Application of the Element Free Galerkin Method to Elastic Rods

by

Daniel Dreyer

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Civil and Environmental Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000

Author .....
Department of Civil and Environmental Engineering
September 14, 1999

Certified by .....
Eduardo Kausel
Professor
Thesis Supervisor

Accepted by .....
Daniele Veneziano
Chairman, Departmental Committee on Graduate Studies

Application of the Element Free Galerkin Method to Elastic Rods

by

Daniel Dreyer

Submitted to the Department of Civil and Environmental Engineering
on September 14, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science in Civil and Environmental Engineering

Abstract

This thesis presents an application of the Element Free Galerkin Method, or EFGM for short, to elastic rods. The method employs the so-called Moving Least Squares Interpolants as shape functions. This approximation method is reviewed together with the following topics: numerical integration; the strong form of the problem, that is, the differential equation; the weak form, which is derived using the Principle of Virtual Work; the method of Lagrange multipliers as well as the method of weighted residuals. A brief historical account of the origin of the EFGM is also given. The EFGM is examined and similarities with other field solution methods are outlined. A Matlab program, based on the forementioned theory and employing Lagrange multipliers to impose essential boundary conditions, is developed. Several numerical examples are computed, with varying order of integration, for smooth and C^0-continuous solutions. The obtained results are compared with closed-form solutions, indicating conclusions similar to previously reported results.

Thesis Supervisor: Eduardo Kausel
Title: Professor

Acknowledgments

My studies at the MIT could only have taken place with the loving trust, support and encouragement of my parents, Jutta and Dieter Dreyer, and my sister, Dagny. They helped me so much, I cannot find any appropriate expression for my gratitude.

My research advisor, Professor Eduardo Kausel, patiently showed me the way to independent research, he was available to share his knowledge through excellent explanations.

Much of my appreciation is due to my office colleagues in the basement of building 1: Karen P. L. Veroy, who was always available for questions and enlightened the basement with friendliness; Joonsang Park, who shared his extensive knowledge and was an outstanding example; Sung-June Kim, whose arguments on research were some of the best and richest I ever encountered in a basement; Mike Cusack, who always 'pushed the envelope', not only with respect to work ethic; sa'Toshi' Suzuki, who just by his presence amidst engineering students gave valuable diversity to the office; and last but not least David, Dominic, Vince, Pong and many more.

I also wish to thank Jane Dunphy. She taught, in her honest-enthusiastic illuminating manner, how to write a thesis...

My gratitude is due to the Dr.-Ing. Jürgen Ulderup Stiftung and the Studienstiftung des deutschen Volkes for their support.

I am also indebted to all my friends, whether or not they were around.

Thank you, Jaana.

Contents

List of Figures

List of Tables

Notation

α_i	Normalized weight in Gaussian quadrature
δ_{ij}	Kronecker delta
λ	Lagrange multiplier, Lamé constant
μ	Lamé constant
Π	Functional
ϕ	Interpolation / approximation function
$\Phi(x)$	Moving Least Squares Interpolation functions
$a(\cdot,\cdot)$	Bilinear operator
$\mathbf{a}$	Interpolation / approximation coefficients
A	Cross section of rod
$\mathbf{A}$	Matrix containing basis vectors $\mathbf{p}(x_i)$
$\tilde{\mathbf{A}}(x)$	Weighted moment matrix in Moving Least Squares Interpolation
$\mathbf{b}$	Interpolation / approximation data points
$\mathbf{b}_0$	Body force vector
$\tilde{\mathbf{B}}(x)$	Weighted data points matrix in Moving Least Squares Interpolation
$\mathfrak{B}$	Domain of problem statement
$\mathfrak{B}_0$	Interior of $\mathfrak{B}$
c_i	Distance of most remote node in minimal domain of influence
C^n	Space of n times differentiable continuous functions
C	Elasticity tensor
d_{factor}	Multplicator for domain of influence

d_{mi}	Radius of domain of influence of i-th node
$\partial\mathfrak{B}$	Boundary of $\mathfrak{B}$
E	Young's modulus
$\mathbf{E}$	Infinitesimal strain tensor
$\mathfrak{E}^h$	Strain energy of EFGM solution
f	Function to be interpolated / approximated
f^b	Body force, in force per unit length
F	Point force
$\mathbf{F}$	Load vector
$\mathbf{F}^*$	Load vector, amended by constraint u^*
G	Shear modulus
$\mathbf{G}$	Constraint terms for essential boundary conditions
$\mathfrak{H}^2$	Sobolev space
$\mathbf{I}$	Identity matrix
$\mathbf{K}$	Stiffness matrix
$\mathbf{K}^*$	Stiffness matrix, amended by constraint terms $\mathbf{G}$
L	Length of rod
LM	Vector relating global and local node numbering
$\mathfrak{L}_2$	Space of square integrable functions
L_{2m}	Differential operator of order $2m$
m	Order of basis vector $\mathbf{p}(x)$
N	Number of nodes
$\mathbf{p}(x)$	Basis vector in Moving Least Squares Interpolation
P_1	Space of linear polynomials
r_i	Normalized coordinate in Gaussian quadrature
$\mathbf{r}$	Residual

$\mathbf{S}$	First Piola-Kirchhoff stress tensor
S_u	Boundary of $\mathfrak{B}$ with prescribed essential boundary conditions
S_f	Boundary of $\mathfrak{B}$ with prescribed natural boundary conditions
$\mathbf{T}$	Cauchy stress tensor
u^*	Prescribed displacement
$u(x)$	Exact displacement solution
$\mathbf{U}$	Nodal displacements
$\mathbf{U}^*$	Nodal displacements vector, amended by Lagrange multiplier λ
v	Test function
V	Total solution space, including exact solution
V_h	Approximate solution space
$\mathbf{V}$	Vandermonde matrix
$w(d_i)$	Weight function
$\mathbf{W}(x)$	Diagonal weight function matrix

Chapter 1

Introduction

1.1 Solution Methods for Field Problems

Solution methods for field problems are widely used mathematical tools in engineering analysis. These methods are applied in such areas as the anaysis of solids and structures, heat transfer, fluids and almost any other area of engineering analysis.

One of the best known and most widely used methods is the Finite Element Method, or FEM in short. Its first occurence, or the roots of its development, date back to over four decades ago [3]. Since then, significant changes have taken place. Nowadays, there seems to be almost no field in which the FEM is not present.

Much of the success of the FEM is due to its intuitively physical approach and its great versatility. Indeed, the FEM may be applied in almost any area, its underlying mathematical structure enabling widespread use.

But despite — or even because of — this generality, there are certain areas in engineering analysis that are not well suited for analyses by the FEM. One example is the area of fracture analysis. Even three dimensional dynamic crack growth may be simulated, but at rather high expenses. Remeshing, at least locally, is necessary and results in loss in accuracy and computational cost.

Recently, especially in the last decade, interest has grown in so-called meshless methods.

These methods are not based on the notion of an element, as in the FEM, but instead attempt to approximate the given geometry through other procedures. One of them is the Element Free Galerkin Method, or EFGM.

1.2 Meshless Methods

The area of research of meshless methods is in a developing state and much has been done already. An extensive, but already somewhat outdated review is given by Belytschko et al. in [6], which also mention the EFGM. The formulation of the EFGM is similar to that of the FEM in that it is based on a Galerkin weak form of the problem; however, instead of using fixed, local interpolants, it uses Moving Least Squares Interpolants.

1.3 Development of the EFGM

The name Element Free Galerkin Method was first coined by Belytschko et al. in [7]. It is an improvement of the diffuse element method, which was introduced by Nayroles et al. [34]. They did not note that the employed interpolants are in fact Moving Least Squares Interpolants, for example as studied in [30]. Belytschko et al. improved the diffuse element method significantly: accurate evaluation of the derivatives, high order of integration and exact enforcement of essential boundary conditions enabled the method to pass the patch test [7].

Since then, improvements to, and analyses of the method itself have been made (see Chapter 3). EFGM has been extensively applied to various field problems. The method performs especially well in the area of fracture analysis [8, 10, 32, 35, 36, 41], due to its dynamic connectivity of nodes. Further, it seems to exhibit no volumetric locking (provided appropriate basis functions are employed) [7, 28].

1.4 About this Thesis

1.4.1 Intention

The objective of this research is to evaluate and examine the Element Free Galerkin Method. The method is applied to elastic rods subject to point forces and body forces. Its accuracy, characteristics and computational effort are examined.

1.4.2 Structure

The structure of this thesis is designed such that the EFGM is gradually developed. In Chapter 2 the fundamental utility in the EFGM is described, the Moving Least Squares Interpolants. Lagrange multipliers and numerical integration are mentioned, different mathematical forms of the problem statement and error-distribution methods are reviewed.

Chapter 3 gives an introduction to the EFGM, describing its characteristics in discretization, imposition of boundary conditions, numerical integration and post processing. Convergence of the method is mentioned with respect to the conforming and nonconforming EFGM. This Chapter closes with some relationships of the EFGM with other meshless methods and the FEM.

Chapter 4 describes the implementation of the EFGM using Matlab for the case of elastic rods subject to point and body forces, imposing essential boundary conditions via Lagrange multipliers.

Chapter 5 reviews numerical examples computed with the Matlab routines and examines the accuracy of the obtained results. Further, some considerations with respect to different weight functions are given.

Chapter 6 discusses the results.

Appendix A develops the mathematical-mechanical background of the Matlab implementation in Chapter 4. The amended weak form for a rod (essential for the EFGM employing Lagrange multipliers) is derived.

Appendix B tabulates numerical examples. Rods subject to point forces at the tip and in the middle and subject to constant and quadratic body forces are computed. In addition,

typical weight functions, employed in this thesis, are outlined.

Chapter 2

Mathematical Background

2.1 Introduction

This chapter reviews some of the formulations and methods necessary for the Element Free Galerkin Method. The Moving Least Squares Interpolants are at the core of the method. These functions are gradually developed from the basics in interpolation.

Imposition of constraints by Lagrange multipliers is described, and some characteristics of numerical integration are mentioned. Different mathematical formulations of problems in elasticity are reviewed, including the weak form. The latter leads to error-distribution methods, including the Galerkin method.

2.2 Moving Least Squares Method

2.2.1 Introduction

In the following, interpolation and approximation by a polynomial is reviewed, leading to the method of Moving Least Squares Interpolation, which are interpolation functions with adjustable support. The more general case of multiple variables can be found in [30]. In the following only the one dimensional case is considered.

Note that:

- approximation in general is considered as the problem of representing some given function $f(x)$ by some function $\phi(x)$, which comes close in value but may not be exactly equal to $f(x)$.

- interpolation, roughly-speaking, denotes the approximate representation of some given function $f(x)$ in terms of N discrete data points $(x_i, f(x_i))$ and interpolation functions $\phi(x; x_1, x_2, \ldots, x_N)$. In general, $\phi(x; x_1, x_2, \ldots, x_N)$ passes the value of $f(x_i)$ "through" when being evaluated at $x = x_i$, that is:

$$\phi(x_i; x_1, x_2, \ldots, x_N) = f(x_i) \qquad \forall\, i = 1, \ldots, N. \tag{2.1}$$

  Note, that eq. (2.1) is sometimes referred to as the interpolation condition, cf. eq. (3.1). When $\phi(x; x_1, x_2, \ldots, x_N)$ does not satisfy eq. (2.1) exactly, $\phi(x; x_1, x_2, \ldots, x_N)$ is therefore non-interpolating and resembles an approximation function.

For ease of notation, in the following the dependency of ϕ of $x_1, x_2, \ldots, x_N$ is omitted.

2.2.2 Interpolation Using a Polynomial

An interpolation function approximates some given function by estimating its shape between discrete data points (later referred to as nodes).

In Figure 2-1 some given function

$$f : D \to \mathbb{R}, \qquad f(x) = y \tag{2.2}$$

with discrete data points

$$(x_i, y_i), \qquad f(x_i) = y_i, \qquad i = 1, \ldots, N \tag{2.3}$$

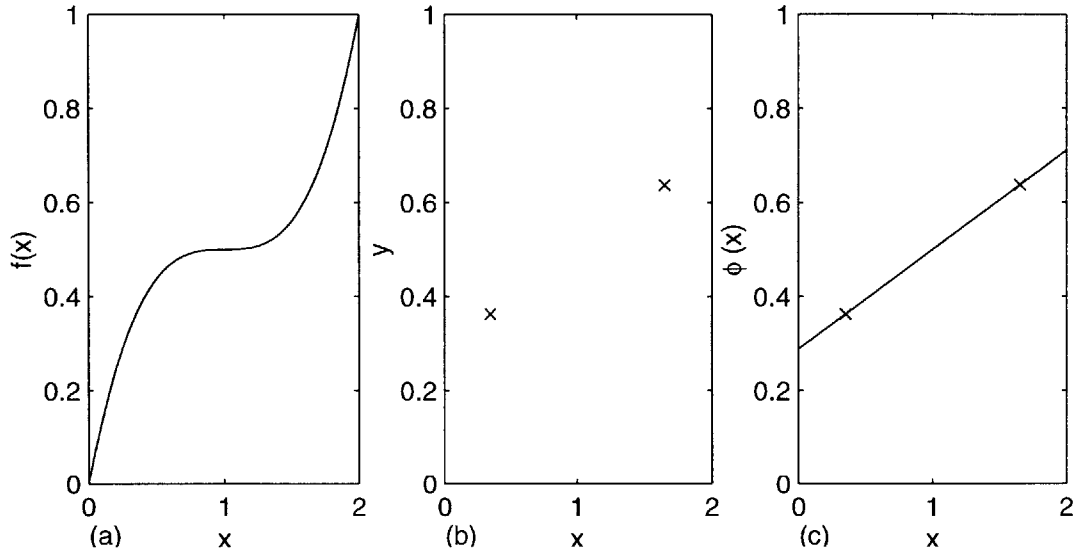

Figure 2-1: Interpolation of (a) $f(x)$ based on (b) two points by (c) $\phi(x)$

is approximated by a line $\phi(x)$ interpolating the given two data points. In general (for 2.2),

$$\begin{aligned} \phi(x) &= \mathbf{p}^T(x)\mathbf{a} \\ &= a_0 + a_1\, x + a_2\, x^2 + \cdots + a_{m-1}\, x^{m-1} \end{aligned} \tag{2.4}$$

with

$$\mathbf{a}^T = \begin{pmatrix} a_0 & a_1 & a_2 & \dots & a_{m-1} \end{pmatrix}, \quad \mathbf{p}^T = \begin{pmatrix} 1 & x & x^2 & \dots & x^{m-1} \end{pmatrix}, \tag{2.5}$$

where the $a_i \in \mathbb{R}$ determine the shape of the interpolant. Note, that in the following $\mathbf{p}$ is said to be of order $m = 2$ when it contains at most the linear term: $\mathbf{p}^T = \begin{pmatrix} 1 & x \end{pmatrix}$.

In order to obtain a unique linear interpolation, at least two independent points are required. For higher-order interpolating polynomials, obviously more data points are necessary to obtain unique interpolants. In general, $N = m$ points are required to obtain a unique polynomial of order m.

Determining the interpolation function reduces to solving a system of equations. One may compute the coefficients a_i in eq. (2.4) by employing the Vandermonde[1] matrix [2]:

$$\mathbf{Va} = \mathbf{b}, \tag{2.6}$$

where

$$\mathbf{V} = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^{m-1} \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{m-1} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix}.$$

2.2.3 Least Squares Approximation

Interpolation functions as presented in Section 2.2.2 are limited in applicability. In general, N data points require the interpolating polynomial to be of order $m = N$. It is not possible to interpolate an arbitrary set of points by a polynomial of fixed order[2]. Mathematically, the system of equations for $N > m$ in eq. (2.6) is not solvable in general.

When the number of data points is high, approximation instead of interpolation is used. The function $\phi(x)$ (to be determined) no longer interpolates the data points. For example, consider trend prediction and averaging in experimental data. Some error (called residual) in approximation is accepted.

One method to obtain such approximations is the least squares approach[3], where the error in interpolation at each data point is defined as:

$$r_i = \phi(x_i) - y_i, \qquad i = 1, \dots, N. \tag{2.7}$$

[1] Alexandre Théophile Vandermonde (1735-1796).
[2] This is similar to a desk with four legs (data points) rocking on an uneven floor (interpolation function).
[3] Problem definition and solution based on a method developed by Carl Friedrich Gauß (1777-1855).

The sum of these errors yields the residual of the approximation. In order to obtain the best approximation in the least squares sense (that means, minimizing the square of the residual) one has to determine the a_i from eq. (2.4) such that

$$\sum_{i=1}^{N} {r_i}^2 = \| \mathbf{r} \|_2^2 = \mathbf{r}^T \mathbf{r} \tag{2.8}$$

is minimized ($\| \cdot \|_2$ is the Euclidean norm). Resorting to matrix notation, with:

$$\mathbf{A} = \begin{pmatrix} 1 & x_1 & \dots & x_1^{m-1} \\ 1 & x_2 & \dots & x_2^{m-1} \\ 1 & x_3 & \dots & x_3^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_N & \dots & x_N^{m-1} \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix}, \tag{2.9}$$

eq. (2.7) can be expressed as:

$$\mathbf{r} = \mathbf{A}\,\mathbf{a} - \mathbf{b}. \tag{2.10}$$

Requiring $\mathbf{r}^T\mathbf{r}$ minimized leads to:

$$\mathbf{A}^T\mathbf{A}\,\mathbf{a} = \mathbf{A}^T\mathbf{b}. \tag{2.11}$$

Remark: Supposed, $\mathbf{A} \in \mathbb{R}^{(N,m)}$ is of rank(A) $= m$, $\mathbf{b} \in \mathbb{R}^N$ and $N \geq m$ holds, then it can be shown that eq. (2.11) has a unique solution $\mathbf{a} \in \mathbb{R}^m$.

2.2.4 Moving Least Squares Interpolation

Least squares approximation, as described in Section 2.2.3, minimizes the square of the residual over the whole set of data points (x_i, y_i), $i = 1, \dots, N$, equally weighted[4]. The

[4] The interpolation functions span the whole domain. Note the similarity to the Standard Ritz Method, which spans over the whole region — in opposite to the Finite Element Method.

approximating curve spans over the whole domain, or set of (x_i, y_i). It loses local accuracy. However, loss in accuracy, and wide-spanned domains may be circumvented by introducing localization terms in eq. (2.11).

Consider in eq. (2.11), instead of the moment matrix $\mathbf{A}^T\mathbf{A}$, some weighted matrix product:

$$\mathbf{A}^T\mathbf{W}(x)\mathbf{A}\,\mathbf{a}(x) = \mathbf{A}^T\mathbf{W}(x)\mathbf{b}, \tag{2.12}$$

with $\mathbf{W}(x)$ a diagonal matrix with N weight functions:

$$W_{ij}(x) = w(d_i)\,\delta_{ij}, \tag{2.13}$$

and $\mathbf{a}(x)$ the coefficients of the approximation function, as before. Note that both terms $\mathbf{W}(x)$ and $\mathbf{a}(x)$ are functions of x, while $\mathbf{A}$, similar to the Vandermonde matrix, is a quasi-constant matrix storing the basis vectors $\mathbf{p}(x_i)$ of nodes with nonzero weights, cf. eqs. (2.15, 2.24). The weight functions in $\mathbf{W}(x)$ limit the influence of its nodes to the corresponding domain of influence.

When computed by eq. (2.12), the coefficients $a_i(x)$ are not constant, as in eq. (2.4). Instead, they change continuously over the whole domain, introducing the localized weight of $\mathbf{W}(x)$ in the definition of $\phi_i(x)$. Note that due to the localization of the interpolation, multiple interpolation functions exist, one interpolation function for each data point x_i.

Eq. (2.12) is the definition of the Moving Least Squares Interpolant (or approximating function) in one dimension[5]. The derivation for the general case may be found in [30].

Weight Function and Domain of Influence

The localization term $\mathbf{W}(x)$, defined in eq. (2.13), distinguishes the Moving Least Squares Method from the standard least squares method. The choice of the weight function and

[5]Note, that *Moving* Least Squares Interpolation implies the notion of the support moving with the evaluation point x.

the definition of the domain of influence is an important part in Moving Least Squares Interpolation.

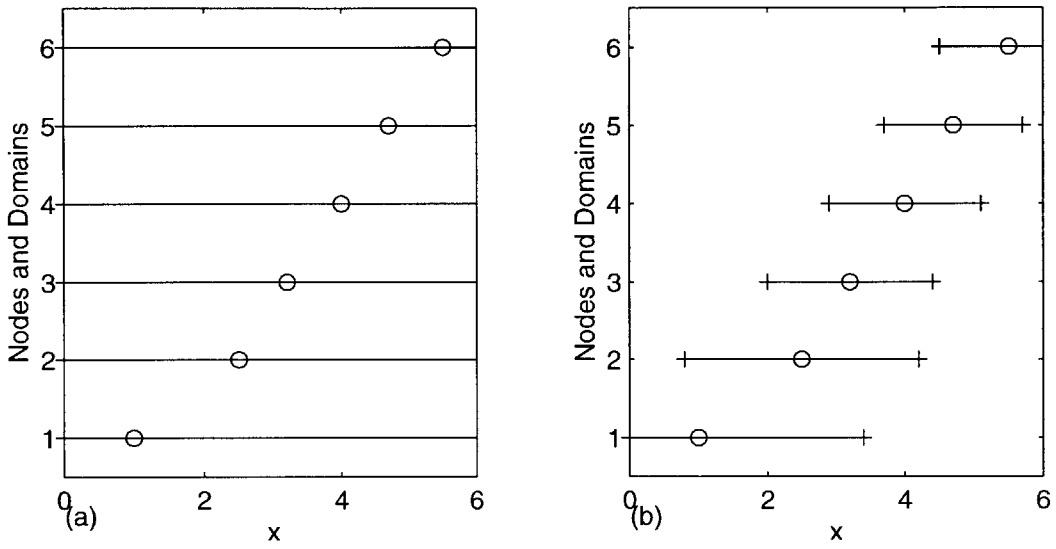

Figure 2-2: Domains of influence in (a) least squares approximation, (b) Moving Least Squares Approximation

The size of the domain of influence is determined by the definition of the weight function. Constant weight functions, which have unit value over the whole domain, yield the regular least squares approximation. Smooth weight functions with circular domains[6] and limited radius are employed in general. The circular domain (or in three dimensions: the ball-shaped domain) degenerates for one dimension to a line, where the weight function of the i-th node may be defined as:

$$w(d_i) = \begin{cases} w(d_i) & \text{if } d_i \leq d_{mi}, \\ 0 & \text{if } d_i > d_{mi}, \end{cases} \tag{2.14}$$

with d_{mi} the radius of the domain of influence [7] and d_i the distance of the i-th node

[6]Rectangular domains may also be used [17].

from the evaluation point x:

$$d_i = |x - x_i|.$$

It is important that the domain of influence is chosen such that $\mathbf{A}^T\mathbf{W}(x)\mathbf{A}$ is not a singular matrix. This requires that at least m nodes have nonzero weights. The process of determining the domain of influence (d_{mi} varies in general with x and is not constant for irregular meshes) consists of the following steps:

- determine some c_i defining the distance d_i of the most remote node in the domain for a minimum set of nodes,

- enlarge this minimum domain by defining $d_{mi} = d_{factor}\, c_i$, where d_{factor} may be chosen to be 1 and higher[7]. With $d_{factor} = 1$, piecewise linear interpolants are obtained, leading to the linear FEM as a special case of the EFGM [36]. Note that values $d_{factor} \geq 3$ may yield too smooth interpolants, leading to underestimation of results.

Moving Least Squares Interpolants may be interpolating and non-interpolating, depending on whether the weight function is singular at $x = x_i$. In general, the non-interpolating version is employed.

Some weight functions used for Moving Least Squares Interpolants are given in Appendix B. Note the difference between the singular weight functions in Figures B-1 and B-2 by Lancaster et al. [30] and the smooth weight functions in Figures B-3 and B-4, respectively, by [20, 41]. Employing the latter gives non-interpolating Moving Least Squares Interpolants, while using the singular weight functions yields interpolating Moving Least Squares functions. Note that the singularity requires some attention in numerical methods.

[7] Organ [36] recommended d_{factor} values (denoted as d_{max}) ranging from 2.01 to 2.51, while Belytschko et al. mentioned $2.0 \leq d_{factor} \leq 3.0$ for elastodynamics [8].

Formulation

It should be noted that the Moving Least Squares Interpolants are not polynomials, but rational functions. Furthermore, due to the complexity, $\phi_i(x)$ may not be computed analytically. One has to resort to numerical evaluation of the interpolants at the required points. For the case of the interpolant $\phi_i(x)$ itself, $a_i(x)$ has to be computed from eq. (2.12). That is, a set of linear equations has to be solved.

To obtain $\frac{d\phi_i(x)}{dx} = \phi_{i,x}(x)$, some attention may be required. Employing the following definitions:

$$\tilde{\mathbf{A}}(x) = \mathbf{A}^T\mathbf{W}(x)\mathbf{A}, \tag{2.15}$$

$$\tilde{\mathbf{B}}(x) = \mathbf{A}^T\mathbf{W}(x), \tag{2.16}$$

$$\mathbf{U} = \mathbf{b},$$

where $\mathbf{U}$ contains the nodal unknowns, one arrives at the definition of the Moving Least Squares interpolants for the EFGM. The basic approach is similar to eq. (2.4), where the analytical solution $u(x)$ is approximated as follows:

$$u(x) \approx \Phi(x)\,\mathbf{U} = \mathbf{p}^T(x)\,\mathbf{a}(x), \tag{2.17}$$

$$u_{,x}(x) \approx \Phi_{,x}(x)\,\mathbf{U} = \mathbf{p}^T_{,x}(x)\,\mathbf{a}(x) + \mathbf{p}^T(x)\,\mathbf{a}_{,x}\,(x), \tag{2.18}$$

where

$$\mathbf{a}(x) = \tilde{\mathbf{A}}^{-1}(x)\,\tilde{\mathbf{B}}(x)\,\mathbf{U}, \tag{2.19}$$

$$\mathbf{a}_{,x}(x) = \Big(\tilde{\mathbf{A}}^{-1}_{,x}(x)\,\tilde{\mathbf{B}}(x) + \tilde{\mathbf{A}}^{-1}(x)\,\tilde{\mathbf{B}}_{,x}(x)\Big)\mathbf{U}.$$

Avoiding the computationally intensive term $\tilde{\mathbf{A}}^{-1}_{,x}(x)$ leads to:

$$\mathbf{a}_{,x}(x) = \Big(-\tilde{\mathbf{A}}^{-1}(x)\,\tilde{\mathbf{A}}_{,x}(x)\,\tilde{\mathbf{A}}^{-1}(x)\,\tilde{\mathbf{B}}(x) + \tilde{\mathbf{A}}^{-1}(x)\,\tilde{\mathbf{B}}_{,x}(x)\Big)\mathbf{U}. \tag{2.20}$$

The shape functions (and its derivatives) for the EFGM may be computed from eqs. (2.17, 2.18) by substition of eqs. (2.19, 2.20), respectively.

Note that:

- the dimensions of the matrices are $\tilde{\mathbf{A}}(x) \in \mathbb{R}^{(m,m)}$ and $\tilde{\mathbf{B}}(x) \in \mathbb{R}^{(m,N)}$, where m denotes the order of the basis vector $\mathbf{p}(x)$ and N is the number of nodes, which weights are nonzero. For one dimensional problems choices for $\mathbf{p}(x)$ are:

$$\mathbf{p}^T(x) = \begin{pmatrix} 1 & x \end{pmatrix}, \quad m = 2, \tag{2.21}$$

$$\mathbf{p}^T(x) = \begin{pmatrix} 1 & x & x^2 \end{pmatrix}, \quad m = 3, \tag{2.22}$$

$$\mathbf{p}^T(x) = \begin{pmatrix} 1 & x & x^2 & x^3 \end{pmatrix}, \quad m = 4, \tag{2.23}$$

  etc.

- In the routines in Sections 4.2.7 and 4.2.8 $\tilde{\mathbf{A}}(x)$, $\tilde{\mathbf{B}}(x)$, $\tilde{\mathbf{A}}_{,x}(x)$ and $\tilde{\mathbf{B}}_{,x}(x)$ are computed as follows:

$$\tilde{\mathbf{A}}(x) = \sum_{i=1}^{N} w(d_i)\, \mathbf{p}(x_i)\, \mathbf{p}^T(x_i), \tag{2.24}$$

$$\tilde{\mathbf{B}}(x) = \begin{pmatrix} w(d_1)\, \mathbf{p}(x_1) & w(d_2)\, \mathbf{p}(x_2) & \dots & w(d_N)\, \mathbf{p}(x_N) \end{pmatrix}, \tag{2.25}$$

$$\tilde{\mathbf{A}}_{,x}(x) = \sum_{i=1}^{N} w_{,x}(d_i)\, \mathbf{p}(x_i)\, \mathbf{p}^T(x_i), \tag{2.26}$$

$$\tilde{\mathbf{B}}_{,x}(x) = \begin{pmatrix} w_{,x}(d_1)\, \mathbf{p}(x_1) & w_{,x}(d_2)\, \mathbf{p}(x_2) & \dots & w_{,x}(d_N)\, \mathbf{p}(x_N) \end{pmatrix}. \tag{2.27}$$

2.3 Lagrange Multiplier

The method of Lagrange[8] multipliers is a convenient way to solve problems subject to some given constraints. As an example, consider the problem of obtainig stationary values of

[8] Joseph Louis Lagrange (1736-1813).

some function $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$:

$$df = \frac{\partial f}{\partial x_1}\, dx_1 + \frac{\partial f}{\partial x_2}\, dx_2 + \cdots + \frac{\partial f}{\partial x_n}\, dx_n = 0 \tag{2.28}$$

subject to the two constraints

$$c_1(\mathbf{x}) = 0, \tag{2.29}$$

$$c_2(\mathbf{x}) = 0. \tag{2.30}$$

By amending $f(\mathbf{x})$ with the given constraints:

$$f^* = f + \sum_i \lambda_i\, c_i = f + \lambda_1\, c_1 + \lambda_2\, c_2$$

and taking the variation of f^* yields:

$$\frac{\partial f}{\partial x_i} + \lambda_1 \frac{\partial c_1}{\partial x_i} + \lambda_2 \frac{\partial c_2}{\partial x_i} = 0 \qquad \forall\, i = 1, \ldots, n. \tag{2.31}$$

Note that the conditions in (2.31) are the conditions of f^* be stationary [21].

To show that eq. (2.31) is in fact equivalent to eq. (2.28), subject to eqs. (2.29, 2.30), consider the following relations:

$$\sum_{i=1}^{n} \frac{\partial c_1}{\partial x_i}\, dx_i = 0, \tag{2.32}$$

$$\sum_{i=1}^{n} \frac{\partial c_2}{\partial x_i}\, dx_i = 0. \tag{2.33}$$

They follow directly from eqs. (2.29, 2.30). Multiplying eqs. (2.32, 2.33) by λ_1 and λ_2, respectively, and adding both equations to eq. (2.28) gives:

$$\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} + \lambda_1 \frac{\partial c_1}{\partial x_i} + \lambda_2 \frac{\partial c_2}{\partial x_i} \right) dx_i = 0. \tag{2.34}$$

From eq. (2.34), it can be seen that eq. (2.31) indeed holds. In general, the method of Lagrange multipliers may be used with more than two terms.

In Section A.3.2 Lagrange multipliers are employed to impose essential boundary conditions.

2.4 Numerical Integration

Numerical integration (in the following only for the case of one dimension) is the process of approximately integrating some function numerically, that is, based on the function itself instead of the analytical expression of $F = \int f \, dx$. For example, one basic method is Simpson's Rule.

The most widely-used method of numerical integration is Gaussian quadrature, but as mentioned in [37], in some cases even more accurate formulas with less sampling points have been developed. In Gaussian quadrature polynomials $\phi(x)$, as defined in eq. (2.4), may be integrated exactly, provided the required number of sampling points (sometimes called Gauss stations) is used.

The basic assumption in Gaussian numerical integration is that the integral can be approximated by a finite number of weighted sampling points:

$$\int_a^b f(x)\, dx = w_1\, f(x_1) + w_2\, f(x_2) + \ldots + w_\nu\, f(x_\nu) + R_\nu, \tag{2.35}$$

where the last term, R_ν, is the error in the approximation.

Consider some polynomial, $\phi(x)$, $x \in \mathbb{R}$, as defined in eq. (2.4). It can be shown, that polynomials of order $(2\mu - 1)$ are integrated exactly, provided μ sampling points are used [3]:

$$\int_a^b \phi(x)\, dx = \sum_{i=1}^{\mu} w_i \phi(x_i). \tag{2.36}$$

Eq. (2.35) is not used in practice. Instead, the weights and sampling points are transformed to normalized representation, based on the following expressions:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{m} w_i f(x_i), \tag{2.37}$$

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} r_i, \qquad r \in [-1;1] \tag{2.38}$$

$$w_i = \frac{b-a}{2} \alpha_i. \tag{2.39}$$

The sampling points r_i may be determined employing the Legendre-Polynomial [2]. An extensive collection of sampling points and weights may be found in [1].

So far only integration of polynomials is considered. However, rational, or nonpolynomial functions may be integrated using Gaussian quadrature. Obviously, no exact results are obtained, only approximations. Section 3.3 deals with the accuracy in integration of rational functions, like Moving Least Squares Approximations.

2.5 Problem Formulation

The following Sections are concerned with the various formulations and expressions which are involved in processes such as the development of the weak form as in the Appendix A or the convergence of the EFGM itself [29]. Most of the following departs from the well-known statement in eq. (2.40).

2.5.1 Differential Form

Consider some given boundary value problem. For the sake of simplicity, only one dimensional problems in the region $\mathfrak{B} = \{\, x \mid 0 \leq x \leq L \,\}$ are considered:

$$\frac{d}{dx}\left(EA\,\frac{du}{dx}\right) + f^b = 0, \tag{2.40}$$

where the displacements are prescribed on S_u and tractions / forces act on S_f.

Note that boundary conditions in general may be distinguished by the order of their differential operations [24], where $2m$ is the order of the differential operator in eq. (2.41):

- $0 \ldots m-1$: essential, geometric or Dirchlet boundaries,
- $m \ldots 2m-1$: natural, additional or Neumann boundaries.

2.5.2 Operational Form

The differential form of eq. (2.40) may be formulated as follows [37]: Find a unique correspondence, that is, find some relation between some given inhomgeneous term f and some u for the region $\mathfrak{B}$, each of them members of some spaces of functions, which satisfies the given differential equation and boundary conditions.

This process of matching spaces of functions is generally written by using differential operators:

$$\mathsf{L}_{2m} u = f \quad \text{in the region } \mathfrak{B} \subset \mathbb{R}, \tag{2.41}$$

where L_{2m} is a differential operator of order[9] $2m$ which in general establishes a unique one-to-one mapping of some f to some u. L_{2m} acts on a certain class of functions — those which in some sense satisfy the boundary conditions, prescribed on $\partial\mathfrak{B} = \mathsf{S}_u \cup \mathsf{S}_f$, and which can be differentiated $2m$-th times (denoted as C^{2m} functions).

Note that eq. (2.41) is the Euler[10] equation.

Consider the class of admissible functions of u and f. Only f with finite energy are admitted:

$$\int_{\mathfrak{B}} (f)^2 \, dx < \infty. \tag{2.42}$$

In this case, the integral expression in eq. (2.42) is the square sum over all body forces applied to the rod, therefore it is reasonable to require this integral to be finite. The space

[9] If not otherwise mentioned, the order of the differential operator is $2m = 2$.
[10] Leonhard Euler (1707-1783).

of functions which satisfy eq. (2.42) is often denoted $\mathfrak{L}_2$, which here is defined as:

$$\mathfrak{L}_2(M) = \{\omega \mid \omega \text{ defined in } M \text{ and } \int_M (\omega)^2 dV < \infty\}. \tag{2.43}$$

It follows that the space of admissible functions for eq. (2.41) is the Sobolev space $\mathfrak{H}^2$. That is, where: (1) the interpolation function, (2) its first derivatives and (3) its second derivatives are required to be in $\mathfrak{L}_2$ [22, p. 267]. Obviously, only functions in $\mathfrak{H}^2$ are allowed which satisfy the given boundary conditions.

Once these requirements on u and f are satisfied, and L_{2m} is in fact a one-to-one transformation[11], it can be shown that the given boundary-value problem has a unique solution u. Physically, the existence of a unique u for the operator L_{2m} expresses the following: For a problem within the linearized theory of elasticity there exists one unique deformed shape u for each load f.

2.5.3 Variational Form

An alternative form to the statements in Sections 2.5.1 and 2.5.2 is the variational form. Consider eq. (2.40), weighted by test functions v and integrated over the domain $\mathfrak{B}$[12]:

$$a(\mathsf{L}_{2m}, v) = a(f, v) \quad \text{for every } v. \tag{2.44}$$

This equation is the result of varying the quadratic $I(v) = a(\mathsf{L}_{2m}v, v) - 2\,a(f, v)$, that is minimizing $I(v)$.

Note that due to integration by parts, eq. (2.44) does not contain second derivatives of u, see eq. (A.15). In fact, the space of admissible solutions in the minimization is $\mathfrak{H}^1$, where the functions are only required to satisfy the essential boundary conditions in advance.

The fundamental enlargement of admissible functions to piecewise linear C^0-continuous

[11] This requires L_{2m} to be self-adjoint and positive definite.

[12] Here $a(\cdot, \cdot)$ is a bilinear form corresponding to the considered model problem. "Bilinear" denotes that this form is linear in both elements on which it operates.

functions[13] enables, for example, the use of two-node elements in elasticity problems in the Finite Element Method.

In EFGM this enlargement by piecewise linear C^0-continuous functions of the space of admissible functions is not necessary. However, the admissible functions have to satisfy the essential boundary conditions. This is not satisfied in general by the interpolation functions employed in EFGM, the Moving Least Squares Interpolants.

2.6 Method of Weighted Residuals

2.6.1 Introduction

The method of weighted residuals seeks to determine a best approximate solution to a differential equation, subject to boundary conditions, through the use of trial (and test) functions [11]. It focuses on the error in satisfying the given problem, similar to the interpolation in Section 2.2.3.

Error-distribution methods yield results which in some sense are the best approximations to the exact solution. Different methods are available [11], where $\mathfrak{B} = \partial\mathfrak{B} + \mathfrak{B}_0$, so that $\partial\mathfrak{B}$ denotes the boundary and $\mathfrak{B}_0$ is the interior of $\mathfrak{B}$:

- *boundary method*: The differential equation is satisfied exactly in the interior $\mathfrak{B}_0$, and the unknowns are selected to fit the boundary conditions in some best sense.

- *interior method*: The trial functions are chosen such that they satisfy the boundary conditions exactly, and the residual is minimized over the whole interior $\mathfrak{B}_0$.

- *mixed method*: Neither the differential equation in $\mathfrak{B}_0$, nor the boundary conditions on $\partial\mathfrak{B}$ are satisfied. Here, the unknowns to be determined are selected to satisfy both the differential equation and the boundary condition in some best sense.

[13] Slope discontinuity allowed, but discontinuities in u itself prohibited.

2.6.2 Formulation

Given some problem statement, cf. eq. (2.41):

$$\mathsf{L}_{2m} u - f = 0 \quad \text{in the region } \mathfrak{B} \tag{2.45}$$

with boundary conditions prescribed on $\partial\mathfrak{B} = \mathsf{S}_u \cup \mathsf{S}_f$. Consider the following approach by the interior method: Some approximate solution of the differential equation is assumed:

$$u \approx \bar{u} = \sum_{i=1}^{N} \phi_i(x)\, u_i,$$

where the u_i are the unknowns to be determined, and the $\phi_i(x)$ are trial functions which are chosen to satisfy all boundary conditions. This approximation, employed in eq. (2.45), is not likely to satisfy the differential equation exactly. Some error, called residual, remains:

$$r(x) = \mathsf{L}_{2m}\bar{u} - f.$$

This residual is required to vanish or being minimized over the interior $\mathfrak{B}_0$, appropriately weighted by test functions[14]:

$$\int_{\mathfrak{B}} v_i(x)\, r(x)\, dx = 0, \qquad \forall\, i = 1, \ldots, N, \tag{2.46}$$

where the $v_i(x)$ are N linearly independent functions. The exact solution is obtained if eq. (2.46) holds for any complete set of functions $v_i(x)$, with $N \to \infty$. In practice, only a limited number of functions is chosen.

The various choices of test (or weight) functions gave birth to several methods within the class of error-distribution methods. In the following, some widely-used methods are described.

[14] In [13], $r(x)$ is said to be orthogonalized to $v_i(x)$ over the interior $\mathfrak{B}_0$.

2.6.3 The Galerkin Method and other Error-Distribution Methods

Next to the Galerkin method[15], which is *"the obvious discretization of the weak form"* [37, p. 117][16], other error-distribution methods have been established and used.

Collocation Method

In the collocation method[17] the residual is required to vanish at N points. That is, the $v_i(x)$ are chosen to be the Dirac delta function $\delta(x-x_i)$. However, the approximate solution does not coincide at the points x_i with the exact solution [24]. Furthermore, the obtained results may fluctuate widely between the x_i [11].

Least Squares Method

This method was first mentioned by Picone in 1928 [18]. Here, the integral of the square of the residual is not forced to be zero, but instead required to be a minimum with respect to the unknown u_i. However, the involved integrations may often be complicated [21].

Subdomain Method

In the subdomain[18] or partition method the region $\mathfrak{B}$ is divided in subdomains. The integral of the residual is required to be zero over each subdomain. Hence, the weight functions may be considered unit step functions, which are unity in the i-th domain and zero elsewhere.

Galerkin Method

In the Galerkin method the test functions $v_i(x)$ are chosen to be the trial functions $\phi_i(x)$ themselves. That is, the residual is forced to be orthogonal to the space of trial functions. Strang and Fix [37] expressed Galerkin's rule as follows (note that the approximate solution space V_h, determined by the interpolation functions, is denoted S^h, where $\bar{u} \in S^h$):

[15] Boris Grigorewitsch Galerkin (1871-1945) mentioned this method in 1915 [18].
[16] Note that eq. (2.45) is the differential form, or strong form, and not the weak form.
[17] First mentioned in 1937 by Frazer, Jones and Skan [18].
[18] Finlayson et al. [18] name Biezeno and Koch as the first to mention this method in 1923.

The residual $L\bar{u}-f$ will not be identically zero unless the true solution u happens to lie in the trial space S^h, but its components in S^h should vanish.

Chapter 3

Element Free Galerkin Method

3.1 Introduction

This Chapter describes the characteristics of the Element Free Galerkin Method and gives an overview of the EFGM and some already examined features.

The Element Free Galerkin Method is a numerical method for approximate analysis. Some notes on its development are given in Section 1.3. Descriptions of the EFGM may be found in most of the publications named in the Bibliography. It should be noted that the area of research of meshless methods, and therefore of the EFGM, is steadily developing. The following descriptions do not claim to be "state of the art" or "cutting edge". But they may be satisfactory as an introductory approach to the EFGM, mentioning limitations without lacking reasonable accuracy and some actuality. The reader may miss some of the mathematical notation. This is outlined in parts in Chapter 2 and not repeated here. Additionally, Appendix A describes the complete derivation of the formulation for an elastic rod.

Possibly the most significant drawback of the EFGM is that the computational effort is high, the method is expensive. But the whole area of meshless numerical methods is still developing, and when considering the computational effort spent on Finite Element Analyses with millions or more degrees of freedom, this disadvantage may fade in future.

3.2 Discretization and Boundary Conditions

As in all meshless methods, the difficulties arising from essential boundary conditions are directly related to the interpolation functions [25, 29]. Moving Least Squares Interpolants are generally employed in the EFGM, which directly affect the process of meshing.

3.2.1 Meshing

The EFGM belongs to the class of so-called meshless methods. That is, there exists no consideration of elements as in the FEM. The region of the field problem, $\mathfrak{B}$, is not required to be subdivided into separate cells, or elements. The procedure of "meshing" reduces — in the pure sense of a meshless method — to distributing nodes in an arbitrary shape over the domain[1].

The difficulties in meshing, and especially, in remeshing[2], are avoided. For example, even adaptive schemes (variable node density based on interpolation error estimation) have been developed to automate and accelerate the solution of some time dependent problems [20].

3.2.2 Shape Functions

In the EFGM in general Moving Least Squares Interpolants are employed as shape functions (also called interpolation functions).

They span the space of discrete solutions V_h (note that V denotes the total solution space, including the exact solution). Alternatives or modifications may be used, like pseudo-derivatives of Shepard functions [26] or modified Moving Least Squares Interpolants with orthogonalized basis functions [31]. Even accelerated computation of derivatives [5] and enrichment of the basis to improve the representation of crack tip fields have been developed

[1] Obviously, the density of nodes should be based on the expected field solution.

[2] For example, remeshing is of importance in field problems with time-dependent domains, that is, changing geometry. This advantage explains much of the success of the EFGM in the area of fracture analysis .

[40]. In this thesis Moving Least Squares Interpolants, as described in Section 2.2, are employed.

Moving Least Squares Interpolants are superior to finite element shape functions with respect to adaptivity. Nodal connectivity is not defined in advance, but may change together with the geometry. Nodes may also be added and subtracted from the region quite easily. This advantage comes along with the disadvantage of high computational cost in evaluating the shape functions at the quadrature points[3]. Another drawback, next to this computational burden, is the loss of the physical meaning of the nodal unknowns[4]. The displacement at node i does not represent the displacement at this location . As mentioned in Section 2.2.4, Moving Least Squares Interpolants, as used in the EFGM, do not satisfy the interpolation condition:

$$\phi_i(x_j) \neq \delta_{ij}. \tag{3.1}$$

Instead:

$$u(x_i) = \sum_{j=1}^{N} \phi_j(x_i)\, U_j \neq U_i. \tag{3.2}$$

This disadvantage affects the imposition of boundary conditions.

3.2.3 Imposition of Constraints

Approximation functions used in the weak form are required to satisfy the essential boundary conditions. Similar to the principle of virtual work, the virtual displacement has to vanish at supports, while natural boundary conditions do not have to be satisfied in advance.

[3]Noted in Section 2.2.4, a linear set of equations (2.17, 2.18) has to be solved for each evaluation point.

[4]As long as non-singular weight functions are employed (which is recommended, since the singularity imposes certain difficulties in the numerical implementation of the interpolants), the interpolation condition is not satisfied.

Essential Boundary Conditions

Essential boundary conditions, which are prescribed displacements, have to be satisfied in advance by the approximation functions. As in other meshless methods, this requirement is not satisfied exactly by the Moving Least Squares Interpolants, as long as no additional constraints are present. In the EFGM, prescribed displacements may not be imposed directly on the system of equations, for example by eliminating rows and columns, or by the penalty method. Instead, Lagrange multipliers [7] , coupling with finite elements [25, 38, 39] or modified variational formulations [31] have to be used. Note that the accurate enforcement of essential boundary conditions is one of the differences between the Diffuse Element Method and the EFGM.

In this thesis, for the one dimensional case, Lagrange multipliers are used to satisfy essential boundary conditions. For problems with increased complexity, coupling of EFGM with FEM domains using ramp functions should be used to avoid the difficulties arising from essential boundary conditions acting on meshless domains.

Natural Boundary Conditions

Natural boundary conditions, like tractions in the case of elasticity problems, do not have to be satisfied in advance by the shape functions (in opposition to essential boundary conditions). The process of obtaining the best approximate solution employing the weak form of the given problem includes already the imposed natural boundary conditions.

Note, that in the case of point forces , some considerations on the accurate formulation are necessary. A point load may be seen as some distributed load with the Dirac delta function as weight function [23]. Suppose within some EFGM discretization, the source point of a point load may coincide with the position of a node. Still the point force has to be distributed using Moving Least Squares Interpolants.

3.3 Integration

The underlying error-distribution method in EFGM is the Galerkin method. Integration is an essential part in Galerkin methods, but it increases the computational burden.

Integration in meshless methods, as in the EFGM, may be some source of confusion. Despite its meshless character, there exists in EFGM at least some subdivision of the domain required for integration. These integration cells, also called background mesh , however, are in general not related to the nodal distribution [7]. But there exist implementations which connect the quadrature cells to the nodes [41].

Integration in the EFGM is performed numerically. In general, Gaussian integration is used. But determining the required order of integration is not as straightforward as in the FEM. In FEM, there exist exact definitions of the terms "full" and "reduced integration", and one can easily determine the required amount of quadrature points for full integration.

As mentioned in Section 2.2.4, the interpolation functions in the EFGM are rational, and therefore cannot be integrated exactly. Still, numerical integration is sufficient, as long as the error in integration is small enough not to deteriorate convergence. The effects of integration on accuracy and convergence have been mentioned [16, 17]. For a too low order of integration, meaningless (oscillating) results were obtained (see also Appendix B).

Attempting to reduce the computational cost of integration, Beissel and Belytschko implemented a nodal integration scheme of the EFGM, but failed for the case of the unstabilized form [4].

3.4 Post Processing

As noted in [7], the EFGM relaxes the requirements on smoothing of results during post processing. The highly smooth Moving Least Squares Interpolants do not exhibit any jump discontinuities (as it is the case in the FEM with linear shape functions) in derivatives, not considering fracture analysis.

However, since the Moving Least Squares Interpolants do not satisfy the interpolation

condition (see eq. (3.1) in Section 3.2.2), the solution $\mathbf{U}$ lacks some physical meaning. In order to obtain the smoothed interpolated solution $\Phi(x)\mathbf{U}$, the Moving Least Squares Interpolants have to be evaluated. Again, as in the integration process, this increases computational cost.

3.5 Convergence

Convergence of the EFGM may be shown as in [37, p. 18] for the FEM:

> *Consistency and stability imply convergence.*

Krysl and Belytschko adapted some of Strang and Fix's approach in [29].

From a less mathematical point of view, Hughes' notes of the requirements on shape functions for convergence in FEM may be considered [22]. The shape functions are required to be:

- smooth (at least C^1) on each element interior,
- continuous across each element boundary and
- complete.

Not all three requirements are directly applicable. One of them is satisfied in advance, the smoothness criterion[5]. The completeness criterion, however, requires more attention:

> *Completeness requires that the rigid body displacements and constant strain states be possible.* [3, p. 377]

> *Completeness requires that the element interpolation function is capable of exactly representing an arbitrary linear polynomial when the nodal degrees of freedom are assigned values in accordance with it.* [22, p. 111]

[5]Note that in fracture analysis the discontinuous shape functions, constructed by the visibility criterion [8, 10, 29] do not satisfy this requirement in advance. However, this study is not concerned with fracture analysis, and therefore, the interpolation functions are smooth over the entire domain.

That is: an element / mesh / discretization must exactly represent all rigid motions (rigid translations and rotations) and constant strain states. Both descriptions of completeness[6] are formulated for the FEM, but are also applicable to the EFGM.

Patch Test

A well-known (but in its numerical implementation not sufficient) test for completeness is the patch test. Krysl and Belytschko showed the following [29]:

$$a(\tilde{u}, \phi_i) = a_h(\tilde{u}, \phi_i),$$

where a is the analytical bilinear operator, a_h the discrete operator (the EFGM formulation), ϕ as again the Moving Least Squares Interpolants and $\tilde{u}$ any arbitrary linear polynomial:

$$\tilde{u} \in P_1, \quad P_1 = \left\{ p \,|\, p = c_0 + c_1\, x, \;\; c_0,\, c_1 \text{ const.} \in \mathbb{R} \right\}$$

Rigid motions may be represented in this case with $c_1 = 0$, while the constant strain state requires $c_1 \neq 0$.

In [29] it is shown that for the case of discontinuous shape functions (see Section 3.5) this requirement is only satisfied in the limit, but it is said that convergence is demonstrated. Continuous shape functions, however, though being rational, can interpolate polynomials of certain order exactly. See Belytschko et al. [7] and Nayroles et al. [34].

3.6 Relations of Field Solution Methods

Without any intention of completeness, some relationships within the area of approximate field solution methods are described.

[6] Note, that the term "consistency" is equivalent to "completeness" [37, p. 175].

3.6.1 EFGM — DEM

The EFGM is an improved formulation of the diffuse element method [7]. The formulation of derivatives, the imposition of essential boundary conditions and the integration accuracy have been modified to yield the EFGM.

3.6.2 EFGM — FEM

The Finite Element Method with linear interpolation functions may be considered a special case of the Element Free Galerkin Method with piecewise constant weight functions and limited domains of influence [36], see also Figure 5-4.

One main advantage of the EFGM is the simplified process of remeshing. This advantage is clearly visible in the case of fracture analysis [8, 10, 32, 35, 36, 41]. Arbitrary crack growth, simulated by FEM, involves extensive remeshing, including an additional source of error and computational cost. However, the computational burden induced by the Moving Least Squares Interpolants in EFGM may balance this disadvantage of the FEM. There are attempts to decrease this cost [5, 26, 31], but it seems to be certain that the EFGM will not reach the efficiency of the FEM.

In the EFGM, there are no single elements which may be examined in the patch test. Instead, the disretization in its entirety may be examined.

One of the more subtle but important differences between FEM and EFGM may not be assigned directly to some characteristics in the computational process. The maturity of the FEM, its long history and the abundant amount of experience gained with the FEM, in comparison to the EFGM, is the most important argument voting for the FEM. The EFGM has to "compete" with the established rival FEM — and obviously, this situation is difficult for the newcomer.

3.6.3 EFGM — hp Clouds

In some cases the construction of a signed partition of unity, like in the method of hp clouds, leads to identical shape functions as in the EFGM [5].

3.6.4 EFGM — RKPM

The resulting shape functions of the EFGM and the Reproducing Kernel Particle Method are equivalent [16]. Belytschko et al. showed that in general, when the consistency requirement is imposed, approximations constructed by Kernel methods and Moving Least Squares Interpolants are identical [6].

3.6.5 EFGM — SPH

In the Smooth Particle Hydrodynamics Method, where in general the governing differential equations are usually imposed directly at the nodes, no quadrature is needed [32]. However, minimizing the residual essentially by a collocation method, as in SPH, yields less accurate results than Galerkin methods, since in the latter the residual is minimized over the whole domain. But due to quadrature, the EFGM is substantially more expensive than SPH [6].

It is shown in [32], that the EFGM with Shepard interpolants (Moving Least Squares Interpolants with a constant basis), is almost identical to the SPH method.

Chapter 4

Implementation

4.1 Introduction

In this thesis the Element Free Galerkin Method is applied to elastic rods using Matlab. General considerations of this implementation are given, followed by pictorial and textual descriptions of the routines.

4.2 Program Description

Flowcharts of most of the routines are shown here. Oval boxes represent subroutines and rectangular boxes represent steps performed within the routine. A summary of the required input is given in Section 4.3.

Flowcharts of the routines `weight.m`, `pbase.m` and `gausstable.m` are not shown. These routines are straightforward in computation, work simply like functions and need not be described below.

The routines are not designed with the aim of superior efficiency. Some of the applications may be designed faster, like:

- accelerated computation of the Moving Least Squares Interpolants as in [5].
- efficient storage of discretizaion data and employing search algorithms as mentioned

in [7].

- computation of weight functions in normalized coordinates using $\bar{d}_i = \frac{d_i}{d_{mi}}$.

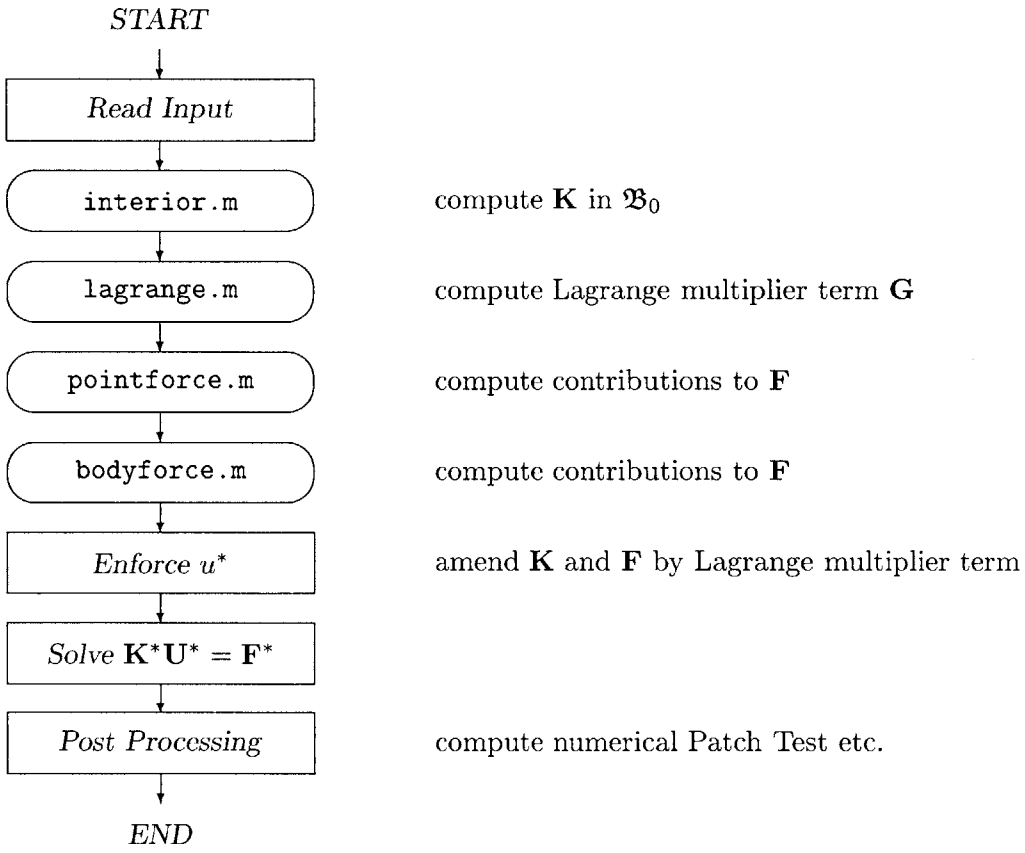

Figure 4-1: Flowchart of `process.m`

4.2.1 `process.m`

The routine `process.m` is the main part of the program, it reads the input data, calls all subroutines, assembles and solves the whole system and processes the output.

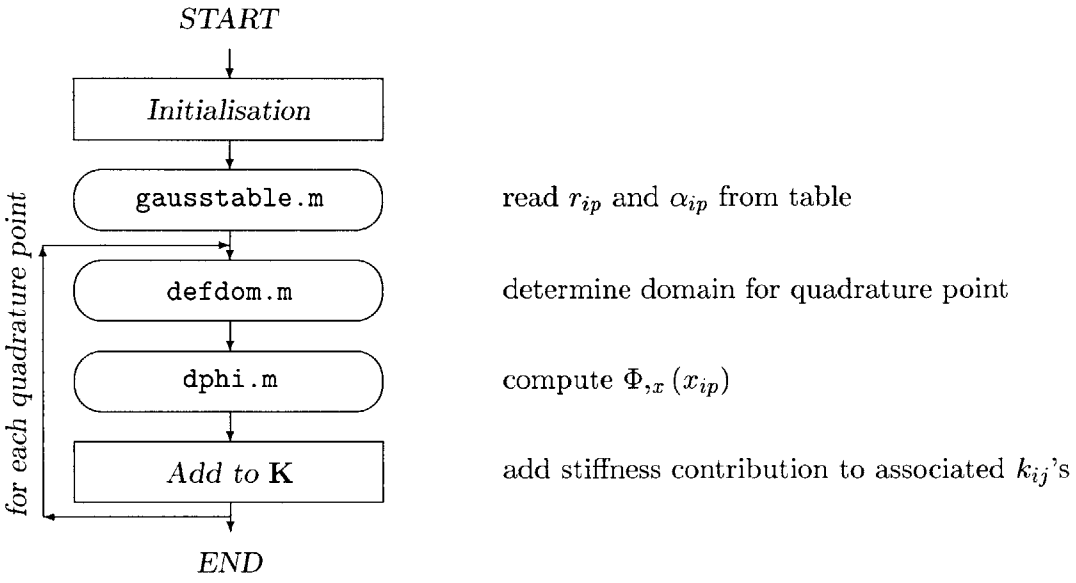

Figure 4-2: Flowchart of `interior.m`

4.2.2 `interior.m`

The function `interior.m` computes the stiffness matrix of the rod under consideration. It loops over each quadrature point, denoted x_{ip}, within each integration cell. Each contribution is added to the total stiffness matrix $\mathbf{K}$ using the LM vector defined by `defdom.m`. Note that this $\mathbf{K}$ matrix is not amended yet by constraints placed by essential boundary conditions.

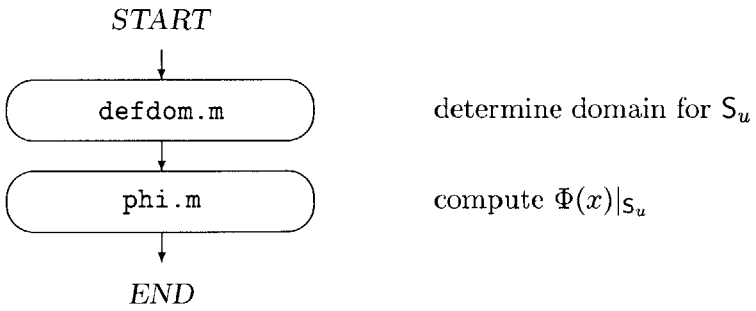

Figure 4-3: Flowchart of `lagrange.m`

4.2.3 `lagrange.m`

The function `lagrange.m` computes the terms imposing the essential boundary condition within the mathematical model, see eq. (A.28). It employs, as in the definition in eq. (A.28), the interpolation function, `phi.m`, evaluated at S_u.

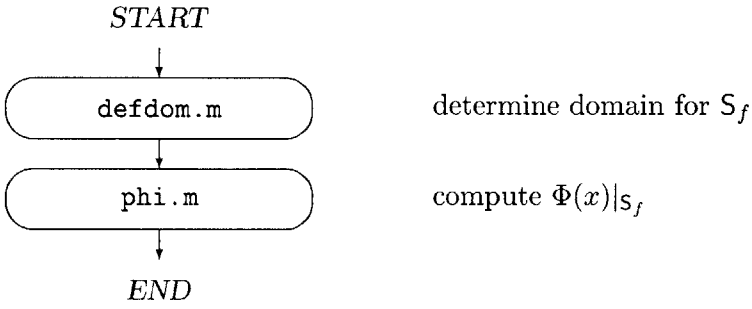

Figure 4-4: Flowchart of `pointforce.m`

4.2.4 `pointforce.m`

The function `pointforce.m` determines the contribution of a point force applied to the rod, see eq. (A.30). It is similar to `lagrange.m`, since both functions determine the influence of some point load on the system (cf. the physical meaning of the Lagrange multiplier in Section A.3.2).

4.2.5 `bodyforce.m`

For the case of a distributed load, `bodyforce.m` computes the contribtutions of this load to the vector $\mathbf{F}$. It is similar to `interior.m`, but integrates over the product of the shape function of the force with $\Phi(x)$, while `interior.m` integrates over the product $\Phi_{,x}^T(x)\Phi_{,x}(x)$.

4.2.6 `defdom.m`

This routine computes the domain of influence and the LM vector (relating the node numbering in the domain of influence to the global node numbering) for the evaluation point

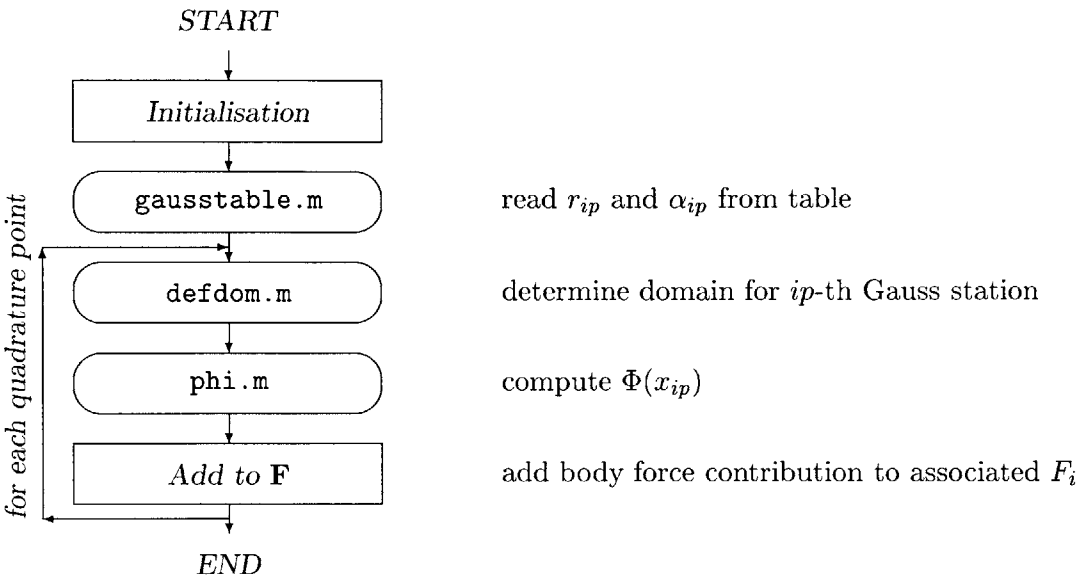

Figure 4-5: Flowchart of `bodyforce.m`

x. `defdom.m` sorts, in essence, the nodes by distance from the evaluation point, cuts off the sorted vector after m elements: it computes c_i (in general the distance between x and the m-th next node), assures interpolation instead extrapolation[1]. Then it determines d_{mi} and establishes the domain vector containing all nodes with nonzero weights.

Some computational effort is spent for this routine. Subdivision of $\mathfrak{B}$ prior to discretization (the assignment of nodes to cells enables shortened neighborhoodlists), storage of neighborhood data, or as mentioned in the introduction, employing other search algorithms, may reduce this effort.

4.2.7 `dphi.m`

`dphi.m` computes the first derivative of the interpolation functions $\Phi_{,x}(x)$. It employs the subroutines `weight.m` and `pbase.m` and is a subroutine itself for `interior.m`, enabling the computation of the stiffness matrix.

[1] Belytschko et al. expressed this (for two dimensions) by requiring that the minimum set of neighboring nodes construct a polygon around point $\mathbf{x}_I$ [7].

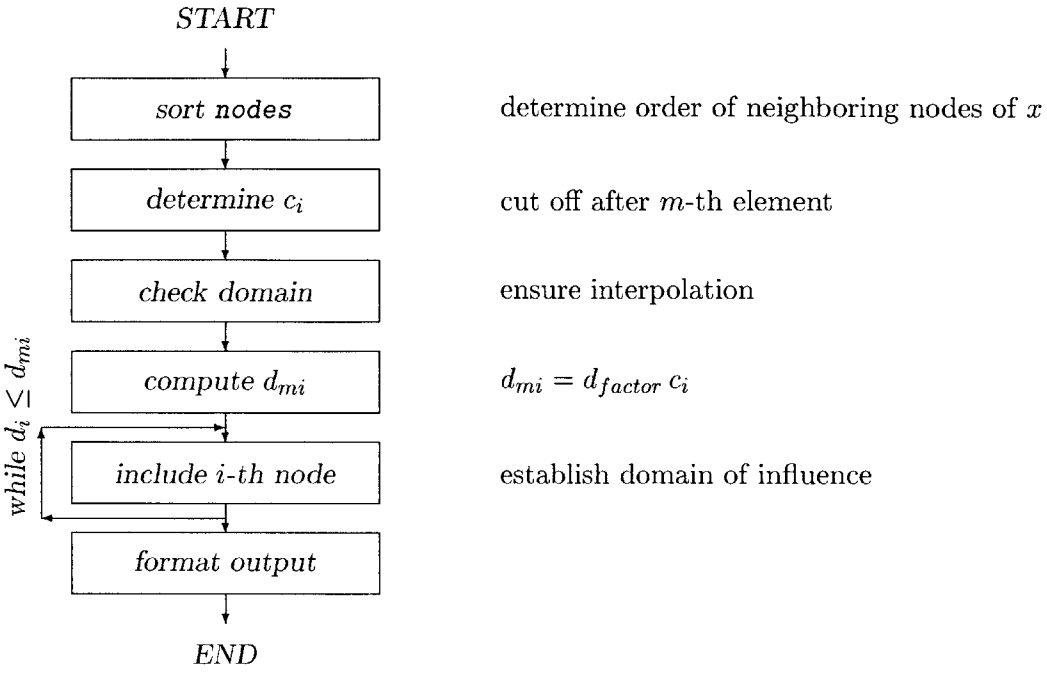

Figure 4-6: Flowchart of `defdom.m`

Note that d_i is not computed as the absolute value, $|x - x_i|$, as mentioned in Section 2.2.4 and in [7]. Instead, d_i in `dphi.m` and `phi.m` may also be negative. However, this is not erroneous. Consider the first derivative of the weight function:

$$\frac{dw}{dx} = \frac{dw}{dd}\,\frac{dd}{dx},$$

$$\frac{dd}{dx} = \begin{cases} -1 & \text{if } d < 0, \\ +1 & \text{if } d > 0. \end{cases}$$

Note that $w(d)$ is axially symmetric, $w(d) = w(-d)$, and that $w_{,x}(-d) = -w_{,x}(d)$ holds for the employed weight functions. Therefore, the formulation of the one dimensional case of the Moving Least Squares Interpolants as in `dphi.m` and `phi.m` is correct.

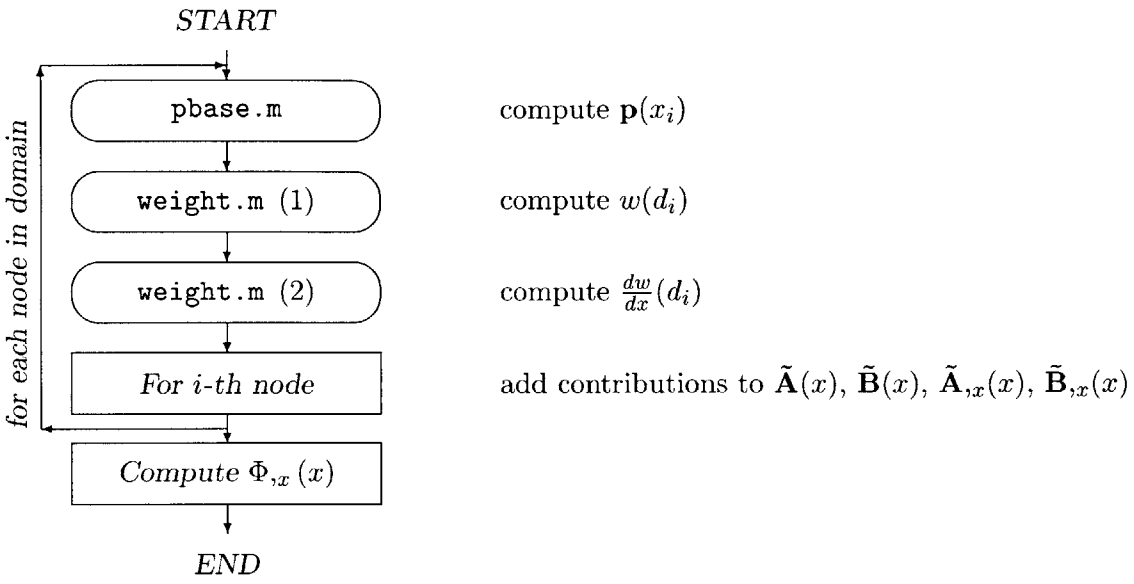

Figure 4-7: Flowchart of `dphi.m`

4.2.8 `phi.m`

The routine `phi.m` computes numerically (similar to `dphi.m`) the Moving Least Squares Interpolants $\Phi(x)$. It calls the subroutines `weight.m` and `pbase.m`. Note the remarks on the definition of d_i for `dphi.m`.

4.2.9 `weight.m`

The weight functions defined in Section B.5 are computed here. This routine also computes the derivatives of the weight functions necessary for $\Phi_{,x}(x)$.

4.2.10 `pbase.m`

This Matlab function is rather short and only determines the i-th derivative of the basis vector $\mathbf{p}(x)$.

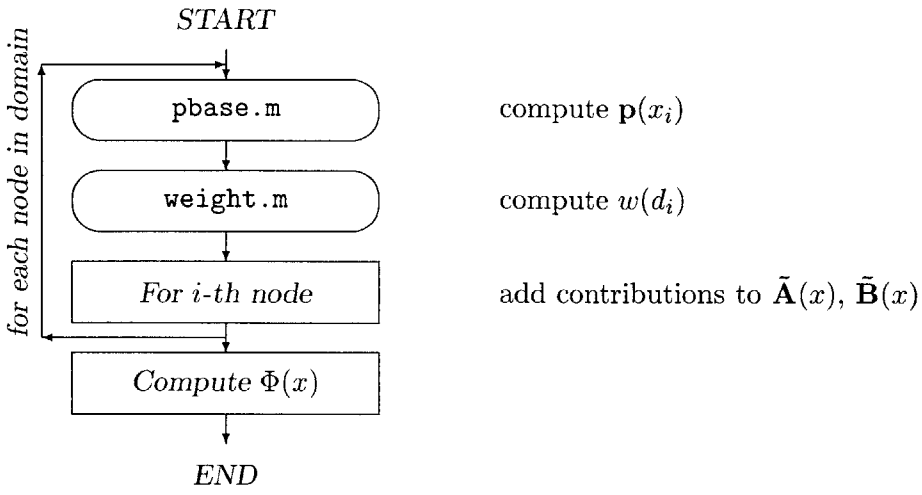

Figure 4-8: Flowchart of `phi.m`

4.2.11 `gausstable.m`

The normalized coordinates and weights for Gaussian quadrature, up to order six, are stored in this function.

4.3 Program Input

The routine `process.m` is the main routine. It solves the final system of equations and produces an output sheet containing results. The following data has to be input by the user.

- `boundary`: 2×1 vector containing x-coordinates of ends of rod.
- `nodes`: $N \times 1$ vector containing x-coordinates of nodes.
- `approxcelllength`: approximate integration cell length (exact cell length to be determined by routine).
- `integrationorder`: order of Gaussian quadrature in each cell.
- `essbc`: x-coordinate of essential boundary condition (by default set to first element of vector boundary).
- `ustar`: prescribed displacement at `essbc`.
- `pointforceswitch`: if point load is applied, this value has to be set to one.
- `pointforcex`: x-coordinate of point force (by default set to last element of vector boundary).
- `pointforcevalue`: value of point force.
- `bodyforceswitch`: has to be set to 1 if distributed load is applied.
- `bodyforceboundary`: 2×1 vector containing interval in which bodyforce is applied.
- `bodyforcevalue`: value of distributed load.
- `bodyforcetype`: shape of applied distributed load:
  - 0: constant load.
  - 1: linear load, with $f^b = 0$ at x equal to first element of `bodyforceboundary`.

  - 2: quadratic load, with $f^b = 0$ at boundaries of distributed load.

- `mlsdfactor`: factor determing d_{mi}, in general: $2.0 \leq d_{factor} \leq 3.0$.

- `mlsm`: dimension of basis, m. Determines basis vector $\mathbf{p}$ and minimum amount of nodes required in domain of influence.

- `mlswtype`: type of weight function:

  - $2, 4, 6, 8, 10$: singular weight functions from [30]. As mentioned in Section 2.2.4, some approximation required for the case $x = x_i$.

  - 11: weight function defined in [41].

  - 14: weight function defined in [20].

Chapter 5

Computation

5.1 Introduction

This Chapter summarizes some of the numerical examples stated in Appendix B. The nodal distribution, nodal density and integration accuracy are varied, numerical Patch tests are performed, examples with different weight functions are computed.

In general, the rod in Figure A-1 with unit stiffness ($EA = 1$) is discretized with one node on each boundary $\partial\mathfrak{B} = \mathsf{S}_u \cup \mathsf{S}_f$ and nodes in the interior $\mathfrak{B}_0$. The background quadrature cells are distributed regularly with constant cell length.

The total strain energy of the EFGM solution is computed as:

$$\mathfrak{E}^h = \frac{1}{2}\mathbf{U}^{\mathbf{T}}\mathbf{K}\mathbf{U} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{U}^{\mathbf{T}}\mathbf{F}.$$

Note that $\mathbf{K}$, $\mathbf{U}$ and $\mathbf{F}$ do not include the additional terms stemming from the Lagrange multipliers.

Several examples with rods subject to point loads and distributed loads are examined. In each case, various node distributions are employed. These distributions are shown in Figure 5-1. If not otherwise mentioned, in general $d_{factor} = 2.0$.

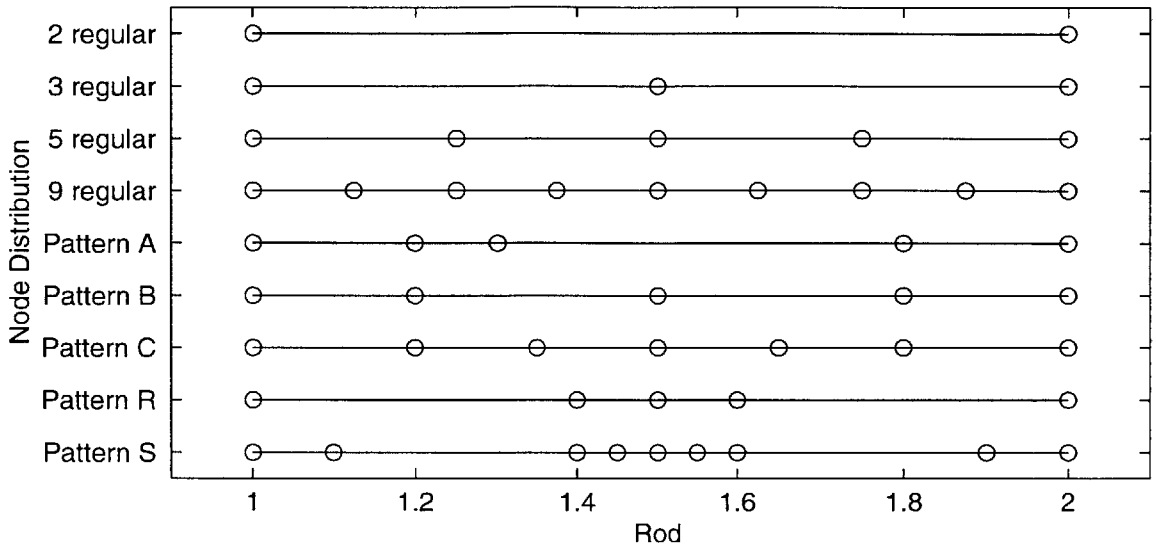

Figure 5-1: Node distributions employed in numerical examples

5.2 Accuracy of Results

The numerical examples stabilize with increasing order of integration. In general, the computations, employing an at most quadratic basis ($m = 3$), perform better than computations with $m = 2$, while the former require more operations. All examples stated in Appendix B converge to the exact solution, except for two node distributions in the case of a point load in the middle.

Consider Figures 5-2, 5-3 and 5-6 to 5-13. In general, the EFGM seems to perform well for smooth solution fields, while discontinuities deteriorate the accuracy of the method. Moving Least Squares Interpolants are unable to represent jump discontinuities (for example due to point forces) exactly[1].

In Figures 5-4 and 5-5 the size of the domain of influence is changed. Shortening the

[1]Note that the Moving Least Squares Interpolants which are used here are continuous and in general not employed in fracture analysis (see conforming EFGM in [29]). Instead, employing the visibility criterion, fracture analyses are performed in general with the non-conforming EFGM using discontinuous Moving Least Squares Interpolants.

domains to their limits by employing $d_{factor} \gtrapprox 1.0$ yields results similar to the FEM. But the obtained solution is smooth, since the weight functions are not discontinuous as in the FEM (1.0 in the neighboring elements and 0 elsewhere).

The case of a widened domain of influence is shown in Figure 5-5. The displacements and strains are underestimated and too smooth, as mentioned in [36].

5.2.1 Numerical Patch Test

For each discretization, the sum over each row is computed. This sum is a numerical patch test for rigid motions, at best it should be equal 0, but errors due to roundoff etc. have to be accepted.

The sum of the rows of the unconstrained stiffness matrices $\mathbf{K}$ are: (1) for the regularly spaced discretizations in all examples in Appendix B always below 10^{-11}, and (2) for the irregular spaced patterns always upper-bounded by 10^{-8}. Therefore, it seems that the patch test for rigid body displacements is passed.

The examples in Section B.2.1 are in essence a Patch test for the constant strain state. A linear displacement shape and constant strain are the exact results for this load case. All examples with a point force applied at the tip converge to the exact strain energy, that is, the constant strain state can be represented (see also [7]).

5.2.2 Point Force Applied in the Middle

Not all results in Appendix B converge to the exact solution. An exception is the case of a point force applied in the middle of the rod, while supported at the left end. Neither the two node regular pattern nor the irregular node distribution Pattern A converge to the exact strain energy (Tables B.3, B.4). Instead, the two node discretization converges to 1/2 of the exact solution. However, this erroneous result, due to the coarseness of the discretization, has to be accepted. The result is equivalent to solutions obtained with the FEM employing a single two node truss element.

In the case of nodes distributed in the shape of Pattern A and employing a linear basis

($m = 2$), the strain energy oscillates with increasing order of integration near $\mathfrak{E}^h \approx 0.1866$. This result is not acceptable. However, as in the forementioned case, this result is similar to the solution of a finite element discretization employing the same node distribution. The strain energy obtained from two node linear truss elements is exactly $\mathfrak{E}^h_{FEM} = 0.1900$. As expected, neither FEM nor EFGM are able to pick up exactly a point load when no nodes are in the proximity.

5.2.3 Effects of Discretization

The employed discretizations:

- regularly spaced grid with 5 nodes,
- Pattern B, Pattern C (moderately irregularly spaced) and
- Pattern A (highly irregularly spaced)

employ the same number of nodes.

Neglecting the exceptional case of a point force applied in the middle, Section 5.2.2, the Element Free Galerkin Method does not exhibit significant changes in strain energy for modest variations of the node distribution.

5.3 Computational Effort

The total number of flops (floating point operations, see Matlab documentation) increases significantly with increasing order of basis. Computations with $m = 3$ in general require $2.0 \ldots 2.5$ as many flops as computations with $m = 2$.

Holding the amount of nodes, dimension of basis, m, and the type of weight function fixed, the computational cost remains constant.

5.4 Variation of Weight Function

All of the results stated in Tables B.1 - B.8 are obtained using the exponential weight function defined by Belytschko et al. [7] and refined by Häussler-Combe et al. [20]. However, various weight functions have been employed within the EFGM, especially polynomials, splines as in [14, 25, 28].

Within this research, three weight functions are compared. They are defined in Section B.5. The results are given in Tables B.9 - B.11. As expected, the singular weight functions L2, L4 and L6 perform rather poorly. The results obtained with the weight functions Ta and HC do not differ much, but improved results are obtained employing weight function type HC for the case of $m = 3$.

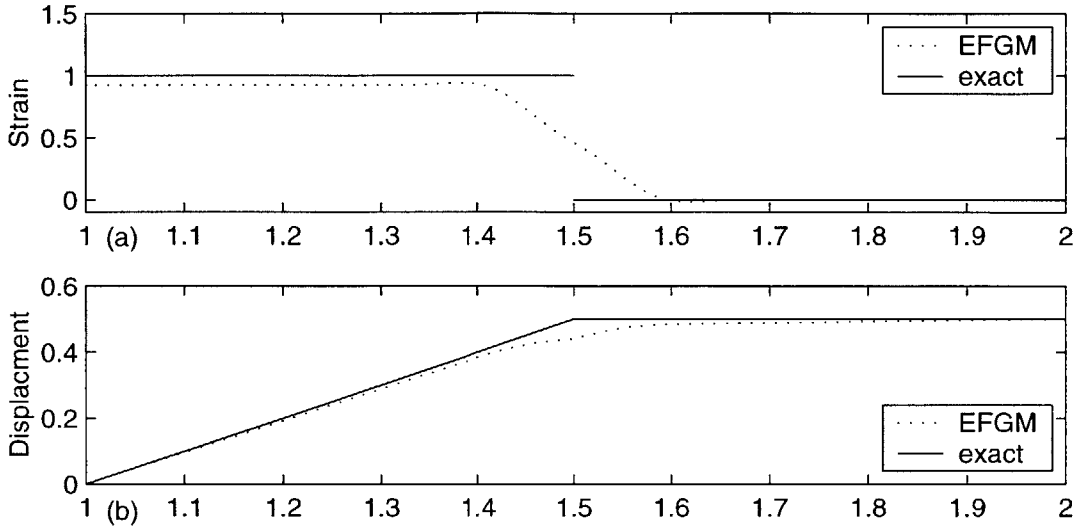

Figure 5-2: (a) Strain, (b) displacement of rod subject to point force in the middle ($x = 1.5$), 3 nodes, regularly distributed, $m = 2$

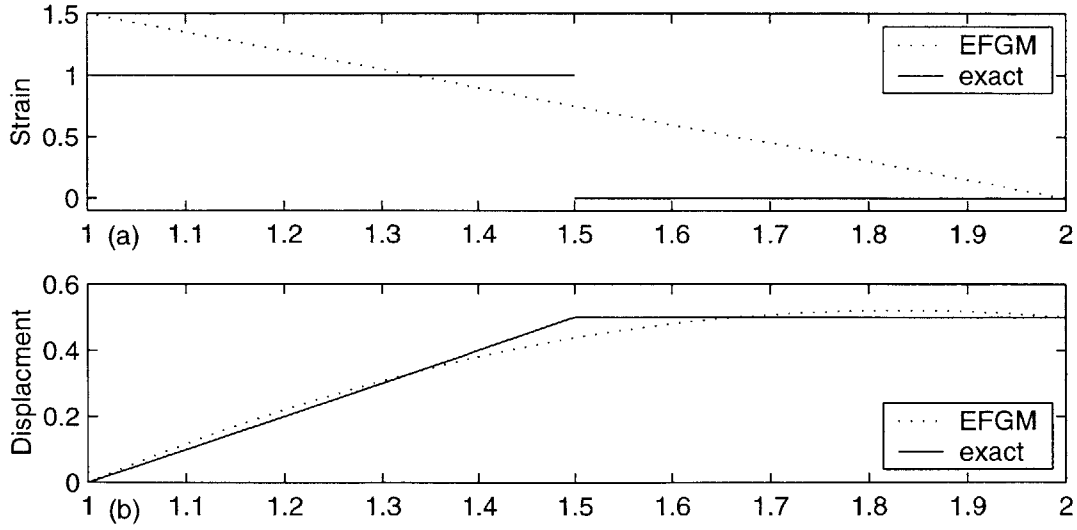

Figure 5-3: (a) Strain, (b) displacement of rod subject to point force in the middle ($x = 1.5$), 3 nodes, regularly distributed, $m = 3$

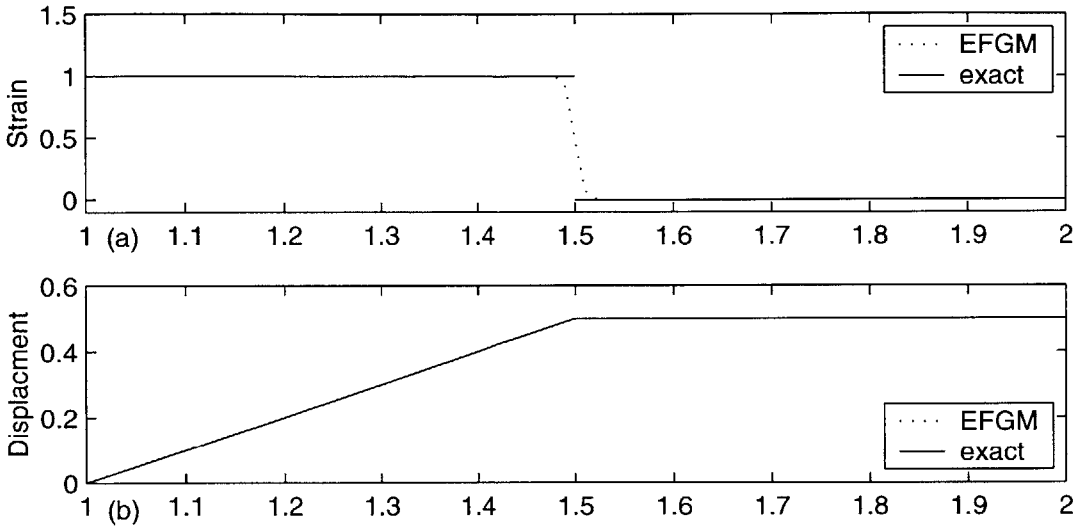

Figure 5-4: (a) Strain, (b) displacement of rod subject to point force in the middle ($x = 1.5$), 3 nodes, regularly distributed, $m = 2$, $d_{factor} = 1.01$

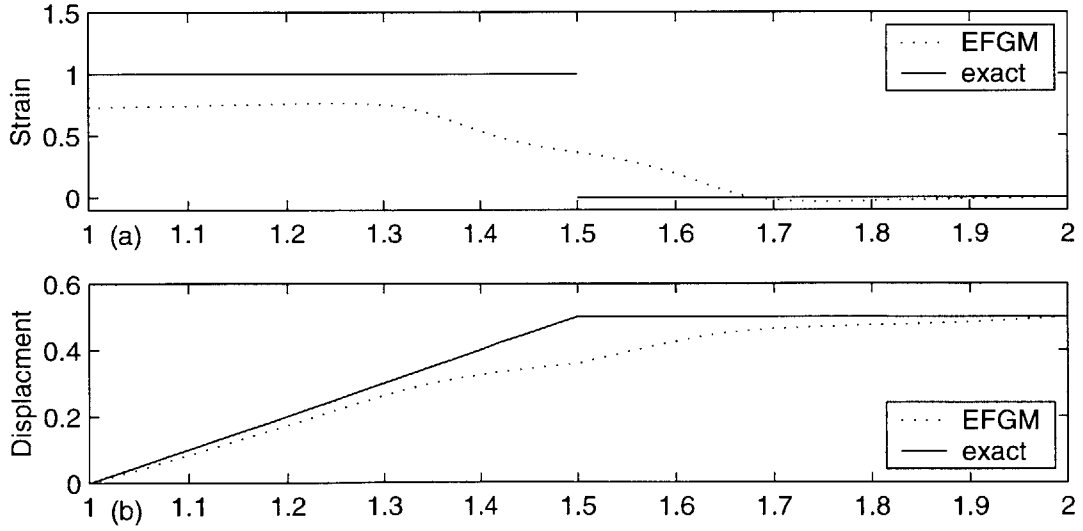

Figure 5-5: (a) Strain, (b) displacement of rod subject to point force in the middle ($x = 1.5$), 3 nodes, regularly distributed, $m = 2$, $d_{factor} = 3.0$

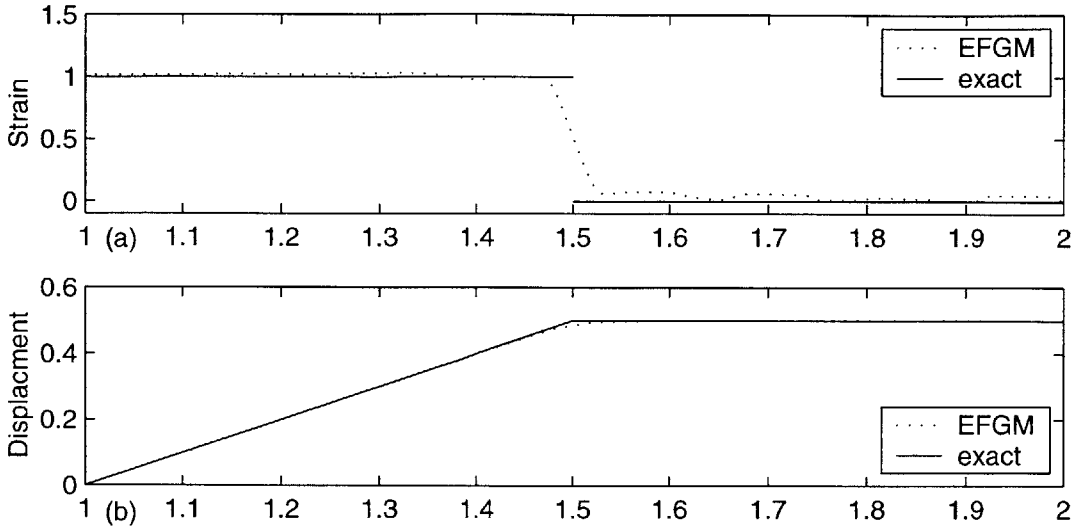

Figure 5-6: (a) Strain, (b) displacement of rod subject to point force in the middle ($x = 1.5$), 9 nodes, regularly distributed, $m = 2$

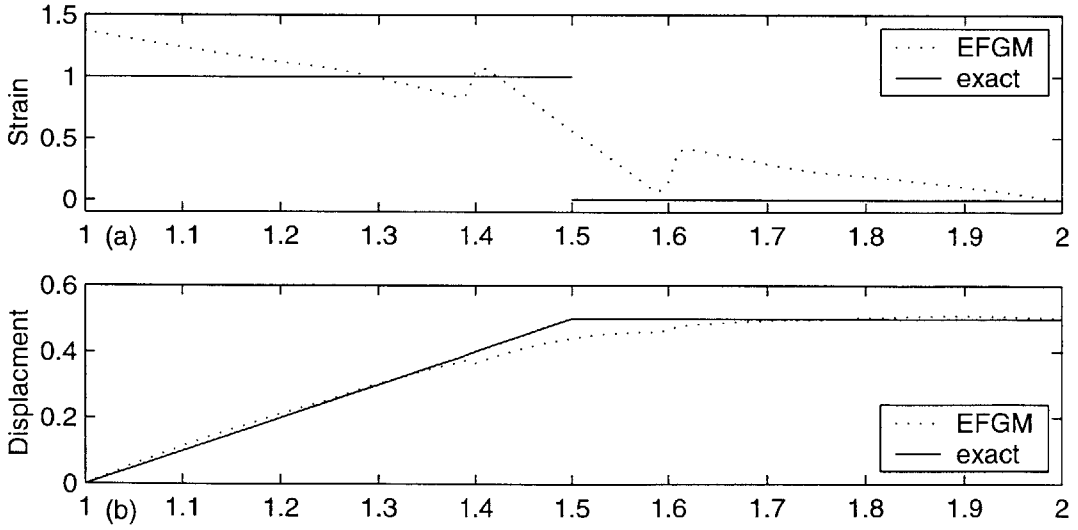

Figure 5-7: (a) Strain, (b) displacement of rod subject to point force in the middle ($x = 1.5$), 5 nodes, Pattern R, $m = 2$

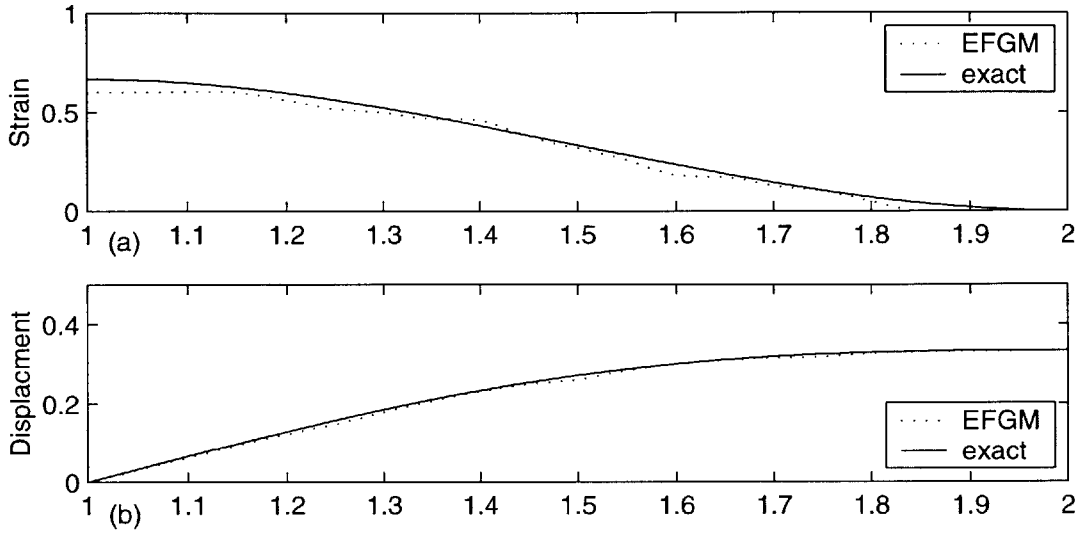

Figure 5-8: (a) Strain, (b) displacement of rod subject to quadratic body force, 5 nodes, regularly distributed, $m = 2$

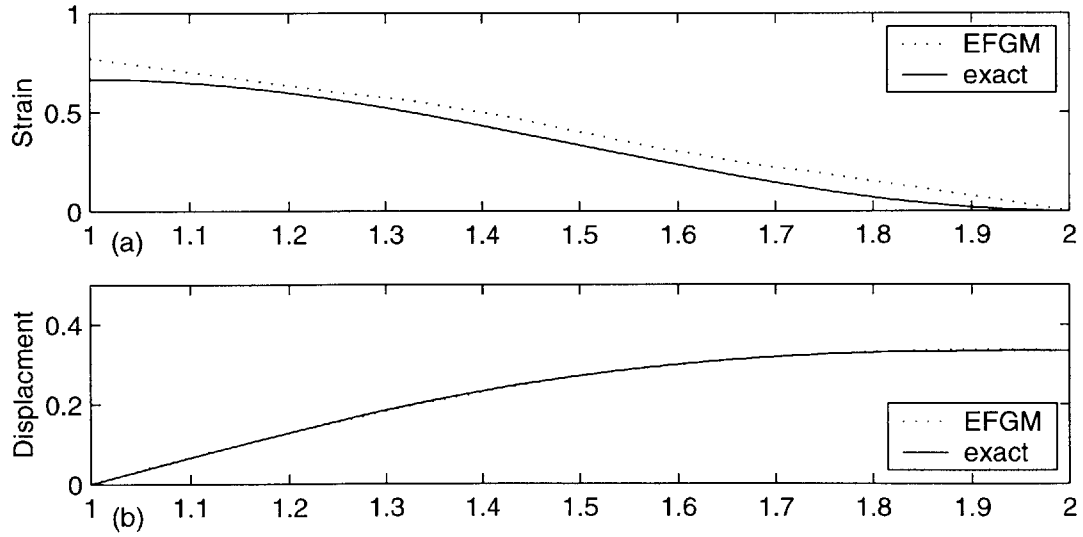

Figure 5-9: (a) Strain, (b) displacement of rod subject to quadratic body force, 5 nodes, regularly distributed, $m = 3$

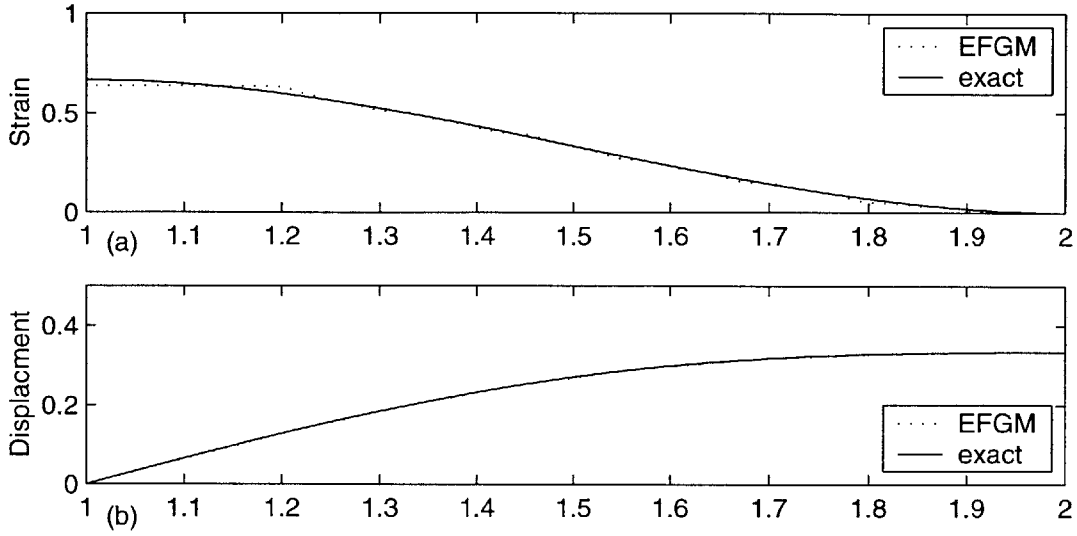

Figure 5-10: (a) Strain, (b) displacement of rod subject to quadratic body force, 9 nodes, regularly distributed, $m = 2$

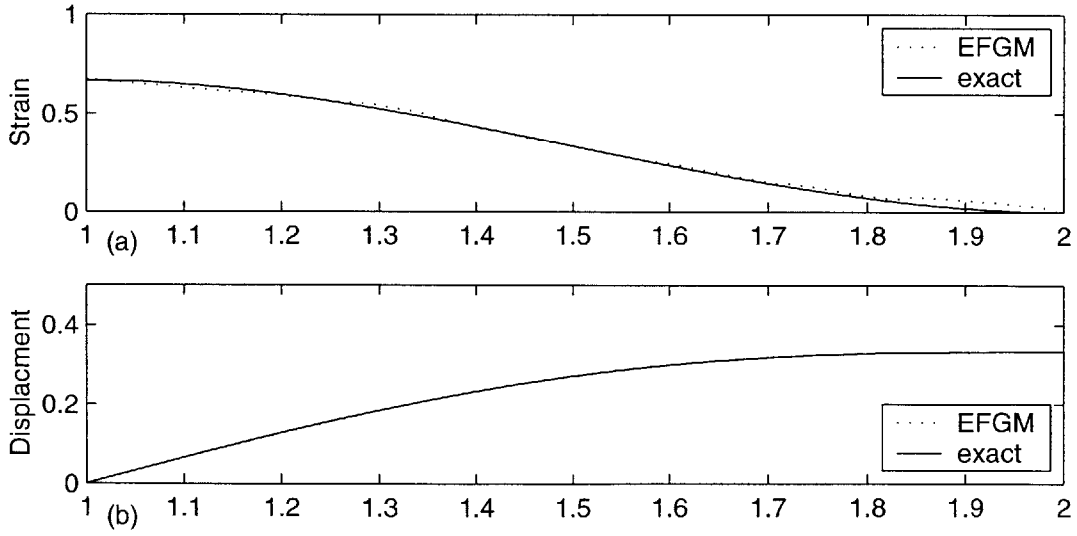

Figure 5-11: (a) Strain, (b) displacement of rod subject to quadratic body force, 9 nodes, regularly distributed, $m = 3$

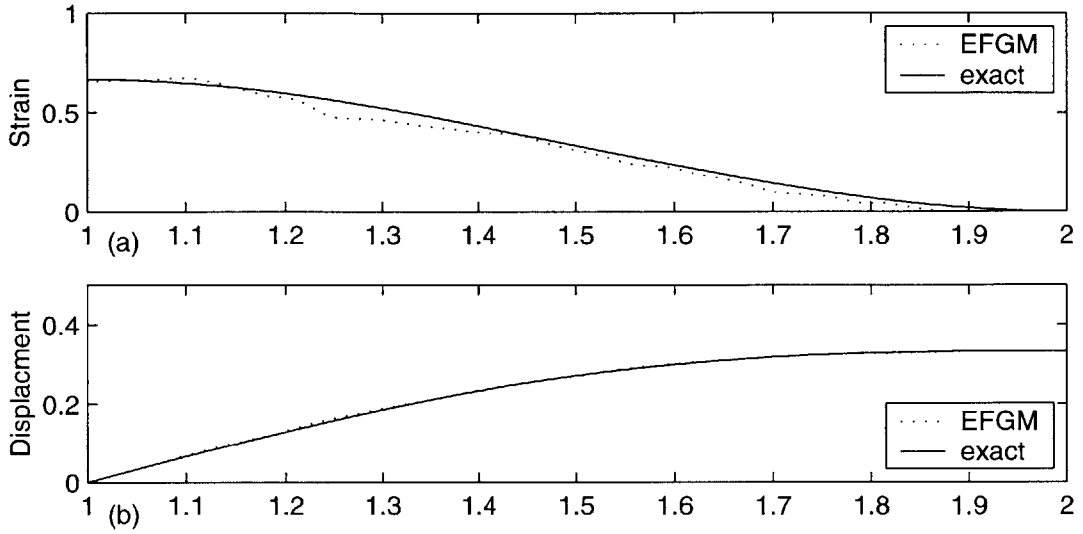

Figure 5-12: (a) Strain, (b) displacement of rod subject to quadratic body force, 7 nodes, Pattern C, $m = 2$

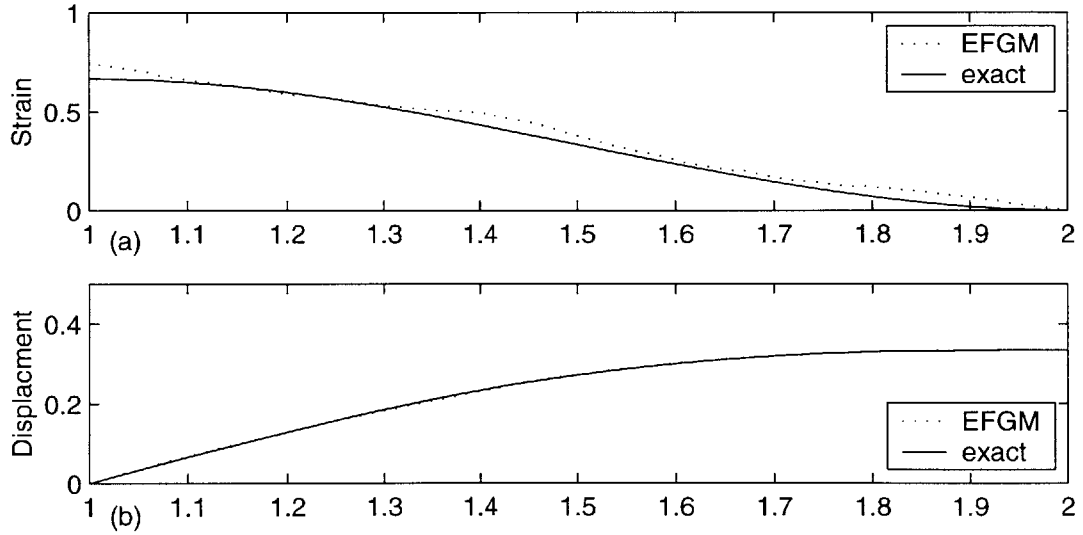

Figure 5-13: (a) Strain, (b) displacement of rod subject to quadratic body force, 7 nodes, Pattern C, $m = 3$

Chapter 6

Conclusion

6.1 Summary

The intent of this thesis is to implement and examine the Element Free Galerkin Method, a meshless method, for the analysis of elastic rods.

First, in Chapter 1, an introduction to approximate field solution methods was given, followed by a brief overview of the EFGM. Then, in Chapter 2, the mathematical tools, like Moving Least Squares Interpolation, Lagrange multipliers, numerical integration and error distribution methods were reviewed. Chapter 3 developed specifics of the Element Free Galerkin Method and explained relations between different numerical field solution methods. Chapter 4 described the specific structure of the implementation of the EFGM employing Matlab. The mathematical formulation for these routines was developed in Appendix A. In Chapter 5 numerical examples, stated in Appendix B, were summarized and significant results outlined.

6.2 Results

The EFGM has been successfully applied to elastic rods. The implementation of the EFGM in this thesis uses Lagrange multipliers to enforce essential boundary conditions and employs continuous Moving Least Squares Interpolants (also referred to as the conforming EFGM).

The following characteristics of the EFGM have been shown:

- In general, the EFGM (neglecting any considerations with respect to fracture and similar analyses) is well-suited for smooth problems. However, jump discontinuities in derivatives (for example induced by point forces) may result in deficient performance of the method. This erroneous behaviour can in general be circumvented by employing appropriately highly refined disretizations and by the accurate imposition of natural boundary conditions [23].

- In accord with other research [7, 16], the EFGM requires a rather large order of integration to assure convergence and accuracy.

- Excluding discontinuous solution fields, the EFGM seems rather insensitive to modest changes in nodal spacing (also noted by Belytschko et al. [7]).

- The accuracy of the EFGM is in general superior to the accuracy of a Finite Element solution employing the same amount of nodes (due to the smoothness of the Moving Least Squares Interpolants).

Recently much advance took place in the area of the EFGM. However, despite accelerated computation [5], the method is still computationally expensive. But its FEM-exceeding accuracy may offer new possibilities. And with the ever-increasing capabilities in computation, the detriment of high computational cost of the EFGM may fade in future.

Appendix A

Rod in Tension

A.1 Introduction

In this Appendix the mathematical formulation of the EFGM for the case of a rod, constrained by Lagrange multipliers, is developed. Departing from fundamental considerations, the differential equation of equilibrium for a rod and its weak form, amended by additional constraints, is derived. Approximating the displacement field $u(x)$ by Moving Least Squares Interpolants leads to the formulation of the EFGM for an axially loaded rod.

A.2 The Differential Equation of Equilibrium

A.2.1 Introduction

In this Section the basic equations of the linearized theory of elasticity for a one dimensional problem (rod) is derived. As far as the generalized theory is concerned, the notation of [19] is used. For an extensive and accurate derivation the reader may refer to [19, 33].

A.2.2 Fundamental Principles in Linearized Elasticity

In the following, the mathematical problem statement of a rod subject to axial forces is derived from the general principles in the theory of linearized elasticity. Note that this

section does not intend to give some rigorous derivation.

Consider the constitutive law , or stress-strain-relationship, within the linearized theory in three dimensions:

$$\mathbf{S} = \mathsf{C}\,\mathbf{E}, \tag{A.1}$$

with $\mathbf{S}$ denoting the first Piola-Kirchhoff stress tensor[1], C the elasticity tensor and $\mathbf{E}$ the infinitesimal strain tensor. $\mathbf{E}$ is defined as the symmetric part of the displacement gradient:

$$\mathbf{E} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T). \tag{A.2}$$

The strain-displacement relationship in eq. (A.2) assures compatibility of the fields $\mathbf{E}$ and $\mathbf{u}$.

In general, C is a fourth-order tensor with $3\times3\times3\times3 = 81$ elements. But in the linearized theory it is assumed that $\mathbf{S} \approx \mathbf{T}$, that means the Cauchy stress tensor is a sufficiently accurate representation of the first Piola-Kirchhoff stress tensor. The Cauchy stress tensor $\mathbf{T}$ and the infinitesimal strain tensor $\mathbf{E}$ (by definition) are symmetric. Therefore, with no loss of generality, it is acceptable to assume $6 \times 6 = 36$ independent coefficients in C.

For the case of an isotropic material[2], the constitutive relationship, eq. (A.1), may be represented by only two independent coefficients, the Lamé constants :

$$\mathbf{S} = 2\mu\mathbf{E} + \lambda(\mathrm{tr}\,\mathbf{E})\mathbf{I}, \tag{A.3}$$

where

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)},$$
$$\mu = G = \frac{E}{2(1+\nu)},$$

[1]Note: In general, the first Piola-Kirchhoff stress tensor is not symmetric.
[2]Isotropic materials have no preferred direction.

with E the modulus of elasticity, or Young's modulus, ν the Poisson ratio and G the shear modulus.

Note that the preceding assumptions were considering an infinitesimally small element only. Neither have assumptions been made yet on the homogeneity of the material, nor on the geometry of the body. For a homogeneous body, λ and μ are constant throughout the whole body. But for the case of inhomogeneities it is important to include the material properties exactly in eq. (A.4).

The differential equation of equilibrium enforces equilibrium of the external forces / tractions and internal forces / stresses. It is based on the equation of motion, which, for the case of negligible inertia effects, or the static case (when $\ddot{\mathbf{u}} = 0$) reduces to:

$$\operatorname{Div} \mathbf{S} + \mathbf{b}_0 = \mathbf{0}. \tag{A.4}$$

This can be written, using eqs. (A.2, A.3), in the form:

$$\mu \, \Delta \mathbf{u} + (\lambda + \mu) \nabla \operatorname{Div} \mathbf{u} + \mathbf{b}_0 = \mathbf{0}. \tag{A.5}$$

Now consider the case of a rod subject to tension, simplified to a one-dimensional problem. The stress tensor is required to take the form:

$$\mathbf{S} = \begin{pmatrix} \sigma & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

with some σ.

To obtain the corresponding strain field, eq. (A.3) has to be inverted:

$$\mathbf{E} = \frac{1}{2\mu} \left[\mathbf{S} - \frac{\lambda}{2\mu + 3\lambda} (\operatorname{tr} \mathbf{S}) \mathbf{I} \right]. \tag{A.6}$$

This yields the following result:

$$\mathbf{E} = \begin{pmatrix} \varepsilon & 0 & 0 \\ 0 & -\nu\varepsilon & 0 \\ 0 & 0 & -\nu\varepsilon \end{pmatrix},$$

with

$$\varepsilon = \frac{1}{E}\sigma,$$
$$\frac{\partial u}{\partial x} = \varepsilon, \qquad \frac{\partial u}{\partial y} = -\nu\varepsilon, \qquad \frac{\partial u}{\partial z} = -\nu\varepsilon.$$

Using these results for the strain field in eq. (A.5) leads finally to:

$$\frac{d}{dx}\left(EA\frac{du}{dx}\right) + f^b = 0, \tag{A.7}$$

where the integration perpendicular to the axis degenerates to the cross section A. It should be noted that eq. (A.7) has to be amended by:

- essential boundary conditions in order to obtain a unique (particular) solution of the differential equation, that means to eliminate rigid body modes and
- natural boundary conditions, if necessary.

A.3 The Weak Form

A.3.1 Introduction

The formulation of eq. (A.7), amended by boundary conditions is the differential form of the problem. An error distribution method, see Section 2.6, may directly be applied to the differential form. For this case (called the strong form), the approximating functions are required to satisfy all boundary conditions [15, p. 231]. However, as in the FEM, the mathematical framework in the EFGM is the underlying variational principle. This is

equivalent to the differential form (Section 2.5), but includes a lower order of differentiation of u. It is called the weak form [37, pp. 3-16, 299-300], where "weak" represents in some sense the constraints imposed on the admissible space of solutions of u. In eq. (A.7) the second derivatives of the field u are required to have finite energy. However, in the weak form, the space of approximate admissible solutions is much larger, since only the first derivatives are required to have finite energy. The restrictions on the approximate solution space are weaker than in the strong form of eq. (A.7).

A.3.2 Derivation

In the following, the weak form of eq. (A.7), subject to some boundary conditions, is derived. The steps are based on the rather physical approach of the method of virtual displacements, or Principle of Virtual Work. For the case of essential boundary conditions the variational formulation is used to incorporate the prescribed displacements via Lagrange multipliers.

Method of Virtual Displacements

Consider the following rod:

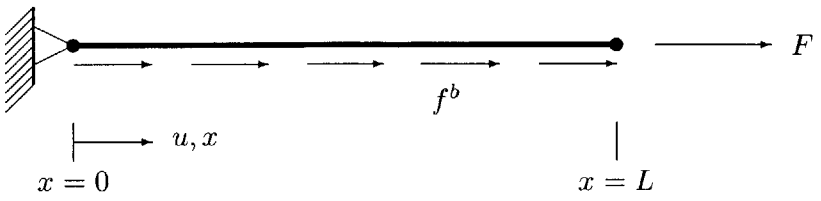

Figure A-1: Rod, axially loaded

with:

E = Young's modulus, u = displacement in x-direction,

F = point force, f^b = body force, in force per unit length,

A = cross section of rod, L = length of rod,

$\mathfrak{B} = \{\, x \mid 0 \leq x \leq L \,\}$, $\partial\mathfrak{B} = \mathsf{S}_u \cup \mathsf{S}_f$,

$\mathsf{S}_u = \{x \mid x = 0\}$, $\mathsf{S}_f = \{x \mid x = L\}$.

Differential equation of equilibrium:

$$\frac{d}{dx}\left(EA\,\frac{du}{dx}\right) + f^b = 0, \tag{A.8}$$

with the following boundary conditions:

- essential boundary condition, prescribed on S_u:

$$u|_{x=0} = u^*, \tag{A.9}$$

- natural boundary condition, prescribed on S_f:

$$EA\,\frac{du}{dx}\bigg|_{x=L} = F. \tag{A.10}$$

Rewrite eq. (A.9) as:

$$u|_{x=0} - u^* = 0. \tag{A.11}$$

In the principle of virtual displacements, the total virtual work of the system is computed by considering a virtually deformed shape. The assumption that the performed virtual work is at some minimum with respect to displacement leads to variational considerations. In mathematical terms, eq. (A.8) is pre-multiplied by some virtual displacement δu, provided

the virtual displacement vanishes on S_u (but is not required to satisfy the natural boundary conditions, see Section 2.5.3):

$$\delta u \frac{d}{dx}\left(EA\frac{du}{dx}\right) + \delta u\, f^b = 0. \tag{A.12}$$

Eq. (A.12) holds for the whole region $\mathfrak{B}$, and therefore may be rewritten as:

$$\int_0^L \delta u \frac{d}{dx}\left(EA\frac{du}{dx}\right) dx + \int_0^L \delta u\, f^b\, dx = 0. \tag{A.13}$$

In order to lower the order of differentiation of u, the following identity (integration by parts) is employed:

$$\int_0^L \delta u \frac{d}{dx}\left(EA\frac{du}{dx}\right) dx = \delta u\, EA \frac{du}{dx}\Big|_{x=0}^{x=L} - \int_0^L \frac{d\,\delta u}{dx} EA \frac{du}{dx}\, dx. \tag{A.14}$$

Rewriting eq. (A.13):

$$\int_0^L \frac{d\,\delta u}{dx} EA \frac{du}{dx}\, dx - \delta u\, EA \frac{du}{dx}\Big|_{x=L} + \delta u\, EA \frac{du}{dx}\Big|_{x=0} - \int_0^L \delta u\, f^b\, dx = 0. \tag{A.15}$$

The second term in eq. (A.15), by using eq. (A.10), can be rewritten as $\delta u|_{x=L}\, F$.

It should be noted that it is assumed, that δu vanishes on S_u. Therefore the third term in eq. (A.15) is zero. But, as mentioned in Section 2.2 and 3.2.3, this assumption is not valid for Moving Least Squares Interpolants $\phi_i(x)$. In other words, prescribed essential boundary conditions are not satisfied exactly by the weak formulation of eq. (A.8), if the test and trial functions are represented using Moving Least Squares Interpolants. Therefore eq. (A.15) has to be amended by additional constraints satisfying the essential boundary conditions. Lagrange multipliers are employed to impose the prescribed displacements. Roughly-speaking: the stiffness matrix obtained from eq. (A.15) is employed to satisfy eq. (A.8) in the interior of region $\mathfrak{B}$ (the rod), while Lagrange multipliers are used to satisfy

the boundary conditions in eq. (A.9).

Therefore, to be consistent with the variational principle, or method of virtual work, it is valid to neglect the third term in eq. (A.15).

Variational Formulation

The variational principle and principle of virtual displacements are consistent and lead to identical results. Resorting to the variational form, considering δu in the following as a variation of u (instead of some virtual displacement) and employing the following identity (chain rule):

$$\delta \left(\frac{du}{dx}\right)^2 = 2\,\frac{du}{dx}\,\delta\,\frac{du}{dx} = 2\,\frac{du}{dx}\,\frac{d\,\delta u}{dx}$$

yields:

$$\delta \left[\int_0^L \frac{EA}{2} \left(\frac{du}{dx}\right)^2 dx - \int_0^L u\, f^b\, dx - u|_{x=L}\, F \right] = \delta\, \Pi. \tag{A.16}$$

Eq. (A.16) is the variation of the functional Π and equivalent to eq. (A.15). This functional is amended by additional constraints imposed by Lagrange multipliers (where c_i denotes the i-th constraint equation):

$$\Pi^* = \Pi + \sum_i \lambda_i\, c_i. \tag{A.17}$$

Here, eq. (A.11) is the only constraint:

$$\Pi^* = \int_0^L \left[\frac{EA}{2} \left(\frac{du}{dx}\right)^2 - u\, f^b \right] dx - u|_{x=L}\, F + \lambda(u|_{x=0} - u^*). \tag{A.18}$$

The functional in eq. (A.16) may be denoted as the total strain energy of the body under consideration. Requiring the minimum strain energy with respect to displacements is

equivalent to invoking the stationarity of eq. (A.18):

$$\delta \Pi^* = 0. \tag{A.19}$$

Note that the variation of the additional constraints requires some attention:

$$\int_0^L \frac{d\,\delta u}{dx}\, EA\, \frac{du}{dx}\, dx - \int_0^L \delta u\, f^b\, dx - \delta u|_{x=L}\, F + \delta\lambda(u|_{x=0} - u^*) + \lambda\, \delta u|_{x=0} = 0. \tag{A.20}$$

Resorting to the principle of virtual work: Eq. (A.20) has to hold for any arbitrary virtual displacement δu and any variation of the Lagrange multiplier $\delta\lambda$. Employing the description of S_u for $x = 0$ and S_f for $x = L$, eq. (A.20) finally leads to:

$$\int_0^L \frac{d\,\delta u}{dx}\, EA\, \frac{du}{dx}\, dx + \lambda\, \delta u|_{\mathsf{S}_u} = \int_0^L \delta u\, f^b\, dx + \delta u|_{\mathsf{S}_f}\, F, \tag{A.21}$$

$$u|_{\mathsf{S}_u} = u^*. \tag{A.22}$$

Eqs. (A.21, A.22) are the weak form of the mathematical model stated in eqs. (A.8, A.9, A.10), amended by constraints. The well-known variational statement of a one-dimensional structure is a special case of eq. (A.21), where the Lagrange multiplier term is not present. Eq. (A.21) reveals the physical meaning of the Lagrange multiplier λ: λ is the negative value of the reaction force, acting at $x = 0$, necessary to prevent rigid body motion.

A.4 Discretization

A.4.1 Introduction

In Section A.3 the continuous formulation of the physical problem is developed. However, the weak form in eqs. (A.21, A.22) of the problem includes only first derivatives of field variables and is therefore more suitable for approximate solution methods than the system of eqs. (A.8, A.9, A.10).

A.4.2 Formulation

Eqs. (A.21, A.22) are only equivalent to eqs. (A.8, A.9, A.10) if they hold for any arbitrary virtual displacement δu, this is for an infinite number of independent test and trial functions. However, in practice, only a finite number of test and trial functions are employed, obviously. But as mentioned in Section 2.6, in the Galerkin method the residual, in satisfying eq. (A.8), is orthogonalized with respect to the solution space V_h. Therefore, the obtained solution is the best approximation in V_h to the exact solution u in V.

Once the expressions in eqs. (A.21, A.22) are obtained, the problem of determining the displacement field $u \in V$ reduces to employing the appropriate solution space V_h which should be at best equal to V or at least be able to come arbitrarily close to functions in V. In Finite Elements, this solution space is determined by the type of elements. In EFGM, Moving Least Squares Approximations are employed to form the space V_h:

$$u \approx \sum_{i=1}^{N} \phi_i(x)\, U_i = \Phi\, \mathbf{U}, \tag{A.23}$$

with

$$\Phi = \begin{pmatrix} \phi_1(x) & \phi_2(x) & \dots & \phi_N(x) \end{pmatrix} \tag{A.24}$$

where $\phi(x)$ is defined in eq. (2.17). As in the Galerkin method, orthogonalizing the residual to V_h leads to:

$$\int_0^L \Phi_{,x}^T\, EA\, \Phi_{,x}\, dx\, \mathbf{U} + \Phi^T\big|_{\mathsf{S}_u}\, \lambda = \int_0^L \Phi^T f^b\, dx + \Phi^T\big|_{\mathsf{S}_f}\, F, \tag{A.25}$$

$$\Phi\big|_{\mathsf{S}_u}\, \mathbf{U} = u^*. \tag{A.26}$$

In accord with the direct stiffness approach[3] N equations concerning the interior are ob-

[3]The direct stiffness approach — in opposite to the flexibility approach — may be described as [24] :

- Prevent any motion in the discretized system by considering all degrees of freedom (U_i) to be fixed,

tained, amended by one additional equation enforcing the essential boundary condition:

$$\begin{pmatrix} \mathbf{K} & \mathbf{G}^T \\ \mathbf{G} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ u^* \end{pmatrix}. \tag{A.27}$$

with the following definitions:

$$\mathbf{G} = \Phi\big|_{\mathsf{S}_u}, \tag{A.28}$$

$$\mathbf{K} = \int_0^L \Phi_{,x}^T \, EA \, \Phi_{,x} \, dx, \tag{A.29}$$

$$\mathbf{F} = \int_0^L \Phi^T f^b \, dx + \Phi^T\big|_{\mathsf{S}_f} F, \tag{A.30}$$

$$\mathbf{U} = \begin{pmatrix} U_1 & U_2 & \dots & U_N \end{pmatrix}^T, \tag{A.31}$$

$$\lambda = \text{the negative of the reaction force acting on } \mathsf{S}_f. \tag{A.32}$$

For ease of notation, let eq. (A.27) be written as:

$$\mathbf{K}^* \, \mathbf{U}^* = \mathbf{F}^*,$$

where $\mathbf{K}^*$, $\mathbf{U}^*$ and $\mathbf{F}^*$ represent the stiffness matrix, amended by the constraint equation, the vector of unknowns and the inhomogeneity terms, respectively.

Note that the point load F in eq. (A.30) is not directly added to the resulting force vector, but instead distributed by the interpolation term $\Phi^T\big|_{\mathsf{S}_f}$, since in the case of Moving Least Squares Interpolants, $\phi_i(x_j) \neq \delta_{ij}$, see Jirásek [23] and Section 3.2.3.

Eqs. (A.27 - A.32) represent the discrete formulation of the weak form in terms of the

---

- Release, one at a time, each degree of freedom, and impose a unit displacement at this degree of freedom. The forces (or moments) required to impose the unit displacements form the elements of the stiffness matrix.

This rather physical approach is in essence the same as variational considerations, or the method of virtual work. The applied unit displacements coincide with the applied virtual deformations, or variations, respectively.

Element Free Galerkin Method, similar to eq. (2.44), with u, v approximated by Moving Least Squares Interpolants. Once established, the system of equations (A.27) has to be solved for the U_i and the (negative) reaction force λ. In order to determine the displacement shape, the solution $\mathbf{U}$ has to be introduced in eq. (A.23) to obtain the deformed shape.

Appendix B

Numerical Examples

B.1 Introduction

Computational examples, rods subject to point loads and distributed loads are presented here. The order of integration, the basis vector $\mathbf{p}$ and the type of weight function are varied. The employed node distributions are shown in Figure 5-1. In Section B.5 typical weight functions are stated.

Highly erroneous results (results like "not a number" given by Matlab), due to a too low order of integration, are marked with a - in the column for total strain energy.

B.2 Point Force

B.2.1 Point Force Applied at Tip

- Rod Geometry and Boundary Conditions:

$$\begin{aligned} EA &= 1, & \mathfrak{B} &= \{\, x \mid 1 \leq x \leq 2 \,\}, \\ u^* &= 0, & \mathsf{S}_u &= \{\, x \mid x = 1 \,\}, \\ F &= 1, & \mathsf{S}_f &= \{\, x \mid x = 2 \,\}, \\ f^b &= 0. \end{aligned}$$

- Moving Least Squares Approximation: The weight function defined by Häussler-Combe et al. [20] is employed, with $d_{factor} = 2.0$.

Table B.1: Results: point force applied at tip, regular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.5000
2	regular	2	1.000	1	1323	0.5000
2	regular	2	1.000	2	1766	0.5000
2	regular	2	1.000	4	2656	0.5000
2	regular	2	1.000	6	3546	0.5000
3	regular	2	1.000	2	1844	0.5000
3	regular	2	1.000	4	2738	0.5000
3	regular	2	1.000	6	3908	0.5000
5	regular	2	1.000	4	3211	0.5521
		3			6990	0.5054
5	regular	2	1.000	6	4111	0.5022
		3			9994	0.5003
5	regular	2	0.500	2	2943	0.5000
		3			7452	0.5013
5	regular	2	0.500	4	5009	0.5000
		3			13000	0.5001
5	regular	2	0.500	6	7355	0.5000
		3			18090	0.5000
9	regular	2	0.500	4	6385	0.5521
		3			13816	0.5062
9	regular	2	0.500	6	8201	0.5022
		3			20310	0.5003
9	regular	2	0.250	2	5577	0.5000
		3			15230	0.5009
9	regular	2	0.250	4	10015	0.5000
		3			26812	0.5002
9	regular	2	0.250	6	14737	0.5000
		3			37934	0.5002
9	regular	2	0.125	2	11135	0.5000
		3			25430	0.5002
9	regular	2	0.125	4	20313	0.5000

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.5000
9	regular	3	0.125	6	48132	0.5001
		2			27561	0.5000
		3			73182	0.5001

Table B.2: Results: point force applied at tip, irregular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.5000
5	Pattern A	2	0.500	2	4004	-
		3			7453	0.5880
5	Pattern A	2	0.500	4	6641	1.4390
		3			13005	0.5143
5	Pattern A	2	0.500	6	9441	0.6115
		3			18557	0.5086
5	Pattern A	2	0.250	2	6653	0.5572
		3			13015	0.5098
5	Pattern A	2	0.250	4	12107	0.5044
		3			23888	0.5007
5	Pattern A	2	0.250	6	17410	0.5123
		3			34536	0.5008
5	Pattern A	2	0.125	2	12125	0.5087
		3			23910	0.5018
5	Pattern A	2	0.125	4	23018	0.5111
		3			45654	0.5008
5	Pattern A	2	0.125	6	34052	0.5112
		3			67623	0.5008
5	Pattern A	2	0.062	2	23062	0.5115
		3			45700	0.5007
5	Pattern A	2	0.062	4	44966	0.5067
		3			89865	0.5007
5	Pattern A	2	0.062	6	66868	0.5051
		3			133805	0.5007

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.5000
5	Pattern A	2	0.031	2	45056	0.5053
		3			89959	0.5007
5	Pattern A	2	0.031	4	88579	0.5053
		3			177837	0.5007
5	Pattern A	2	0.031	6	132249	0.5054
		3			265715	0.5007
5	Pattern B	2	0.500	2	2943	0.5200
		3			7452	0.5008
5	Pattern B	2	0.500	4	5299	0.5209
		3			13000	0.5019
5	Pattern B	2	0.500	6	8511	0.5121
		3			18090	0.5005
5	Pattern B	2	0.250	2	5861	0.5046
		3			12552	0.5020
5	Pattern B	2	0.250	4	10863	0.5018
		3			22738	0.5005
5	Pattern B	2	0.250	6	15015	0.5000
		3			32932	0.5010
5	Pattern B	2	0.125	2	10333	0.5000
		3			23218	0.5014
5	Pattern B	2	0.125	4	18635	0.5001
		3			44046	0.5009
5	Pattern B	2	0.125	6	27781	0.5001
		3			64866	0.5010
5	Pattern B	2	0.062	2	19231	0.5002
		3			43634	0.5011
5	Pattern B	2	0.062	4	37243	0.5007
		3			86180	0.5010
5	Pattern B	2	0.062	6	54695	0.5009
		3			128734	0.5010
7	Pattern C	2	0.500	4	6297	0.5178
		3			13538	0.5057
7	Pattern C	2	0.500	6	8391	0.5035
		3			20026	0.5011

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{C}^h$
	Exact Solution					0.5000
7	Pattern C	2	0.250	2	6047	0.5104
		3			13548	0.5034
7	Pattern C	2	0.250	4	10485	0.5026
		3			26048	0.5001
7	Pattern C	2	0.250	6	14357	0.5005
		3			37154	0.5003
7	Pattern C	2	0.125	2	10217	0.5006
		3			25610	0.5008
7	Pattern C	2	0.125	4	18813	0.5008
		3			47360	0.5004
7	Pattern C	2	0.125	6	28239	0.5005
		3			70030	0.5004

B.2.2 Point Force Applied in the Middle

- Rod Geometry and Boundary Conditions:

$$EA = 1, \qquad \mathfrak{B} = \{\, x \mid 1 \leq x \leq 2 \,\},$$
$$u^* = 0, \qquad \mathsf{S}_u = \{\, x \mid x = 1 \,\},$$
$$F_t = 0, \qquad \mathsf{S}_f = \{\, x \mid x = 2 \,\},$$
$$F_m = 1, \qquad x_{F_m} = 1.5,$$
$$f^b = 0.$$

- Moving Least Squares Approximation: The weight function defined by Häussler-Combe et al. [20] is employed, with $d_{factor} = 2.0$.

Table B.3: Results: point force applied in the middle, regular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.2500
2	regular	2	1.000	1	1324	0.1250
2	regular	2	1.000	2	1766	0.1250
2	regular	2	1.000	4	2654	0.1250
2	regular	2	1.000	6	3542	0.1250
3	regular	2	1.000	2	1907	0.2103
3	regular	2	1.000	4	2799	0.2103
3	regular	2	1.000	6	3967	0.2071
3	regular	2	0.500	2	3075	0.2075
3	regular	2	0.500	4	5139	0.2247
3	regular	2	0.500	6	6927	0.2197
5	regular	2	1.000	6	4171	0.2298
		3			9881	0.2438
5	regular	2	0.500	2	3005	0.2301
		3			7341	0.2456
5	regular	2	0.500	4	5067	0.2354
		3			12885	0.2390

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{C}^h$
	Exact Solution					0.2500
5	regular	2	0.500	6	7409	0.2375
		3			17971	0.2440
5	regular	2	0.250	2	5627	0.2287
		3			12437	0.2545
5	regular	2	0.250	4	10037	0.2374
		3			23065	0.2458
5	regular	2	0.250	6	13623	0.2349
		3			34151	0.2496
9	regular	2	0.500	6	8255	0.2443
		3			20192	0.2451
9	regular	2	0.250	2	5635	0.2401
		3			15116	0.2498
9	regular	2	0.250	4	10065	0.2427
		3			26690	0.2468
9	regular	2	0.250	6	14779	0.2437
		3			37804	0.2498
9	regular	2	0.125	2	11185	0.2394
		3			25308	0.2556
9	regular	2	0.125	4	20347	0.2437
		3			47994	0.2510
9	regular	2	0.125	6	27579	0.2424
		3			73028	0.2523
9	regular	2	0.062	2	18479	0.2438
		3			50388	0.2532
9	regular	2	0.062	4	34861	0.2425
		3			98106	0.2527
9	regular	2	0.062	6	53185	0.2426
		3			143476	0.2529

Table B.4: Results: point force applied in the middle, irregular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.2500
5	Pattern A	2	0.500	6	9627	0.3236
		3			18545	0.2984
5	Pattern A	2	0.250	2	6843	0.2529
		3			13007	0.2809
5	Pattern A	2	0.250	4	12289	0.1871
		3			23872	0.2442
5	Pattern A	2	0.250	6	17584	0.1870
		3			34512	0.2452
5	Pattern A	2	0.125	2	12307	0.1852
		3			23894	0.2386
5	Pattern A	2	0.125	4	23184	0.1871
		3			45622	0.2457
5	Pattern A	2	0.125	6	34202	0.1871
		3			67575	0.2449
5	Pattern A	2	0.062	2	23228	0.1874
		3			45668	0.2459
5	Pattern A	2	0.062	4	45100	0.1864
		3			89801	0.2464
5	Pattern A	2	0.062	6	66970	0.1867
		3			133709	0.2478
5	Pattern A	2	0.031	2	45190	0.1866
		3			89895	0.2475
5	Pattern A	2	0.031	4	88649	0.1866
		3			177709	0.2476
5	Pattern A	2	0.031	6	132255	0.1866
		3			265523	0.2476
5	Pattern B	2	0.500	6	8697	0.2369
		3			18183	0.2460
5	Pattern B	2	0.250	2	6051	0.2271
		3			12649	0.2565
5	Pattern B	2	0.250	4	11045	0.2326
		3			22827	0.2442
5	Pattern B	2	0.250	6	15189	0.2319
		3			33013	0.2506

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.2500
5	Pattern B	2	0.125	2	10515	0.2346
		3			23307	0.2537
5	Pattern B	2	0.125	4	18801	0.2317
		3			44119	0.2502
5	Pattern B	2	0.125	6	27931	0.2316
		3			64923	0.2510
5	Pattern B	2	0.062	2	19397	0.2314
		3			43707	0.2511
5	Pattern B	2	0.062	4	37377	0.2317
		3			86221	0.2509
5	Pattern B	2	0.062	6	54797	0.2318
		3			128743	0.2507
5	Pattern B	2	0.031	2	36913	0.2318
		3			86773	0.2507
5	Pattern B	2	0.031	4	72875	0.2317
		3			171357	0.2507
5	Pattern B	2	0.031	6	109685	0.2317
		3			255483	0.2508
7	Pattern C	2	0.500	6	8379	0.2444
		3			19802	0.2534
7	Pattern C	2	0.250	2	6039	0.2410
		3			13328	0.2577
7	Pattern C	2	0.250	4	10469	0.2437
		3			25820	0.2548
7	Pattern C	2	0.250	6	14333	0.2416
		3			36918	0.2503
7	Pattern C	2	0.125	2	10201	0.2394
		3			25382	0.2526
7	Pattern C	2	0.125	4	18781	0.2418
		3			47116	0.2534
7	Pattern C	2	0.125	6	28191	0.2415
		3			69770	0.2539
7	Pattern C	2	0.062	2	19111	0.2428
		3			47622	0.2544

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.2500
7	Pattern C	2	0.062	4	37111	0.2414
		3			92482	0.2533
7	Pattern C	2	0.062	6	54267	0.2413
		3			136408	0.2535
5	Pattern R	2	0.500	2	4552	0.9476
		3			7337	0.6631
5	Pattern R	2	0.500	4	6914	0.2394
		3			11973	0.2380
5	Pattern R	2	0.500	6	9244	0.2456
		3			16151	0.2202
5	Pattern R	2	0.250	2	6626	0.2396
		3			11525	0.2226
5	Pattern R	2	0.250	4	12174	0.2448
		3			21245	0.2205
5	Pattern R	2	0.250	6	17752	0.2463
		3			31423	0.2204
5	Pattern R	2	0.125	2	12494	0.2452
		3			21725	0.2203
5	Pattern R	2	0.125	4	24198	0.2435
		3			42521	0.2226
5	Pattern R	2	0.125	6	35876	0.2420
		3			62859	0.2250
5	Pattern R	2	0.062	2	23946	0.2424
		3			42109	0.2248
5	Pattern R	2	0.062	4	46468	0.2421
		3			82333	0.2244
5	Pattern R	2	0.062	6	69288	0.2424
		3			123015	0.2244
9	Pattern S	2	0.500	4	9003	5.5489
		3			17549	0.2969
9	Pattern S	2	0.500	6	10537	0.2570
		3			24527	0.2626
9	Pattern S	2	0.250	2	9021	-
		3			17563	0.4110

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.2500
9	Pattern S	2	0.250	4	15035	0.2572
		3			32461	0.2479
9	Pattern S	2	0.250	6	21545	0.2532
		3			46903	0.2532
9	Pattern S	2	0.125	2	15053	0.2577
		3			32483	0.2413
9	Pattern S	2	0.125	4	31813	0.2438
		3			63753	0.2449
9	Pattern S	2	0.125	6	48021	0.2429
		3			94985	0.2463
9	Pattern S	2	0.062	2	31849	0.2418
		3			63301	0.2477
9	Pattern S	2	0.062	4	61541	0.2431
		3			123903	0.2444
9	Pattern S	2	0.062	6	91443	0.2425
		3			184007	0.2442
9	Pattern S	2	0.031	2	61979	0.2429
		3			124503	0.2443
9	Pattern S	2	0.031	4	117579	0.2437
		3			242765	0.2435
9	Pattern S	2	0.031	6	173311	0.2449
		3			362031	0.2438

B.3 Body Force

B.3.1 Constant Shape

- Rod Geometry and Boundary Conditions:

$$EA = 1, \qquad \mathfrak{B} = \{\, x \mid 1 \leq x \leq 2 \,\},$$
$$u^* = 0, \qquad \mathsf{S}_u = \{\, x \mid x = 1 \,\},$$
$$F = 0, \qquad \mathsf{S}_f = \{\, x \mid x = 2 \,\},$$
$$f^b = 1.$$

- Moving Least Squares Approximation: The weight function defined by Häussler-Combe et al. [20] is employed, with $d_{factor} = 2.0$.

Table B.5: Results: constant body force, regular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					$0.166\bar{6}$
2	regular	2	1.000	2	2006	0.1250
2	regular	2	1.000	4	3326	0.1250
2	regular	2	1.000	6	4646	0.1250
3	regular	2	1.000	2	2093	0.1473
3	regular	2	1.000	4	3421	0.1536
3	regular	2	1.000	6	5161	0.1540
3	regular	2	0.500	2	3833	0.1553
3	regular	2	0.500	4	6909	0.1597
3	regular	2	0.500	6	9573	0.1582
3	regular	2	0.250	2	6529	0.1604
3	regular	2	0.250	4	12261	0.1583
3	regular	2	0.250	6	18405	0.1584
5	regular	2	0.500	2	3641	0.1618
		3			9521	0.1671
5	regular	2	0.500	4	6713	0.1638
		3			17705	0.1669

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{e}^h$
	Exact Solution					$0.166\bar{6}$
5	regular	2	0.500	6	10205	0.1640
		3			25217	0.1669
5	regular	2	0.250	2	7553	0.1637
		3			17053	0.1668
5	regular	2	0.250	4	14129	0.1652
		3			32745	0.1668
5	regular	2	0.250	6	19477	0.1647
		3			49109	0.1668
5	regular	2	0.250	6	19477	0.1647
		3			49109	0.1668
5	regular	2	0.125	4	24869	0.1648
		3			65513	0.1668
5	regular	2	0.125	6	38029	0.1648
		3			96893	0.1668
9	regular	2	0.250	2	7199	0.1655
		3			20622	0.1671
9	regular	2	0.250	4	13815	0.1660
		3			37718	0.1668
9	regular	2	0.250	6	20859	0.1661
		3			54138	0.1668
9	regular	2	0.125	2	15495	0.1659
		3			35710	0.1667
9	regular	2	0.125	4	29191	0.1663
		3			69222	0.1667
9	regular	2	0.125	6	40007	0.1662
		3			106162	0.1667

Table B.6: Results: constant body force, irregular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					$0.166\bar{6}$
5	Pattern A	2	0.500	2	5172	-
		3			9413	0.1917
5	Pattern A	2	0.500	4	9082	0.6473
		3			17599	0.1725
5	Pattern A	2	0.500	6	13236	0.2102
		3			25785	0.1704
5	Pattern A	2	0.250	2	9106	0.1839
		3			17619	0.1702
5	Pattern A	2	0.250	4	17192	0.1606
		3			33651	0.1669
5	Pattern A	2	0.250	6	25058	0.1628
		3			49349	0.1668
5	Pattern A	2	0.125	2	17228	0.1606
		3			33695	0.1673
5	Pattern A	2	0.125	4	33382	0.1627
		3			65749	0.1668
5	Pattern A	2	0.125	6	49736	0.1628
		3			98137	0.1668
5	Pattern B	2	0.500	2	3641	0.1628
		3			9521	0.1696
5	Pattern B	2	0.500	4	7141	0.1653
		3			17705	0.1674
5	Pattern B	2	0.500	6	11909	0.1637
		3			25217	0.1668
5	Pattern B	2	0.250	2	7985	0.1634
		3			17053	0.1672
5	Pattern B	2	0.250	4	15413	0.1629
		3			32077	0.1668
5	Pattern B	2	0.250	6	21585	0.1640
		3			47105	0.1669
5	Pattern B	2	0.125	2	14633	0.1643
		3			32793	0.1671
5	Pattern B	2	0.125	4	26985	0.1637
		3			63509	0.1669

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{C}^h$
	Exact Solution					$0.166\bar{6}$
5	Pattern B	2	0.125	6	40581	0.1636
		3			94221	0.1670
7	Pattern C	2	0.500	2	4443	-
		3			9278	-
7	Pattern C	2	0.500	4	8355	0.1665
		3			18150	0.1684
7	Pattern C	2	0.500	6	11471	0.1655
		3			27714	0.1671
7	Pattern C	2	0.250	2	7983	0.1688
		3			18170	0.1684
7	Pattern C	2	0.250	4	14599	0.1658
		3			36602	0.1668
7	Pattern C	2	0.250	6	20375	0.1666
		3			52990	0.1669
7	Pattern C	2	0.125	2	14215	0.1669
		3			35970	0.1671
7	Pattern C	2	0.125	4	27027	0.1670
		3			68066	0.1669
7	Pattern C	2	0.125	6	41067	0.1666
		3			101514	0.1669

B.3.2 Quadratic Shape

- Rod Geometry and Boundary Conditions:

$$EA = 1, \qquad \mathfrak{B} = \{\, x \mid 1 \le x \le 2 \,\},$$
$$u^* = 0, \qquad \mathsf{S}_u = \{\, x \mid x = 1 \,\},$$
$$F = 0, \qquad \mathsf{S}_f = \{\, x \mid x = 2 \,\},$$
$$f^b = 4\Big((x-1)(2-x)\Big).$$

- Moving Least Squares Approximation: The weight function defined by Häussler-Combe et al. [20] is employed, with $d_{factor} = 2.0$.

Table B.7: Results: quadratic body force, regular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.0825
2	regular	2	1.000	2	2018	0.0556
2	regular	2	1.000	4	3350	0.0556
2	regular	2	1.000	6	4682	0.0556
3	regular	2	1.000	2	2105	0.0655
3	regular	2	1.000	4	3445	0.0749
3	regular	2	1.000	6	5197	0.0755
3	regular	2	0.500	2	3857	0.0766
3	regular	2	0.500	4	6957	0.0794
3	regular	2	0.500	6	9645	0.0784
3	regular	2	0.250	2	6577	0.0800
3	regular	2	0.250	4	12357	0.0785
3	regular	2	0.250	6	18549	0.0785
5	regular	2	0.500	2	3665	0.0797
		3			9545	0.0825
5	regular	2	0.500	4	6761	0.0812
		3			17753	0.0822
5	regular	2	0.500	6	10277	0.0813
		3			25289	0.0822

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{C}^h$
	Exact Solution					0.0825
5	regular	2	0.250	2	7601	0.0809
		3			17101	0.0823
5	regular	2	0.250	4	14225	0.0819
		3			32841	0.0823
5	regular	2	0.250	6	19621	0.0816
		3			49253	0.0823
5	regular	2	0.125	2	13053	0.0820
		3			33557	0.0824
5	regular	2	0.125	4	25061	0.0816
		3			65705	0.0823
5	regular	2	0.125	6	38317	0.0816
		3			97181	0.0823
9	regular	2	0.500	2	5282	-
		3			10362	-
9	regular	2	0.500	4	8443	0.0908
		3			18598	0.0836
9	regular	2	0.500	6	11179	0.0824
		3			28198	0.0825
9	regular	2	0.250	2	7247	0.0819
		3			20670	0.0826
9	regular	2	0.250	4	13911	0.0822
		3			37814	0.0826
9	regular	2	0.250	6	21003	0.0822
		3			54282	0.0826
9	regular	2	0.125	2	15591	0.0821
		3			35806	0.0827
9	regular	2	0.125	4	29383	0.0824
		3			69414	0.0826
9	regular	2	0.125	6	40295	0.0823
		3			106450	0.0826

Table B.8: Results: quadratic body force, irregular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{E}^h$
	Exact Solution					0.0825
5	Pattern A	2	0.500	2	5196	-
		3			9437	0.1015
5	Pattern A	2	0.500	4	9130	0.3530
		3			17647	0.0870
5	Pattern A	2	0.500	6	13308	0.1063
		3			25857	0.0856
5	Pattern A	2	0.250	2	9154	0.0906
		3			17667	0.0851
5	Pattern A	2	0.250	4	17288	0.0767
		3			33747	0.0823
5	Pattern A	2	0.250	6	25202	0.0776
		3			49493	0.0823
5	Pattern A	2	0.125	2	17324	0.0766
		3			33791	0.0823
5	Pattern A	2	0.125	4	33574	0.0776
		3			65941	0.0823
5	Pattern A	2	0.125	6	50024	0.0776
		3			98425	0.0823
5	Pattern B	2	0.500	2	3665	0.0797
		3			9545	0.0862
5	Pattern B	2	0.500	4	7189	0.0815
		3			17753	0.0824
5	Pattern B	2	0.500	6	11981	0.0805
		3			25289	0.0824
5	Pattern B	2	0.250	2	8033	0.0803
		3			17101	0.0827
5	Pattern B	2	0.250	4	15509	0.0803
		3			32173	0.0825
5	Pattern B	2	0.250	6	21729	0.0809
		3			47249	0.0827
5	Pattern B	2	0.125	2	14729	0.0812
		3			32889	0.0829
5	Pattern B	2	0.125	4	27177	0.0808
		3			63701	0.0827

continued from previous page

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Floating Point Operations	Strain Energy $\mathfrak{C}^h$
	Exact Solution					0.0825
5	Pattern B	2	0.125	6	40869	0.0807
		3			94509	0.0827
7	Pattern C	2	0.500	2	4467	-
		3			9302	-
7	Pattern C	2	0.500	4	8403	0.0825
		3			18198	0.0842
7	Pattern C	2	0.500	6	11543	0.0817
		3			27786	0.0826
7	Pattern C	2	0.250	2	8031	0.0831
		3			18218	0.0832
7	Pattern C	2	0.250	4	14695	0.0821
		3			36698	0.0825
7	Pattern C	2	0.250	6	20519	0.0826
		3			53134	0.0825
7	Pattern C	2	0.125	2	14311	0.0829
		3			36066	0.0825
7	Pattern C	2	0.125	4	27219	0.0828
		3			68258	0.0825
7	Pattern C	2	0.125	6	41355	0.0826
		3			101802	0.0825
7	Pattern C	2	0.062	2	27727	0.0826
		3			69026	0.0825
7	Pattern C	2	0.062	4	54779	0.0825
		3			135450	0.0825
7	Pattern C	2	0.062	6	80575	0.0824
		3			200506	0.0825

B.4 Comparison of Weight Functions

Table B.9: Results obtained with different weight functions, point force applied at tip, regular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Weight Function	Strain Energy $\mathfrak{E}^h$
		Exact Solution				0.5000
2	regular	2	1.000	6	L2	0.5010
2	regular	2	1.000	6	L4	0.5010
2	regular	2	1.000	6	L6	-
2	regular	2	1.000	6	Ta	0.5000
2	regular	2	1.000	6	HC	0.5000
3	regular	2	1.000	6	L2	0.5010
3	regular	2	1.000	6	L4	0.5010
3	regular	2	1.000	6	L6	-
3	regular	2	1.000	6	Ta	0.5000
3	regular	2	1.000	6	HC	0.5000
5	regular	2	0.500	6	L2	0.5010
5	regular	2	0.500	6	L4	0.5010
5	regular	2	0.500	6	L6	0.5326
5	regular	2	0.500	6	Ta	0.5000
5	regular	2	0.500	6	HC	0.5000
9	regular	2	0.125	4	L2	0.5010
9	regular	2	0.125	4	L4	0.5010
9	regular	2	0.125	4	L6	0.5010
9	regular	2	0.125	4	Ta	0.5000
9	regular	2	0.125	4	HC	0.5000

Table B.10: Results obtained with different weight functions, point force applied in the middle, regular and irregular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Weight Function	Strain Energy $\mathfrak{E}^h$
		Exact Solution				0.2500
5	regular	2	0.125	6	Ta	0.2256
		3				0.2598
5	regular	2	0.125	6	HC	0.2353
		3				0.2508
9	regular	2	0.062	6	Ta	0.2378
		3				0.2624
9	regular	2	0.062	6	HC	0.2426
		3				0.2529
5	Pattern A	2	0.062	6	Ta	0.1888
		3				0.2413
5	Pattern A	2	0.062	6	HC	0.1867
		3				0.2478
5	Pattern B	2	0.062	6	Ta	0.2174
		3				0.2495
5	Pattern B	2	0.062	6	HC	0.2318
		3				0.2507
7	Pattern C	2	0.062	6	Ta	0.2357
		3				0.2658
7	Pattern C	2	0.062	6	HC	0.2413
		3				0.2535

Table B.11: Results obtained with different weight functions, quadratic body force, regular and irregular node distribution

Number of Nodes	Nodal Distribution	Basis m	Integration Cell Length	Integration Order	Weight Function	Strain Energy $\mathfrak{E}^h$
		Exact Solution				0.0825
5	regular	2	0.125	6	Ta	0.0816
		3				0.0822
5	regular	2	0.125	6	HC	0.0816
		3				0.0823
9	regular	2	0.125	6	Ta	0.0823
		3				0.0827
9	regular	2	0.125	6	HC	0.0823
		3				0.0826
5	Pattern A	2	0.125	6	Ta	0.0801
		3				0.0828
5	Pattern A	2	0.125	6	HC	0.0776
		3				0.0823
5	Pattern B	2	0.125	6	Ta	0.0802
		3				0.0824
5	Pattern B	2	0.125	6	HC	0.0807
		3				0.0827
7	Pattern C	2	0.062	6	Ta	0.0827
		3				0.0825
7	Pattern C	2	0.062	6	HC	0.0824
		3				0.0825

B.5 Weight Functions

Some typical weight functions, employed in the Moving Least Squares Method, are shown in Figures B-1, B-2, B-3 and B-4. The domain of influence is set to $d_{mi} = 1.0$. Note the difference between the singular and smooth weight functions. Smooth weight functions, as applied in the EFGM, yield noninterpolating Moving Least Squares Interpolants, while singular weight functions give interpolating Moving Least Squares Interpolants. The following weight functions are implemented:

- Lancaster et al. [30], `mlswtype` $= 2 \ldots 10$:

$$w^{(L\alpha)}(d_i) = |x - x_i|^{-\alpha}, \quad \alpha \in \{2, 4, 6, 8, 10\},$$

- Tabbara et al. [41], `mlswtype` $= 11$:

$$w^{(Ta)}(d_i) = e^{-\left(\frac{2.5\, d_i}{d_{mi}}\right)^2},$$

- Häussler-Combe et al.[1] [20], `mlswtype` $= 14$:

$$w^{(HC)}(d_i) = \frac{e^{-\left(\frac{d_i}{H_c\, d_{mi}}\right)^2} - e^{-\frac{1}{H_c^2}}}{1 - e^{-\frac{1}{H_c^2}}}.$$

[1]Note that this weight function is a special case of the exponential weight function WFA defined by Belytschko et al. [7]. They do not recommend any value for H_c, while Häussler-Combe et al. recommend $H_c = 1/3$.

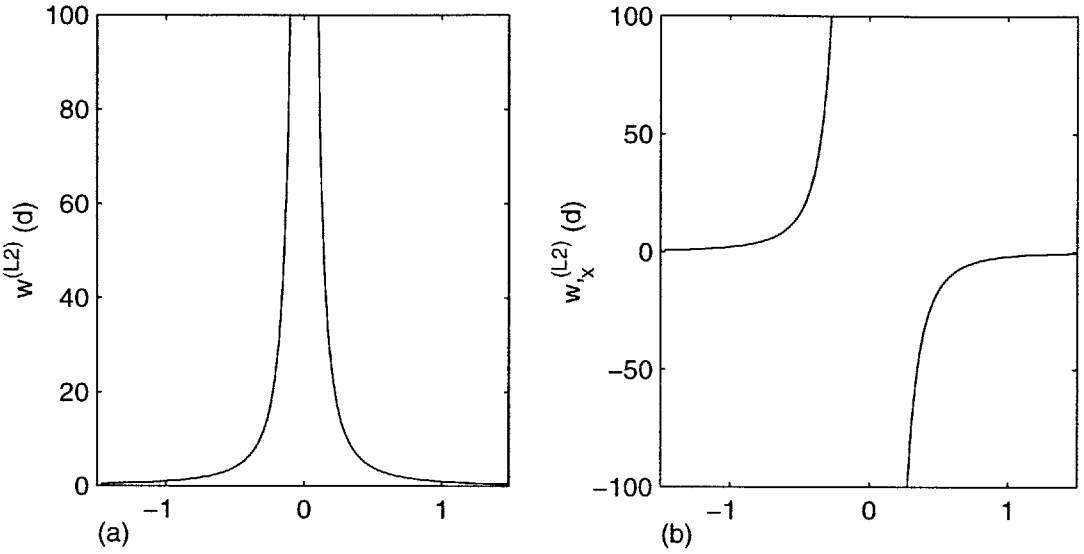

Figure B-1: (a) Singular second order weight function, (b) derivative [30]

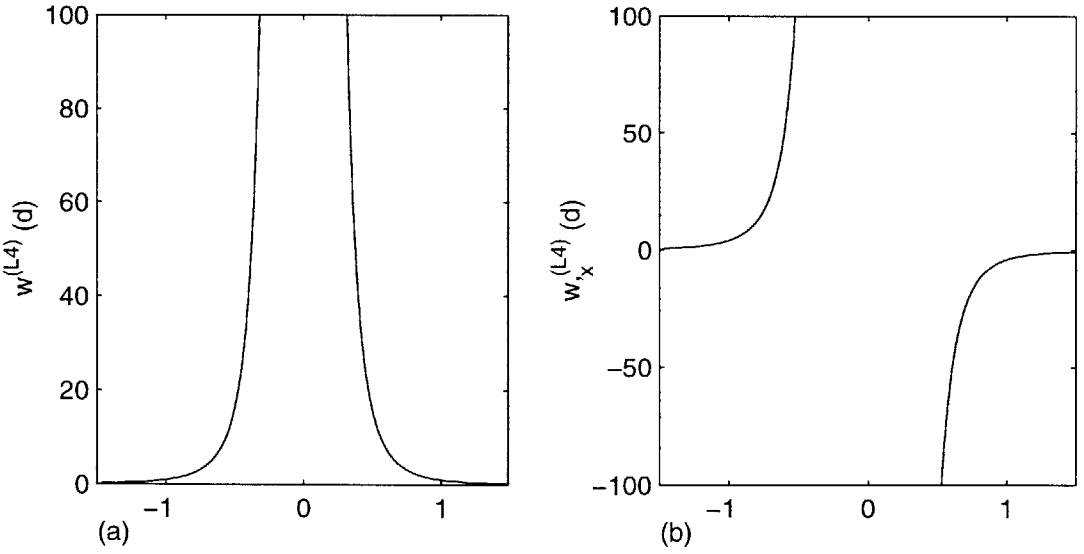

Figure B-2: (a) Singular fourth order weight function, (b) derivative [30]

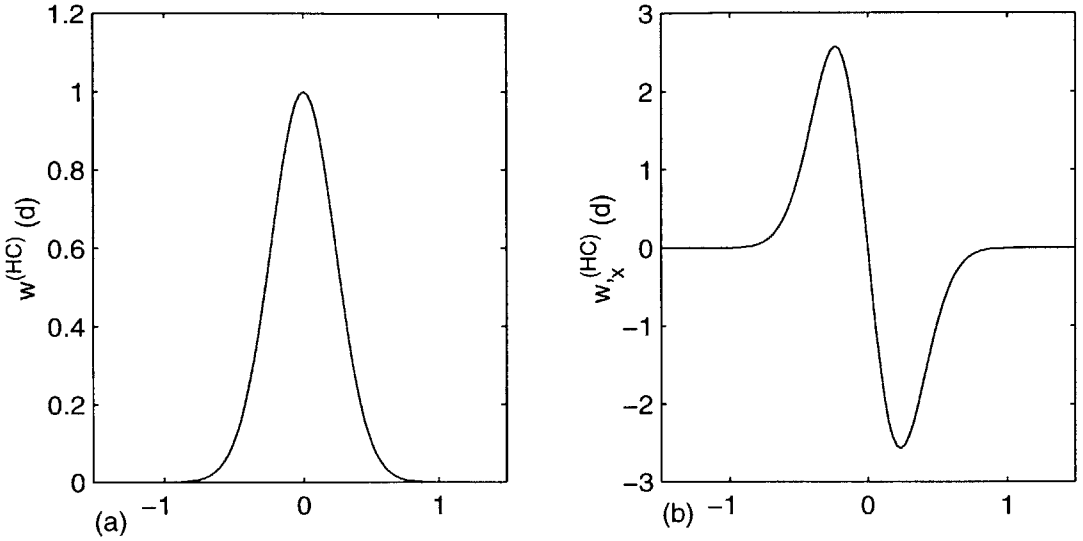

Figure B-3: (a) Smooth weight function, (b) derivative [20]

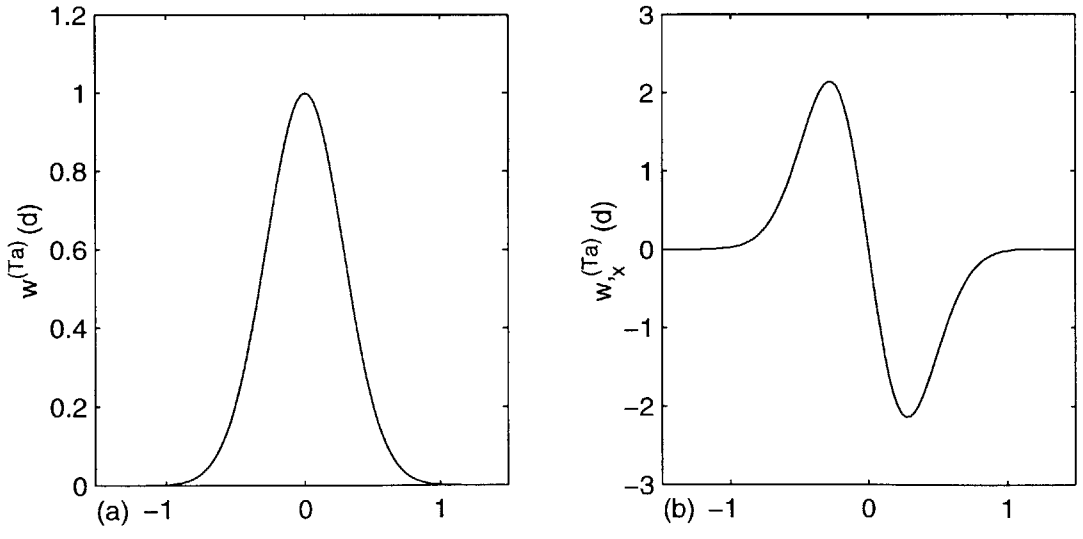

Figure B-4: (a) Smooth weight function, (b) derivative [41]

Bibliography

[1] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* United States Department of Commerce, National Bureau of Standards, 1964.

[2] Rainer Ansorge and Hans Joachim Oberle. *Mathematik für Ingenieure*, volume 1 and 2. Akademie Verlag, Berlin, Germany, 1994.

[3] Klaus-Jürgen Bathe. *Finite Element Procedures.* Prentice-Hall, 1996.

[4] S. Beissel and T. Belytschko. Nodal integration of the element-free galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139:49–74, 1996.

[5] T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, and W. K. Liu. Smoothing and accelerated computations in the element free galerkin method. *Journal of Computational and Applied Mathematics*, 74:111–126, 1996.

[6] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

[7] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[8] T. Belytschko and D. Organ. Element-free galerkin methods for dynamic fracture in concrete. In D. R. J. Owen, E. Oñate, and E. Hinton, editors, *Computational Plasticity:*

Fundamentals and Applications, Part 1. Proceedings of the Fifth International Conference on Computational Plasticity held in Barcelona, Spain, pages 304–321. CIMNE, March 1997.

[9] T. Belytschko, D. Organ, and Y. Krongauz. A coupled finite element-element free galerkin method. *Computational Mechanics*, 17:186–195, 1995.

[10] T. Belytschko and M. Tabbara. Dynamic fracture using element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 39:923–938, 1996.

[11] Arthur P. Boresi and Ken P. Chong. *Approximate Solution Methods in Engineering Mechanics.* Elsevier Science Publishing Co., Inc., 1991.

[12] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik.* Harri Deutsch, fourth edition, 1999.

[13] Lothar Collatz. *The Numerical Treatment of Differential Equations.* Springer-Verlag, third edition, 1960.

[14] L. W. Cordes and B. Moran. Treatment of material discontinuity in the element-free galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139:75–89, 1996.

[15] Stephen H. Crandall. *Engineering Analysis, A Survey of Numerical Procedures.* Robert E. Krieger Publishing Company, 1956.

[16] J. Dolbow and T. Belytschko. Numerical integration of the galerkin weak form in meshfree methods. *Computational Mechanics*, 23:219–230, 1999.

[17] John E. Dolbow. Numerical integration in meshfree methods. Master's thesis, Northwestern University, 1998.

[18] B. A. Finlayson and L. E. Scriven. The method of weighted residuals - a review. *Applied Mechanics Reviews*, 19(9):735–748, 1966.

[19] Morton E. Gurtin. *An Introduction to Continuum Mechanics.* Academic Press, 1981.

[20] U. Häussler-Combe and C. Korn. An adaptive approach with the element free galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 162:203–222, 1998.

[21] Francis B. Hildebrand. *Methods of Applied Mathematics.* Dover Publications, second edition, 1965.

[22] Thomas J. R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis.* Prentice-Hall, 1987.

[23] Milan Jirásek. Element-free galerkin method applied to strain-softening materials. In R. De Borst, Bicanic N., Mang H., and Meschke G., editors, *Computational Modeling of Concrete Structures: Proceedings of the EURO-C Conference 1998, Badgastein*, 1998.

[24] Eduardo Kausel. Lecture notes in structural dynamics, 1998.

[25] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133–145, 1996.

[26] Y. Krongauz and T. Belytschko. Consistent pseudo-derivatives in meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 146:371–386, 1997.

[27] P. Krysl and T. Belytschko. Analysis of thin plates by the element-free galerkin method. *Computational Mechanics*, 17:26–35, 1995.

[28] P. Krysl and T. Belytschko. Analysis of thin shells by the element-free galerkin method. *International Journal of Solids and Structures*, 33(20-22):3057–3080, 1996.

[29] P. Krysl and T. Belytschko. Element-free galerkin method: Convergence of the continuous and discontinuous shape functions. *Computer Methods in Applied Mechanics and Engineering*, 148:257–277, 1997.

[30] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155), 1981.

[31] Y. Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element free galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 113:397–414, 1994.

[32] Y. Y. Lu, T. Belytschko, and M. Tabbara. Element free galerkin method for wave propagation and dynamic fracture. *Computer Methods in Applied Mechanics and Engineering*, 126:131–153, 1995.

[33] Lawrence E. Malvern. *Introduction to the Mechanics of a Continuous Medium.* Prentice-Hall, 1969.

[34] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.

[35] D. Organ, M. Fleming, T. Terry, and T. Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics*, 18:1–11, 1996.

[36] Daniel J. Organ. *Numerical Solutions to Dynamic Fracture Problems Using the Element-Free Galerkin Method.* PhD thesis, Northwestern University, 1996.

[37] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[38] N. Sukumar. Application of the coupled FE-EFG method to material discontinuities in 1D and 2D. `http://www.tam.nwu.edu/suku/meshless/meshless.html`, May 1996.

[39] N. Sukumar. Patch test in 1D: Coupled finite element - element free galerkin method. `http://www.tam.nwu.edu/suku/meshless/meshless.html`, October 1996.

[40] N. Sukumar, B. Moran, T. Black, and T. Belytschko. An element-free galerkin method for three-dimensional fracture mechanics. *Computational Mechanics*, 20:170–175, 1997.

[41] M. R. Tabbara and C. M. Stone. A computational method for quasi-static fracture. *Computational Mechanics*, 22:203–210, 1998.

[42] Kyuichiro Washizu. *Variational Methods in Elasticity and Plasticity.* Pergamon Press, third edition, 1982.

[43] O. C. Zienkiewicz and R. L. Taylor. *Finite Element Method*, volume 1: Basic Formulation and Linear Problems. McGraw-Hill International, fourth edition, 1994.

Index

W

Y