems. These methods introduces several new nodes into the mesh, each time when crack extends. As a result, they require mesh generation related algorithms to modify the mesh

appropriately, and necessitates to ensure proper load balancing in a parallel simulation. However, usually in a general FE code which is generalized to address multiscale and multiphysics problems, such fracture-specific and mesh-modification routines are not encouraged.

Enriched partition of unity methods (EPUM): These methods represent state of the art in computational fracture mechanics, and gained maximum attention in recent years. They include extended finite element methods (XFEM) [65–68] and generalized finite element methods (GFEM) [69,70], and these class of methods are originally developed with an objective to eliminate the adaptive remeshing and its associated complex and time-consuming operations. The fundamental idea behind this method is to enrich the finite element solution space with additional problem-specific enrichment functions. In crack propagation problems, the enrichment includes a heaviside function to model the discontinuity across the crack surface, and the Westergaard crack tip displacement functions to accurately represent the singular fields associated with the crack tip. In this class of methods, the crack can propagate within the interior of an element, in contrast to adaptive remeshing based methods in which the crack propagates only along the mesh edges. Hence, this method makes it possible to simulate crack propagation without modifying the underlying discretization. Though these methods are demonstrated to be powerful, the following points hamper their easier implementation into an existing structural mechanics solver. The numerical integration of weak forms, which is taken for granted in standard FE method, is still an active area of research in EPUM [71–77]. This is owing to the singularity of the enrichment functions used over the elements that contains a crack tip. In addition, they are not optimal for handling more than one crack. More importantly, EPUMs model the crack as a surface of discontinuity. This means that the crack face opening is not explicitly modeled, and hence additional complications arise while dealing with the fluid flow introduced traction forces on the crack faces while modeling fluid-structure-fracture interaction. Some additional points on this issue are presented in section 4.3.

Owing to the aforementioned general implementation issues and the specific difficulties associated with FSI coupling, neither EPUM nor adaptive remeshing methods are ideal for developing fluid-structure-fracture interaction methods. In order to meet our requirements, a mesh refitting procedure is developed to model mixed-mode crack propagation. The objective is not to devise a method which is competitive to EPUM or adaptive remeshing method in terms of computational efficiency or accuracy. Rather, the focus is on developing a crack propagation approach which facilitates, with minimal implementation efforts, (1) to update the existing large scale structural mechanics solver into a robust tool to handle single and multiple cracks, and (2) to couple the crack propagation method with existing FSI approach to model fluid-structure-fracture interaction. These objectives are accomplished by utilizing a mesh refitting procedure together with the nodal releasing technique, the details of which are presented in the next section.

The present method shares a similarity with arbitrary Lagrangian Eulerian (ALE) based methods that it involves a mesh modification step. However, instead of describing the governing equations with respect to a moving mesh, the Lagrangian description is retained. Before explaining the complete method, the ALE based methods for fracture mechanics are briefly recalled. The use of ALE in computational fracture mechanics is not widespread. Only few studies employed ALE to address crack propagation problems. In the following, a brief account of all the works that relies on ALE formulation is provided.

Existing ALE based crack propagation methods: Though ALE formulations are widely used in several solid mechanics applications (refer to [78] for an overview), less than a handful of

researchers used ALE to handle fracture mechanics problems. The first use of ALE is described in [79, 80] to model dynamic crack propagation. The governing equations on a moving mesh, its associated finite element weak form, and the computation of dynamic energy release rate and stress intensity factors are explained in detail. The authors of these articles demonstrated the capability of their method by simulating a few mode-I dynamic crack propagation, and comparing them with analytical relations. In this work, the material separation is not explicitly modeled. Moreover the existing Lagrangian FE code framework cannot be directly extended to include this model. In order to achieve this, an ALE based method has been developed in [81] in which ALE boundary conditions are consistently treated using a special procedure. The method is applied to simulate both mode-I as well as mixed-mode dynamic crack propagation examples. Another method that uses isoparametric mapping based mesh motion algorithm is presented in [82]. A self-similar dynamic crack propagation problem in double cantilever beam is solved. The conclusion of this work make a note that ALE methods are more robust, and can be a powerful alternative to remeshing. For better understanding of dynamic crack growth in FRP composites, an ALE based method together with a contact mechanics approach is developed in [83]. A very fine mesh in the neighborhood of the crack tip is maintained throughout the simulation with the help of remeshing procedure. This method is then used to study the interfacial debonding phenomenon in FRP strengthened reinforced concrete beams. An attractive method that combines the advantages of element free Galerkin (EFG) method and ALE is presented in [84]. This method moves a cloud of nodes along with the crack tip, so that the vicinity of crack tip is always adequately resolved. As EFG is a meshless method that does not require nodal connectivities, such implementation of ALE to maintain high nodal density in the preferred region is accomplished effectively without resorting to remeshing strategies (refer to [85] for complications involved in implementing such a method in mesh-based FEM). An approach to transfer the solution over the moving mesh is also detailed, and the method is successfully applied to simulate wave propagation and dynamic crack propagation.

To summarize, neither a complex crack trajectory of single crack nor simple propagation of multiple cracks within a material is modeled until now using ALE based methods. This is predominantly due to the fact that the mesh modification method used in ALE, in its classical sense, cannot handle the mesh topology changes that are introduced by propagating a crack through the FE mesh. Continual remeshing is mandatory to avoid mesh tangling problems. In this work, this is avoided by using an additional step in the mesh refitting procedure that allows to modify element connectivity locally to preserve the quality of mesh. The complete details of the methodology is presented next.

2.5 The mesh refitting approach

One of the major reasons why devising a computational methodology to deal with fracture mechanics is challenging is the fact that the crack propagates in arbitrary directions through the material. If the dynamics of crack were known a priori, one can design an optimal mesh that allows the propagation of a crack through a preexisting mesh at each instant. Since this is not the usual case, the mesh has to be repeatedly modified to accommodate the advancement of cracks within the FE mesh, if one does not use EPUM. In the present method, the mesh refitting procedure is used to allow for the appropriate mesh modification, together with nodal releasing approach.

The complete numerical methodology, together with the computer implementation aspects, of the present approach to model crack propagation are presented in this section. The structural

material is assumed to be homogeneous, isotropic and time- and rate-independent hyperelastic, and it is assumed that the fracture behavior of the material is completely characterized by J -integral.

Algorithm 1 Computational crack propagation procedure at each time step

- 1: Construct local coordinate system at crack tip
 - 2: Compute vector J -integral
 - 3: Check whether crack propagates or not, from crack propagation criterion
 - 4: **if** Crack propagation criterion is not satisfied **then**
 - 5: continue to the next time step
 - 6: **end if**
 - 7: Obtain direction of crack propagation from crack kinking criterion
 - 8: Find the new crack tip nodes
 - 9: Apply Mesh refitting procedure
 - 10: Propagate the crack by nodal releasing technique
 - 11: Check whether crack has reached the end of the structure
-

The focus of the present work is to simulate through-thickness mixed-mode crack propagation within a structure using brick elements. The current work is an extension of Tabiei & Wu [86] which aims at implementation of crack module in DYNA3D FE package, and shares similarities with Miehe & Gürses [54] in which a thermodynamically consistent method is proposed. Both works do not involve a complete mesh modification step; rather only the nodes through which the crack is predicted to propagate in the next time step are repositioned appropriately. Though this approach works for relatively simpler examples, when the crack tip moves along a curved path (refer section 2.6.3) or exhibits a complex trajectory (refer section 2.6.7), it does not lead to satisfactory results. Moreover, the complex geometry related operations like deciding the new crack tip nodes are not addressed in depth. These details are crucial for implementation of the method. The method proposed in [54] was called as r -adaptive method, but in order to avoid confusion with complex r -adaptive mesh redistribution methods [87, 88], the present method is labelled as mesh refitting approach. The following section presents the complete implementation details of the present method.

At each time step, the governing equations are solved using nonlinear FE method. Then, the solution obtained is used to perform crack propagation related operations as described in Algorithm 1.

2.5.1 Solve the governing equations

The solution of structural dynamic equations using finite element methods is briefly recalled here. For a more elaborate discussion of FEM, the reader can refer to the literature [50, 89].

The first step in FE solution procedure is to derive the weak form associated with the governing equations and boundary conditions. The strong form given in equation (2.19) is multiplied by appropriate test functions ($\delta \mathbf{d}^s$) and are integrated over the structural domain to obtain weak form which is stated as,

Find $\mathbf{d}^s \in \mathcal{W}_d$ such that for all $\delta \mathbf{d}^s \in \mathcal{V}_d$, the following holds

$$(\delta \mathbf{d}^s, \rho^s \ddot{\mathbf{d}}^s)_{\Omega_0^s} + (\text{Grad } \delta \mathbf{d}^s, \mathbf{FS})_{\Omega_0^s} = (\delta \mathbf{d}^s, \rho^s \mathbf{b}^s)_{\Omega_0^s} + \langle \delta \mathbf{d}^s, \bar{\mathbf{h}}^s \rangle_{\Gamma_N^s} \quad (2.26)$$

where $(\cdot, \cdot)_{\Omega_0^s}$ and $\langle \cdot, \cdot \rangle_{\Gamma_N^s}$ mean the standard L^2 -inner product over the reference domain and Neumann part of the boundary, respectively. The test function is denoted as $\delta \mathbf{d}^s$, since the weak form associated with the structural equations is a representation of the principle of virtual work, with $\delta \mathbf{d}^s$ being the virtual displacements.

The solution space and the test function space are defined as

$$\mathcal{W}_d = \{\mathbf{d}^s \in \mathbf{H}^1(\Omega_0^s) \mid \mathbf{d}^s = \bar{\mathbf{d}}^s \text{ on } \Gamma_D^s\} \quad (2.27)$$

$$\mathcal{V}_d = \{\delta \mathbf{d}^s \in \mathbf{H}^1(\Omega_0^s) \mid \delta \mathbf{d}^s = 0 \text{ on } \Gamma_D^s\} \quad (2.28)$$

In order to convert the above integral equations into a system of algebraic equations, FEM discretizes the domain Ω_0^s into a number of non-overlapping simple shaped regions called elements. Elements are defined by their nodes, and interpolation or shape functions (N) can be formed over each element such that

$$\mathbf{d}_h^{s,(e)}(\mathbf{X}, t) = \sum_{i=1}^{N_n} N_i^{(e)}(\mathbf{X}) \mathbf{d}_i^s(t) \quad (2.29)$$

where subscript \cdot_h represents a discretized quantity, N_n is the number of nodes in the considered element (e), and \mathbf{d}_i^s is the nodal displacements. For structural mechanics problems, it is the usual practice to use Bubnov-Galerkin methods in which the test functions $\delta \mathbf{d}^s$ are also interpolated using the same shape functions. Moreover, the use of isoparametric concept enables us to use the same functions to interpolate the geometry at reference and current configurations.

Plugging in the discrete approximation of displacement (equation (2.29)) and other relevant quantities into the weak form given in equation(2.26), one arrives at the following semi-discrete system of nonlinear differential equations.

$$\mathbf{M}^s \ddot{\mathbf{d}}^s + \mathbf{F}_{\text{int}}^s(\mathbf{d}^s) - \mathbf{F}_{\text{ext}}^s = 0 \quad (2.30)$$

where \mathbf{M}^s is mass matrix, $\mathbf{F}_{\text{int}}^s$ represents nonlinear vector of internal forces, and $\mathbf{F}_{\text{ext}}^s$ is the vector of external forces. For brevity the subscript \cdot_h is omitted.

The above nonlinear differential equations are discretized in space but continuous in time. The next step is to perform time discretization i.e., to replace the time derivatives by corresponding discrete differentials. Among many available time integration methods [89], the implicit generalized- α method introduced by Chung and Hulbert [90] is employed. This is based on the well-known Newmark method [91] that allows expressing the quantities at time level t^{n+1} completely in terms of known quantities at t^n , i.e.,

$$\dot{\mathbf{d}}^{s,n+1} = \frac{\gamma}{\beta \Delta t} (\mathbf{d}^{s,n+1} - \mathbf{d}^{s,n}) - \frac{\gamma - \beta}{\beta} \dot{\mathbf{d}}^{s,n} - \frac{\gamma - \beta}{2\beta} \Delta t \ddot{\mathbf{d}}^{s,n} \quad (2.31)$$

$$\ddot{\mathbf{d}}^{s,n+1} = \frac{1}{\beta \Delta t^2} (\mathbf{d}^{s,n+1} - \mathbf{d}^{s,n}) - \frac{1}{\beta \Delta t} \dot{\mathbf{d}}^{s,n} - \frac{1 - 2\beta}{2\beta} \Delta t \ddot{\mathbf{d}}^{s,n} \quad (2.32)$$

where the parameters $\gamma \in [0, 1]$ and $\beta \in [0, \frac{1}{2}]$ characterize the method. Generalized- α method shifts the evaluation from t^{n+1} to the midpoints $t^{n+1-\alpha_f}$ and $t^{n+1-\alpha_m}$ introduced in the method.

Quantities at these midpoints are defined by the following interpolation

$$\mathbf{d}^{s,n+1-\alpha_f} = (1 - \alpha_f)\mathbf{d}^{s,n+1} + \alpha_f\mathbf{d}^{s,n} \quad (2.33a)$$

$$\dot{\mathbf{d}}^{s,n+1-\alpha_f} = (1 - \alpha_f)\dot{\mathbf{d}}^{s,n+1} + \alpha_f\dot{\mathbf{d}}^{s,n} \quad (2.33b)$$

$$\ddot{\mathbf{d}}^{s,n+1-\alpha_f} = (1 - \alpha_f)\ddot{\mathbf{d}}^{s,n+1} + \alpha_f\ddot{\mathbf{d}}^{s,n} \quad (2.33c)$$

with parameters $\alpha_f, \alpha_m \in [0, 1]$. The space and time discretized finite element weak form at the intermediate points is written as follows

$$\mathbf{M}^s \ddot{\mathbf{d}}^{s,n+1-\alpha_m} + \mathbf{F}_{\text{int}}^s(\mathbf{d}^{s,n+1-\alpha_f}) - \mathbf{F}_{\text{ext}}^{s,n+1-\alpha_f} = 0 \quad (2.34)$$

with

$$\mathbf{F}_{\text{ext}}^{s,n+1-\alpha_f} = (1 - \alpha_f)\mathbf{F}_{\text{ext}}^{s,n+1} + \alpha_f\mathbf{F}_{\text{ext}}^{s,n} \quad (2.35)$$

The parameters α_f and α_m enable us to control the amount of numerical dissipation introduced in the scheme, which is an advantage when compared to the Newmark method.

In order to obtain the approximate solution using the FEM, the system of nonlinear algebraic equations represented in equation (2.34) must be solved at each time step. In this thesis, the Newton-Raphson method is utilized for this purpose. Equation(2.34) is recast into residual form as follows.

$$\mathbf{r}^s(\mathbf{d}^{s,n+1}) = \mathbf{M}^s \ddot{\mathbf{d}}^{s,n+1-\alpha_m} + \mathbf{F}_{\text{int}}^s(\mathbf{d}^{s,n+1-\alpha_f}) - \mathbf{F}_{\text{ext}}^{s,n+1-\alpha_f} = 0 \quad (2.36)$$

The Newton-Raphson method involves repeated linearization of the residual presented in equation (2.36), solution of the resulting linear system of equations, and updating the unknown field variables until convergence is met. Using the Taylor series, the nonlinear residual is linearized at the end of time step t^{n+1} as

$$\mathbf{r}^s(\mathbf{d}_{i+1}^{s,n+1}) = 0 = \mathbf{r}^s(\mathbf{d}_i^{s,n+1}) + \left. \frac{\partial \mathbf{r}^s(\mathbf{d}^{s,n+1})}{\partial \mathbf{d}^{s,n+1}} \right|_i \Delta \mathbf{d}_{i+1}^{s,n+1} + \text{h.o.t.} \quad (2.37)$$

where i and $i + 1$ denote iteration counter within the nonlinear solution loop. Neglecting the higher order terms (h.o.t.), the linear system can be rewritten as

$$\left. \frac{\partial \mathbf{r}^s(\mathbf{d}^{s,n+1})}{\partial \mathbf{d}^{s,n+1}} \right|_i \Delta \mathbf{d}_{i+1}^{s,n+1} = -\mathbf{r}^s(\mathbf{d}_i^{s,n+1}) \quad (2.38)$$

The equation above represents a system of linear algebraic equations that can be solved to obtain the incremental displacements ($\Delta \mathbf{d}_{i+1}^{s,n+1}$). After obtaining the incremental values, the displacements are updated within the iterative procedure.

$$\mathbf{d}_{i+1}^{s,n+1} = \mathbf{d}_i^{s,n+1} + \Delta \mathbf{d}_{i+1}^{s,n+1} \quad (2.39)$$

These iterative steps are successively carried out until a user-defined convergence criterion is met. At this point the converged solutions for displacements are obtained.

2.5.2 Perform computational crack propagation procedure

The solution obtained from the previous step by freezing the crack in the structure is used to perform crack propagation procedure by computing vector J -integral. It involves eight discrete steps, each of which are detailed below. Since the present work simulates only through-thickness crack, for brevity, Quad and Tri elements are used to represent hexahedral and Wedge elements, respectively.

Step 1: Construct local coordinate system at crack tip

To compute fracture mechanics quantities from FE solution, and to decompose these quantities into their corresponding modes in a mixed-mode problem, it is essential to construct local coordinate system $(\underline{\xi}, \underline{\eta})$ at the crack tip (\mathbf{x}_c) as shown in figure 2.6. Let $(\mathbf{e}_1 - \mathbf{e}_2)$ represents the base vector in $(\underline{\xi}, \underline{\eta})$. In order to do so, elements that are attached to crack tip (shaded quadrilateral areas in figure 2.6) are considered, and the nodes that are connected to \mathbf{x}_c and lying on the crack surface (open circles in figure 2.6) are extracted. Then, the point \mathbf{x}_m which is the middle point of the nodes lying on the crack surface is defined. The base vectors are computed as follows. \mathbf{e}_1

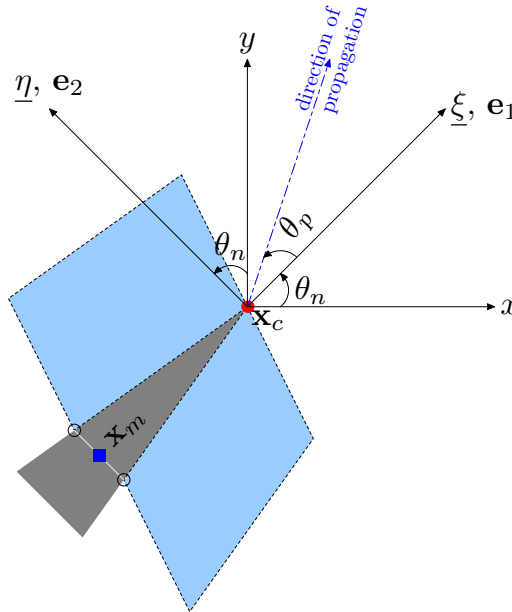


Figure 2.6: Construction of local coordinate system at crack tip. The shaded Quads represent finite elements.

is given by,

$$\mathbf{e}_1 = \frac{\mathbf{x}_c - \mathbf{x}_m}{\|\mathbf{x}_c - \mathbf{x}_m\|} \quad (2.40)$$

\mathbf{e}_2 is directly obtained by computing normal to \mathbf{e}_1 in right hand coordinate system.

Step 2: Compute vector J -integral

As already stated, J -integral quantifies the strength of singularity at the crack tip in nonlinear elastic materials. Moreover, it is a single parameter that dictates whether the crack propagates or

not, if at all it propagates in which direction it advances. Hence, it is essential to accurately evaluate the J -integral from the displacements and stresses obtained from FE solution of structural equations.

With respect to the spatial configuration, the energy release rate along the direction of crack is defined as,

$$J = \oint_{\gamma_j} \left(w n_{\underline{\xi}} - \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \frac{\partial \mathbf{d}}{\partial \underline{\xi}} \right) d\gamma \quad (2.41)$$

where γ_j is the integration contour, w is the strain energy stored per unit deformed volume, \mathbf{n} is the normal to contour γ_j and $\boldsymbol{\sigma}$ is the Cauchy stress tensor. All these quantities are defined in the current spatial configuration as shown in figure 2.7a. Since the current work employs total Lagrangian formulation, it is convenient to express J -integral in the reference configuration using pull-back operations. As shown in [92], it reads as

$$J = \oint_{\Gamma_j} \left(W N_{\underline{\Xi}} - \mathbf{N} \cdot \mathbf{P} \cdot \frac{\partial \mathbf{d}}{\partial \underline{\Xi}} \right) d\Gamma \quad (2.42)$$

where Γ_j , W , \mathbf{N} are the corresponding quantities in reference configuration, \mathbf{P} is the first Piola-Kirchhoff stress tensor and $(\underline{\Xi}, \underline{H})$ denote the crack tip coordinate system in the reference configuration.

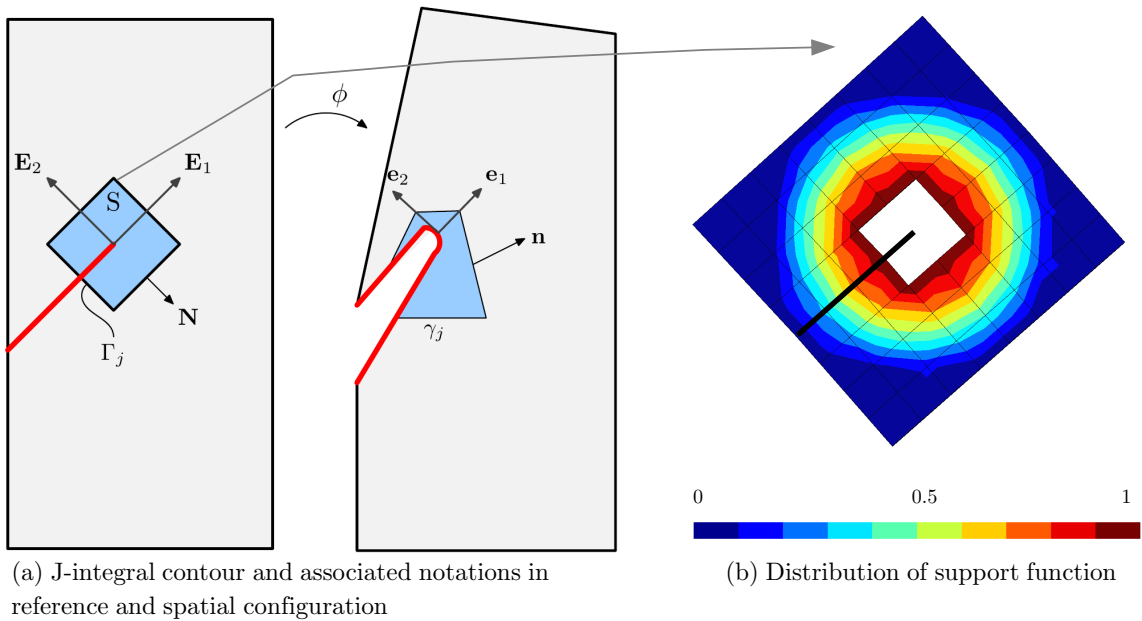


Figure 2.7: J -integral: notation and distribution of support function

In FEM, the contour integrals are cumbersome to implement because the material variables are available only at the Gauss points. Interpolating these variables over the desired contour presents complications, in addition to introducing interpolation errors. Hence, several studies [93,94] have proposed the idea of converting the contour integral into integration evaluated over a finite domain around the crack tip by applying the divergence theorem. This procedure is straightforward to implement, as it requires only the quantities at Gauss points within the elements falling over the finite domain. The equivalent domain vector J -integral for finite strain problems is given

as [92, 95],

$$\mathbf{J} = \int_S \left(\frac{\partial \mathbf{d}^\top}{\partial X} \cdot \mathbf{P} - W\mathbf{I} \right) \cdot \nabla_0(q) dS \quad (2.43)$$

where S is the domain enclosed by Γ_j , and $\nabla_0(q)$ is the gradient of support function q with respect to the reference configuration.

In order to construct q , using nodal connectivity information, all the elements that are located on n -layers around the crack tip (see figure 2.7b with $n = 4$) are considered. From this, the elements connected to crack tip are deleted. Then, all the nodes that are falling on the outer boundary of this element set are located, and among these nodes, the one which has the shortest distance (r_{\min}) from the crack tip is chosen. Then the support function is initialized to take a value of unity at the inner layer of nodes, and drops smoothly to zero when the distance of a node from crack tip is more than or equal to r_{\min} . The distribution of q within the integration domain is given in figure 2.7b.

Step 3: Check crack propagation criterion

Crack propagation criterion determines whether the existing crack propagates through the structure under the current stress state. The crack propagation occurs when the driving force reaches or exceeds the material resistance. The J -integral provides a measure of driving force for fracture. The fracture toughness is the material's resistance to crack propagation. As stated earlier, critical values of J -integral quantifies the fracture toughness, which is assumed to be a material property.

The present work makes use of the vector J - integral based crack propagation criterion proposed by Ma & Korsunsky [96]. This criterion requires, first, the calculation of maximum strain energy release rate, which is given by the magnitude of J -vector.

$$G_{max} = \sqrt{J_1^2 + J_2^2} \quad (2.44)$$

where J_1 and J_2 are the strain energy release rate along the direction of \mathbf{e}_1 and \mathbf{e}_2 respectively, which are given by simple dot products $J_1 = \mathbf{J} \cdot \mathbf{e}_1$ and $J_2 = \mathbf{J} \cdot \mathbf{e}_2$.

The crack extension under the given loading conditions occur, when G_{max} reaches the critical value of J - integral

$$G_{max} \geq J_c \quad (2.45)$$

In the above expression, G_{max} represents the maximum available crack driving force and J_c is the measure of fracture toughness. When the above condition is met, the crack propagates along the direction of G_{max} , which is given in the next step.

If the crack propagation criterion is not satisfied, then there is no need to perform the remaining operations. The present time step is completed, then the algorithm moves to the next time step to solve governing equations of the structure.

Step 4: Determine crack kinking criterion

The aforementioned crack propagation criterion decides only whether the current material state enables the crack to propagate or not; it does not provide any information about the direction of crack propagation. Hence, after confirming that crack propagates, the next logical step is to determine along which direction it is going to advance through the material. Crack kinking criterion provides us the direction of crack propagation.

There are several methods put forward to determine the crack kinking direction, and the most important methods are

- Maximum circumferential stress criterion [97]
- Minimum strain energy density criterion [98]
- Maximum energy release rate criterion (MERR) [96, 99]

It is concluded in a comparative study [100] that the minimum strain energy density criterion is less accurate, and the accuracy of maximum circumferential stress criterion and maximum energy release rate criterion are equivalent in all the tests considered. For more information on the details of these different methods, readers can refer to [100].

This work incorporates the maximum energy release rate criterion, proposed in [96], which is consistent with the crack propagation criterion given in the last section. It is stated that the crack propagates when G_{max} reaches or exceeds the characteristic fracture toughness of the material. MERR predicts the crack propagation direction (θ_p) to be the direction of G_{max} , which is simply given as

$$\theta_p = \tan^{-1} \left(\frac{J_2}{J_1} \right) \quad (2.46)$$

It is to be remembered that θ_p is measured with respect to the crack normal, as indicated in figure 2.6. Moreover, in this work, the extent of crack propagation is always set to be the length of one complete edge of an element.

Step 5: Find new crack tip nodes

Having computed the crack propagation direction from \mathbf{J} , the next essential step is to determine the new crack tip nodes—nodes in the FE mesh through which crack must be propagated. A geometry based method is developed in this work to decide the new tip nodes; the complete procedure is explained as follows, and an example situation is given in figure 2.8 to aid the understanding of the procedure.

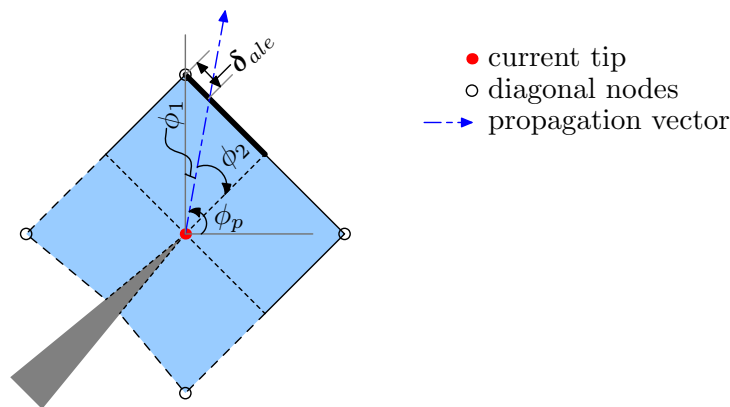


Figure 2.8: Procedure to find new crack tip nodes

The first step is to identify all the elements that are connected to the current tip node (figure 2.8). All the operations stated here are with respect to the current spatial configuration.

Then the angle θ_p is converted into ϕ_p that is measured with respect to the horizontal. Then it is necessary to loop through all the edges of these elements to decide which edge contains the new tip node. First, it involves identifying all the edges that contain the current tip nodes and neglecting these edges (short dotted lines in figure). Then, the angle formed by the line joining the current tip node to the edge nodes, and the crack propagation direction is calculated (ϕ_1 and ϕ_2 in figure). If both ϕ_1 and ϕ_2 are more than 90° , then this edge is neglected (long dotted lines in figure). The ordering of nodes in the edge does not influence the algorithm.

If ϕ_1 or ϕ_2 is less than `tol-zero` ($= 10^{-5}$ radians in all the simulations) that represents zero, and if the corresponding node is not a diagonal node, then it is marked as the new tip node. For the configuration shown in figure 2.8, ϕ_1 represents diagonal node and if $\phi_2 < \text{tol-zero}$, the crack propagates through an already existing edge in the mesh. In such a case, no additional procedure like nodal repositioning or element splitting is required, as will be clarified later.

If the above condition is not met i.e., the crack does not propagate along an existing mesh, then the focus will be on finding an edge that is intersected by the propagation vector. In order for this, the following condition is checked: if the sign of both ϕ_1 and ϕ_2 are the same, then the crack propagation vector does not intersect this edge. Hence, this edge is neglected (thin continuous lines in figure). If this condition is checked on all edges, one arrives at the edge intersected by propagation vector (thick continuous line in figure). Either one of the nodes of this edge should be the new crack tip node.

After getting the required intersecting edge, the next step is to check whether the crack propagates along the diagonal. This is realized by the condition $\phi_1 < \text{diag-tol}$ ($= 0.25$ radians in all the simulations). In this case, the diagonal node corresponding to ϕ_1 is marked as new tip node. Since the crack propagates through a diagonal of the element, this element must be split along this diagonal to accommodate crack propagation through the mesh; in this step, only the elements to be split, and the nodes along which they are split are marked. The actual splitting operation is performed in the next step. Moreover, if ϕ_1 is less than `tol-zero`, then no mesh modification is needed. If not, the distance between the point at which the propagation vector intersects the edge, and node is calculated (δ_{ale} in figure) and this value will subsequently be used as a boundary condition in the mesh refitting step.

If above check is not satisfied, the non-diagonal node is identified, and this is marked as the new tip node. Moreover, the corresponding δ_{ale} is calculated for further nodal repositioning step. It is worth mentioning that all the angles in this procedure are within $[-\pi, \pi]$ interval, and all these new tip nodes with $\delta_{ale} \neq 0$ are stored in \mathcal{R}_{ale} .

Step 6: Mesh refitting procedure

The necessity of performing mesh refitting step arises due to the fact that the crack does not always propagate along the existing mesh surface. If this were the case, $\phi_2 \approx 0$ and hence $\delta_{ale} \approx 0$ in figure 2.8, then this step would be absent. However, since the crack path is not known a priori, it is not possible to design a mesh that satisfies this criterion, and a dynamic nodal repositioning algorithm is mandatory. This means that at the end of this step, the mesh is refitted in such a way that the crack propagates along an existing edge in the refitted mesh.

The mesh refitting procedure used in the present work involves two discrete operations listed as follows.

1. Nodal repositioning
2. Splitting Hex into Wedge elements

In the first step, the nodes are repositioned without touching the elements. This means that the element topology—shape and total number of elements, or the connectivity between elements—remains unchanged. The elements only deform due to the movement of the nodes.

There are diverse methods available to perform mesh movement within the domain. All these methods solve a boundary value problem with conditions specified throughout the boundary of the domain. These methods include

- Springs [101–104]
- Laplacian method [105]
- Biharmonic equation [106, 107]
- Radial basis function method [108]
- Elastostatic equation [109, 110]

All of the aforementioned methods are developed in the context of arbitrary Lagrangian Eulerian (ALE) based FSI, where the fluid mesh is relaxed owing to the deformation of the structural boundary. In this case, a high aspect ratio viscous mesh is used to resolve the boundary layer around the structure, and these high aspect ratio elements pose a lot of challenge to the ALE approach. This phenomenon is highlighted in [111] where different methods of mesh motion schemes are compared; it is concluded that biharmonic method is capable of dealing with viscous meshes better than other alternatives at the expense of increased computational cost. However, in case of fracture problems, such problematic high aspect ratio elements do not exist, and the mesh movement in all the presented examples are amenable to elastostatics equations.

In the elastostatic approach, the mesh is treated as an elastic body, and the governing equations of the mesh movement together with the boundary conditions are

$$\nabla \cdot \boldsymbol{\sigma}^m = 0 \quad \text{on } \Omega^s \quad (2.47a)$$

$$\mathbf{d}^m = 0 \quad \text{on } \partial\Omega_{ale} \quad (2.47b)$$

$$\mathbf{d}^m = \boldsymbol{\delta}_{ale} \quad \text{on } \mathcal{R}_{ale} \quad (2.47c)$$

where $\boldsymbol{\sigma}^m$ is the Cauchy stress tensor, \mathbf{d}^m denotes the displacements at each node within the mesh, Ω^s represents the whole structural domain, and $\partial\Omega_{ale}$ denotes the boundary for ALE computations which is the closure of Ω including the crack surfaces: $\partial\Omega_{ale} = \Gamma_D^s \cup \Gamma_N^s \cup \Gamma_c^s$. Displacements at the new crack tip nodes are set to be $\boldsymbol{\delta}_{ale}$ that is computed in the previous step. It is to be mentioned that when $\boldsymbol{\delta}_{ale} = 0$ at all the crack tip nodes, then the crack propagates along the existing FE edge within a mesh, and the nodal repositioning step need not be performed.

The stress-strain relationship is given under the assumption of St.-Venant-Kirchhoff material,

$$\boldsymbol{\sigma}^m = \lambda^m \text{tr}(\boldsymbol{\epsilon}^m) \mathbf{I} + 2\mu^m \boldsymbol{\epsilon}^m \quad (2.48)$$

where λ^m and μ^m are Lamé's constants that characterizes the material, and the linear strain measure is given by

$$\boldsymbol{\epsilon}^m = \frac{1}{2} \left[\nabla \mathbf{d}^m + (\nabla \mathbf{d}^m)^\top \right] \quad (2.49)$$

The weak form of equation (2.47), obtained after multiplying with the test functions $\delta \mathbf{d}^m$ and integrating over Ω^s , is written as

$$(\delta \mathbf{d}^m, \nabla \cdot \boldsymbol{\sigma}^m)_{\Omega^s} = 0 \quad (2.50)$$

The above equation, with both material- and geometric-nonlinearity are neglected, renders a linear system of equations that are solved using the `Trilinos` package [112] to obtain the mesh displacement \mathbf{d}^m .

In problems which involve only very small mesh displacements, it is possible to formulate the above equations to evaluate the stiffness matrix only once [113], and proceed with this stiffness matrix throughout the simulation. However, in crack propagation simulations, in order to account for change in mesh topology due to crack propagation, and to accommodate crack curvature as in examples presented in section 2.6, the stiffness matrix associated with these equations are formulated at each time step.

In the next step of mesh refitting procedure, the `Hex` elements that are marked to be split are cut into two `Wedge` elements. This happens when the crack propagates very close to the diagonal of a `Hex` element (figure 2.9a). As can be seen in figure 2.9a, the quad element is split along the diagonal that has old and new tip nodes as its ends, In the present simulations, the hex elements are split similarly into two wedges. This process does not involve introducing new nodes in to the mesh. By comparing figure 2.8 and 2.9a, the effect of mesh refitting procedure is clear: the new tip nodes are first moved to the desired location using the nodal repositioning step, and then the `Hex` element is appropriately split into `Wedge` elements to enable crack propagation along the diagonal. In short, the combination of nodal repositioning and element splitting ensure that after the mesh modifications, the crack propagates along an edge in the new mesh.

One of the main reasons for the failure of ALE based methods in handling large deformation or topology change is that such methods maintain their nodal connectivity during the entire simulation. This splitting procedure alleviates this problem by enabling us to modify the connectivity between the elements locally. This is an essential step without which the nodal repositioning method cannot handle the change in mesh topology that is inherent to crack propagation problems, without resorting to complicated and time-consuming remeshing procedures.

After the nodes are repositioned, the field values at these new locations are simply interpolated from the converged solution. Since the nodes move only small distances, this does not seem to affect the solution significantly. However, this should be studied in the future.

Step 7: Nodal releasing technique

The two previous steps have enabled us to identify new tip nodes, and to move these nodes to match the computed propagation angle. The material separation is not yet included within FE procedure. In order to achieve this, and to form physical crack surfaces the crack propagation leaves behind, the nodal releasing technique is used.

In order to represent the material separation, the element connectivity at the current tip node must be modified; a duplicate node is created at the same location where current tip resides. Few elements are released from the current tip node, and are assigned with new duplicate node. This, in turn, generates new crack surfaces. In order not to destroy the FE mesh during this process, a consistent way of determining which elements get duplicate nodes is used; the remaining elements retain the current tip node.

In this procedure, two angles are defined: one is ϕ_p already defined in figure 2.8, and other is the angle formed by the negative normal at crack tip to the horizontal (ϕ_{-n} in figure 2.9a). Then, for each element, the angle (ϕ_g) formed by the line connecting current tip to the centroid of the element to the horizontal is computed. The element is released and gets the duplicate node, if $\phi_g \notin [\phi_p, \phi_{-n}]$. In this process, all angles are in the interval $[0, 2\pi)$. The elements that retain the current tip node are shaded in figure 2.9a. After nodal releasing and modifying element

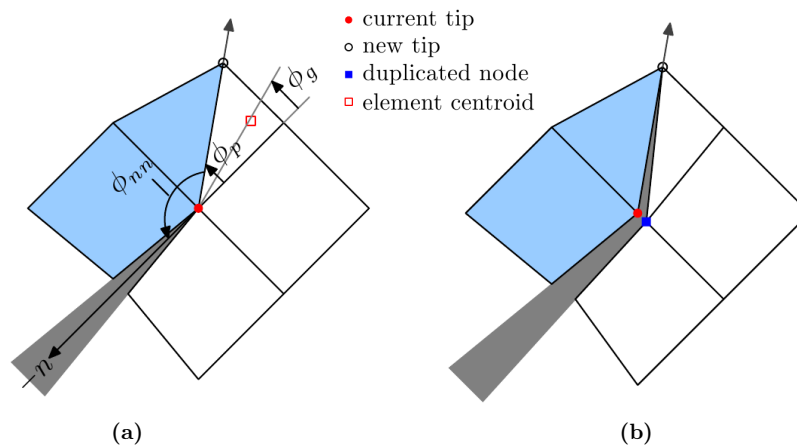


Figure 2.9: Nodal release technique (a) splitting an element to allow crack propagation along diagonal (b) modify connectivity locally near crack tip; the crack opening is shown only for visualization. Legend is common for both figures

connectivity, the mesh close to the crack tip is plotted in figure 2.9b. The crack has propagated through the mesh, and the material separation is introduced.

It is to be mentioned that the field values at the current tip nodes are copied to the newly generated duplicate nodes to enable time integration at the next step.

Step 8: Check whether crack tip reached the boundary

When the tip of the propagating crack reaches the boundary of the structure, the configuration resembles two completely different structures joined at a single point as shown in figure 2.10a. The matrix resulting out of the FE discretization, in such cases, is ill-conditioned, and the simulation eventually crashes. In order to avoid this, the configuration corresponding to this case is identified; after performing crack propagation at each time step, it is checked whether the crack has reached the boundary of the structure. To enable this, all the boundary nodes (including nodes falling on initial crack surface) are explicitly identified at the beginning of the simulation, and every new crack tip and duplicate nodes are added to this list. Once the crack tip has reached the boundary, then the nodal release technique explained in the previous step is once again applied; a duplicate node is created in the position of crack tip, and the connectivity information is modified to separate the structure into two different bodies (figure 2.10b).

2.6 Numerical examples

Several examples of varying complexity are solved to demonstrate the effectiveness of the proposed method. These examples exhibit single and mixed-mode behavior, involving mono- and multimaterials. Whenever possible, the crack path obtained from the present method is compared with experiments or results obtained from adaptive FEM or meshless methods. This is necessary in order to closely examine the accuracy of the method.

Plane strain condition is assumed in all the examples. The load is increased smoothly from zero to the given loading in order to eliminate any influence of inertia.

The first two examples consider stationary crack, and the quantities calculated are compared with an XFEM study [92]. These examples consider highly nonlinear effects evident from the crack

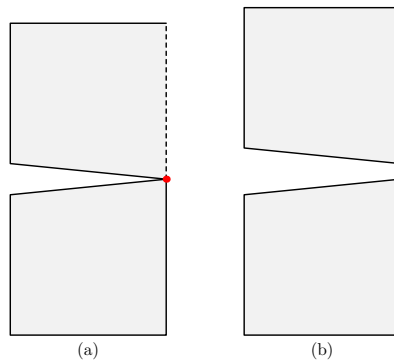


Figure 2.10: Splitting a single structure into two different structures (a) crack tip reaches structure surface (b) application of nodal release technique

tip blunting observed in the results. In addition to these two stationary crack examples, only one simulation involving crack propagation through the material with high material nonlinearity is solved. All the other examples focus on simulating the crack propagation examples presented in the literature and to thoroughly compare the results from the present method to the reported results.

2.6.1 Crack tip blunting

Consider a material, which is characterized by nonlinear elastic behavior, containing an initial crack. When the material deforms, the crack surfaces move apart, and the initially sharp crack will blunt significantly. In order to demonstrate this, a single edge notched specimen with dimensions $2 \text{ mm} \times 6 \text{ mm}$ is considered. The crack occupies half-width as shown in figure 2.11a. The material properties are $E = 0.4225 \text{ MPa}$ and $\nu = 0.3$. The top surface is subjected to the fixed displacement of 4 mm . These geometric configuration, material properties and boundary conditions are same as the example given in the study of XFEM-based large strain crack analysis [92]. The only difference is that the XFEM study considers incompressible materials, whereas the present work deals with compressible one.

The deformed configuration of the structure is shown in figure 2.11b. As expected, the initial sharp crack blunts due to the nonlinear elasticity. The vertical displacement of crack surface nodes are plotted against their horizontal position in the reference state in figure 2.11c; the values from the XFEM study [92] is also shown for the comparison. This matching between the present results and XFEM clearly shows that the final configuration obtained from the proposed method is accurate.

2.6.2 J -integral computation

In order to study the accuracy of J -integral evaluation in finite-strain nonlinear elastic conditions, it is computed at different stretch levels for the example shown in the previous section.

The plot of J -integral against stretch ratio (λ) is given in figure 2.12, where λ is defined as the ratio of deformed length to the original length. The J -integral values are computed for different Poisson's ratio and are compared with results presented for incompressible material [92]. It can be seen from the figure that as ν is increased, the curve moves towards the reference value given

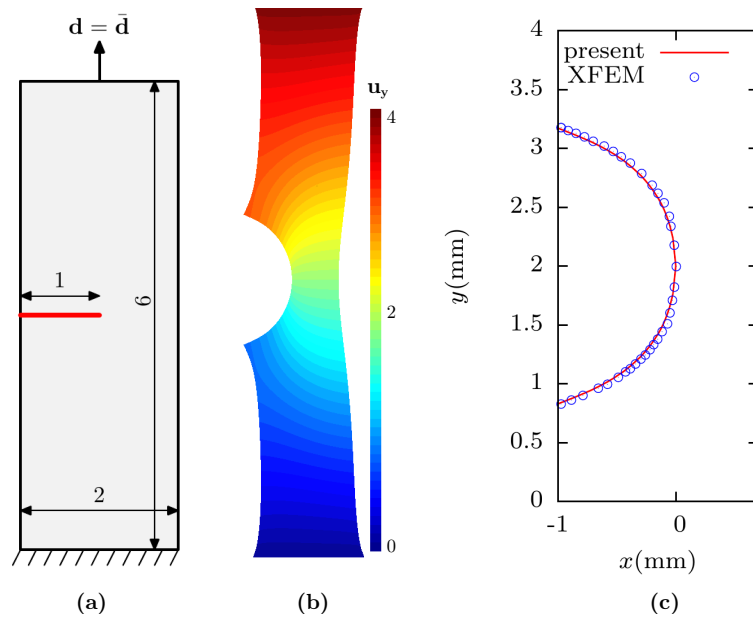


Figure 2.11: Crack tip blunting. (a) Geometry. All dimensions are in mm (b) Deformed configuration (c) Plot of vertical displacement of crack surface nodes against their horizontal position in reference state, and comparison with XFEM results [92].

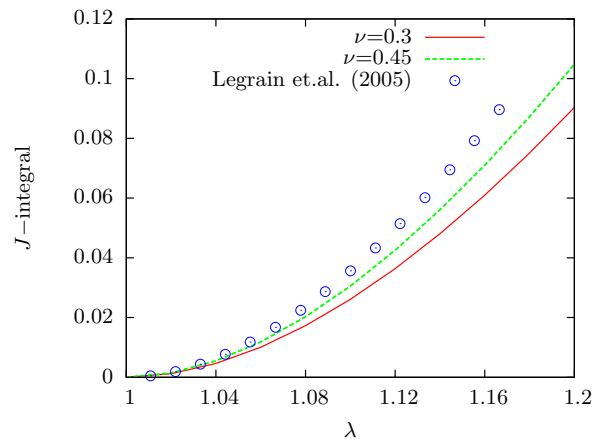


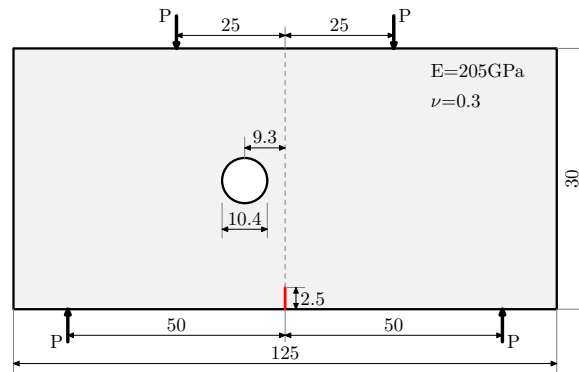
Figure 2.12: Computation of J -integral for different stretch ratios and comparison with XFEM results [92].

at incompressibility limit i.e., $\nu = 0.5$. Hence, it can be concluded that the J -integral obtained are in consistent with the reported results.

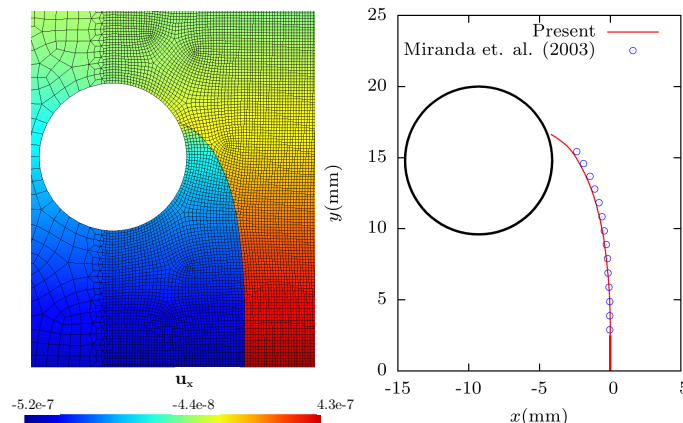
Having simulated a stationary crack, the remaining examples consider complex crack propagation through the structure. The comparison between the present results and the results obtained from the literature demonstrates the accuracy of the method.

2.6.3 Single edge notched beam with a hole

This simulation corresponds to the mixed-mode crack propagation in a single edge notched beam that contains an off-centre circular hole. The geometry, loading, and material properties of the simulation are illustrated in figure 2.13a. The beam is loaded at two points, and is supported from below at two points. The experimental observation of crack propagation for this specimen has been performed by [56], and several researchers (for example [114]) have studied this numerically.



(a) Geometry, material parameters and loading conditions (not to scale). All dimensions are in *mm*.



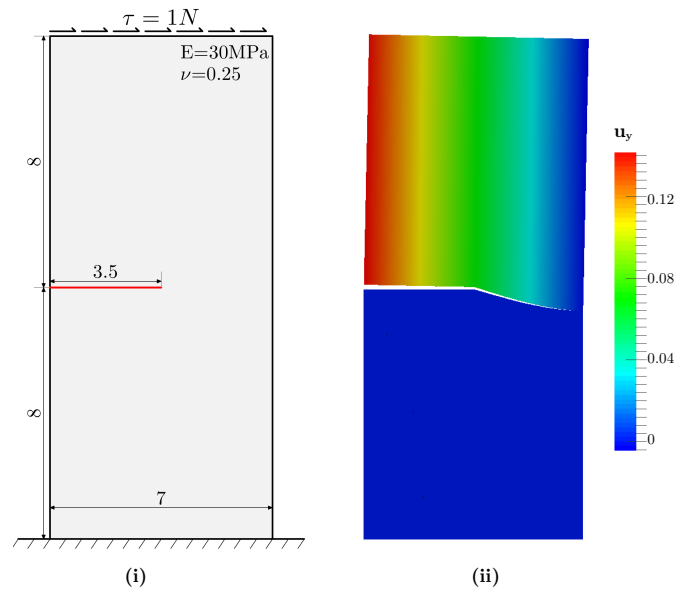
(b) Crack tip trajectory comparison with experiments of Miranda et. al. [56]

Figure 2.13: Single edge notched beam

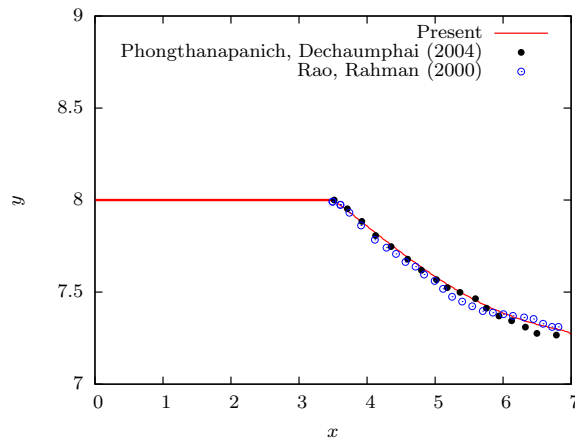
The contours of displacement in x -direction at the end of simulation time is given in figure 2.13b, from which it is possible to identify the crack path from the discontinuous displacement field. As can be clearly observed, the crack initially starts growing from the notch almost vertically before taking a curved path. The crack is then attracted towards the hole, and subsequently ends on the hole. These qualitative features are in very good agreement as reported in earlier studies. In order to further assess the accuracy of the present simulation, the trace of the location of crack tip is compared with the experimental results [56]. The comparison of crack paths is plotted in figure 2.13b, and it can be seen clearly that the results obtained from the present numerical simulation exactly matches with the experimentally reported crack tip trajectory.

2.6.4 Single edge cracked plate under mixed-mode loading

In this example, as shown in figure 2.14a(i), the plate which is fixed at the bottom edge, is subjected to a far-field shear stress $\tau = 1N$ on the top edge. The initial edge crack length is half of the plate width, and upon loading, the crack takes a slightly curved path and reaches the other end of the plate. The material parameters used in the simulation are also specified in figure 2.14a(i).



(a) (i) Geometry, material parameters and loading conditions; All dimensions are in mm. (ii) Contours of displacement in y -direction.



(b) Crack tip trajectory

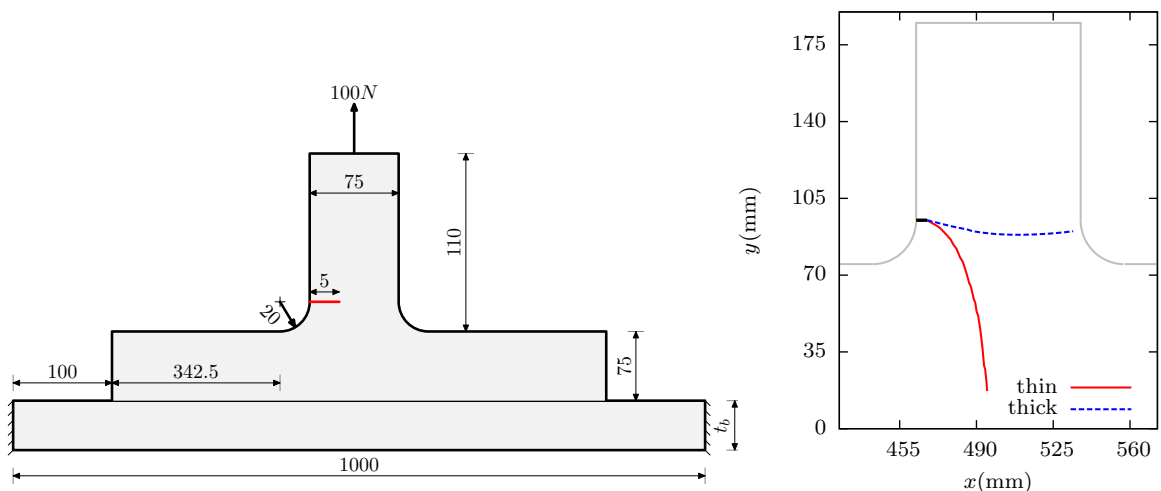
Figure 2.14: Single edge crack plate under mixed mode loading.

Contours of displacement in y -direction after the crack reaches the other end of the plate are displayed in figure 2.14a(ii). The crack surface is directly evident from the discontinuous displacement field, and the crack path qualitatively resembles the expected behavior. In order to provide detailed comparison, a zoomed view of crack path is plotted in figure 2.14b, and the

result obtained from present simulation is compared with two other studies: one utilizes meshless method [115], and another study based on adaptive FEM [57]. It can be seen that the predicted crack tip trajectory matches very well with the results obtained from other two studies.

2.6.5 Crack growth from a fillet

This example considers the growth of crack from a fillet in a structural member. The complete simulation setup is depicted in figure 2.15a. The configuration is inspired from the experimental study of Sumi [116] who included the influence of the bending stiffness and the welding residual stresses between the bottom beam and the structural member, on crack growth. In the present simulations, the effect of residual stresses are neglected, and as in [100, 117, 118], the bending stiffness of the structure is adjusted by varying the thickness of the bottom beam (t_b) shown in figure 2.15a. Two thickness values of bottom beam considered are 15mm and 315mm ; low thickness value corresponding to a flexible bottom beam, and higher value represents a rigid beam. The effect of this thickness value on crack path is reported. The material parameters considered are Young's modulus, $E = 200\text{GPa}$ and Poisson ratio, $\nu = 0.3$.



(a) Simulation setup (not to scale). Value of t_b is set to 15 and 315 for different cases. All dimensions are in mm

(b) Crack tip trajectory

Figure 2.15: Crack growth from fillet

The crack tip trajectory is shown in figure 2.15b, for both values of bottom beam thickness. When the beam is thin, the crack path takes a sharp curve immediately after the loading, and the crack proceeds towards the bottom. In contrast, when the thickness of the bottom beam is high, the crack grows almost straight and ends in the opposite fillet. These observations are in consistent with both the experimental [116] as well as numerical [100, 117, 118] observations.

It is essential to mention that in the region containing the crack and the fillet, same mesh is used for both simulations. This means that the different crack path predicted from the numerical simulations are due solely to the influence of the change in bending stiffness, not due to the effect of meshing pattern.

2.6.6 Crack deflection due to inclusion

Crack growth in the presence of an inclusion is studied in this example. This simulation tests the applicability of the present method to multimaterial applications. Geometry, loading, and boundary conditions are given in figure 2.16a; it is taken from [119]. The configuration consists of a rectangular plate which contains an off-centre circular inclusion. The young's modulus (E_{plate}) and Poisson ratio (ν) of the plate are assumed to be 20MPa and 0.3 respectively. The objective of this study is to check whether the method is capable of accurately predicting the influence of this inclusion on crack propagation, which is already reported in [100, 119].

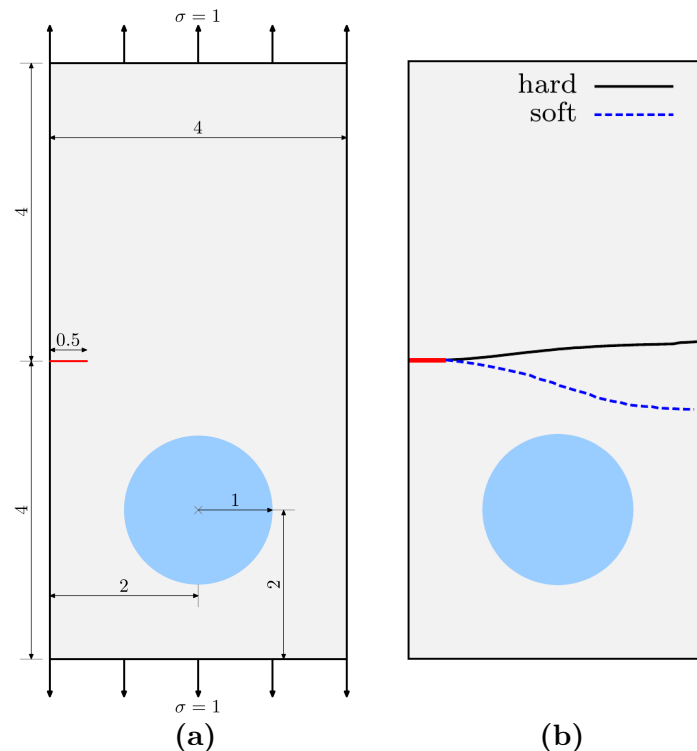


Figure 2.16: Crack deflection due to inclusion (a)Geometry and loading conditions (b)crack path

The inclusion is characterized by the ratio of Young's modulus of the plate to that of the inclusion ($r = E_{plate}/E_{incl.}$). Two values are considered; $r = 10$ which means that Young's modulus of inclusion is 10 times lower than that of the plate which is referred to as "soft" inclusion, and $r = 0.1$ that is referred as "hard" inclusion. The Poisson ratio is assumed to be same as that of the plate.

The effect of inclusion on crack tip trajectory is shown in figure 2.16b. For soft inclusion, the crack is attracted towards the side of inclusion; however, crack does not end in inclusion, as reported in section 2.6.3. In case of hard inclusion, the crack deflects away from the inclusion. These observations are in consistent with the already reported results [100, 119].

2.6.7 Crack in a drilled plate

To demonstrate further the accuracy of the proposed method to simulate crack path, the example given in [58] is considered. It reported the propagation of crack from initial notch in a Polymethylmethacrylat (PMMA) beam which has three drilled holes. The study carried out both

experimental and numerical tests, and observed a curvilinear crack propagation within the drilled plate. The geometrical configuration, material properties, and the loading conditions are given in figure 2.17(a). In this example, the stress/strain fields are influenced by the presence of holes in the beam, and this provides interesting curvilinear crack tip trajectories. There are two simulation cases considered based on the location of the initial notch. These are dictated by the choice of a and b in figure 2.17(a) whose values are given in table 2.1 for simulation-1 and simulation-2.

Simulation	a	b
1	5.0	1.5
2	6.0	1.0

Table 2.1: Geometric parameters defining notch location for Bittencourt’s drilled plate problem shown in figure 2.17(a).

Interestingly, the crack path follows different trajectory based on the choice of a and b . The crack paths from the present simulations, and the comparison with the experimental data provided in [58] are described as follows. Same mesh is used for both simulations.

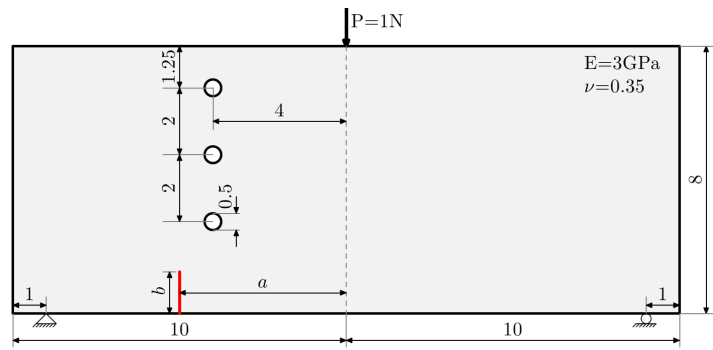
Simulation-1: The location of initial notch is given by $a = 5mm$ and $b = 1.5mm$. The crack is initially attracted towards the bottom hole, propagates near this hole, and got deflected away to end in the middle hole as shown in figure 2.17(b). This is in accordance with the experimental results of [58], and other numerical studies [54,57,120]. Comparison with the experimental results show that the present simulation produces very good results; even the crack deflection near the bottom hole is predicted well in the simulation as can be directly seen from figure 2.17(b). This is one of the very challenging validation test cases for fracture mechanics simulations, owing to the complex crack tip trajectory involved. The developed methodology can be said to be accurate as it produces results that are matching very well with the experimental values even for this complex configuration.

Simulation-2: In this example, for which $a = 6mm$ and $b = 1mm$, the crack is attracted towards the middle hole, and directly ends in it (figure 2.17(c)). There are no crack deflections observed, and this example is easier when compared to simulation 1. For this example as well, the results match excellently with the experiment (figure 2.17(c)).

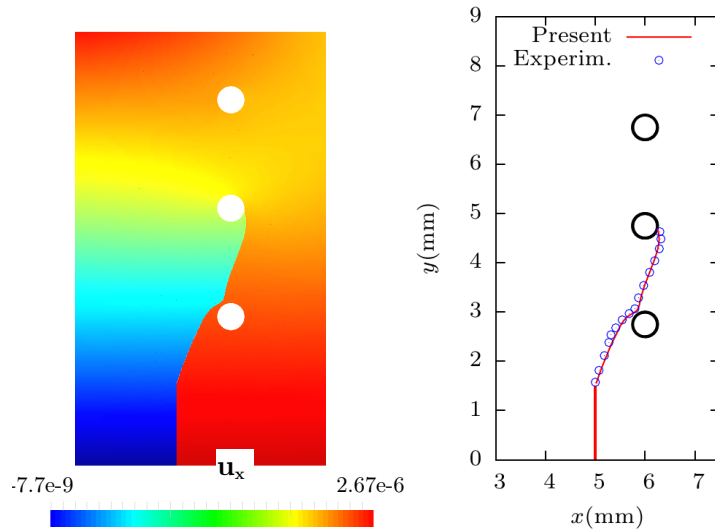
2.6.8 Four point beam with two notches

In order to test the performance of the present method to simulate multiple cracks in a structure, four point bending beam with two preexisting notches, shown in figure 2.18a, is simulated. The beam is supported from below at two points, and is loaded at two other points. The material properties are also given in figure 2.18a. This example is proposed by Bocca et. al. [121] who performed experiments on the structure, and also simulated them numerically. Of many variants of the experiments by [121], this particular configuration is chosen.

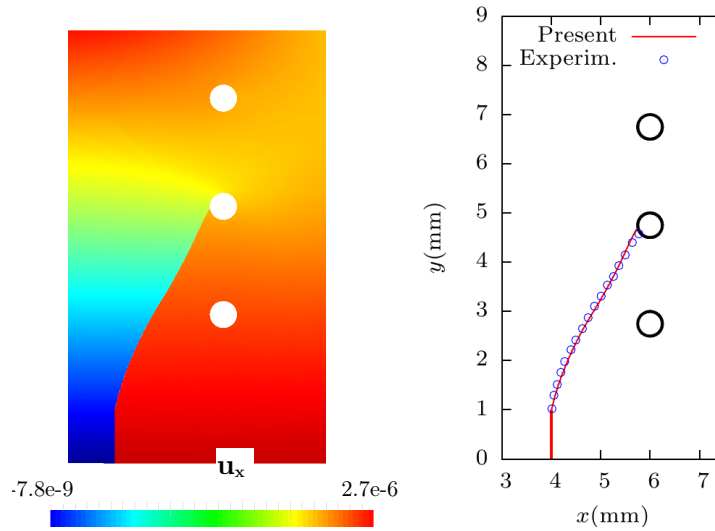
The contours of x-direction displacement, at the end of simulation, is presented in figure 2.18b, from which the two crack paths on the structure can be inferred. To demonstrate the accuracy of simulation, the crack paths obtained from the present method are compared with the results reported using a meshless method that incorporates crack tip singular fields as enrichments [122]; results presented for the finest meshless node distribution is used for the comparison. The comparison of crackpaths is plotted in figure 2.18c. It can be seen that for both crack tips,



(a) Geometry. All dimensions are in *mm*.

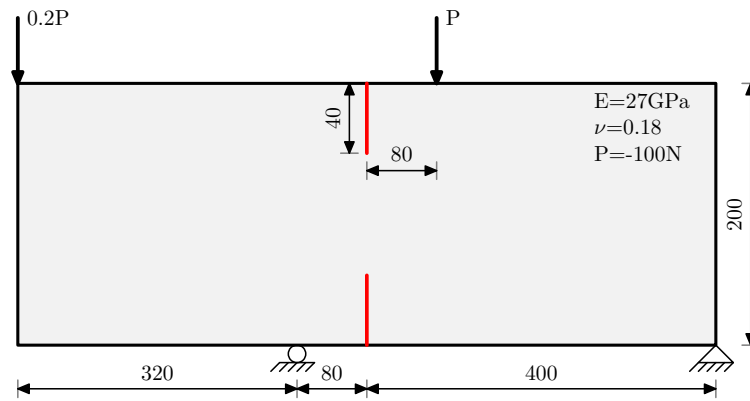


(b) Simulation-1

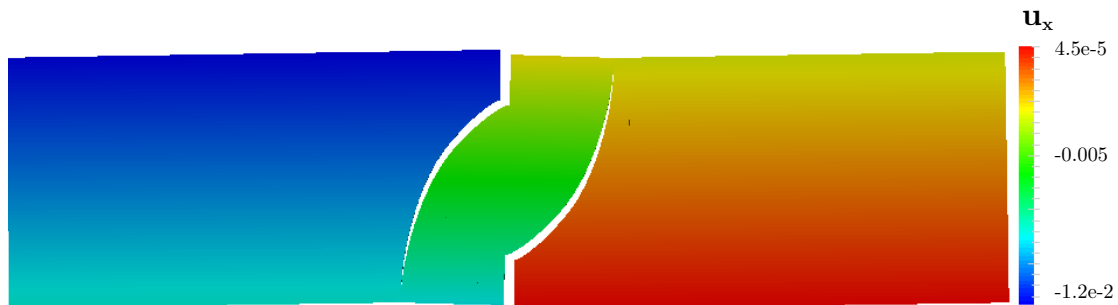


(c) Simulation-2

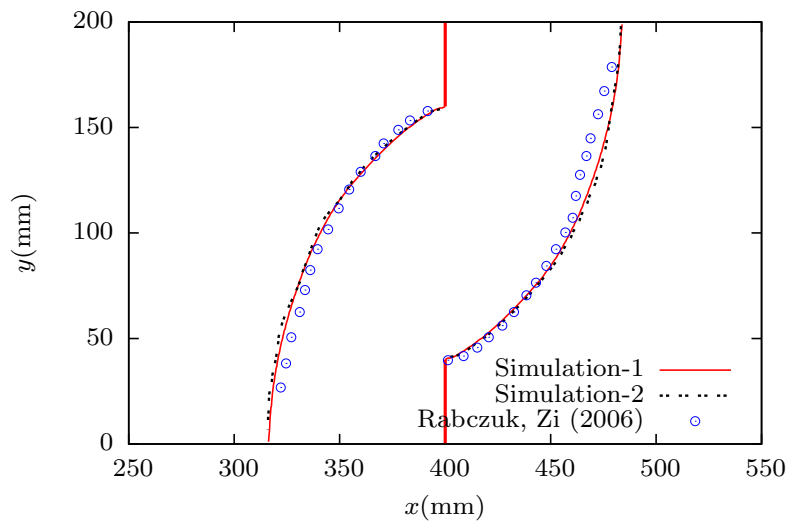
Figure 2.17: Bittencourt's drilled plate problem. Experimental values for comparison are taken from [58]



(a) Geometry, material parameters and loading conditions (not to scale). All dimensions are in *mm*.



(b) Contours of displacement in x-direction



(c) Comparison of crack tip trajectories

Figure 2.18: Four point bending beam with two notches

the tip trajectory obtained from the present simulations (simulation-1 and simulation-2 that are explained below) matches very well with the reported value. In [122], the authors reported that their results are in well agreement with the experimental values (comparison is not plotted). Since

the present results matches with those of [122], it can be concluded that the proposed method also compares well with experiments.

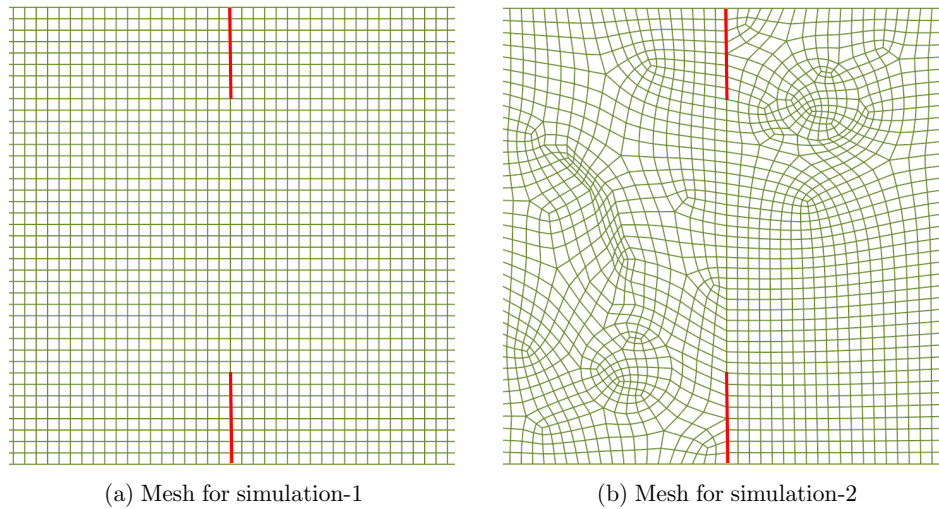


Figure 2.19: Close-up view of mesh patterns for the simulation of four point bending beam with two notches. For brevity, coarse version of actual meshes uses in the simulations are presented.

In this problem, two cracks grow simultaneously within the structure. Due to the complexity of the considered simulation, the results can be very sensitive to the orientation of mesh near the crack tip if the numerical methodology is not robust. In order to test the influence of mesh orientation, two simulations, denoted as simulation-1 and simulation-2 in figure 2.18c, are performed. Both simulations employ approximately same number of nodes and elements; only the orientation of mesh with respect to crack surfaces are significantly different from each other. Moreover, on purpose, unsymmetrical mesh over the domain with respect to crack surfaces are generated for simulation-2 in order to obtain meaningful conclusions. For brevity, coarser version of meshes used in these simulations are illustrated in figure 2.19. Figure 2.18c clearly demonstrates that despite this difference, both simulations produce almost same crack tip trajectory. From this, it can be concluded that the present methodology is robust enough to not get influenced significantly by mesh orientation.

2.6.9 Nonlinear elastic plate with a hole

The above examples considered crack propagation with little material nonlinearity. This is evident from the fact that the crack remains sharp even after several propagation steps. This example considers crack propagation involving high material nonlinearity under large deformation. A small off-centre hole is introduced in the geometric configuration considered in section 2.6.1, and this simulation allows the crack to propagate through the material. The mechanical properties of the material are $E = 1\text{GPa}$, $\nu = 0.3$ and critical J -integral, $J_c = 50\text{ Jm}^{-2}$. The top surface is subjected to displacement of 0.5 mm. The geometric configuration of this example is presented in figure 2.20a.

As with linear elastic examples, the loading (or the corresponding Dirichlet boundary condition here) is increased very smoothly from the zero initial value so that the influence of inertia is neglected. When the material starts deforming, as expected, the crack starts to blunt, and the J -integral value starts to increase. When J reaches J_c , then the crack starts to propagate; the

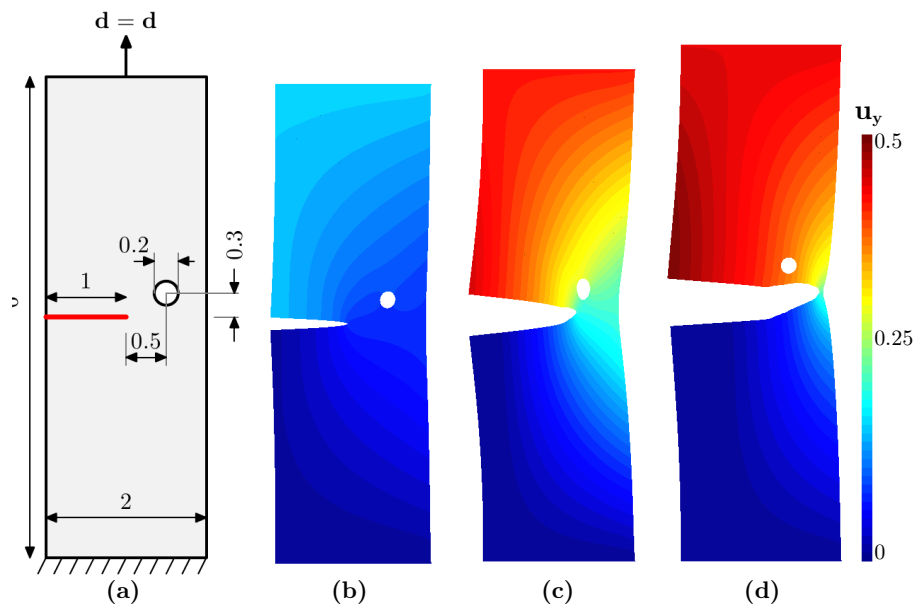


Figure 2.20: Plate with a hole. (a) Geometry (not to scale). All dimensions are in mm (b) The configuration at which crack starts propagating (c) An intermediate configuration (d) Final configuration

deformed configuration of the structure at which the crack starts propagating is depicted in figure 2.20b. Due to the presence of the hole, the crack slightly deflects upwards, as can be seen from figure 2.20b and 2.20c. From all these plots, one can infer that the crack tip is always blunt owing to the nonlinear stress-strain behavior.

2.7 Closure

The central focus of the current chapter was the development of a finite element methodology to model mixed-mode crack propagation through homogeneous, isotropic, nonlinear elastic materials undergoing finite strain deformations. The proposed method invoked the assumption that the fracture is completely controlled by vector J -integral. It involves two steps: in the first step, the governing equations of structure are solved using nonlinear FE method by freezing the crack in the structure; in the next step, the solution obtained from the FE method is used to propagate the crack based on maximum energy release rate criterion. Advancing the crack through a FE mesh requires a continual change in topology of the mesh, which is achieved in this work by utilizing a mesh refitting approach. This method, as the name suggests, refit the mesh at each instant of crack advancement in such a way that the crack propagates through an existing mesh edge in the modified mesh. The mesh movement strategies (for example used in ALE based methods) usually result in mesh tangling issues when attempt to handle topology changes in the mesh. This problem is circumvented in this work by splitting the hexahedral elements into wedge elements in the crack tip neighborhood, which allows the possibility of local mesh connectivity to be modified. This step is crucial to preserve the quality of mesh throughout the simulation, without which the mesh movement methods will fail.

The accuracy of the methods was tested initially by simulating several examples of crack propagation through linear elastic material under small strain conditions. The obtained results

are compared with experimental and other available computational methods. The comparison demonstrated that the present method accurately predicted the fracture behavior of all the examples considered. Finally, the applicability of the method is presented for crack propagation through nonlinear elastic materials under finite strain deformations.

3 Fluid: Robust numerical integration in embedded interface methods

The overall objective of this thesis is the development of a computational approach for fluid-structure interaction in the presence of cracking structures. The previous chapter was centered towards an FE approach for fracture mechanics. Now, the development of an accurate and robust computational approach to solve flow around a moving obstacle is an essential prerequisite to address fluid-structure-fracture interaction.

Despite several impressive advances in computational mechanics, solving the Navier-Stokes equations around dynamically moving obstacles still offers a number of challenges. Some of the problems are highlighted in section 1.4, and it was pointed out that fixed grid methods can circumvent these problems. The focus of this chapter is to present a fixed grid embedded interface approach developed in [123–126] to solve the Navier-Stokes equation in the presence of moving interfaces. Emphasis is placed on the accurate numerical integration of the associated weak forms, which is one of the crucial steps in dictating the accuracy of fixed grid methods.

This chapter is organized as follows. It first recalls the equations governing the dynamics of fluid flow—the Navier-Stokes equations—together with the relevant continuum mechanics concepts. Then it provides a short introduction to the embedded interface method used in this work. This chapter then proceeds with the FE weak form associated with the Navier-Stokes equations; stabilization and the weak enforcement of boundary conditions are detailed. This is followed by the description of special time integration technique used to obtain accurate solutions in fixed grid methods. One of the main contributions of the present thesis is the development of accurate and robust techniques for weak form integration. In the next section, the existing numerical integration methods are reviewed, and their deficiencies are recalled. After this, the two new numerical integration methods developed in this thesis, namely the generalised moment fitting method and the direct divergence method are explained in detail. Emphasis is placed on the computer implementation of the proposed methods. In the final section of the chapter, several numerical examples are solved to study the performance of integration strategies. These examples demonstrate that the direct divergence method is the most accurate and robust one among all the available methods.

3.1 Navier-Stokes equations: strong form

The governing equations of fluid dynamics are usually written in Eulerian frame of reference because the fluid flow is always associated with large deformations. Let Ω^f be the domain occupied by the fluid which is bounded by its boundary Γ^f . Within this domain, an obstacle is embedded, and the interface between the obstacle and the domain is denoted as Γ^i . The behavior of isothermal fluid flow within the domain is governed by the conservation equations for mass and momentum; together they are referred to as Navier-Stokes equations. This work deals with unsteady incompressible Newtonian fluid for which the conservation equations of mass and

momentum are written as

$$\nabla \cdot \mathbf{u}^f = 0 \quad \text{on } \Omega^f \times (0, T) \quad (3.1a)$$

$$\rho^f \frac{D\mathbf{u}^f}{Dt} - \nabla \cdot \boldsymbol{\sigma}^f = \rho^f \mathbf{b}^f \quad \text{on } \Omega^f \times (0, T) \quad (3.1b)$$

where \mathbf{u}^f is the fluid velocity, ρ^f is the fluid density, $\boldsymbol{\sigma}^f$ is the Cauchy stress tensor, and \mathbf{b}^f is externally applied body force per unit mass. In Eulerian frame of reference, the material derivative of velocity is the sum of the local and convective parts as written below.

$$\frac{D\mathbf{u}^f}{Dt} = \frac{\partial \mathbf{u}^f}{\partial t} + \mathbf{u}^f \cdot \nabla \mathbf{u}^f \quad (3.2a)$$

$$= \dot{\mathbf{u}}^f + \mathbf{u}^f \cdot \nabla \mathbf{u}^f \quad (3.2b)$$

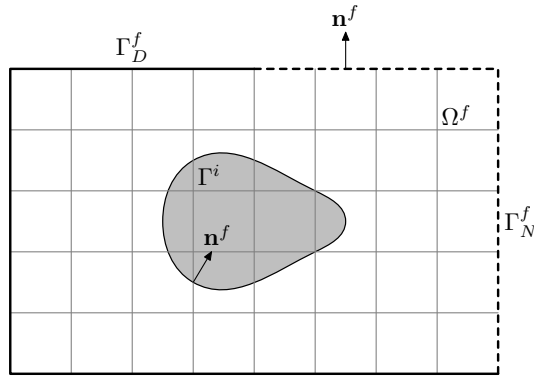


Figure 3.1: A schematic illustration of fluid domain. Shaded area represents embedded object. Continuous and dotted boundaries denote Dirichlet and Neumann portions.

The Cauchy stress tensor is expressed as a sum of two parts, of which one is an isotropic tensor and another is a function of the strain rate tensor $\boldsymbol{\epsilon}^f$

$$\boldsymbol{\sigma}^f = -p^f \mathbf{I} + \boldsymbol{\tau}(\boldsymbol{\epsilon}^f) \quad (3.3a)$$

$$\text{with } \boldsymbol{\epsilon}^f = \frac{1}{2} [\nabla \mathbf{u}^f + (\nabla \mathbf{u}^f)^\top] \quad (3.3b)$$

where p^f is the dynamic pressure. For a Newtonian fluid, the tensor $\boldsymbol{\tau}(\boldsymbol{\epsilon}^f)$ is the linear function of $\boldsymbol{\epsilon}^f$, and is given as $\boldsymbol{\tau}(\boldsymbol{\epsilon}^f) = 2\mu^f \boldsymbol{\epsilon}^f$ where μ^f is the dynamic viscosity of fluid.

From the above expressions, the governing equations (3.1) can be rewritten as

$$\nabla \cdot \mathbf{u}^f = 0 \quad \text{on } \Omega^f \times (0, T) \quad (3.4a)$$

$$\underbrace{\rho^f \dot{\mathbf{u}}^f + \rho^f \mathbf{u}^f \cdot \nabla \mathbf{u}^f}_{\text{convection}} + \nabla p^f - \underbrace{2\mu^f \nabla \cdot \boldsymbol{\epsilon}^f}_{\text{diffusion}} = \rho^f \mathbf{b}^f \quad \text{on } \Omega^f \times (0, T) \quad (3.4b)$$

The presence of convection terms in the above equation introduces a lot of additional complexities, when compared to the structural equations presented in section 2.3. Achieving a stable solution is very challenging, especially when the convection terms dominate the diffusion. This issue will be addressed in section 3.3.

The aforementioned governing equations are specified together with proper initial and boundary conditions. Velocity field throughout the domain is specified as initial conditions.

$$\mathbf{u}^f|_{t=0} = \mathbf{u}_0^f \text{ on } \Omega^f \quad (3.5)$$

To define a well-posed problem the initial velocity field must be divergence-free i.e., $\nabla \cdot \mathbf{u}_0^f = 0$. Initial condition for pressure is not required as it has no associated evolution equation. In the incompressible Navier-Stokes equations, the pressure acts as a Lagrange multiplier to enforce incompressibility condition.

As shown in figure 3.1, the boundary of the domain Γ^f is split into Γ_D^f and Γ_N^f that hold Dirichlet and Neumann boundary conditions respectively. In addition, Dirichlet boundary conditions are specified on the interface Γ^i . The boundary conditions are stated as follows

$$\mathbf{u}^f = \bar{\mathbf{u}}^f \text{ on } \Gamma_D^f \times (0, T) \quad (3.6a)$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^f = \bar{\mathbf{h}}^f \text{ on } \Gamma_N^f \times (0, T) \quad (3.6b)$$

$$\mathbf{u}^f = \bar{\mathbf{u}}^i \text{ on } \Gamma^i \times (0, T) \quad (3.6c)$$

where $\bar{\mathbf{u}}^f$ and $\bar{\mathbf{u}}^i$ are the prescribed velocity values over their corresponding boundaries, $\bar{\mathbf{h}}^f$ is the traction vector, and \mathbf{n}^f is the unit outward pointing normal to the boundary.

3.2 Introduction to embedded interface methods

Conventional FEMs do not allow for the representation of discontinuous fields within an element; a mesh that is aligned along any discontinuity is generated, and the numerical solution can be obtained without any difficulties, using standard polynomial approximation spaces. However, when the discontinuity surface, for example the interface Γ^i in figure 3.1, is dynamic, then these methods require remeshing. The additional complications associated with such procedure are already elaborated in section 1.4. In order to circumvent such complications, methods that do not require an interface-fitted mesh have been developed in recent years. Such methods are termed as embedded interface methods (EIM) which include immersed boundary [127, 128], immersed interface [129, 130] and fictitious domain [131] methods. A common disadvantage of all these methods is that they do not provide an accurate solution in the neighborhood of interface. This shortcoming motivated Gerstenberger & Wall [123, 124] to develop EIMs specifically suited for FSI applications by enabling a sharp interface representation. These methods are later improved by Schott & Wall [126].

As the interface is not directly accounted for in the mesh, its presence has to be explicitly introduced through modifications in the discretization of the governing equations. Different EIMs are classified based on the nature of modifications through which the interface effects are included. The present work follows Schott & Wall [126], that uses cut element based discretization approach, and employs Nitsche's method to enforce boundary conditions on the interface weakly. Let \mathcal{T} denotes a non-interface fitted background mesh covering domain Ω^{ex} in which an interface Γ^i is embedded arbitrarily (figure 3.2). In this work, Γ^i is represented by the trace mesh of an embedded mesh. The interest here is on the physical fluid domain Ω^f that is the area not enclosed by Γ^i . Elements (K) that are partly or completely inside Ω^f are given by $\Omega_{\mathcal{T}} := \{K \in \mathcal{T} | K \cap \Omega^f \neq \emptyset\}$. The set of elements cut by Γ^i are given by $G_h := \{K \in \mathcal{T} | K \cap \Gamma^i \neq \emptyset\}$. Γ^i decomposes cut elements $K \in G_h$ into a number of arbitrary polyhedra, called as volume-cells (V). Such cells falling into the physical fluid domain are called

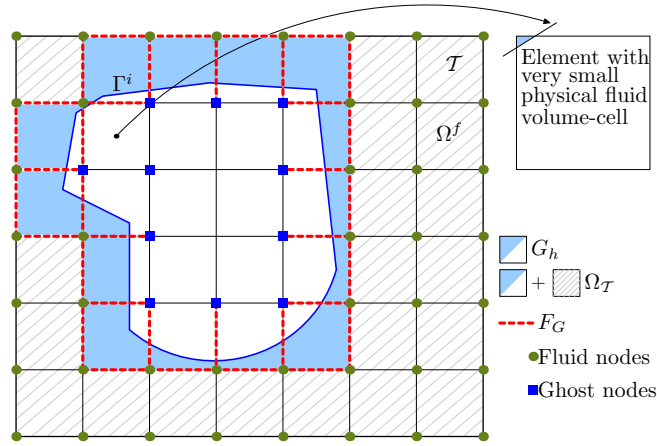


Figure 3.2: Notation used for mesh and volume-cell description.

fluid volume-cells. Then, the physical fluid volume of an element (shaded polyhedra in figure 3.2) is given as $\Omega_K^f = \{V \subseteq K \cap \Omega^f \text{ polyhedra} \mid V_i \cap V_j = \emptyset \text{ for } i \neq j\}$. For non-cut fluid elements $K \in \Omega_T \setminus G_h$, the whole element is considered to be the volume-cell. Based on this notation, one can write the whole physical domain as

$$\Omega^f = \bigcup_{K \in \mathcal{T}} \Omega_K^f = \bigcup_{K \in \mathcal{T}} \bigcup_{V \in \Omega_K^f} V \quad (3.7)$$

Despite the volume-cell approximation, owing to the use of EIM, the interpolations within the volume-cells are still carried out using the standard shape functions of the complete cut elements. It is worthwhile to mention here that while handling thin-interfaces and some special cut situations, more than one set of degrees of freedom (DOF) may have to be assigned to a node. For more details on this issue, refer to Schott & Wall [126] and Shahmiri [28].

3.3 Navier-Stokes equations: weak form

3.3.1 Time discretization

One-step-theta (OST) method is used for time integration, in which the acceleration \mathbf{a} is defined as a combination of the values at time level n and $n + 1$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \theta \mathbf{a}^{n+1} + (1 + \theta) \mathbf{a}^n \quad (3.8)$$

where θ is the parameter defining the behavior of OST, and Δt is the discrete time step. For brevity, the superscript f is omitted from pressure, velocity and density.

The time-discretized Navier-Stokes equations, after applying OST scheme, can be written as

$$\left[\frac{\rho \mathbf{u}}{\Theta} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot \boldsymbol{\tau} - \mathbf{b} \right]^{n+1} - \frac{\rho}{\Theta} \mathbf{u}^{\text{hist}} = 0 \quad (3.9a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (3.9b)$$

where $\Theta = \theta \Delta t$ and $\mathbf{u}^{\text{hist}} = \mathbf{u}^n + \Delta t(1 - \theta)\mathbf{a}^n$. For simplicity, the superscript f is omitted in the above equation. More details on OST scheme can be found in several text books, including Donea & Huerta [132].

3.3.2 Spatial discretization

This section presents the spatial discretization associated with the EIM, independent of the time integration used. It is originally proposed by Schott and Wall [126], the extension of which is also presented for two-phase flows [133] and to embed one mesh into another non-confirming mesh [28, 134].

For brevity, the complete weak form is split into five terms as described below.

$$R_{NS} + R_{STAB}^{resi} + R_{emBC} + R_{STAB}^{edge} = f_{NS} \quad (3.10)$$

where R_{NS} and f_{NS} are the terms arising from the discretization of Navier-Stokes equations, R_{STAB}^{resi} contains residual-based stabilization terms, R_{emBC} includes the terms associated with the weak enforcement of Dirichlet boundary conditions on the embedded interface, and R_{STAB}^{edge} are edge-based stabilization terms. This section provides a brief explanation for the source and significance of each terms in the above equation.

Before proceeding to the weak form, it is essential to denote the appropriate function spaces for velocity and pressure. The discrete trial and test function spaces for velocities are given as

$$\mathcal{S}_h^u = \left\{ \mathbf{u} \in [C^0(\Omega^f)]^3 : \mathbf{u}|_V \in [Q^l(K)]^3 \quad \forall V \in \Omega_K^f, \forall K \in \mathcal{T} \mid \mathbf{u} = \bar{\mathbf{u}}^f \text{ on } \Gamma_D^f \right\} \quad (3.11a)$$

$$\mathcal{V}_h^u = \left\{ \mathbf{v} \in [C^0(\Omega^f)]^3 : \mathbf{v}|_V \in [Q^l(K)]^3 \quad \forall V \in \Omega_K^f, \forall K \in \mathcal{T} \mid \mathbf{v} = 0 \text{ on } \Gamma_D^f \right\} \quad (3.11b)$$

Since there is no Dirichlet boundary condition for pressure, the trial and test spaces are the same, given as

$$\mathcal{Q}_h^p = \left\{ q \in C^0(\Omega^f) : q|_V \in Q^l(K) \quad \forall V \in \Omega_K^f, \forall K \in \mathcal{T} \right\} \quad (3.12)$$

where $Q^l(K)$ is the polynomial of order l in each direction, defining the interpolation within an element K .

Having defined the function spaces, the next step is to multiply the Navier-Stokes equations with appropriate test functions, and integrate the product over the fluid domain. The resulting weak form, after integration by parts, reads as: find $(\mathbf{u}_h, p_h) \in \mathcal{S}_h^u \times \mathcal{Q}_h^p$ such that for all $(\mathbf{v}_h, q_h) \in \mathcal{V}_h^u \times \mathcal{Q}_h^p$, the equation $R_{NS} = f_{NS}$ holds; where,

$$R_{NS} = (\mathbf{v}_h, \rho \dot{\mathbf{u}}_h) + (\mathbf{v}_h, \rho \mathbf{u}_h \cdot \nabla \mathbf{u}_h) + (q_h, \nabla \cdot \mathbf{u}_h) - (\nabla \cdot \mathbf{v}_h, p) + (\epsilon(\mathbf{v}_h), 2\mu\epsilon(\mathbf{u}_h)) \quad (3.13)$$

$$f_{NS} = (\mathbf{v}_h, \rho \mathbf{b}) + \left\langle \mathbf{v}_h, \bar{\mathbf{h}} \right\rangle_{\Gamma_N^f} \quad (3.14)$$

where $(\cdot, \cdot) = (\cdot, \cdot)_{\Omega^f}$ is the usual L^2 -inner product defined over the physical fluid domain. This means that for cut elements, this integral is defined only on the arbitrary polyhedral-shaped fluid volume-cells which offers a lot of complications in the weak form integration as will be dealt with in section 3.5. The above equation represents the unstabilized weak form without considering the presence of the embedded interface. The forthcoming text explains the need and significance of other terms in equation (3.10)

Residual-based stabilization:

The need for stabilization in FEM for fluids arises due to two reasons: appearance of convective terms in the governing equations because of the use of Eulerian reference frame, and the incompressibility constraint. Since neither of these two are present in the structural equations presented in the last chapter, stabilization was not needed. Here, the widely used residual based stabilization is described [132, 135–140] that contains the following three components.

$$R_{STAB}^{resi} = \underbrace{(\rho \mathbf{u}_h \cdot \nabla \mathbf{v}_h, \tau_M \mathbf{r}_{M,h})}_{\text{SUPG}} + \underbrace{(\nabla q_h, \tau_M \mathbf{r}_{M,h})}_{\text{PSPG}} + \underbrace{(\nabla \cdot \mathbf{v}_h, \tau_C r_{C,h})}_{\text{LSIC}} \quad (3.15)$$

where $\mathbf{r}_{M,h}$ and $r_{C,h}$ are the residual of momentum and continuity equations respectively.

$$\mathbf{r}_{M,h} = \rho \frac{\partial \mathbf{u}_h}{\partial t} + \rho \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla p_h - 2\mu \nabla \cdot \boldsymbol{\epsilon}(\mathbf{u}_h) - \rho \mathbf{b} \quad (3.16a)$$

$$r_{C,h} = \nabla \cdot \mathbf{u}_h \quad (3.16b)$$

The corresponding stabilization parameters are defined as presented in [141, 142]

$$\tau_M = \frac{1}{\sqrt{\left(\frac{2\rho}{\Delta t}\right)^2 + \rho \mathbf{u}_h \cdot \mathbf{G} \rho \mathbf{u}_h + C_I \mu^2 \mathbf{G} : \mathbf{G}}} \quad (3.17a)$$

$$\tau_C = \frac{1}{\tau_M \text{tr}(\mathbf{G})} \quad (3.17b)$$

where $G_{kl} = \sum_{d=1}^3 (\partial \xi_d / \partial x_k) (\partial \xi_d / \partial x_l)$ utilizes the natural coordinate system $\boldsymbol{\xi}$ of the parent element, Δt is the time step, C_I is set to be 36.0 and 144.0 for linear and quadratic hexahedral elements respectively.

Unless the focus is on simulating creeping flows, the convection terms in the Navier-Stokes equations (equation (3.4)) dominate the diffusion. In this case, the standard Galerkin methods are inherently unstable, and lead to non-physical oscillations and wiggles in the solution field. The role of streamline/upwind Petrov-Galerkin (SUPG) is to add, in a consistent manner, additional dissipation or equivalently to introduce upwinding into the discretization without affecting the solution accuracy.

Another source of instability arises from the incompressibility of Navier-Stokes equations: pressure appears in momentum equations, but there is no evolution equation for pressure. It acts as a Lagrangean multiplier to enforce incompressibility constraint. As a result of this, the discretized matrix system will contain zero submatrix on the diagonal. Such a saddle-point system is solvable only if the velocity and pressure solution spaces are chosen to satisfy the well-known inf-sup or Ladyzhenskaya-Babuška-Brezzi (LBB) condition [143]. However, in our formulations, equal order pressure and velocity interpolation functions are chosen, and circumvent the necessity of satisfying the LBB condition by introducing pressure stabilizing Petrov-Galerkin (PSPG) terms. While convective instability dominates only at sufficiently high Reynolds number (Re), this instability is present at all situations.

The least-squares incompressibility constraint (LSIC), as the name implies, is a stabilization procedure of least-squares type. This term is adopted from Hansbo [144], and as shown by Linke [145], this term provides an enhancement of discrete mass conservation.

The presence of second-order derivatives in the stabilization terms entail additional continuity requirements. In order to avoid this problem, the stabilization terms are assumed to act only in the interior of the elements.

R_{NS} and R_{STAB}^{resi} are present even in conventional FEMs for fluid flow. This means that until now, the influence of Γ^i into the discretization is not introduced.

Weak enforcement of boundary conditions:

When Dirichlet boundary conditions are applied strongly, as in Γ_D^f , the trial functions satisfy the Dirichlet conditions explicitly (equation (3.11)a), and the test functions vanish on Γ_D^f (equation (3.11)b). In contrast, since the interface Γ^i is not aligned with the background mesh, the Dirichlet boundary conditions have to be enforced weakly. R_{emBC} describes additional terms that are included in the weak form to enable weak imposition of boundary conditions on the embedded interface Γ^i .

$$\begin{aligned} R_{emBC} = & + \langle \mathbf{v}_h, p_h \cdot \mathbf{n} \rangle_{\Gamma^i} - \langle \mathbf{v}_h, 2\mu\epsilon(\mathbf{u}_h) \cdot \mathbf{n} \rangle_{\Gamma^i} \\ & - \langle q_h \cdot \mathbf{n}, \mathbf{u}_h - \bar{\mathbf{u}}^i \rangle_{\Gamma^i} - \langle \beta \cdot 2\mu\epsilon(\mathbf{v}_h) \cdot \mathbf{n}, \mathbf{u}_h - \bar{\mathbf{u}}^i \rangle_{\Gamma^i} \\ & + \sum_{K \in G_h} \sum_{V \in \Omega_K^f} \langle \gamma_u \cdot \mathbf{v}_h, (\mathbf{u}_h - \bar{\mathbf{u}}^i) \rangle_{\Gamma_K \cap V} \end{aligned} \quad (3.18)$$

where Γ_K is the part of interface that lies in element K . The terms on the first line, called as standard-consistency terms, result from the partial integration of pressure and viscous Galerkin terms. They are retained because of non-vanishing test functions i.e., $\mathbf{v}_h|_{\Gamma^i} \neq 0$, and these terms remain irrespective of the method of choice for weak boundary condition imposition.

This work, as proposed in [126], utilizes Nitsche's method [146, 147], and the second line of equation (3.18) are added to enforce $\mathbf{u}_h - \bar{\mathbf{u}}^i = 0$ weakly on Γ^i . It is possible to choose $\beta = 1$ or $\beta = -1$ without affecting the consistency. The anti-symmetric formulation with choice $\beta = -1$ results in suboptimal convergence or larger L^2 -errors [148, 149], though it leads to better stability behavior. Hence the symmetric variation with $\beta = 1$, also called as adjoint-consistent formulation, is chosen.

The choice $\beta = 1$, unfortunately, leads to lack of coercivity and the terms described in the last line of equation (3.18) are mandatory to account for this. Including this term ensures a non-singular system of equations. The choice of stabilization parameter in this term is given by [126, 133, 134]

$$\gamma_u := \max(\gamma_v, \gamma_{conv}, \gamma_t) := \max\left(\alpha\mu h_K^{-1}, \frac{\rho\|\mathbf{u}_h\|_\infty\alpha}{6}, \frac{\rho h_K\alpha}{12\theta\Delta t}\right) \quad (3.19)$$

which takes into account a viscous part γ_v to balance the lack of coercivity, a convective part γ_{conv} for convective instabilities, and a time dependent part γ_t in case of unsteady simulations. In the above formula, α is a stabilization parameter and h_K is the characteristic element length scale.

Alternative to Nitsche's method, it is also possible to use interface Lagrange multiplier method [150–152] or the boundary conditions can be weakly enforced by introducing an volumetric Lagrangean multiplier that involves an additional element-wise discontinuous flux field [124, 153, 154].

Edge-based stabilization

Shapes of the volume-cells obtained for an element depend on how the element is cut by the interface. In a moving interface simulations, peculiar cut configurations can result in tiny volume-cells as depicted in figure 3.2. Residual-based stabilization terms alone are not suffi-

cient in such cases, since such terms involve integrals over the volume-cells. As the size of the volume-cells reduces, the residual-based stabilization terms tend to zero, and hence additional means of stabilization is needed to obtain oscillation-free solutions. This work makes use of edge-based stabilization which involves adding terms to penalize the jumps of gradients across element boundaries i.e., mesh edges in the interface region. It has three contributions as stated below.

$$R_{STAB}^{edge} = j_{GP}(\mathbf{v}_h, \mathbf{u}_h) + j_{stream}(\mathbf{v}_h, \mathbf{u}_h) + j_p(q_h, p_h) \quad (3.20)$$

The term $j_{GP}(\mathbf{v}_h, \mathbf{u}_h)$ overcomes the stability problems related to Nitsche's method on cut elements [155], and is defined as

$$j_{GP}(\mathbf{v}_h, \mathbf{u}_h) = \sum_{F \in F_G} \sum_{i=1}^k \int_F \left(\alpha_{GP,\mu} \cdot \mu + \alpha_{GP,t} \frac{\rho}{\theta \Delta t} h_F^2 \right) h_F^{2(i-1)+1} \llbracket D^i \mathbf{v}_h \rrbracket : \llbracket D^i \mathbf{u}_h \rrbracket ds \quad (3.21)$$

where F_G contains all element surfaces shared by two volume-cells belonging to different elements, say K_1 and K_2 among which at least one is a cut element (figure 3.2). $\alpha_{GP,\mu}$ and $\alpha_{GP,t}$ imply viscous and transient ghost-penalty parameters respectively, h_F is the characteristic length scale of element's surface and $\llbracket v \rrbracket = v|_{K_1} - v|_{K_2}$ denotes jump in a quantity across the face shared by two adjacent elements, and k is the highest polynomial degree in \mathbf{v}_h .

$j_{stream}(\mathbf{v}_h, \mathbf{u}_h)$ and $j_p(q_h, p_h)$ are edge-based stabilization versions of SUPG and PSPG respectively. $j_{stream}(\mathbf{v}_h, \mathbf{u}_h)$ controls convective instabilities, similar to SUPG. It involves penalizing the jumps of velocity gradients along the edges [156–158], and is given as

$$j_{stream}(\mathbf{v}_h, \mathbf{u}_h) = \sum_{F \in F_G} \int_F \gamma_{stream} h_F^2 \llbracket \nabla \mathbf{v}_h \rrbracket : \llbracket \nabla \mathbf{u}_h \rrbracket ds \quad (3.22)$$

where the convection stabilization parameter is defined as $\gamma_{stream} = \alpha_{stream} \cdot \rho |\mathbf{u}_h \cdot \mathbf{n}_F|$. To avoid adding diffusion in the direction normal to the flow, the streamline nature of flow is taken into consideration through the term $|\mathbf{u}_h \cdot \mathbf{n}_F|$.

In order to enable the use of inf-sup unstable pressure-velocity pairs, pressure gradient based stabilization term $j_p(q_h, p_h)$ is introduced [155, 159]

$$j_p(q_h, p_h) = \sum_{F \in F_G} \int_F \gamma_p h_F^{2i} \llbracket D^i q_h \rrbracket : \llbracket D^i p_h \rrbracket ds \quad (3.23)$$

where the pressure stabilization parameter is defined as,

$$\gamma_p = \alpha_p \cdot \left(\frac{\mu}{h_F} + \frac{\rho \|\mathbf{u}_h\|_\infty}{6} + \frac{\rho h_F}{12\theta \Delta t} \right)^{-1} \quad (3.24)$$

The terms $j_{stream}(\mathbf{v}_h, \mathbf{u}_h)$ and $j_p(q_h, p_h)$ help to introduce physical information from fluid domain to velocities and pressure at ghost nodes (see figure 3.2), respectively.

The stabilization parameters are chosen to be $\alpha_{GP,\mu} = \alpha_{stream} = \alpha_p = 10$ and $\alpha_{GP,t} = 0.05$ from error analysis presented in [126] and further extensive numerical experiments.

It is to be remarked that the use of edge-based stabilization requires additional data structure to hold the information about the faces; moreover it increases the bandwidth of the resulting matrix system which lead higher computational costs. However, in order to obtain robust and accurate embedded interface formulations, such stabilization is mandatory.

3.4 A note on time integration

When the embedded interface moves within the background mesh from time step t^n to t^{n+1} , there are few nodes that were in the solid domain at t^n will be injected to the fluid domain at t^{n+1} (figure 3.3). These nodes, referred to as injected-nodes, offer some difficulties for the time integration, since values of \mathbf{u}^{hist} given in equation (3.9) are not available at these nodes. This section expresses two possible cases, and explains how these are dealt with.

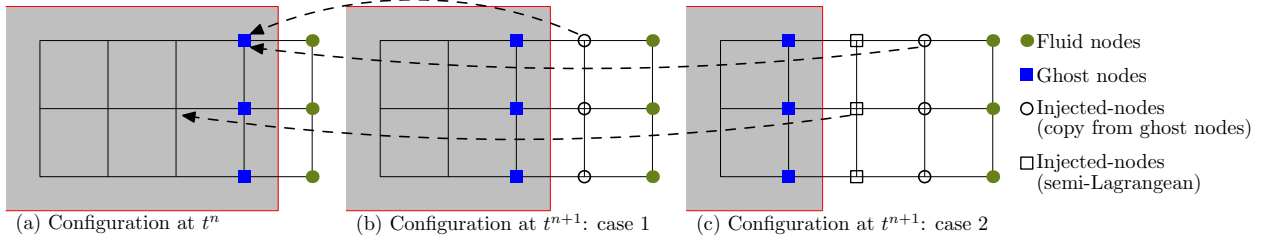


Figure 3.3: Interface movement and semi-Lagrangian time integration

Case 1: Interface does not cross one complete element. In this case, the extent of movement of the interface is restricted not to cross one complete background element. The injected-nodes here were the ghost nodes at t^n (figure 3.3), and \mathbf{u}^{hist} values are directly computed from the solution available at ghost nodes.

Case 2: Interface traverses one or more complete elements. In this case, only for some of the injected-nodes, the ghost values can be copied (figure 3.3). Others were completely within the solid at t^n , and in order to construct \mathbf{u}^{hist} values for these nodes, the semi-Lagrangian time integration proposed by Henke et. al. [160] is used. This procedure involves tracking back the Lagrangean origin of these nodes at t^n which, in principle, must be located in the fluid domain at t^n . The field values at these Lagrangean origins are utilized to construct \mathbf{u}^{hist} to perform time integration using OST. For more details on the semi-Lagrangian method, refer to [160].

In both scenarios, the initial conditions for the ghost nodes at t^{n+1} are constructed by solving a small system of equations. This equation system is derived using equation (3.21), requiring that the solution field is kink-free and continuous over the cut element. Once the history values are constructed at all the nodes, the time integration using OST scheme can be carried out without further difficulties.

3.5 Numerical integration of weak forms

In order to extract the stiffness matrix represented by the FE weak form given in equation (3.10), it is essential to accurately evaluate all the integrals in these equations. In a standard FEM, this integration is accomplished in a straightforward manner using the Gauss quadrature rule which is stated as follows.

Gauss quadrature rule: Let $f(\mathbf{x})$ be the integrand which is defined and continuous over Ω , and $\omega(\mathbf{x})$ be the weight function, then the quadrature approximates the integral over Ω in the

following form.

$$\int_{\Omega} \omega(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^N w_i f(\mathbf{x}_i) \quad (3.25)$$

where \mathbf{x}_i and w_i denote the location and the weights of quadrature points, respectively, and N is the number of points necessary for the approximation to be sufficiently accurate.

In a standard FEM or in uncut elements in embedded interface methods, the integration domain $\Omega = (\Omega_K^f)$ is the whole element that has a predefined simple shape: either a hexahedron, tetrahedron, pyramid, wedge or a prism. For these elements, the location \mathbf{x}_i and the weights w_i of the Gauss quadrature rule are readily available with which one can easily evaluate the weak form integrals using equation (3.25).

In EIMs, the interface which is embedded within the non-body conforming FE mesh cuts few elements of the background mesh and as a result split the element into two or more non-overlapping arbitrary complex shaped polyhedra. As already stated, the elements that are cut by the interface are called cut elements, and the polyhedra produced by the interface cut are referred to as volume-cells. An example of such a cut configuration is illustrated in figure 3.4. These cut elements offer significant difficulties: in order to extract the stiffness matrix for these cut elements, the weak form has to be integrated over the polyhedral-shaped volume-cells. Gauss quadrature can't be used because such rules are available only for simple shaped volumes, not for arbitrary polyhedra.

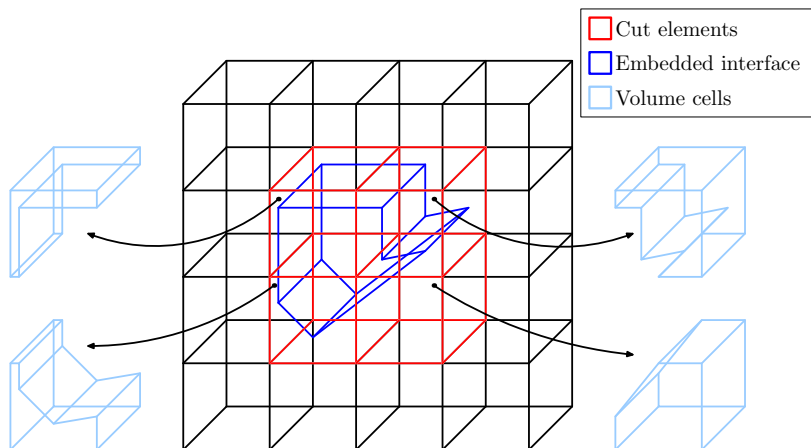


Figure 3.4: Interface cut and the resulting complex polyhedral shaped volume-cells for a small embedded interface method example

Accuracy of the weak form integration over the cut elements largely influences the overall solution accuracy in EIMs. Moreover, in transient FSI and multiphase flow problems, since the interface changes its position and shape with time, the geometrical cutting operations and the construction of integration method for the cut elements are performed at each time step. The inapplicability of standard Gauss quadrature rule in the aforementioned situations, together with the importance of accurate weak form integration have motivated the researchers to develop various integration strategies that can be used directly in EIMs.

One of the essential points to be mentioned here is that in FE methods the integrand is not available as an explicit mathematical function. It is constructed during the solution process, and can be interpolated at any point within the domain. Hence, while developing integration methods

aiming at weak form integration in FE methods, this has to be bear in mind. The following section reviews the available methods which can handle integrands that are not explicitly prespecified.

3.5.1 Review of existing integration methods

A detailed review of the available numerical integration strategies used in the embedded interface methods is presented here; emphasis is placed on robustness and accuracy of each methods. Only the methods that are appealing for FEM implementation and are relevant to the present work are reviewed here. The existing integration methods can be grouped into either one of the following categories.

- Tessellation [65, 66, 123, 161–163]
- Adaptive quadrature [164–170]
- Conformal mapping [171, 172]
- Equivalent polynomial approach [173–175]
- Moment fitting method [76, 169, 170, 176–180]
- Methods based on the divergence theorem [181–192]

Apart from these methods, there are different methods that smooth the discontinuity over a distance [193–196], and other methods addressing how to deal with singular integrands [71–77]. Since the focus here is on sharp representation of the interface which is crucial for the accuracy of FSI problems, and only the usual non-singular polynomial shape functions are used, they are not reviewed here.

For the following discussion, let Ω represents the cut element, and \mathcal{R} denotes a volume-cell. \mathcal{R}^f and \mathcal{R}^s represent volume-cells that are located on physical fluid and solid part respectively.

3.5.1.1 Tessellation

The most widely used method for integration over the volume-cells is the tessellation approach [65, 66, 123, 161–163]. This approach involves a volume decomposition process in that the complex shaped volume-cells are decomposed into a number of simple shaped subvolumes. The subvolumes are referred to as integration-cells, and are generally of tetrahedral or hexahedral shaped for which the Gauss quadrature rules are readily available. It is ensured that the integration-cells are aligned along the interface so that the necessity to integrate a discontinuous function within them is circumvented. Figure 3.5 shows a representative example of the tessellation process. These integration-cells are used only to perform weak form integration over the volume-cells. It does not affect the approximation property of FEM nor does it introduce additional nodal variables. Moreover, since integration-cells are not used to form solution spaces, the aspect ratio of such cells does not influence the solution accuracy [162].

The volume decomposition process associated with tessellation is very complicated to implement within the FE framework, and hence external packages such as qhull [197] or TetGen [198] are often used to accomplish this. Once the volume decomposition is successfully carried out, the integration rule for a volume-cell is constructed by collecting all the integration-cells that belong to this volume-cell. This involves two mapping steps, as shown in equation (3.26).

$$(\tilde{\mathbf{x}}, \tilde{w})_{\xi_{in}} \xrightarrow{F_1} (\bar{\mathbf{x}}, \bar{w})_{\mathbf{x}_{in}} \xrightarrow{F_2^{-1}} (\mathbf{x}, w)_{\xi_e} \quad (3.26)$$

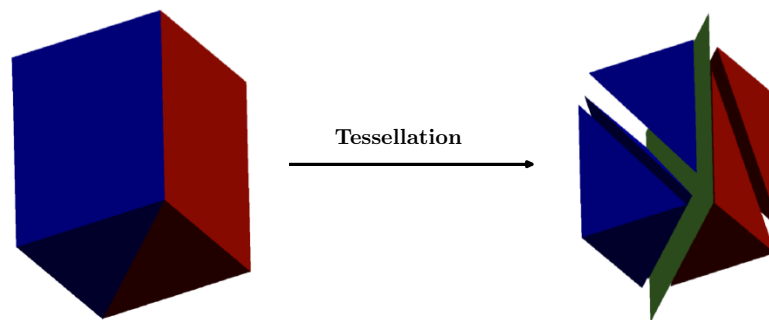


Figure 3.5: Tesselation approach. Red and blue shapes on the left figure indicate volume-cells, and the green surface on the right shows the interface.

where ξ_{in} and \mathbf{x}_{in} are the natural- and the physical-coordinate system of the integration-cells respectively, and ξ_e is the natural coordinate system of the corresponding cut element.

In the first mapping F_1 , the quadrature rule available in the natural coordinate system of integration-cells $(\tilde{\mathbf{x}}, \tilde{w})$ is mapped into the physical integration-cell coordinates, and is denoted as $(\bar{\mathbf{x}}, \bar{w})$. This integration rule is, using another mapping F_2^{-1} , transformed into the required integration rule defined with respect to natural coordinate system of the background element.

Though tesselation is working well for many cases, the time required to split the volume-cell into integration-cells is very high. Moreover, in several instances, the volume decomposition process associated with tesselation breaks down for some complex cut configurations. In other words, tesselation lacks robustness. This is particularly a big disadvantage, when simulating moving or deforming interface problems, since the tesselation has to be performed at each time step to account for the change in position or shape of the interface.

Tesselation seems to be the most intuitive method when one aims at integration over arbitrary volumes. However, due to the complicated geometrical cutting operations, and the associated robustness issues, it is not the ideal choice for weak form integration in EIMs. In fact, most of the methods devised in the literature are centered towards eliminating the volume subdivision.

3.5.1.2 Adaptive quadrature

The adaptive quadrature method [164, 165, 169, 170] also involves a decomposition process; but instead of the volume-cells as in the tesselation, the cut elements are split into integration-cells. As mentioned in [169], this method involves three steps: an approximate estimation of local integration using a quadrature rule, computation of local integration error, and partition of integration domain into smaller divisions of same shape. These steps are repeated until the error in the integration is less than the predefined value. In other words, the decomposition is carried out recursively until the given error criterion is met. Figure 3.6a illustrates an example in which the elements cut by the interface are recursively split four times. The accuracy of these methods are studied in depth in [165], in which the adaptive integration method is used in an XFEM based n -phase flow computations. It has been shown that even in two-dimensional problems, to obtain relative error in the order of 10^{-6} , more than 20,000 points is necessary. This method is extremely expensive owing to the fact that the integration-cells, unlike tesselation, do not conform to the interface and hence an error is introduced into the integration. To keep this error negligible, the size of the integration-cells is required to be very small and thus results in more number of quadrature points than in the tesselation procedure.

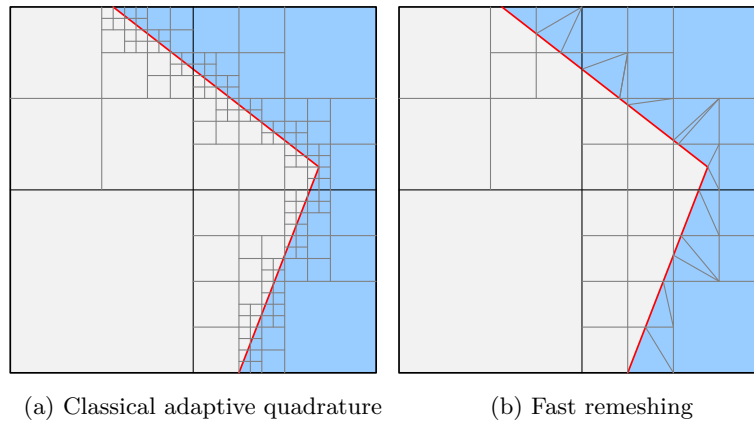


Figure 3.6: Adaptive quadrature (a) the splitting is performed upto 4 levels (b) 2 levels of adaptive splitting and then confirming subcells are formed

A combination of tessellation and adaptive quadrature has been proposed in [199] for GFEM, and in [166–168] for two-dimensional non-Newtonian flow simulations. This method is termed as fast remeshing approach in [199] which consists of two steps described as follows. Adaptive quadrature step: In this step, using quadtree subdivision algorithm, all Quad fluid elements that are intersected by the interface are subdivided into many Quad subcells. Tessellation step: after adaptive subdivision, all the quadrilateral (Quad) subcells that are intersected by the interface are further triangulated in such a way that the resulting Tri elements are aligned along the interface. So in this method, no discontinuous function is integrated over the subcells, and hence this method is expected to be more efficient than the adaptive quadrature method. Figure 3.6b depicts an example in which before performing tessellation step, the cut elements are recursively split two times. However, extension of the method to 3D is not yet reported.

3.5.1.3 Conformal Mapping

Aimed at weak form integration on polygonal FEM, an integration method based on conformal mapping is developed in [171]. Schwarz-Christoffel mapping is used to map a polygon into a unit disk. The integration rule on a polygon is obtained by mapping the quadrature rule on the unit disk to the polygon with the help of Schwarz-Christoffel inversion formula. This approach is shown to be applicable for both convex and concave shaped polygons. The method proposed in [171] is applied in XFEM based fracture mechanics simulations, to integrate the weak form over the cut elements [172]. It is shown to be slightly more efficient than tessellation.

3.5.1.4 Equivalent polynomial approach

Ventura [173] has proposed a method which performs integration of both strongly and weakly discontinuous functions over the cut elements without resorting to any volume decomposition process. This is accomplished by replacing the discontinuous function by an equivalent polynomial which, when integrated over the cut element, gives the value of integration of the discontinuous function. To illustrate the essence of this method, consider a function $H(\mathbf{x}) \cdot f(\mathbf{x})$, where $H(\mathbf{x})$ is a Heaviside step function which is discontinuous across the interface, and $f(\mathbf{x})$ is a polynomial. Though $H(\mathbf{x}) \cdot f(\mathbf{x})$ is discontinuous on Ω , it is continuous on both volume-cells, \mathcal{R}^f and \mathcal{R}^s .

The method aims at deriving an equivalent polynomial $\check{H}(\mathbf{x})$ such that

$$\int_{\Omega} \check{H}(\mathbf{x}) \cdot f(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{R}^f} H(\mathbf{x}) \cdot f(\mathbf{x}) d\mathbf{x} + \int_{\mathcal{R}^s} H(\mathbf{x}) \cdot f(\mathbf{x}) d\mathbf{x} \quad (3.27)$$

Since the modified integrand is a polynomial and is continuous over the whole cut element, the standard Gauss integration rules can be used. In [173] equivalent polynomial is developed for both weak and strong discontinuities when the interface crosses the element completely. It is shown that the equivalent polynomial $\check{H}(\mathbf{x}) \cdot f(\mathbf{x})$ has higher polynomial order than $H(\mathbf{x}) \cdot f(\mathbf{x})$, and hence higher order quadrature is required. This is not an issue when comparing the advantage that the volume decomposition can altogether be neglected.

The method proposed in [173] is used to solve XFEM based plasticity computations in [174]. This study made use of linear elements in 2D and 3D. It demonstrates that despite the equivalent polynomial approach requires much less integration points than tessellation, the convergence rate remains unaffected for both elasticity and plasticity simulations. This study constructs the equivalent polynomials using a numerical approach during which it assumes that the quadrature rules for integration-cells are available. This is a disadvantage, because the fundamental motivation of the equivalent polynomial approach is to eliminate the generation of integration-cells.

The main difficulty of extending [173] to tackle hexahedral elements is that when an interface cuts such elements, the resulting volume-cells, \mathcal{R}^f and \mathcal{R}^s , are arbitrary polyhedra. Hence an analytical evaluation of r.h.s of equation (3.27) is impossible. In a recent work [175], this has been addressed by replacing $H(\mathbf{x})$ with its regularized version that is analytically integrated over the whole cut element Ω to get r.h.s of equation (3.27). This method seems to be able to construct quadrature rules over all element shapes when a straight interface completely intersects the element.

3.5.1.5 Moment fitting method

The moment fitting method is a promising approach to construct quadrature schemes over complex 2D and 3D shapes. In this method, the following equations, known as moment fitting equations [200–202] are solved to obtain quadrature point locations and weights.

$$\begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_N) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_N) \\ \vdots & & & \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} \int_{\mathcal{R}} \omega(\mathbf{x}) \phi_1(\mathbf{x}) d\mathbf{x} \\ \int_{\mathcal{R}} \omega(\mathbf{x}) \phi_2(\mathbf{x}) d\mathbf{x} \\ \vdots \\ \int_{\mathcal{R}} \omega(\mathbf{x}) \phi_m(\mathbf{x}) d\mathbf{x} \end{bmatrix} \quad (3.28)$$

where \mathbf{x} and w are the location of quadrature points and their weights respectively, $\{\phi_1, \phi_2, \dots, \phi_m\}$ are the predefined base functions, N is the number of quadrature points, and m is the number of base functions. The base functions are chosen in such a way that with the combinations of base functions, one can construct the integrand of interest. The fundamental idea behind the method is that since the base functions are explicitly defined, it can be easily integrated over the integration domain \mathcal{R} to obtain the right hand side of equation (3.28). Once this is done, since usually $m \neq N$, the resulting nonlinear equations are solved using Newton's least squares method to obtain quadrature point locations and their weights.

Moment fitting equations are solved to obtain quadrature rules over triangles and squares to integrate polynomial degrees of upto 50 in [177]. The procedure utilizes a node elimination algorithm to eliminate the integration points that are not highly contributing. It is reported that the quadrature rules generated using this method have minimal number of integration points when

compared to all other available works. This method has been extended to deal with arbitrary convex and concave polygons [178]. The generated quadrature rules have desirable properties of interiority of nodes and positivity of weights, and are used to construct mass and stiffness matrices in a polygonal FEM. The method devised in [178] is used together with a special transformation technique for the weak form integration in an XFEM based fracture mechanics solver [76]. In all these works, nonlinear least squares problem is solved to obtain both quadrature point position and their weights. Without explicitly referring to moment fitting equations, this method is used in [176] to construct integration schemes over cut elements that possess strong discontinuity. Quadrature rules are derived for triangular and tetrahedral elements to integrate quadratic polynomials multiplied by a Heaviside or Dirac delta function.

The first use of moment fitting equations in 3D is reported in [169], wherein the base functions are integrated using Lasserre's method [203, 204]. The location of integration points are fixed by distribution of points within the domain of interest, and the weights are obtained by solving the linear system of equations. The resulting quadrature is applied to integrate polynomials over irregular volumes. Owing to the use of Lasserre's method, the applicability of this procedure is limited to convex volumes. This method is used in [179] with application to magneto-mechanical problems, where a procedure is used to ensure that all volume-cells are convex shaped.

3.5.1.6 Methods based on the divergence theorem

Various researchers have used the divergence theorem of multivariable calculus, stated as follows, to perform integration over arbitrary polygons and polyhedra. Let \mathcal{R} be the region in \mathbb{R}^3 bounded by the closed surface \mathcal{S} . Let $\hat{\mathbf{n}}$ denote the outward pointing unit normal of \mathcal{R} on \mathcal{S} then, the divergence theorem states that, for any vector field \mathbf{F} defined on \mathcal{R}

$$\int_{\mathcal{R}} \nabla \cdot \mathbf{F} \, dV = \int_{\mathcal{S}} \mathbf{F} \cdot \hat{\mathbf{n}} \, dA \quad (3.29)$$

As is evident from equation (3.29), this theorem can be used to simplify the domain integral to integral along the boundaries of the domain that can be easily evaluated. In order to directly utilize this theorem, almost all the available methods of this kind either assume that the integrand of interest is predefined [181–186], or rely on symbolic computations to integrate generalized functions [187, 188]. There are severe arguments against using methods based on these assumptions in a general finite element framework, since

- the integrand is not explicitly available in FEM (but can be computed at any point within the domain of interest)
- the FE framework targets large scale problems, and coupling such efficiently tuned codes (usually written in C++) with symbolic computation packages significantly slows down the execution speed.

In recent works [189, 191], aimed at efficient integration in boundary element methods, the researchers eliminate the symbolic computation by using radial basis functions or Gauss quadrature. A similar procedure for 2D polygons is described also in [190]. However, [189, 190] deal only with polygons, and in [191] the method is used for integration over simple volumes only. These references do not address the integration over complex volumes, and more importantly the efficiency of these methods is not reported. For a coherent presentation, detailed explanation of these methods is shifted to section 3.5.3.1.

In summary, each of the aforementioned available methods has its own shortcomings: the volume decomposition process involved in tessellation is highly complicated to implement and usually lacks robustness, the adaptive quadrature is extremely expensive, the conformal mapping is currently available only for two-dimensional computations, the equivalent polynomial approach have not yet been demonstrated to handle arbitrary cut configurations, moment fitting methods are able to handle only convex shapes, and the existing methods based on the divergence theorem cannot be used in a general FEM framework.

In this work, two new integration methods aiming for robust and accurate integration over arbitrary polyhedra are proposed. In the first, the moment fitting method proposed in [169] is extended to deal with both convex and concave volumes. Then, another approach is proposed in which the divergence theorem is used directly to integrate the FE weak form. The necessity of symbolic computation is eliminated with the help of one-dimensional Gauss quadrature, and hence the method can be used in a general FE framework. The following text details these two methods.

Throughout the present work it is assumed that the volume-cells are bounded by plane surfaces which are in turn bounded by straight lines. This is not a severe limitation because most of the time the geometry in FEM is approximated by straight line segments. An extension of the method to higher order interpolation functions or isogeometric analysis can be easily pursued in the future.

3.5.2 Generalized moment fitting method

This section presents a new integration method in which the moment fitting equations are solved to construct quadrature rule over arbitrary polyhedra. The method is the extension of [169], and is generalized in the sense that it can tackle both convex as well as concave shapes using a single algorithm; it does not even require to decide whether the volume of interest is convex or concave. The material presented in this section is based on [180].

Quadrature schemes to integrate 6th order polynomial are constructed, hence the base functions consist of all the monomials until order 6, i.e. $\phi = \{x^i y^j z^k, i + j + k \leq 6\}$. In total, there are 84 base functions. Similar to [169], the position of the integration points are predefined, and the resulting equation system arising from equation (3.28) is solved using a linear least squares method. The complete description of the quadrature construction method necessitates a detailed explanation for the following three steps: integration of predefined base functions, distribution of quadrature points over arbitrary polyhedra, and the solution of the resulting algebraic system of equations using a linear least squares method.

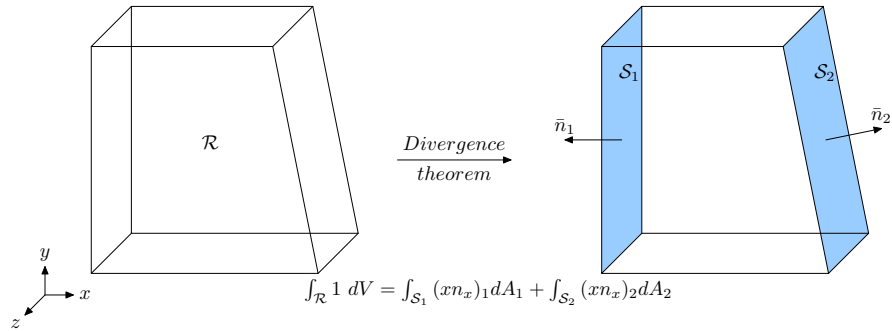
The details of these 3 steps, together with computer implementation aspects are presented in the forthcoming text.

3.5.2.1 Integration of base functions

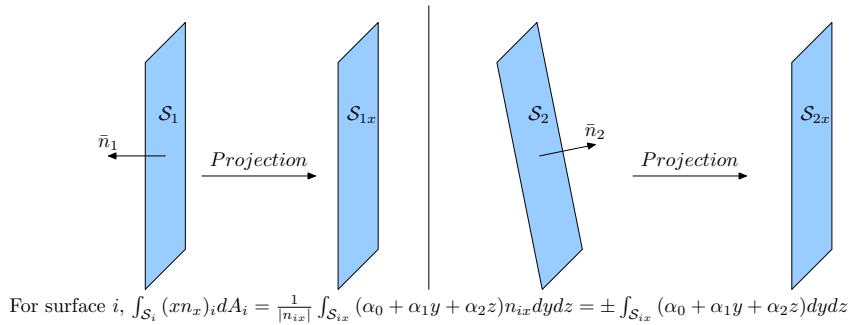
In the present formulation, the divergence theorem of multivariable calculus is used to integrate the base functions within the complex convex/concave volumes over which the quadrature is constructed. This step is essential to compute the r.h.s. of equation (3.28).

The integration of a polynomial, using the divergence theorem, over three dimensional regions is explained in [183, 188]. Following a similar methodology, the integration of the base function over \mathcal{R} involves three steps. In the first step, the divergence theorem is applied to convert the volume integral into surface integrals. Next, the surfaces of the volume which lie in an arbitrarily oriented plane are projected into the appropriate coordinate plane. Finally, the surface integral is reduced to a contour integral by applying the divergence theorem once again. The values of

Step 1: Conversion of a volume integral to surface integrals



Step 2: Projection of arbitrarily oriented plane to coordinate plane



Step 3: Conversion of a surface integral to contour integrals

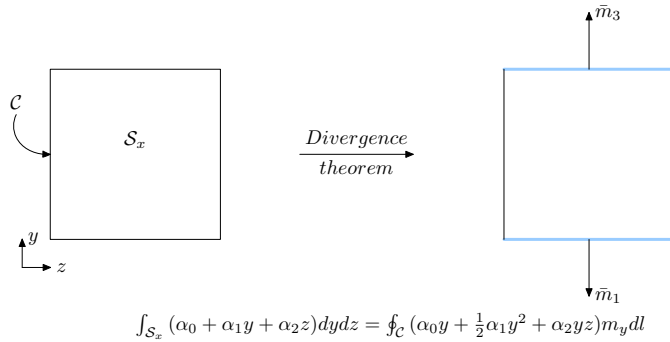


Figure 3.7: Integration of a base function using the divergence theorem

the contour integral in this step are evaluated by using the traditional one dimensional Gaussian quadrature rules. These steps are explained in detail in this section, and an example of base function integration over a representative volume-cell is illustrated in Figure 3.7.

Step 1: Conversion of volume integral to surface integrals

The integral over the volume-cell is decomposed into a sum of integrals over its facets using the well-known divergence theorem stated in equation (3.29). Since, the integrand of interest is a scalar function, it is necessary to construct an equivalent vector field so that the divergence

theorem can be directly used. This is a very simple and straightforward step. In the example presented in Figure 3.7, it is of interest to evaluate $\int_{\mathcal{R}} 1 dV$ over the volume denoted by \mathcal{R} . To convert the volume integral to surface integrals using equation (3.29), a vector field \mathbf{F} is to be found such that $\nabla \cdot \mathbf{F} = 1$. As stated in [183], there can be a number of possible choices for \mathbf{F} which includes $\{x, 0, 0\}$, $\{0, y, 0\}$, $\{0, 0, z\}$. Then, the divergence theorem leads to

$$\int_{\mathcal{R}} 1 dV = \int_{\mathcal{S}} xn_x dA = \int_{\mathcal{S}} yn_y dA = \int_{\mathcal{S}} zn_z dA \quad (3.30)$$

Among all possible choices of \mathbf{F} , the above three are advantageous because two of its components are zero, and hence the dot product with the normal vector gives only one term. In our implementation, the \mathbf{F} whose x -component is non-zero is always used. This implies that only the surfaces whose normal component in x -direction is non-zero are considered for the integration process. For the example shown in Figure 3.7, the surface integration is to be performed only on the shaded surfaces (\mathcal{S}_1 and \mathcal{S}_2) because on the remaining surfaces $n_x = 0$. In general, for the volume-cells produced in EIM, four surfaces are excluded from the surface integration (because generally hexahedral elements are used in the background mesh, and the integration is carried out in element local coordinate system). This feature helps to reduce the computational cost of the quadrature construction.

Step 2: Projection of arbitrarily oriented plane to coordinate plane

The facets of the volume-cell are arbitrarily oriented in $x - y - z$ space. If the surface integral is evaluated over these facets directly, a complicated procedure of the change of variables needs to be performed. In the next step, to avoid the process of change of variables, the surface integral over the arbitrary plane is projected into integral over a coordinate plane with the help of the following theorem.

Let \mathcal{S} be a surface in $x - y - z$ space which is contained in the plane $ax + by + cz = d$ whose normal is $\hat{\mathbf{n}}$. Let \mathcal{S}_x be the projection of \mathcal{S} into the $y - z$ plane, then

$$\int_{\mathcal{S}} f(x, y, z) dA = \frac{1}{|n_x|} \int_{\mathcal{S}_x} f(h(y, z), y, z) dydz \quad (3.31)$$

where,

$$h(y, z) = \frac{d - by - cz}{a} = \alpha_0 + \alpha_1 y + \alpha_2 z \quad (3.32)$$

In the example presented in Figure 3.7, both the surfaces (\mathcal{S}_1 and \mathcal{S}_2) should be projected into the $y - z$ plane by applying the above theorem. After applying equations (3.31) and (3.32), equation (3.30) becomes, for surface \mathcal{S}_i ,

$$\begin{aligned} \int_{\mathcal{S}_i} (xn_x)_i dA_i &= \frac{1}{|n_{ix}|} \int_{\mathcal{S}_{ix}} (\alpha_0 + \alpha_1 y + \alpha_2 z) n_{ix} dydz \\ &= \pm \int_{\mathcal{S}_{ix}} (\alpha_0 + \alpha_1 y + \alpha_2 z) dydz \end{aligned} \quad (3.33)$$

This step requires to find only the equation of plane which contains the arbitrarily oriented facets. The procedure used to find the equation of plane is described in Appendix B.

Step 3: Conversion of surface integral to contour integrals

In the final step, the divergence theorem is applied again to reduce the surface integral into a contour integral. Let \mathcal{S}_x be the surface bounded by a closed contour \mathcal{C} and $\hat{\mathbf{m}}$ be the outward

pointing normal to S_x along \mathcal{C} , then for any vector field \mathbf{H} ,

$$\int_{S_x} \nabla \cdot \mathbf{H} dA = \oint_{\mathcal{C}} \mathbf{H} \cdot \hat{\mathbf{m}} dl \quad (3.34)$$

The vector field equivalent to the scalar integrand can be found by the same procedure explained in step 1. For our example, applying the above theorem to equation (3.33), it becomes

$$\int_{S_x} (\alpha_0 + \alpha_1 y + \alpha_2 z) dy dz = \oint_{\mathcal{C}} (\alpha_0 y + \frac{1}{2} \alpha_1 y^2 + \alpha_2 y z) m_y dl \quad (3.35)$$

As shown in Figure 3.7, the contour integration is performed only along the lines for which $m_y \neq 0$. The contour integral in equation (3.34) can be estimated using the well known Gauss quadrature rule in one dimension.

In steps 1 and 3 of the above procedure, the normal to the facets ($\hat{\mathbf{n}}$) and contours ($\hat{\mathbf{m}}$) are computed, respectively. A key point to be noted is that the divergence theorem yields correct results only if these normal vectors are outward pointing to the considered volume-cell. In Appendix B, the method for ensuring this requirement in EIM simulations is explained.

The following important point must be noted regarding the Gauss quadrature used to perform contour integration. If \mathcal{F}_1 is the base function to be integrated over the volume, when applying the divergence theorem another function \mathcal{F}_2 is integrated along the contours of the volume. For instance, in the example explained above, to obtain the integral value of 1 over the volume-cell, an equivalent function $(\alpha_0 y + \frac{1}{2} \alpha_1 y^2 + \alpha_2 y z) m_y$ is integrated along the contours. It is very straightforward to see that the polynomial order of \mathcal{F}_2 is more than that of \mathcal{F}_1 by 2. It should be made sure that the Gauss quadrature rule used for the contour integral is accurate enough to integrate \mathcal{F}_2 . As already stated, the maximum polynomial order of base functions considered in the present work is 6, and hence the Gaussian rule used in contour integration should be able to integrate polynomials of order 8.

The use of the divergence theorem to estimate the volume integral requires that the integrand of interest (\mathcal{F}_1) must be known explicitly. In such cases, the equivalent function (\mathcal{F}_2) to be integrated along the contours can be derived, and the divergence theorem can be directly applied. However, in FEM simulations the integrand is not known explicitly, but it can be evaluated at any point within the volume-cell. So a quadrature rule which is capable of numerically estimating the integral value in such situations is needed. As already discussed, using moment fitting equations, one can arrive at such quadrature rules. With the help of this base function integration, r.h.s. of equation (3.28) can be evaluated. Since the base functions are predefined, if the position of quadrature points are known, then one can also compute the coefficient matrix associated with equation (3.28). In this work, similar to [169], it is assumed that the integration points' locations are predefined. This requires an algorithm to distribute points within the arbitrary polyhedral shaped volume-cells, which is the focus of the next section.

3.5.2.2 Quadrature point distribution

Attempts to solve for quadrature point locations over non-convex regions resulted in points placed outside the domain even in two dimensions [190, 205]. In these references, it was made sure that all the points lie within the bounding box constructed over the region of interest, and it was assumed that the integrand is continuous inside the bounding box. Such a flexibility is not available in FEM simulations because the shape functions are defined only inside the element and also they are discontinuous across the interface in the cut elements. Hence, it is mandatory that

all the quadrature points are strictly positioned inside the volume-cell. To avoid the problem of outside lying quadrature points, as in [169], the points are distributed within the volume and the corresponding weights are calculated after solving the moment fitting equations.

The distribution of points within a concave volume is not straightforward. An easy brute force method is to construct a bounding box over the arbitrary volume, distribute points uniformly within the box, and decide whether the points lie within the volume-cell. This procedure, though very simple to implement, is prohibitively expensive when the volume-cell contains very thin regions in two of its dimensions. An example of such a shape is shown in Figure 3.8. It can be seen that the bounding box contains a large region outside the volume-cell and when points are distributed and checked, only few points lie inside the volume-cell. As can be seen from Figure 3.8, out of 100 points only 5 are inside the volume. This results in unacceptably large computational time to arrive at a reasonable number of points within the volume-cell.

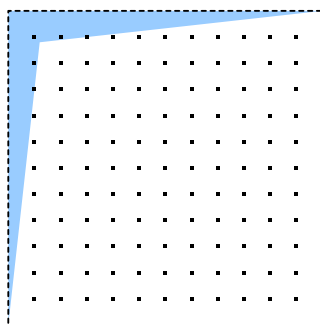


Figure 3.8: Distribution of quadrature points within the bounding box. Shaded area shows the region of interest and the dotted line marks the bounding box

An efficient method for point distribution is the use of barycentric or mean-value coordinates. Though these coordinates are well defined and widely used for convex volumes, their generalization to concave shapes in three dimensions is still lacking. The recently proposed maximum entropy coordinates are applicable for concave volumes, but they necessitate to solve a constrained optimization problem for each volume to get the coordinates [206] and the procedure has to be carried out at every time step for each volume in case of moving boundary simulations. Such complicated algorithms for the distribution of points can adversely affect the robustness of the quadrature construction scheme. To this end, a very simple yet efficient strategy is proposed for quadrature point distribution inside the concave volumes whose performance does not depend on the complexity of the volume-cell.

The present method makes use of ray tracing to distribute the points within the volume-cell. This method neither requires to find whether a point lies inside the volume-cell (as it is guaranteed to be) nor uses any special coordinate system. The volume under consideration is cut with a number of plane surfaces (Algorithm 2), and the quadrature points are distributed over the intersection regions between the plane surface and the volume (Algorithm 3).

As explained in Algorithm 2, the point distribution starts with the construction of a bounding box over the volume-cell under consideration. The corner points of the bounding box are denoted by $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$ as shown in Figure 3.9a. The next step is to find the bounding planes (denoted as z_1- and z_N- planes in algorithm 2) of the volume. For this, at first the intersection region between the volume and the plane $z_1 = z_{min} + \epsilon\Delta z$ is found. If the intersection area is significant (at least 1% of the area of bounding box at this plane), then this z_1- plane is termed as bounding plane 1 (Figure 3.9b), and quadrature points are distributed

Algorithm 2 Distribution of quadrature points within the considered volume-cell

```

1: Construct the bounding box with corner points  $(x_{min}, y_{min}, z_{min})$  and  $(x_{max}, y_{max}, z_{max})$ ;
2:  $\Delta x = x_{max} - x_{min}$ ;  $\Delta y = y_{max} - y_{min}$ ;  $\Delta z = z_{max} - z_{min}$ ;  $\epsilon = 0.01$ ;
3:  $x_1 = x_{min} - \epsilon \Delta x$ ;
4:  $y_1 = y_{min} + \epsilon \Delta y$ ;  $y_N = y_{max} - \epsilon \Delta y$ ;
5:  $z_1 = z_{min} + \epsilon \Delta z$ ;  $z_N = z_{max} - \epsilon \Delta z$ ;
6: while  $z_1 < z_N$  do
7:   Distribute quadrature points within  $z_1$  plane (Algorithm 3);
8:   if number of quadrature points  $> 0$  then
9:     Break;
10:  else
11:     $z_1 = z_1 + \epsilon \Delta z$ ;
12:  end if
13: end while
14: while  $z_N > z_1$  do
15:   Distribute quadrature points within  $z_N$  plane (Algorithm 3);
16:   if number of quadrature points  $> 0$  then
17:     Break;
18:   else
19:     $z_N = z_N - \epsilon \Delta z$ ;
20:   end if
21: end while
22: if  $(z_N - z_1) > \epsilon \Delta z$  then
23:   for  $i=2$  to  $N-1$  do
24:     $z_i = z_{i-1} + \frac{z_N - z_1}{N-1}$ ;
25:    Distribute quadrature points within  $z_i$  plane (Algorithm 3);
26:   end for
27: else
28:   Neglect this too small volume-cell;
29: end if

```

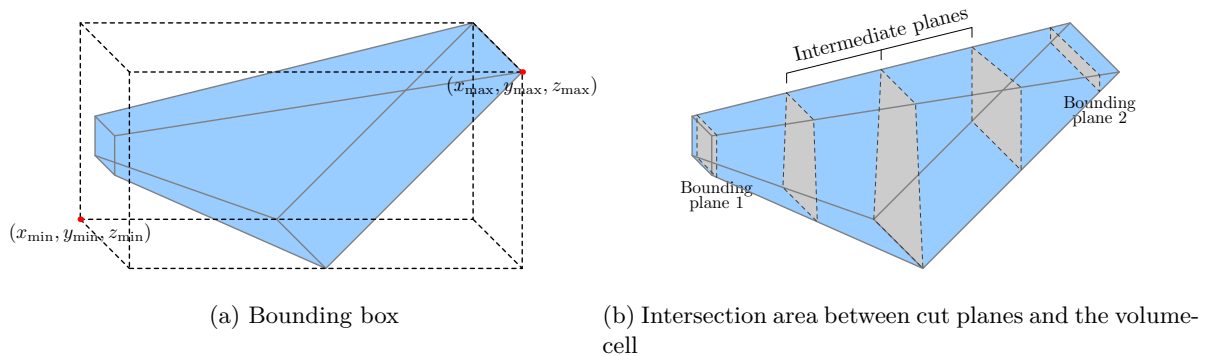


Figure 3.9: Bounding box and cut planes defined for a volume-cell

within the intersection area. Otherwise, the value of z_1 is moved within the volume in steps of $\epsilon\Delta z$ until it intersects the volume with significant area. A similar procedure is used to identify the bounding plane 2 with initial choice of $z_N = z_{max} - \epsilon\Delta z$. Once the bounding planes ($z = z_1$ and $z = z_N$) are identified, specified number of intermediate planes are generated and the quadrature points are distributed over the corresponding intersection areas. The definition of bounding planes implies that the region of volume lying outside these bounding planes is negligible, and is not considered while distributing the points. However, it is to be noted that the bounding planes are used only during the distribution of quadrature points. For the actual integration of base functions, exact geometrical details of the volume are used.

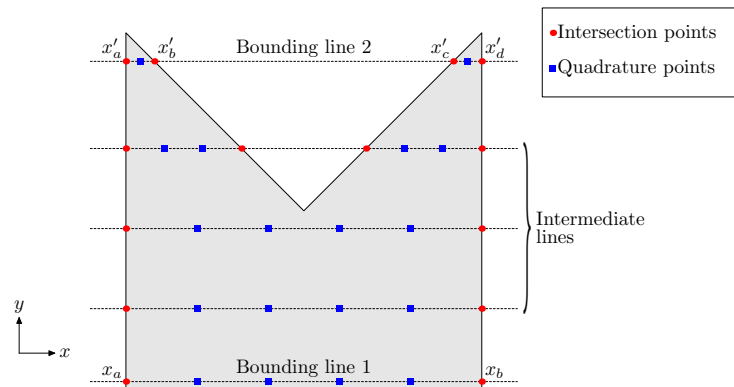


Figure 3.10: Distribution of quadrature points on the intersection region between the volume-cell and the cut plane

The procedure to decide whether the intersection area is significant and the strategy of distributing quadrature points within the intersection area are explained in algorithm 3 and are schematically shown in Figure 3.10. Ray tracing is used to distribute points in the considered plane. As a first step, the bounding lines are identified in the plane. For this, a ray is shot along the x -direction at $y_1 = y_{min} + \epsilon\Delta y$, and intersection points of the ray with the facets of the volume are identified. If the intersection length, for example $(x_b - x_a)$ for the bounding line 1 in Figure 3.10, is higher than 1% of the box's dimension in x -direction, the quadrature points are generated on the intersection length. Otherwise y_1 is increased in steps of $\epsilon\Delta y$ until it encounters

significant intersection length. Using a similar procedure, bounding line 2 is identified. When multiple intersection points are obtained, as in bounding line 2 in Figure 3.10, the quadrature points should be distributed only in the regions which lie inside the volume. Once the bounding lines are obtained, a number of rays in between them are generated, and the points are distributed over the intersection length. This completes the distribution of points in the intersection region. When this step is repeated for all the intersection region, the required quadrature points' distribution within the volume is obtained.

Algorithm 3 Distribution of quadrature points within the considered z_i -plane

```

1: Input:  $x_1, y_1, y_N, z_i$ 
2: Output: quadrature points in the  $z_i$ -plane
3: while  $y_1 < y_N$  do
4:   Shoot a ray along  $x$ -direction from  $(x_1, y_1, z_i)$ ;
5:   Find the intersection length ( $L_{int}$ );
6:   if  $L_{int} > \epsilon\Delta x$  then
7:     Distribute quadrature points within the intersection length;
8:     Break;
9:   else
10:     $y_1 = y_1 + \epsilon\Delta y$ ;
11:   end if
12: end while
13: while  $y_N < y_1$  do
14:   Shoot a ray along  $x$ -direction from  $(x_1, y_N, z_i)$ ;
15:   Find the intersection length ( $L_{int}$ );
16:   if  $L_{int} > \epsilon\Delta x$  then
17:     Distribute quadrature points within the intersection length;
18:     Break;
19:   else
20:     $y_N = y_N - \epsilon\Delta y$ ;
21:   end if
22: end while
23: if  $(y_N - y_1) > \epsilon\Delta y$  then
24:   for  $j=2$  to  $N-1$  do
25:     $y_j = y_{j-1} + \frac{y_N - y_1}{N-1}$ ;
26:    Shoot a ray along  $x$ -direction from  $(x_1, y_j, z_i)$ ;
27:    Find the intersection length ( $L_{int}$ );
28:    if  $L_{int} > \epsilon\Delta x$  then
29:      Distribute quadrature points within the intersection length;
30:    end if
31:   end for
32: else
33:   Return zero quadrature points;
34: end if

```

Some representative volume-cells over which quadrature points distributions are performed are shown in Figures 3.20 and 3.25. It is worthwhile to mention that differences in time taken between distributing the points within such complex volumes and within simple shapes such as a cube are negligible. This fact proves that the algorithm for distributing the quadrature points within the

volume is very efficient. The complexity of volumes it can handle demonstrates that the algorithm is robust. Such a performance is expected owing to the fact that all the steps in the algorithm involves straightforward algebraic calculations; there are no complex or time-consuming iterative steps involved at any stage of the algorithm.

3.5.2.3 Solution of equation system

Once a method of base function integration, and a strategy to distribute points within any arbitrary polyhedron are devised, the coefficient matrix and r.h.s. associated with equation (3.28) can be evaluated, which is given in the following form.

$$Aw = b \tag{3.36}$$

from which one solves for quadrature weights, w . The above equation generally represents an over-determined system, since the number of quadrature points is more than the number of base functions. In order to solve this, a linear least squares method is used, which modifies the above system as

$$(A^T A)w = A^T b \tag{3.37}$$

where $A^T A$ is a dense (non-sparse) square matrix. The inverse of $A^T A$ is computed using `Epetra` package of `Trilinos` [112], which uses `LAPACK` libraries [207] to perform LU-decomposition.

Having obtained the position and weights of quadrature points, the FE weak form can be integrated over the volume-cells using equation (3.25).

As discussed in section 3.6.2, with the help of appropriate numerical examples, this method is shown to be more robust than tessellation, but in certain cut configurations it can be less accurate. The next section presents an integration approach which is robust as well as accurate.

3.5.3 Direct divergence method

The previous section explained how to use the divergence theorem to integrate explicitly specified functions over arbitrary polyhedra. In order to use such a procedure to deal with unspecified functions, some researchers [187,188] utilized symbolic computations. It has already been pointed out in section 3.5.1.6 that this presents difficulties to implement in a general FE framework. In this section, an accurate, robust, efficient and easy-to-implement method is presented for the numerical integration of polynomials that are not explicitly prespecified, over arbitrarily complex shaped polyhedra. Our method involves the application of the divergence theorem to convert the volume integral to surface integrals over the facets of the polyhedra. The need for symbolic computation is eliminated by using one-dimensional Gauss quadrature rule, as in [190,191]. The material presented in this section is based on [192].

3.5.3.1 Integration over polygons

Though the main objective of this work is the integration of polynomials over complex 3D polyhedra, for brevity, the method is first explained for 2D polygons. As already stated, this method makes use of the divergence theorem to convert the domain integral into boundary integral.

Since our integrand (\mathcal{F}) is a scalar polynomial function, to utilize the divergence theorem (equation (3.29)), it should be expressed as the divergence of a vector. This is very straightforward [180,183], and can be written as

$$\nabla \cdot \mathbf{F} = \mathcal{F} \tag{3.38}$$

It can be assumed without losing generality that

$$\mathbf{F} = \mathcal{G}(\mathbf{x})\hat{i} + 0\hat{j} + 0\hat{k} \quad (3.39)$$

where \hat{i} , \hat{j} and \hat{k} are unit normal vector along x-, y- and z-directions, respectively. It follows directly from equation (3.38) and equation (3.39) that,

$$\mathcal{G}(\mathbf{x}) = \int_{\kappa}^{\mathbf{x}} \mathcal{F} dx \quad (3.40)$$

where κ is a reference point, which is explained later in this section. By substituting equations (3.38) and (3.39) into equation (3.29), one get

$$\int_{\mathcal{R}} \mathcal{F} d\mathcal{R} = \int_{\mathcal{S}} \mathcal{G}(\mathbf{x}) n_x d\mathcal{S} \quad (3.41)$$

where, n_x is the component of $\hat{\mathbf{n}}$ along x -direction. Thus, the domain integral is converted to a surface integral. Prior to the evaluation of this surface integral, one needs to perform the line integral given by equation (3.40) to compute $\mathcal{G}(\mathbf{x})$ on \mathcal{S} . In short, the problem of evaluating the domain integral is converted into performing two separate integrations: one over the surfaces of the domain (equation (3.41)) and one line integration (equation (3.40)). It is shown here that these two integrals can be computed easily and accurately.

It is to be recalled that the scope of the present work is to deal with \mathcal{F} that are not explicitly available, but are constructed during the solution process, as in typical finite element simulations. In such cases, the evaluation of $\mathcal{G}(\mathbf{x})$ using equation (3.40) is usually achieved by using symbolic computation [187, 188]. In order to eliminate symbolic computations, the one dimensional Gauss quadrature rule is used to compute $\mathcal{G}(\mathbf{x})$.

Consider the integration of a 5th order polynomial over the shaded polygon shown in Figure 3.11a. In EIM simulations, this can arise when the background element shown in the dotted line is cut by the interface (continuous line). The integration procedure explained in [189–191] can be interpreted as follows. First the domain integral is converted into integration over the boundaries of the domain using the divergence theorem. Since, the boundaries of the polygon are its edges, the application of the divergence theorem yields,

$$\int_{\mathcal{R}} \mathcal{F} d\mathcal{R} = \sum_{i=1}^{N_{ed}} \int_{E_i} \mathcal{G}(\mathbf{x}) n_x dE_i \quad (3.42)$$

where E_i denotes the i^{th} edge of the polygon, and N_{ed} is the total number of edges that defines the polygon (in the present example shown in Figure 3.11a, $N_{ed} = 5$). The integration over the edges can be performed using the one dimensional Gauss quadrature rule by mapping this rule over the edges of the polygon (Figures 3.11b and 3.11c). If $\mathcal{G}(\mathbf{x})$ is available at all these edge Gauss points, equation (3.42) can be directly evaluated.

To compute $\mathcal{G}(\mathbf{x})$ at any point \mathbf{X}_i , equation (3.40) is represented as,

$$\mathcal{G}(\mathbf{X}_i) = \int_{\kappa}^{\mathbf{X}_i} \mathcal{F} dx \quad (3.43)$$

where κ is a reference point chosen as shown in Figure 3.11b. All these edge Gauss points are connected to κ via an imaginary line, and the integral in equation (3.43) is evaluated along this imaginary line by using a more complicated radial integration [189], or by using Gauss quadrature

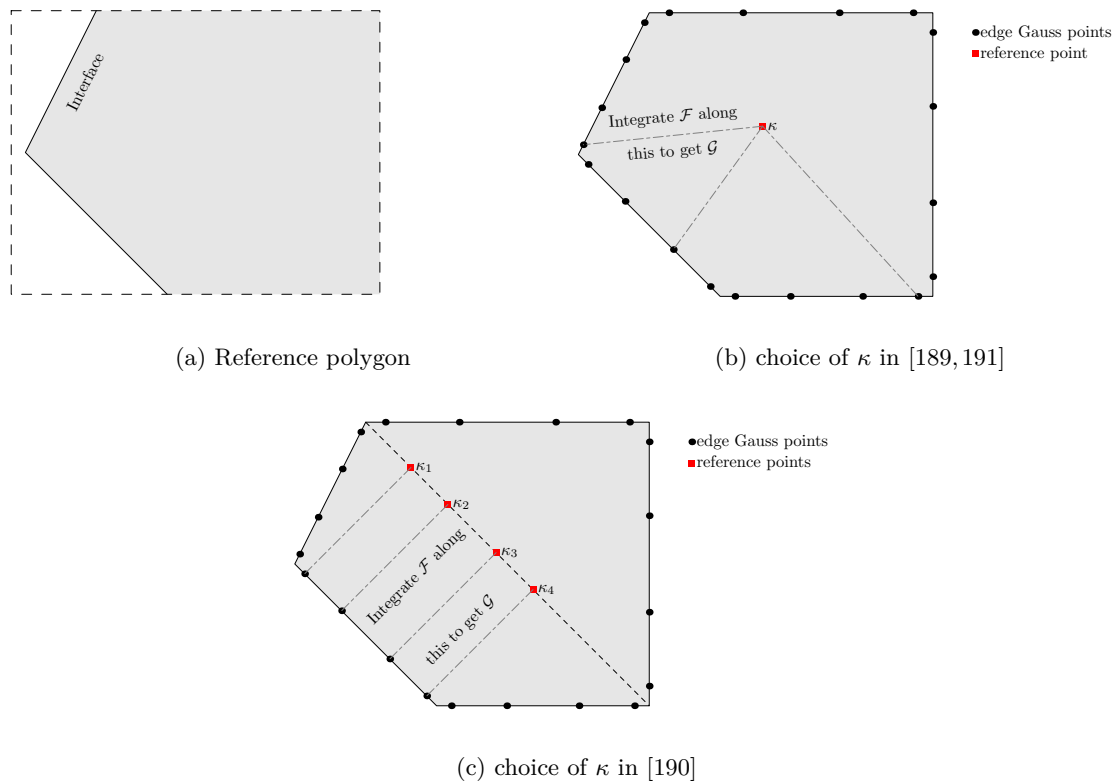


Figure 3.11: Integration over polygons as explained in [189–191]

once again [191]. In these works, κ is the same for all edge Gauss points. The approach presented in [190] is different; the edge Gauss points are projected over one of the diagonals of the polygon as shown in Figure 3.11c. Every edge point has its own κ , contrary to [189, 191]. Along the line connecting the edge point and its κ , equation (3.43) is integrated to evaluate $\mathcal{G}(\mathbf{x})$. The present method, for 2D polygons, is similar to [190]. However, instead of choosing a diagonal, the present method chooses one of the edges of the polygon to get κ for edge Gauss points. The complete procedure of the present method is explained below.

The present method of integration begins by listing the edges of the polygon for which $n_x \neq 0$. These edges are named as integration edges (Figure 3.12a). It can be seen from equation (3.42) that only edges with $n_x \neq 0$ give non-zero integral values, and hence only integration edges are considered further in our method. In the next step, the reference line which is defined as an infinite line that passes along one of the integration edges, is marked (Figure 3.12b). The role of this reference line will be explained in the subsequent steps. The edge that coincides with the reference line is no more an integration edge. Then, the one dimensional Gauss quadrature rule is mapped to the remaining integration edges (Figure 3.12c); these points are named as main Gauss points. Let (\mathbf{X}, W) denote the location and weight of these points. As explained in the above paragraph, to evaluate the domain integral using equation (3.42), the value of $\mathcal{G}(\mathbf{x})$ is required at all these main Gauss points. In order to compute $\mathcal{G}(\mathbf{x})$, all the main Gauss points are projected onto the reference line as shown in Figure 3.12d. These projected points are chosen as κ in equation (3.43), and similar to [190], κ is different for each main Gauss point. In the next step, the one dimensional Gauss quadrature rule is mapped again, but now over each interval between κ and \mathbf{X} . These quadrature points are denote as internal Gauss points (Figure 3.12e).

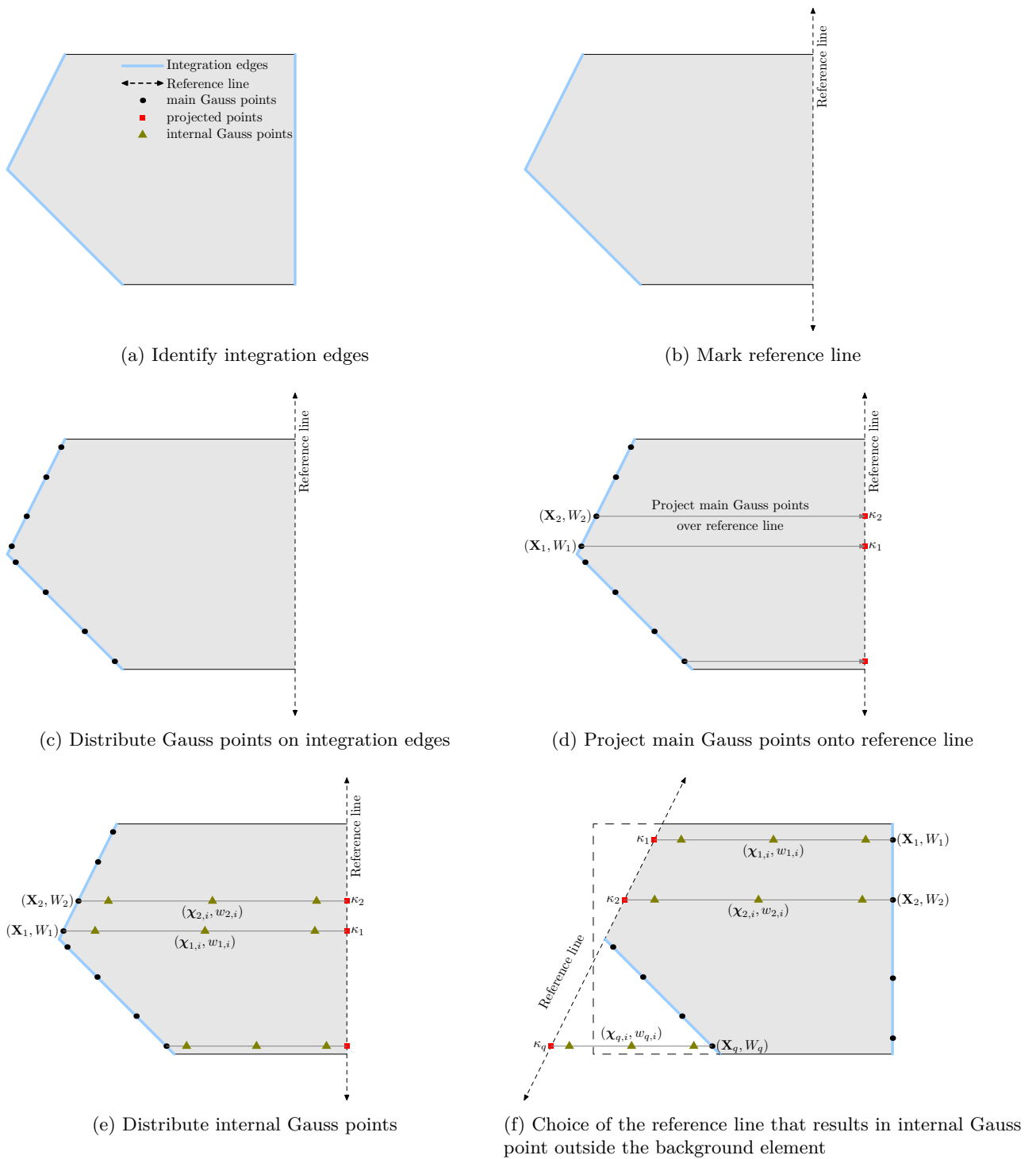


Figure 3.12: Integration over a polygon using direct divergence method. Legend shown in (a) is applicable for all the subfigures.

Now each main Gauss point has its own internal Gauss rule denoted as (χ, w) , which is used to evaluate $\mathcal{G}(\mathbf{x})$.

Evaluation of $\mathcal{G}(\mathbf{x})$ at every main Gauss point is achieved by expressing equation (3.43) in terms of the internal Gauss rule

$$\mathcal{G}(\mathbf{X}_i) = \sum_{j=1}^{NI_i} \mathcal{F}(\boldsymbol{\chi}_{i,j}) w_{i,j} \quad (3.44)$$

where NI_i is the number of internal Gauss points available for the considered main Gauss point \mathbf{X}_i .

Once $\mathcal{G}(\mathbf{x})$ is evaluated at all main Gauss points, the domain integral is computed by evaluating the right hand side of equation (3.41) as,

$$I_{\mathcal{R}} = \sum_{i=1}^{NM} \mathcal{G}(\mathbf{X}_i) n_x(\mathbf{X}_i) W_i \quad (3.45)$$

where NM is the total number of main Gauss points. All the steps involved in the integration over a polygon, with the computer implementation perspective, are summarized in algorithm 4.

Owing to the choice of the reference line in the present method, the integration edge that coincides with the reference line is eliminated from the calculations. This results in a reduced number of integration points when compared to [189–191]. In the present example, since the 4 point Gauss rule is used on the integration edges, and every edge point has a 3 point internal Gauss rule, totally 16 points are eliminated when using the present method. Moreover, when the polygon is concave, it is sometimes possible that more than one edge may coincide with the reference line, in which case the present method is even more efficient. Unlike other methods, the internal Gauss rule is always mapped along a horizontal line in our method. Hence, it is necessary only to map the x -component of the internal Gauss rule; y - and z - components are equal to the considered main Gauss point.

Algorithm 4 Integration over {polygons}[polyhedra]

- 1: Identify and store the integration {edges}[facets];
 - 2: Select the reference {line} [plane];
 - 3: Delete the {edges}[facets] falling on reference {line}[plane] from integration {edges}[facets];
 - 4: [Split the integration facets into Tri and Quad cells when necessary]
 - 5: Distribute main Gauss points (\mathbf{X}, W) on all integration {edges}[facets];
 - 6: $I_{\mathcal{R}} = 0$;
 - 7: **for** $i=1$ to number of main Gauss points **do**
 - 8: Project \mathbf{X}_i onto the reference {line}[facet] to get κ_i
 - 9: Distribute internal Gauss points $(\boldsymbol{\chi}, w)$ between \mathbf{X}_i and κ_i
 - 10: $\mathcal{G}(\mathbf{X}_i) = 0$
 - 11: **for** $j=1$ to number of internal Gauss points of \mathbf{X}_i **do**
 - 12: $\mathcal{G}(\mathbf{X}_i) = \mathcal{G}(\mathbf{X}_i) + \mathcal{F}(\boldsymbol{\chi}_{i,j}) w_{i,j}$
 - 13: **end for**
 - 14: $I_{\mathcal{R}} = I_{\mathcal{R}} + \mathcal{G}(\mathbf{X}_i) n_x(\mathbf{X}_i) W_i$
 - 15: **end for**
-

There are two questions that need further clarification regarding the present method.

- Can the reference line be chosen along any of the integration edges?

- How to choose the quadrature order for internal and edge Gauss points when the polynomial order of \mathcal{F} is known?

Choice of reference line: The polygon considered in the present example is formed when an interface cuts the background quadrilateral element (Figure 3.11a). It can be seen that the edges of the polygon are formed either from the interface or from the background element edges. When the reference line is chosen along an edge formed by the interface, it is sometimes possible that more than one edge can coincide with the reference line (as will be shown in section 3.6.2) which results in a reduced number of integration points. Hence, in our simulations, whenever possible, the reference line is set to coincide with one of the interface edges. However, while doing so, in certain cases, some of the internal Gauss points may fall outside the background element (Figure 3.12f). This is undesirable in FEM simulations, since the shape functions, and hence the integrands of the weak form, are defined only within the element. In such cases, as shown in Figure 3.12e (in general for any polygon formed after interface cut in EIM) when an edge formed by the background element is chosen as the reference edge, all internal points are within the background element. In very rare cases, for example when the element is cut by more than one interface, neither of these options is possible. In such case, an imaginary line with $x = \frac{x_{max} + x_{min}}{2}$ is chosen as the reference line; where x_{max} and x_{min} are the maximum and minimum value of x -coordinate over the whole polygon. Instead, it is possible to choose the longest diagonal as suggested in [190]. However, this possibility is not adopted because its extension to 3D is not clear. This problem is specific to FEM simulations; when the present method is used for integration of a polynomial in other applications, the reference line can coincide with any edge.

Order of quadrature rules: As stated above, the internal Gauss points are used to integrate \mathcal{F} (equation (3.44)), and the main points are used to integrate $\mathcal{G}(\mathbf{x})$ (equation (3.45)). Since $\mathcal{G}(\mathbf{x})$ is computed by integrating \mathcal{F} , the polynomial order of $\mathcal{G}(\mathbf{x})$ is increased by one. Hence the order of the the main Gauss rule should be sufficient enough to integrate $\mathcal{G}(\mathbf{x})$. This is the reason why 3-point Gauss rule is used for internal points, and 4-point Gauss rule is used for the main points in our example, to integrate a 5th order polynomial.

3.5.3.2 Integration over polyhedra

In this section, the direct divergence method is explained for performing integration over arbitrary polyhedra. The polyhedra are defined by their facets, the same way as the polygons are built by their edges. Application of the divergence theorem converts the domain integral into integrals evaluated over the facets

$$\int_{\mathcal{R}} \mathcal{F} d\mathcal{R} = \sum_{i=1}^{N_f} \int_{F_i} \mathcal{G}(\mathbf{x}) n_x dF_i \quad (3.46)$$

where F_i denotes the i^{th} facet of the polyhedra, and N_f is the total number of facets.

The procedure is very similar to the one described for polygons. All operations are carried out on the facets of the polyhedra, like on the edges of the polygon. However, here an additional step of splitting the facets into triangular and quadrilateral cells is involved, as explained below.

Various steps involved in the integration over polyhedra are illustrated in Figure 3.13, and are also explained in Algorithm 4. The method begins with identification of integration facets, facets for which $n_x \neq 0$ (Figure 3.13a). Then, the reference plane that coincides with one of the integration facets is chosen (Figure 3.13b). The facet that is in the reference plane is no more

an integration facet. To integrate equation (3.46), it is necessary to map quadrature points over the integration facets. The facets of the polyhedra are simple polygons with arbitrary number of vertices, but the quadrature rules are available only for triangle and quadrilateral. Hence, the integration facets, which are arbitrary polygons, are split into a number of triangular and quadrilateral cells (Figure 3.13c). Detailed explanation of the splitting procedure is given in the next section. Over each of these cells, the quadrature rules are mapped (Figure 3.13d), and these points are denoted as the main Gauss points (\mathbf{X}, W) . After this, the main Gauss points are projected onto the reference plane (Figure 3.13e). Then, internal Gauss points $(\boldsymbol{\chi}, w)$ are distributed along the lines connecting the main Gauss points to the projected points (Figure 3.13f).

After obtaining the Gauss points, the required integral is evaluated using equations (3.44) and (3.45). The procedure of obtaining the main and internal quadrature points are referred to as “quadrature construction” or “construction of integration scheme” in this thesis. The method of obtaining the reference plane, and how to choose the order of internal and main Gauss points have already been described in the previous section.

Shape of the eliminated facet	6 th order polynomial			10 th order polynomial		
	main ^a	internal	total	main ^b	internal	total
Triangle	13	13 × 4	65	27	27 × 6	189
Quadrilateral	16	16 × 4	80	36	36 × 6	252
Arbitrary polygon ^c	3 × 13	39 × 4	195	3 × 27	81 × 6	567

^a7th order quadrature

^b11th order quadrature

^cwhich is split at least into 3 triangles, so minimal number of reduction in points is reported

Table 3.1: Reduction in number of integration points for present method when compared to [190, 191] to integrate 6th and 10th order polynomials over a polyhedron due to elimination of one facet

The advantage of using the present method, when compared to other methods based on the divergence theorem [190, 191], is that the facet that is on the reference plane is eliminated from the computation. This results in a reduced number of integration points as shown in Table 3.1. When a triangular facet is eliminated, and if a 6th order polynomial is integrated over the polyhedra, then the facet would have had 13 main Gauss points (As pointed out in the last section, the quadrature order used for main Gauss points is more than the polynomial order of \mathcal{F} by 1). Each of these points holds 4 internal points, and hence 65 points are eliminated when using the present method. When a quadrilateral is eliminated, reduction in the number of Gauss points will raise to 80. The method is more advantageous when the eliminated facet is an arbitrary shaped polygon that will be split into a minimum of 3 triangles to map the main Gauss points; so the effect is equivalent to eliminating 3 triangular facets which will results in a reduction of 195 points. The reduction in Gauss points increases rapidly when the order of polynomial is increased. This can be seen from Table 3.1 by comparing the numbers for 10th order and 6th order polynomials. In EIM simulations, it is possible that more than one facet may coincide with the reference plane as will be shown in section 3.6.2. In such cases, the present method is even more efficient because all these facets will be eliminated from the calculations. In order to compare the numbers presented in Table 3.1 with the total number of integration points for a polyhedron, consider a polyhedron with NIF number of integration facets, out of which NE number of facets coincide with the reference plane. If one assumes that all the integration facets have the same shape, the percentage reduction in total number of integration points is $\frac{NE}{NIF} \times 100\%$. Another advantage is that facet

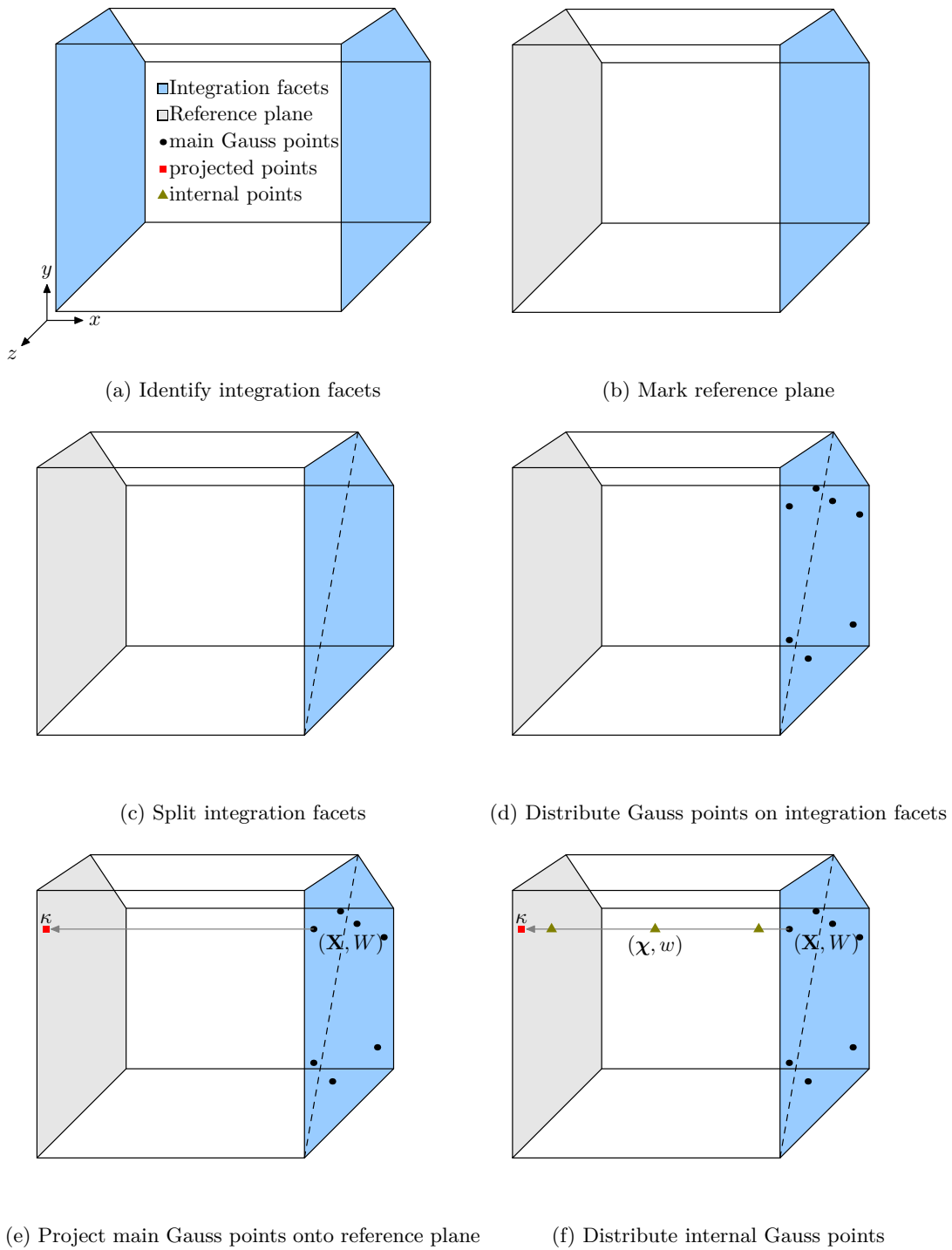


Figure 3.13: Integration over a polyhedron using direct divergence method. Legend shown in (a) is applicable for all the subfigures.

splitting is used to enable the mapping of main Gauss points over the integration facets as shown in Figure 3.13c, instead of conventional triangulation (facet splitting procedure is explained in

the next section). The implication of these advantages on the computational efficiency of EIM simulations is quantified in section 3.6.7.

In this work, the domain integral is converted into facet integrals by applying the divergence theorem. Many of the available methods of this kind use the same theorem once again to convert the facet integral into contour integrals. Though this is a possibility, it is avoided here because the implementation would become more cumbersome. It is easy to imagine that in such a procedure, three sets of Gauss rules should be handled (refer to figure 3.14) instead of two (internal and main) as in the present method. More importantly, it wouldn't be possible to make use of the optimal Gauss rules that are available for triangular and quadrilateral regions in such case. This would result in higher number of integration points, and would subsequently increase the computational time.

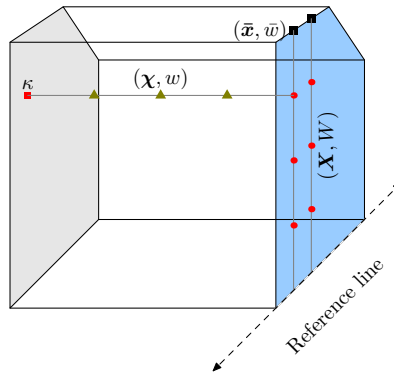


Figure 3.14: Three sets of integration rule if surface integral were converted to contour integral

Since the facets of polyhedra are simple polygons, facet splitting i.e., splitting the facet into triangular and quadrilateral cells can be easily achieved, as explained in the next section.

3.5.3.3 Facet splitting

As explained above, when a facet of a polyhedron has more than 4 vertices, mapping the main Gauss points over these facets is not straightforward. Such facets can be triangulated, the mapping can be done over each of the resulting Tri. (In this section, the abbreviations Tri and Quad are used for triangular cell and quadrilateral cell respectively.) However, as will be shown in the following text, triangulation will result in a large number of main Gauss points, which may deteriorate the efficiency of integration. To address this issue, this method uses a facet splitting procedure that decomposes a facet into a combination of Quad and Tri, by always forming as many Quads as possible. The following example explains why such a splitting procedure is advantageous over triangulation.

Assume that we encounter a nonagon (a polygon with 9 sides) shown in Figure 3.15 as one of the integration facets when performing integration over a polyhedron. When triangulated, it results in 7 Tri (Figure 3.15a); for the same facet, facet splitting produces 3 Quad and 1 Tri (Figure 3.15b). As explained in Figure 3.15, 91 and 61 main Gauss points are generated when using triangulation and facet splitting, respectively. Moreover, in our integration method, each main Gauss point holds NI number of internal Gauss points, so the total reduction in number of integration points, when using facet splitting will be $(91 - 61) * NI$. This example clearly demonstrates that it is more efficient to use facet splitting for the present method than the conventional triangulation procedure.

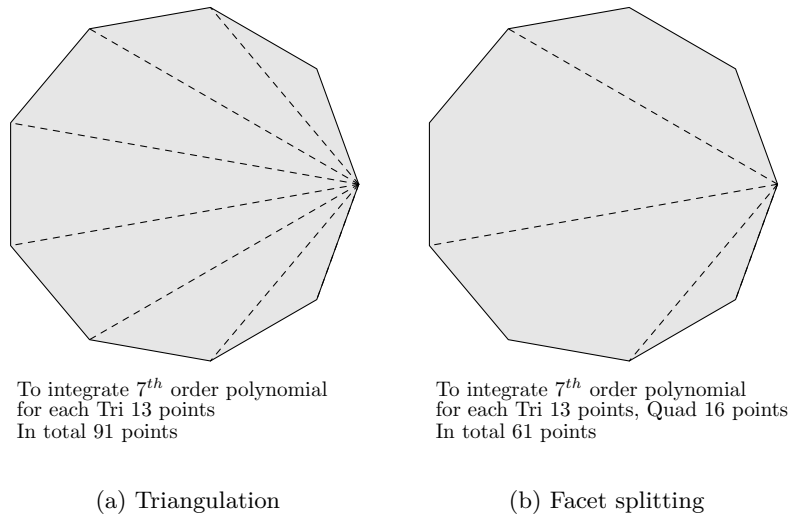


Figure 3.15: Comparison of number of main Gauss points when using triangulation and facet splitting

Facets of the polyhedra resulting from EIM simulations are always simple polygons: the edges of the facets intersect each other only at the vertices. In order to achieve the splitting of such simple polygons into Tri and Quad, the well known earclipping method of triangulation [208, 209] is modified in such a way that whenever possible, the algorithm forms a Quad shaped ear.

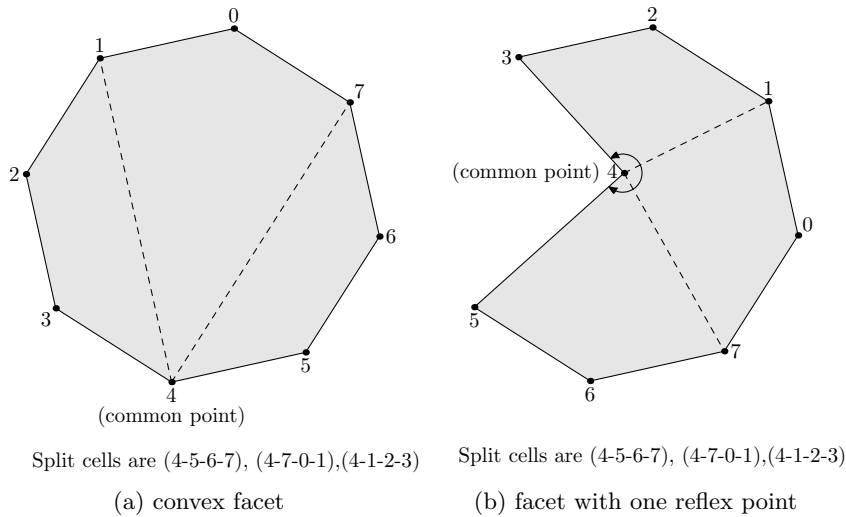


Figure 3.16: Simple facet splitting procedure for special shapes

Facet splitting is very simple for a convex facet, and for a facet with only one reflex vertex. (A vertex is reflex, if the internal angle formed at the vertex is more than 180° as in Figure 3.16b.) These facets are special facets. The first step is to choose a common point: if the facet is convex, any vertex can be chosen; if not, the reflex vertex must be chosen (vertex-4 in Figure 3.16a and 3.16b). Then, the first Quad cell is formed by connecting the common point with the subsequent three vertices in order. This is shown in Figure 3.16, where the first cell is formed by

connecting vertex-4, which is the common point, to the subsequent vertices 5, 6, and 7. The subsequent cells are formed by connecting the common point (vertex-4), the last point of the previous cell (vertex-7), and subsequent two vertices in order (vertices 0 and 1). The procedure is repeated until all the vertices of the polygon are processed. It is to be noted that when the number of vertices of the facet is an odd number, then the final cell is a Tri (Figure 3.15b). It is taken for granted that, for these special facets, all the cells formed using the aforementioned procedure lie completely within the facet i.e., these cells are the ears of the facet. If any of the Quad formed is not convex (a Quad with no self-intersection can have only one reflex point), it is split into two appropriate Tri. This is essential because quadrature rules are available only for convex Quad, and such rules cannot be mapped to a Quad with reflex points.

When a facet has more than one reflex vertex, the procedure is not straightforward. In such cases, it is necessary to form a Tri or Quad, and make sure no reflex vertices are located within this cell. The complete procedure for splitting such facets is presented in Algorithm 5. The facet is referred to as a `polygon` in the algorithm.

The algorithm first checks whether any two adjacent edges of the polygon are on the same line. If so, then these edges are merged into one. This step is important to make sure that the vertices of Tri formed during the splitting do not fall on a line. In the next step, the reflex points of the polygon are identified. Once the above two steps are completed, Tri and Quad cells can be constructed, and it can be checked whether they form the ears of the considered polygon.

To form the first cell, the first reflex point is chosen as the `firstPt`. `secondPt` is the next polygonal point to `firstPt`. If `secondPt` is a reflex point, the resulting cell cannot be an ear; it always falls outside the polygon as shown in Figure 3.17a. In such cases, the `firstPt` is shifted to the `secondPt`, and this is repeated until the `secondPt` is a non-reflex point. Then, `thirdPt` which is next polygonal point to `secondPt` in order is identified. (See lines 1 to 16 in algorithm 5) At this point, there are two possibilities.

Possibility 1: `thirdPt` is a reflex point: If one attempts to create a Quad cell, it results in either a concave Quad or a Quad with self intersecting edges. In order to avoid this, a Tri cell is formed by connecting the three available points as shown in Figure 3.17b. If no reflex points of the polygon are within this Tri, it forms an ear and is added directly to the `splitcells` that holds all the Tri and Quad cells formed during the splitting process. Then, this Tri is deleted and a new polygon is formed (compare Figures 3.17b and 3.17c). The whole procedure is repeated for this new polygon. (Refer to lines 17-19 and 22-30 in algorithm 5)

Possibility 2: `thirdPt` is a non-reflex point: A Quad is formed by connecting `firstPt`, `secondPt`, `thirdPt` and `fourthPt` (which is the next point to `thirdPt`) in cycle as shown in Figure 3.17c. This Quad forms an ear, if none of the reflex points lie within this cell. If so, the algorithm calls the routine `SplitConcaveQuadCell`. This routine checks whether the Quad is convex, in which case it is directly added to `splitcells`; but if the Quad is non-convex, it is split into two appropriate Tri and these are stored in `splitcells`. Then this Quad is deleted, and a new polygon is formed (compare Figures 3.17c and 3.17d), as in the lines between 19-21 and 32-35 in the algorithm.

If the Quad or Tri, as described above, is found not to be an ear, the algorithm moves all the way to the beginning, to modify the `firstPt`, and repeat the whole procedure.

The splitting process is completed, when all the vertices of the original polygon are part of one of the `splitcells`. From an implementation perspective, the algorithm ends after dealing with one of the following cases.

Algorithm 5 Splitting a simple polygon that has at least 2 concave vertices into Tri and Quad cells

```

1: Merge adjacent edges falling along a same line;
2: Identify reflex vertices and store them in ptReflex;
3: num=number of polygon points; numReflex=number of reflex points;
4: while (splitDone==false) do
5:   for i=0 to num-1 do
6:     Clear the data in newCell;
7:     if i==0 then
8:       firstPt=ptReflex[0];
9:     else
10:      firstPt++;
11:    end if
12:    secondPt=next polygonal point to firstPt;
13:    if secondPt is a reflex point then
14:      continue;
15:    end if
16:    thirdPt=next polygonal point to secondPt;
17:    if thirdPt is a reflex point then
18:      newCell=TriCell(firstPt,secondPt,thirdPt);
19:    else
20:      fourthPt=next polygonal point to thirdPt;
21:      newCell=QuadCell(firstPt,secondPt,thirdPt,fourthPt);
22:    end if
23:    if any reflex point is within newCell then
24:      continue;
25:    else
26:      break;
27:    end if
28:  end for
29:  if newCell is a TriCell then
30:    Add newCell to splitcells;
31:    Delete secondPt from polygon;
32:  else
33:    SplitConcaveQuadCell(newCell);
34:    Delete secondPt and thirdPt from polygon;
35:  end if
36: Merge adjacent edges falling along a same line;
37: Compute reflex points of the new polygon;
38: if num==3 then
39:   Add polygon to splitcells;
40:   splitDone=true;
41: else if num==4 then
42:   SplitConcaveQuadCell(polygon);
43:   splitDone=true;
44: else if numReflex≤1 then
45:   call SplitSpecialFacets(polygon);
46:   splitDone=true;
47: end if
48: end while

```

1. Number of vertices in the new polygon is 3: In this case, the polygon is directly added to `splitcells` as a `Tri`.
2. Number of vertices in the new polygon is 4: Here, `SplitConcaveQuadCell` routine is called.
3. There is only one or no reflex point in the new polygon: In this case, the procedure used to split the special facets (as shown in Figure 3.16) is called.

This completes the facet splitting process. These final steps are explained in lines 38-47 in Algorithm 5.

When a facet has `num` number of vertices and `numReflex` number of reflex points, and if $(\text{num} - \text{numReflex}) < 4$, then no `Quad` can be created for such facets. An example is shown in Figure 3.17e for which `num`=5 and `numReflex`=2, and hence convex `Quad` cannot be produced. Conventional earclipping procedure [208,209] is used to split such polygons into a number of `Tri`.

It is worth mentioning that the aspect ratio of the cells produced are not controlled in our splitting procedure. Cells with high aspect ratio do not deteriorate the solution accuracy when the present method is used for weak form integration in EIM. This is because the cells are used only for integration purpose, and not for constructing any shape functions, as reported also in [162].

Note: In both moment fitting and direct divergence methods, while using the divergence theorem over the volume-cell, it is essential to make sure that the normal vector on each facets is outward pointing. A procedure used to ensure this requirement is given in Appendix B.

3.6 Numerical examples

Through numerical examples, the characteristics of different integration schemes are analyzed in this section. The considered integration schemes are tessellation, generalized moment fitting method [180], and direct divergence method [192]. A simple fluid statics problem is used to clearly demonstrate the need for accurate weak form integration in fluid simulations. Then, to study the robustness, the methods are applied to generate quadrature schemes over complex polyhedra, and the errors incurred in integrating a given polynomial functions are assessed. Before analyzing the order of convergence through Beltrami flow, the accuracy of weak form integration is studied with the help of a simple fluid flow example. Finally, simulations of flow past stationary and moving interfaces are carried out to investigate the efficiency of integration schemes. Throughout this section, the generalized moment fitting method is referred simply as moment fitting method.

All the three integration methods are implemented in `BACI`, and they use same libraries for performing geometrical cutting operations. Moreover, all of them are optimized to the best of our efforts so that the comparison of computational time presented here is meaningful.

3.6.1 Fluid statics example: Significance of accurate weak form integration

Tessellation method breaks down at several instances during our simulations as already stated in section 3.5.1.1. It is possible to make the tessellation working, when the tolerances used in the geometrical cutting operations are relaxed to a sufficient value. This means small volume-cells are automatically neglected, and are not required to be split into tetrahedra. This is already proposed in [66] when simulating fracture mechanics problems using XFEM. A question that naturally arises now is: instead of developing different integration strategies, why not neglect such small cells, and proceed with the slightly reduced accurate solution? Unfortunately, this is

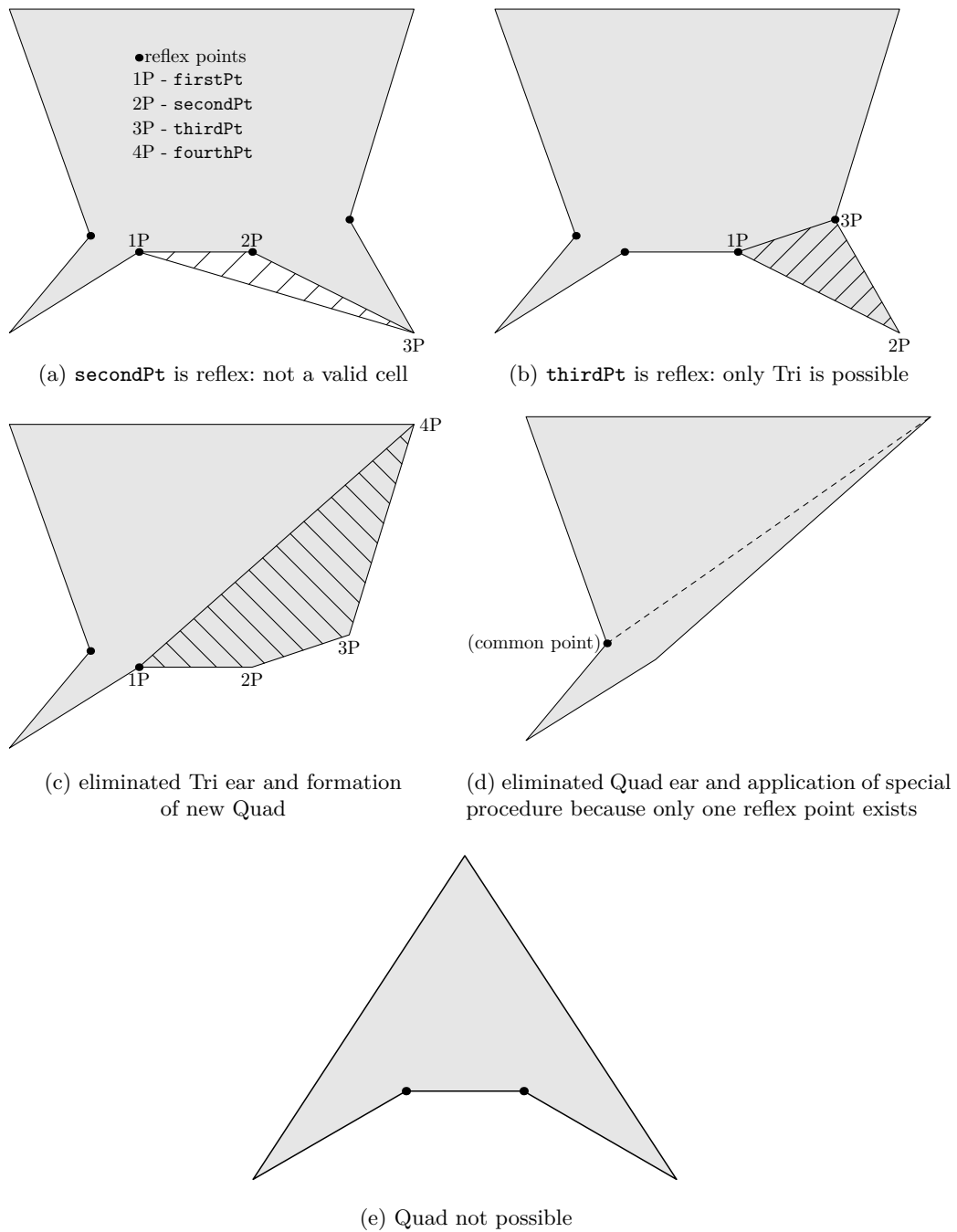


Figure 3.17: Splitting a facet with more than one reflex point. 1P, 2P, 3P and 4P represents **firstPt**, **secondPt**, **thirdPt** and **fourthPt** respectively. Shaded region is the facet, and the hatched region is a Tri or Quad cell. Legend shown in (a) is applicable for all the subfigures.

not feasible in fluid flow simulations, as pressure is very sensitive to the accuracy of weak form integration. The following example demonstrates the problem confronted when eliminating small cells.

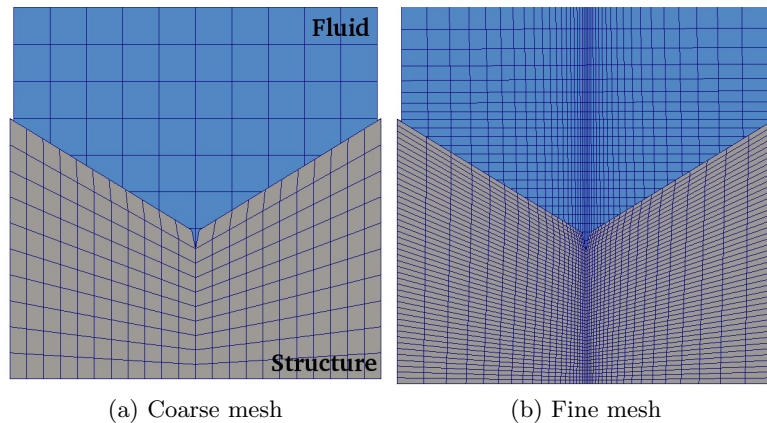


Figure 3.18: Geometry and mesh details of fluid statics problem

Consider the background fluid mesh and the structural domain shown in figure 3.18. All velocity components are set to zero over the boundaries of fluid domain, and a constant body force along the downward vertical direction is applied to the fluid. With such conditions, the Navier-Stokes equation is simplified to

$$\frac{\partial p}{\partial y} = B_0 \quad (3.47)$$

This represents a fluid statics problem, where all velocity components are zero and pressure varies only due to the addition of body forces. One can easily imagine that the pressure contours for this problem should represent a pressure field varying linearly in y -direction. Two meshes shown in figure 3.18 are considered: one coarse mesh and another fine mesh.

The pressure contours for both coarse and fine meshes, when using three different integration strategies are presented in figure 3.19. On the coarse mesh, all three integration methods yield a correct pressure field. However, on the fine grid, the tessellation procedure broke down initially. In order to avoid the failure of volume decomposition associated with tessellation, the tolerance used in the cut library is adjusted slowly until it worked. At this point, the small volume-cells produced after interface cut are neglected in tessellation. Owing to these missing volume-cells, though they are very small, the simulation does not represent a linearly varying pressure field accurately. This can be seen from the plot corresponding to tessellation for fine mesh in figure 3.19, wherein the region of inaccurate pressure field is marked with a dotted ellipse. It can be seen that both moment fitting and direct divergence method result in correct pressure field. Since these methods do not involve any volume decomposition, they can be used with tight tolerances, and hence no volume-cells are neglected.

This example ensures that neglecting the small volume-cells that is usually followed in structural mechanics simulations, is not feasible in fluid flow simulations. Indeed, it confirms that the development of accurate and robust integration methods is an essential prerequisite for developing efficient EIMs.

In this work, the volume-cells produced after the interface cut, however small it may be, are not neglected during the numerical integration. Though this is necessary to obtain non-oscillatory pressure field in fluid flow simulations, such a small volume-cells lead to a very high condition number of the resulting matrices, and the solution may become unstable which results in non-physical oscillations. To avoid this, the procedure of eliminating very small volume-cells is advocated even for XFEM based two-phase fluid flow simulations in [210,211]. Detailed pressure

field results are not reported in these studies to make conclusions regarding whether such approximations results in pressure oscillations as observed in our simulations. In [163,168], an interesting alternative is proposed to address this issue. When an element produces a very small volume-cell, the background fluid mesh points are repositioned in such a way that the interface cut introduces sufficiently large volume-cells in all the background cut elements. A Poisson equation is solved to decide the movement of the mesh points. In the present work, the edge-based stabilized formulations described in the previous sections automatically eliminate the ill-conditioning problem associated with small volume-cells, as clearly analyzed in [126].

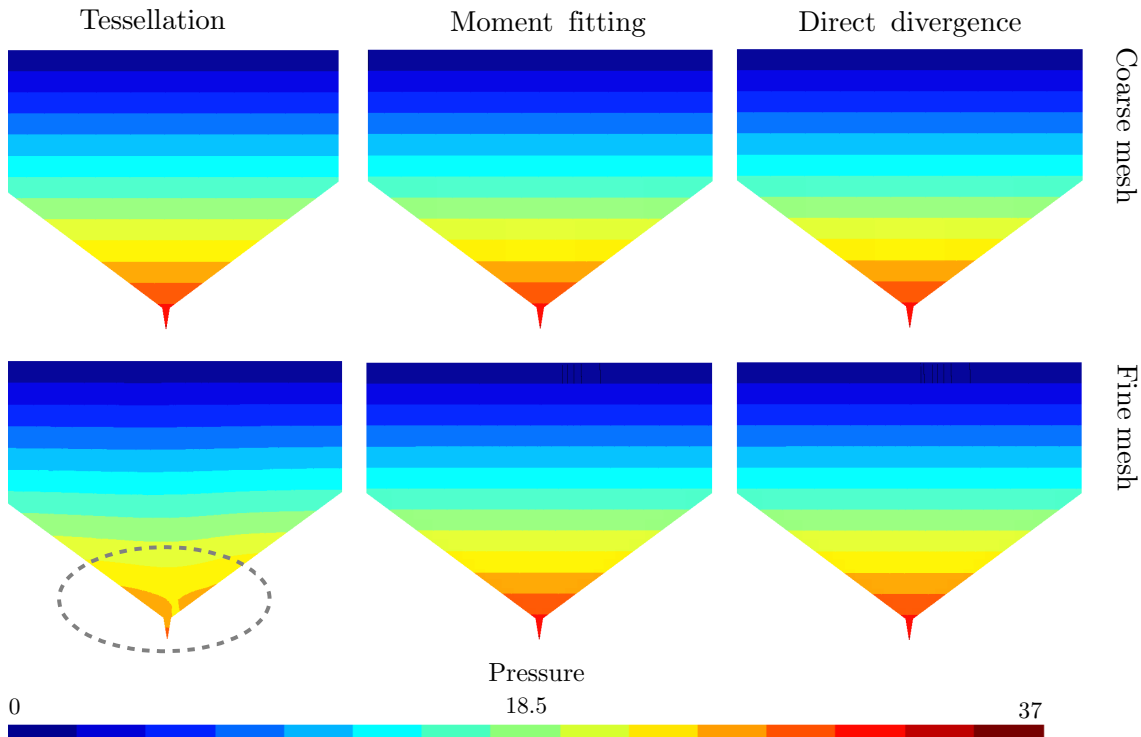


Figure 3.19: Pressure contours of fluid statics problem in coarse and fine mesh

3.6.2 Integration over arbitrary polyhedra: Robustness

In EIMs, as already described, the interface cut produces highly complex polyhedra. Any method developed for weak form integration in EIMs should be robust enough to handle such complex shapes. In order to test the robustness of the considered methods, integration of a predefined polynomial is performed over some selected polyhedra that are encountered in our simulations, and study the accuracy of such integrations.

The polyhedra considered (P1 to P4) and the corresponding integration points' distribution are shown in figure 3.20. Plots corresponding to direct divergence method need further explanation: the facet that coincides with the reference plane is marked in red color; as already stated in section 3.5.3.2, all the facets that are falling on the reference plane will be eliminated from the calculations, and these facets are marked as 'E'; moreover, since the function evaluation takes place only on internal Gauss points, they alone are included. For P3, when using direct divergence method, some of the internal points fall outside the polyhedron (Figure 3.20c). This is

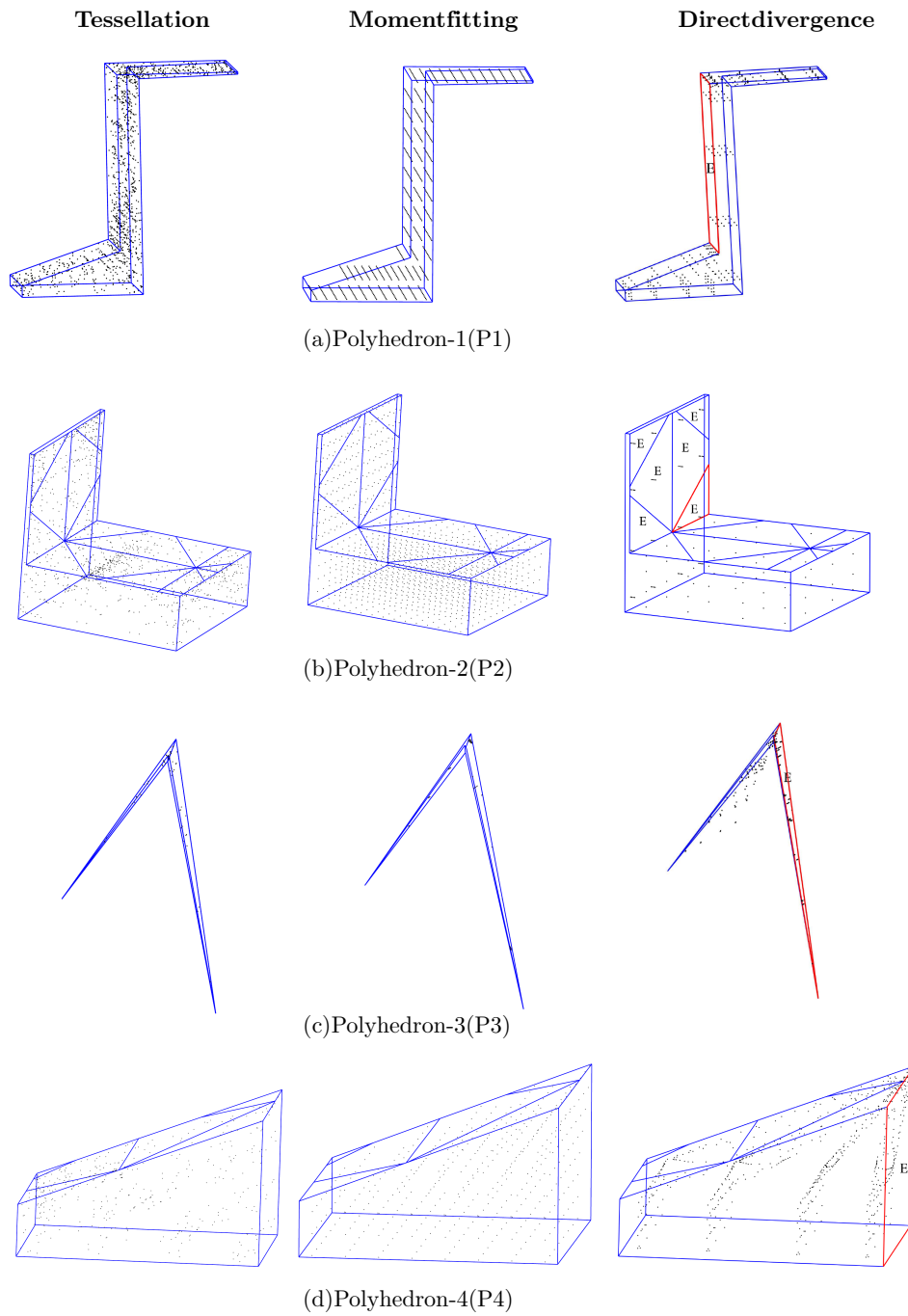


Figure 3.20: Geometry of polyhedra and integration points' distribution used to study the robustness of different integration schemes. Left column corresponds to tessellation, middle is moment fitting, and right column represents direct divergence method. Facet coincides with the reference plane is marked in red. Facets that are eliminated from calculations are marked as 'E'.

not a problem because it is always ensured that all the points are inside the background element, and hence the integrand of weak form is defined at these points also. Over these polyhedra, a 6th order polynomial $p(\mathbf{x}) = x^6 + xy^4z + x^2y^2z^2 + z^6$ is integrated. The number of integration

points and the accuracy are reported in Tables 3.2 and 3.3, respectively. The term “similar” in these tables refers to other methods based on the divergence theorem [190, 191]. The differences between these methods and the direct divergence method are already stated in section 3.5.3.2.

The number of integration points required to integrate $p(\mathbf{x})$ are given in Table 3.2. The number of tetrahedra generated after the volume decomposition process of tessellation is also marked within parentheses. The quadrature rules for these tetrahedra are obtained from **Intrepid** library of the Trilinos project [112]. For the direct divergence and similar methods, internal Gauss points are used for comparison, since the function evaluation is carried out only on these points. Also, the total (main+internal) number of Gauss points are marked within parentheses.

Polyhedron	Tessellation ^a	Moment fitting	Direct divergence ^b	Similar ^b
P1	1248 (52 Tet)	756	752 (940)	1040 (1300)
P2	1032 (43 Tet)	1232	128 (160)	728 (910)
P3	48 (2 Tet)	125	220 (275)	312 (390)
P4	456 (19 Tet)	424	620 (775)	884 (1105)

^aNumber of integration points (number of tetrahedra after volume decomposition)

^bNumber of internal Gauss points (total number of integration points)

Table 3.2: Number of quadrature points generated by different integration schemes when performing integration of a 6th order polynomial over polyhedra shown in Figure 3.20.

Table 3.2 indicates that for P1 and P2, the number of integration points required for the direct divergence method is less than for the other methods. For P2, due to the choice of the reference plane, 6 facets are eliminated from calculations since all of them lie on the reference plane (Figure 3.20b). As a result, for P2 the direct divergence method requires very few integration points (Table 3.2). Elimination of more than one facet for P2 is possible only because the reference plane coincides with a facet formed from the interface (refer to the discussion on the choice of reference line in section 3.5.3.1). For P3 and P4, the direct divergence method results in more integration points. The reason for this is explained as follows. P3 is a special shape that is composed of two sliver tetrahedra. Since tessellation recognizes these special features, it is more efficient for P3. For P4, only 1 out of 8 integration facets is eliminated due to the reference plane concept. This is not sufficient enough to make the direct divergence method more efficient than tessellation. It is to be mentioned that 1 out of 5, and 6 out of 8 integration facets are eliminated for P1 and P2, respectively. As the ratio of number of eliminated facets to the total number of integration facets is high, the direct divergence method requires less integration points than tessellation. In general, when a polyhedron is simple shaped, as P4 or posses special geometrical features like P3, tessellation requires a smaller number of integration points; when the considered volume-cell is complex shapes, like P1 and P2, moment fitting and direct divergence outperforms tessellation. For all the polyhedra considered here, the number of integration points from direct divergence method is less than those from similar methods, for reasons stated in section 3.5.3.2. The influence of this on the computational efficiency of EIM simulations are reported in section 3.6.7.

P4 in Figure 3.20 is very important, since it represents a typical polyhedron that is generally encountered in EIM simulations. The fact that for P4, the direct divergence method requires more points than tessellation and moment fitting affects the efficiency as will be shown in the forthcoming sections.

Accuracy of the quadrature rule is quantified by the quadrature error which is defined as

$$e_Q = \left| \frac{I_\Omega - I_Q}{I_\Omega} \right| \quad (3.48)$$

where I_Ω is the exact integral value of the considered polynomial over the volume, and I_Q is the value obtained from the considered method. The value of I_Ω is obtained by applying the divergence theorem.

A comparison of e_Q produced when integrating $p(\mathbf{x})$, given in Table 3.3, indicates that the direct divergence method, tessellation and similar methods give the same level of accuracy. Even for P3, for which the moment fitting struggles to construct efficient integration scheme, the error from these methods is $\sim 10^{-15}$. It can be concluded that the direct divergence method is robust since it can construct accurate integration schemes for all the complex polyhedra considered here.

Polyhedron	Tessellation	Moment fitting	Direct divergence	Similar
P1	3.7345×10^{-12}	1.5544×10^{-11}	3.7374×10^{-12}	3.7380×10^{-12}
P2	1.0472×10^{-15}	4.5992×10^{-13}	2.5132×10^{-15}	5.6548×10^{-15}
P3	1.7522×10^{-15}	4.4125×10^{-06}	6.3159×10^{-15}	6.3157×10^{-15}
P4	5.5618×10^{-11}	3.8046×10^{-11}	1.8687×10^{-13}	8.8994×10^{-14}

Table 3.3: Error produced in integrating $p(\mathbf{x})$ using different integration schemes over the polyhedra shown in Figure 3.20

The number of integration points in tessellation is fixed by the number of integration-cells it produces, and in direct divergence method, the number of points is dictated by the number of facets. For a given volume-cell, the number of points has a unique value to integrate a polynomial of certain degree. However, as already explained, the integration points in moment fitting method are distributed, and the quadrature weights at the corresponding points are obtained. In such a procedure, a question that naturally arises is “how to decide the number of quadrature points?”. In all our moment fitting examples, the L_∞ -norm of e_Q is calculated for each base function, and the number of quadrature points is increased until L_∞ -norm falls below 10^{-10} for each base function.

In order to better understand the behavior of moment fitting method with respect to number of integration points, a few polynomials up to sixth order are integrated within the complex volumes. The convergence plots of e_Q for the chosen polynomials with number of quadrature points are shown in Figures 3.21. The constant term is used to check whether the volume of the cell is predicted properly. It is very important to make sure that the quadrature is able to integrate a constant term correctly. Otherwise it will be necessary to use a large number of integration points for accurate calculations [74]. Figures 3.21 clearly shows that e_Q does not reduce monotonically with increasing number of quadrature points. Importantly, for P3, even if the number of points are increased to 5000, the error cannot be reduced below 10^{-8} . This may be due to the fact that the moment fitting matrix may not be well-conditioned for the considered number of, and distribution of quadrature points. Moreover, the quadrature points’ refinement is achieved by simply increasing the number of cut planes and the number of points to be generated in the intersection length in our point distribution algorithm, explained in section 3.5.2.2. This means that the geometrical details of the volume are not considered, and hence the refinement process is not optimal. Another reason which can be of minor influence is the following: there is an error introduced in integrating each base function while solving the least squares problem. When a polynomial is constructed as the combination of base functions, every term in the polynomial

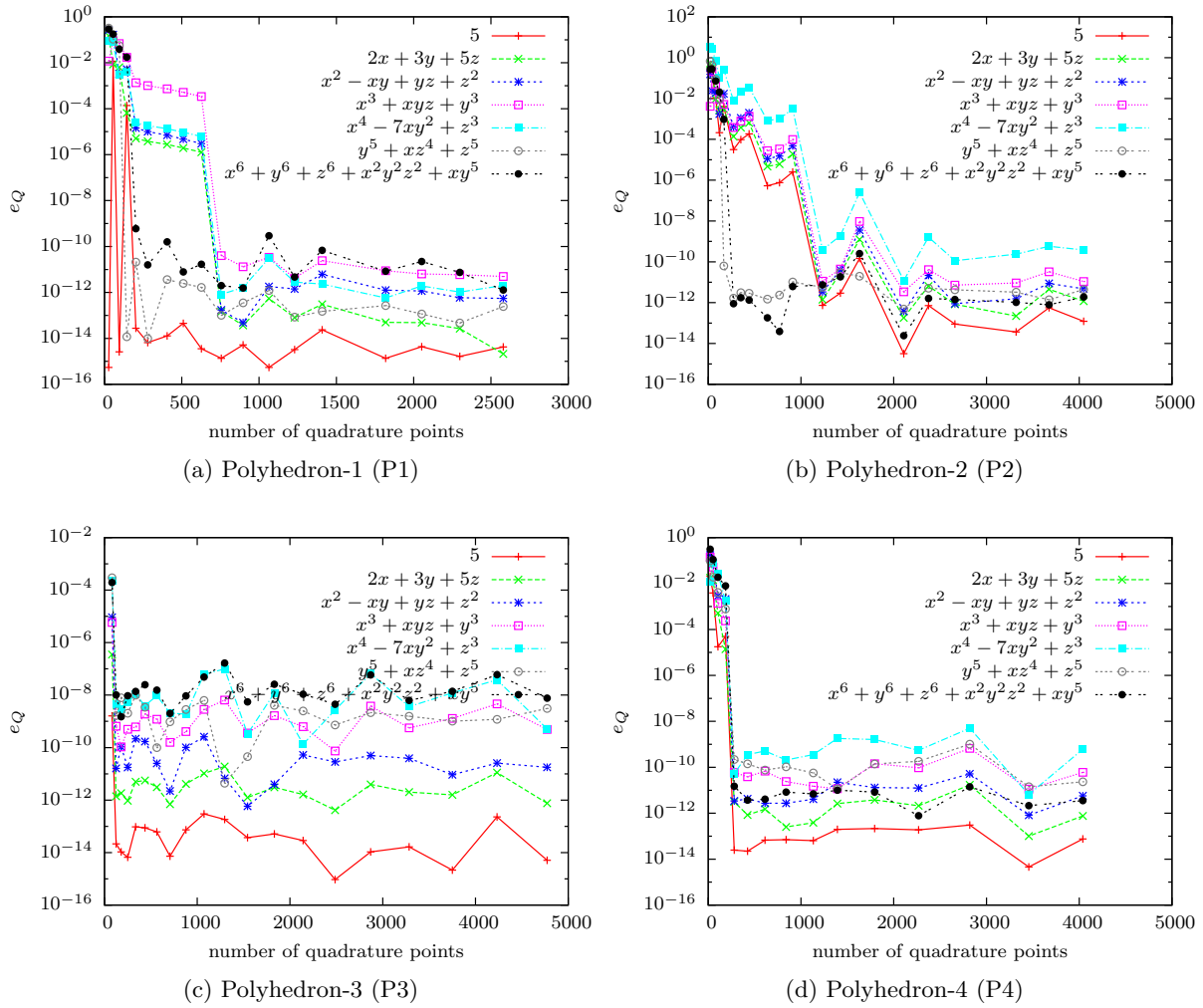


Figure 3.21: Convergence plot for known polynomials when using moment fitting methods

contributes to the total error. Depending upon the sign of error introduced in each term of the polynomial, the error oscillates.

A recent study [212] points out the conditioning problem in moment fitting methods is due to the use of monomials as base functions. In order to circumvent the problems, Gram-Schmidt procedure is applied on the monomials, and the resulting orthogonal functions are used as base functions, and this is shown to provide improved conditioning.

3.6.3 A simple test case : Accuracy

Having discussed the accuracy of integrating the known polynomials over the irregularly shaped volumes to study the robustness, the next step is to analyze the accuracy of the stiffness matrix computation, i.e., for the integration of the weak form of the governing equations. In order to do this, a very simple fluid flow problem is considered for which the exact solution for velocities (u, v and w) and pressure (p) are known.

The background mesh and the cut interface are shown in Figure 3.22. On all the surfaces of the domain, a constant velocity boundary condition is applied. Same traction condition is given

at the inlet and the outlet which are marked in Figure 3.22. Body force of appropriate magnitude is applied, so as to keep the pressure constant throughout the domain. So for the considered problem, the velocity and the pressure field are constant in the domain, and are confirming to the boundary conditions applied. The interface cuts the background mesh and the same velocity conditions as applied on the boundaries of the domain are imposed on the interface. Figure 3.22 also shows the tessellation of the volume-cells into integration-cells.

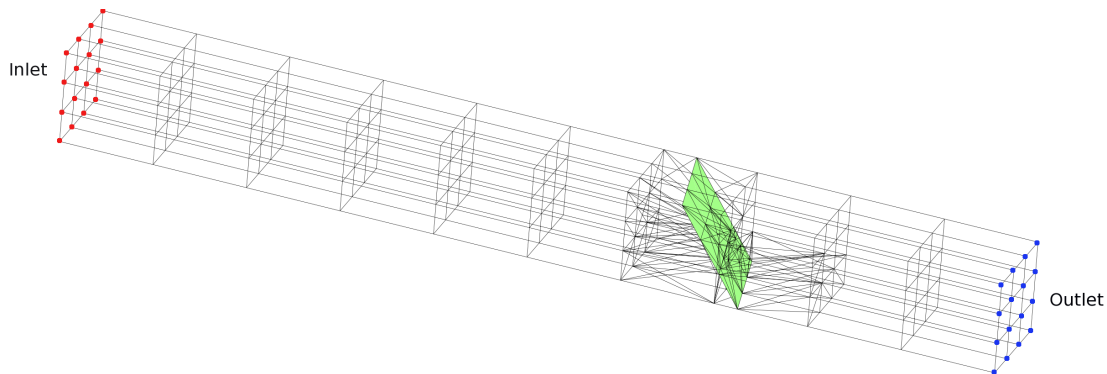


Figure 3.22: Geometry of the simple test case used to study the error in weak form integration. The decomposition of cut elements into tetrahedra shown is performed only for Tessellation.

In this test case, the errors produced in the value of velocity and pressure at any node of a cut element are direct measures of accuracy in integrating the weak form of the Navier-Stokes equations. Such errors induced by different integration methods are given in Table 3.4. It can be seen that the direct divergence method induces the least error on the computed flow quantities. Since the geometry, mesh, and boundary conditions are the same, and only the method to perform integration over the cut elements is changed, one can conclude that the direct divergence method is more accurate when compared to the other two alternatives. Even for the pressure, the most sensitive quantity, the direct divergence method gives error $\sim 10^{-15}$.

Quantity	Tessellation	Moment fitting	Direct divergence
u	8.88×10^{-16}	2.22×10^{-10}	$< 10^{-18}$
v	9.19×10^{-16}	6.87×10^{-12}	1.11×10^{-17}
w	5.01×10^{-16}	3.95×10^{-12}	1.62×10^{-17}
p	1.23×10^{-13}	5.96×10^{-09}	8.88×10^{-15}

Table 3.4: Error introduced in velocity and pressure on a node of a cut element by different integration methods

The error induced by moment fitting is larger than the other two integration methods, as quantified in Table 3.4. The reason for this is clear from the discussion of previous section, wherein it was shown that the moment fitting method achieves lowest accuracy when known polynomials are integrated over the volume-cells. This is the cause of observed less accuracy in weak form integration. Table 3.4 demonstrates also that the errors in pressure dominate, which is no surprise because pressure is the most sensitive quantity. Though the errors produced in the moment fitting method is higher than other methods, the error is small enough that it does not affect the order of convergence of EIM, as will be shown using the next example.

3.6.4 Beltrami flow : Order of convergence

The Beltrami flow is an ideal choice to study the influence of the quadrature error on the convergence of FE solution because while deriving the analytical solution to the Beltrami flow, no terms in the Navier-Stokes equations were eliminated [213].

The exact solution for the Beltrami problem is given by,

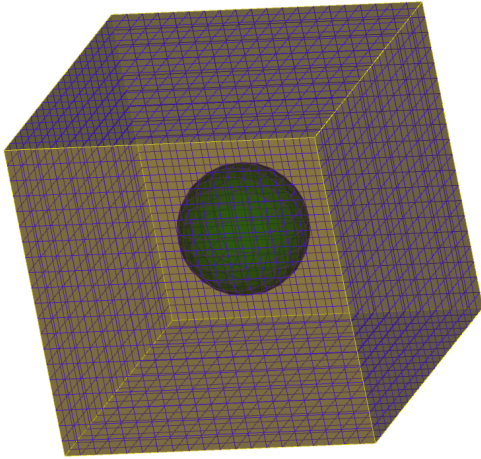
$$u(x, y, z) = -a [e^{ax} \sin(ay + dz) + e^{az} \cos(ax + dy)] e^{-\nu d^2 t} \quad (3.49a)$$

$$v(x, y, z) = -a [e^{ay} \sin(az + dx) + e^{ax} \cos(ay + dz)] e^{-\nu d^2 t} \quad (3.49b)$$

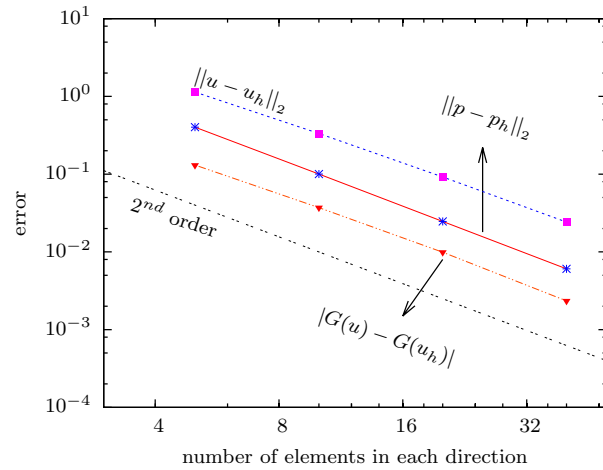
$$w(x, y, z) = -a [e^{az} \sin(ax + dy) + e^{ay} \cos(az + dx)] e^{-\nu d^2 t} \quad (3.49c)$$

$$\begin{aligned} p(x, y, z) = & -\frac{a^2}{2} \left[e^{2ax} + e^{2ay} + e^{2az} + 2\sin(ax + dy)\cos(az + dx)e^{a(y+z)} \right. \\ & + 2\sin(ay + dz)\cos(ax + dy)e^{a(z+x)} \\ & \left. + 2\sin(az + dx)\cos(ay + dz)e^{a(x+y)} \right] e^{-2\nu d^2 t} \end{aligned} \quad (3.49d)$$

The initial and boundary conditions are obtained from the above equations. Following [213], the constants are chosen as $a = \pi/4$ and $d = \pi/2$. The time step used in the simulation is $\Delta t = 0.05$. To reduce the mesh resolution requirements, a diffusion dominated flow with $\nu = 1$ is simulated.



(a) Spherical domain (the embedded mesh) cuts the background mesh in the cubical domain



(b) Convergence plot

Figure 3.23: Beltrami flow example

The simulation setup for Beltrami flow is shown in Figure 3.23a. A cube of length 2.5 serves as the background mesh, and a sphere of unit diameter whose centre is at the origin is embedded within the cube. The surface of the sphere represents the embedded interface. It is to be noted that the spherical domain does not represent a solid body. It is just an embedded mesh that contains the same fluid as in the cubical domain. The initial conditions are applied to both cubical and spherical domains, and on the surfaces of the cube the boundary conditions from the exact solutions are applied at each time step. At every time step, the solution over the cubical and spherical domain are solved in a coupled manner. It is to be mentioned that, in addition to the present work, the integration schemes presented in this thesis are also applicable for embedding

an arbitrary fluid patch into a background mesh as discussed in [28]. This example uses such an approach.

Several authors [214–216] have reported the effect of numerical integration on the accuracy of FE solutions, among which [216] is important for us. In [216] it has been shown that when the data of the problem has increased smoothness, not only the quantities computed from FEM, but also any functional derived from these quantities should converge at optimal order. This can be used as an accuracy test for our quadrature scheme. Since $(\mathbf{u}, p) \in C^\infty(\Omega)$ for the Beltrami flow, \mathbf{u}, p and any derived functional should converge at second order for an appropriately accurate quadrature scheme. In order to test this the following functional is defined,

$$G(u) = \int_{\Omega} \sin(x) \frac{\partial u}{\partial x} d\Omega \quad (3.50)$$

and its convergence rate is checked. The convergence of \mathbf{u}, p and $G(u)$ with increasing number of elements in each direction is shown in Figure 3.23b. It can be seen that both \mathbf{u} and p in their L_2 -norm, and $|G(u) - G(u_h)|$ converge almost with second order; where $G(u)$ and $G(u_h)$ are computed from analytical and FE solution using equation (3.50), respectively. Since all three methods converge at the same order and produced same error values, different integration methods are not identified in figure 3.23b.

From the above analyses, it can be concluded that all the three integration methods are of sufficiently accurate to be used for weak form integration in EIM.

Having discussed the robustness, accuracy, and convergence, the forthcoming sections report the computational efficiency of the integration schemes.

3.6.5 Stationary interface simulations : Efficiency

As a next step, flow past stationary interfaces is simulated to study the computational efficiency of different integration schemes. The interface shapes considered are a circular cylinder and a star. Shape of the interfaces and mesh details are shown in Figure 3.24. The mesh is fine near the interface and is coarsened away from it. The mesh contains one cell in the third dimension, so that the interface cut produces polyhedra.

The reasons for choosing such interface shapes are as follows. Flow past a circular cylinder is a widely used test case for validation of numerical schemes. Hence, the flow quantities computed from our simulations can be compared with the available literature results. However, since the circular cylinder has smooth geometry, it introduces only volumes with moderate complexity over which integration needs to be performed. In order to study the integration methods in the presence of highly complicated polyhedra, the star configuration is chosen. Some of the complex polyhedra resulting from the interface cut for flow past the star are shown in Figure 3.25.

Simulations are carried out at Reynolds number, $Re = U_\infty D / \nu$ of 20; where U_∞ is the free stream velocity, ν is the kinematic viscosity of the fluid, and D is the diameter for circular cylinder and mean-diameter for the star (see Figure 3.24a). Unsteady simulations with $\Delta t = 0.1$ are performed for 1200 time steps. Though the time integration is carried out for 1200 steps, because the interface is stationary in these examples, the geometrical cutting operations and the construction of integration schemes are carried out only once.

Comparison of drag coefficient (C_D) for flow past the circular cylinder and the star are given in Table 3.5. It can be seen that results obtained when using all the three integration methods match very well with the reported literature values for the circular cylinder example. The important point is that for both cases the direct divergence method produces exactly the same results as the tessellation, while the results from moment fitting are slightly different. This is because the

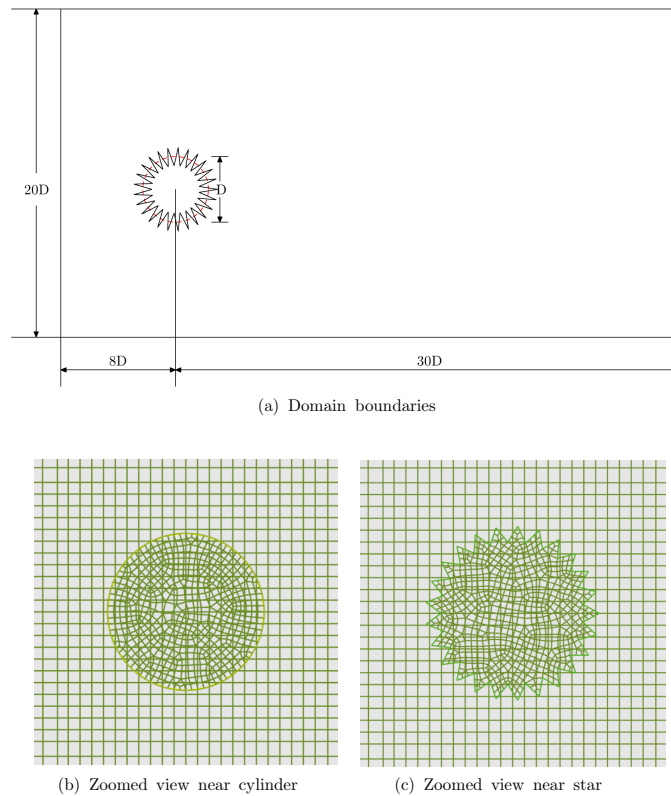


Figure 3.24: Mesh details for stationary interface simulations

	Circular cylinder	Star
Tritton [217]	2.22	-
Calhoun [218]	2.19	-
Russel and Wang [219]	2.13	-
Tessellation	2.1252	1.8749
Moment Fitting	2.1276	1.8754
Direct divergence	2.1252	1.8749

Table 3.5: Drag coefficient (C_D) for flow past the circular cylinder and star at $Re = 20$

error produced by the moment fitting method is larger than other methods, as explained in the previous sections. Though results from moment fitting are slightly different, they also agree very well with the reported values. The pressure field over the star after the simulation attained steady state is pictured in Figure 3.26 which shows that the results are qualitatively similar.

The time required for the quadrature construction and for the total simulation are compared in Tables 3.6 and 3.7, respectively. Table 3.6 indicates that moment fitting is the costliest for quadrature construction. This is due to the fact that for the construction of quadrature rules using the moment fitting method, a small system of linear algebraic equations (equation (3.28)) needs to be solved for every volume-cell. The coefficient matrix associated with this system of equations

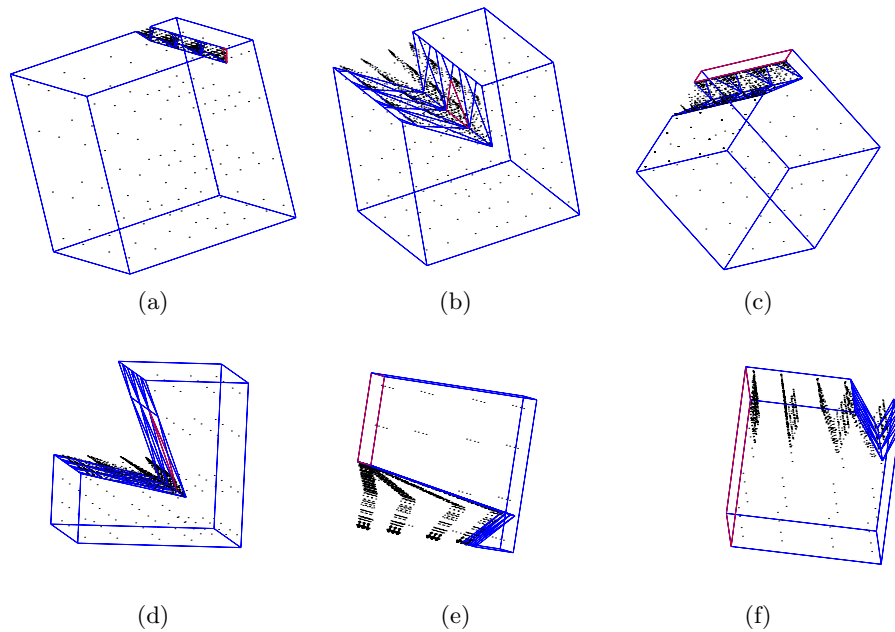


Figure 3.25: Some polyhedra generated when solving flow past the star. Integration points shown are for direct divergence method.

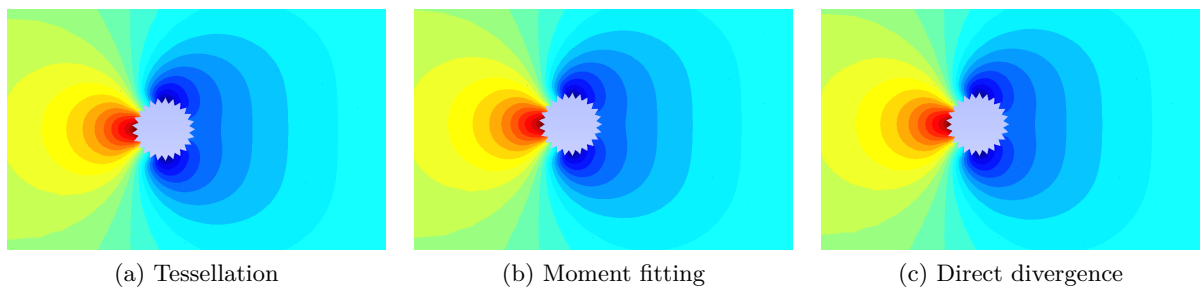


Figure 3.26: Steady state pressure field over flow past the star at $Re = 20$

is always a full (non-sparse) matrix. It is obvious that solving such a system is time consuming. Tessellation involves a volume splitting process which, though is complicated to implement, is faster than moment fitting method. Direct divergence method outperforms the other two in quadrature construction time, since it involves only some mapping operations on Gauss points over triangles, quadrilaterals and lines, and it requires the rather simple facet splitting procedure. Both these mapping and facet splitting are performed very quickly when compared to the matrix inversion of moment fitting and volume splitting of tessellation. This is why direct divergence method is faster than the other two; it is more than 8 and 78 times faster than tessellation and moment fitting, respectively for circular cylinder example. The ratios go to 36 and 51 for the star problem.

As far as the total simulation time is concerned, moment fitting is the most efficient for stationary interface problems (Table 3.7). It is directly evident from Table 3.7 that the direct divergence method is almost as efficient as tessellation. Detailed discussion on total simulation time is given in the next section.

	no. of quadrature construction	Tessellation	Moment Fitting	Direct divergence
Stationary cylinder	1	0.4003	3.78	0.048
Stationary star	1	8.422	11.97	0.2312
Accelerating airfoil	160	330.65	1.056×10^4	42.992
Oscillating hemisphere	100	925.71	6424.69	105.47

Table 3.6: Comparison of quadrature construction time in seconds

	no. of time steps	Tessellation	Moment Fitting	Direct divergence
Stationary cylinder	1200	1.486×10^4	1.310×10^4	1.487×10^4
Stationary star	1200	10.35×10^4	9.407×10^4	10.72×10^4
Accelerating airfoil	160	1.606×10^4	2.576×10^4	2.130×10^4
Oscillating hemisphere	100	12.85×10^4	13.41×10^4	14.49×10^4

Table 3.7: Comparison of total simulation time in seconds

3.6.6 Moving interface simulations: Efficiency

Results of the previous section prove that for stationary interface simulations, the proposed method is as accurate and almost as efficient as tessellation. In FSI, one problem class of our interest, the interface changes its shape and position at every time step. This requires the geometrical cutting and the construction of integration schemes to be performed at each time step. To test the efficiency of integration methods in such situations, two simulations are performed: flow over a rapidly accelerating airfoil, and an inline oscillating hemisphere. The former serves as a validation study for moving interface simulations, and the later is a full three-dimensional simulation, as opposed to the cases reported in the previous section wherein there is only one cell in the third dimension.

The first moving interface example considered is the flow field around rapidly accelerated airfoil in a still fluid, as described in [220]. The NACA0012 airfoil is used in our simulations. It accelerates from rest to a constant velocity, $U = 1$, and then it traverses with this constant velocity. Throughout the simulation, the airfoil maintains a constant angle of attack of 35° . The unsteady displacement of the airfoil is given as follows,

$$x(\tau) = -\frac{1}{2} \left[\tau - \frac{\tau_a}{\pi} \sin \left(\frac{\pi\tau}{\tau_a} \right) \right] \quad \text{for } 0 \leq \tau \leq \tau_a \quad (3.51a)$$

$$x(\tau) = \frac{\tau_a}{2} - \tau \quad \text{for } \tau > \tau_a \quad (3.51b)$$

where $\tau = \frac{tU}{c}$ is the non-dimensional time, $\tau_a = 0.8$ is the acceleration duration, c is the chord length of the airfoil. Reynolds number, the non-dimensional number that characterizes the flow field, $Re = \frac{Uc}{\nu}$ is set to be 100, where ν is the kinematic viscosity of fluid. The computational domain is taken to be $24c \times 20c$, where c is the chord length of the airfoil. The complete subdomain within which the airfoil traverses is discretized with a very fine mesh, and a coarse mesh is used away from this region. The time step used in the simulation is $\Delta\tau = 0.02$.

The evolution of vorticity field around the airfoil at different instances of time is illustrated in figure 3.27. It can be seen that negative vorticity is generated around leading-edge and the upper surface, whereas positive vorticity is generated over the bottom surface and the trailing-

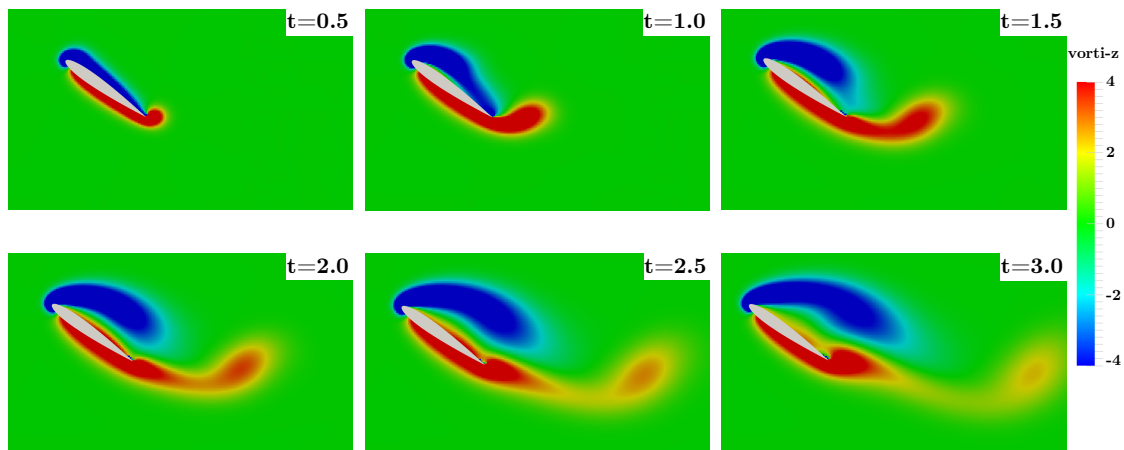


Figure 3.27: Evolution of vorticity field around airfoil

edge region. As the airfoil starts moving, over the period of time, the positive vorticity is shed from the trailing edge to the wake region. These qualitative features agree very well with the observations reported in [220].

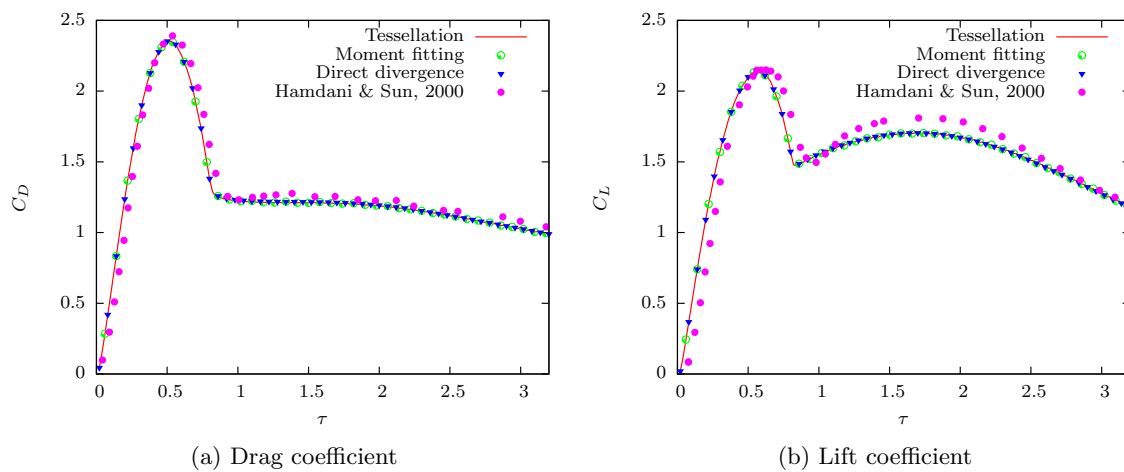


Figure 3.28: Comparison of drag and lift coefficients for rapidly accelerated airfoil

The simulation is also validated by computing the drag and lift coefficients, and comparing them with the results provided in [220]. The drag and lift coefficients are defined as follows,

$$C_D(\tau) = \frac{2[D(\tau) - I_a(\tau)]}{\rho_f U^2 c b}; \quad C_L(\tau) = \frac{2L(\tau)}{\rho_f U^2 c b} \quad (3.52)$$

where $L(\tau)$ and $D(\tau)$ are the lift and drag forces acting on the airfoil, ρ_f is the density of fluid, b is the thickness of the airfoil in the direction perpendicular to the paper, and $I_a(\tau)$ is the inertial force due to the acceleration of the airfoil, which is defined as follows

$$I_a(\tau) = \rho S b a(\tau) \quad (3.53)$$

where ρ is the density of airfoil; since the buoyancy effects are not considered, $\rho = \rho_f$. S is the area of airfoil section; this is approximated by the area of ellipse having major and minor axes equal to the chord length and thickness of the foil. $a(\tau)$ is the acceleration. During the constant velocity translation part, $I_a(\tau) = 0$. The time evolution of lift and drag coefficients is shown in figure 3.28. Lift and drag start raising during the acceleration phase, and reach their peak values. After this, they fall down, and settle almost to a constant value during the constant velocity translation phase. It can be seen that the results produced from the simulations when using different integration schemes falls over a single line, and they are matching very well with the reported unsteady drag and lift coefficients [220].

In the second example, the hemisphere oscillates with velocity, $u(t) = -U_m \cos(2\pi ft)$, where U_m is the maximum velocity and f is the frequency of oscillation. The non-dimensional parameters, Reynolds number $Re = U_m D / \nu$ and Keulegan-Carpenter number $KC = U_m / fD$ are set to be 10 and 5, respectively. To define the parameters at these values, $U_m = 1$, $D = 1$, $\nu = 0.1$ and $f = 0.2$ are chosen. One cycle of oscillation is simulated i.e, until total time $T = 5$ with time step $\Delta t = 0.05$.

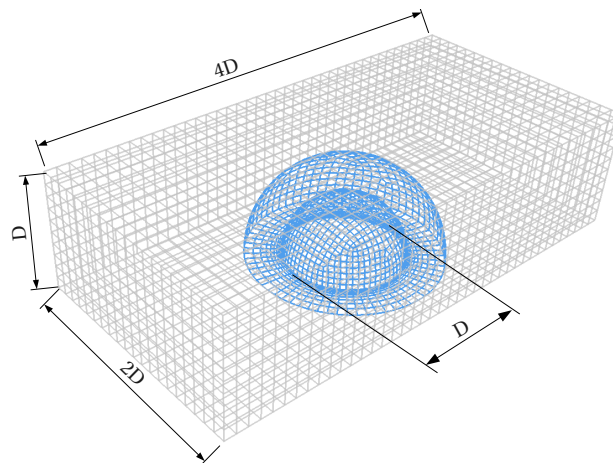


Figure 3.29: Background and embedded grid for flow past the oscillating hemisphere

This examples is solved using embedded mesh approach [28,221]. A structured grid is generated over the hemisphere by defining a larger hemispherical domain as shown in Figure 3.29. This grid, called as embedded domain, is embedded within the background domain discretized with a Cartesian grid. Both background and embedded domains are solved in a coupled manner using the procedure explained in [28,221]. This requires integration over polyhedral cells created at the intersection of both meshes. No slip boundary condition is applied on all but the bottom surface of the background domain.

At all time steps of the simulation, the solution computed using different integration methods are the same. In order to show this, the pressure field over the surface of the embedded interface is plotted at $t = 2.5$ in Figure 3.30. The quantities computed over the interface are directly affected by the error introduced by the integration methods, and since pressure is the most sensitive quantity in flow simulations, it can be concluded that the choice of integration method does not affect the solution accuracy for this problem.

The quadrature construction time and total simulation time are given in Table 3.6 and 3.7, respectively. As far as the quadrature construction time is concerned, all three methods behave the same way as in stationary interface simulations: the direct divergence method is the fastest

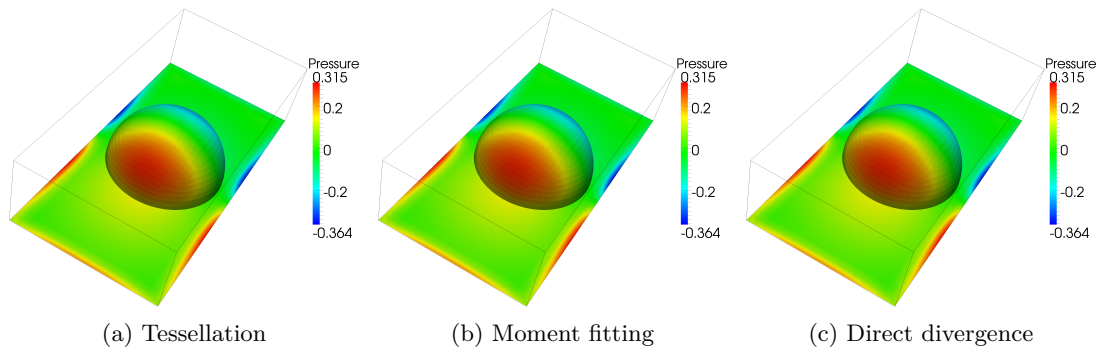


Figure 3.30: Pressure distribution over the embedded interface at $t = 2.5$ for flow past the oscillating hemisphere when using different integration methods

and the moment fitting method is the slowest, for reasons stated above. For this moving interface problem, moment fitting method does not outperform tessellation in total simulation time. This is because the quadrature construction is carried out at each time step, and the fact that the moment fitting takes longer for quadrature construction slows down its performance. The direct divergence method is 32.6% slower than tessellation and is 17.3% faster than moment fitting for airfoil problem; for hemisphere simulation, it is 12.7% and 8.05% slower than tessellation and moment fitting, respectively. The simulations reported here, and also the majority of the real world applications, involve a number of simple and complex polyhedra over which integration needs to be performed. For all the simple shapes, tessellation is more efficient than the direct divergence method as already shown for P4 in section 3.6.2. This is the reason why, though the direct divergence method outperforms tessellation in quadrature construction time, it is in many cases not more efficient in terms of the total simulation time. Direct divergence method is favorable when used for more complex polyhedral shapes.

Though, the direct divergence method is slightly slower, it excels through its superior robustness properties as it works for polyhedra of arbitrary complexity. From our experience, tessellation breaks down at times, and due to this the simulation crashes; as already stated in section 3.6.2, moment fitting is not accurate enough for certain polyhedra. The direct divergence method never breaks down, and it is accurate for any polyhedra. More importantly, it is much easier for computer implementation than tessellation and moment fitting.

3.6.7 Comparison with other methods based on the divergence theorem

Having discussed the efficiency of tessellation, moment fitting and the direct divergence method in the previous sections, it is very clear that the direct divergence method has superior accuracy and robustness properties than other reported methods. Hence, it is the preferred method in all our simulations. Having concluded on the choice of the integration method, it is now worthwhile to compare the efficiency of our direct divergence method with other methods based on the divergence theorem [190, 191] (referred to as similar methods).

As explained already in section 3.5.3.2, the present method is different from “similar methods” in the following aspects:

1. The concept of reference plane
2. Facet splitting instead of triangulation

The influence of these differences are given in Table 3.8. It shows that the present method is more efficient than similar methods, both in terms of quadrature construction time and total simulation time. The reasons for this are explained below.

	Quadrature construction (s)		Simulation time (s)	
	Present	Similar	Present	Similar
Stationary cylinder	0.0480	0.0666	1.487×10^4	1.654×10^4
Stationary star	0.2312	0.3005	10.72×10^4	11.59×10^4
Accelerating airfoil	42.992	90.522	2.130×10^4	3.896×10^4
Oscillating hemisphere	105.47	149.86	14.49×10^4	15.88×10^4

Table 3.8: Comparison of computational time for the quadrature construction and total simulation time for the present and similar methods

Due to the introduction of a reference plane in the direct divergence method, one or more facets that are on this plane are eliminated from the calculations. This means that on these facets, the following operations with regard to quadrature construction are eliminated: facet splitting, mapping main points to these facets, and obtaining internal Gauss points for these (unavailable) main Gauss points. Hence, the concept of reference plane leads to reduction in quadrature construction time. Moreover, elimination of these facets results in a reduced number of integration points as quantified in Table 3.1 and section 3.6.2, and as a result the total simulation time is reduced.

As explained in Figure 3.15 and the corresponding text in section 3.5.3.3, when using facet splitting, the number of integration points is smaller than when using triangulation procedure. This will obviously lead to increased computational efficiency, as shown in Table 3.8.

	Present	Similar
Stationary cylinder	0.06	11.31
Stationary star	3.57	11.98
Accelerating airfoil	32.63	142.6
Oscillating hemisphere	12.76	23.58

Table 3.9: Percentage increase in total simulation time when compared to tessellation

The percentage increase in total simulation time when compared to tessellation is quantified in Table 3.9. This Table shows that the modifications proposed in this work take the integration methods based on the divergence theorem one step closer to the tessellation.

Accuracy of the results produced by the direct divergence method and similar methods are exactly the same, because the above two modifications do not introduce any change in the underlying principle on which these methods operate. However, the modifications proposed in this paper leads to significant reduction in computational cost when compared to similar methods.

Note 1 In addition to the above said two modifications, the use of Gauss-Lobatto rules is considered to distribute main Gauss points on the integration facets. The number of integration points for triangle and quadrilateral in Lobatto rule are greater than those in the Gauss quadrature. The idea was that since some of the integration points in Lobatto rules are located on the edges, adjacent facets or adjacent cells generated from facet splitting can share these edge located points which would perhaps result in a reduced number of integration points. However, initial comparison over a few polyhedra indicated that the number of integration points is greater when using Lobatto rules than for Gauss quadrature.

Note 2 Integration of non-polynomials over polygons using the divergence theorem based method is presented in [190]. It has also been reported that their method works without any problems even for polynomials of degree in hundreds. Since the direct divergence method shares some ideas from [190], it is expected to exhibit similar behavior. Moreover, stability and error estimates are also derived for divergence theorem based methods in [190] for integration over polygons.

3.7 Closure

This chapter presented the mathematical formulations and the associated FE weak form of the Navier-Stokes equations discretized using an embedded interface method. The specific focus of the chapter was the accurate and robust numerical integration of weak forms over the elements that are cut by the interface. Existing numerical integration methods were reviewed and the shortcomings for each methods were identified. In order to circumvent the robustness issues associated with the existing methods, two new methodologies have been developed in this work: generalized moment fitting method and direct divergence method.

In the generalized moment fitting method, quadrature rules were developed to integrate polynomials over arbitrary polyhedra. The method is generalized in the sense that it is applicable for both convex and concave shapes. An efficient point distribution strategy was proposed and the quadrature weights at the corresponding points were obtained by solving the moment fitting equations. The integration of base functions in the moment fitting equations was accomplished with the help of the divergence theorem. Though the method works well in most instances, for a few polyhedral shapes it produced less accurate quadrature schemes. One of the main features of this methods is that the base functions can be chosen arbitrarily, and the quadrature rule is fit to integrate these base functions accurately. This could be very helpful in meshless methods, another class of methods that experiences problems with weak form integration. The problem in meshless methods stems from the fact that the integrands associated with the weak forms are not polynomials; they involve trigonometric or even exponential functions [118, 222–224]. An interesting study would be to choose appropriate base functions, and check if the resulting quadrature rule obtained from moment fitting equations can eliminate the weak form integration problem in meshless methods.

The direct divergence method, as the name implies, utilizes the divergence theorem to integrate the weak form integrand in FEM. It is more robust and accurate than other methods, and it is presented in a form that can be directly implemented in a large scale FE package. This method neither requires a complicated volume decomposition nor involves new mathematical formulations. It utilizes the divergence theorem, and the quadrature rules available for a line, triangle and quadrilateral to derive quadrature scheme for complex polyhedra. Contrary to the majority of existing methods of this kind, the direct divergence method does neither assume that the integrand is known nor does it involve symbolic computations. Hence it is extremely easy to implement. In our computations, it has never failed even in the instances where the tessellation method broke down, and hence it is the preferred integration method in all the computations presented in the next chapter.

The proposed integration methods are capable of evaluating integrals of polynomials over arbitrary complex shaped polyhedra accurately, and are generalized in a sense that they do not make use of any concepts specific to FEM. Owing to this fact, it is not only useful in embedded interface methods, but can also be utilized in polygonal and polyhedral finite element methods

[186, 188, 225–228], level set based methods [229–232], electronic structure calculations [233, 234], aerospace applications [235–237], computational geometry [181–183] etc.

4 Fluid-structure-fracture interaction

The main focus of this thesis is to introduce the possibility of fracture within the structure in FSI modeling. As a prerequisite to achieve this, a mesh refitting based fracture mechanics solver is devised in chapter 2, and the embedded interface finite element method to resolve fluid flow around moving interfaces is already described in chapter 3. The only question that remains to be answered is: how are these fluid and structure fields appropriately coupled to model FSFI effects? The answer to this question is precisely the focus of this chapter, to detail the partitioned coupling strategy used in this work.

This chapter is organized as follows. It begins with the summary of monolithic and partitioned coupling schemes, and explains why the partitioned approach is chosen for this work. Then, the governing equations for fluid and structure, together with the coupling conditions are presented. After this, the strongly coupled partitioned approach to simulate fluid-structure-fracture interaction is described. Dirichlet-Neumann partitioning is used in this work together with Aitken's Δ^2 method as convergence accelerator. Some numerical examples of increasing complexity are illustrated to demonstrate the applicability of the devised coupling scheme to solve complete fracture of structure due to the fluid loading. Finally, some unsolved issues of the present method are detailed.

4.1 Partitioned vs. Monolithic coupling

The numerical methodologies used to solve FSI problems, based on how the fluid and structure are coupled, are broadly classified into two classes: monolithic and partitioned schemes. For a detailed overview of further classifications, refer to [238]. The following text summarizes the advantages and shortcomings of monolithic and partitioned approaches, and explains why the partitioned coupling is preferred in this work.

In the monolithic approach, the governing equations of fluid and structure are solved simultaneously within a unified algorithm. The conditions that are to be enforced at the interface are treated implicitly. The main advantages of this approach are that it provides more accurate solutions and is numerically very stable. The major drawback is that the cost of solving the coupled system of equations is very high. Moreover, it involves very high initial development costs as the existing solvers for the fluid and structure part cannot be utilized.

In contrary, the two subproblems, fluid and structure, are treated separately using the respective numerical schemes in the partitioned approach. The communication between the two subproblems is achieved explicitly with the help of interfacial conditions. The major drawback of this approach is its lack of stability: even when the numerical methodologies used for both fluid and structure are stable and convergent, the resulting coupling scheme using this approach may be unstable. However, the partitioned approach offers a number of advantages. Primarily the already existing well-validated numerical methodologies for each field can be directly used. Moreover, it is possible to use different discretization strategies for each subproblem; but here, the finite element method is used for both fluid and structure. In addition, the use of partitioned strategy enables one to use different time step sizes for each subproblem.

Since the objective of the present work is to reuse the existing solution methods for fracture mechanics developed in chapter 2 and for fluid problems described in chapter 3, the partitioned approach is utilized to solve the coupled fluid-structure-fracture problem.

The partitioned approach is classified further into loosely coupled and strongly coupled approaches. In the former, the structure and the fluid governing equations are solved only once at each time step; it does not involve any iterative process. Since the interface conditions are not satisfied exactly, this scheme introduces interface errors which are accumulated over the subsequent time steps. Despite this drawback, this method is shown to be sufficiently accurate while dealing with compressible flows and for incompressible flows interacting with very stiff structures (for example aeroelasticity problems) [239–242]. However, the loosely coupled methods are not applicable for incompressible flows interacting with highly flexible structures; an instability called ‘artificial added mass effect’ appears and the simulations blow-up [110, 243–247]. The strongly coupled schemes involve an iterative process in which the fluid and structure fields are solved several times within a time step until the coupling conditions are satisfied. This scheme leads to solution equivalent to monolithic approaches, and can handle the interaction of incompressible flows with light weight structures. Hence, strongly coupled partitioned schemes are preferred in this work.

4.2 Problem definition

The complete definition of the coupled fluid-structure-fracture interaction problem consists of description of fluid and solid fields together with appropriate coupling conditions at the shared FSFI interface. In addition, this work assumes the points from which the crack starts propagating within the structure are predefined. An example FSFI configuration is illustrated in figure 4.1. A solid object Ω^s is enclosed by the fluid domain Ω^f , which is covered with a Cartesian mesh. The shared interface $\Gamma^i = \Gamma^c \cup \Gamma^s \setminus \Gamma_D^s$ defines the FSFI interface, where Γ^s is the boundary of Ω^s . Moreover, the blue dot on Γ^c denote the point of stress concentration, and hence a crack can propagate from this point. This work, though assumes the location from which crack starts to propagate, the crack path within the structure is assumed to be unknown a priori.

For completeness, this section recalls the already presented governing equations for fluid and structure, and describes the coupling conditions to be enforced on the FSFI interface.

4.2.1 Structure

The dynamic equilibrium of the structure, in the Lagrangian frame of reference, stated with respect to the reference configuration reads as,

$$\rho^s \ddot{\mathbf{d}}^s - \text{Div}(\mathbf{FS}) = \rho^s \mathbf{b}^s \quad \text{in } \Omega_0^s \times (0, T) \quad (4.1)$$

where ρ^s is the density of the structure, $\ddot{\mathbf{d}}^s$ represents the second time derivative of displacements, \mathbf{F} and \mathbf{S} are the deformation gradient and second Piola-Kirchhoff stress tensor respectively. \mathbf{b}^s is the external body force per unit mass and T is the end of considered time interval. For more details on these equations, the reader may refer to section 2.3.

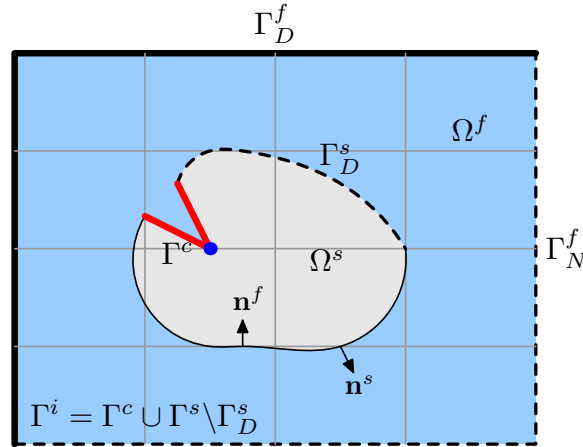


Figure 4.1: Notations used in FSFI coupling. Ω^f and Ω^s denote fluid and structural domain respectively. Thick red lines represents the preexisting notch and the blue dot denotes the point of peak stresses from which fracture gets initiated.

In order to solve the system, initial conditions on displacements \mathbf{d}^s and its first derivative $\dot{\mathbf{d}}^s$ are specified together with the appropriate boundary conditions.

$$\mathbf{d}^s|_{t=0} = \mathbf{d}_0^s \quad \text{in } \Omega_0^s \quad (4.2a)$$

$$\dot{\mathbf{d}}^s|_{t=0} = \dot{\mathbf{d}}_0^s \quad \text{in } \Omega_0^s \quad (4.2b)$$

$$\mathbf{d}^s = \bar{\mathbf{d}}^s \quad \text{on } \Gamma_D^s \times (0, T) \quad (4.2c)$$

$$(\mathbf{FS}) \cdot \mathbf{n}^s = \bar{\mathbf{h}}_f^i \quad \text{on } \Gamma^i \times (0, T) \quad (4.2d)$$

where $\bar{\mathbf{h}}_f^i$ defines the traction applied on the interface. It contains a subscript f to denote that this is not a user-defined value, but this traction is obtained from the fluid solution.

Crack propagation studies reported in chapter 2 assumed that the crack surfaces are traction-free. This is not the case in FSFI simulations as the crack surfaces experience traction from the fluid loads. This has implications on the procedure of estimating the vector J -integral as will be explained in section 4.3.

4.2.2 Fluid

The flow of an incompressible isothermal viscous fluid is governed by conservation of mass and momentum. In the Eulerian frame of reference, they are given by,

$$\nabla \cdot \mathbf{u}^f = 0 \quad \text{in } \Omega^f \times (0, T) \quad (4.3a)$$

$$\rho^f \dot{\mathbf{u}}^f + \rho^f \mathbf{u}^f \cdot \nabla \mathbf{u}^f + \nabla p^f - 2\mu \nabla \cdot \boldsymbol{\epsilon}^f = \rho^f \mathbf{b}^f \quad \text{in } \Omega^f \times (0, T) \quad (4.3b)$$

where \mathbf{u}^f , ρ^f , p^f and \mathbf{b}^f are the fluid velocity, density, pressure and externally applied body force respectively. The strain rate tensor is given by $\boldsymbol{\epsilon}^f = \frac{1}{2}[\nabla \mathbf{u}^f + (\nabla \mathbf{u}^f)^\top]$.

The governing equations are complemented by the following appropriate initial and boundary conditions,

$$\mathbf{u}^f|_{t=0} = \mathbf{u}_0^f \quad \text{in } \Omega^f \quad (4.4a)$$

$$\mathbf{u}^f = \bar{\mathbf{u}}^f \quad \text{on } \Gamma_D^f \times (0, T) \quad (4.4b)$$

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^f = \bar{\mathbf{h}}^f \quad \text{on } \Gamma_N^f \times (0, T) \quad (4.4c)$$

$$\mathbf{u}^f = \bar{\mathbf{u}}_s^i \quad \text{on } \Gamma^i \times (0, T) \quad (4.4d)$$

Here, the fluid velocity condition applied at the interface $\bar{\mathbf{u}}_s^i$ is dictated by the movement of the structure, and hence the subscript s .

4.2.3 Interface

FSI is a surface coupled problem: the different fields involved here, fluid and structure, interact through the interface surface Γ^i . It is clearly evident from equations (4.2d) and (4.4d) that the interface conditions to be applied on one field is dictated by the behavior of other field. The appropriate interface conditions, which include a kinematic and a dynamic condition, are enforced to ensure proper coupling of fluid and structure.

Kinematic condition: This enforces the no-slip condition on Γ^i . The kinematic condition arises from two physical constraints. Firstly, there can be no flow of fluid across the interface, which requires that the normal velocity components of fluid and structure must match at the interface. Secondly, since the present work deals with viscous fluids, the fluid cannot slip at the interface; in other words, the tangential movement of fluid must match that of the structure. These two constraints can be combined to state that the velocity of fluid and structure must be equal at the interface.

$$\mathbf{u}^f = \frac{\partial \mathbf{d}^s}{\partial t} \quad \text{on } \Gamma^i \times (0, T) \quad (4.5)$$

Dynamic condition: This ensures the force equilibrium at the interface, by requiring the surface traction from both fields to be equal

$$\boldsymbol{\sigma}^f \cdot \mathbf{n}^f = -\boldsymbol{\sigma}^s \cdot \mathbf{n}^s \quad (4.6)$$

The opposite sign is introduced to take into account for the fact that the normal vector for fluid and structure are acting in opposite directions (figure 4.1)

4.3 The strongly coupled partitioned approach

The coupling algorithm presented here is originally developed in [244, 248, 249] for FSI problems, and it is extended here to handle the FSFI. The method involves Dirichlet-Neumann partitioning and the coupling is achieved using fixed-point iterative methods employing Aitken's Δ^2 relaxation as convergence accelerator. The name Dirichlet-Neumann partitioning is derived from the nature of coupling: the Dirichlet condition for the fluid velocity on the interface is dictated by the deformation of the structure (equation (4.4d)), and the Neumann condition for the structure is given by the fluid loads acting on it (equation (4.2d)).

The complete coupling strategy when moving from time level t^n to t^{n+1} is described in Algorithm 6, and figure 4.2 is used to aid the understanding of the steps involved. For brevity,

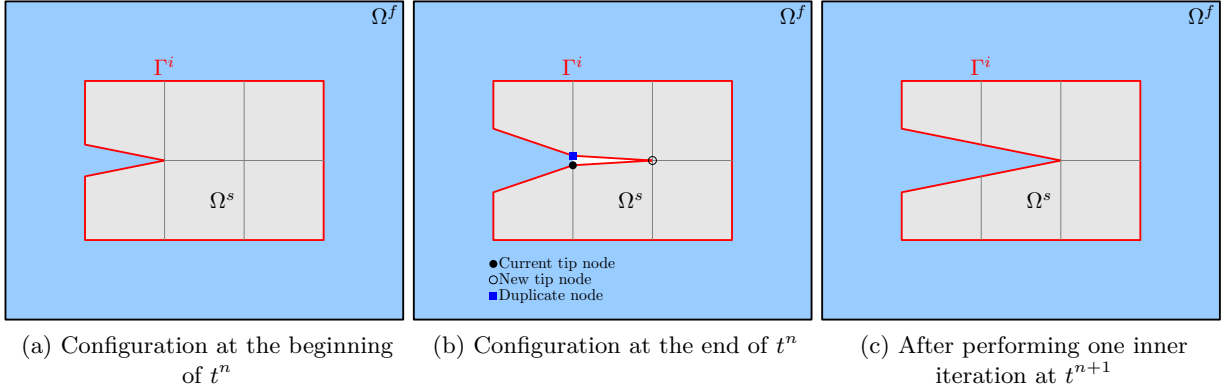


Figure 4.2: Fluid-structure-fracture interaction and the associated interface update. In (b) the crack opening is shown only for the visualization. Physically the crack faces are not moved apart at this instant.

the field quantities associated with inner iterations are marked without time level, for example \mathbf{d}_{it}^i actually implies $\mathbf{d}_{\text{it}}^{i,n+1}$. The method begins with initializing all field variables, and then it involves a predictor for interface forces. Here, it is simply set to be the converged value from t^n . The predictor does not have a strong influence in the present work as it employs strong coupling. However, the choice of predictor decides the temporal accuracy and moreover it dictates the onset of artificial added mass instabilities in loosely coupled partitioned schemes [247].

Structure

The next step is to solve the structural governing equations. The nonlinear system of equations resulting from the FE weak form can be written as

$$\mathbf{F}^s(\mathbf{d}^s) = 0 \quad (4.7)$$

Let the matrix form resulting from the linearization of the above equations be given as,

$$\mathbf{A}^s \mathbf{d}^s = \mathbf{f}^i + \mathbf{f}^e \quad (4.8)$$

The right hand side is conveniently split into two components. The interface forces \mathbf{f}^i are dictated by the fluid solution as given in the next subsection. \mathbf{f}^e accounts for any other external forces like body force. The above system is solved to obtain the structural displacements \mathbf{d}_{it}^s at current iteration level, from which the interface displacements \mathbf{d}_{it}^i are extracted. It is to be mentioned that the structure uses interface-fitted mesh (interface is embedded only in the fluid domain), and hence structural FE nodes are located on the interface, from which \mathbf{d}_{it}^i is obtained.

In order to enable the coupling, this interface displacement has to be transferred to the fluid system. This is achieved here in two steps. First, \mathbf{d}_{it}^i is used to move the interface within the fluid domain. The interface movement implies different cut configurations in the fluid domain (see figure 3.2) according to the position of Γ^i . In the second step, the interface velocity to be enforced ($\bar{\mathbf{u}}_s^i$ in equation (4.4d)) is extracted from \mathbf{d}^i as follows.

$$\bar{\mathbf{u}}_s^i = \frac{\mathbf{d}_{\text{it}}^i - \mathbf{d}_{\text{it}-1}^i}{\Delta t} \quad (4.9)$$

The above interpolation is only first order accurate. It is possible to design second order accurate schemes based on the trapezoidal rule. However, since such schemes are shown to be oscillatory [247], they are not used in this work.

Algorithm 6 Strongly coupled Dirichlet-Neumann partitioned algorithm

- 1: Initialize field variables: $\mathbf{d}_0^i = \mathbf{d}^{i,n}$; $\mathbf{d}_0^s = \mathbf{d}^{s,n}$; $\mathbf{u}_0^f = \mathbf{u}^{f,n}$; $p_0^f = p^{f,n}$
 - 2: Predict interface forces: $\mathbf{f}_0^i = \mathbf{f}^{i,n}$
 - 3: Set iteration count, $\text{it}=0$ and $\text{converged}=\text{false}$
 - 4: **while** ($\text{converged}=\text{false}$) **do**
 - 5: $\text{it}++$
 - 6: Solve structural equations $\mathbf{F}^s(\mathbf{d}^s) = 0$ to get interface displacements \mathbf{d}_{it}^i
 - 7: Update the interface location by displacing it with \mathbf{d}_{it}^i
 - 8: Compute interface velocity \mathbf{u}_{it}^i from \mathbf{d}_{it}^i using equation (4.9)
 - 9: Solve fluid equations $\mathbf{F}^f(\mathbf{u}^f) = 0$ to get \mathbf{u}_{it}^f and p_{it}^f
 - 10: Compute interface forces $\tilde{\mathbf{f}}_{\text{it}}^i$ from the fluid solution using equation (4.13)
 - 11: **if** Convergence criterion satisfied **then**
 - 12: Update solution fields:
 - 13: $\mathbf{d}^{i,n+1} = \mathbf{d}_{\text{it}}^i$; $\mathbf{d}^{s,n+1} = \mathbf{d}_{\text{it}}^s$;
 - 14: $\mathbf{f}^{i,n+1} = \tilde{\mathbf{f}}_{\text{it}}^i$; $\mathbf{u}^{f,n+1} = \mathbf{u}_{\text{it}}^f$; $p^{f,n+1} = p_{\text{it}}^f$;
 - 15: $\text{converged}=\text{true}$
 - 16: **else**
 - 17: Compute relaxation parameter ω_{it} using equation (4.16)
 - 18: Relax interface forces $\mathbf{f}_{\text{it}}^i = \omega_{\text{it}}\tilde{\mathbf{f}}_{\text{it}}^i + (1 - \omega_{\text{it}})\mathbf{f}_{\text{it}-1}^i$
 - 19: **end if**
 - 20: **end while**
 - 21: Perform crack propagation steps given in Algorithm 1 of Chapter 2
 - 22: Copy the values at new interface nodes appropriately
-

Fluid

The next step is to solve the governing equations of the fluid. The nonlinear system of equations resulting from the discretization of the Navier-Stokes equations can be written as

$$\mathbf{F}^f(\mathbf{u}^f) = 0 \quad (4.10)$$

where \mathbf{u}^f contains both pressure and velocity unknowns. Let the matrix form of the discretized fluid equations be given as,

$$\mathbf{A}^f \mathbf{u}^f = \mathbf{f}^f \quad (4.11)$$

It is to be mentioned that the above equation uses the interface velocity condition obtained from equation (4.9), and is solved to get \mathbf{u}_{it}^f and p_{it}^f .

The fluid solution is used to extract the interface forces. In the weak form of the structural equations, \mathbf{f}^i as introduced in equation (4.8) is calculated as

$$\tilde{\mathbf{f}}^i = \int_{\Gamma^i} \delta \mathbf{d}^s (\boldsymbol{\sigma}^s \cdot \mathbf{n}^s) d\Gamma \quad (4.12)$$

The $(\tilde{\cdot})$ denotes forces before performing relaxation. After applying the dynamic interface condition (equation (4.6)), the above equation can be rewritten as follows.

$$\tilde{\mathbf{f}}^j = - \int_{\Gamma^i} \delta \mathbf{d}^s (\boldsymbol{\sigma}^f \cdot \mathbf{n}^f) d\Gamma = - \int_{\Gamma^i} \delta \mathbf{d}^s (-p^f \mathbf{I} + 2\mu^f \boldsymbol{\epsilon}^f) \cdot \mathbf{n}^f d\Gamma \quad (4.13)$$

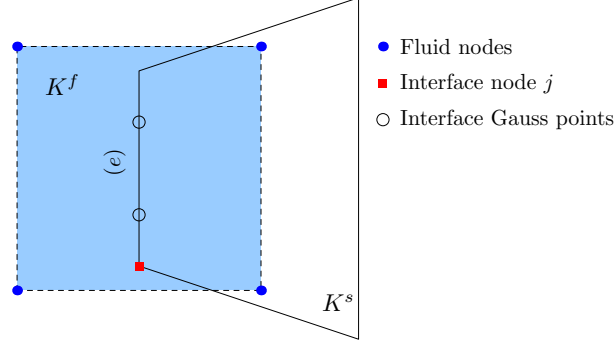


Figure 4.3: Illustration used to explain the calculation of consistent interface traction.

Interface forces are evaluated using the consistent force transfer method [102]. Let K^s denote a finite element in the structural domain which contains a part of the interface, i.e., $K^s \cap \Gamma^i \neq 0$. Consider the fluid element K^f that is intersected by the interface segment (figure 4.3). In terms of a matrix system, the contribution of the aforementioned term to node j of the interface element (e) can be written as

$$\tilde{\mathbf{f}}_j^{i,(e)} = - \int_{\Gamma^i} N_j^{(e)} (-p^f \mathbf{I} + 2\mu^f \boldsymbol{\epsilon}^f) \cdot \mathbf{n}^f d\Gamma \quad (4.14)$$

where N_j is the shape functions defined on the interface elements. This term is directly evaluated using the Gauss-quadrature rule defined on the interface elements. It involves interpolating the velocity and pressure available at the fluid nodes to the Gauss points and corresponding multiplication of integration weights.

The solution is said to be converged if $\|\mathbf{r}_{it}^i\| < \epsilon$, where the interface residual \mathbf{r}_{it}^i is given by

$$\mathbf{r}_{it}^i = \tilde{\mathbf{f}}_{it}^i - \mathbf{f}_{it-1}^i \quad (4.15)$$

If convergence is achieved, the field values are updated, and the algorithm goes to the next time step. If not, then the dynamic relaxation parameter ω_{it} is computed using Aitken's Δ^2 method [250]. The relaxation parameter is given by,

$$\omega_{it} = -\omega_{it-1} \frac{(\mathbf{r}_{it-1}^i)^\top (\mathbf{r}_{it}^i - \mathbf{r}_{it-1}^i)}{|\mathbf{r}_{it}^i - \mathbf{r}_{it-1}^i|^2} \quad (4.16)$$

For more details on this method, the reader can refer to [249]. Then, the interface forces are updated using this relaxation parameter.

$$\mathbf{f}_{it}^i = \omega_{it} \tilde{\mathbf{f}}_{it}^i + (1 - \omega_{it}) \mathbf{f}_{it-1}^i \quad (4.17)$$

The fluid and structural equations enforcing the coupling conditions are solved iteratively until the specified convergence criterion is met.

Fracture

After the converged solutions for fluid and structural fields are obtained, the crack propagation steps presented in Algorithm 1 of Chapter 2 are carried out. This involves the computation of vector \mathbf{J} -integral, checking whether crack propagation criterion is satisfied, and appropriately modifying the mesh to accommodate crack propagation through the structural domain. All these steps are elaborated in detail in section 2.5.

The computation of the \mathbf{J} -integral requires further discussion. In Chapter 2 it was assumed that the crack surfaces are traction-free. However, in FSFI, the crack surfaces are subjected to pressure and viscous forces due to their interaction with the fluid. In such a case, an additional term needs to be added to equation (2.43). The vector \mathbf{J} -integral becomes,

$$\mathbf{J} = \int_S \left(\left(\frac{\partial \mathbf{d}^s}{\partial \mathbf{X}} \right)^\top \cdot \mathbf{P} - W\mathbf{I} \right) \cdot \nabla_0(q) dS - \int_{\gamma_j^c} \left(\frac{\partial \mathbf{d}^s}{\partial \mathbf{x}} \right)^\top \cdot \mathbf{t}^c q d\gamma \quad (4.18)$$

where γ_j^c is the part of crack surface falling on the J -integral domain, \mathbf{P} is the first-Piola Kirchhoff stress tensor, W is the strain energy per unit volume, q is the support function, and \mathbf{t}^c is the externally applied traction acting on the crack surfaces, here $\mathbf{t}^c = -\boldsymbol{\sigma}^f \cdot \mathbf{n}^f$. The first term already appears in equation (2.43), and the second term accounts for the external traction on the crack faces. After applying the dynamic coupling conditions and performing a pull-back operation on the second term, the vector \mathbf{J} -integral can be established as,

$$\mathbf{J} = \int_S \left(\left(\frac{\partial \mathbf{d}^s}{\partial \mathbf{X}} \right)^\top \cdot \mathbf{P} - W\mathbf{I} \right) \cdot \nabla_0(q) dS - \int_{\Gamma_j^c} \left(\frac{\partial \mathbf{d}^s}{\partial \mathbf{X}} \right)^\top \cdot \mathbf{P} \cdot \mathbf{N}^s q d\gamma \quad (4.19)$$

where \mathbf{N}^s is the unit normal vector to the structural domain in reference configuration. One can refer to step 2 of section 2.5.2 for more details on the computation of \mathbf{J} .

If the \mathbf{J} -based crack propagation criterion (step 3 in section 2.5.2) is satisfied, the crack advances through the structural domain. Since the interface discretization in the present work is given by the trace mesh of the structural mesh, it is rebuilt after each crack propagation step. The rebuilding of the interface introduces two sets of new nodes, marked as duplicate and new crack tip nodes in figure 4.2(b). At these new nodes, the field values must be initialized appropriately. As far as $\mathbf{d}^{i,n+1}$ is concerned, the values at these new nodes are set from the corresponding nodal values of the structural discretization. For interface forces $\mathbf{f}^{i,n+1}$, the values at the duplicated nodes are copied from the current crack tip nodes, and $\mathbf{f}^{i,n+1} = 0$ at the new tip nodes. This is because of the following reason. At the instant of crack propagation, the old crack tip and the duplicated nodes are at the same location. Though the crack has propagated through the structure and the new crack surfaces are created, these surfaces have not yet moved relative to each other. This means that there is no opening for the fluid to fill. Moreover, the new crack tip node has not yet come into contact with the fluid flow, and hence the interface force is set to zero. This is the reason why in figure 4.2(b), the triangular region in the area of old and new crack tip nodes are not marked as fluid domain. However, in the next step, when structural equations are solved, the crack faces move relative to each other and the fluid starts to fill this opening (figure 4.2(c)). This work assumes that once the crack is propagated, the opening introduced is immediately filled with fluid. It means there is no fluid-lag as in hydraulic fracture.

This interface rebuilding process, and the subsequent geometrical cutting operations will be highly cumbersome when XFEM is used for crack propagation. This is because in XFEM, the crack is not aligned along the element edges and hence the trace mesh of the structural domain cannot be used directly as the interface discretization. Hence, XFEM is not preferred in this work for fracture.

4.4 Numerical examples

In this section, selected numerical examples of increasing complexity are presented to evaluate the performance of the proposed fluid-structure-fracture coupling algorithm. In the beginning, two simple examples are introduced to demonstrate the interface update procedure explained in the previous section and to check whether the simulation produces meaningful results even after the structure is completely broken into two distinct parts. After this, a thin filament undergoing FSFI is studied to present flow features within the crack opening area. Finally, a bending structure example is solved, in which after the complete fracture of the structure, one portion is allowed to undergo rigid body motion and gets convected by the flow.

In all the examples, the velocity is increased from zero to the given value using a smooth cosine profile. This serves two purposes: to obtain better convergence of flow solver and to apply smoothly varying load on the structure for enabling quasi-static crack propagation. The response of the structure under loading is modeled using the Neo-Hookean constitutive relations given in section 2.3.4. The critical J -integral (J_c) is assumed a very low value here, because in reality under static conditions the materials fail under fatigue. This means that the material withstands a large number of repeated cyclic loading before failure. Since it is assumed here that the failure occur with monotonically increasing load, a small value of J_c is assumed to enable the crack propagation through the material under given setting.

4.4.1 Failure of a structure supporting a static fluid

This simple test case is solved to check the implementation of the interface rebuilding process described in figure 4.2. The configuration is depicted in figure 4.4a, and it is the same as the fluid statics test case presented in section 3.6.1.

The structural domain in this example can be assumed to consist of two parts that are joint together with a weak adhesive along the dotted line. It contains a sharp notch (blue dot in figure 4.4a), which acts as the point of stress concentration and the crack gets initiated from this point. This structural domain supports a static fluid as shown in figure 4.4a. The fluid is subjected to a constant body force acting along the vertical downward direction. The velocities on the left and right boundaries of fluid domain are set to zero, and on the top boundary a “do-nothing” boundary condition is specified. This is because, when the structure fractures, the volume occupied by the fluid domain increases slightly. Since this is an incompressible flow simulation, in order to compensate for the change in volume, an equivalent amount of fluid should enter the domain. This is the reason why Dirichlet condition is not applied on the top boundary. As the initial condition, all velocities are set to zero throughout the domain.

Though the interface exhibits movement due to the fluid pressure, the interface velocity at all instances is set to zero. This make the present simulation to represent a fluid-statics example, despite the small amount of fluid flow entering the top boundary. In such a case, the fluid pressure increases linearly with the depth, and this imparts loading on the structure. As expected, stress concentration develops at the sharp notch, which leads to crack propagation through the

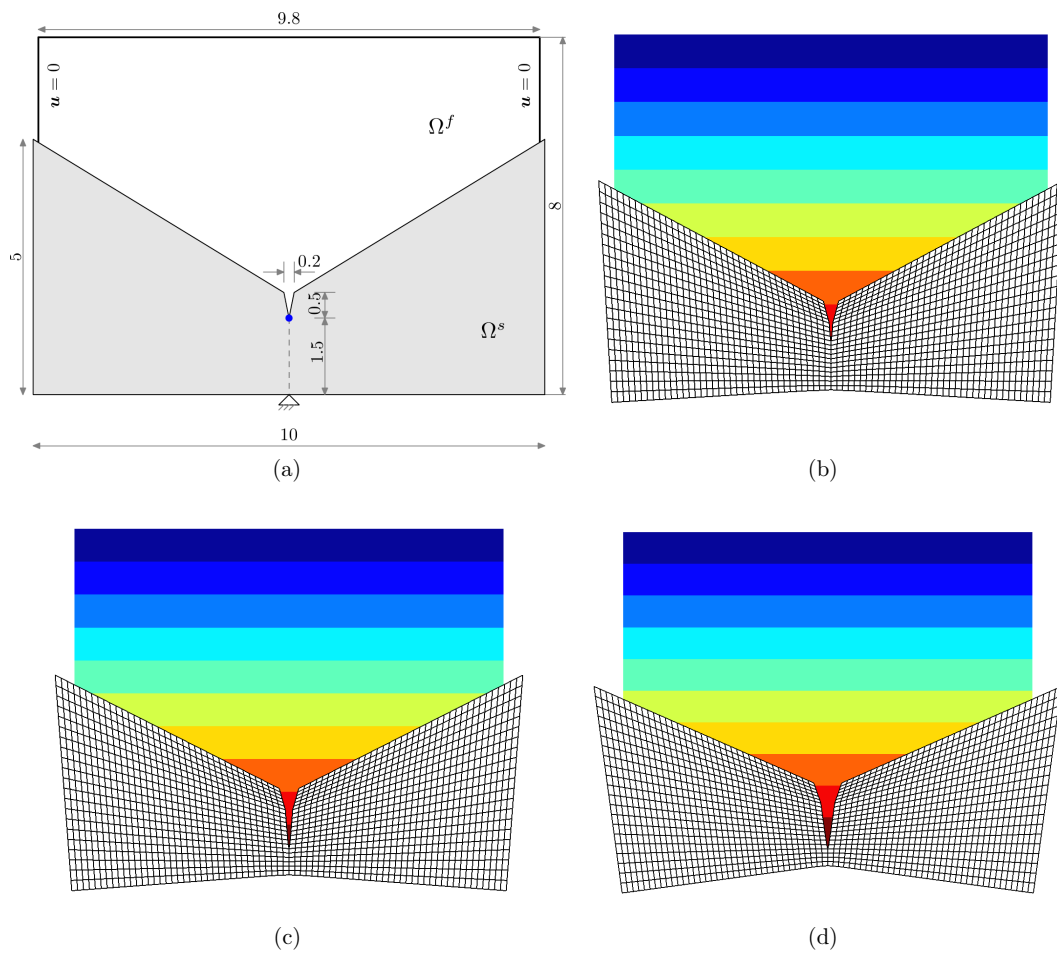


Figure 4.4: Failure of the structure supporting static fluid. (a) Simulation setup. (b)-(d) Palettes of pressure distribution over various stages of the crack propagation.

structure. In this example, critical J -integral is set to a very small value that allows crack propagation under given loading.

The results of the coupled fluid-structure-fracture modeling on this configuration are presented in figure 4.4(b-d). Owing to the fluid loading, crack starts propagating from the sharp notch. Because of the symmetry of the geometry and loading, the crack propagates vertically downwards. Further loading causes the relative movement of crack faces introducing an opening, and this opening is filled by the fluid. Since there are no external forces acting on the structure, it is clear that the fluid pressure interacting with the crack surfaces assist in opening of the crack surfaces. Moreover, it is directly evident from figure 4.4(b-d) that the linear variation of pressure with respect to depth is maintained at all instances, as expected for the fluid-statics problem.

4.4.2 Breaking fluid channel due to external loads

The last section presented an example in which the fluid pressure interacts with the structure, and drives the crack propagation through it. However, the simulation was highly simplified to represent a fluid statics problem. Moreover, the complete fracture of the structure was not modeled and the interface velocity was set to zero at all instances. The current example presents the

breakage of a fluid carrying channel subjected to external loading. Contrary to the previous test case, the interface velocity is dictated by the structural movement in this example. The objective here is to check whether the FSFI algorithm presents any difficulty when the structure is completely broken into two. The flow field characteristics associated with FSFI are not discussed here, but are delayed to the next two examples.

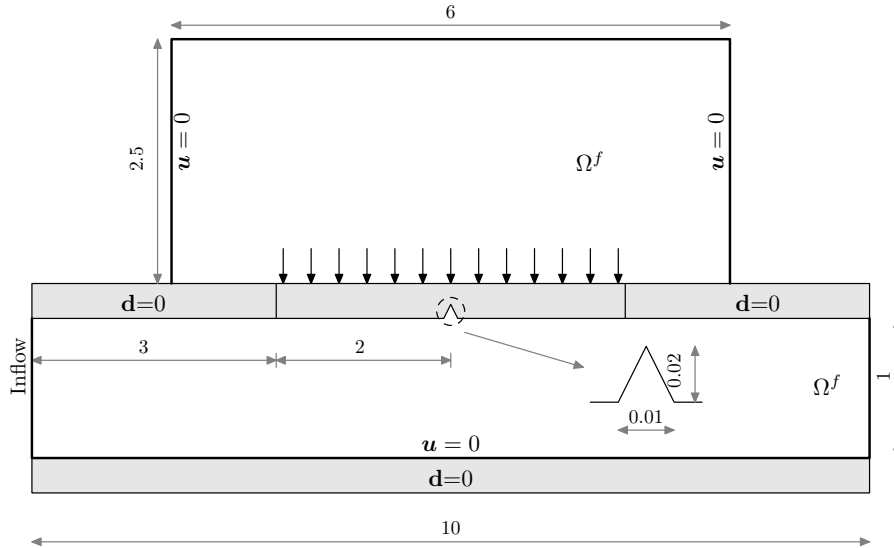


Figure 4.5: Computational setup for breaking channel example

The configuration of the current example is presented in figure 4.5. It involves solving the flow through a channel. The bottom wall and a large portion of the top wall are completely rigid, and only a region of the top wall is allowed to deform. This flexible region of the top wall is subjected to an external load as shown in figure 4.5. The bottom wall of the channel is not a part of FSFI interface; a mesh conforming to this wall is generated and a no-slip boundary condition is enforced strongly. Only the top wall represents the complete FSFI interface.

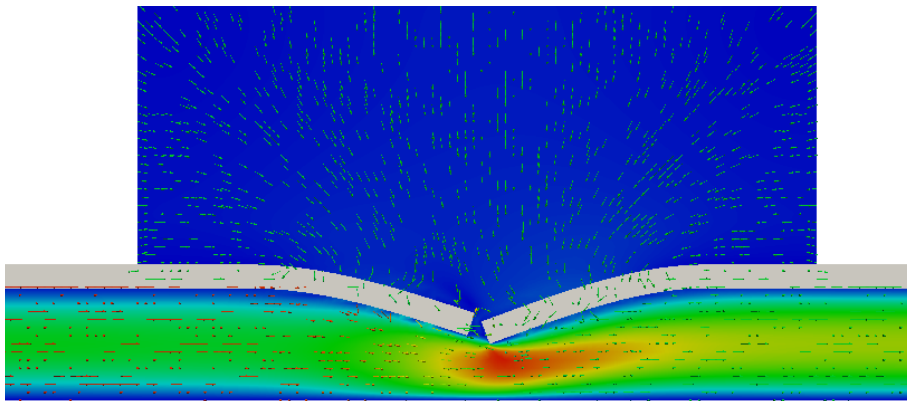


Figure 4.6: Palette of u_x and velocity vectors after the channel fractured completely

A parabolic fluid velocity profile is given at the inflow with $u_{\max} = 1$, and the fluid properties are $\rho^f = 1\text{kg/m}^3$, $\mu^f = 0.01\text{Ns/m}^2$. The material properties of the channel wall are: $E^s = 10000\text{Pa}$, $\nu^s = 0.3$ and $\rho^s = 50\text{kg/m}^3$.

The flexible portion of the top wall starts deforming and the deformation increases until the magnitude of J -integral reaches J_c . Once this condition is met, crack starts propagating from the notch. The crack advances through the structure continuously until it fractures completely. The velocity vectors are presented in figure 4.6, after the complete fracture of the structure. It can be seen that the FSFI algorithm proposed here works without any problems even after the structural domain and the interface are split into two.

The previous and the present examples, though involve FSFI effects, the detailed flow features near the crack surfaces are not discussed. In order to do so, the following two examples are solved to detail the flow field characteristics in the region of crack opening at various levels of crack propagation.

4.4.3 A filament fixed at both ends subjected to fluid loading

The last section presented the breaking channel example, in which the crack surfaces and the fluid flow interacted with each other. However, the fracture is not the result of the fluid loads acting on them but is induced by externally applied forces on the top surface of the pipe. In order to represent the capabilities of the present method to simulate fracture within the structure due to fluid loading, a thin filament subjected to fluid flow is considered.

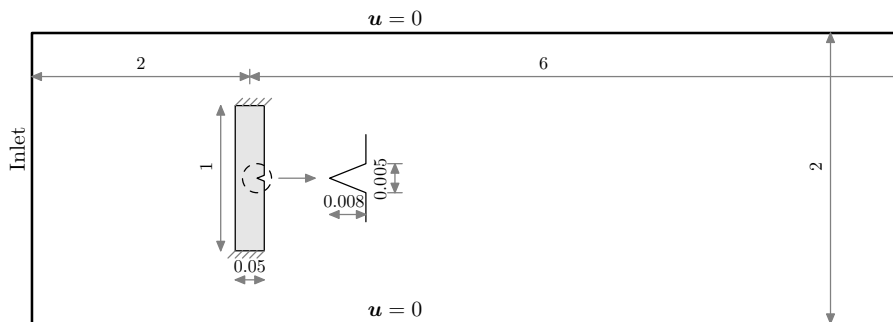


Figure 4.7: Computational setup for filament fracture example

The complete configuration is shown in figure 4.7. The filament, as in the other examples, contain a notch, which serves as the stress concentration point and the point of crack initiation. The notch geometry is symmetric with respect to the filament along the direction of the flow. It is fixed on top and bottom to arrest rigid body motion after the filament fracture. The material properties of the filament are: $E^s = 5000\text{Pa}$, $\nu^s = 0.3$, $\rho^s = 10\text{kg}/\text{m}^3$ and $J_c = 0.2\text{N}/\text{m}$. On the fluid domain, a parabolic inflow velocity profile with $u_{\text{max}} = 1$ is specified at the inlet. The top and bottom are no-slip boundaries. The fluid properties used in the simulation are $\rho^f = 1\text{kg}/\text{m}^3$, $\mu^f = 0.01\text{Ns}/\text{m}^2$. This corresponds to Reynolds number of $Re = 100$, where the length scale is the height of the filament.

Owing to the interaction with the fluid flow, the filament starts to bend from its initial configuration. As the flow velocity is increased smoothly, the bending and hence the magnitude of vector \mathbf{J} -integral also increases smoothly. Once $|\mathbf{J}|$ reaches its critical value J_c the two necessary conditions of crack propagation are satisfied: there is adequate strain energy available in the material to form new crack surfaces and sufficient stresses are acting on the notch. Hence the crack starts growing in the filament. Once new crack surfaces are formed, the subsequent loading will move these crack faces apart and introduces an additional area in the fluid domain. This sudden opening introduces low pressure zone within the crack region, and the fluid rush into the

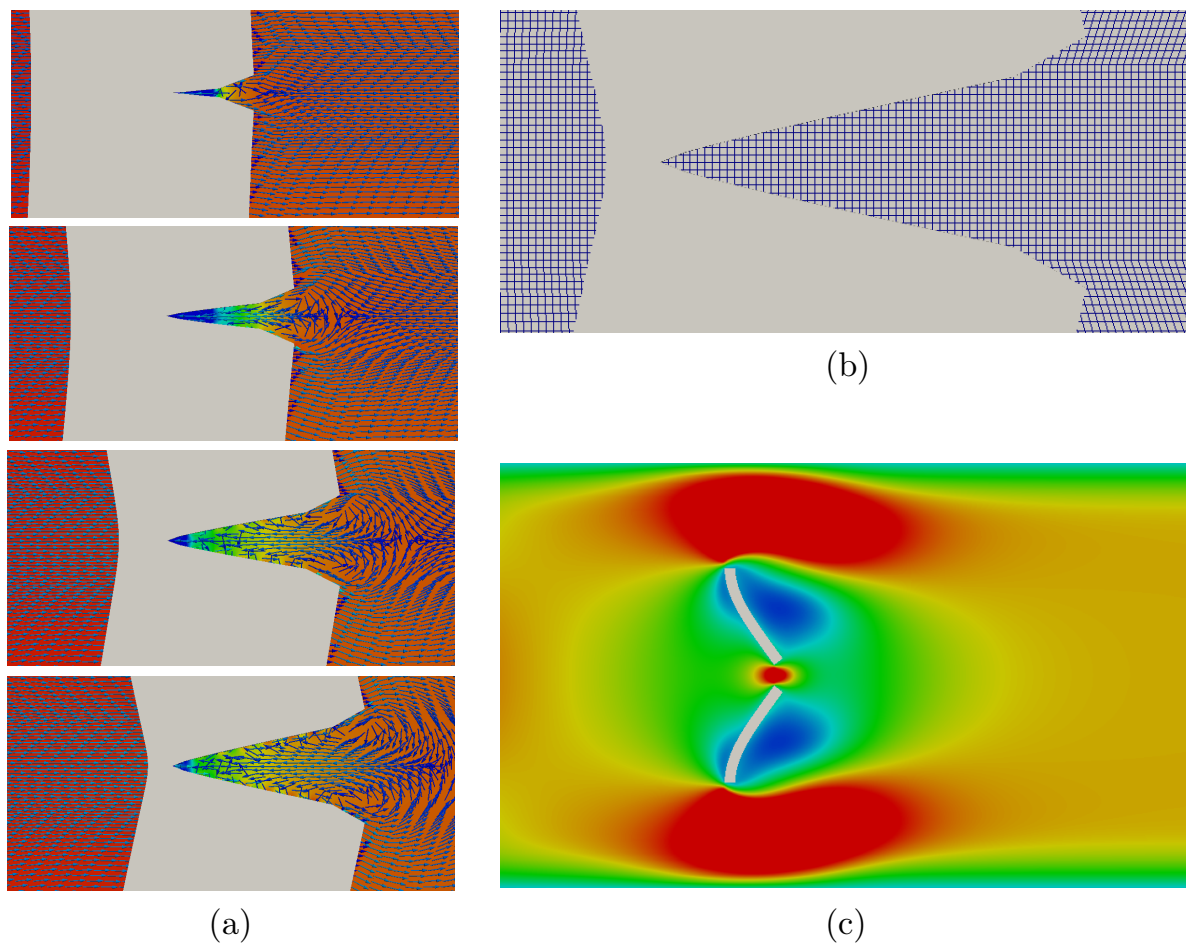


Figure 4.8: Results of filament fracture example. (a) Closeup view of pressure palettes and velocity vectors in the crack opening. (b) Closeup view of mesh in the crack region. (c) Palettes of u_x and velocity vectors after the filament completely broke into two and attain steady state.

opening, owing to the pressure difference. This can be seen clearly from figure 4.8a, which shows the zoomed view of pressure palette and velocity vectors for several levels of crack opening in the vicinity of crack surfaces. In order to capture this phenomenon, a fine mesh is used in the opening region as depicted in figure 4.8b.

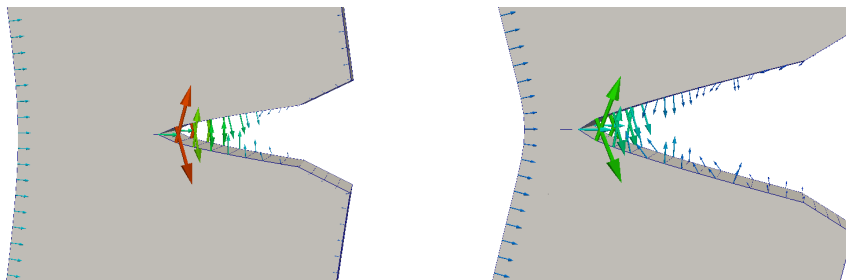


Figure 4.9: Interface traction acting on the crack surfaces.

The crack continuously advances through the filament owing to the fluid loads acting on it, and ultimately result in complete fracture of the filament i.e, the filament breaks completely into two. From this point, the problem is a pure FSI problem, and the filament continue to deflect due to the fluid forces until both the fluid and structure reach the steady state conditions. The palette of u_x at this steady state condition is plotted in figure 4.8c.

Another interesting question is: How does the fluid flow entering the crack opening region influence the subsequent behavior of the structure? In order to answer this question, the interface traction acting on the crack faces, imparted by the fluid flow are plotted in figure 4.9. Instead of assisting the crack faces to open further, the traction on the crack surfaces tend to bring the crack faces together. The reason for this is that the crack opening area is the low pressure region, and since the pressure dominates viscous effects at the considered Reynolds number of 100, the interface traction tries to close the crack faces. However, the fluid induced loads on all other parts of the interface are assisting the crack faces to open. Since the crack surfaces are a small part of the complete interface, the crack faces continue to open until the entire structure is fractured.

4.4.4 Fracture of a bending structure

The numerical examples presented in the previous sections have demonstrated the ability of the present method to model FSFI. However, they are simplified in the following two aspects. First, due to the symmetry of loading and boundary conditions, the crack propagation is of single mode. This is identified by the fact that the crack propagates either vertically or horizontally. Second, the Dirichlet boundary conditions were specified on the structural domain such that after the structure is broken completely into two, both parts are avoided to have rigid body motion. In this section, the bending structure example is presented which introduces both of these complex phenomena, which were not addressed in the previous examples, into the computation.

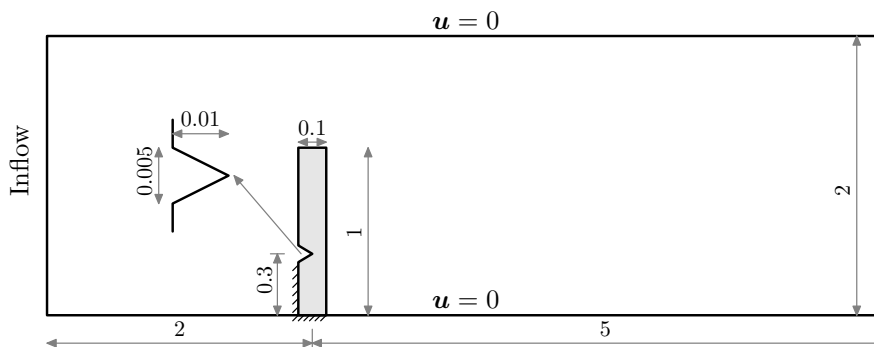


Figure 4.10: Geometry and boundary conditions for bending structure test case.

The computational setup for the present simulation is depicted in figure 4.10. A structure having a notch is set to face the fluid flow in this example. It is fixed at the bottom and over the portion of the left surface as can be seen in figure 4.10. Due to this non-symmetric boundary conditions, mixed-mode crack propagation is expected to occur through the structure. The material parameters are given as, $E^s = 5000Pa$, $\nu^s = 0.3$, $\rho^s = 10kg/m^3$ and $J_c = 0.2N/m$. On the fluid domain, the top and bottom surfaces are no-slip boundaries, and a parabolic velocity profile with $u_{max} = 1$ is specified on the left boundary. The density and the dynamic viscosity of the fluid are set to be $\rho^f = 1kg/m^3$ and $\mu^f = 0.01Ns/m^2$. This corresponds to the Reynolds number of $Re = 100$, where the length scale is the height of the structure.

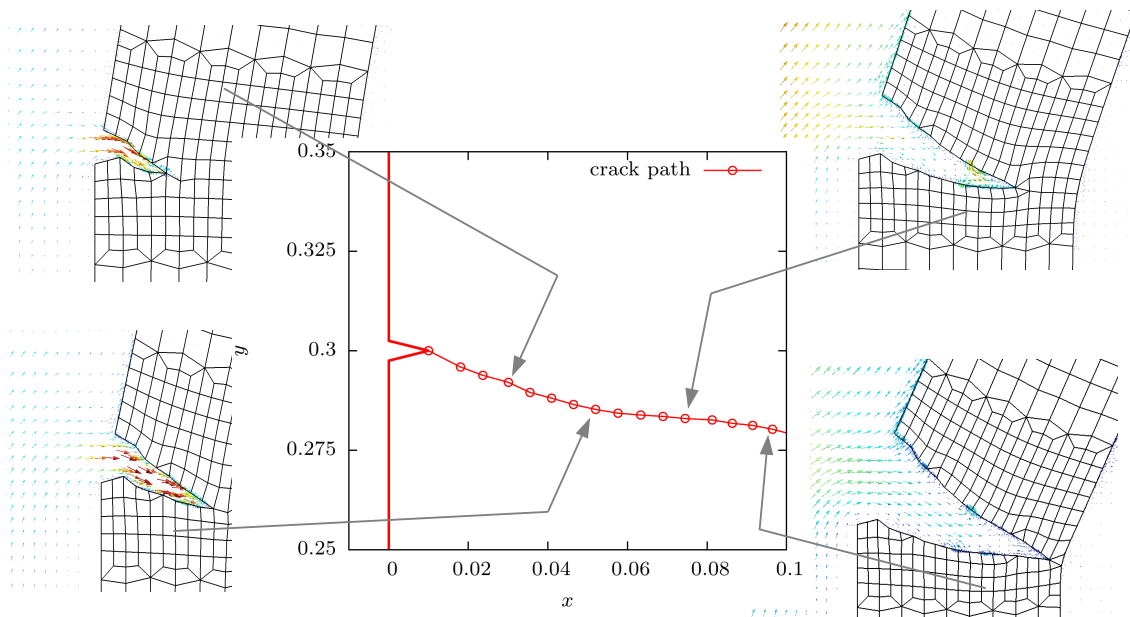


Figure 4.11: Mixed-mode crack propagation path through the bending structure and the corresponding fluid flow behavior. Trace of all nodal positions in reference coordinate system through which the crack tip moves are plotted to denote the crack path.

The structure starts bending, when the fluid flows past it. The bending deflection induces stress concentration at the notch tip and $|\mathbf{J}|$ starts raising with increasing deflection. As in other examples, once $|\mathbf{J}| = J_c$, since the crack propagation criterion is satisfied, crack starts propagating through the structure. However, unlike other examples, the notch is subjected to unsymmetrical structural boundary conditions and fluid loading. Hence a mixed-mode crack propagation occurs in this example. The crack propagates continuously until the structure is completely broken into two. The crack path and the associated fluid flow through the crack opening are illustrated in figure 4.11. The crack starts at the notch, and takes a slightly curved path and reaches the other end of the structure. The behavior of the structure is in some sense similar to the fracture test case presented in section 2.6.4, which considers the single edge cracked plate under mixed-mode loading. The crack path obtained from this example is consistent with the results in presented in section 2.6.4.

When the crack faces move apart, the low pressure region is formed inside the crack opening. This low pressure region creates sufficient driving force to attract the flow through the opening. The flow fields at various levels of crack propagation are depicted in figure 4.11. The interface forces behave the same way as in the previous example.

The crack traverses completely through the structure and split it into two separate portions. As can be seen from figure 4.10 that the structure is fixed only on the regions below the notch and once it is completely broken, the top portion is fully free to exhibit rigid body motion. This is the special feature of this example that after complete fracture, one portion is carried away by the flow. Figure 4.12 plots, at a few instances, the location of the structure when it is exhibiting rigid body motion and the corresponding palette of u_y .

The present example involves a number of complex phenomena: mixed-mode crack propagation through the structure, fluid starts to fill the crack opening, complete fracture of the structure

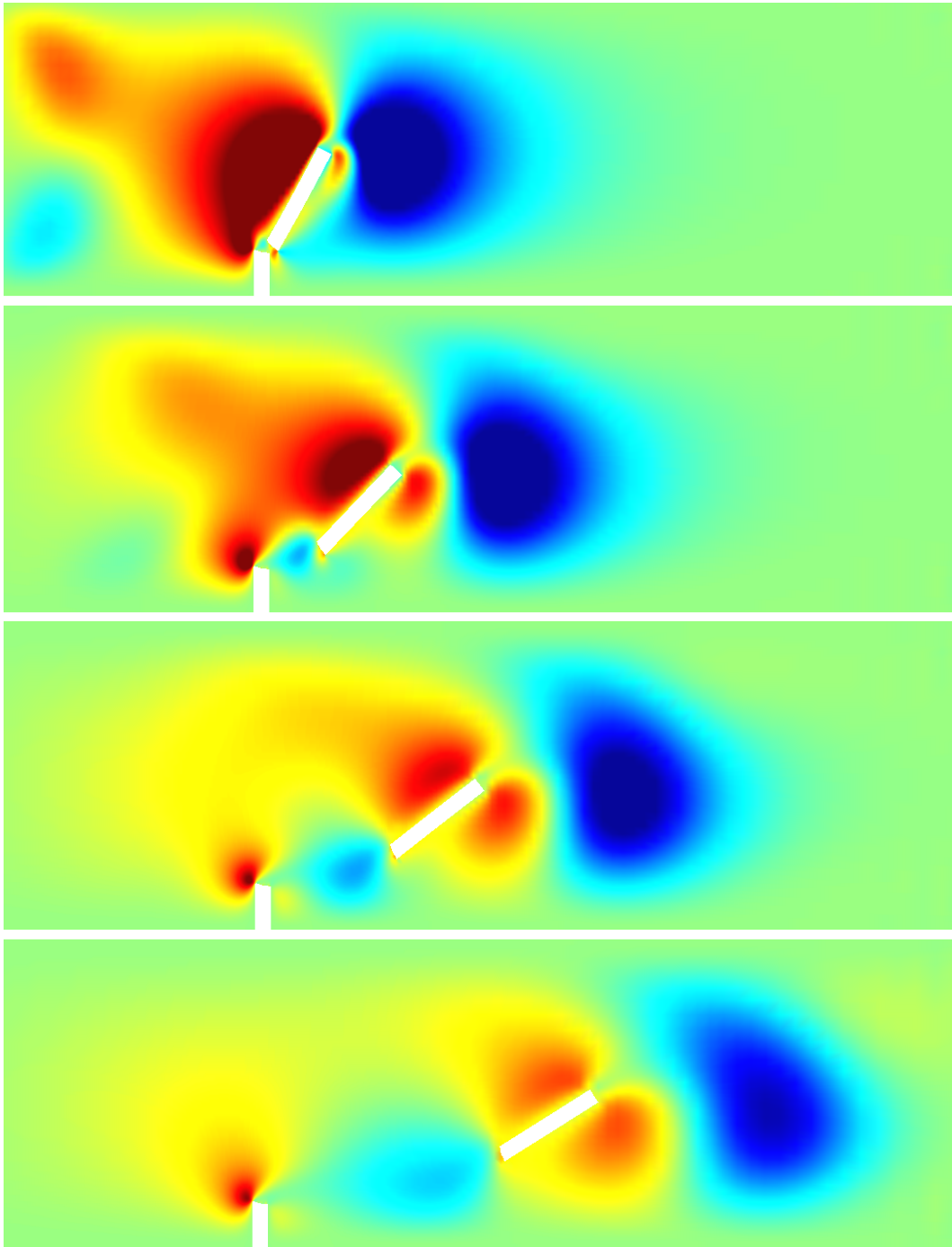


Figure 4.12: Fracture of the bending structure: Palettes of u_y when one part after fracture gets convected by the flow.

into two pieces of which one is carried away by the flow. It can be seen from the presented results that the present method is capable of capturing all these phenomena.

4.5 Unsolved issues

The examples presented in the previous section have demonstrated the capabilities of the current method to simulate fluid-structure-fracture interaction. The present method is clearly an improvement over the existing FSI solvers that assume that the structure deforms only within the elastic limit. However, the solutions presented in the previous examples contain two important issues that are yet to be addressed. In order to explain them, the close up view of pressure palettes near the crack opening for filament fracture problem are considered as shown in figure 4.13.

The first issue is concerned with the temporal pressure oscillations observed in the solution immediately after the complete separation of the structure. The reason for these oscillations are explained as follows. When the crack opens, the sudden movement of the crack faces introduces an opening and the pressure values inside this opening take very low values. Consider the pressure field just before the body completely breaks into two (figure 4.13a). Inside the crack surfaces, the pressure values are very low, and in the neighborhood region that is facing the flow, the pressure values are very close to stagnation pressure, which is the highest pressure in the fluid domain. These two regions are separated by the portion of the structure which is yet to be fractured. When the body breaks into two, these two regions are merged (figure 4.13b). This introduces a very strong local pressure gradient within the fluid domain. This strong local gradient causes the temporal pressure oscillations within the domain after the body completely breaks into two. How to suppress these oscillations are not clear now, and needs further attention.

The spatial oscillations of fluid solution in the region of crack surfaces is the second issue (figure 4.13b). At present, the reason for this oscillations is not clear. One factor that influences this could be the use of the semi-Lagrangian time integration described in section 3.4. Since the present study employs very fine mesh in the neighborhood of crack opening, many times the interface crosses many elements in one time step. In such cases, it has been observed in pure fluid simulations that the smoothness of the solution is negatively affected. Moreover, this special time integration involves tracing back the Lagrangian origin of the current node at the previous time step. Though physically the Lagrangian origin of a node located in the fluid domain at current time step must be in the fluid domain at the previous time step, in the discrete setting, this is not always achieved. This means there are some fluid nodes for which the Lagrangian origin is located within the structural domain. For these nodes, the velocity values are simply set to zero, which also introduces some error in the solution. However, to what extent this error in time integration affects the solution accuracy is not yet clear and requires further investigation.

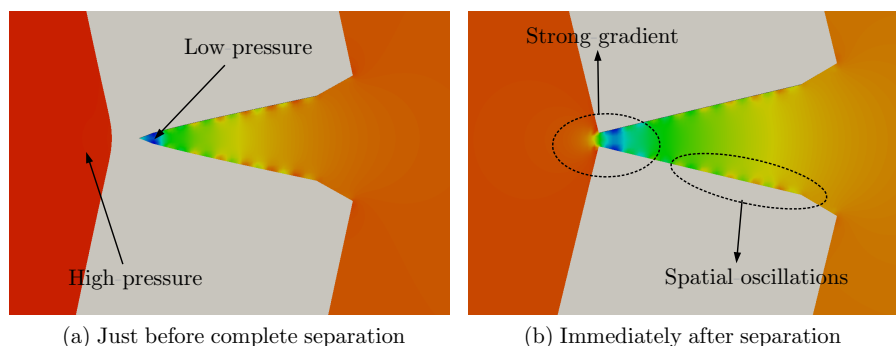


Figure 4.13: Pressure palettes near the crack surfaces for filament fracture problem.

4.6 Closure

This chapter detailed the coupling method used in the proposed fluid-structure-fracture interaction solver. Since the objective of the present work is to reuse the crack propagation approach developed in chapter 2 for fracture and the embedded interface fluid formulations described in chapter 3, the partitioned scheme is chosen for the coupling. The governing equations of fluid and structure together with the interface coupling conditions are recalled in this chapter, before presenting the coupling algorithm. The proposed FSFI method uses the strongly coupled partitioned scheme with Aitken's Δ^2 method as the convergence accelerator. The interface rebuilding process and the computation of vector J -integral for loaded crack surfaces were emphasized. Simulations of varying complexity were presented to demonstrate the ability of the proposed method to capture FSFI effects. The most complex examples involved the fracture of a bending structure subjected to fluid loads. This involved a number of complex phenomena: it induces mixed-mode crack propagation through the structure, fluid starts to fill the crack opening, complete fracture of the structure into two pieces of which one is carried away by the flow. Finally, some unsolved issues in the current method were described.

5 Summary and Outlook

5.1 Summary

The central focus of this thesis is the development of a computational methodology for simulating fluid-structure-fracture interaction (FSFI) phenomena. Precisely, this work is aimed towards modeling the quasi-static fracture of hyperelastic structures due to the fluid loads acting on them, in scenarios where the structure exhibits large deformation and the fluid-structure interaction (FSI) effects are significant. The present study has invoked the assumption that the fracture is brittle, and the fluid flow is modeled as incompressible.

FSI has gained great attention in the past decades because of its importance in a variety of natural phenomena and engineering applications. Almost all of the existing FSI studies assume that the structure deforms completely within the elastic limit, which precludes any plastic deformation and failure by fracture. Though studies under this assumption are very useful in a number of practical applications, there are scenarios in which the FSI loads acting on the structure reach a very high level and subsequently lead to fracture. This phenomenon in which the fluid loads result in crack propagation, in addition to deflection of the structure, is termed in this thesis as fluid-structure-fracture interaction. The present work is focused on developing an embedded interface finite element method to simulate FSFI.

The contributions of this thesis are three-fold. First, on the structure part, a mesh refitting approach is devised to simulate the quasi-static brittle fracture within nonlinear hyperelastic materials. Then, on the fluid part, accurate and robust numerical integration methods are developed for enhancing the robustness of embedded interface methods (EIM) to handle fluid flow past arbitrarily moving/deforming bodies. Finally, on the coupling part, the interaction of fluid flow with cracking structures is modeled using a strongly coupled partitioned approach. In the following, the aforementioned contributions are detailed.

The first step in the development of a computational methodology to handle FSFI is to equip the structural analysis with a fracture mechanics solver. In this work, a mesh refitting approach is developed to simulate single and multiple crack propagation through an isotropic homogeneous hyperelastic material. Each time when the crack propagates, it induces topology change in the structural domain. In order to deal with this, the initial mesh covering the structure must be adjusted continuously. The majority of the existing methods utilize adaptive remeshing strategies to deal with this issue. However, for practical reasons related to the computational costs associated with FSFI simulations, adaptive methods are not preferred in this work. Instead, a two step mesh-modification algorithm is proposed. In the first step, the computational nodes are repositioned in such a way that the crack in the new mesh propagates along an existing edge in the mesh. In the next step, if necessary, the element connectivity is modified locally to enable arbitrary crack paths through the mesh. The excellent performance of this method is demonstrated by simulating crack propagation examples of varying complexity and comparing the results with those available in the literature.

One of the crucial steps that dictates the accuracy and robustness of EIMs is the accuracy of weak form integration. In contrast to the conventional body-fitted methods, the interfaces in EIMs are not aligned along the edges of the background mesh i.e., the interface cuts the elements

of the background mesh in an arbitrary manner. Over these cut elements the weak form must be integrated in the region that is located within the fluid part. This requires accurate numerical integration strategies that work for arbitrary shaped volumes. The most widely used volume decomposition based methods lack robustness i.e., they fail at certain cut configurations and the simulation crashes. To address this issue, two new numerical integration strategies, namely the generalized moment fitting method and the direct divergence method are constructed in this work. In the first method, generalized quadrature rules are developed to integrate polynomials over arbitrary polyhedra. An efficient point distribution strategy is proposed and the quadrature weights at the corresponding points are obtained by solving the moment fitting equations. In the direct divergence method, as the name implies, the divergence theorem of multivariable calculus is used. Both methods are designed to work in situations, where the integrand is not explicitly prespecified, which is essential for their applications in finite element methods. Moreover, they can handle convex as well as concave shaped polyhedra. Numerical examples demonstrate that the direct divergence method is the most accurate and posses superior robustness characteristics when compared to all other available methods.

Having dealt with fracture and the robustness of fluid formulations, a coupling approach is developed to model the fluid-structure-fracture interaction in the final step. As stated before, this enables the possibility of simulating fracture within the structure owing to the FSI loads. Since the objective is to use the existing methodologies for fracture and fluid field, and because this work considers highly flexible structures, a strongly coupled partitioned approach is utilized. The simple Dirichlet-Neumann partitioning, in which the Dirichlet condition for the fluid velocity on the interface is dictated by the deformation of the structure, and the Neumann condition for the interface traction is given by the fluid loads acting on it, is used to couple the structure and fluid fields. The convergence is accelerated using dynamic relaxation employing Aitken's Δ^2 method. Numerical examples demonstrate that the developed method can handle complete fracture of the structure due to fluid induced FSI loads. An Example involving a structure breaking completely into two, of which one part is carried away by the flow is also presented.

In short, this thesis has taken the numerical modeling of FSI to the next step by introducing the possibility of fracture within the structure due to the fluid induced FSI loads. To be more precise, the aims are to

- enable the possibility of crack initiation and propagation within the structure due to FSI loads
- model the mutual interaction of crack surfaces and the fluid entering the crack opening

These aims are achieved by developing an embedded interface finite element methodology to handle fluid-structure-fracture interaction which combines the knowledge from three different fields: structural mechanics, fluid dynamics and fracture mechanics.

5.2 Outlook

As already mentioned in several places within the thesis, the present work is the first step towards modeling FSFI in biological applications. In order to enable the proposed method to deal with practical scenarios, the following essential extensions need to be considered.

The first issues to be addressed are the temporal pressure oscillations observed in the domain when the interface completely broke into two, and the spatial oscillations in field variables appearing in the crack opening region. The reasons for these oscillations are explained in section 4.5.

The most important future work is to carry out detailed investigations on how to suppress these oscillations.

The computational methodology developed in this thesis is applicable only for structures that fail by monotonically increasing load. The majority of the structures in engineering and nature fail by fatigue. The failure occurs when they are subjected to a large number of repeated loading cycles even when the maximum stress experienced is far below the strength of the material. In order to apply the proposed method to real biological applications, the possibility of modeling fatigue failure must be introduced in the fracture module. While progressive failure due to fatigue occurs, the crack faces move apart and come together owing to the cyclic loadings. This means that in each loading cycle the crack faces come into contact with each other which necessitates the use of computational contact mechanics [251].

The present work is limited to the fracture of time-independent rate-independent hyperelastic materials. However, almost all of the biological materials are viscoelastic in nature. In such materials, the stresses experienced are dependent on the rate of loading. Hence, in order to broaden the applicability of the proposed method, the fracture of viscoelastic materials should be included.

Another possibility is to extend the presented strongly coupled partitioned FSFI method to a monolithic framework. In biological applications, the density of the structure is equal to that of the fluid. In such cases, an increased number of inner iterations is required for the convergence of the strongly coupled schemes or even convergence may not be obtained. This is observed for the bending structure example presented in section 4.4.4. In order to avoid this, the whole problem can be recast into the monolithic framework [125, 252], in which the governing equations of fluid and structure are solved simultaneously within a unified algorithm. The conditions that are to be enforced at the interface are treated implicitly, which results in more accurate and stable computational approach.

The aforementioned potential extensions enable the proposed methodology to be useful to model FSFI in practical applications. In addition, the computational methodologies for fracture and the numerical integration techniques developed in this thesis can be enhanced for general applications.

The mesh refitting crack propagation approach provides a valuable basis for further research and development in fracture mechanics, and the following future improvements of this method are possible.

Though the mesh refitting approach assumes that the crack path is a priori unknown, a preexisting notch or a crack is assumed to exist. This notch or crack provides a clearly defined point of stress concentration and the crack starts propagating from this point. In other words, the crack initiation is not modeled in this work, only the propagation of a preexisting flaw is simulated. The proposed approach can be extended to model crack initiation using cohesive elements [59–63]. These elements are added at the interface between each neighboring finite elements, the response of which are given by the traction-separation law. When the structure is subjected to sufficient external loads, these cohesive elements break and give rise to crack within the material.

Despite significant advances in fracture mechanics, the computational modeling of rubber-like materials still remains as an open problem. The present method utilizes a nonlinear compressible hyperelastic material and the comparison of present results are consistent with those reported for rubber-like incompressible materials [92]. Moreover, section 2.6.9 reports crack propagation through the material that experiences large strain to such an extent that the crack tip is always blunt. Not many existing computational methods can model the crack propagation in such scenarios. A valuable extension is to include an incompressible material model within this approach to enable the mesh refitting method capable of modeling the fracture in rubber-like materials.

Brittle fracture is the focus of the present work. This inherently assumes that all the dissipative effects are confined to a very small region in the vicinity of crack tip and hence are neglected. The majority of the materials used in engineering applications are ductile, in which the bulk plastic dissipative effects are significant. An interesting extension of the present work is to combine the mesh refitting method with the approach developed at our institute [253] to model plasticity in finite strain region. This will enable the applicability of crack propagation approach to deal with ductile fracture.

The numerical integration methods developed in this thesis, namely the generalized moment fitting and the direct divergence methods, are applicable only for linear polyhedra. Each facet of the polyhedra is assumed to be a plane surface which is in turn bounded by straight lines. In order to deal with warped fluid elements in the embedded interface methods, these integration schemes need to be extended to perform integration over volumes that are bounded by curved surfaces. Another potential application of such an extension is the multiphase flow problems with higher order levelset formulations. However, before attempting to extend the numerical integration methods, the algorithm used to perform geometrical cutting operations has to be enhanced to deal with curved surfaces.

A A simple crack propagation approach for linear elastic fracture mechanics

When one simulates fracture problems using linear FEM, steps 2–4 given in the mesh refitting procedure (section 2.5.2) can be replaced with the following simpler steps. This alternative procedure involves displacement correlation technique to compute stress intensity factor, and corresponding crack kinking and crack propagation criterion.

Step 3: Stress intensity factor computation

As already stated in section 2.2, stress, strain, displacement, and crack propagation direction are all decided by a single parameter in LEFM theory. This single crack tip characterizing parameter is termed as stress intensity factor (SIF). SIF quantifies the strength of singularity at the crack tip, and the extraction of SIF from FE data is essential to decide whether the crack propagates or not, if at all it propagates in which direction it advances.

There are a number of available methods used to estimate SIF from finite element solution, of which the displacement correlation technique (DCT) [56,58,254–256] is the most simple method. Displacement correlation technique directly interpolates the nodal displacements obtained from FE solution to obtain SIF. These methods are extremely easy to implement as no additional procedure is mandatory. Though they require sufficiently fine mesh in the crack tip vicinity [58], they are still used by several researchers and as noted in [256] most commercial FE codes uses this technique. Moreover, this method does not require any extra-work in case when crack faces are applied with external traction. This is very important since in case of fluid-structure-crack interaction, owing to the fluid loading, traction is always applied on the crack faces.

The stress intensity factors are given by [86],

$$K_{\text{I}} = \frac{E}{(1 + \nu)(1 + \kappa)} \sqrt{\frac{\pi}{2r}} (\mathbf{d}_{t_{node2}} - \mathbf{d}_{t_{node1}}) \quad (\text{A.1a})$$

$$K_{\text{II}} = \frac{E}{(1 + \nu)(1 + \kappa)} \sqrt{\frac{\pi}{2r}} (\mathbf{d}_{n_{node2}} - \mathbf{d}_{n_{node1}}) \quad (\text{A.1b})$$

where r is the distance between tip node and node 1 (or 2), E is the Young's modulus, ν is Poisson's ratio, \mathbf{d}_t and \mathbf{d}_n are displacement components along the direction t and n given in figure A.1. κ is given by,

$$\kappa = \begin{cases} 3 - 4\nu & \text{plane strain} \\ \frac{3-\nu}{1+\nu} & \text{plane stress} \end{cases} \quad (\text{A.2})$$

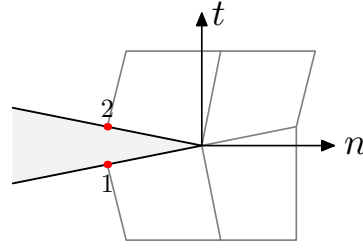


Figure A.1: Displacement correlation technique

Step 4: Check crack propagation criterion

After extracting the SIFs, the next step is to decide whether crack propagates or not. This can be done using the SIF-based mixed-mode criterion, expressed as

$$\left(\frac{K_I}{K_{Ic}}\right)^\alpha + \left(\frac{K_{II}}{K_{IIc}}\right)^\beta = 1 \quad (\text{A.3})$$

where K_{Ic} and K_{IIc} denote the critical SIFs in corresponding modes. The critical values are called fracture toughness, and in LEFM, they are considered to be material properties. These values are specified as input to the simulation.

The constant parameters α and β should be determined empirically for each configuration. However, since it is not feasible in practice, we use $\alpha = 2$ and $\beta = 2$ which leads to the well-known criterion proposed in [257]. If the crack propagation criterion is not satisfied, then the crack does not propagate at this time step, all the remaining operations are not performed and the algorithm goes to step 1 of next time level.

Step 5: Determine crack kinking criterion

The next step involves the computation of crack kinking angle using maximum circumferential stress criterion (MCSC). In this criterion, the crack propagates in the direction in which the circumferential stress is maximum in the region close to crack tip. It is proposed by Erdogan [97] for 2D mixed-mode fracture problems, and subsequently extended to 3D configuration. The following derivation follows from [56]. The stresses at the crack tip for mixed-mode problems are given by adding the stresses corresponding to individual modes that are written as,

$$\sigma_r = \frac{1}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left[K_I \left(1 + \sin^2 \frac{\theta}{2} \right) + \frac{3}{2} K_{II} \sin \theta - 2K_{II} \tan \frac{\theta}{2} \right] \quad (\text{A.4a})$$

$$\sigma_\theta = \frac{1}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left[K_I \cos^2 \frac{\theta}{2} - \frac{3}{2} K_{II} \sin \theta \right] \quad (\text{A.4b})$$

$$\tau_{r\theta} = \frac{1}{2\sqrt{2\pi r}} \cos \frac{\theta}{2} [K_I \sin \theta + K_{II} (3 \cos \theta - 1)] \quad (\text{A.4c})$$

The above equations are valid for both plane stress and plane strain problems. As per MCSC, the crack tends to propagate along the plane on which σ_θ is maximum, i.e., $\tau_{r\theta} = 0$. Equating

$\tau_{r\theta} = 0$ from equation A.4c,

$$\cos \frac{\theta}{2} [K_I \sin \theta + K_{II} (3 \cos \theta - 1)] = 0 \quad (\text{A.5})$$

The above equation has two solutions: one trivial solution $\cos \frac{\theta}{2} = 0$, and hence $\theta = \pm\pi$, and a non-trivial solution for which $K_I \sin \theta + K_{II} (3 \cos \theta - 1) = 0$. Solving this equation for θ and denoting the resulting value as θ_p , the propagaing angle, yields the following expression [258]

$$\theta_p = \begin{cases} 2 \tan^{-1} \left(\frac{K_I}{4K_{II}} - \frac{1}{4} \sqrt{\left(\frac{K_I}{K_{II}}\right)^2 + 8} \right) & \text{for } K_{II} \geq 0 \\ 2 \tan^{-1} \left(\frac{K_I}{4K_{II}} + \frac{1}{4} \sqrt{\left(\frac{K_I}{K_{II}}\right)^2 + 8} \right) & \text{for } K_{II} \leq 0 \end{cases} \quad (\text{A.6})$$

This formula is not well suited for numerical simulations because when K_{II} is very close to zero, it leads to very high values. Mi [R] has alleviated this problem by devising a different expression for the mixed-mode crack propagation that takes into account different signs of K_{II} . It is expressed as follows [86].

$$\theta_p = 2 \tan^{-1} \left(\frac{-2K_{II}}{K_I + \sqrt{(K_I)^2 + 8(K_{II})^2}} \right) \quad (\text{A.7})$$

This criterion provides existence of limiting crack propagation angle: for mode I problems $K_{II} = 0$, so $\theta_p = 0$ which means that the crack advances along the crack line; for mode II cases, $K_I = 0$, and $\theta_p = \pm 70.53^\circ$ depending on the sign of K_{II} .

This procedure is tested to solve all the linear problems presented in section 2.6, and produces results that are matching with J -integral approach. However, owing to the use of displacement correlation technique, this procedure requires higher mesh density at the crack tip.

B Implementation aspects of numerical integration techniques

B.1 Computation of normal of a polyhedron

While using the divergence theorem in moment fitting and direct divergence methods, it is necessary to make sure the direction of normals calculated for every facet of the volume-cells are outward pointing. The procedure to handle this issue in FEM simulations is explained here. Consider a very simple configuration in which a hexahedral element is cut by a triangulated interface as shown in Figure B.1, and it is of interest to construct a quadrature rule for the volume-cell Ω^+ . The facets of the volume-cell are either formed from the sides of the hexahedral element (known as regular facets), or formed by the cut interface (called as cut facets). The boundary of the cut facets is drawn with a thick line in Figure B.1. For the regular facets, there is always a corresponding parent side from the hexahedral element. In FEM simulations, the nodes of any side of an element are numbered to give outward normals. By calculating the normal of a regular facet, and comparing it with the corresponding parent side's normal, it can be decided whether the regular facet's normal is outward pointing or not. For the cut facets, the normal vector of the interface is used as the reference. The interface is generally represented as a triangulated surface. In any simulation, the direction of normal vector to these triangulated facets are known. For example, let us assume that the configuration shown in Figure B.1 arises in a simulation of fluid flow over a solid body, and the triangulated interface represents the surface of the solid object. In such cases, the normal vector calculated at the triangulated interface is always pointing towards the fluid side (towards Ω^+ side). Hence for the cut facet considered in this example, if the normal to the facet and the interface normal are pointing in the opposite direction then the cut facet's normal is outward pointing, and vice versa. When the interface is represented with a level set function instead of a triangulated surface, the direction of the normal to this level set side is always known and can be used as the reference. For any facet, if the normal vector is not outward pointing, the sign of the corresponding surface integral value is changed to account for this issue.

B.2 Computation of normals and equation of plane for a simple polygon

The surfaces of a regular finite element is a regular shaped convex shape, for which the usual 3-point method for calculation of normal vector works fine. However, when constructing the integration schemes using moment fitting and the direct divergence methods, it is essential to calculate the normal (or equation of plane) of facets of the volume-cells. Such facets can be concave in shape, in addition to having collinear points. In such instance, the 3-point method fails. In order to compute correct normal vector and equation of plane of facets of the volume-cells, which is crucial for constructing accurate integration schemes, we use Newell's method which is applicable even for concave shaped polygons having collinear points [259].

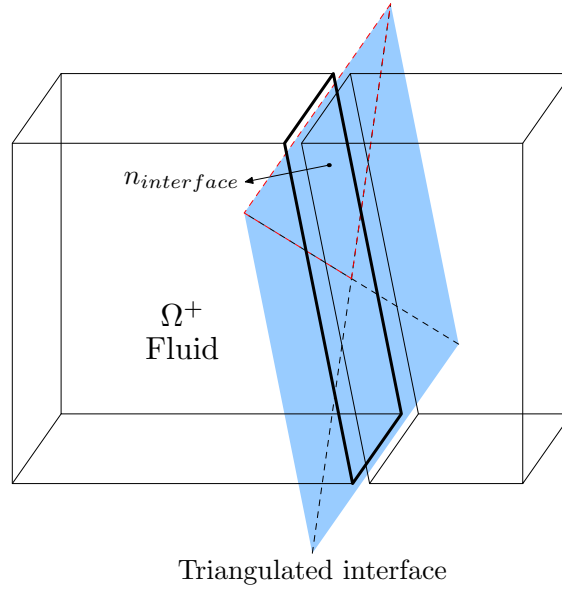


Figure B.1: Method to ensure the outward normal for every facet of a volume-cell

Let the facet has n number of vertices that are denoted as $v_1, v_2, v_3 \dots v_n$ in cyclic order. The vertex v_i is given by its Cartesian coordinates $v_i = (x_i, y_i, z_i)$. Let P denotes middle point of the polygon, given as,

$$P = \frac{1}{n} \sum_{i=1}^n v_i \quad (\text{B.1})$$

The equation of plane is denoted as $Ax + By + Cz = D$. The coefficients of this equations are given as following,

$$A = \sum_{i=1}^n (y_i - y_{i\oplus 1})(z_i + z_{i\oplus 1}) \quad (\text{B.2a})$$

$$B = \sum_{i=1}^n (z_i - z_{i\oplus 1})(x_i + x_{i\oplus 1}) \quad (\text{B.2b})$$

$$C = \sum_{i=1}^n (x_i - x_{i\oplus 1})(y_i + y_{i\oplus 1}) \quad (\text{B.2c})$$

$$D = P \cdot N \quad (\text{B.2d})$$

where $N = (A, B, C)$. If $fac = \sqrt{A^2 + B^2 + C^2}$, then the normal vector, in Cartesian coordinate system, is given by

$$\hat{\mathbf{n}} = \frac{A}{fac} \hat{i} + \frac{B}{fac} \hat{j} + \frac{C}{fac} \hat{k} \quad (\text{B.3})$$

Bibliography

- [1] C. Picioreanu, M. C. van Loosdrecht, and J. J. Heijnen. Two-dimensional model of biofilm detachment caused by internal stress from liquid flow. *Biotechnology and Bioengineering*, 72:205–218, 2001.
- [2] D. Keller and J. W. Costerton. *Effects of Oral Biofilm on the Immune System*. CDE World, 2002.
- [3] E. Alpkvist and I. Klapper. Description of mechanical response including detachment using a novel particle model of biofilm/flow interaction. *Water Science and Technology*, 55:265–273, 2007.
- [4] J. F. Hammond, E. J. Stewart, J. G. Younger, M. J. Solomon, and D. M. Bortz. Spatially heterogeneous biofilm simulations using an immersed boundary method with lagrangian nodes defined by bacterial locations. arXiv:1302.3663, 2013.
- [5] M. Denny and B. Gaylord. The mechanics of wave-swept algae. *The Journal of Experimental Biology*, 205:1355–1362, 2002.
- [6] K. J. Mach, D. V. Nelson, and M. W. Denny. Techniques for predicting the lifetimes of wave-swept macroalgae: a primer on fracture mechanics and crack growth. *The Journal of Experimental Biology*, 210:2213–2230, 2007.
- [7] K. J. Mach, B. B. Hale, M. W. Denny, and D. V. Nelson. Death by small forces: a fracture and fatigue analysis of wave-swept macroalgae. *The Journal of Experimental Biology*, 210:2231–2243, 2007.
- [8] K. J. Mach. Mechanical and biological consequences of repetitive loading: crack initiation and fatigue failure in the red macroalga *Mazzaella*. *The Journal of Experimental Biology*, 212:961–976, 2009.
- [9] P. E. O’Donoghue, S. T. Green, M. F. Kamminen, and P. K. Bowles. The development of a fluid/structure interaction model for flawed fluid containment boundaries with applications to gas transmission and distribution piping. *Computers & Structures*, 38:501–513, 1991.
- [10] C. J. Greenshields, G. P. Venizelos, and A. Ivankovic. A fluid-structure model for fast brittle fracture in plastic pipes. *Journal of Fluids and Structures*, 14:221–234, 2000.
- [11] F. Cirak, R. Deiterding, and S. P. Mauch. Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to shocks and detonations. *Computers & Structures*, 85:1049–1065, 2007.
- [12] T. Rabczuk, R. Gracie, J. H. Song, and T. Belytschko. Immersed particle method for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 81:48–71, 2010.
- [13] K. G. Wang. *A computational framework based on an embedded boundary method for nonlinear multi-phase fluid-structure interactions*. PhD thesis, Stanford University, 2011.
- [14] H. O. Nordhagen, S. Kragset, T. Berstad, A. Morin, C. Dørum, and S. T. Munkejord. A monolithic approach to fluid-structure interaction. *Computers & Structures*, 94-95:13–21, 2012.
- [15] P. D. Lea. *Fluid Structure Interaction with Applications in Structural Failure*. PhD thesis, Northwestern University, 2013.
- [16] J. D. Baum, H. Luo, E. Mestreau, R. Löhner, D. Pelessone, and C. Charman. A coupled CFD/CSD methodology for modeling weapon detonation and fragmentation, 1999. AIAA-99-0794.
- [17] J. F. Moxnes, A. K. Prytz, Ø. frøyland, S. Klokkehaug, S. Skriudalen, E. Friis, J. A. Teland, C. Dørum, and G. Ødegårdstuen. Experimental and numerical study of the fragmentation of expanding warhead casings by using different numerical codes and solution techniques. *Defence Technology*, 10:161–176, 2014.
- [18] D. Taherzadeh, C. Picioreanu, U. Küttler, A. Simone, W. A. Wall, and H. Horn. Computational study of the drag and oscillatory movement of biofilm streamers in fast flows. *Biotechnology and Bioengineering*, 105:600–610, 2010.
- [19] M. Coroneo, L. Yoshihara, and W. A. Wall. Biofilm growth: A multi-scale and coupled fluid-structure interaction and mass transport approach. *Biotechnology and Bioengineering*, 111:1385–1395, 2014.
- [20] L. Yoshihara, M. Coroneo, A. Comerford, G. Bauer, T. Klöppel, and W. A. Wall. A combined fluid-structure interaction and multi-field scalar transport model for simulating mass transport in biomechanics. *International Journal for Numerical Methods in Engineering*, 100:277–299, 2014.

- [21] J. Adachi, E. Siebrits, A. Peircec, and J. Desrochesd. Computer simulation of hydraulic fractures. *International Journal of Rock Mechanics & Mining Sciences*, 44:739–757, 2007.
- [22] S. Secchi and B. A. Schrefler. A method for 3-D hydraulic fracturing simulation. *International Journal of Fracture*, 178:245–258, 2012.
- [23] R. W. Zimmerman and I. W. Yeo. Fluid flow in rock fractures: From the Navier-Stokes equations to the cubic law. In B. Faybishenko, P. A. Witherspoon, and S. M. Benson, editors, *Dynamics of fluids in fractured rock*, Geophysical Monograph 122, pages 213–224. 2000.
- [24] D. J. Brush and N. R. Thomson. Fluid flow in synthetic rough-walled fractures: Navier-Stokes, Stokes, and local cubic law simulations. *Water Resources Research*, 39:doi:10.1029/2002WR001346, 2003.
- [25] M. B. Cardenas, D. T. Slottke, R. A. Ketcham, and J. M. Sharp Jr. Navier-Stokes flow and transport simulations using real fractures shows heavy tailing due to eddies. *Geophysical Research Letters*, 34:L14404, 2007.
- [26] A. Amini, F. Togashi, O. A. Soto, J. D. Baum, and R. Löhner. Blast propagation in a generic facility. In *Proceedings of MABS-21 Conference*, Jerusalem, Israel, 2010.
- [27] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 2002.
- [28] S. Shahmiri. *A hybrid fixed-grid-ALE approach for fluid-structure interaction*. PhD thesis, Technische Universität München, Germany, 2014.
- [29] M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams. An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.
- [30] K. B. Broberg. *Cracks and Fracture*. Academic Press, 1999.
- [31] T. G. Boue, R. Harpaz, J. Fineberg, and E. Bouchbinder. Failing softly: A fracture theory of highly-deformable materials. *arXiv:1502.04848v1*, 2015.
- [32] D. R. Curran, L. Seaman, and D. A. Shockey. Dynamic failure of solids. *Physics Reports*, 147:253–388, 1987.
- [33] T. L. Anderson. *Fracture Mechanics: Fundamentals and Applications*. Taylor & Francis Group, 2005.
- [34] J. R. Rice and R. Thomson. Ductile versus brittle behavior of crystals. *Philosophical Magazine*, 29:73–97, 1974.
- [35] C. F. Tipper. The fracture of metals. *Metallurgia*, 39:133–137, 1949.
- [36] J.M. Barsom and ASTM Committee E-24 on Fracture Testing. *Fracture Mechanics Retrospective: Early Classic Papers, 1913-1965*. ASTM retrospective publication series. ASTM, 1987.
- [37] C. E. Inglis. Stresses in a plate due to the presence of cracks and sharp corners. *Transactions of the Institute of Naval Architects*, 55:219–241, 1913.
- [38] A. A. Griffith. The phenomena of rupture and flow in solids. *Philosophical Transactions A*, 221:163–198, 1920.
- [39] G. R. Irwin. *Fracture Dynamics*, pages 147–166. Fracturing of Metals, American Society for Metals, Cleveland, 1948.
- [40] E. Orowan. Fracture and strength of solids. *Reports on Progress in Physics*, XII:185–232, 1948.
- [41] G. R. Irwin. Onset of fast crack propagation in high strength steel and aluminum alloys. *Sagamore Research Conference Proceedings*, 2:289–305, 1956.
- [42] G. R. Irwin. Analysis of stresses and strains near the end of a crack traversing a plate. *Journal of Applied Mechanics*, 24:361–364, 1957.
- [43] H. M. Westergaard. Bearing pressures and cracks. *Journal of Applied Mechanics*, 6:49–53, 1939.
- [44] M. L. Williams. On the stress distribution at the base of a stationary crack. *Journal of Applied Mechanics*, 24:109–114, 1957.
- [45] A. A. Wells. Unstable crack propagation in metals: Cleavage and fast fracture. In *Proceedings of the Crack Propagation Symposium*, volume 1, Cranfield, UK, 1961.
- [46] J. R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.
- [47] J. W. Hutchinson. Singular behavior at the end of a tensile crack tip in a hardening material. *Journal of the Mechanics and Physics of Solids*, 16:13–31, 1968.

-
- [48] J. R. Rice and G. F. Rosengren. Plane strain deformation near a crack tip in a power-law hardening material. *Journal of the Mechanics and Physics of Solids*, 16:1–12, 1968.
- [49] W. M. Lai, D. Rubin, and E. Krempl. *Introduction to Continuum Mechanics*. Butterworth-Heinemann, Third edition, 1999.
- [50] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.
- [51] M. E. Gurtin. *An introduction to continuum mechanics*. Academic Press, 1981.
- [52] G. A. Holzapfel. *Nonlinear solid mechanics: a continuum approach for engineering*. Wiley, 2000.
- [53] P. A. Wawrzynek and A. R. Ingraffea. An edge-based data structure for two-dimensional finite element analysis. *Engineering with Computers*, 3:13–20, 1987.
- [54] C. Miehe and E. Gürses. A robust algorithm for configurational-force-driven brittle crack propagation with R-adaptive mesh alignment. *International Journal for Numerical Methods in Engineering*, 72:127–155, 2007.
- [55] P. O. Bouchard, F. Bay, Y. Chastel, and I. Tovenà. Crack propagation modelling using an advanced remeshing technique. *Computer Methods in Applied Mechanics and Engineering*, 189:723–742, 2000.
- [56] A. C. O. Miranda, M. A. Meggiolaro, J. T. P. Castro, L. F. Martha, and T. N. Bittencourt. Fatigue life and crack path predictions in generic 2D structural components. *Engineering Fracture Mechanics*, 70:1259–1279, 2003.
- [57] S. Phongthanapanich and P. Dechaumphai. Adaptive Delaunay triangulation with object-oriented programming for crack propagation analysis. *Finite Elements in Analysis and Design*, 40:1753–1771, 2004.
- [58] T. N. Bittencourt, P. A. Wawrzynek, and A. R. Ingraffea. Quasi-automatic simulation of crack propagation for 2D LEFM problems. *Engineering Fracture Mechanics*, 55:321–334, 1996.
- [59] G. T. Camacho and M. Ortiz. Computational modeling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33:2899–2938, 1996.
- [60] M. Ortiz and A. Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International Journal for Numerical Methods in Engineering*, 44:1267–1282, 1999.
- [61] R. de Borst. Numerical aspects of cohesive-zone models. *Engineering Fracture Mechanics*, 70:1743–1757, 2003.
- [62] A. Turon, C. G. Dávila, P. P. Camanho, and J. Costa. An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models. *Engineering Fracture Mechanics*, 74:1665–1682, 2007.
- [63] K. Park and G. H. Paulino. Cohesive zone models: A critical review of traction-separation relationships across fracture surfaces. *Applied Mechanics Reviews*, 64:060802, 2011.
- [64] W. Celes, G. H. Paulino, and R. Espinha. A compact adjacency-based topological data structure for finite element mesh representation. *International Journal for Numerical Methods in Engineering*, 64:1529–1556, 2005.
- [65] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.
- [66] N. Sukumar, N. Moës, B. Moran, and T. Belytschko. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering*, 48:1549–1570, 2000.
- [67] J. Réthoré, A. Gravouil, and A. Combescure. An energy-conserving scheme for dynamic crack growth using the eXtended finite element method. *International Journal for Numerical Methods in Engineering*, 63:631–659, 2005.
- [68] Q. Z. Xiao and B. L. Karihaloo. Improving the accuracy of XFEM crack tip fields using higher order quadrature and statistically admissible stress recovery. *International Journal for Numerical Methods in Engineering*, 66:1378–1410, 2006.
- [69] V. Gupta, D. J. Kim, and Duarte C. A. Analysis and improvements of global–local enrichments for the generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 245–256:47–62, 2012.
- [70] V. Gupta, C. A. Duarte, I. Babuška, and U. Banerjee. A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, 266:23–39, 2013.

- [71] A. Nagarajan and S. Mukherjee. A mapping method for numerical evaluation of two-dimensional integrals with $1/r$ singularity. *Computational Mechanics*, 12:19–26, 1993.
- [72] E. Béchet, H. Minnebo, N. Moës, and B. Burgardt. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering*, 64:1033–1056, 2005.
- [73] P. Laborde, J. Pommier, Y. Renard, and M. Salaün. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering*, 64:354–381, 2005.
- [74] K. Park, J. P. Peraira, C. A. Duarte, and G. H. Paulino. Integration of singular enrichment functions in the generalized/extended finite element method for three-dimensional problems. *International Journal for Numerical Methods in Engineering*, 78:1220–1257, 2009.
- [75] S. E. Mousavi and N. Sukumar. Generalized Duffy transformation for integrating vertex singularities. *Computational Mechanics*, 45:127–140, 2010.
- [76] S. E. Mousavi and N. Sukumar. Generalized Gaussian quadrature rules for discontinuities and crack singularities in the extended finite element method. *Computer Methods in Applied Mechanics and Engineering*, 199:3237–3249, 2010.
- [77] H. Minnebo. Three-dimensional integration strategies of singular functions introduced by the XFEM in the LEFM. *International Journal for Numerical Methods in Engineering*, 92:1117–1138, 2012.
- [78] J. Donea, A. Huerta, J. P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. In *Encyclopedia of computational mechanics*, pages 1–38, Hawaii, USA, 2004.
- [79] H. M. Koh and R. B. Haber. A mixed Eulerian-Lagrangian model for the analysis of dynamic fracture. Technical Report 524, Civil Engineering Studies, Structural Research Series, Dept. of Civil Engineering, University of Illinois at Urbana-Champaign, Illinois, 1986.
- [80] H. M. Koh, H. S. Lee, and R. B. Haber. Dynamic crack propagation analysis using Eulerian-Lagrangian kinematic descriptions. *Computational Mechanics*, 3:141–155, 1988.
- [81] A. I. Abdelgalil. *Modeling of dynamic fracture problems using ALE finite element formulation*. PhD thesis, The University of British Columbia, 2002.
- [82] M. R. Amini and A. R. Shahani. Finite element simulation of dynamic crack propagation process using an arbitrary Lagrangian Eulerian formulation. *Fatigue & Fracture of Engineering Materials & Structures*, 36:533–547, 2013.
- [83] D. Bruno, F. Greco, and P. Lonetti. A fracture-ALE formulation to predict dynamic debonding in FRP strengthened concrete beams. *Composites: Part B*, 46:46–60, 2013.
- [84] J. P. Ponthot and T. Belytschko. Arbitrary Lagrangian-Eulerian formulation for element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 152:19–46, 1998.
- [85] M. M. Rashid. The arbitrary local mesh replacement method: An alternative to remeshing for crack propagation analysis. *Computer Methods in Applied Mechanics and Engineering*, 154:133–150, 1998.
- [86] A. Tabiei and J. Wu. Development of the DYNA3D simulation code with automated fracture procedure for brick elements. *International Journal for Numerical Methods in Engineering*, 57:1979–2006, 2003.
- [87] P. A. Browne, C. J. Budd, C. Piccolo, and M. Cullen. Fast three dimensional r-adaptive mesh redistribution. *Journal of Computational Physics*, 275:174–196, 2014.
- [88] C. J. Budd, R. D. Russell, and E. Walsh. The geometry of r-adaptive meshes generated using optimal transport methods. *Journal of Computational Physics*, 282:113–137, 2015.
- [89] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2001.
- [90] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- α method. *Journal of Applied Mechanics*, 60:371–375, 1993.
- [91] N. M. Newmark. A method of computation for structural dynamics. *ASCE Journal of Engineering Mechanics Division*, 85:67–94, 1959.
- [92] G. Legrain, N. Moës, and E. Verron. stress analysis around crack tips in finite strain problems using the eXtended finite element method. *International Journal for Numerical Methods in Engineering*, 63:290–314, 2005.
- [93] B. Moran and C. F. Shih. Crack tip and associated domain integrals from momentum and energy balance. *Engineering Fracture Mechanics*, 27:615–642, 1987.

-
- [94] K. N. Shivakumar and I. S. Raju. An equivalent domain integral method for three-dimensional mixed-mode fracture problems. *Engineering Fracture Mechanics*, 42:935–959, 1992.
- [95] J. Dolbow and A. Devan. Enrichment of enhanced assumed strain approximations for representing strong discontinuities: addressing volumetric incompressibility and the discontinuous patch test. *International Journal for Numerical Methods in Engineering*, 59:47–67, 2004.
- [96] L. Ma and A. M. Korsunsky. On the use of vector J -integral in crack growth criteria for brittle solids. *International Journal of Fracture*, 133:L39–L46, 2005.
- [97] F. Erdogan and G. C. Sih. On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engineering*, 85:519–527, 1963.
- [98] G. C. Sih. Strain-energy-density factor applied to mixed-mode fracture problems. *International Journal of Fracture*, 10:305–321, 1974.
- [99] M. A. Hussain, S. L. Pu, and J. H. Underwood. Strain energy release rate for a crack under combined mode I and mode II. *Fracture Analysis, ASTM STP*, 560:2–28, 1974.
- [100] P. O. Bouchard, F. Bay, and Y. Chastel. Numerical modelling of crack propagation: automatic remeshing and comparison of different criteria. *Computer Methods in Applied Mechanics and Engineering*, 192:3887–3908, 2003.
- [101] J. T. Betina. Unsteady euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA Journal*, 29:327–333, 1991.
- [102] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163:231–245, 1998.
- [103] F. Blom. Considerations on the spring analogy. *International Journal for Numerical Methods in Fluids*, 32:647–668, 2000.
- [104] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 80:305–316, 2002.
- [105] P. Crumpton and M. Giles. Implicit time-accurate solutions of unstructured dynamic grid. *International Journal for Numerical Methods in Fluids*, 25:1285–1300, 1997.
- [106] L. Tysell. Grid deformation of 3d hybrid grids. In *Proceedings of the 8th International Conference on Numerical Grid Generation in Computational Field Simulations*, pages 265–274, Hawaii, USA, 2002. International Society of Grid Generation.
- [107] B. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 56:1007–1021, 2003.
- [108] A. de Boer, M. S. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers and Structures*, 85:784–795, 2007.
- [109] A. Johnson and T. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.
- [110] W. A. Wall. *Fluid-Struktur Interaktion mit stabilisierten Finiten Elementen*. PhD thesis, Institut für Baustatik und Baudynamik, Universität Stuttgart, Germany., 1999.
- [111] T. Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89:1456–1467, 2011.
- [112] M. A. Heroux and J. M. Willenbring. Trilinos Users Guide. Technical Report SAND2003-2952, Sandia National Laboratories, 2003.
- [113] D. R. Lynch. Unified approach to simulation on deforming elements with application to phase change problems. *Journal of Computational Physics*, 47:387–411, 1982.
- [114] D. Azocar, M. Elgueta, and M. C. Rivara. Automatic LEFM crack propagation method based on local Lepp-Delaunay mesh refinement. *Advances in Engineering Software*, 41:111–119, 2010.
- [115] B. N. Rao and S. Rahman. An efficient meshless method for fracture analysis of cracks. *Computational Mechanics*, 26:398–408, 2000.
- [116] Y. Sumi, C. Yang, and Z. N. Wang. Morphological aspects of fatigue crack propagation part II-effects of stress biaxiality and welding residual stress. *International Journal of Fracture*, 82:221–235, 1996.

- [117] M. Fleming, Y. A. Chu, B. Moran, and T. Belytschko. Enriched element-free Galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering*, 40:1483–1504, 1997.
- [118] J. Dolbow and T. Belytschko. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, 23:219–230, 1999.
- [119] S. Natarajan. *Enriched finite element methods: Advances & Applications*. PhD thesis, Cardiff university, 2011.
- [120] P. Areias, D. Dias-da-Costa, J. Alfaiate, and E. Júlio. Arbitrary bi-dimensional finite strain cohesive crack propagation. *Computational Mechanics*, 45:61–75, 2009.
- [121] P. Bocca, A. Carpinteri, and S. Valente. Mixed mode fracture of concrete. *International Journal of Solids and Structures*, 27:1139–1153, 1991.
- [122] T. Rabczuk and G. Zi. A meshfree method based on the local partition of unity for cohesive cracks. *Computational Mechanics*, 39:743–760, 2006.
- [123] A. Gerstenberger and W. A. Wall. An extended finite element method / Lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197:1699–1714, 2008.
- [124] A. Gerstenberger and W. A. Wall. An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 82:537–563, 2010.
- [125] A. Gerstenberger. *An XFEM based fixed-grid approach to fluid-structure interaction*. PhD thesis, Technische Universität München, Germany, 2010.
- [126] B. Schott and W. A. Wall. A new face-oriented stabilized XFEM approach for 2D and 3D incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 276:233–265, 2014.
- [127] C. S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10:252–271, 1972.
- [128] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
- [129] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31:1019–1044, 1994.
- [130] L. Lee and R. J. LeVeque. An immersed interface method for incompressible Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 25:832–856, 2003.
- [131] R. Glowinski, T. W. Pan, A. J. Kearsley, and J. Periaux. Numerical simulation and optimal shape for viscous flow by a fictitious domain method. *International Journal for Numerical Methods in Fluids*, 20:695–711, 2005.
- [132] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons Ltd., 2003.
- [133] B. Schott, U. Rasthofer, V. Gravemeier, and W. A. Wall. A face-oriented stabilized Nitsche-type extended variational multiscale method for incompressible two-phase flow. *International Journal for Numerical Methods in Engineering*, DOI: 10.1002/nme.4789, 2014.
- [134] S. Shahmiri, B. Schott, R. Kruse, and W. A. Wall. A stabilized Nitsche-type extended embedding mesh approach for 3D low- and high-Reynolds number flows. under preparation.
- [135] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.
- [136] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the Babuska-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolation. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.
- [137] T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.
- [138] P. M. Gresho and R. L. Sani. *Incompressible flow and the finite element method*. John Wiley & Sons Ltd., 2000.
- [139] L. P. Franca, G. Hauke, and A. Masud. Revisiting stabilized finite element methods for the advective-diffusive equation. *Computer Methods in Applied Mechanics and Engineering*, 195:1560–1572, 2006.

-
- [140] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized Oseen problem. *Computer Methods in Applied Mechanics and Engineering*, 196:853–866, 2007.
- [141] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering*, 158:155–196, 1998.
- [142] C. H. Whiting and K. E. Jansen. A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis. *International Journal for Numerical Methods in Fluids*, 35:93–116, 2001.
- [143] F. Brezzi and K. J. Bathe. A discourse on the stability conditions for mixed finite element formulations. *Computer Methods in Applied Mechanics and Engineering*, 82:27–57, 1990.
- [144] P. Hansbo. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 84:175–192, 1990.
- [145] A. Linke. Collision in a cross-shaped domain – a steady 2d Navier-Stokes example demonstrating the importance of mass conservation in CFD. *Computer Methods in Applied Mechanics and Engineering*, 198:3278–3286, 2009.
- [146] P. Hansbo. Nitsche’s method for interface problems in computational mechanics. *GAMM-Mitteilungen*, 28:183–206, 2005.
- [147] E. Burman and P. Hansbo. Fictitious domain methods using cut elements: II. a stabilized Nitsche method. *Applied Numerical Mathematics*, 62:328–341, 2012.
- [148] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39:1749–1779, 2002.
- [149] E. Burman. A penalty-free nonsymmetric Nitsche-type method for the weak imposition of boundary conditions. *SIAM Journal of Numerical Analysis*, 50:1959–1981, 2012.
- [150] H. Ji and J. E. Dolbow. On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 61:2508–2535, 2004.
- [151] N. Moës, E. Béchet, and M. Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 67:1641–1669, 2006.
- [152] M. Hautefeuille, C. Annavarapu, and J. E. Dolbow. Robust imposition of Dirichlet boundary conditions on embedded surfaces. *International Journal for Numerical Methods in Engineering*, 90:40–64, 2012.
- [153] U. M. Mayer, A. Popp, A. Gerstenberger, and W. A. Wall. 3D fluid-structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 46:53–67, 2010.
- [154] J. Baiges, R. Codina, F. Henke, S. Shahmiri, and W. A. Wall. A symmetric method for weakly imposing Dirichlet boundary conditions in embedded finite element meshes. *International Journal for Numerical Methods in Engineering*, 90:636–658, 2012.
- [155] E. Burman and P. Hansbo. Fictitious domain methods using cut elements: III. a stabilized Nitsche method for Stokes’ problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:859–847, 2014.
- [156] E. Burman and P. Hansbo. Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1437–1453, 2004.
- [157] E. Burman and P. Zunino. A domain decomposition method based on weighted interior penalties for advection-diffusion-reaction problems. *SIAM Journal on Numerical Analysis*, 44:1612–1638, 2006.
- [158] E. Burman and Fernández. Finite element methods with symmetric stabilization for the transient convection-diffusion-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 198:2508–2519, 2009.
- [159] E. Burman and P. Hansbo. Edge stabilization for the generalized Stokes problem: A continuous interior penalty method. *Computer Methods in Applied Mechanics and Engineering*, 195:2393–2410, 2006.
- [160] F. Henke, M. Winklmaier, W. A. Wall, and V. Gravemeier. A semi-lagrangian time integration approach for extended finite element methods. *International Journal for Numerical Methods in Engineering*, 98:174–202, 2014.
- [161] U. M. Mayer, A. Gerstenberger, and W. A. Wall. Interface handling for three-dimensional higher-order XFEM-computations in fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 79:846–869, 2009.
- [162] J. P. Pereira, C. A. Duarte, D. Guoy, and X. Jiao. *hp*-Generalized FEM and crack surface representation for non-planar 3-D cracks. *International Journal for Numerical Methods in Engineering*, 77:601–633, 2009.

- [163] Y. J. Choi, M. A. Hulsen, and H. E. H. Meijer. Simulation of the flow of a viscoelastic fluid around a stationary cylinder using an extended finite element method. *Computers & Fluids*, 57:183–194, 2012.
- [164] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 47:1401–1417, 2000.
- [165] S. Zlotnik and P. Díez. Hierarchical X-FEM for n -phase flow ($n > 2$). *Computer Methods in Applied Mechanics and Engineering*, 198:2329–2338, 2009.
- [166] Y. J. Choi, M. A. Hulsen, and H. E. H. Meijer. An extended finite element method for the simulation of particulate viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, 165:607–624, 2010.
- [167] M. G. H. M. Baltussen, Y. J. Choi, M. A. Hulsen, and P. D. Anderson. Weakly-imposed Dirichlet boundary conditions for non-Newtonian fluid flow. *Journal of Non-Newtonian Fluid Mechanics*, 166:993–1003, 2011.
- [168] Y. J. Choi. *Modeling Particulate Complex Flows using XFEM*. PhD thesis, Technische Universiteit Eindhoven, 2011.
- [169] S. E. Mousavi and N. Sukumar. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Computational Mechanics*, 47:535–554, 2011.
- [170] B. Mueller, F. Kummer, M. Oberlack, and Y. Wang. Simple multidimensional integration of discontinuous functions with application to level set methods. *International Journal for Numerical Methods in Engineering*, 92:637–651, 2012.
- [171] S. Natarajan, S. Bordas, and D. R. Mahapatra. Numerical integration over arbitrary polygonal domains based on Schwarz-Christoffel conformal mapping. *International Journal for Numerical Methods in Engineering*, 80:103–134, 2009.
- [172] S. Natarajan, D. R. Mahapatra, and S. P. A. Bordas. Integrating strong and weak discontinuities without integration subcells and example applications in an XFEM/GFEM framework. *International Journal for Numerical Methods in Engineering*, 83:269–294, 2010.
- [173] G. Ventura. On the elimination of quadrature subcells for discontinuous functions in the eXtended Finite-Element Method. *International Journal for Numerical Methods in Engineering*, 66:761–795, 2006.
- [174] A. Martin, J.-B. Esnault, and P. Massin. About the use of standard integration schemes for X-FEM in solid mechanics plasticity. *Computer Methods in Applied Mechanics and Engineering*, DOI: 10.1016/j.cma.2014.09.028, 2014.
- [175] G. Ventura and E. Benvenuti. Equivalent polynomials for quadrature in heaviside function enriched elements. *International Journal for Numerical Methods in Engineering*, DOI: 10.1002/nme.4679, 2014.
- [176] D. J. Holdych, D. R. Noble, and R. B. Secor. Quadrature rules for triangular and tetrahedral elements with generalized functions. *International Journal for Numerical Methods in Engineering*, 73:1310–1327, 2008.
- [177] Z. Xiao, H. Gimbutas. A numerical algorithm for the construction of efficient quadratures in two and higher dimensions. *Computer Methods in Applied Mechanics and Engineering*, 59:663–676, 2010.
- [178] S. E. Mousavi, H. Xiao, and N. Sukumar. Generalized gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering*, 82:99–113, 2010.
- [179] M. Kästner, S. Müller, J. Goldmann, C. Spieler, J. Brummund, and V. Ulbricht. Higher-order extended FEM for weak discontinuities – level set representation, quadrature and application to magneto-mechanical problems. *International Journal for Numerical Methods in Engineering*, 93:1403–1424, 2013.
- [180] Y. Sudhakar and W. A. Wall. Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. *Computer Methods in Applied Mechanics and Engineering*, 258:39–54, 2013.
- [181] H. G. Trimmer and J. M. Stern. Computation of global geometric properties of solid objects. *Computer-Aided Design*, 12:301–304, 1980.
- [182] C. Cattani and A. Paoluzzi. Boundary integration over linear polyhedra. *Computer-Aided Design*, 22:130–135, 1990.
- [183] B. Mirtich. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics, GPU and Game tools*, 1:31–50, 1996.
- [184] J. P. M. Almeida and O. J. B. A. Pereira. A set of hybrid equilibrium finite element models for the analysis of three-dimensional solids. *International Journal for Numerical Methods in Engineering*, 39:2789–2802, 1996.

- [185] C. Gonzalez-Ochoa, S. Mccammon, and J. Peters. Computing moments of objects enclosed by piecewise polynomial surfaces. *ACM Transactions on Graphics*, 17:143–157, 1998.
- [186] P. Kaufmann, S. Martin, M. Botsch, and M. Gross. Flexible simulation of deformable models using discontinuous galerkin FEM. *Graphical Models*, 71:153–167, 2009.
- [187] H. T. Rathod and H. S. Govinda Rao. Integration of polynomials over linear polyhedra in Euclidean three-dimensional space. *Computer Methods in Applied Mechanics and Engineering*, 126:373–392, 1995.
- [188] G. Dasgupta. Integration within Polygonal Finite Elements. *Journal of Aerospace Engineering*, 16(1):9–18, 2003.
- [189] X. W. Gao. The radial integration method for evaluation of domain integrals with boundary-only discretization. *Engineering Analysis with Boundary Elements*, 26:905–916, 2002.
- [190] A. Sommariva and M. Vianello. Product gauss cubature over polygons based on green’s integration formula. *BIT Numerical Mathematics*, 47:441–453, 2007.
- [191] M. R. Hematiyan. A general method for evaluation of 2D and 3D domain integrals without domain discretization and its application in BEM. *Computational Mechanics*, 39:509–520, 2007.
- [192] Y. Sudhakar, J. P. M. Almeida, and W. A. Wall. An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: Application to embedded interface methods. *Journal of Computational Physics*, 273:393–415, 2014.
- [193] A. K. Tornberg. Multi-dimensional quadrature of singular and discontinuous functions. *BIT Numerical Mathematics*, 42:644–669, 2002.
- [194] B. Patzák and M. Jirásek. Process zone resolution by extended finite elements. *Engineering Fracture Mechanics*, 70:957–977, 2003.
- [195] E. Benvenuti, A. Tralli, and G. Ventura. A regularized XFEM model for the transition from continuous to discontinuous displacements. *International Journal for Numerical Methods in Engineering*, 74:911–944, 2008.
- [196] E. Benvenuti, G. Ventura, N. Ponara, and A. Tralli. Accuracy of three-dimensional analysis of regularized singularities. *International Journal for Numerical Methods in Engineering*, DOI: 10.1002/nme.4788, 2014.
- [197] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM transactions on Mathematical Software*, 22:469–483, 1996.
- [198] Si H. *Tetgen: a quality tetrahedral mesh generator and three dimensional Delaunay triangulator*. 2007, Available from: tetgen.berlios.de.
- [199] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:4081–4193, 2001.
- [200] J. N. Lyness and D. Jespersen. Moderate degree symmetric quadrature rules for the triangle. *Journal of the Institute of Mathematics and its Applications*, 15:19–32, 1975.
- [201] J. N. Lyness and G. Monegato. Quadrature rules for regions having regular hexagonal symmetry. *SIAM Journal on Numerical Analysis*, 14:283–295, 1977.
- [202] D. A. Dunavant. High degree efficient symmetrical gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering*, 21:1129–1148, 1985.
- [203] J. B. Lasserre. Integration on a convex polytope. *Proceedings of the American Mathematical Society*, 126(8):2433–2441, 1998.
- [204] J. B. Lasserre. Integration and homogeneous functions. *Proceedings of the American Mathematical Society*, 127(3):813–818, 1999.
- [205] A. Sommariva and M. Vianello. Gauss-green cubature and moment computation over arbitrary geometries. *Journal of Computational and Applied Mathematics*, 231:886–896, 2009.
- [206] K. Hormann and N Sukumar. Maximum entropy coordinates for arbitrary polytopes. In *Eurographics Symposium on Geometry Processing*, volume 27, pages 1513–1520, 2008.
- [207] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users’ Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
- [208] G. H. Meisters. Polygons have ears. *The American Mathematical Monthly*, 82:648–651, 1975.

- [209] D. Eberly. Triangulation by Ear Clipping. Available at <http://www.geometrictools.com/Documentation/TriangulationByEarClipping.pdf>, 2008.
- [210] A. Reusken. Analysis of an extended pressure finite element space for two-phase incompressible flows. *Computing and Visualization in Science*, 11:293–305, 2008.
- [211] H. Sauerland and T. P. Fries. The stable XFEM for two-phase flows. *Computers & Fluids*, 87:41–49, 2013.
- [212] B. Mueller, F. Kummer, and M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering*, 96:512–528, 2013.
- [213] C. R. Ethier and D. A. Steinman. Exact fully 3D Navier-Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, 19(1):369–375, 1994.
- [214] U. Banerjee and J. E. Osborn. Estimation of the effect of numerical integration in finite element eigen value approximation. *Numerische Mathematik*, 56:735–762, 1990.
- [215] U. Banerjee and M. Suri. The effect of numerical quadrature in the p -version of the finite element method. *Mathematics of Computation*, 59:1–20, 1992.
- [216] I. Babuška, U. Banerjee, and H. Li. The effect of numerical integration on the finite element approximation of linear functionals. *Numerische Mathematik*, 117:65–88, 2011.
- [217] D. J. Tritton. Experiments on the flow past a circular cylinder at low reynolds numbers. *Journal of Fluid Mechanics*, 6(4):547–567, 1959.
- [218] D. Calhoun. A cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *Journal of Computational Physics*, 176:231–275, 2002.
- [219] D. Russell and Z. J. Wang. A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. *Journal of Computational Physics*, 191:177–205, 2003.
- [220] H. Hamdani and M. Sun. Aerodynamic forces and flow structures of an airfoil in some unsteady motions at small reynolds number. *Acta Mechanica*, 145:173–187, 2000.
- [221] S. Shahmiri, A. Gerstenberger, and W. A. Wall. An XFEM based embedding mesh technique for incompressible viscous flows. *International Journal for Numerical Methods in Fluids*, 65:166–190, 2011.
- [222] I. Babuška, U. Banerjee, and Q. Osborn, J. Li. Quadrature for meshless methods. *International Journal for Numerical Methods in Engineering*, 76:1434–1470, 2008.
- [223] I. Babuška, U. Banerjee, and Q. Osborn, J. E. Zhang. Effect of numerical integration on meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 198:2886–2897, 2009.
- [224] K. Nissen, C. J. Cyron, V. Gravemeier, and W. Al. Wall. Information-flux method: a meshfree maximum-entropy Petrov-Galerkin method including stabilised finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 241-244:225–237, 2012.
- [225] N. Sukumar and A. Tabarraei. Confirming polygonal finite elements. *International Journal for Numerical Methods in Engineering*, 61:2045–2066, 2004.
- [226] M. M. Rashid and M. Selimotic. A three-dimensional finite element method with arbitrary polyhedral elements. *International Journal for Numerical Methods in Engineering*, 67:226–252, 2006.
- [227] M. Wicke, M. Botsch, and M. Gross. A finite element method on convex polyhedra. *Computer Graphics Forum*, 26:355–364, 2007.
- [228] P. Milbradt and T. Pick. Polytope finite elements. *International Journal for Numerical Methods in Engineering*, 73:1811–1835, 2008.
- [229] C. Min and F. Gibou. Geometric integration over irregular domains with application to level-set methods. *Journal of Computational Physics*, 226:1432–1443, 2007.
- [230] Y. T. Ng, C. Min, and F. Gibou. An efficient fluid-solid coupling algorithm for single-phase flows. *Journal of Computational Physics*, 228:8807–8829, 2009.
- [231] U. Rasthofer, F. Henke, W. A. Wall, and V. Gravemeier. An extended residual-based variational multi-scale method for two-phase flow including surface tension. *Computer Methods in Applied Mechanics and Engineering*, 200:1866–1876, 2011.
- [232] M. Theillard, L. F. Djodom, J. Vié, and F. Gibou. A second-order sharp numerical method for solving the linear elasticity equations on irregular domains and adaptive grids – application to shape optimization. *Journal of Computational Physics*, 233:430–448, 2013.

- [233] P. M. Boerrigter, G. Te Velde, and E. J. Baerends. Three-dimensional numerical integration for electronic structure calculations. *International Journal of Quantum chemistry*, XXXIII:87–113, 1988.
- [234] A. Alam, S. N. Khan, B. G. Wilson, and D. D. Johnson. Efficient isoparametric integration over arbitrary space-filling Voronoi polyhedra for electronic structure calculations. *Physical Review B*, 84:045105, 2011.
- [235] A. R. Dobrovolskis. Inertia of any polyhedron. *Icarus*, 124:698–704, 1996.
- [236] B. Singh and D. Guptasarma. New method for fast computation of gravity and magnetic anomalies from arbitrary polyhedra. *Geophysics*, 66:521–526, 2001.
- [237] D. Tsoulis. Analytical computation of the full gravity tensor of a homogeneous arbitrarily shaped polyhedra source using line integrals. *Geophysics*, 77:F1–F11, 2012.
- [238] D. Jürgens. Survey on software engineering for scientific applications. Technical report, Informatikbericht, Institute for Scientific Computing, Braunschweig, Germany, 2009.
- [239] M. Lesoinne and C. Farhat. A higher-order subiteration free staggered algorithm for nonlinear transient aeroelastic problems. *AIAA Journal*, 36:1754–1756, 1998.
- [240] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems—Part II: Energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3147–3170, 2001.
- [241] C. Farhat, P. Geuzaine, and G. Brown. Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers & Fluids*, 32:3–29, 2003.
- [242] C. Farhat, K. G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.
- [243] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190:3039–3067, 2001.
- [244] D. P. Mok and W. A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In W. A. Wall, K. U. Bletzinger, and K. Schweitzerhof, editors, *Proceedings of Trends in Computational Structural Mechanics*, pages 689–698, 2001.
- [245] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194:4506–4527, 2005.
- [246] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.
- [247] C. Förster. *Robust methods for fluid-structure interaction with stabilised finite elements*. PhD thesis, Institut für Baustatik und Baudynamik, Universität Stuttgart, Germany., 2007.
- [248] W. A. Wall, D. P. Mok, and E. Ramm. Partitioned analysis approach of the transient coupled response of viscous fluid and flexible structures. In W. Wunderlich, editor, *Solids, structures and coupled problems in engineering, ECCM '99, Munich*, 1999.
- [249] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.
- [250] B. Irons and R. C. Tuck. A version of the Aitken accelerator for computer implementation. *International Journal for Numerical Methods in Engineering*, 1:275–277, 1969.
- [251] A. Popp. *Mortar methods for computational contact mechanics and general interface problems*. PhD thesis, Technische Universität München, Germany, 2012.
- [252] M. Mayr, T. Klöppel, W. A. Wall, and M. W. Gee. A temporal consistent monolithic approach to fluid-structure interaction enabling single field predictors. *SIAM Journal on Scientific Computing*, 37:B30–B59, 2015.
- [253] A. Seitz, A. Popp, and W. A. Wall. A semi-smooth Newton method for orthotropic plasticity and frictional contact at finite strains. *Computer Methods in Applied Mechanics and Engineering*, 285:228–254, 2015.
- [254] C. F. Shih, H. G. de Lorenzi, and M. D. German. Crack extension modeling with singular quadratic isoparametric elements. *International Journal of Fracture*, 12:647–651, 1976.
- [255] T. N. Bittencourt. *Computer simulation of linear and nonlinear crack propagation in cementitious materials*. PhD thesis, Cornell University, 1993.

- [256] G. V. Guinea, J. Planas, and M. Elices. K_I evaluation by the displacement extrapolation technique. *Engineering Fracture Mechanics*, 66:243–255, 2000.
- [257] E. M. Wu. Application of fracture mechanics to anisotropic plates. *Journal of Applied Mechanics*, 34:967–974, 1967.
- [258] D. Broek. *Elementary Engineering Fracture Mechanics*. Martinus Nijhoff Publishers, 1986.
- [259] Dan Gordan, Computing the plane equation of a polygon. Class notes, Department of Computer Science, University of Haifa, Israel.

Reports of the Institute for Computational Mechanics at Technische Universität München

- 28 (2015) Sudhakar Yogaraj:**
An embedded interface finite element method for fluid-structure-fracture interaction.
- 27 (2015) Ursula Rasthofer:**
Computational multiscale methods for turbulent single and two-phase flows.
- 26 (2015) Hamman de Vaal:**
Computational modelling, clinical comprehension and improvement of aortic manipulation.
- 25 (2015) Tobias Wiesner:**
Flexible aggregation-based algebraic multigrid methods for contact and flow problems.
- 24 (2014) Jakob Huemer:**
Einfluss instationärer aerodynamischer Kräfte auf die Fahrdynamik von Personenkraftwagen.
- 23 (2014) Robert Metzke:**
Modeling and experimental investigation of the mechanobiological environment associated with alveolar pneumocytes.
- 22 (2014) Shadan Shahmiri:**
A hybrid ALE-fixed-grid approach for fluid-structure interaction.
- 21 (2014) Caroline Danowski:**
Computational modelling of thermo-structure interaction with application to rocket nozzles.
- 20 (2014) Kei Müller:**
Simulation of self-assembly and mechanics of transiently crosslinked, semiflexible biopolymer networks.
- 19 (2014) Mahmoud Ismail:**
Reduced dimensional modeling of the entire human lung.
- 18 (2013) Florian Henke:**
An extended finite element method for turbulent premixed combustion.
- 17 (2012) Markus Gitterle:**
A dual mortar formulation for finite deformation frictional contact problems including wear and thermal coupling.
- 16 (2012) Andreas Maier:**
Computational modeling of rupture risk in abdominal aortic aneurysms.

- 15 (2012) Georg Bauer:**
A coupled finite element approach for electrochemical systems.
- 14 (2012) Alexander Popp:**
Mortar methods for computational contact mechanics and general interface problems.
- 13 (2012) Thomas Klöppel:**
A finite element model for the human red blood cell.
- 12 (2012) Sophie Rausch:**
Computational and experimental modeling of lung parenchyma.
- 11 (2011) Christian Cyron:**
Micromechanical continuum approach for the analysis of biopolymer networks.
- 10 (2011) Lena Wiechert:**
Computational modeling of multi-field and multi-scale phenomena in respiratory mechanics.
- 9 (2010) Peter Gamnitzer:**
Residual-based variational multiscale methods for turbulent flows and fluid-structure interaction.
- 8 (2010) Axel Gerstenberger:**
An XFEM based fixed grid approach to fluid-structure interaction.
- 7 (2009) Ulrich Küttler:**
Effiziente Lösungsverfahren für Fluid-Struktur-Interaktions-Probleme.
- 6 (2009) Moritz Frenzel:**
Advanced structural finite element modeling of arterial walls for patient-specific geometries.
- 5 (2007)¹ Christiane Förster:**
Robust methods for fluid-structure interaction with stabilised finite elements.
- 4 (2004)¹ Tobias Erhart:**
Strategien zur Numerischen Modellierung transienter Impaktvorgänge bei nichtlinearem Materialverhalten.
- 3 (2004)¹ Michael Gee:**
Effiziente Lösungsstrategien in der nichtlinearen Schalenmechanik.
- 2 (2003)¹ Volker Gravemeier:**
The variational multiscale method for laminar and turbulent incompressible flow.
- 1 (2001)¹ Daniel Mok:**
Partitionierte Lösungsverfahren in der Strukturdynamik und der Fluid-Struktur-Interaktion.

¹This dissertation was supervised by Prof. Dr.-Ing. Wolfgang A. Wall at the Institute for Structural Mechanics at the University of Stuttgart and is published in the respective report series.

Lebenslauf

- Name: Sudhakar Yogaraj
- Geburtsdatum: 12. Februar 1985
- Geburtsort: Chatrapatti, India
- Nationalität: Indien
- 1989 – 1995: Grundschule, Indien
- 1995 – 2000: Highschool, Indien
- 2000 – 2002: Höheren Schule, Indien
- 2002 – 2006: Bachelor Studium
Universität: Anna University, Indien
Schwerpunkt: luftfahrt; Punktzahl: 86.3%
Bester Bachelorabschluss im Bereich Luftfahrttechnik an der Anna University
- 2007 – 2009: Master studium (Master of Science by Research)
Universität: Indian Institute of Technology Madras, Indien
Schwerpunkt: Angewandte Mechanik; Durchschnittspunktzahl: 10.0/10.0
Diss.: Aerodynamics of flapping insect wings in inclined stroke plane hovering
- 2010 – 2011: Scientist/Engineer an der Indian Space Research Organization
- 2011 – 2013: Marie Curie Early Stage Researcher,
Lehrstuhl für Numerische Mechanik,
Technische Universität München
- 2014 – 2015: Wissenschaftlicher Mitarbeiter,
Lehrstuhl für Numerische Mechanik,
Technische Universität München