

EDGE FLUX INTENSITY FUNCTIONS IN THREE  
DIMENSIONAL DOMAINS

By  
Netta Omer

SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE  
AT  
BEN-GURION UNIVERSITY OF THE NEGEV  
BEER-SHEVA, ISRAEL  
OCTOBER 2003

© Copyright by Netta Omer, 2003

BEN-GURION UNIVERSITY OF THE NEGEV  
DEPARTMENT OF  
MECHANICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Engineering for acceptance a thesis entitled “**Edge Flux Intensity Functions in Three Dimensional Domains**” by **Netta Omer** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: October 2003

Supervisor:

\_\_\_\_\_  
Prof. Zohar Yosibash

Readers:

\_\_\_\_\_  
???

\_\_\_\_\_  
???

## ABSTRACT

The solution to scalar elliptic boundary value problems in three-dimensional polyhedra domains in the vicinity of an edge is provided in an explicit form. It involves an asymptotic expansion of eigen-functions and their shadows, and the associated coefficients (which are functions along the edges) called edge flux intensity functions (EFIFs). Using the dual eigen-functions and special constructed extraction polynomials we present a new extraction method for the computation of the EFIFs. The method called *quasidual function method* can be interpreted as an extension of the dual singular function contour integral method in 2-D domains, and involves the computation of a surface integral  $J[R]$  along a cylindrical surface of radius  $R$  away from the edge as presented in a general framework in [10].

This accurate and efficient method provides the *polynomial representation of the EFIF along the edge* and is implemented as a post-solution operation in conjunction with the p-version finite element method. Numerical realization of some of the anticipated properties of the  $J[R]$  are provided, and it is used for extracting EFIFs associated with different scalar elliptic problems in 3-D domains. The numerical examples demonstrate the efficiency, robustness and high accuracy of the proposed quasi-dual method.

## ACKNOWLEDGEMENTS

I wish to express my deepest appreciation to Professor Zohar Yosibash, my supervisor, for his guidance, assistance and encouragement.

Furthermore, I would like to thank Professors Monique Dauge and Martin Costabel of the University of Rennes 1, France, for their valuable suggestions and helpful discussions.

And finally, I would like to thank to Mr. Ilan Gilad for his support in the administrative aspects of this work.

# CONTENTS

<i>Abstract</i> . . . . .	IV
<i>Acknowledgements</i> . . . . .	V
<i>List of Tables</i> . . . . .	IX
<i>List of Figures</i> . . . . .	XI
<i>List of Symbols</i> . . . . .	XIV
<b>1. Introduction</b> . . . . .	1
1.1 Two dimensional domains with reentrant corners . . . . .	2
1.1.1 Computation of the eigen-pairs . . . . .	3
1.1.2 The path independent integral . . . . .	7
1.1.3 Extracting the GFIF . . . . .	11
1.2 Finite Element Method . . . . .	12
1.2.1 The variational formulation . . . . .	14
1.2.2 Structure Discretization . . . . .	15
1.2.3 The h and p versions of the FEM . . . . .	16
1.3 Three dimensional domain with an edge singularity . . . . .	17
1.3.1 Extracting the eigen-pairs . . . . .	19
1.3.2 Extracting the EFIF . . . . .	20

2.	<i>The Scalar Elliptic Problems of Interest &amp; Singular Solution Near Edges</i>	27
2.1	Notations & Problems of Interest	27
2.2	The Eigen-Pairs and computation of their Shadow Functions	29
2.2.1	Eigen-functions, their shadow functions and duals for cases 1-4	32
2.2.2	Eigen-functions, their shadow functions and duals for case 5	42
3.	<i>The Quasi-Dual Function Method for Extracting EFIFs, and the <math>J[R]</math> Integral</i>	46
3.1	The quasi-dual extraction functions.	48
3.2	Analytical Solutions for Validating the $J[R]$ Theorem	50
3.2.1	Discretization of the model by p-FEM	51
3.2.2	Numerical Tests Using $K_0^{(\alpha_1)}$ for Case 1 – 5	51
3.2.3	Numerical Tests Using $K_0^{(\alpha_1)} - K_3^{(\alpha_1)}$ for Case 5	53
3.2.4	Numerical Results Using $K_2^{(\alpha_1)}$ for Case 1 – 4	54
3.3	Numerical errors due to the numerical integration and finite element solution.	58
3.3.1	Errors due to finite element approximation	58
3.3.2	Errors due to the Gauss quadrature computation	58
4.	<i>Computing EFIFs Using a "Simple" Family of Extraction Polynomials</i>	61
4.1	Constructing a "simple" family of extraction polynomials $B_2^{(k)}(x_3)$	62
4.2	Numerical results - using $BC_4$ with $K_{0k}^{(\alpha_1)}$ - Influence of $R$	65
4.3	Numerical results - using $BC_4$ with $K_{2k}^{(\alpha_1)}$ - Influence of $R$	66
4.4	Numerical results - using $BC_4$ with $K_{2k}^{(\alpha_1)}$ for cases 1-4	67
5.	<i>Computing EFIFs Using a Hierarchical Family of Extraction Polynomials</i>	72
5.1	Constructing a hierarchical family of extraction polynomials $BJ_m^{(k)}(x_3)$	72
5.2	Numerical results - using $BC_4$ with $K_{2k}^{(\alpha_1)}$	74

6. <i>A Polynomial Representation of a Non-Polynomial EFIF</i> . . . . .	79
6.1 Computing EFIFs Using an Hierarchical Family of Extraction Polynomials . . . . .	83
6.2 Computing EFIFs Using a "Simple" Family of Extraction Polynomials	90
7. <i>Summary, Conclusions and Suggestions for Future Research</i> . . . . .	94
7.1 Summary and Conclusions . . . . .	94
7.2 Suggestions for Future Research . . . . .	96
<i>Appendix</i>	97
A. <i>Computer program</i> . . . . .	98
<i>Bibliography</i> . . . . .	103

## LIST OF TABLES

1.1	The physical significance of vector $\mathbf{u}$ and vector $\mathbf{f}$ in FEM applications	14
2.1	Notation of the various Cases considered as model problems . . . . .	29
2.2	Main coefficients of the considered cases . . . . .	39
3.1	Values of $J[R]/J_{ex}$ , $BC_2$ , using $K_0^{(\alpha_1)}$ . $B_0(x_3) = 1$ , and $J_{ex} = 8/3$ .	53
3.2	Values of $J[R]/J_{ex}$ , for case 5 with $BC_2$ . . . . .	55
3.3	Values of $J[R]/J_{ex}$ , $BC_2$ , using $K_2^{(\alpha_1)}$ . . . . .	56
3.4	Values of $J[R]/J_{ex}$ , using case 5 with $BC_2$ and $R = 0.02$ . . . . .	59
3.5	Values of $J[R]/J_{ex}$ , using case 5 with $BC_2$ and $R = 0.02$ . . . . .	59
4.1	Values of extracted $a_i^{(\alpha_1)}$ , $BC_4$ , using $K_{0k}^{(\alpha_1)}$ at different $R$ s . . . . .	65
4.2	Richardson's extrapolation for $a_0^{(\alpha_1)}$ and $a_1^{(\alpha_1)}$ with $\mathcal{O}(R^2)$ . . . . .	67
4.3	Values of extracted $a_i^{(\alpha_1)}$ , $BC_4$ , using $K_{2k}^{(\alpha_1)}$ at different $R$ s . . . . .	68
4.4	Richardson's extrapolation for $a_0^{(\alpha_1)}$ and $a_1^{(\alpha_1)}$ with $\mathcal{O}(R^4)$ . . . . .	69
4.5	Values of extracted $a_i^{(\alpha_1)}$ , $BC_4$ , using $K_{2k}^{(\alpha_1)}$ at $R = 0.05$ . . . . .	69
5.1	Values of extracted $a_i^{(\alpha_1)}$ , $BC_4$ , using $K_2^{(\alpha_1)}$ and $BJ_2^{(m)}(x_3)$ , $m \leq 4$ at different $R$ s. . . . .	76
5.2	Values of extrapolated $a_i^{(\alpha_1)}$ , $BC_4$ , using $K_2^{(\alpha_1)}$ and $BJ_2^{(m)}(x_3)$ , $m \leq 4$ , based on $R = 0.9, 0.5$ of Table 5.1. . . . .	77
6.1	Values of extracted $a_i^{(\alpha_1)}$ for the case with $d = 1.05$ , using $K_{2k}^{(\alpha_1)}$ and $BJ_2^{(k)}(x_3)$ , $k \leq 19$ at different $R$ s, and extrapolated. . . . .	88

6.2 Values of extracted  $a_i^{(\alpha_1)}$ , for  $d = 1.05$  using the “simple” extraction polynomials,  $K_{2k}^{(\alpha_1)}$  at different  $R$ s . . . . . 90

## LIST OF FIGURES

1.1	Schematic 2-D domain, $G$ . . . . .	3
1.2	A variable mapping transforming the domain $G$ , into the domain $G'$ . . . . .	5
1.3	The sub-domain $\Omega$ with the boundaries $\Gamma_i$ . . . . .	8
1.4	Considered paths in the domain $G$ . . . . .	9
1.5	Diagram of the Finite Element Method Process. . . . .	13
1.6	The concept of variation of the functional $\mathbf{u}(\mathbf{x})$ . . . . .	14
1.7	FEM discretization of a structure . . . . .	15
1.8	Schematic 3-D domain, $\Omega$ . . . . .	18
1.9	Edge and vertex singularities in a 3-D domain. . . . .	19
1.10	Three surface integrals in the 3-D domain. . . . .	20
2.1	Schematic domains of interest $\Omega$ . . . . .	28
2.2	The plane domain $G$ before and after change of variables. . . . .	33
2.3	The eigen-functions and dual eigen-functions associated with the first eigen-value, $\alpha_1$ , for case 1. . . . .	39
2.4	The eigen-functions and dual eigen-functions associated with the first eigen-value, $\alpha_1$ , for case 2. . . . .	40
2.5	The eigen-functions and dual eigen-functions associated with the first eigen-value, $\alpha_1$ , for case 3. . . . .	40
2.6	The eigen-functions and dual eigen-functions associated with the first eigen-value, $\alpha_1$ , for case 4. . . . .	41

2.7	The eigen-functions and dual eigen-functions associated with the first eigen-value, $\alpha_2$ , for case 5. . . . .	45
3.1	Cylindrical surface, $\Gamma_R$ . . . . .	47
3.2	The p-FEM models. . . . .	52
3.3	Converge rates of $J[R]$ , using $BC_2$ , $K_0^{(\alpha_1)}$ , $B_0(x_3) = 1$ . . . . .	54
3.4	Converge rates of $J[R]$ , for case 5 with $BC_2$ and $K_m^{\alpha_1}$ , as presented in Table 3.2. . . . .	55
3.5	Converge rates of $J[R]$ for Cases 1-4, using $BC_4$ , $K_2^{(\alpha_1)}$ , $B_2(x_3) = (x_3 - 1)^2$ as presented in Table 3.3. . . . .	57
4.1	Relative error (%) of the extracted EFIF using $K_{0k}^{(\alpha_1)}$ and the "simple" family $B_2^{(k)}$ at various $R$ 's. $A^{(\alpha_1)}(x_3) = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ . . . . .	66
4.2	The exact and computed EFIF fourth-order polynomial, using $K_{2k}^{(\alpha_1)}$ at $R = 0.05$ . $A^{(\alpha_1)} = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ . . . . .	70
4.3	Relative error (%) of the extracted EFIF using $K_{2k}^{(\alpha_1)}$ at $R = 0.05$ . The exact EFIF is a fourth-order polynomial. . . . .	70
5.1	Relative error (%) of the extracted EFIF at $R = 0.05$ and the exact one using $K_{2k}^{(\alpha_1)}$ and the hierarchical family $BJ_2^{(k)}(x_3)$ (of degree $k$ ). The exact EFIF is a fourth-order polynomial $A_{ex}^{(\alpha_1)} = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ . . . . .	75
5.2	Relative error (%) of the extracted EFIF at $R = 0.05$ using $K_{2k}^{(\alpha_1)}$ and the hierarchical family $BJ_2^{(k)}(x_3)$ , with $k$ taken up to 15 <sup>th</sup> degree. . . . .	76
5.3	Relative error (%) of the extracted EFIF at $R = 0.5, 0.05$ and using Richardson's extrapolation from data at $R = 0.9, 0.5$ . EFIF computed using $K_{2k}^{(\alpha_1)}$ and the hierarchical family $BJ_2^{(k)}(x_3)$ , with $k \leq 4$ . The exact EFIF is a fourth-order polynomial $A_{ex}^{(\alpha_1)} = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ . . . . .	77

6.1	The p-FEM model for the problem having a non-polynomial EFIF tending to large values at $x_3 = \pm 1$ . . . . .	80
6.2	Relative error in FE solution and its derivatives (%) at $p = 8$ for $(r, \theta, x_3) = (0.05, \theta = 135^\circ, [-1, 1])$ for the three problems defined by $d = 2, 1.5, 1.05$ . . . . .	82
6.3	The exact and extracted EFIF, using the hierarchical extraction polynomials of degree $k$ , $K_{2k}^{(\alpha_1)}$ , $p = 8$ , at $R = 0.05$ and 54 Gauss points. . . . .	85
6.4	Relative error (%) of extracted EFIF, using the hierarchical extraction polynomials of degree $k$ , $K_{2k}^{(\alpha_1)}$ , $p = 8$ , at $R = 0.05$ and 54 Gauss points. . . . .	86
6.5	Relative error zoomed around 0% of extracted EFIF, using the hierarchical extraction polynomials of degree $k$ , $K_{2k}^{(\alpha_1)}$ , $p = 8$ , at $R = 0.05$ and 54 Gauss points. . . . .	87
6.6	Relative error (%) of the extracted EFIF at $R = 0.2, 0.05$ and using Richardson's extrapolation from data at $R = 0.9, 0.5, 0.2, 0.05$ . EFIF computed using $K_{2k}^{(\alpha_1)}$ and the hierarchical family $BJ_2^{(k)}(x_3)$ , with $k \leq 19$ . The exact EFIF is $A^{(\alpha_1)}(x_3) = \frac{\sin x_3}{(d-x_3^2)}$ , with $d = 1.05$ . . . . .	89
6.7	The exact and extracted EFIF for the problem with $d = 1.05$ , using "simple" extraction polynomials $K_{2k}^{(\alpha_1)}$ , $p = 8$ , 54 Gauss points, at different $R$ s. Bottom graph is zoomed. . . . .	92
6.8	Relative error (%) of extracted EFIF for the problem with $d = 1.05$ , using "simple" extraction polynomials $K_{2k}^{(\alpha_1)}$ , $p = 8$ , 54 Gauss points, at different $R$ s. Bottom graph is zoomed. . . . .	93
A.1	Visual Basic Form for cases 1-4, EFIF Computational. . . . .	100
A.2	Visual Basic Form for case 5, EFIF and $J[R]$ Computational. . . . .	101
A.3	Diagram of the Computer Program . . . . .	102

## LIST OF SYMBOLS

$\Omega$	3-D Domain
$\omega$	Open angle of the 3-D domain
$\Gamma_1, \Gamma_2$	Flat planes that intersect at the edge
$L$	Partial differential scalar elliptic operator
$\tau(\mathbf{x})$	Solution of the scalar elliptic problem
$\tau_{FE}(\mathbf{x})$	Finite element solution of the scalar elliptic problem
$\kappa(\mathbf{x})$	Dual solution of the scalar elliptic problem
$\alpha_i$	Eigen value
$\Phi_j^{(\alpha_i)}(r, \theta)$	Primal shadow function associated with the $i$ 'th eigen value, including $r$ dependency
$\varphi_j^{(\alpha_i)}(\theta)$	Primal shadow function associated with the $i$ 'th eigen value
$\Psi_j^{(\alpha_i)}(r, \theta)$	Dual shadow function associated with the $i$ 'th eigen value including $r$ dependency
$\psi_j^{(\alpha_i)}(\theta)$	Dual shadow function associated with the $i$ 'th eigen value,
$A^{(\alpha_i)}(x_3)$	Edge flux intensity function associated with the $i$ 'th eigen-value
$c_0^{(\alpha_i)}$	Coefficient of the dual shadow function, $\Psi_0^{(\alpha_i)}(r, \theta)$
$T(R)$	Neumann trace operator
$\Gamma_R$	Cylindrical surface with radius $R$
$J[R]$	Anti-symmetric boundary integral defined on the surface $\Gamma_R$
$m$	Number of dual shadow functions
$k$	Number of extraction polynomial
$B(x_3)$	Extraction function
$K_m^{(\alpha_i)}[B]$	Quasidual singular function

$B_m^{(k)}(x_3)$	”Simple” family of extraction polynomials
$a_i$	Coefficients of the EFIF polynomial approximation
$BJ_m^{(k)}(x_3)$	Hierarchical family of extraction polynomials
$\tilde{a}_i$	Coefficients of the Jacobi EFIF polynomial approximation
$J_n(x_3)$	Jacobi polynomial of degree $n$
$w(x_3)$	Weight function of Jacobi polynomials
$n$	Degree of the polynomial approximating the EFIF
$BC_2$	Exact EFIF is a polynomial of degree 2
$BC_4$	Exact EFIF is a polynomial of degree 4

## 1. INTRODUCTION

The fluxes associated with the solution of scalar elliptic equations (representing heat transfer in engineering science) and the stresses associated with the solution of elasticity equations may be “*singular*” (their values tend to infinity) in domains containing reentrant corners. These singularities may be the result of the domains topology such as a corner created by an intersection of boundaries, may be due to abrupt change in material properties or in the boundary conditions. The characteristic of the singular solution is of great engineering importance because structural failure theories involve it. For example, there is an entire field in mechanical engineering called Linear Elastic Fracture Mechanics, (which relies on the so-called stress intensity factors), aimed at predicting failures because of cracks in solids.

This thesis is aimed at the description of the singular solution of scalar elliptic problems in the vicinity of edges in three dimensional domains, and the computation of the so called eigen-pairs and edge flux intensity functions. This is a first step towards the computation of the edge stress intensity functions in elasticity.

As a starting point, the characteristics of the singular solution of the Laplace equation in the vicinity of a reentrant corner in a 2-D domain is addressed, and methods for the computation of the eigen-pairs and GFIFs (Generalized Flux Intensity Factors) are reviewed. The solution of the Laplace equation or elasticity equations, in a three dimensional domain containing vertex or edge singularities will be thereafter described. Due to different geometrical configuration, the solutions in the vicinity of an edge, a vertex and the intersection between the edge and the vertex are different.

Each one of these solutions is mathematically complicated, however can be extended from its two dimensional counterpart.

### 1.1 Two dimensional domains with reentrant corners

Vast amount of research has focused on the 2-D solutions in the neighborhood of singular points. The domain  $G$  of interest is a two dimensional domain with a solid angle  $\omega$ , created by the intersection of two straight edges  $\Gamma_1$  and  $\Gamma_2$ , as shown in Figure 1.1. The coordinates  $r$  and  $\theta$  are Cylindrical coordinates of a system located in the singular point. The coordinates  $x_1$  and  $x_2$  are the Cartesian coordinates, also located in the singular point.

Over the domain  $G$  we consider herein a homogeneous scalar second order partial differential operator of the form:

$$L = \sum_{i=1}^2 \sum_{j=1}^2 k_{ij} \partial_i \partial_j \quad \text{with} \quad \partial_1 = \frac{\partial}{\partial x_1}, \quad \partial_2 = \frac{\partial}{\partial x_2} \quad (1.1)$$

where  $k_{ij}$  form a symmetric  $2 \times 2$  matrix ( $k_{ij}$  represent the heat conduction coefficients for the heat transfer problem). The  $k_{ij}$ 's are constant real coefficients, which satisfy the ellipticity condition ( $\sqrt{k_{11}k_{22} - k_{12}^2} > 0$ ).

We consider homogenous Dirichlet boundary conditions on the edges  $\Gamma_1$  and  $\Gamma_2$ . (The solution of the problem is denoted by  $\tau(r, \theta)$ ):

$$\tau(r, 0) = \tau(r, \omega) = 0 \quad (1.2)$$

It can be shown (see [3] for example) that the solution in the vicinity of a general singular point can be expanded as:

$$\tau(r, \theta) = \sum_{i=1}^{\infty} \sum_{j=0}^J \sum_{k=0}^K A_{ijk} r^{\alpha_i + j} (\log(r))^k \phi_{ijk}(\theta) \quad (1.3)$$

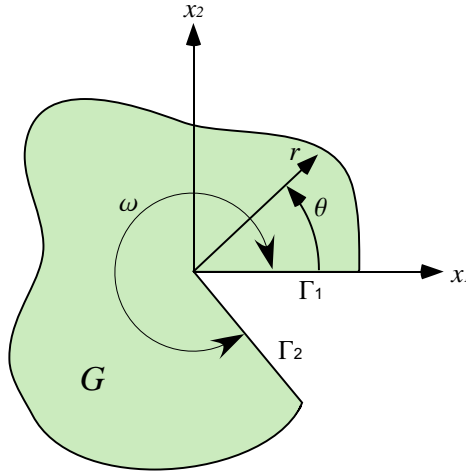


Fig. 1.1: Schematic 2-D domain,  $G$ .

where  $A_{ijk}$  are the coefficient of the asymptotic expansion (called the generalized flux intensity factors, GFIF's),  $\alpha_i$  are the eigen-values ( $\alpha_{i+1} > \alpha_i$ ) and  $\phi_{ijk}$  are the eigen-functions associated with the  $i$ 'th eigen-value.  $K \geq 0$  is an integer which is zero for most problems, except for the special case where  $\alpha_i$  is an integer, and  $J > 0$  for curved boundaries intersecting at the singular point. We are interested in cases where  $J = K = 0$ . The eigen-values and eigen-functions depend on the geometry of the domain, the boundary conditions and the  $k_{ij}$  coefficients of the operator  $L$  in the vicinity of the singular point. When the elasticity equation is addressed,  $A_i$  are called the generalized stress intensity factors, GSIF.

### 1.1.1 Computation of the eigen-pairs

The computation of eigen-pairs may be accomplished by several techniques. An analytical method for computing two dimensional isotropic eigen-pairs is provided by Dauge in [11] as well as many prior publications as [28, 23, 25]. Another analytical method is proposed by Grisvard in [15] for eigen-pairs extraction of elliptic problems. Leguillon and Sanchez-Palencia in [21] provide a method to compute the eigen-pairs

of the Laplace equation by separation of variables, where as the eigen-pairs of an elastic isotropic domain are numerically computed. The separation of variables method is also presented by Szabo and Babuška in [26] for the Laplace equation where a numerical approach is addressed for the elasticity problem. Another numerical approach for the computation of the eigen-pairs is presented by Yosibash and Szabo in [35]. The approach is based on the modified Steklov formulation and is applicable to elliptic scalar problems as well as elasticity problems for isotropic or anisotropic two dimensional domains. An analytical approach for extracting the eigen-pairs for scalar elliptic problems in anisotropic domains with multi-material corners is presented by Mantic, Paris and Berger in [22] where the concept of a  $2 \times 2$  transfer matrix is used. The eigen-pairs computation is reduced to the evaluation of a product of the transfer matrices, one for each homogenous wedge. For a general scalar elliptic problem, i.e.  $k_{11} \neq k_{22}$  and  $k_{12} \neq 0$ , we compute by separation of variables in a two dimensional domain the eigen-pairs and the so called dual eigen-pairs following [21] and [26].

For a general scalar elliptic differential equation, we use a change of variables such that the operator  $L$  in the new variables is the Laplace operator in the mapped domain  $G'$ :

$$L' = \frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2}, \quad (1.4)$$

where the new variables are:

$$\xi(x_1, x_2) = k_{11}^{-1} \sqrt{k_{11}k_{22} - k_{12}^2} x_1 \stackrel{\text{def}}{=} k_1 x_1 \quad (1.5)$$

$$\eta(x_1, x_2) = x_2 - k_{11}^{-1} k_{12} x_1 \stackrel{\text{def}}{=} x_2 - k_2 x_1 \quad (1.6)$$

The edges  $\Gamma_1$  and  $\Gamma_2$  which are defined by  $\theta = 0$  and  $\theta = \omega$  respectively for the domain  $G$  are transformed into the edges  $\Gamma'_1$  and  $\Gamma'_2$  which are defined by  $\gamma = \omega_1$  and  $\gamma = \omega_2$  (See Figure 1.2). The boundary conditions prescribed on the edges  $\Gamma'_1$

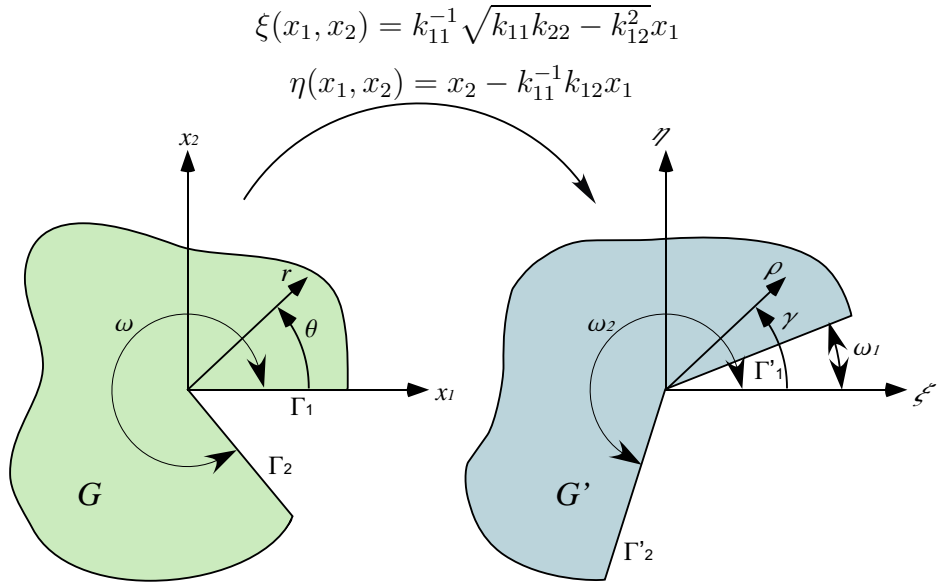


Fig. 1.2: A variable mapping transforming the domain  $G$ , into the domain  $G'$ .

and  $\Gamma'_2$  are Dirichlet boundary conditions. The new angles  $\omega_1$  and  $\omega_2$  are given by:

$$\omega_1 = \arctan\left(\frac{-k_2}{k_1}\right) \quad (1.7)$$

$$\omega_2 = \arctan\left(\frac{\tan(\omega) - k_2}{k_1}\right) \quad (1.8)$$

For the Laplace equation, the solution is easily obtained by separation of variables:

$$\tau(\rho, \gamma) = \rho^\alpha [A \cos(\alpha\gamma) + B \sin(\alpha\gamma)] \quad (1.9)$$

where  $\alpha$  is called the eigen-value. Although both positive and negative values of  $\alpha$  satisfy equation (1.9), the restriction  $\alpha \geq 0$  is imposed to insure that  $|\tau| \leq \infty$  in the vicinity of the singular point. The negative eigen-values are associated with the solution  $\kappa$  which will be referred as the dual solution:

$$\kappa(\rho, \gamma) = \rho^{-\alpha} [C \cos(\alpha\gamma) + D \sin(\alpha\gamma)] \quad (1.10)$$

The solutions  $\tau$  and  $\kappa$  must satisfy the homogenous Dirichlet boundary conditions

over the edges  $\Gamma'_1$  and  $\Gamma'_2$ :

$$\begin{pmatrix} \cos(\alpha\omega_1) & \sin(\alpha\omega_1) \\ \cos(\alpha\omega_2) & \sin(\alpha\omega_2) \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (1.11)$$

A non-trivial solution to the system (1.11) is obtained for the case when the determinant is zero, i.e:

$$\cos(\alpha\omega_1) \sin(\alpha\omega_2) - \sin(\alpha\omega_1) \cos(\alpha\omega_2) = 0 \quad (1.12)$$

and therefore:

$$\alpha_i = \frac{\pm i\pi}{\omega_2 - \omega_1}, \quad i = 1, 2, \dots \quad (1.13)$$

where each  $\alpha_i$  is associate with an  $A_i$  and a  $B_i$ . The ratios  $A_i/B_i$  and  $C_i/D_i$  are determined by:

$$\frac{A_i}{B_i} = \frac{C_i}{D_i} = -\tan(\alpha_i\omega_1) = -\tan(\alpha_i\omega_2) \quad (1.14)$$

and the solution of the operator  $L'$  on the domain  $G'$  is:

$$\tau(\rho, \gamma) = \sum_{i=1} B_i \rho^{\alpha_i} [-\tan(\alpha_i\omega_1) \cos(\alpha_i\gamma) + \sin(\alpha_i\gamma)] \quad (1.15)$$

$$\kappa(\rho, \gamma) = \sum_{i=1} D_i \rho^{\alpha_i} [-\tan(\alpha_i\omega_1) \cos(\alpha_i\gamma) + \sin(\alpha_i\gamma)] \quad (1.16)$$

To obtain the solution in terms of  $r$ ,  $\theta$ , a reverse mapping is performed on the solutions  $\tau$  and  $\kappa$  (where the unknown coefficients  $B_i$  and  $D_i$  are now denoted by  $A_i$  and  $C_i$  respectively for convenience reasons):

$$\tau(r, \theta) = \sum_{i=1} A_i r^{\alpha_i} \varphi_i(\theta) \quad (1.17)$$

$$\kappa(r, \theta) = \sum_{i=1} C_i r^{-\alpha_i} \psi_i(\theta) \quad (1.18)$$

where  $\varphi_i(\theta)$  are the eigen-function and  $\psi_i(\theta)$  are the dual eigen functions:

$$\begin{aligned} \varphi_i(\theta) = & - \left( (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right)^{\alpha_i/2} \sec(\alpha_i\omega_1) \times \\ & \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (1.19)$$

$$\begin{aligned} \psi_i(\theta) = & - \left( (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right)^{-\alpha_i/2} \sec(\alpha_i \omega_1) \times \\ & \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (1.20)$$

### 1.1.2 The path independent integral

Once the eigen-functions are obtained, one can proceed to the computation of the coefficients of the series expansion  $A_i$ 's called GFIFs. There are several methods for extracting GFIFs. One of the methods involves a path independent integral which is developed herein. The integral in question is a path integral in the domain of interest such that for any chosen path the integration yields the same value. A detailed explanation of the path independent integral for the Laplace equation and elastic isotropic equation are presented by Szabo and Babuška in [26] and by Legullon and Sanchez in [21]. A path integral, based on Betti's law, for an isotropic elastic domain was developed also by Carpenter in [6] and in further work Carpenter and Byers extended the method to bi-material domains in [8]. Another path independent integral for a bimaterial noched domain based on Betti's law was presented by Banks-Sills and Sherer in [4]. Labossiere and Dunn proposed a path integral for anisotropic domains in [18] and in their further work, [19], they expanded the path integral to bi-material domains. The path integral presented in this work is based on [26] and is applicable to the general scalar elliptic problem.

Let  $\Omega$  be a sub-domain of  $G$ , such that  $\Omega \subset G$  with the boundaries  $\partial\Omega = \cup_i \Gamma_i$ . The boundaries  $\Gamma_1$  and  $\Gamma_2$  of the sub-domain  $\Omega$  are the edges of the domain  $G$  as illustrated in Figure 1.3.

The problem of interest in the vicinity of the singular point  $(0, 0)$  is:

$$L(\tau) \triangleq \bar{\nabla} \cdot ([k] \bar{\nabla} \tau) = 0 \quad \text{in } G \quad (1.21)$$

$$\tau = 0 \quad \text{on } \Gamma_1 \cup \Gamma_2 \quad (1.22)$$

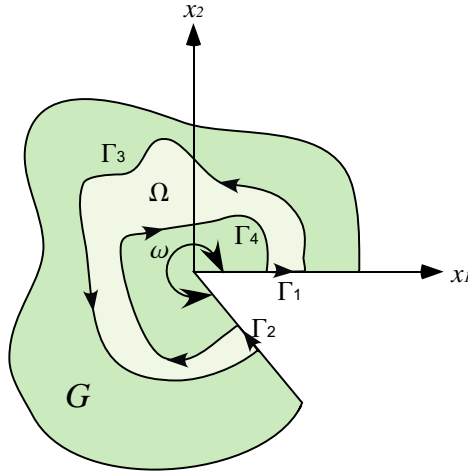


Fig. 1.3: The sub-domain  $\Omega$  with the boundaries  $\Gamma_i$ .

where:

$$[k] = \begin{pmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{pmatrix} \quad (1.23)$$

Since  $\Omega \subset G$ , (1.21) holds in  $\Omega$  and since each  $\tau_i = A_i r^{\alpha_i} \rho_i(\theta)$  satisfy (1.21) and (1.22), we have in  $\Omega$ :

$$L(\tau_i) = 0 \quad \text{in } \Omega \quad (1.24)$$

$$\tau_i = 0 \quad \text{on } \Gamma_1 \cup \Gamma_2 \quad (1.25)$$

Multiplying equation (1.24) by a function  $\kappa_j$  and integrating over the sub-domain  $\Omega$  one obtains:

$$\int_{\Omega} L(\tau_i) \kappa_j d\Omega = \int_{\Omega} (\bar{\nabla} \cdot ([k] \bar{\nabla} \tau_i)) \kappa_j d\Omega \quad (1.26)$$

where  $\kappa_j$  is chosen to be the dual solution associated with the  $j$ 'th dual eigen pair.

Applying Green's formula on (1.26):

$$\begin{aligned} & \int_{\Omega} (\bar{\nabla} \cdot ([k] \bar{\nabla} \tau_i)) \kappa_j d\Omega = \\ & = \oint_{\partial\Omega} ([k] \bar{\nabla} \tau_i \cdot \bar{n}) \kappa_j dl - \oint_{\partial\Omega} ([k] \bar{\nabla} \kappa_j \cdot \bar{n}) \tau_i dl + \int_{\Omega} (\bar{\nabla} \cdot ([k] \bar{\nabla} \kappa_j)) \tau_i d\Omega \end{aligned} \quad (1.27)$$

where  $\bar{n}$  is the outward normal vector to the boundary. Since both  $\tau_i(r, \theta)$  and  $\kappa_j(r, \theta)$  satisfy the general homogenous scalar elliptic equation,  $L(\tau_i) = L(\kappa_j) = 0$ ,

equation (1.27) simplifies to:

$$\oint_{\partial\Omega} ([k]\bar{\nabla}\tau_i \cdot \bar{n})\kappa_j dl = \oint_{\partial\Omega} ([k]\bar{\nabla}\kappa_j \cdot \bar{n})\tau_i dl \quad (1.28)$$

The boundaries  $\Gamma_1$  and  $\Gamma_2$  of the sub-domain  $\Omega$  are the edges of the domain  $G$  where the homogenous boundary conditions are prescribed. Therefore the integration over the edges  $\Gamma_1$  and  $\Gamma_2$  vanishes. By changing the direction of integration on  $\Gamma_3$ , equation (1.28) is simplified:

$$\begin{aligned} \int_{\Gamma_3} (([k]\bar{\nabla}\tau_i \cdot \bar{n})\kappa_j - ([k]\bar{\nabla}\kappa_j \cdot \bar{n})\tau_i) dl &= \\ = \int_{\Gamma_4} (([k]\bar{\nabla}\tau_i \cdot \bar{n})\kappa_j - ([k]\bar{\nabla}\kappa_j \cdot \bar{n})\tau_i) dl & \end{aligned} \quad (1.29)$$

Since the edges  $\Gamma_3$  and  $\Gamma_4$  are randomly selected, the right hand side as well as the left hand side of relation (1.29) may be considered as an invariant where  $\Gamma$  is a path in  $G$  which starts at one edge and ends at the other edge of the domain, so that

$$J = \int_{\Gamma} (([k]\bar{\nabla}\tau_i \cdot \bar{n})\kappa_j - ([k]\bar{\nabla}\kappa_j \cdot \bar{n})\tau_i) dl \quad (1.30)$$

is path independent, where various paths  $\Gamma$  are illustrated in figure 1.4.

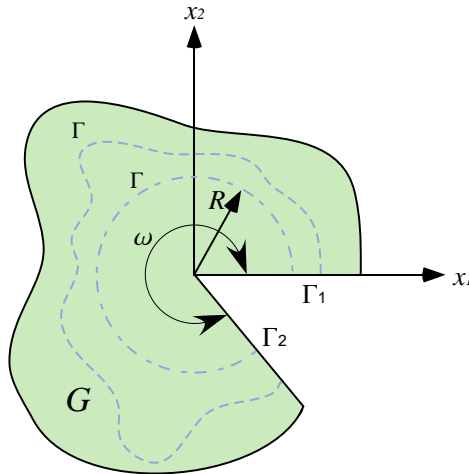


Fig. 1.4: Considered paths in the domain  $G$ .

Let  $\Gamma$  be a circular path chosen over the domain  $G$ , as illustrated in Figure 1.4. The normal  $\bar{n}$  along the circular path  $\Gamma$  is at  $\hat{r}$  direction such that:

$$\begin{aligned} [k]\bar{\nabla}\tau_i \cdot \bar{n} &= \left[ \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} \begin{pmatrix} \partial_1 \\ \partial_2 \end{pmatrix} \tau_i \right]^T \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \\ &= \left[ \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} \begin{pmatrix} \cos(\theta)\partial_r - \frac{1}{r}\sin(\theta)\partial_\theta \\ \sin(\theta)\partial_r + \frac{1}{r}\cos(\theta)\partial_\theta \end{pmatrix} \tau_i \right]^T \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \end{aligned} \quad (1.31)$$

the explicit form is:

$$\begin{aligned} [k]\bar{\nabla}\tau_i \cdot \bar{n} &= (k_{11}\cos^2(\theta) + k_{12}\sin(2\theta) + k_{22}\sin^2(\theta))\partial_r\tau_i \\ &\quad - \frac{1}{r}\left(\frac{1}{2}(k_{11} - k_{22})\sin(2\theta) - k_{12}\cos(2\theta)\right)\partial_\theta\tau_i \\ &= A_i r^{\alpha_i-1} [\alpha_i(k_{11}\cos^2(\theta) + k_{12}\sin(2\theta) + k_{22}\sin^2(\theta))\varphi_i(\theta) \\ &\quad - \left(\frac{1}{2}(k_{11} - k_{22})\sin(2\theta) - k_{12}\cos(2\theta)\right)\varphi'_i(\theta)] \\ &\stackrel{\text{def}}{=} A_i r^{\alpha_i-1} H_i^\varphi(\theta) \end{aligned} \quad (1.32)$$

and

$$\begin{aligned} [k]\bar{\nabla}\kappa_j \cdot \bar{n} &= C_j r^{-\alpha_j-1} [-\alpha_j(k_{11}\cos^2(\theta) + k_{12}\sin(2\theta) + k_{22}\sin^2(\theta))\psi_j(\theta) \\ &\quad - \left(\frac{1}{2}(k_{11} - k_{22})\sin(2\theta) - k_{12}\cos(2\theta)\right)\psi'_j(\theta)] \\ &\stackrel{\text{def}}{=} C_j r^{-\alpha_j-1} H_j^\psi(\theta) \end{aligned} \quad (1.33)$$

The path independent integral  $J$  is:

$$\begin{aligned} J &= \int_\Gamma (([k]\bar{\nabla}\tau_i \cdot \bar{n})\kappa_j - ([k]\bar{\nabla}\kappa_j \cdot \bar{n})\tau_i) R d\theta \\ &= \int_\Gamma \left( A_i R^{\alpha_i-1} H_i^\varphi(\theta) C_j R^{-\alpha_j} \psi_j(\theta) - C_j R^{-\alpha_j-1} H_j^\psi(\theta) A_i R^{\alpha_i} \varphi_i(\theta) \right) R d\theta \\ &= A_i C_j R^{\alpha_i-\alpha_j} \int_0^\omega \left( H_i^\varphi(\theta)\psi_j(\theta) - H_j^\psi(\theta)\varphi_i(\theta) \right) d\theta \end{aligned} \quad (1.34)$$

Since  $J$  is an invariant regarding any chosen path, it can not be dependent on  $R$  and therefore for  $i \neq j$  the integral in (1.34) vanishes. The orthogonality property

of the eigen function and their dual is therefore established:

$$\begin{cases} i \neq j & J = 0, \\ i = j & J = A_i C_i \int_0^\omega \left( H_i^\varphi(\theta) \psi_i(\theta) - H_i^\psi(\theta) \varphi_i(\theta) \right) d\theta \end{cases} \quad (1.35)$$

### 1.1.3 Extracting the GFIF

As was mentioned in the previous section, there are few methods for extracting the GFIF. Two numerical methods for the GFIF calculation are presented in [2] by Babuska and Miller. The first method is the dual function method while the other method is based on the energy release principle. A numerical method based on the complementary energy formulation and the p-version of finite element method for computation of the GFIF's for two dimensional scalar elliptic problems and elasticity problems is presented by Szabo and Yosibash in [27]. In a later publication, [34], Yosibash and Szabo extended the methods to anisotropic materials (GSIF) and multi-material domains. Carpenter presented a method for extracting GSIF for isotropic domains with complex eigen values in [7]. The method is based on a path independent integral and requires two computations of the integral at two different paths. Banks-Sills and Sherer, [4], extracted the GFIF of bimaterial notched domains where the solution contains both real and complex eigen values. The extraction method presented below is based on the path integral method for extraction of the GFIF. The method involves integrating the analytical dual solution and the finite element approximation of  $\tau$  along  $\Gamma$  in the vicinity of the singular point, as shown below.

Relation (1.35) shows that when one takes the solution  $\tau = \sum_{i=1} A_i r^{\alpha_i} \varphi_i(\theta)$  together with the dual solution  $\kappa_j = C_j r^{-\alpha_j} \psi_j(\theta)$  and inserts them in the  $J$  integral (1.30), where  $C_j$  is chosen as:

$$C_j = \left( \int_0^\omega \left( H_i^\varphi(\theta) \psi_i(\theta) - H_i^\psi(\theta) \varphi_i(\theta) \right) d\theta \right)^{-1} \quad (1.36)$$

then using (1.35), equation (1.30) becomes:

$$A_i = \int_{\Gamma} (([k]\bar{\nabla}\tau \cdot \bar{n})\kappa_i - ([k]\bar{\nabla}\kappa_i \cdot \bar{n})\tau) dl \quad (1.37)$$

Since the exact solution  $\tau(r, \theta)$  is unknown ( $A_i$ , the GFIF, are unknown), one cannot compute the RHS of (1.37) analytically. Therefore a numerical method is addressed where the exact solution  $\tau(r, \theta)$  is replaced by a finite element approximation,  $\tau(r, \theta)_{FE}$ . Applying the finite element solution in equation (1.37), we finally obtain:

$$A_i = \int_{\Gamma} (([k]\bar{\nabla}\tau_{FE} \cdot \bar{n})\kappa_i - ([k]\bar{\nabla}\kappa_i \cdot \bar{n})\tau_{FE}) dl \quad (1.38)$$

In order to receive the  $i$ 'th GFIF, the dual solution  $\kappa(r, \theta)$  in the RHS of equation (1.38) is considered with the  $i$ 'th dual eigen-pairs only. The solution approximation  $\tau_{FE}$  and the approximation of the exact solution derivatives  $\partial_r\tau_{FE}$   $\partial_\theta\tau_{FE}$  are considered in (1.38) computation. Since the  $A_i$  computation does not depend on the selected radius  $R$  of the path  $\Gamma$ , the path is chosen away from the singular point in order to avoid the discretization errors of the finite element approximation, inherent in the elements adjacent to the singularity.

## 1.2 Finite Element Method

There are many complex problems in the engineering design field where the mathematical formulation cannot be solved by analytical methods. The finite element method (FEM) is a powerful numerical method for a wide range of engineering problems. The process steps of solving a physical problem with the finite element method is presented in Figure 1.5.

As shown in Figure 1.5, each simulation step introduces a source of error. Although the modelling errors are very important in engineering practice, the errors

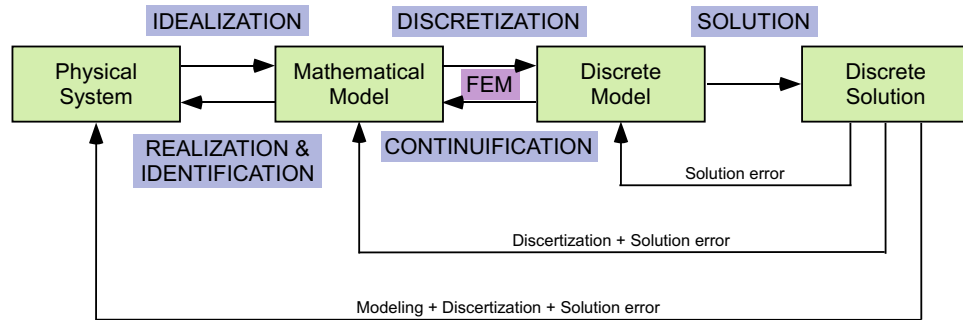


Fig. 1.5: Diagram of the Finite Element Method Process.

due to the mathematical modelling are difficult to evaluate since the solution must be compared with experimental results which are not always available. The next in order of importance are the discretization errors. The errors of the discretization together with the errors of the solution can be approximated by substituting the solution in the mathematical model where the solution error approximation itself is given by substituting the solution in the discrete model. By evaluating the errors, one can decide when to terminate the iterative procedure.

The basic concept of the FEM is the subdivision of the mathematical model into components of simple geometry called *elements*. The response of each element is expressed in terms of a finite number of degrees of freedom characterized as the value of an unknown function of functions at a set of *nodal points* (also called *nodes*). The response of the mathematical model is then considered to be approximated by that of the discrete model obtained by connecting or assembling the collection of all elements.

The physical relation of each element and the physical system of equations that describe the mathematical model is a linear relation between  $\mathbf{u}$  and  $\mathbf{f}$  of the form:

$$\mathbf{k}\mathbf{u} = \mathbf{f} \quad (1.39)$$

where  $\mathbf{k}$  is the stiffness matrix which represents the physical quantities of the structure and the vectors  $\mathbf{u}$  and  $\mathbf{f}$  vary according to the application being modelled, as shown in Table 1.1. Relation (1.39) is derived from the variational formulation of

minimum potential energy.

*Tab. 1.1:* The physical significance of vector  $\mathbf{u}$  and vector  $\mathbf{f}$  in FEM applications

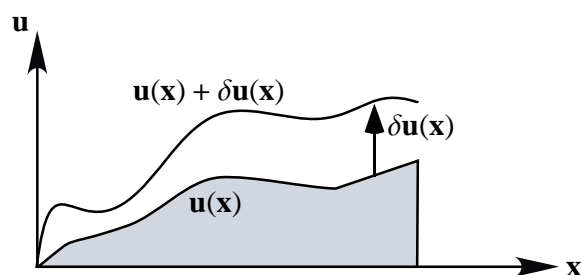
Application Problem	vector $\mathbf{u}$	vector $\mathbf{f}$
Structures and solid mechanics	Displacement	Mechanical force
Heat conduction	Temperature	Heat flux
Acoustic fluid	Displacement potential	Particle velocity
Potential flows	Pressure	Particle velocity
General flows	Velocity	Fluxes
Electronics	Electric potential	Charge density
Magnetostatics	Magnetic potential	Magnetic intensity

### 1.2.1 The variational formulation

The potential energy of the structure depends on the internal energy,  $U$ , and the external energy,  $W$ , of the structure such that:

$$\Pi = U - W \quad (1.40)$$

where  $\Pi$  is the total potential energy. The total potential energy is a functional depending only on the vector  $\mathbf{u}$ . The concept of the variation formulation is to describe every admissible function of  $\mathbf{u}$  with respect to the actual  $\mathbf{u}$  vector, as shown in Figure 1.6 for a 1-D problem.



*Fig. 1.6:* The concept of variation of the functional  $\mathbf{u}(\mathbf{x})$

The functionals  $\delta \mathbf{u}$  and  $\delta \Pi$  are the variations of  $\mathbf{u}$  and  $\Pi$  respectively. Both  $\mathbf{u}$  and  $\mathbf{u} + \delta \mathbf{u}$  are admissible in the sense of the principle of virtual work, they are both continuous over  $\mathbf{x}$  and satisfy the boundary conditions which are specific over the domain. The minimum potential energy principle states that the actual  $\mathbf{u}$  vector satisfies equation (1.40) as well as:

$$\delta \Pi = \delta U - \delta W = 0 \quad (1.41)$$

and therefore the solution of equations (1.40) and (1.41) is the vector  $\mathbf{u}$  of the mathematical model. Since the stiffness matrix  $\mathbf{k}$  is unknown yet, it is impossible to solve equations (1.40) and (1.41) and therefore the variational formulation is prescribed over a discrete number of elements such that the stiffness matrix for each element may be established. By assembling the elements, an approximated system of equation is obtained for the mathematical model.

For further details on the variational formulation see [14], [20].

### 1.2.2 Structure Discretization

In order to apply the variation formulation to the derivation of the finite element method, the mathematical model is replaced by a discrete one, as shown in Figure 1.7. The functional (1.40) may be decomposed into the sum of contributions of

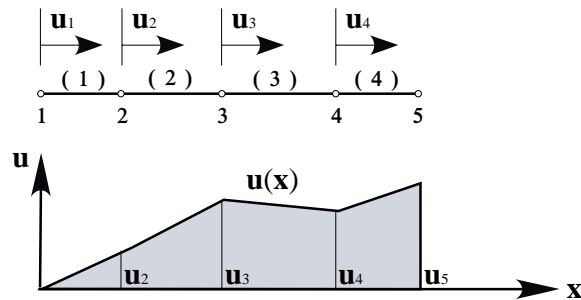


Fig. 1.7: FEM discretization of a structure

individual elements:

$$\Pi = \Pi^{(1)} + \Pi^{(2)} + \dots + \Pi^{(N_e)} \quad (1.42)$$

where  $N_e$  is the number of elements. The same decomposition applies on the internal and external energy as well as on the variation formulation:

$$\delta\Pi^{(e)} = \delta U^{(e)} - \delta W^{(e)} = 0 \quad (1.43)$$

The variation formulation in equation (1.43) is the basis of the derivation of the stiffness equations that describe the matrix  $\mathbf{k}$ .

Further details about the finite element method, the FEM procedure, the shape functions and variational formulation of elastic and heat transfer example problems can be found in [13], [17], [26] and [29].

### 1.2.3 The $h$ and $p$ versions of the FEM

The finite element method is an iterative method, as shown in Figure 1.5. The convergence rate of the solution is calculated in each iteration and compared to the previous rate, until a satisfying rate is achieved and the sequence of approximations is terminated.

There are two iteration versions in the finite element method: the  $h$ -version and the  $p$ - or  $hp$ -version. The convergence rate of each version is examined over a uniform mesh and over an optimally refined mesh. In the  $h$ -version of the finite element method one subdivides each element and therefore increase the number of elements in the structure. The convergence rate of the  $h$ -version is measured by the examination of error in the energy norm. It can be shown that when a uniform mesh is in question, the relation between  $\log \|e\|$  and  $\log N$  where  $\|e\|$  is the error in energy norm and  $N$  is the number of degrees of freedom, is a straight line. This convergence rate is called *algebraic convergence*. In the  $p$ -version of the FEM, the degree of the approximated

polynomial is increased in each element at each iteration. The convergence rate of the p-version over a uniform mesh when the exact solution is a smooth function is much more rapid than the algebraic convergence rate. If the exact solution is not smooth the convergence rate may change between the algebraic convergence rate at least twice the algebraic convergence rate of the h-version at most.

It can be shown that if an optimally refined mesh is used, both versions improve their convergence rate, where the convergence rate of the p- version is still faster than the rate of the h-version. More details about the h-version and the p-version can be found in [26].

Since the hp-version of the FEM has a faster convergence rate than the h-version, the hp-version was chosen as the iterative version of FEM in this work. The finite element software that was selected in this work is StressCheck<sup>1</sup>, see [12]. The software is capable of working with the p-version of the FEM up to the 8 degree of the approximated polynomial at each element.

### 1.3 Three dimensional domain with an edge singularity

Although the behavior of a two dimensional flux field in the vicinity of a singular point is well known, in reality three dimensional domains are present. Therefore a refined analysis for three dimensional solutions in the vicinity of an edge is required.

Consider a 3 –  $D$  domain  $\Omega$  with a solid angle  $\omega$ , created by an intersection of two flat surfaces, as shown in Figure 1.8. We denote the flat surfaces by  $\Gamma_1$  and  $\Gamma_2$ .

There are three different expansions of the solution to second order scalar elliptic problems, depending on whether it is in the vicinity of an edge, a vertex or the

---

<sup>1</sup> StressCheck is a trade mark of Engineering Software Research and Development, Inc., 10845 Olive Blvd., Suite 170, St. Louis, MO 63141, U.S.A.

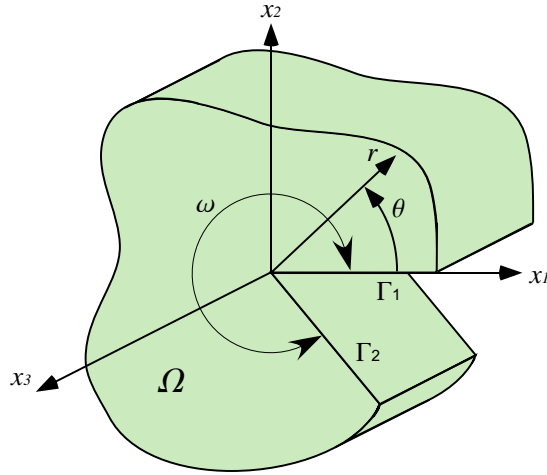


Fig. 1.8: Schematic 3-D domain,  $\Omega$ .

intersection of the edge with the vertex (see figure 1.9). The solution in the vicinity of a vertex can be represented as:

$$\tau = \sum_{i=1} \rho^{\gamma_i} g_i(\theta, \varphi) \quad (1.44)$$

where  $\rho, \theta, \varphi$  are coordinates of a spherical coordinate system located in the vertex. The scalar  $\gamma_i$  are denoted as eigen-values with their correspondent analytic eigen-function  $g_i(\theta, \varphi)$  away from the edge. The eigen-functions may have edge singularities in the vicinity of the edge. Herein, we do not address vertex singularities, and for further details the reader is referred to [15]. In this work we consider the solution in the vicinity of an edge (which will be denoted by  $E$ ) only.

The series expansion of the solution in the vicinity of the edge (as can be found in [5], [30], [31], [16] and [32]) is of the form:

$$\tau(r, \theta, x_3) = \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} A_{ij}(x_3) r^{\alpha_i+j} \varphi_{ij}(\theta) \quad (1.45)$$

where each edge singularity along  $E$  is characterized:

- by an *exponent*  $\alpha$  which depends only on the geometry and the operator, and

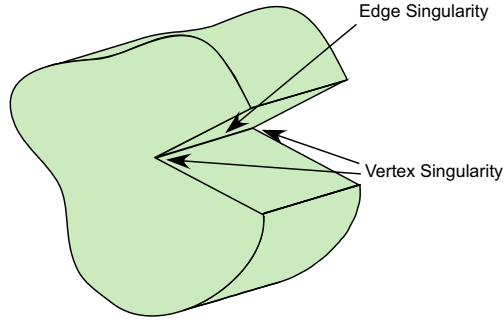


Fig. 1.9: Edge and vertex singularities in a 3-D domain.

which determines the level of non-smoothness of the singularity (and its corresponding eigen-function). The *eigen-value*  $\alpha_i$  is extracted from the  $2 - D$  problem.

- by an *eigen-function*  $\varphi_{ij}(\theta)$  which depends on the geometry of the domain and the operator. The *eigen-functions*  $\varphi_{i0}$  are extracted from the  $2 - D$  problem.
- and by a *function* along the edge  $E$ , denoted by  $A_{ij}(x_3)$  ( $x_3$  is a coordinate along the edge) and called the “Edge Flux Intensity Function” (EFIF) which determines the “amount of energy” residing in each singularity.

### 1.3.1 Extracting the eigen-pairs

A common method for extracting the eigen-pairs of the  $3 - D$  problem is to extract the eigen-pairs of the related  $2 - D$  problem and use those eigen-pairs in the  $3 - D$  solution. This method is useful when a scalar elliptic equation or an isotropic elastic equation is in question but the method is not applicable in other problems such as the anisotropic elastic equation. Few methods for eigen-pairs extraction in three dimensional domains have been approached in the past for the anisotropic elastic equation where most of them are numerical methods. One of the three dimensional anisotropic elastic domains and multi-material domains approach is presented by Pageau and

Biggers in [24]. The method is based on the finite element method where the geometry of the domain is divided into a number of quadratic elements. The authors show that the method converges as the number of elements in the domain increases. Another numerical method, based on the p-FEM was proposed by Yosibash in [30], [32], [33]. The method is applicable to anisotropic multi-material three dimensional domains and is based on the modified Steklov formulation. The method presented by Costabel, Dauge and Lafranche in [9] is a semi-analytic method for elastic anisotropic materials.

### 1.3.2 Extracting the EFIF

The path independent integral method in two dimensional domains cannot be extended easily to the case of edge singularities in three dimensional domains. In  $3-D$  the path independent integral has to be considered on three different surface integrals when extended to the 3-D domain, as shown in figure 1.10. The first integral is a circular cylindrical surface integral that starts at the boundary  $\Gamma_1$  and ends at the boundary  $\Gamma_2$ . The other two integrals are circular surface integrals at the bases of the domain  $\Omega$  (at a specific  $x_3$ ). The two surface integrals at the bases of the domain

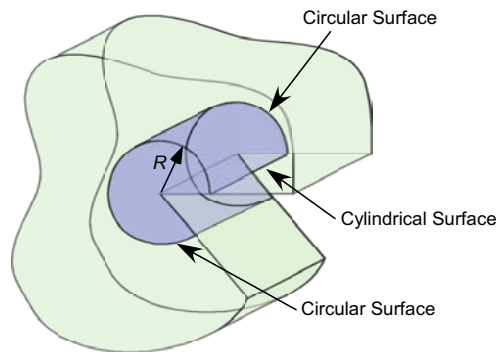


Fig. 1.10: Three surface integrals in the 3-D domain.

depend on the selection of the radius  $R$  and therefore the method of a surface independent integral does not seem to work. Other methods of EFIF's extraction have been developed. The first method for EFIF extraction was presented by Bochniak and Sandig in [5]. The method involves Mellin transformations, Pencil operators and their Jordan-chains. Another method which is based on a two dimensional projection, Richardson extrapolation and the p-finite element solution was presented by Yosibash, Actis and Szabo in [33] for point-wise extraction of EFIF's.

*The EFIF point-wise extraction method for Laplace operator in 3 – D*

The point-wise extraction method (as presented in [33]) is based on a 2-D projection, and applies to the Laplace operator in 3–D domains. The three-dimensional solution can be constructed by the two dimensional eigen-pairs:  $\alpha_n$  (the eigen-value) and  $\varphi_n(\theta)$  (the eigen-function), where the eigen-functions are orthogonal with respect to the path integral in a plane perpendicular to the edge:

$$\int_{\theta=0}^{\omega} \varphi_n(\theta)\varphi_m(\theta)d\theta = 0 \quad \text{for } n \neq m \quad (1.46)$$

The orthogonal relation is based on the contour integral method for GFIF extraction (see previous section). First let us derive the expansion of the solution for the Laplace operator in the vicinity of an edge. Let  $A_n(x_3)$  be the edge flux intensity function associated with the n'th 2-D eigen-pair. It is clear that  $\tau_n = A_n(x_3)r^{\alpha_n}\rho_n(\theta)$  does not satisfy the three dimensional Laplace equation:

$$\begin{aligned} \nabla_{3D}^2 \tau_n &= \left( \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2} \right) A_n(x_3)r^{\alpha_n}\varphi_n(\theta) = \\ &= \left( \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \right) A_n(x_3)r^{\alpha_n}\varphi_n(\theta) + A_n''(x_3)r^{\alpha_n}\varphi_n(\theta) = \\ &= A_n''(x_3)r^{\alpha_n}\varphi_n(\theta) \neq 0 \end{aligned} \quad (1.47)$$

To avoid the residual in (1.47) one may add  $\frac{-1}{4(\alpha_n+1)}A_n''(x_3)r^{\alpha_n+2}\varphi_n(\theta)$  to  $A_n(x_3)r^{\alpha_n}\varphi_n(\theta)$  but the combination does not satisfy the three- dimensional Laplace equation either:

$$\begin{aligned} \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}\right) \left(A_n(x_3)r^{\alpha_n}\varphi_n(\theta) + \frac{-1}{4(\alpha_n+1)}A_n''(x_3)r^{\alpha_n+2}\varphi_n(\theta)\right) = \\ = \frac{-1}{4(\alpha_n+1)}A_n''''(x_3)r^{\alpha_n+2}\varphi_n(\theta) \neq 0 \end{aligned} \quad (1.48)$$

In order to eliminate the new residual, another term must be added to the combination of eigen-functions. It is clear, though, that the new combination will not satisfy the three dimensional Laplace equation either but will create a new residual. The combination of functions that will satisfy the three dimensional Laplace equation for the  $n$ 'th eigen-pair is the infinite series:

$$\tau_n(r, \theta, x_3) = r^{\alpha_n}\varphi_n(\theta) \sum_{i=0}^{\infty} \partial_3^{2i} A_n(x_3) r^{2i} \frac{(-1/4)^i}{\prod_{j=0}^i j(\alpha_n + j)} \quad (1.49)$$

where  $\partial_3^{2i} = \frac{\partial^{2i}}{\partial x_3^{2i}}$  (see details in [33]), so that:

$$\tau(r, \theta, x_3) = \sum_{n=1}^{\infty} \tau_n(r, \theta, x_3) = \sum_{n=1}^{\infty} r^{\alpha_n}\varphi_n(\theta) \sum_{i=0}^{\infty} \partial_3^{2i} A_n(x_3) r^{2i} \frac{(-1/4)^i}{\prod_{j=0}^i j(\alpha_n + j)} \quad (1.50)$$

Once the series expansion that describes the 3-D Laplace solution in the vicinity of an edge is known, the next step is to extract the EFIFs,  $A_i(x_3)$ . For extraction purposes, the 3-D solution is projected onto a two dimensional section of the 3-D domain at a specific point  $x_3 = x_3^*$ , such that the solution in a plane perpendicular to the edge  $E$  at  $x_3^*$  is approximated by:

$$\tau^N = \sum_{i=1}^N B_i(x_3^*) r^{\alpha_i} \varphi_i(\theta) \quad (1.51)$$

One wishes that  $\tau^N$  be the projected best approximation from all functions  $v^N$  so that:

$$\int \int (\tau - \tau^N) v^N r dr d\theta = 0 \quad (1.52)$$

Therefore we can rephrase this restriction into the form:

$$\int_r \int_\theta \tau v^N|_{x_3=x_3^*} r dr d\theta = \int_r \int_\theta \tau^N v^N|_{x_3=x_3^*} r dr d\theta \quad (1.53)$$

for all  $v^N$  which may be constructed by a linear combination of the  $2 - D$  eigenfunction:

$$v^N(x_3^*) = \sum_{i=1}^N C_i r^{\alpha_i} \varphi_i(\theta) \quad (1.54)$$

By substituting equations (1.49), (1.51) and (1.54) into (1.53), and with the use of the orthogonal relation (1.46), a set of equations is obtained for the coefficients  $B_i(x_3^*)$  (for more details see [33]):

$$\begin{aligned} B_i(x_3^*) &= A_i(x_3^*) + R^2 \frac{c_{i1}(\alpha_i + 1)}{\alpha_i + 2} (\partial_3^2 A_i(x_3^*)) + \\ &+ R^4 \frac{c_{i2}(\alpha_i + 1)}{\alpha_i + 3} (\partial_3^4 A_i(x_3^*)) + \mathcal{O}(R^6) \end{aligned} \quad (1.55)$$

It can be shown that by using decreasing values of  $R < 1$ , followed by Richardson's extrapolation, equation (1.55) reduces to  $B_i(x_3^*) = A_i(x_3^*)$ .

The solution  $\tau$  is still unknown and therefore an approximation of the solution is used instead. The approximation which will be used is a finite element solution, denoted by  $\tau_{FE}$ . The values of  $\tau^N$  and  $v^N$  in equation (1.53) are substituted by equations (1.51) and (1.54) respectively where the exact solution in (1.53) is replaced with the approximation  $\tau_{FE}$ . With the use of the orthogonal relation (1.46), an explicit equation for the required element  $B_i(x_3^*)$  is obtained:

$$B_i(x_3^*) = \frac{\int_{r=0}^R \int_{\theta=0}^\omega \tau_{FE}(r, \theta, x_3) r^{\alpha_i+1} \varphi_i(\theta) dr d\theta}{\frac{R^{2\alpha_i+2}}{2\alpha_i+2} \int_{\theta=0}^\omega \varphi_i^2(\theta) d\theta} + \mathcal{O}(R^2) \quad (1.56)$$

where the integrals that involve the finite element solution are numerically computed.

The accuracy of the finite element approximation is low in the elements at the vicinity of the singular point and therefore the integration on a sector from  $r = 0$  to  $r = R$  lowers the accuracy of  $B_i(x_3^*)$ . In order to increase the accuracy, the

integration is performed over a circular ring from  $r = 0.9R$  to  $r = R$  and equation (1.56) becomes:

$$B_i(x_3^*) = \frac{\int_{r=0.9R}^R \int_{\theta=0}^{\omega} \tau_{FE}(r, \theta, x_3) r^{\alpha_i+1} \varphi_i(\theta) dr d\theta}{\frac{(1-0.9^{2\alpha_i+2})R^{2\alpha_i+2}}{2\alpha_i+2} \int_{\theta=0}^{\omega} \varphi_i^2(\theta) d\theta} + \mathcal{O}(R^2) \quad (1.57)$$

Each  $B_i(x_3^*)$  has to be extracted several times at decreasing values of  $R < 1$ , followed by Richardson extrapolation for estimation of the exact value at  $r \rightarrow 0$ .

Although the method provides an accurate value of the EFIF at  $x_3^*$ , the EFIF can not be obtained in a straight forward manner. The method produces discrete values of the EFIF and in order to obtain functional representation of the EFIF it should be computed at many points  $x_3^*$  along the edge. Moreover, the method is applicable to the Laplace equation only and is not easily expanded to a general scalar elliptic equation.

#### *A new method for extracting the functional representation of the EFIF*

The methods presented for EFIF extraction are point-wise, i.e.: the EFIF value at a specific point on the edge is extracted. A newly developed theoretical framework for extracting the functional representation of the EFIF along the edge was recently presented by Costabel, Dauge and Yosibash in [10]. The method involves special extraction formulas and newly generated eigen-functions and their "shadows" for the computation of the EFIF. The implementation and numerical realization of the methods are the aim of this work. We herein implemented the methods as a post-processing step in a p-version finite element code. The methods are adaptive, provide a better functional representation of the EFIF as the special hierarchical family of extraction polynomials (Jacobi orthogonal polynomials, [1]) is increased. The extraction can be applied to any edge (including crack front) of any polyhedron, and is extendable to problems of 3-D elasticity.

The notation of the scalar elliptic problem and the eigen-pairs extraction method are given in Chapter 2. The series expansion of the  $3 - D$  scalar elliptic solution and dual solution in the neighborhood of an edge is constructed by the  $2 - D$  eigen pairs. Each  $2 - D$  eigen function is followed by a series of "shadow functions" in the  $3 - D$  solution such that the scalar elliptic equation is satisfied. Five examples are presented and their eigen-pairs and shadow functions are given explicitly.

The orthogonal relation of the solution and the dual solution is established in Chapter 3. The relation is an extension of the contour integral method, denoted by  $J[R]$ , and is a circular cylindrical surface integral that start at the boundary  $\Gamma_1$  and ends at the boundary  $\Gamma_2$ . The convergence rate of  $J[R]$  with respect to the radius  $R$  depends on the selection of the dual EFIF, denoted by  $B(x_3)$ . The results and convergence rate of  $J[R]$  for the five examples with several selections of the dual EFIF are given.

Since the functional representation of the EFIF is unknown, we are interested in the polynomial approximation of the EFIF. In order to receive an EFIF polynomial of degree  $N$ , the integral  $J[R]$  is computed  $N + 1$  times for  $N + 1$  different  $B(x_3)$  functions, and a system of equations of the EFIF polynomial coefficients is set to be solved. A special family of polynomials (called the "simple" family of polynomials) is constructed in Chapter 4 such that each  $J[R]$  computation with the  $n$ 'th dual EFIF,  $B(x_3)$ , of the "simple" family of polynomials, extract the  $n$ 'th coefficient of the EFIF. Numerical results of several examples of EFIF extractions are given. The results are compared with the exact EFIF and the convergence rate of the extraction method is examined.

Another family of polynomials is constructed in Chapter 5, called the hierarchical family of extraction polynomials. By using the hierarchical family of polynomials for the EFIF extraction, the extraction method becomes adaptive in the sense that when an EFIF polynomial is extracted of  $N$  degree, in order to increase the degree

---

of polynomial to  $N + 1$  one has to compute the  $J[R]$  once more with the  $N + 1$  hierarchical polynomial and follow a short procedure, where in the "simple" family of polynomials a new family of polynomials related to the new requested degree has to be constructed. The hierarchical family of polynomials is based on Jacobi polynomials and the EFIF representation becomes a linear combination of the hierarchical family of polynomials. Numerical results of several examples of EFIF extractions are given. The results are compared with the exact EFIF of the problem and the convergence rate of the extraction method is examined.

Chapter 6 examines the extraction method of the EFIF when the exact EFIF is not a polynomial. Three non-polynomial exact EFIFs are selected in this Chapter and the convergence rate of the extracted EFIF is examined. The convergence rate using different  $R$ 's in the  $J[R]$  computation, and the convergence rate using different polynomial degrees in the FEM approximation were examined for both "simple family" of polynomials and hierarchical family of polynomials. The influence of different degrees of polynomials were examined using the hierarchical family of polynomials only.

The numerical results in Chapters 3, 4, 5 and 6 reveal a powerful method for EFIF extraction. The method is accurate and efficient providing a polynomial representation of the EFIF along the edge.

Via a COM interface, using Visual Basic, a computer code has been developed. Appendix A describes the computational algorithm, starting with activating Stress-Check, loading the parametric FE model, changing the necessary parameters, running the FE analysis and then extracting the solution and its derivatives at points required for  $J[R]$  computation. Finally the code computes  $J[R]$  and thus  $a_i$ .

## 2. THE SCALAR ELLIPTIC PROBLEMS OF INTEREST & SINGULAR SOLUTION NEAR EDGES

The notation of the scalar elliptic problem, the series of expansion of the solution and the eigen-pairs extraction method are presented in this chapter. The eigen-pairs of the  $3 - D$  scalar elliptic equation are constructed by the  $2 - D$  eigen-pairs, where the  $2 - D$  domain is perpendicular to the edge in the  $3 - D$  domain. Since the  $2 - D$  eigen-functions do not satisfy the  $3 - D$  equation, each  $2 - D$  eigen-function is followed by an infinite number of "shadow functions" such that each series of shadow functions and their related  $2 - D$  eigen-function satisfy the  $3 - D$  scalar elliptic problem. The  $2 - D$  eigen-pair extraction and the related shadow function extraction is presented in this chapter for five selected examples.

### 2.1 *Notations & Problems of Interest*

We choose a domain  $\Omega$  such that only one straight edge  $E$  is present. The domain is generated as the product  $\Omega = G \times I$  where  $I$  is the interval  $[-1, 1]$ , and  $G$  is a plane bounded sector of opening  $\omega \in (0, 2\pi]$  and radius 1 (the case of a crack,  $\omega = 2\pi$ , is included), as shown in Figure 2.1. Although any radius or interval  $I$  can be chosen, these simplified numbers have been chosen for simplicity of presentation.

The variables in  $G$  and  $I$  are  $(x_1, x_2)$  and  $x_3$  respectively, where the coordinates  $(x_1, x_2, x_3)$  are denoted by  $\mathbf{x}$ . Let  $(r, \theta)$  be the polar coordinates centered at the vertex of  $G$  so that  $G = \{(x_1, x_2) \in \mathbb{R}^2 \mid r \in (0, 1), \theta \in (0, \omega)\}$ . The edge  $E$  of

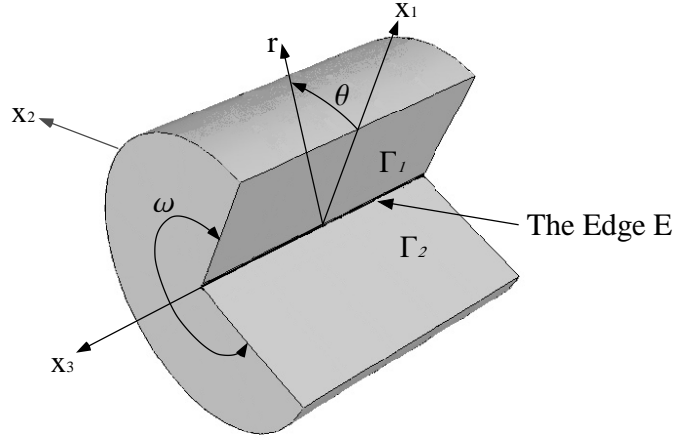


Fig. 2.1: Schematic domains of interest  $\Omega$ .

interest is the set  $\{\mathbf{x} \in \mathbb{R}^3 \mid r = 0, x_3 \in I\}$ . The two flat planes that intersect at the edge  $E$  are denoted by  $\Gamma_1$  and  $\Gamma_2$ . A part of a cylindrical surface is defined as follows:

$$\Gamma_R := \{\mathbf{x} \in \mathbb{R}^3 \mid r = R, \theta \in (0, \omega), x_3 \in I\},$$

The methods presented in this work are restricted to geometries where the edges are straight lines and the angle  $\omega$  is fixed along  $x_3$ .

The considered operator is a homogeneous elliptic second order partial differential operator  $L$  with constant real coefficients  $(k_{ij})$  of the form:

$$L = \sum_{i=1}^3 \sum_{j=1}^3 k_{ij} \partial_i \partial_j \quad \text{with} \quad \partial_1 = \frac{\partial}{\partial x_1}, \quad \partial_2 = \frac{\partial}{\partial x_2}, \quad \partial_3 = \frac{\partial}{\partial x_3}.$$

Where  $k_{ij} = k_{ji}$  form a symmetric matrix  $3 \times 3$  (for heat transfer problems these represent the heat conduction coefficients). The  $k_{ij}$ 's have to satisfy the ellipticity condition, and without loss of generality  $k_{33}$  is set as  $k_{33} = 1$ .

Denoting the solution by  $\tau(\mathbf{x})$ , we consider homogeneous Dirichlet boundary

conditions on  $\Gamma_1$  and  $\Gamma_2$ , i.e.:

$$\tau(r, 0, x_3) = \tau(r, \omega, x_3) = 0, \quad (2.1)$$

but all methods presented herein carry over to Neumann or mixed homogenous boundary conditions also.

For demonstration purposes three specific operators are considered - the Laplace operator,  $k_{ij} = \delta_{ij}$ , a general operator with  $k_{11} = 5, k_{22} = 4, k_{12} = -4$  and  $k_{13} = k_{23} = 0$  and a general operator having also mixed derivatives in the  $x_3$  direction with  $k_{11} = k_{22} = 1, k_{13} = -1$  and  $k_{12} = k_{23} = 0$ . Two domains are considered as benchmark domains, namely one having  $\omega = 3\pi/2$  and the other one is a cracked domain,  $\omega = 2\pi$ . Combination of the two different domains and three different operators provide five specific cases considered in this work according to Table 2.1.

Tab. 2.1: Notation of the various Cases considered as model problems

Case #	$\omega$	The Operator					
		$k_{11}$	$k_{22}$	$k_{33}$	$k_{12}$	$k_{13}$	$k_{23}$
Case 1	$3\pi/2$	1	1	1	0	0	0
Case 2	$3\pi/2$	5	4	1	-4	0	0
Case 3	$2\pi$	1	1	1	0	0	0
Case 4	$2\pi$	5	4	1	-4	0	0
Case 5	$2\pi$	1	1	1	0	-0.5	0

## 2.2 The Eigen-Pairs and computation of their Shadow Functions

The functional representation of the exact solution for the problem  $L(\tau) = 0$  in the neighborhood of the edge  $E$  relies on splitting the operator  $L$  into three parts (as shown in [10]):

$$L = M_0(\partial_1, \partial_2) + M_1(\partial_1, \partial_2)\partial_3 + M_2\partial_3^2 \quad (2.2)$$

The splitting allows consideration of a solution  $\tau$  of the form:

$$\tau = \sum_{j \geq 0} \partial_3^j A(x_3) \Phi_j(x_1, x_2) \quad (2.3)$$

The equation  $L(\tau) = 0$  after inserting (2.3) becomes:

$$\sum_{j \geq 0} \partial_3^j A(x_3) M_0 \Phi_j + \sum_{j \geq 0} \partial_3^{j+1} A(x_3) M_1 \Phi_j + \sum_{j \geq 0} \partial_3^{j+2} A(x_3) M_2 \Phi_j = 0 \quad (2.4)$$

and after rearranging:

$$\begin{aligned} & A(x_3) M_0 \Phi_0 + \partial_3^1 A(x_3) (M_0 \Phi_1 + M_1 \Phi_0) + \\ & + \sum_{j \geq 0} \partial_3^{j+2} A(x_3) (M_0 \Phi_{j+2} + M_1 \Phi_{j+1} + M_2 \Phi_j) = 0 \end{aligned} \quad (2.5)$$

Equation (2.5) has to hold for any smooth function  $A(x_3)$ . Thus, the functions  $\Phi_j$  must satisfy the three equations below:

$$\begin{cases} M_0 \Phi_0 = 0 \\ M_0 \Phi_1 + M_1 \Phi_0 = 0 \\ M_0 \Phi_{j+2} + M_1 \Phi_{j+1} + M_2 \Phi_j = 0, \quad j \geq 0 \end{cases} \quad (x_1, x_2) \in G \quad (2.6)$$

accompanied by homogeneous boundary conditions on the two faces  $\theta = 0, \omega$ .

The first partial differential equation in (2.6) generates the solution  $\Phi_0$  which is of the form

$$\Phi_0 = r^\alpha \varphi_0(\theta)$$

and is denoted as the two-dimensional primal leading function. It is nothing more than the singular solution associated with the eigen-value  $\alpha$  of the degenerated boundary value problem over the 2-D domain  $G$ . The second PDE in (2.6) with homogeneous Dirichlet boundary conditions generates the function  $\Phi_1$  which depends on  $\Phi_0$  and is of the form:

$$\Phi_1 = r^{\alpha+1} \varphi_1(\theta) \quad (2.7)$$

The sequence  $\Phi_j$  (where  $j \geq 2$ ) are the solutions of the third equation of (2.6) with Dirichlet boundary conditions. These are of the form:

$$\Phi_j = r^{\alpha+j} \varphi_j(\theta) \quad (2.8)$$

The  $\Phi_j$ , where  $j > 1$  are called the “shadow” functions associated with the primal leading function  $\Phi_0$ . There are an infinite number of functions  $\Phi_j$  associated with positive eigen-values  $\alpha_i$ , and therefore:

$$\Phi_j^{(\alpha_i)} = r^{\alpha_i+j} \varphi_j^{(\alpha_i)}(\theta) \quad j = 0, 1, \dots \quad (2.9)$$

Thus, for each eigen-value  $\alpha_i$ , the 3-D solution associated with it is:

$$\tau^{(\alpha_i)} = \sum_{j \geq 0} \partial_3^j A^{(\alpha_i)}(x_3) r^{\alpha_i+j} \varphi_j^{(\alpha_i)}(\theta) \quad (2.10)$$

and the overall solution  $\tau$  is:

$$\tau = \sum_{i \geq 1} \sum_{j \geq 0} \partial_3^j A^{(\alpha_i)}(x_3) r^{\alpha_i+j} \varphi_j^{(\alpha_i)}(\theta) \quad (2.11)$$

where  $A^{(\alpha_i)}(x_3)$  is the Edge Flux Intensity Function (EFIF) associated with the  $i^{\text{th}}$  eigen-value.

The solutions associated with the negative eigen-values are called the dual solutions, and are denoted by  $\Psi$ . For example,

$$\Psi_0^{(\alpha_i)} = c_0^{(\alpha_i)} r^{-\alpha_i} \psi_0^{(\alpha_i)}(\theta) \quad (2.12)$$

where  $\Psi_0^{(\alpha_i)}$  is the leading dual eigen-solution and

$$\Psi_j^{(\alpha_i)} = c_0^{(\alpha_i)} r^{-\alpha_i+j} \psi_j^{(\alpha_i)}(\theta) \quad (2.13)$$

are the shadow dual eigen-solutions. Theoretical details and a rigorous mathematical formulation is provided in [10].

## 2.2.1 Eigen-functions, their shadow functions and duals for cases 1-4

For the Cases 1 - 4 (Table 2.1) the operator  $L$  can be split as in (2.6) with:

$$M_0 = k_{11}\partial_1\partial_1 + 2k_{12}\partial_1\partial_2 + k_{22}\partial_2\partial_2, \quad M_1 = 0, \quad M_2 = 1 \quad (2.14)$$

Computing the primal and dual eigen-functions  $\Phi_0$  and  $\Psi_0$ :

$\Phi_0$  and  $\Psi_0$  are the solutions of the first equation in (2.6), where the operator is  $M_0$  on the plane domain  $G$ . A change of variables is performed:

$$\xi(x_1, x_2) = k_{11}^{-1} \sqrt{k_{11}k_{22} - k_{12}^2} x_1 \stackrel{\text{def}}{=} k_1 x_1 \quad (2.15)$$

$$\eta(x_1, x_2) = x_2 - k_{11}^{-1} k_{12} x_1 \stackrel{\text{def}}{=} x_2 - k_2 x_1, \quad (2.16)$$

so that  $M_0$  in the new variables is transformed into the Laplace operator:

$$\frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2}, \quad (2.17)$$

over a plane domain  $G'$ . The two straight lines defined by  $\theta = 0$  and  $\theta = \omega$  in the original domain  $G$  are transformed into the two lines defined by  $\gamma = \omega_1$  and  $\gamma = \omega_2$  in the transformed domain  $G'$ :

$$\omega_1 = \arctan\left(\frac{-k_2}{k_1}\right) \quad (2.18)$$

$$\omega_2 = \arctan\left(\frac{\tan(\omega) - k_2}{k_1}\right), \quad (2.19)$$

as illustrated in Figure 2.2. Since both  $\Phi_0$  and  $\Psi_0$  have to satisfy the homogeneous Dirichlet boundary conditions on  $\theta = 0, \omega$  in the original domain, these conditions become:

$$\Phi_0(\rho, \omega_1) = \Phi_0(\rho, \omega_2) = \Psi_0(\rho, \omega_1) = \Psi_0(\rho, \omega_2) = 0 \quad (2.20)$$

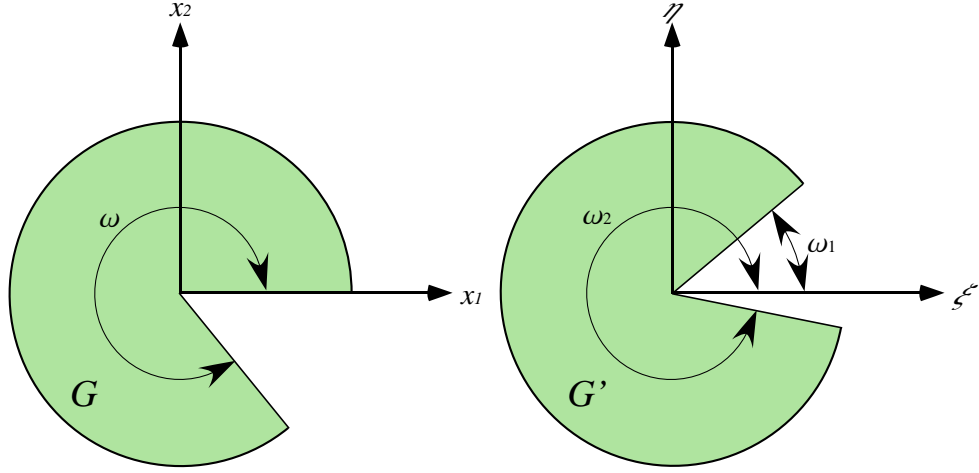


Fig. 2.2: The plane domain  $G$  before and after change of variables.

The solutions of the Laplace equation (by separation of variable) are:

$$\begin{cases} \Phi_0(\rho, \gamma) = \rho^\alpha [A \cos(\alpha\gamma) + B \sin(\alpha\gamma)] \\ \Psi_0(\rho, \gamma) = c_0 \rho^{-\alpha} [A \cos(\alpha\gamma) + B \sin(\alpha\gamma)] \end{cases} \quad (2.21)$$

where both  $+\alpha$  and  $-\alpha$  are eigen-values. The solutions associated with the positive eigen-values are denoted by  $\Phi$  whereas those associated with the negative eigen-values are denoted by  $\Psi$ . The value of the constant  $c_0$  is chosen so to satisfy a specific orthonormal condition as will be discussed next.

Requiring that the solutions satisfy the boundary conditions (2.20) results in the following conditions (the equations present are the equations for  $\Phi_0$  alone, although same equations are obtained for  $\Psi_0$ ):

$$\begin{pmatrix} \cos(\alpha\omega_1) & \sin(\alpha\omega_1) \\ \cos(\alpha\omega_2) & \sin(\alpha\omega_2) \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.22)$$

For a non-trivial solution,  $\alpha$  has to satisfy:

$$\alpha_i = \pm \frac{i\pi}{\omega_2 - \omega_1}, \quad i = 1, 2, \dots \quad (2.23)$$

The positive  $\alpha_i$ 's are associated with the  $\Phi_0$ 's and the negative with the  $\Psi_0$ 's. Since there are an infinite number of distinct  $\alpha_i$ 's, with each there is an associated  $\Phi_0^{(\alpha_i)}$  and  $\Psi_0^{(\alpha_i)}$ , and distinct  $A_i, B_i$ . The ratio  $A_i/B_i$  is given by:

$$A_i = -B_i \tan(\alpha_i \omega_1) = -B_i \tan(\alpha_i \omega_2)$$

The generic constant is omitted, as it will be added to the EFIF in the asymptotic expansion, and thus the reverse transformation of variables is performed and the functions  $\Phi_0^{(\alpha_i)}$  and  $\Psi_0^{(\alpha_i)}$  may be obtained in the polar coordinates  $r, \theta$  in the original  $G$ :

$$\Phi_0^{(\alpha_i)}(r, \theta) = r^\alpha \varphi_0^{(\alpha_i)}(\theta), \quad \Psi_0^{(\alpha_i)}(r, \theta) = c_0^{(\alpha_i)} r^{-\alpha_i} \psi_0^{(\alpha_i)}(\theta) \quad (2.24)$$

where

$$\begin{aligned} \varphi_0^{(\alpha_i)}(\theta) = & - \left( (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right)^{\alpha_i/2} \sec(\alpha_i \omega_1) \times \\ & \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (2.25)$$

and

$$\begin{aligned} \psi_0^{(\alpha_i)}(\theta) = & - \left( (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right)^{-\alpha_i/2} \sec(\alpha_i \omega_1) \times \\ & \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (2.26)$$

One may see that if the coefficients  $k_{ij}$  are chosen such that  $k_{ij} = \delta_{ij}$ , the operator  $L$  simplifies to the Laplace operator, having:

$$k_1 = 1, \quad k_2 = 0, \quad \omega_1 = 0, \quad \omega_2 = \omega.$$

And the eigen-functions and its dual eigen-functions are:

$$\begin{cases} \Phi_0^{(\alpha_i)}(r, \theta) = r^{\alpha_i} \varphi_0^{(\alpha_i)}(\theta) = r^{\alpha_i} \sin(\alpha_i \theta) \\ \Psi_0^{(\alpha_i)}(r, \theta) = c_0^{(\alpha_i)} r^{-\alpha_i} \psi_0^{(\alpha_i)}(\theta) = c_0^{(\alpha_i)} r^{-\alpha_i} \sin(\alpha_i \theta) \end{cases}, \quad \alpha_i = \pm \frac{i\pi}{\omega}$$

$\varphi_0^{(\alpha_i)} = \psi_0^{(\alpha_i)} = \sin(\alpha_i \theta)$  which are the well known expressions for the Laplace problem.

The value of the constant  $c_0^{(\alpha_i)}$  :

The value of the constant  $c_0^{(\alpha_i)}$  is chosen such that the primal and the dual eigenfunction,  $\Phi_0^{(\alpha_i)}$  and  $\Psi_0^{(\alpha_i)}$ , satisfy the orthonormal condition:

$$\int_0^w \left[ \left( T(R) \Phi_0^{(\alpha_i)} \right) \Psi_0^{(\alpha_i)} - \Phi_0^{(\alpha_i)} \left( T(R) \Psi_0^{(\alpha_i)} \right) \right] R d\theta = 1 \quad (2.27)$$

where  $T(R)$  is the Neumann trace operator related to the operator  $M_0$  :

$$\begin{aligned} T(R) = & \left( k_{11} \cos^2(\theta) + k_{12} \sin(2\theta) + k_{22} \sin^2(\theta) \right) \frac{\partial}{\partial r} + \\ & + \left( k_{12} \cos(2\theta) - \frac{1}{2}(k_{11} - k_{22}) \sin(2\theta) \right) \frac{1}{r} \frac{\partial}{\partial \theta} \end{aligned} \quad (2.28)$$

Further explanations on the  $T(R)$  trace operator and the connection to the  $J[R]$  integral are presented in Chapter 3. Further details about equation (2.27) are given in [10]. The value of the constant  $c_0^{(\alpha_i)}$ , is extracted from equation (2.27):

$$c_0^{(\alpha_i)} = \frac{1}{\int_0^w \left[ \left( T(R) \left( r^{(\alpha_i)} \phi_0^{(\alpha_i)} \right) \right) r^{(-\alpha_i)} \psi_0^{(\alpha_i)} - r^{(\alpha_i)} \phi_0^{(\alpha_i)} \left( T(R) \left( r^{(-\alpha_i)} \psi_0^{(\alpha_i)} \right) \right) \right] R d\theta} \quad (2.29)$$

One may note that for the Laplace operator  $k_{ij} = \delta_{ij}$  the Neumann trace operator simplifies to  $T = \frac{\partial}{\partial r}$  and the constant  $c_0$  is  $c_0^{(\alpha_i)} = \frac{1}{\alpha_i w}$ , which is the known coefficient of the dual eigen-function for a two dimensional domain. The explicit value of the constant  $c_0$  for cases 1-4 is computed and presented in Table 2.2.

*The odd shadow functions and the odd dual shadow functions:*

Once the primal eigen-function,  $\Phi_0^{(\alpha_i)}$ , is obtained, the first shadow function  $\Phi_1^{(\alpha_i)}$  may be calculated as the solution of the second equation in (2.6). Since the operator  $M_1$  does not exist, ( $M_1 = 0$ ), the differential equation is homogenous with homogeneous Dirichlet boundary conditions and therefore the first shadow function vanishes:

$$\Phi_1^{(\alpha_i)} = 0 \quad (2.30)$$

The sequence of odd shadow functions,  $\Phi_k^{(\alpha_i)}$  (where  $k = 3, 5, 7, \dots$ ), are calculated as the solution of the third equation in (2.6). For  $\Phi_3^{(\alpha_i)}$  we obtain:

$$M_0 \Phi_3^{(\alpha_i)} = -M_2 \Phi_1^{(\alpha_i)} = 0 \quad (2.31)$$

Once again the differential equation becomes homogeneous with Dirichlet boundary conditions since  $M_1 = 0$  and  $\Phi_1^{(\alpha_i)} = 0$ . For  $\Phi_k^{(\alpha_i)}$  we obtain:

$$M_0 \Phi_k^{(\alpha_i)} = -M_2 \Phi_{k-2}^{(\alpha_i)} = 0 \quad (2.32)$$

and therefore all odd shadow functions for any  $L$  (with  $k_{13} = k_{23} = 0$ ) vanish:

$$\Phi_k^{(\alpha_i)} = 0 \quad k = 3, 5, 7, \dots \quad (2.33)$$

Computing odd dual shadow functions,  $\Psi_k^{(\alpha_i)}$ , is along the lines of primal shadow functions, thus all odd dual shadow functions vanish for any  $L$  with  $k_{13} = k_{23} = 0$ :

$$\Psi_k^{(\alpha_i)} = 0 \quad k = 3, 5, 7, \dots \quad (2.34)$$

*The shadow function  $\Phi_2^{(\alpha_i)}$  and its dual  $\Psi_2^{(\alpha_i)}$ :*

The shadow function  $\Phi_2^{(\alpha_i)}$  and its dual  $\Psi_2^{(\alpha_i)}$  are the solution of the third equation in (2.6) with  $j = 0$ . It is a non-homogeneous differential equation over a two dimensional domain with Dirichlet boundary conditions. Its explicit form in polar coordinates  $\rho, \gamma$  is:

$$k_{11} k_1^2 \left( \frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2}{\partial \gamma^2} \right) \Phi_2^{(\alpha_i)} = \rho^{\alpha_i} [\tan(\alpha_i \omega_1) \cos(\alpha_i \gamma) - \sin(\alpha_i \gamma)] \quad (2.35)$$

The homogeneous solution is:

$$\Phi_2^{(\alpha_i)H} = \rho^f [A^H \cos(f\gamma) + B^H \sin(f\gamma)] \quad (2.36)$$

and the particular solution is:

$$\Phi_2^{(\alpha_i)P} = \frac{1}{4k_{11}k_1^2(\alpha_i + 1)} \rho^{\alpha_i+2} [\tan(\alpha_i\omega_1) \cos(\alpha_i\gamma) - \sin(\alpha_i\gamma)] \quad (2.37)$$

The particular solution identically satisfies the homogeneous Dirichlet boundary conditions so that the homogeneous solution must satisfy these:

$$\begin{cases} \Phi_2^{(\alpha_i)}(\rho, \omega_1) = \Phi_2^{(\alpha_i)H}(\rho, \omega_1) + \Phi_2^{(\alpha_i)P}(\rho, \omega_1) = \Phi_2^{(\alpha_i)H}(\rho, \omega_1) = 0 \\ \Phi_2^{(\alpha_i)}(\rho, \omega_2) = \Phi_2^{(\alpha_i)H}(\rho, \omega_2) + \Phi_2^{(\alpha_i)P}(\rho, \omega_2) = \Phi_2^{(\alpha_i)H}(\rho, \omega_2) = 0 \end{cases} \quad (2.38)$$

The coefficients  $A^H, B^H$  vanish so that  $\Phi_2$  is the particular solution alone. We may conclude that  $\Phi_2^{(\alpha_i)}$  is given by:

$$\Phi_2^{(\alpha_i)}(r, \theta) = r^{\alpha_i+2} \varphi_2^{(\alpha_i)}(\theta) \quad (2.39)$$

where:

$$\begin{aligned} \varphi_2^{(\alpha_i)}(\theta) = & \frac{-\sec(\alpha_i\omega_1)}{4k_{11}k_1^2(\alpha_i+1)} \left( (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right)^{\frac{\alpha_i+2}{2}} \times \\ & \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (2.40)$$

Computing  $\Psi_2^{(\alpha_i)}$  follows same arguments as for  $\Phi_2^{(\alpha_i)}$ . The explicit PDE is:

$$k_{11}k_1^2 \left( \frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2}{\partial \gamma^2} \right) \Psi_2^{(\alpha_i)} = c_0^{(\alpha_i)} \rho^{-\alpha_i} [\tan(\alpha_i\omega_1) \cos(\alpha_i\gamma) - \sin(\alpha_i\gamma)] \quad (2.41)$$

with the particular solution:

$$\Psi_2^{(\alpha_i)P} = \frac{c_0^{(\alpha_i)}}{4k_{11}k_1^2(1 - \alpha_i)} \rho^{-\alpha_i+2} [\tan(\alpha_i\omega_1) \cos(\alpha_i\gamma) - \sin(\alpha_i\gamma)] \quad (2.42)$$

Since the homogeneous solution vanishes when accounting for the boundary conditions, we conclude that  $\Psi_2^{(\alpha_i)}$  is:

$$\Psi_2^{(\alpha_i)}(r, \theta) = c_0^{(\alpha_i)} r^{-\alpha_i+2} \psi_2^{(\alpha_i)}(\theta) \quad (2.43)$$

where:

$$\begin{aligned} \psi_2(\theta)^{(\alpha_i)} &= \frac{-\sec(\alpha_i \omega_1)}{4k_{11}k_1^2(1-\alpha_i)} \left[ (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right]^{\frac{2-\alpha_i}{2}} \times \\ &\quad \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (2.44)$$

The shadow function  $\Phi_4^{(\alpha_i)}$  :

The shadow function  $\Phi_4^{(\alpha_i)}$  is generated by the third differential equation in (2.6) with  $j=2$ . The method of extracting  $\Phi_4^{(\alpha_i)}$  is very similar to the method of  $\Phi_2^{(\alpha_i)}$  extraction. The explicit form of the differential equation in  $\rho, \gamma$  coordinates is:

$$\begin{aligned} k_{11}k_1^2 \left( \frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2}{\partial \gamma^2} \right) \Phi_4^{(\alpha_i)} &= \\ = \frac{1}{4k_{11}k_1^2(\alpha_i+1)} \rho^{\alpha_i+2} [\tan(\alpha_i \omega_1) \cos(\alpha_i \gamma) - \sin(\alpha_i \gamma)] \end{aligned} \quad (2.45)$$

The solution of  $\Phi_4^{(\alpha_i)}$  is based on the particular solution alone since the homogeneous solution vanishes under the homogeneous Dirichlet boundary conditions. The shadow function  $\Phi_4^{(\alpha_i)}$  in  $r, \theta$  polar coordinates is:

$$\Phi_4^{(\alpha_i)}(r, \theta) = r^{\alpha_i+4} \varphi_4^{(\alpha_i)}(\theta) \quad (2.46)$$

Where:

$$\begin{aligned} \varphi_4^{(\alpha_i)}(\theta) &= \frac{\sec(\alpha_i \omega_1)}{32k_{11}^2k_1^4(\alpha_i+1)(\alpha_i+2)} \left( (k_1^2 + k_2^2) \cos^2(\theta) + \sin^2(\theta) - k_2 \sin(2\theta) \right)^{\frac{\alpha_i+4}{2}} \times \\ &\quad \times \sin \left( \alpha_i \left( \arctan \left( \frac{\sin(\theta) - k_2 \cos \theta}{k_1 \cos \theta} \right) - \omega_1 \right) \right) \end{aligned} \quad (2.47)$$

The specific cases (1-4) eigen-functions:

The eigen-functions and the dual eigen-functions are defined by the eigen-value,  $\alpha_i$ , the operator representative coefficients,  $k_1$  and  $k_2$  (equation (2.15), (2.16)), the

representative coefficients of the domain,  $\omega_1$  and  $\omega_2$  (equation (2.18), (2.19)), and the representative coefficient of the dual solution,  $c_0^{(\alpha_i)}$ . The coefficients of the selected cases, related with the first eigen-value are presented in Table 2.2.

Tab. 2.2: Main coefficients of the considered cases

Case #	$k_1$	$k_2$	$\omega_1$	$\omega_2$	$\alpha_1$	$c_0^{(\alpha_1)}$
Case 1	1.0	0.0	0	$1.5\pi$	$2/3$	0.31831
Case 2	0.4	-0.8	$0.3524\pi$	$-0.5\pi$	0.87139	0.26903
Case 3	1.0	0.0	0	$2.0\pi$	0.5	0.31831
Case 4	0.4	-0.8	$0.3524\pi$	$2.3524\pi$	0.5	0.23533

We provide in Figures 2.3-2.6 the graphical representation of the primal and dual eigen-functions  $\varphi_0^{(\alpha_1)}, \varphi_2^{(\alpha_1)}, \varphi_4^{(\alpha_1)}, \psi_0^{(\alpha_1)}, \psi_2^{(\alpha_1)}$  for Cases 1-4.

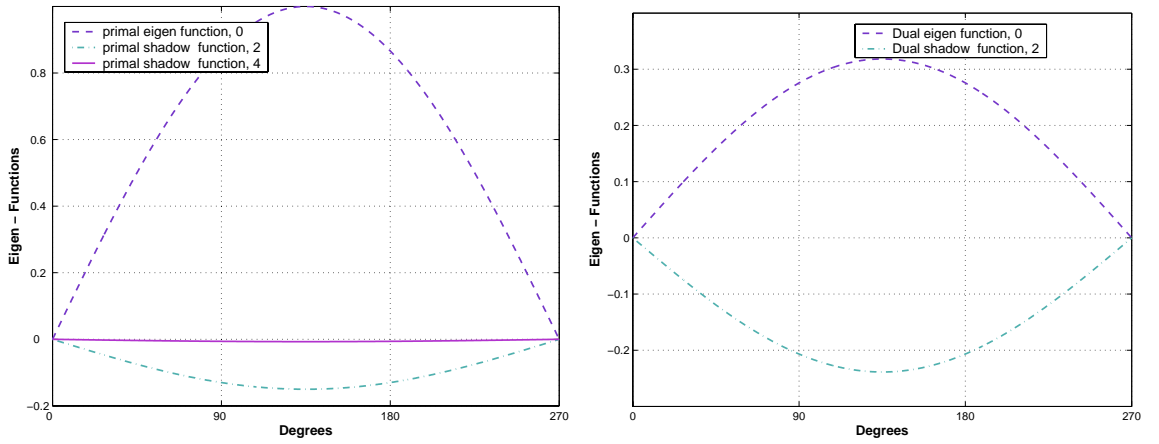


Fig. 2.3: The eigen-functions and dual eigen-functions associated with the first eigen-value,  $\alpha_1$ , for case 1.

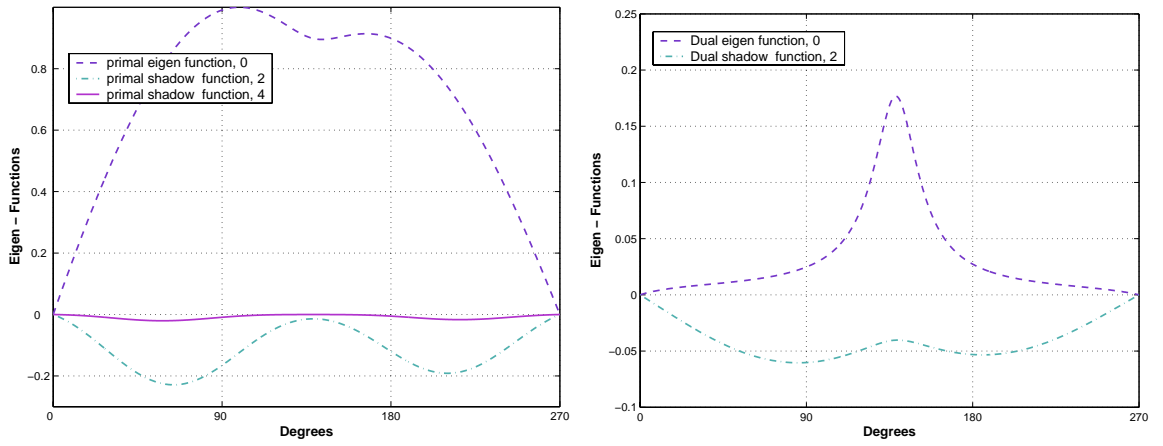


Fig. 2.4: The eigen-functions and dual eigen-functions associated with the first eigen-value,  $\alpha_1$ , for case 2.

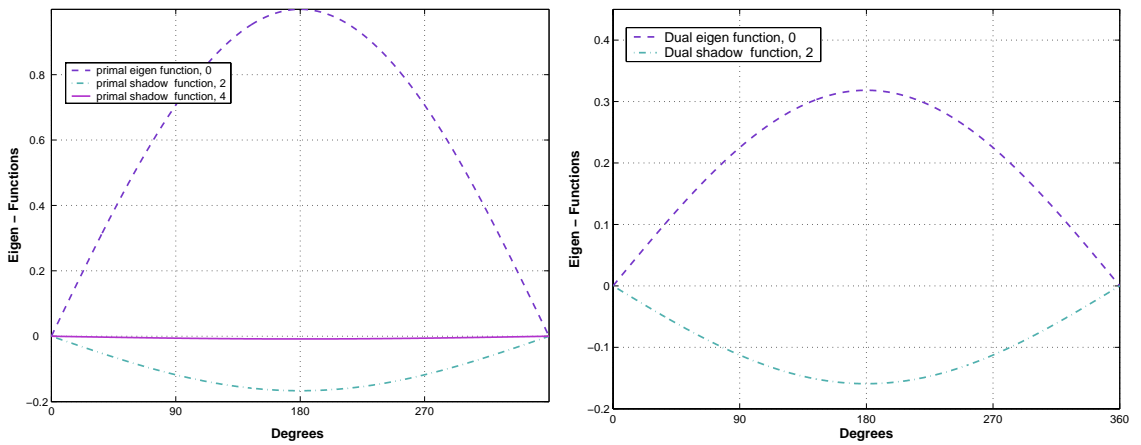


Fig. 2.5: The eigen-functions and dual eigen-functions associated with the first eigen-value,  $\alpha_1$ , for case 3.

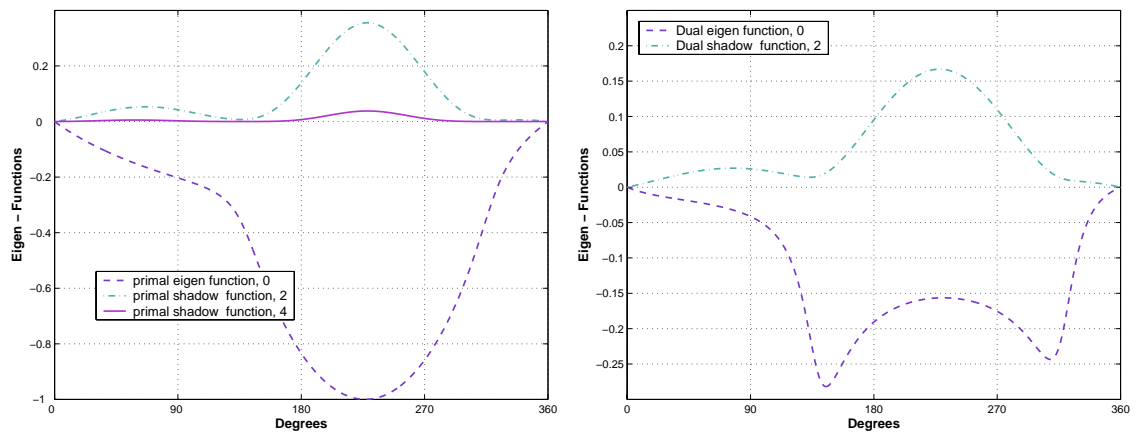


Fig. 2.6: The eigen-functions and dual eigen-functions associated with the first eigen-value,  $\alpha_1$ , for case 4.

## 2.2.2 Eigen-functions, their shadow functions and duals for case 5

The operator  $L$  for case 5, is:

$$L = \partial_1 \partial_1 + \partial_2 \partial_2 - \partial_1 \partial_3 + \partial_3 \partial_3$$

with:

$$M_0 = \partial_1 \partial_1 + \partial_2 \partial_2 \quad M_1 = -\partial_1 \quad M_2 = 1$$

For this case  $\Phi_0^{(\alpha_i)}$  and  $\Psi_0^{(\alpha_i)}$  according to the first equation of system (2.6) are:

$$\Phi_0^{(\alpha_i)}(r, \theta) = r^{\alpha_i} \sin(\alpha_i \theta) \quad (2.48)$$

$$\Psi_0^{(\alpha_i)}(r, \theta) = c_0^{(\alpha_i)} r^{-\alpha_i} \sin(\alpha_i \theta) \quad (2.49)$$

with  $\alpha_i = \frac{i\pi}{\omega}$ ,  $i = 1, 2, \dots$ . The Neumann trace operator for case 5 is simply  $T = \frac{\partial}{\partial r}$  and therefore the coefficient of the dual solution is  $c_0^{(\alpha_i)} = \frac{1}{\alpha_i \omega}$ .

The shadow function  $\Phi_1^{(\alpha_i)}$  is computed by the second differential equation of the system (2.6) and the shadow functions  $\Phi_2^{(\alpha_i)}$  and  $\Phi_3^{(\alpha_i)}$  are computed by the third differential equation of (2.6) with  $j = 0$  and  $j = 1$  respectively.

$$\Phi_1^{(\alpha_i)}(r, \theta) = r^{\alpha_i+1} \left( \frac{1}{4} \sin((\alpha_i - 1)\theta) + \frac{1}{4} d_1^{(\alpha_i)} \sin((\alpha_i + 1)\theta) \right) \quad (2.50)$$

$$\begin{aligned} \Phi_2^{(\alpha_i)}(r, \theta) = & r^{\alpha_i+2} \left( \frac{1}{32} \sin((\alpha_i - 2)\theta) + \frac{(d_1^{(\alpha_i)} \alpha_i + d_1^{(\alpha_i)} - 3)}{16(\alpha_i + 1)} \sin(\alpha_i \theta) + \right. \\ & \left. d_2^{(\alpha_i)} \sin((\alpha_i + 2)\theta) \right) \end{aligned} \quad (2.51)$$

$$\begin{aligned} \Phi_3^{(\alpha_i)}(r, \theta) = & r^{\alpha_i+3} \left( \frac{1}{384} \sin((\alpha_i - 3)\theta) + d_3^{(\alpha_i)} \sin((\alpha_i + 3)\theta) \right. \\ & + \frac{(-6 + d_1^{(\alpha_i)} + d_1^{(\alpha_i)} \alpha_i)}{128(\alpha_i + 1)} \sin((\alpha_i - 1)\theta) \\ & \left. + \frac{1}{64} \left( 16d_2^{(\alpha_i)} - \frac{(3 + 3d_1^{(\alpha_i)} - 3d_1^{(\alpha_i)} \alpha_i)}{(\alpha_i + 1)(\alpha_i + 2)} \right) \sin((\alpha_i + 1)\theta) \right) \end{aligned} \quad (2.52)$$

Where  $d_1^{(\alpha_i)}$ ,  $d_2^{(\alpha_i)}$  and  $d_3^{(\alpha_i)}$  are determined by the homogeneous Dirichlet boundary conditions as:

$$\begin{cases} w = i\pi & d_1^{(\alpha_i)} = 0, \\ w \neq i\pi & d_1^{(\alpha_i)} = -\frac{\sin((\alpha_i-1)w)}{\sin((\alpha_i+1)w)} \end{cases} \quad (2.53)$$

$$\begin{cases} w = i\pi/2 & d_2^{(\alpha_i)} = 0, \\ w \neq i\pi/2 & d_2^{(\alpha_i)} = -\frac{\sin((\alpha_i-2)w)}{32\sin((\alpha_i+2)w)} \end{cases} \quad (2.54)$$

$$\begin{cases} w = i\pi/3 & d_3^{(\alpha_i)} = 0, \\ w \neq i\pi/3 & d_3^{(\alpha_i)} = -\left(\frac{1}{384}\sin((\alpha_i-3)w) + \frac{1}{64}\left(16d_2^{(\alpha_i)} - \frac{(3+3d_1^{(\alpha_i)}-3d_1^{(\alpha_i)}\alpha_i)}{(\alpha_i+1)(\alpha_i+2)}\right)\sin((\alpha_i+1)w) + \frac{(-6+d_1^{(\alpha_i)}+d_1^{(\alpha_i)}\alpha_i)}{128(\alpha_i+1)}\sin((\alpha_i-1)w)\right) \times \frac{1}{\sin((\alpha_i+3)w)} \end{cases} \quad (2.55)$$

The dual shadow function  $\Psi_1^{(\alpha_i)}$  is computed by the second equation of the system (2.6) with Dirichlet boundary conditions. The shadow functions  $\Psi_2^{(\alpha_i)}$  and  $\Psi_3^{(\alpha_i)}$  are computed by the third differential equation of the system (2.6) with  $j = 0$  and  $j = 1$  respectively.

$$\Psi_1^{(\alpha_i)}(r, \theta) = r^{-\alpha_i+1} \left( \frac{c_0^{(\alpha_i)}}{4} \sin((\alpha_i+1)\theta) + \frac{c_1^{(\alpha_i)}}{4} \sin((\alpha_i-1)\theta) \right) \quad (2.56)$$

$$\begin{aligned} \Psi_2^{(\alpha_i)}(r, \theta) = & r^{-\alpha_i+2} \left( \frac{c_0^{(\alpha_i)}}{16} \sin((\alpha_i+2)\theta) + \frac{(-3c_0^{(\alpha_i)} + c_1^{(\alpha_i)} - c_1^{(\alpha_i)}\alpha_i)}{16(1-\alpha_i)} \sin(\alpha_i\theta) + \right. \\ & \left. \frac{c_2^{(\alpha_i)}}{16} \sin((\alpha_i-2)\theta) \right) \end{aligned} \quad (2.57)$$

$$\begin{aligned}
\Psi_3^{(\alpha_i)}(r, \theta) &= r^{-\alpha_i+3} \left( \frac{c_0^{(\alpha_i)}}{192} \sin((\alpha_i + 3)\theta) + \frac{(5c_0^{(\alpha_i)} - c_1^{(\alpha_i)} + c_1^{(\alpha_i)}\alpha)}{128(\alpha_i - 1)} \sin((\alpha_i + 1)\theta) \right. \\
&\quad \left. + \frac{1}{64} \left( c_2^{(\alpha_i)} - \frac{3(c_0^{(\alpha_i)} + c_1^{(\alpha_i)} - c_1^{(\alpha_i)}\alpha)}{(\alpha_i - 2)(\alpha_i - 1)} \right) \sin((\alpha_i - 1)\theta) \right. \\
&\quad \left. + c_3^{(\alpha_i)} \sin((\alpha_i - 3)\theta) \right) \tag{2.58}
\end{aligned}$$

where  $c_1^{(\alpha_i)}$ ,  $c_2^{(\alpha_i)}$  and  $c_3^{(\alpha_i)}$  are determined by the homogeneous Dirichlet boundary conditions as:

$$\begin{cases} w = i\pi & c_1^{(\alpha_i)} = 0, \\ w \neq i\pi & c_1^{(\alpha_i)} = -\frac{c_0^{(\alpha_i)} \sin((\alpha_i+1)w)}{\sin((\alpha_i-1)w)} \end{cases} \tag{2.59}$$

$$\begin{cases} w = i\pi/2 & c_2^{(\alpha_i)} = 0, \\ w \neq i\pi/2 & c_2^{(\alpha_i)} = -\frac{c_0^{(\alpha_i)} \sin((\alpha_i+2)w)}{\sin((\alpha_i-2)w)} \end{cases} \tag{2.60}$$

$$\begin{cases} w = i\pi/3 & c_3^{(\alpha_i)} = 0, \\ w \neq i\pi/3 & c_3^{(\alpha_i)} = \left\{ -\left( \frac{c_0^{(\alpha_i)}}{192} \sin((\alpha_i + 3)\omega) + \frac{(5c_0^{(\alpha_i)} - c_1^{(\alpha_i)} + c_1^{(\alpha_i)}\alpha)}{128(\alpha_i - 1)} \sin((\alpha_i + 1)\omega) \right. \right. \\ &\quad \left. \left. + \frac{1}{64} \left( c_2^{(\alpha_i)} - \frac{3(c_0^{(\alpha_i)} + c_1^{(\alpha_i)} - c_1^{(\alpha_i)}\alpha)}{(\alpha_i - 2)(\alpha_i - 1)} \right) \sin((\alpha_i - 1)\omega) \right) \right\} \\ &\quad \neq \left\{ c_3^{(\alpha_i)} \sin((\alpha_i - 3)\theta) \right\} \end{cases} \tag{2.61}$$

Figures 2.7 presents the eigen-functions, their shadows and their duals for case 5.

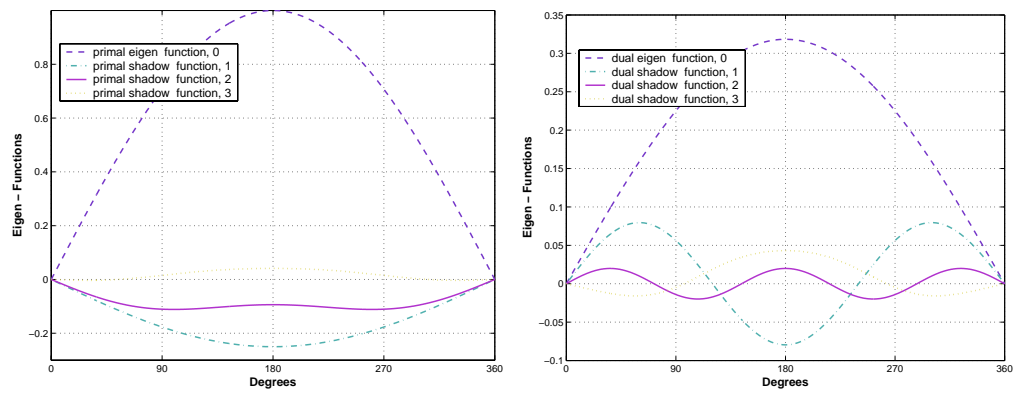


Fig. 2.7: The eigen-functions and dual eigen-functions associated with the first eigen-value,  $\alpha_2$ , for case 5.

### 3. THE QUASI-DUAL FUNCTION METHOD FOR EXTRACTING EFIFS, AND THE $J[R]$ INTEGRAL

At this point, the eigen-functions  $\varphi_j^{(\alpha_i)}$  and the dual eigen functions  $\psi_j^{(\alpha_i)}$  are specified. In order to extract the EFIFs, an orthogonal relation between the functions  $\varphi_j^{(\alpha_i)}$  and  $\psi_j^{(\alpha_i)}$  will be established in this chapter. This orthogonal relation is necessary for the introduction of a special surface integral, denoted by  $J[R]$ , which is an extension of the path independent integral of 2 –  $D$  domains.

For each eigen-value  $\alpha_i$ , a set of *quasi-dual* singular functions  $K_m^{(\alpha_i)}[B]$  are constructed where  $m$  is a natural integer called the *order* of the quasi-dual function, and  $B(x_3)$  is a function (we choose it to be a polynomial) called *extraction polynomial*.

$$K_m^{(\alpha_i)}[B] \stackrel{\text{def}}{=} \sum_{j=0}^m \partial_3^j B(x_3) \Psi_j^{(\alpha_i)} \quad (3.1)$$

By using the quasi-dual functions, one can extract a scalar product of  $A^{(\alpha_i)}(x_3)$  with  $B(x_3)$  on  $E$ . This is accomplished with the help of the *anti-symmetric* boundary integral  $J[R]$ , over the surface  $\Gamma_R$  shown in 3.1. We define  $J[R](u, v)$  to be:

$$J[R](u, v) \stackrel{\text{def}}{=} \int_{\Gamma_R} (Tu \cdot v - u \cdot Tv) dS = \int_I \int_0^w (Tu \cdot v - u \cdot Tv)|_{r=R} R d\theta dx_3 \quad (3.2)$$

where  $I \equiv E$  (the edge) along  $x_3$  axis (Figure 2.1).  $T$  is the radial Neumann trace operator related to the operator  $L$ :

$$T \stackrel{\text{def}}{=} \left\{ \left( \begin{array}{ccc} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & 1 \end{array} \right) \left( \begin{array}{c} \partial_1 \\ \partial_2 \\ \partial_3 \end{array} \right) \right\}^T \left( \begin{array}{c} \cos(\theta) \\ \sin(\theta) \\ 0 \end{array} \right) \quad (3.3)$$

The Neumann trace operator in (2.27) is a  $2 - D$  trace operator where the trace operator presented in (3.3) is a  $3 - D$  operator.

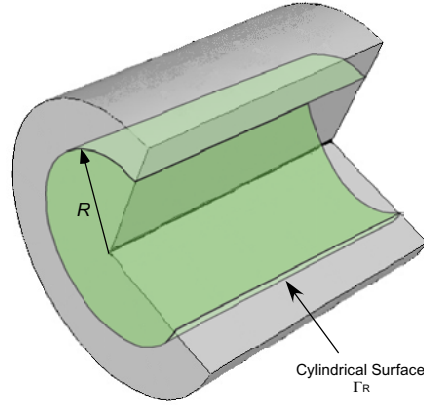


Fig. 3.1: Cylindrical surface,  $\Gamma_R$ .

With the above definition we have the following theorem:

Theorem 3.0.2.1: Take  $B(x_3)$  such that

$$\partial_3^j B(x_3) = 0 \quad \text{for } j = 0, \dots, m - 1 \quad \text{on } \partial I \quad (3.4)$$

then

$$J[R](\tau, K_m^{(\alpha_i)}[B]) = \int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3 + \mathcal{O}(R^{m+1}), \quad \text{as } R \rightarrow 0, \quad (3.5)$$

The proof of the theorem in equations (3.4) and (3.5) can be found in [10]. Equation (3.2) relates between any two functions  $u$  and  $v$ . When replacing those functions with the solution  $\tau$  and the quasidual singular function  $K_m^{(\alpha_i)}[B]$ , relation (3.2) allows an approximation of  $\int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3$ . The accuracy of the approximation depends on the radius of integration surface,  $R$ , and on the order of quasidual function,  $m$ . The integer  $m$  determines the number of homogeneous derivatives required of the extraction polynomial,  $B(x_3)$  on  $\partial I$  as defined in equation (3.4). As shown in

equation (3.1) the integer  $m$  also represents the number of shadow functions in the quasidual singular function  $K_m^{(\alpha_i)}[B]$  and therefore if one chooses to increase the accuracy of (3.5) by increasing  $m$ , one has to increase the number of shadow functions as well as the number of homogeneous derivatives of  $B(x_3)$ . A precise determination of  $\int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3$  can be obtained by computing (3.5) for two or three  $R$  values and Richardson's extrapolation as  $R \rightarrow 0$ .

### 3.1 The quasi-dual extraction functions.

As  $m$  increases, more dual shadow functions are required to represent  $K_m^{(\alpha_i)}[B]$ . Consider the following options of the quasidual extraction functions:

$$K_0^{(\alpha_1)} = B_0(x_3)\Psi_0^{(\alpha_1)}(r, \theta) \quad (3.6)$$

$$K_1^{(\alpha_1)} = B_1(x_3)\Psi_0^{(\alpha_1)}(r, \theta) + \partial_3 B_1(x_3)\Psi_1^{(\alpha_1)}(r, \theta) \quad (3.7)$$

$$K_2^{(\alpha_1)} = B_2(x_3)\Psi_0^{(\alpha_1)}(r, \theta) + \partial_3 B_2(x_3)\Psi_1^{(\alpha_1)}(r, \theta) + \partial_3^2 B_2(x_3)\Psi_2^{(\alpha_1)}(r, \theta) \quad (3.8)$$

and:

$$\begin{aligned} K_3^{(\alpha_1)} = & B_3(x_3)\Psi_0^{(\alpha_1)}(r, \theta) + \partial_3 B_3(x_3)\Psi_1^{(\alpha_1)}(r, \theta) \\ & + \partial_3^2 B_3(x_3)\Psi_2^{(\alpha_1)}(r, \theta) + \partial_3^3 B_3(x_3)\Psi_3^{(\alpha_1)}(r, \theta) \end{aligned} \quad (3.9)$$

According to the theorem above, the difference between  $J[R](\tau, K_m^{(\alpha_i)}[B])$  and  $\int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3$  should be of an order of  $\mathcal{O}(R^{m+1})$ , which is the convergence rate with respect to  $R$ . To obtain the "right" convergence rate, the following conditions should be satisfied for the extraction polynomials, according to equation (3.4):

$$\text{For } B_0 : \quad \text{No condition required.} \quad (3.10)$$

$$\text{For } B_1 : \quad B_1(+1) = B_1(-1) = 0. \quad (3.11)$$

$$\text{For } B_2 : \quad B_2(+1) = B_2(-1) = \partial_3 B_2(+1) = \partial_3 B_2(-1) = 0. \quad (3.12)$$

$$\text{For } B_3 : \quad B_3(+1) = B_3(-1) = \partial_3 B_3(+1) = \partial_3 B_3(-1) = \partial_3^2 B_3(+1) = \partial_3^2 B_3(-1) = 0. \quad (3.13)$$

Since  $B_0$  does not have to satisfy any condition, we chose  $B_0(x_3) = 1$ . In order to check how strict are the conditions on the  $B$ s in (3.12), we used also  $B(x_3)$  functions which do not strictly satisfy (3.12) condition. So we chose  $B_2(x_3) = x_3^2 - 1$  in addition to the right choice of  $B_2(x_3) = (x_3^2 - 1)^2$ .

In general the exact solution  $\tau$  is unknown, so we use instead a finite element approximation  $\tau_{FE}$ . Thus, the integral (3.2) is performed numerically using a Gaussian quadrature of order  $nG$ : (Note that due to the change in integral limits from  $0 \leq \theta \leq \omega$  to  $-1 \leq \tilde{\theta} \leq 1$ , the factor  $\omega/2$  is added).

$$J[R](\tau, K_m^{(\alpha_i)}[B]) = \sum_{k=1}^{nG} \sum_{\ell=1}^{nG} \frac{\omega}{2} w_k w_\ell (T\tau_{FE} \cdot K_m^{(\alpha_i)}[B] - \tau_{FE} \cdot TK_m^{(\alpha_i)}[B])_{\xi_k, \eta_\ell} \quad (3.14)$$

where  $w_k$  are the weights and  $\xi_k$  and  $\eta_\ell$  are the abscissas of a Gaussian quadrature. The Neumann trace operator,  $T$ , operates on both  $\tau$  and  $K_m^{(\alpha_i)}[B]$  in (3.14). For  $T\tau$  we use the numerical approximations  $T\tau_{FE}$  computed by finite elements. We extract in the post-solution phase of the FE analysis  $\tau_{FE}$ ,  $\partial_1 \tau_{FE}$ ,  $\partial_2 \tau_{FE}$ , and  $\partial_3 \tau_{FE}$  whereas  $TK_m^{(\alpha_i)}[B]$  is computed analytically. These values are evaluated at the specific Gaussian points when the integral is computed numerically.

The numerical errors associated with the numerical integration and with replacing the exact solution by the finite element solution are negligible, as will be shown for one example problem in Section 3.3.

### 3.2 Analytical Solutions for Validating the $J[R]$ Theorem

We generate herein analytical solutions against which our numerical experimentations are compared. The exact solution associated with the  $i$ -th eigen-pair,  $\tau_{ex}^{(\alpha_i)}$  is:

$$\tau_{ex}^{(\alpha_i)} = \sum_{j \geq 0} \partial_3^j A^{(\alpha_i)}(x_3) \Phi_j^{(\alpha_i)}(r, \theta) \quad (3.15)$$

So if  $A^{(\alpha_i)}(x_3)$  is a polynomial of order  $2N - 1$ , i.e.

$$A^{(\alpha_i)}(x_3) = a_0^{(\alpha_i)} + a_1^{(\alpha_i)} x_3 + \cdots + a_{2N-1}^{(\alpha_i)} x_3^{2N-1}$$

then (3.15) has a finite number of terms in the sum, because the  $2N$  and higher derivatives are zero. Thus, (3.15) becomes:

$$\tau_{ex}^{(\alpha_i)} = \sum_{j=0}^{2N-1} \partial_3^j A^{(\alpha_i)}(x_3) \Phi_j^{(\alpha_i)}(r, \theta) \quad (3.16)$$

We may specify over the entire boundary of our domain a Dirichlet boundary conditions according to (3.16). This implies that the solution at any point  $r, \theta, x_3$  is exactly (3.16). We prescribed two kinds of boundary conditions, each having a different  $N$ . The first BC, which is denoted by  $BC_2$  is the one for which we take

$$A^{(\alpha_1)}(x_3) = 1 + x_3 + x_3^2$$

i.e.,  $a_0^{(\alpha_1)} = a_1^{(\alpha_1)} = a_2^{(\alpha_1)} = 1$ , and  $a_j^{(\alpha_1)} = 0$  for all  $j \geq 3$ . This means that on the boundary ( $\partial\Omega$  as shown in 2.1) we prescribe the Dirichlet  $BC_2$ :

$$\tau_{ex}^{(\alpha_1)}|_{\partial\Omega} = (1 + x_3 + x_3^2) \Phi_0^{(\alpha_1)}(r, \theta) + (1 + 2x_3) \Phi_1^{(\alpha_1)}(r, \theta) + 2\Phi_2^{(\alpha_1)}(r, \theta) \quad (3.17)$$

The second boundary condition which we consider is denoted by  $BC_4$  for which we take

$$A^{(\alpha_1)}(x_3) = a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 + a_4 x_3^4$$

This means that on the boundary we have the Dirichlet  $BC_4$ :

$$\begin{aligned}
 \tau_{ex}^{(\alpha_1)}|_{\partial\Omega} &= (a_0 + a_1x_3 + a_2x_3^2 + a_3x_3^3 + a_4x_3^4)\Phi_0^{(\alpha_1)}(r, \theta) \\
 &+ (a_1 + 2a_2x_3 + 3a_3x_3^2 + 4a_4x_3^3)\Phi_1^{(\alpha_1)}(r, \theta) \\
 &+ (2a_2 + 6a_3x_3 + 12a_4x_3^2)\Phi_2^{(\alpha_1)}(r, \theta) \\
 &+ (6a_3 + 24a_4x_3)\Phi_3^{(\alpha_1)}(r, \theta) + 24\Phi_4^{(\alpha_1)}(r, \theta)
 \end{aligned} \tag{3.18}$$

The functions  $\Phi_j^{(\alpha_1)}(r, \theta)$  (depend of course on  $\omega$  and  $k_{ij}$ ) are so created as to satisfy identically the differential equation. Therefore, if we specify (3.17) on the boundary then the exact solution is the same function shown in (3.17) when evaluated at each  $\mathbf{x}$ . The same is true for the function in (3.18). This means that our exact solution contains only one edge singularity (no vertex singularities).

### 3.2.1 Discretization of the model by p-FEM

The domains have been discretized by using a p-FEM mesh, with geometrical progression towards the singular edge with a factor of 0.15, having 4 layers of elements. In the  $x_3$  direction, a uniform discretization using 5 elements has been adopted. There are 160 3-D elements (Hexahedral and Pentahedral elements), over which the polynomial degree, using the trunk space (see Szabo and Babuška in [26]) has been increased from 1 to 8.

The opening angle has been assign as a parameter, and in Figure 3.2 we present the meshes used for opening angles of  $\omega = 3\pi/2$  and  $\omega = 2\pi$  (crack).

### 3.2.2 Numerical Tests Using $K_0^{(\alpha_1)}$ for Case 1 – 5

When using the quasi-dual function  $K_0^{(\alpha_1)}$ , with any chosen  $B_0(x_3)$ , the convergence of  $J[R]$  to  $J[0] \stackrel{\text{def}}{=} J_{ex}$ , according to the theorem presented in the previous section should be  $\mathcal{O}(R^2)$  for cases 1-4 (for which  $\Phi_1 = \Psi_1 = 0$ ), and  $\mathcal{O}(R)$  for case 5.

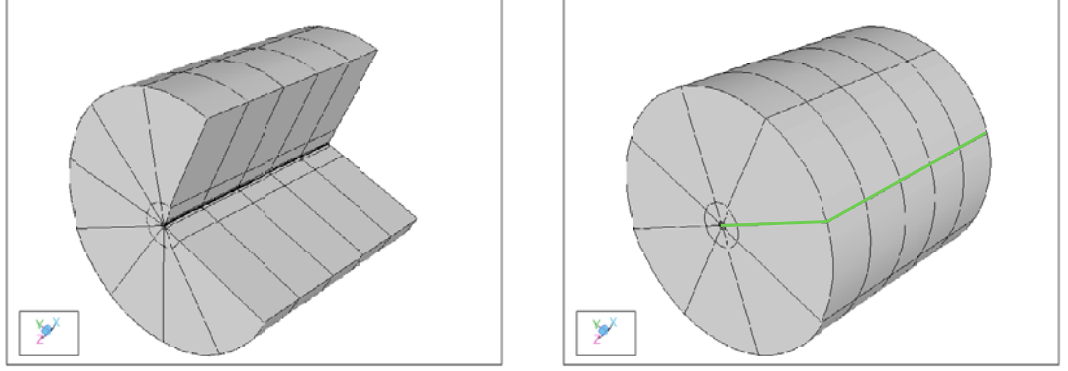


Fig. 3.2: The p-FEM models.

We perform numerical tests for Cases 1-5 taking  $BC_2$  boundary conditions, and computing  $J[R]$  at different values of  $R$ .

According to equation (3.5), we have:

$$\begin{aligned} J[R](\tau, K_m^{(\alpha_i)}[B]) &\stackrel{\text{def}}{=} \int_{\Gamma_R} (T\tau \cdot K_m^{(\alpha_i)} - \tau \cdot TK_m^{(\alpha_i)})_{r=R} R d\theta \\ &= \int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3 + \mathcal{O}(R^{m+1}) \end{aligned} \quad (3.19)$$

so that:

$$J[R](\tau, K_m^{(\alpha_i)}[B])_{R \rightarrow 0} = \int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3 \equiv J_{ex} \quad (3.20)$$

For example, for  $A(x_3) = 1 + x_3 + x_3^2$  and  $K_0^{(\alpha_1)}$  with  $B(x_3) = 1$ :

$$J_{ex} = \int_{-1}^1 (1 + x_3 + x_3^2) \cdot 1 \cdot dx_3 = 2 + \frac{2}{3} = 2\frac{2}{3}$$

Because  $\phi_1^{(\alpha_1)} = \psi_1^{(\alpha_1)} = 0$  for cases 1-4 for example, then  $K_0^{(\alpha_1)} = K_1^{(\alpha_1)}$  so that:

$$J[R] = J_{ex} + \mathcal{O}(R^2) \quad (3.21)$$

and

$$\log(J[R] - J_{ex}) = 2 \log(R) \quad (3.22)$$

We summarize the results in Table 3.1, and plot  $\log(J_{ex} - J[R])$  vis.  $\log(R)$  in Figure 3.3. The  $J[R]$  computation uses 10 integration points for the quadric in both  $\theta$  and  $x_3$  directions with  $p = 6$  in the finite element analysis - It was found that taking 32 integration points and  $p = 8$  does not improve the results considerably.

Tab. 3.1: Values of  $J[R]/J_{ex}$ ,  $BC_2$ , using  $K_0^{(\alpha_1)}$ .  $B_0(x_3) = 1$ , and  $J_{ex} = 8/3$ .

	Case 1	Case 2	Case 3	Case 4	Case5
R = 0.9	0.544627	0.857214	0.393094	0.704199	0.299646
R = 0.8	0.639754	0.886932	0.519466	0.764039	0.430013
R = 0.7	0.724304	0.913497	0.632346	0.819377	0.547270
R = 0.6	0.797786	0.936658	0.730629	0.868391	0.651167
R = 0.5	0.859434	0.955926	0.812665	0.907785	0.742218
R = 0.4	0.909614	0.971456	0.879250	0.939072	0.820100
R = 0.3	0.949535	0.984096	0.932802	0.967113	0.884557
R = 0.2	0.977493	0.992844	0.970071	0.983999	0.936454
R = 0.1	0.994679	0.998349	0.993188	0.996935	0.974619

It is easily visible in Figure 3.3 that the convergence rate in cases 1-4 is  $\mathcal{O}(R^2)$ , whereas the converge rate of case 5 is higher then  $\mathcal{O}(R)$  but lower then  $\mathcal{O}(R^2)$ , as expected.

### 3.2.3 Numerical Tests Using $K_0^{(\alpha_1)} - K_3^{(\alpha_1)}$ for Case 5

Taking  $K_0^{(\alpha_1)}$  up to  $K_m^{(\alpha_1)}$ , where the highest value of  $m$  is 3, with the “proper” choice of  $B(x_3)$ ’s, the convergence of  $J[R]$  to the exact value should be  $\mathcal{O}(R^{m+1})$  for Case 5. We performed numerical tests taking  $BC_2$  boundary conditions, and computing  $J[R]$  at different values of  $R$ . We summarize the results in Table 3.2, and plot  $\log(J_{ex} - J[R])$  vis.  $\log(R)$  in Figure 3.4. We can see in Figure 3.4 that the convergence rate of  $J[R]$  in Case 5 and  $K_0^{(\alpha_1)} - K_3^{(\alpha_1)}$  is of order  $R^{m+1}$  as we expected.

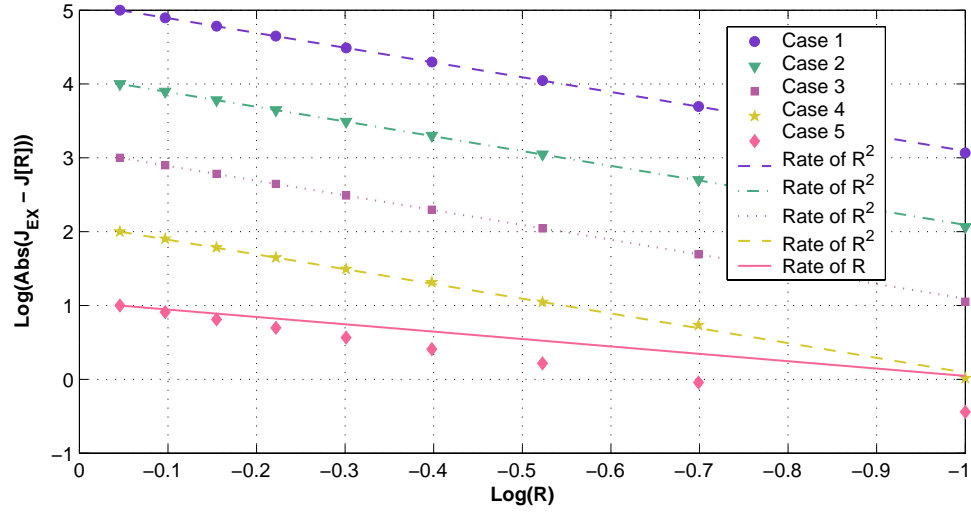


Fig. 3.3: Converge rates of  $J[R]$ , using  $BC_2$ ,  $K_0^{(\alpha_1)}$ ,  $B_0(x_3) = 1$ .

### 3.2.4 Numerical Results Using $K_2^{(\alpha_1)}$ for Case 1 – 4

Using the quasi-dual function  $K_2^{(\alpha_1)}$ , with the “proper” choice of  $B_2(x_3)$  satisfying (3.12) ( $B_2(x_3) = (x_3 - 1)^2$ ), the convergence of  $J[R]$  to the exact value should be  $\mathcal{O}(R^4)$  for Cases 1-4. The question is: Will the convergence be slower for a  $B_2(x_3)$  which does not satisfy (3.12)? We do this test for Cases 1-4 taking  $BC_4$  boundary conditions. We summarize the results in Table 3.3 and in Figure 3.5. The Gauss integration uses 10 points in both  $\theta$  and  $x_3$  directions with  $p = 6$  in the finite element analysis. It was found that taking 32 integration points and  $p = 8$  does not improve the results considerably. Figure 3.5 illustrates the convergence rate of  $J[R]$  when  $B_2(x_3)$  does not satisfy (3.12), ( $B_2(x_3) = x_3^2 - 1$ ). As we can see, the convergence rate is slower than the convergence rate of  $J[R]$  when a  $B_2(x_3)$  that satisfies (3.12) is selected.

Tab. 3.2: Values of  $J[R]/J_{ex}$ , for case 5 with  $BC_2$ .

$B_m(x_3)$	$K_0^{\alpha_1}$	$K_1^{\alpha_1}$	$K_2^{\alpha_1}$	$K_3^{\alpha_1}$
$J_{ex}$	1	$x_3^2 - 1$	$(x_3^2 - 1)^2$	$(x_3^2 - 1)^3$
	8/3	-8/5	128/105	-64/63
R = 0.9	0.299646	0.294691	1.493344	4.476667
R = 0.8	0.430013	0.436022	1.378059	3.255974
R = 0.7	0.547270	0.563480	1.274364	2.368930
R = 0.6	0.651167	0.675921	1.185997	1.762052
R = 0.5	0.742218	0.772991	1.115421	1.377856
R = 0.4	0.820100	0.853594	1.063079	1.158648
R = 0.3	0.884557	0.916867	1.028120	1.051144
R = 0.2	0.936454	0.963079	1.009058	1.010512
R = 0.1	0.974619	0.990599	1.001099	1.000582
R = 0.08	0.980910	0.994130	1.000710	1.000370
R = 0.06	0.986374	0.996604	1.000221	1.000044

It may be seen that the converge rate indeed is  $\mathcal{O}(R^{m+1})$  for  $B_m(x_3)$  that satisfy (3.12).

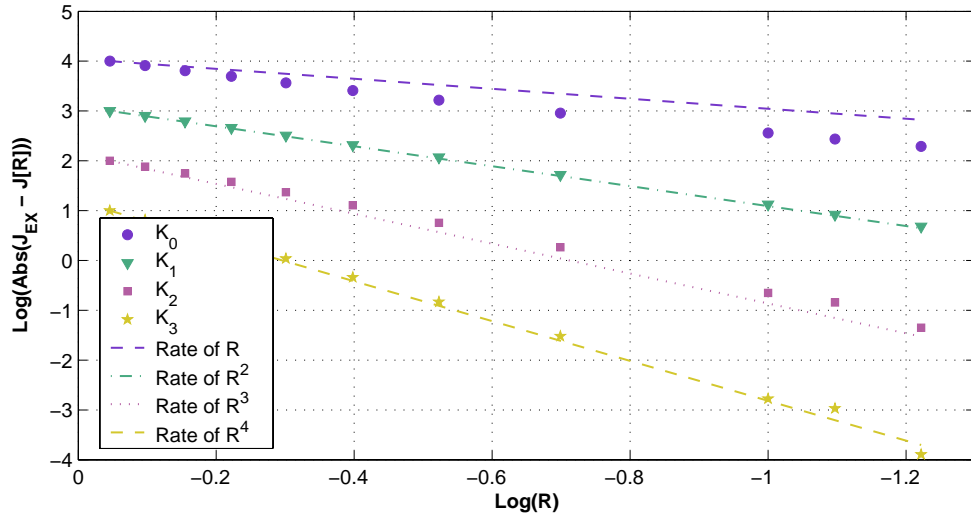


Fig. 3.4: Converge rates of  $J[R]$ , for case 5 with  $BC_2$  and  $K_m^{\alpha_1}$ , as presented in Table 3.2.

Tab. 3.3: Values of  $J[R]/J_{ex}$ ,  $BC_2$ , using  $K_2^{(\alpha_1)}$

$B_2(x_3)$	Case 1		Case 2	
	$x_3^2 - 1$	$(x_3^2 - 1)^2$	$x_3^2 - 1$	$(x_3^2 - 1)^2$
$J_{ex}$	-1.6	1.219047	-1.6	1.219047
R = 0.9	-0.887038	0.999645	0.291835	0.999721
R = 0.8	-0.431005	1.000026	0.478194	1.000058
R = 0.7	-0.053998	1.000175	0.626690	1.000132
R = 0.6	0.252628	1.000142	0.742754	1.000076
R = 0.5	0.496215	0.999906	0.830931	0.999930
R = 0.4	0.685064	0.999680	0.896410	0.999740
R = 0.3	0.826890	1.000238	0.944198	1.000027
R = 0.2	0.924078	0.999921	0.975774	0.999825
R = 0.1	0.981498	1.000289	0.994182	1.000084

$B_2(x_3)$	Case 3		Case 4	
	$x_3^2 - 1$	$(x_3^2 - 1)^2$	$x_3^2 - 1$	$(x_3^2 - 1)^2$
$J_{ex}$	-1.6	1.219047	-1.6	1.219047
R = 0.9	-1.296983	0.999666	-0.128968	0.999040
R = 0.8	-0.771699	0.999908	0.128063	0.998552
R = 0.7	-0.325337	1.000267	0.348874	0.999061
R = 0.6	0.046995	1.000450	0.533524	0.999781
R = 0.5	0.349228	0.999949	0.681029	0.998962
R = 0.4	0.588407	0.999356	0.796871	0.997832
R = 0.3	0.772011	1.000448	0.888854	1.000358
R = 0.2	0.899388	0.999967	0.949504	0.998503
R = 0.1	0.975646	1.000655	0.988488	1.000678

It may be seen that as expected the convergence is  $\mathcal{O}(R^2)$  for  $B(x_3)$  that does not satisfy (3.12), but much higher for  $B(x_3)$  which does!

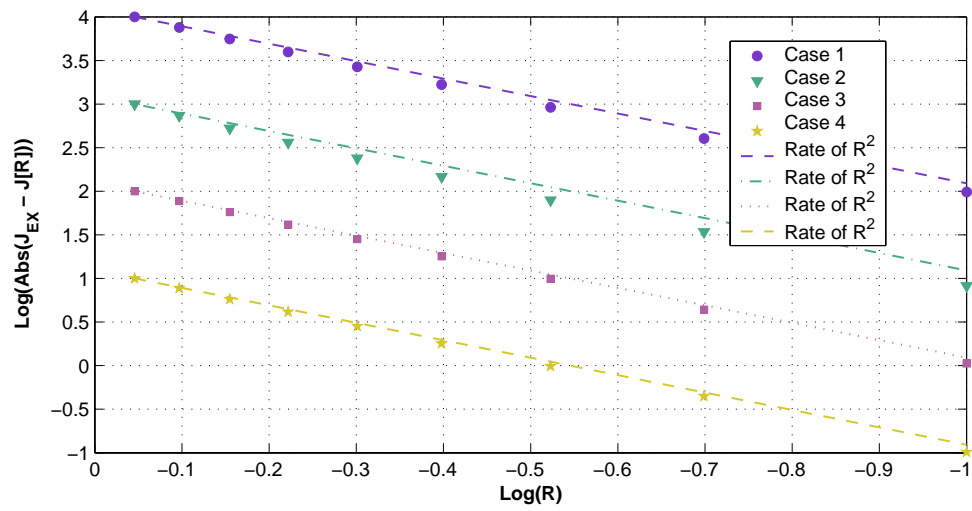


Fig. 3.5: Converge rates of  $J[R]$  for Cases 1-4, using  $BC_4$ ,  $K_2^{(\alpha_1)}$ ,  $B_2(x_3) = (x_3 - 1)^2$  as presented in Table 3.3.

### 3.3 Numerical errors due to the numerical integration and finite element solution.

The integral  $J[R]$  is computed by a Gaussian quadrature, and using the approximated finite element solution instead of the exact solution. These incorporate numerical inaccuracies in our computations which should be controlled and bounded. We use case 5 in this Section to quantify the level of numerical errors, and demonstrate that these are negligible.

#### 3.3.1 Errors due to finite element approximation

By using a finite element solution as an approximation of the exact solution, a numerical error is included in our computations. In order to evaluate the influence of the error, we compute  $J[R]$  at different polynomial degrees of the test and trial functions of the finite elements. The results are summarized in Table 3.4. The results summarized in Table 3.4 are the values of  $J[R]/J_{ex}$  with  $BC_2$  ( $A^{\alpha_1}(x_3) = 1 + x_3 + x_3^2$ ) and  $K_0 - K_3$ , with "proper"  $B(x_3)$ , for case 5. The values of  $J[R]/J_{ex}$  are not influenced by the polynomial degree of the finite element approximation when using a high degree polynomial (5'th degree or higher).

#### 3.3.2 Errors due to the Gauss quadrature computation

The second source of numerical error is due to the Gauss quadrature used to evaluate the integral  $J[R]$ . One needs to evaluate a double integral over  $\theta$  and  $x_3$  (equation (3.2)). The Gaussian quadrature order  $nG$  controls the numerical error in this case. In order to evaluate the influence of the quadrature order,  $J[R]$  was computed with various quadrature orders, and the results are presented in Table 3.5. The results presented in Table 3.4 are the values of  $J[R]/J_{ex}$  with  $BC_2$  ( $A^{\alpha_1}(x_3) = 1 + x_3 + x_3^2$ ) and  $K_0 - K_3$ , with "proper"  $B(x_3)$ , for case 5. The values of  $J[R]/J_{ex}$  are not

Tab. 3.4: Values of  $J[R]/J_{ex}$ , using case 5 with  $BC_2$  and  $R = 0.02$

$B_{(n)}(x_3)$ $J_{ex}$	1 8/3	$(x_3 - 1)$ -8/5	$(x_3 - 1)^2$ 128/105	$(x_3 - 1)^3$ -64/63
$P = 8$	0.996084	0.999703	1.000080	1.000068
$P = 7$	0.995804	0.999439	0.999819	0.999805
$P = 6$	0.995966	0.999646	1.000032	1.000016
$P = 5$	0.997801	1.001348	1.001692	1.001663
$P = 4$	0.994830	0.998490	0.998890	0.998882
$P = 3$	0.989424	0.995263	0.996286	0.996391

As may be observed, the values  $J[R]$  are not influenced by the numerical errors of the finite element approximation when using a high polynomial degree in the finite element analysis.

influenced by the quadrature computation even at low order as 10.

Tab. 3.5: Values of  $J[R]/J_{ex}$ , using case 5 with  $BC_2$  and  $R = 0.02$

$B_{(n)}(x_3)$ $J_{ex}$	1 8/3	$(x_3 - 1)$ -8/5	$(x_3 - 1)^2$ 128/105	$(x_3 - 1)^3$ -64/63
$N = 10$	0.996092	0.999705	1.000079	1.000064
$N = 15$	0.996084	0.999703	1.000080	1.000068
$N = 32$	0.996085	0.999701	1.000077	1.000064

As may be observed, the quadrature order has a negligible numerical error even at a low order as 10.

A relation between the solution,  $\tau$ , and the quasi-dual function  $K_m^{(\alpha_i)}[B]$  was established via the  $J[R]$  integral in this Chapter. The convergence rate of the internal boundary surface integral,  $J[R]$ , is of  $\mathcal{O}(R^{m+1})$  where  $m$  is an integer defining the number of shadow functions in the quasi-dual function and the number of homogeneous derivatives of the extraction polynomial,  $B(x_3)$ . The integral,  $J[R]$ , was examined for all five cases with several quasi-dual functions where the results show that the convergence rate of  $J[R]$  is as expected. Moreover, if the number of shadow functions for constructing  $K_q^{(\alpha_i)}[B]$  is  $q$  and the number of homogeneous derivatives of  $B(\pm 1)$  is  $n$ , where  $q \neq n$ , the convergence rate will be the minimum between  $\mathcal{O}(R^{q+1})$  and  $\mathcal{O}(R^{n+1})$ .

#### 4. COMPUTING EFIFS USING A "SIMPLE" FAMILY OF EXTRACTION POLYNOMIALS

The surface integral  $J[R]$ , which was established in Chapter 3, approximates the integration of the EFIF,  $A^{(\alpha_i)}(x_3)$  multiplied by the extraction function  $B(x_3)$ ,  $(\int_I A^{(\alpha_i)}(x_3) B(x_3) dx_3)$ . We are interested in extracting the EFIF with the help of known values of  $J[R]$ , computed with various numbers of  $B(x_3)$  functions. Since the functional representation of the EFIF is unknown, we would be interested in it's polynomial approximation and therefore the problem of extracting the EFIF is reduced to extracting the coefficients of the polynomial representation of the EFIF.

In order to extract a polynomial approximation of the EFIF of the degree  $N$ ,  $N + 1$  extractions of  $J[R]$  are performed for  $N + 1$  different  $B(x_3)$  polynomials. Then, a system of equation is constructed where the solution of the system will provide  $N + 1$  coefficients of the  $N$ 'th -degree polynomial which approximates the EFIF.

A special family of extraction polynomials is constructed in this Chapter such that each  $B(x_3)$  provides an EFIF coefficient. (The system of equation produced of the family of polynomials is a unity matrix.) This family of polynomials is called "*simple*" family of polynomials and the  $B(x_3)$ 's polynomials of the family satisfy equation (3.4) with  $m = 2$ .

We use cases 1-4 from this point on to demonstrate the method and evaluate it's performance.

#### 4.1 Constructing a "simple" family of extraction polynomials

$$B_2^{(k)}(x_3)$$

Any function  $A^{(\alpha_i)}(x_3)$  can be approximated by a polynomial of a given order, and for practical purposes, we assume that a polynomial of order nine will suffice for approximating the EFIF for any practical consideration:

$$A^{(\alpha_i)}(x_3) \approx a_0^{(\alpha_i)} + a_1^{(\alpha_i)}x_3 + \cdots + a_9^{(\alpha_i)}x_3^9 \quad (4.1)$$

Extracting the ten coefficient  $a_0^{(\alpha_i)} \cdots a_9^{(\alpha_i)}$  is realized by using equation (3.5). Therefore, we are required to compute  $J[R]$  by using a given number,  $k$ , of quasidual functions and a family of extraction polynomials  $B(x_3)$ , that satisfy equation (3.4).

To extract the ten coefficients  $a_0^{(\alpha_i)} \cdots a_9^{(\alpha_i)}$ , we construct ten extraction polynomials denoted by the index  $k$ ,  $B_m^{(k)}(x_3)$ , with the following conditions:

- The polynomials  $B_m^{(k)}(x_3)$  have to satisfy:

$$\partial_3^{m-1} B_m^{(k)}(\pm 1) = 0$$

In particular, for  $m = 2$ :

$$B_2^{(k)}(\pm 1) = \partial_3 B_2^{(k)}(\pm 1) = 0 \quad (4.2)$$

- The polynomials  $B_m^{(k)}(x_3)$  are orthonormal under integration from  $-1$  to  $1$  with respect to any polynomial of order less or equal to nine: i.e.

$$\int_{-1}^1 B_2^{(k)}(x_3) x_3^j dx_3 = \delta_{kj}, \quad j, k = 0, 1, \cdots, 9. \quad (4.3)$$

For example, using the quasi-dual extraction function  $K_2^{(\alpha_1)} = B_2^{(k)}(x_3)\Psi_0^{(\alpha_1)}(r, \theta) + \partial_3^2 B_2^{(k)}(x_3)\Psi_2^{(\alpha_1)}(r, \theta)$  (note that the term associated with  $\partial_3 B_2^{(k)}(x_3)$  is not included

because  $\Psi_1^{(\alpha_1)}(r, \theta) = 0$  for cases 1-4), and in view of (4.3) we obtain by using the ten different  $B_2^{(k)}$ ,  $k = 0, 1, \dots, 9$  a set of ten equations of the form:

$$J[R](\tau_{FE}, K_{2k}^{(\alpha_1)}) = (a_k^{(\alpha_1)})_{FE} + \mathcal{O}(R^4)$$

We denote by 2 indexes the quasi-dual function  $K_{mk}^{(\alpha_i)}$ . The index  $m$  denotes the number of shadow functions in the quasi-dual function and the number of homogeneous derivatives of the extraction function  $B_2^{(k)}(x_3)$  where the index  $k$  denotes which extraction function is used. Notice that  $(a_k^{(\alpha_1)})_{FE}$  has a numerical error because we use the finite element approximation of  $\tau_{ex}$  when computing  $J[R](\tau_{FE}, K_{mk}^{(\alpha_i)})$ . This numerical error can be monitored by ensuring that the error in the energy norm is small and using the finite element solution away from the first row of elements adjacent to the singular edge.

One may choose in addition  $\partial_3^2 B(\pm 1) = 0$  and obtain 16 constrains on  $B$  (Equations 4.2, 4.3) so that by choosing  $B_2^{(k)}(x_3)$  as  $B_2^{(k)} = b_0 + b_1 x_3 + b_2 x_3^2 + \dots + b_{15} x_3^{15}$  we are now able to construct the "simple" extraction family of polynomials:

$$\begin{aligned}
B_2^{(0)}(x_3) &= 3.866243 - 99.23357x_3^2 + 774.021x_3^4 - 2764.36x_3^6 \\
&\quad + 5221.57x_3^8 - 5411.45x_3^{10} + 2913.85x_3^{12} - 638.273x_3^{14} \\
B_2^{(1)}(x_3) &= 240.995x_3 - 4386.12x_3^3 + 28196.5x_3^5 - 88766.7x_3^7 \\
&\quad + 153324x_3^9 - 148606x_3^{11} + 75954.5x_3^{13} - 15956.8x_3^{15} \\
B_2^{(2)}(x_3) &= -87.6348 + 4048.73x_3^2 - 37595.3x_3^4 + 146204x_3^6 \\
&\quad - 290509x_3^8 + 311366x_3^{10} - 171723x_3^{12} + 38296.4x_3^{14} \\
B_2^{(3)}(x_3) &= -3663.13x_3 + 79367.9x_3^3 - 555575x_3^5 + 1839893x_3^7 \\
&\quad - 3286647x_3^9 + 3262744x_3^{11} - 1697808x_3^{13} + 361688x_3^{15} \\
B_2^{(4)}(x_3) &= 499.518 - 27473.5x_3^2 + 277787x_3^4 - 1136404x_3^6 \\
&\quad + 2335249x_3^8 - 2563585x_3^{10} + 1439446x_3^{12} - 325519x_3^{14} \\
B_2^{(5)}(x_3) &= 16484.1x_3 - 388903x_3^3 + 2863740x_3^5 - 9808048x_3^7 \\
&\quad + 17945096x_3^9 - 18137022x_3^{11} + 9570277x_3^{13} - 2061624x_3^{15} \\
B_2^{(6)}(x_3) &= -999.037 + 59831.2x_3^2 - 636386x_3^4 + 2692405x_3^6 \\
&\quad - 5666872x_3^8 + 6333563x_3^{10} - 3606191x_3^{12} + 824649x_3^{14} \\
B_2^{(7)}(x_3) &= -28084.0x_3 + 696995x_3^3 - 5307885x_3^5 + 18619723x_3^7 \\
&\quad - 34683799x_3^9 + 35546745x_3^{11} - 18966944x_3^{13} + 4123248x_3^{15} \\
B_2^{(8)}(x_3) &= 638.273 - 40211.2x_3^2 + 442323x_3^4 - 1916736x_3^6 \\
&\quad + 4107292x_3^8 - 4654930x_3^{10} + 2680111x_3^{12} - 618487x_3^{14} \\
B_2^{(9)}(x_3) &= 15956.8x_3 - 409559x_3^3 + 3194560x_3^5 - 11409144x_3^7 \\
&\quad + 21550606.37x_3^9 - 22334264x_3^{11} + 12026142x_3^{13} \\
&\quad - 2634297.897x_3^{15}
\end{aligned} \tag{4.4}$$

$B_2^{(k)}$  can be used with any  $K_{mk}^{(\alpha_i)}$  where  $m \leq 2$ , so  $B_2^{(k)}$  can also be chosen with

$K_{0k}^{(\alpha_i)}$  since it satisfies equation (3.4):

$$K_{0k}^{(\alpha_i)} = B_2^{(k)} \Psi_0^{(\alpha_i)}$$

#### 4.2 Numerical results - using $BC_4$ with $K_{0k}^{(\alpha_1)}$ - Influence of $R$

For the benchmark problem with  $BC_4$  with  $a_0 = 5, a_1 = 4, a_2 = 9, a_3 = 3, a_4 = 1$  and  $a_5 = \dots = a_9 = 0$ , and using  $B_2^{(0)}(x_3) \dots B_2^{(9)}(x_3)$  in (4.4), and  $K_{0k}^{(\alpha_1)}$ , we extracted the polynomial representation of the EFIF for case 4, for example, at different values of  $R$ . We expect to obtain a  $\mathcal{O}(R^2)$  convergence rate in the computed values. Using 15 integration points and  $p = 8$  in the finite element analysis we extract the coefficients shown in Table 4.1. (We performed the computation of  $J[R](\tau_{FE}, K_{0k}^{(\alpha_1)})$  10 times for  $k = 0, 1, \dots, 9$ ).

Tab. 4.1: Values of extracted  $a_i^{(\alpha_1)}$ ,  $BC_4$ , using  $K_{0k}^{(\alpha_1)}$  at different  $R$ s

	Exact	$R = 0.9$	$R = 0.5$	$R = 0.2$	$R = 0.05$
$a_0^{(\alpha_1)}$	5	2.222990	4.078715	4.852051	4.991024
$a_1^{(\alpha_1)}$	4	0.915898	3.049524	3.851433	3.990502
$a_2^{(\alpha_1)}$	9	6.946990	8.369218	8.904111	8.975718
$a_3^{(\alpha_1)}$	3	2.976952	2.990601	2.987099	2.991134
$a_4^{(\alpha_1)}$	1	0.985236	0.998325	0.984476	1.034880
$a_5^{(\alpha_1)}$	0	0.131439	0.053104	0.049990	0.008539
$a_6^{(\alpha_1)}$	0	0.032395	-0.001769	0.045420	0.048412
$a_7^{(\alpha_1)}$	0	-0.274521	-0.128380	-0.098008	0.027731
$a_8^{(\alpha_1)}$	0	-0.026893	0.000553	-0.031504	-0.074369
$a_9^{(\alpha_1)}$	0	0.171007	0.080450	0.053618	-0.045685

Using the  $a_i$ s we compute the relative error (%) between the extracted EFIF compared to the exact EFIF, as present in Figure 4.1. We can see in Figure 4.1 that when decreasing values of  $R$  are selected for the extraction of the EFIF, the EFIF approximation converges toward the exact EFIF, as expected. In this case (of  $K_{0k}^{(\alpha_1)}$ )

a good approximation of the EFIF is received when extracted with  $R = 0.2$  or lower.

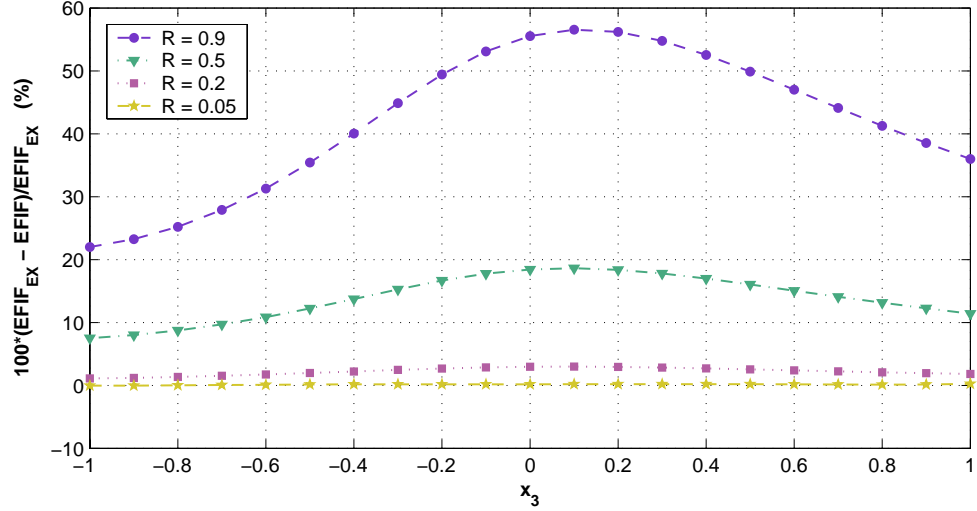


Fig. 4.1: Relative error (%) of the extracted EFIF using  $K_{0k}^{(\alpha_1)}$  and the "simple" family  $B_2^{(k)}$  at various  $R$ 's.  $A^{(\alpha_1)}(x_3) = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ .

To better visualize the convergence behavior of order 2 in  $R$ , we use Richardson's extrapolation for  $a_0$  and  $a_1$ , with the convergence of order 2 to obtain the values as  $R \rightarrow 0$ . Table 4.2 shows the Richardson's extrapolation table. Table 4.2 shows that one can extract the  $a_i$ 's away from the edge and use these to obtain very good approximation at  $R \rightarrow 0$ .

### 4.3 Numerical results - using $BC_4$ with $K_{2k}^{(\alpha_1)}$ - Influence of $R$

We consider exactly the same case as before, only that this time, we extract the EFIF taking  $K_{2k}^{(\alpha_1)}$ . For this case, we expect to obtain higher convergence rate with respect to  $R$ :  $\mathcal{O}(R^4)$  in the computed values. The extracted values are shown in Table 4.3.

Tab. 4.2: Richardson's extrapolation for  $a_0^{(\alpha_1)}$  and  $a_1^{(\alpha_1)}$  with  $\mathcal{O}(R^2)$ 

$a_0^{(\alpha_1)}$				$a_0^{(\alpha_1)}$ exact
$R = 0.9$	2.22299046	4.90716304	5.00414219	5.0
$R = 0.5$	4.07871471	4.99935310		
$R = 0.2$	4.85205096			
$a_1^{(\alpha_1)}$				$a_1^{(\alpha_1)}$ exact
$R = 0.9$	0.91589819	4.00203619	4.00428926	4.0
$R = 0.5$	3.04952446	4.00417799		
$R = 0.2$	3.85143343			

Again, to better visualize the convergence behavior of order 4 in  $R$ , we use Richardson's extrapolation for  $a_0$  and  $a_1$ , with the convergence of order 4 to obtain the values at  $R \rightarrow 0$ . Table 4.4 shows the Richardson's extrapolation table. It is easy to see that by using Richardson's extrapolation with  $a_i$  values, extracted away from the edge, we obtain very good approximation of the EFIF coefficients at  $R \rightarrow 0$ .

#### 4.4 Numerical results - using $BC_4$ with $K_{2k}^{(\alpha_1)}$ for cases 1-4

Although we extract the EFIF for four different Cases, the boundary conditions of all cases is  $BC_4$  and therefore we expect to obtain the same EFIF approximation for all four cases.

For the benchmark problem with  $BC_4$ , having  $a_0 = 5, a_1 = 4, a_2 = 9, a_3 = 3, a_4 = 1$ , and using  $B_2^{(0)}(x_3) \cdots B_2^{(9)}(x_3)$  in (4.4), and  $K_{2k}^{(\alpha_1)}$ , we extracted the polynomial representation of the EFIF. Again we expect to obtain the values above. Using 10 integration points and  $p = 6$  in the finite element mesh, at  $R = 0.05$  we extract  $a_0, a_1, \dots, a_9$  for the cases 1-4. The extracted  $a_i$ 's are summarized in Table 4.5.

Tab. 4.3: Values of extracted  $a_i^{(\alpha_1)}$ ,  $BC_4$ , using  $K_{2k}^{(\alpha_1)}$  at different  $R$ s

	Exact	$R = 0.9$	$R = 0.5$	$R = 0.2$	$R = 0.05$
$a_0^{(\alpha_1)}$	5	5.922487	5.089213	5.005745	5.000573
$a_1^{(\alpha_1)}$	4	4.066124	4.013264	4.005760	3.999830
$a_2^{(\alpha_1)}$	9	9.025389	9.017465	9.013898	8.982407
$a_3^{(\alpha_1)}$	3	1.475587	2.760273	2.956454	2.996282
$a_4^{(\alpha_1)}$	1	0.498056	0.820542	0.906252	1.024553
$a_5^{(\alpha_1)}$	0	8.226022	1.167258	0.167910	-0.026804
$a_6^{(\alpha_1)}$	0	1.782512	0.477125	0.225434	0.080168
$a_7^{(\alpha_1)}$	0	-16.366479	-2.315630	-0.305883	0.103623
$a_8^{(\alpha_1)}$	0	-1.766848	-0.400377	-0.155535	-0.098537
$a_9^{(\alpha_1)}$	0	10.141565	1.426885	0.174665	-0.095684

We present in Figure 4.2 the extracted EFIF compared to the exact EFIF, and to better visualize the difference between them, we present in Figure 4.3 the relative error in percentage between the extracted EFIF and the exact one. As may be seen the difference is in all cases is less than 1%.

Tab. 4.4: Richardson's extrapolation for  $a_0^{(\alpha_1)}$  and  $a_1^{(\alpha_1)}$  with  $\mathcal{O}(R^4)$ 

$a_0^{(\alpha_1)}$				$a_0^{(\alpha_1)}$ exact
$R = 0.9$	5.92248700			
		5.00147778		
$R = 0.5$	5.08921300		5.00355715	5.0
		5.00355208		
$R = 0.2$	5.00574500			
$a_1^{(\alpha_1)}$				$a_1^{(\alpha_1)}$ exact
$R = 0.9$	4.06612400			
		4.00769838		
$R = 0.5$	4.01326400		4.00555763	4.0
		4.00556285		
$R = 0.2$	4.00576000			

Tab. 4.5: Values of extracted  $a_i^{(\alpha_1)}$ ,  $BC_4$ , using  $K_{2k}^{(\alpha_1)}$  at  $R = 0.05$ 

	Case 1	Case 2	Case 3	Case 4
$a_0^{(\alpha_1)}$	4.984674759	4.995171982	4.980150073	4.991610613
$a_1^{(\alpha_1)}$	3.983542333	3.995024025	3.977277342	3.98261663
$a_2^{(\alpha_1)}$	8.990337394	8.995653993	8.991452698	9.00697531
$a_3^{(\alpha_1)}$	3.012099046	2.997408248	3.039678119	3.134998507
$a_4^{(\alpha_1)}$	0.993718479	0.995999184	0.987616268	0.963262624
$a_5^{(\alpha_1)}$	-0.057008407	0.002681025	-0.167575617	-0.565818761
$a_6^{(\alpha_1)}$	0.03361419	0.021475805	0.06111061	0.136589959
$a_7^{(\alpha_1)}$	0.074843576	-0.024476334	0.259820759	0.925421554
$a_8^{(\alpha_1)}$	-0.044252728	-0.027455195	-0.081877613	-0.185272667
$a_9^{(\alpha_1)}$	-0.050032691	0.008661221	-0.161719553	-0.561026566

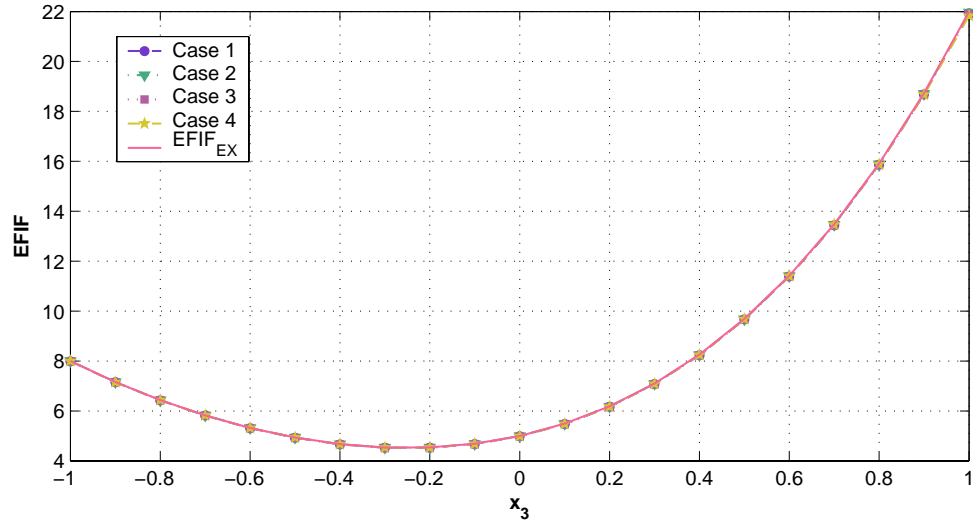


Fig. 4.2: The exact and computed EFIF fourth-order polynomial, using  $K_{2k}^{(\alpha_1)}$  at  $R = 0.05$ .  $A^{(\alpha_1)} = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ .

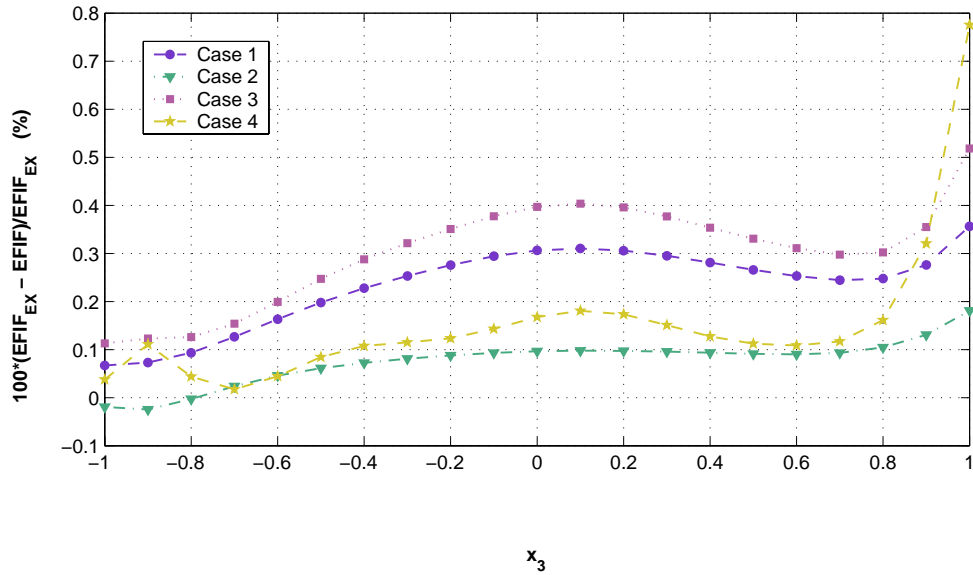


Fig. 4.3: Relative error (%) of the extracted EFIF using  $K_{2k}^{(\alpha_1)}$  at  $R = 0.05$ . The exact EFIF is a fourth-order polynomial.

The results presented in this Chapter show an accurate method for extracting EFIF. The accuracy of the polynomial approximation of the EFIF depends on the selection of the extraction Radius  $R$  and the number of shadow functions ( $m$ ) in the quasi-dual extraction function. The examples presented in this Chapter refer to  $m = 0, 2$  and various radius extractions  $R$ . The results show that even when  $m = 0$  the approximation of the EFIF is very good. By comparing the EFIF's, extracted at different  $R$ s, it is easy to see that the EFIF converges to the exact EFIF as  $R$  decreases. By using Richardson extrapolation for the coefficients of the EFIF, extracted away from the edge, the results obtained are very accurate approximations of the coefficients at  $R \rightarrow 0$ . The extraction method was also examined for cases 1-4. In all four cases the same boundary conditions were applied (same  $A^{(\alpha_1)}(x_3)_{ex}$ ) and as expected, all four cases provided excellent approximations of the EFIF.

Therefore the extraction method using the "simple" family of polynomials is an accurate method for extracting the EFIF.

## 5. COMPUTING EFIFS USING A HIERARCHICAL FAMILY OF EXTRACTION POLYNOMIALS

The ten coefficients  $(a_i)_{FE}$  of a 9-th order polynomial approximation of the EFIF were obtained by using the “simple” extraction polynomials as presented in Chapter 4. Because we are interested in an adaptive procedure, i.e., would like to find out how well does the 9-th order polynomial approximate  $A(x_3)$ , we would like to find the 10-th order polynomial approximation, the 11-th and so on. Using these “simple” extraction polynomials, we need to generate each time  $n+1$  polynomials, if interested in the  $n$ -th polynomial approximation of  $A(x_3)$ . This is a tedious and inefficient procedure. Therefore a more sophisticated method has to be developed, which may easily compute a polynomial approximation of order  $n+1$  of  $A(x_3)$  once the  $n$ 'th order approximation is already computed: i.e. one has to find a hierarchical family of extraction polynomials.

### 5.1 *Constructing a hierarchical family of extraction polynomials*

$$BJ_m^{(k)}(x_3)$$

We would like to construct an adaptive class of orthonormal polynomials with a given weight  $w(x_3) = (1 - x_3^2)^m$  so to represent  $B_m(x_3)$ . This suggests the use of Jacobi polynomials as a natural basis. In this way, if  $A(x_3)$  is a polynomial of degree  $n$ , it can be represented by a linear combination of Jacobi polynomials as:

$$A(x_3) \approx \tilde{a}_0 J_0 + \tilde{a}_1 J_1(x_3) + \cdots + \tilde{a}_n J_n(x_3) \quad (5.1)$$

where [1, pp. 773-774]:

$$J_i(x_3) = \frac{1}{i^2 + 7i + 12} \sum_{l=0}^i \frac{(i+l+4)!}{2^l l! (i-l)! (2+l)!} (x_3 - 1)^l \quad (5.2)$$

The hierarchical family of extraction polynomials, denoted by  $BJ_m^{(k)}(x_3)$ , have to be chosen so to satisfy  $BJ_m^{(k)}(\pm 1) = \partial_3 BJ_m^{(k)}(\pm 1) = \dots = \partial_3^{m-1} BJ_m^{(k)}(\pm 1) = 0$ . So these may be used with  $m = 2$  for generating both  $K_{0k}$  and  $K_{2k}$  dual functions, we choose:

$$BJ_2^{(k)}(x_3) = (1 - x_3^2)^2 \frac{J_k(x_3)}{h_k} \quad (5.3)$$

The factor  $(1 - x_3^2)^2$  is to ensure that  $BJ_m^{(k)}(\pm 1) = \partial_3 BJ_m^{(k)}(\pm 1) = 0$ . With  $J_k(x_3)$  being the Jacobi polynomials, denoted in literature by  $P_k^{(2,2)}(x_3)$  with the following important property [1, pp. 773-774] :

$$\int_{-1}^1 (1 - x_3^2)^2 J_n(x_3) J_k(x_3) dx_3 = \delta_{nk} h_k \quad (5.4)$$

such that

$$h_k = \frac{2^5 (k+1)(k+2)}{(2k+5)(k+3)(k+4)} \quad (5.5)$$

By choosing these extraction polynomials, we can adaptively increase the polynomial order of the EFIFs. Of course,  $A^{(\alpha_i)}(x_3)$  is an unknown function, and we wish to find a projection of it into the space of polynomials. It is expected that as we increase the polynomial space, the approximation is better (at least in a given region). For example, let us say that we are interested in projecting  $A^{(\alpha_i)}(x_3)$  into the space of polynomials of degree up to  $n$ . Thus we have to compute the  $n+1$  coefficients  $a_0 \dots a_n$ . To accomplish this we use the  $n+1$  extraction polynomials  $BJ_2^{(0)}(x_3), \dots, BJ_2^{(n)}(x_3)$  defined in (5.3). So that we get according to (5.4):

$$\int_{-1}^1 A^{(\alpha_i)}(x_3) BJ_2^{(j)}(x_3) dx_3 = \tilde{a}_j^{(\alpha_i)} \quad j = 0, 1, \dots, n \quad (5.6)$$

This way, if we want to increase the space in which  $A^{(\alpha_i)}(x_3)$  is projected, all which is needed is the computation of (5.6) for  $j = n + 1$ . This way, the new  $A^{new}(x_3) = A^{previous}(x_3) + \tilde{a}_{n+1}J_{n+1}(x_3)$ .

Inserting (5.5) and (5.2) in (5.3), we finally obtain:

$$BJ_2^{(i)}(x_3) = \frac{(2i+5)(i+3)(i+4)}{2^5(i+1)(i+2)} \frac{(1-x_3^2)^2}{i^2+7i+12} \sum_{l=0}^i \frac{(i+l+4)!}{2^l l! (i-l)! (2+l)!} (x_3-1)^l \quad (5.7)$$

## 5.2 Numerical results - using $BC_4$ with $K_{2k}^{(\alpha_1)}$

For the benchmark problem with  $BC_4$  for which the exact EFIF is a polynomial of degree 4 in  $x_3$ :  $a_0 = 5, a_1 = 4, a_2 = 9, a_3 = 3, a_4 = 1$  and  $a_5 = \dots = a_9 = 0$ , and using the extraction polynomials  $BJ_2^{(0)}(x_3) \dots BJ_2^{(k)}(x_3)$ , where  $0 \leq k \leq 4$ , we extract the EFIF for case 2 at  $R = 0.05$ . Of course as  $k > 4$ , we should fully recover the EFIF. Using 15 integration points and  $p = 8$  in the finite element mesh we extract the coefficients and compute the EFIFs. Figure 5.1 presents the relative error in percentage between the extracted EFIF and the exact one, for the family of extraction polynomials  $BJ_2^{(k)}(x_3)$ . As may be seen for the family of degree 4 we indeed fully recover the exact EFIF.

As one increases the order of the hierarchical family, the results do not improve, but we obtain an oscillatory behavior of the solution due to numerical errors (the finite element solution is not exact), with a very small amplitude as demonstrated in Figure 5.2.

To illustrate the convergence of the extracted values as a function of  $R$ , we present in Table 5.1 the coefficients of the extracted polynomial at  $R = 0.9, 0.5, 0.2, 0.05$ . We use Richardson's extrapolation, with the error behaving as  $\mathcal{O}(R^4)$ , and the coefficients at  $R = 0.9, 0.5$  to extrapolate to  $R = 0$ . These extrapolation results are shown in Table 5.2.

We illustrated the relative error in the extrapolated EFIF using the data at  $R =$

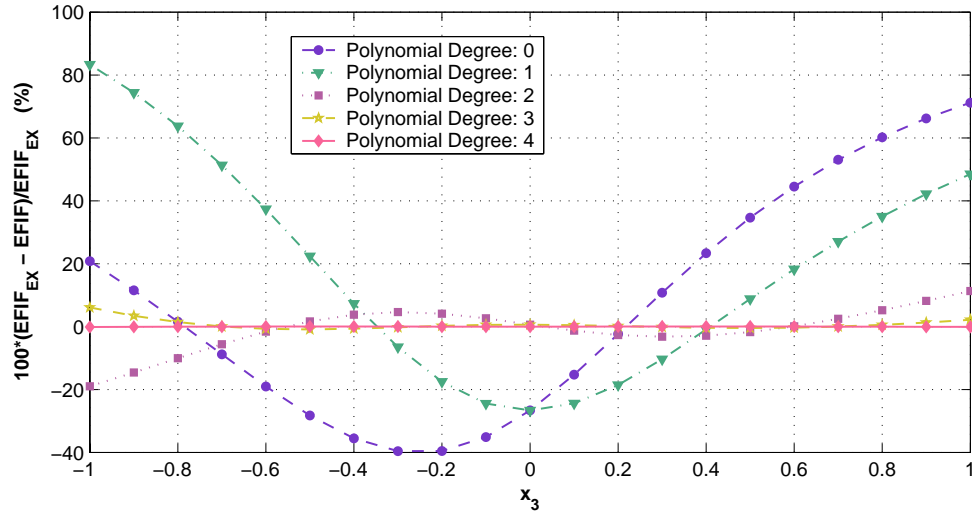


Fig. 5.1: Relative error (%) of the extracted EFIF at  $R = 0.05$  and the exact one using  $K_{2k}^{(\alpha_1)}$  and the hierarchical family  $BJ_2^{(k)}(x_3)$  (of degree  $k$ ). The exact EFIF is a fourth-order polynomial  $A_{ex}^{(\alpha_1)} = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ .

0.9, 0.5 in Figure 5.3. By extracting the EFIF from the FE solution away from the singular edge (where usually the numerical data is polluted), we demonstrate that a very good approximation is obtained by Richardson's extrapolation, taking into consideration that the error behaves as  $\mathcal{O}(R^4)$ . Practically, the relative error in the extrapolated EFIF is as obtained very close to the singular edge ( $R = 0.05$ ), and much better than the values obtained when extraction is performed at  $R = 0.5$ .

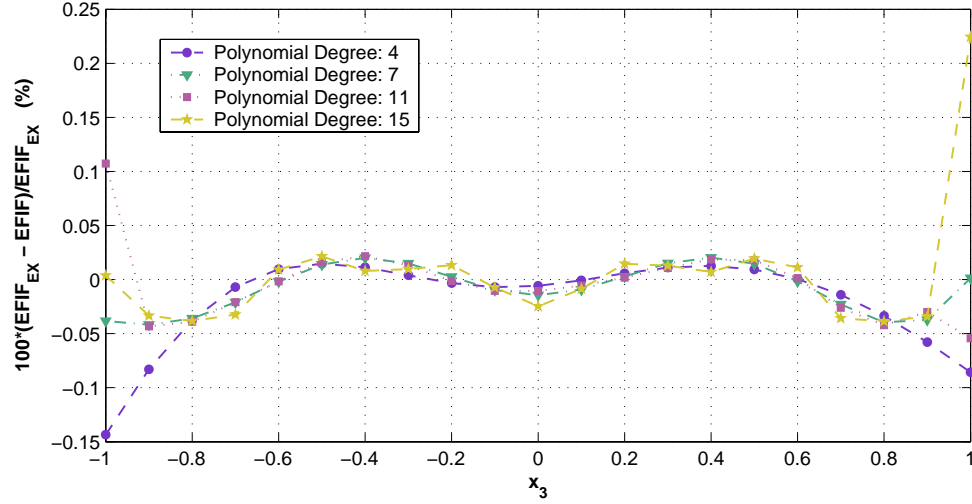


Fig. 5.2: Relative error (%) of the extracted EFIF at  $R = 0.05$  using  $K_{2k}^{(\alpha_1)}$  and the hierarchical family  $BJ_2^{(k)}(x_3)$ , with  $k$  taken up to  $15^{th}$  degree.

Tab. 5.1: Values of extracted  $a_i^{(\alpha_1)}$ ,  $BC_4$ , using  $K_2^{(\alpha_1)}$  and  $BJ_2^{(m)}(x_3)$ ,  $m \leq 4$  at different  $R$ s.

	Exact	$R = 0.9$	$R = 0.5$	$R = 0.2$	$R = 0.05$
$a_0^{(\alpha_1)}$	5	5.920806968	5.089253508	5.005993235	5.000288235
$a_1^{(\alpha_1)}$	4	4.004545148	4.002303539	4.002751475	3.99852796
$a_2^{(\alpha_1)}$	9	9.047407703	9.00825309	9.001724824	8.989161317
$a_3^{(\alpha_1)}$	3	2.985298783	2.995871625	3.001625541	3.005167695
$a_4^{(\alpha_1)}$	1	0.90483039	0.98390502	1.007098452	1.025721321

Tab. 5.2: Values of extrapolated  $a_i^{(\alpha_1)}$ ,  $BC_4$ , using  $K_2^{(\alpha_1)}$  and  $BJ_2^{(m)}(x_3)$ ,  $m \leq 4$ , based on  $R = 0.9, 0.5$  of Table 5.1.

	Exact	Extrapolated Using $R = 0.9, 0.5$
$a_0^{(\alpha_1)}$	5	5.001699446
$a_1^{(\alpha_1)}$	4	4.002067521
$a_2^{(\alpha_1)}$	9	9.00413051
$a_3^{(\alpha_1)}$	3	2.996984837
$a_4^{(\alpha_1)}$	1	0.992230769

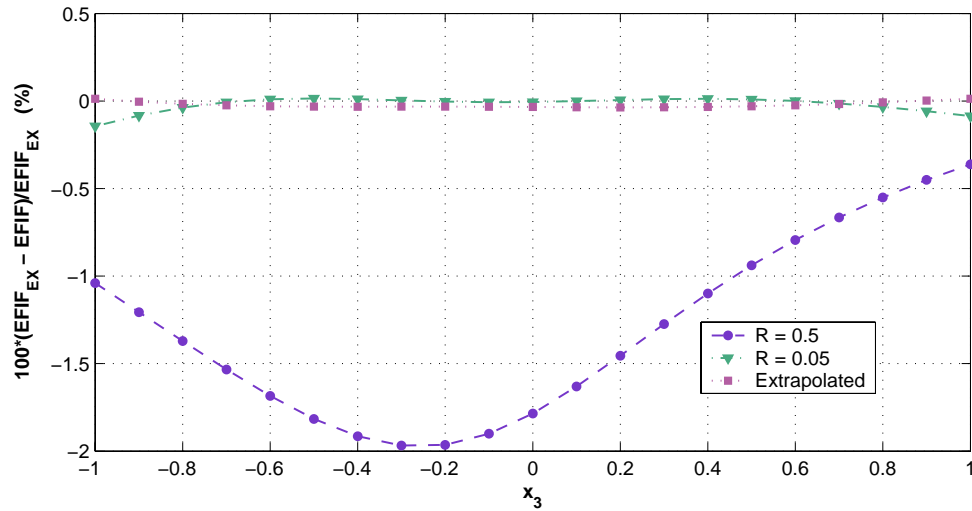


Fig. 5.3: Relative error (%) of the extracted EFIF at  $R = 0.5, 0.05$  and using Richardson's extrapolation from data at  $R = 0.9, 0.5$ . EFIF computed using  $K_{2k}^{(\alpha_1)}$  and the hierarchical family  $BJ_2^{(k)}(x_3)$ , with  $k \leq 4$ . The exact EFIF is a fourth-order polynomial  $A_{ex}^{(\alpha_1)} = 5 + 4x_3 + 9x_3^2 + 3x_3^3 + x_3^4$ .

The hierarchical family of polynomials presented in this Chapter provides an adaptive method for extracting the EFIF. The relative error between the extracted EFIF and the exact EFIF is reduced as the degree of extracted EFIF polynomial is increased. When the EFIF polynomial reaches a relative error of  $\sim \pm 0.1\%$ , which is a very good result, increasing the degree of polynomial does not improve the results (due to numerical errors). By using Richardson extrapolation, one can receive a very good approximation of the EFIF while the coefficients are extracted away from the edge.

## 6. A POLYNOMIAL REPRESENTATION OF A NON-POLYNOMIAL EFIF

We have demonstrated so far that the quasi-dual extraction method performs very well if the exact EFIF is a polynomial.

A natural question is - what if the EFIF is not a polynomial? In this case an adaptive hierarchical algorithm for polynomial space enrichment is demonstrated, so to provide an estimate of the convergence of the polynomial approximation to the exact EFIF. Herein we investigate the performance of such hierarchical space enrichment for the case where the exact EFIF is a general function, and furthermore, it tends to very high values at the ends of the edge. For example, consider case 2, where the EFIF is a function of the form:

$$A^{(\alpha_1)}(x_3) = \frac{\sin x_3}{(d - x_3^2)} \quad (6.1)$$

where  $d$  is a given number<sup>1</sup>. As  $d$  approaches 1, the EFIF approaches infinity at the vertices  $x_3 = \pm 1$ . We chose three values of  $d = 2, 1.5, 1.05$ .

Consider the following problem:

$$\begin{aligned} L(\tau) &= \partial_3^2 A^{(\alpha_1)}(x_3) \Phi_0^{(\alpha_1)}(r, \theta) \quad \text{in } \Omega \\ \tau &= A^{(\alpha_1)}(x_3) \Phi_0^{(\alpha_1)}(r, \theta) \quad \text{on } \partial\Omega, \end{aligned} \quad (6.2)$$

for which the exact solution is simply  $\tau_{ex} = A^{(\alpha_1)}(x_3) \Phi_0^{(\alpha_1)}(r, \theta)$ .

---

<sup>1</sup> It is important to notice that we assume dimensionless quantities in our computation so that  $\sin(x_3)$ ,  $x_3^2$  and  $d$  are dimensionless.

A refined finite element model graded towards  $x_3 = \pm 1$  was generated as shown in Figure 6.1. It has 25 elements in the  $x_3$  direction and a total of 800 solid finite elements.

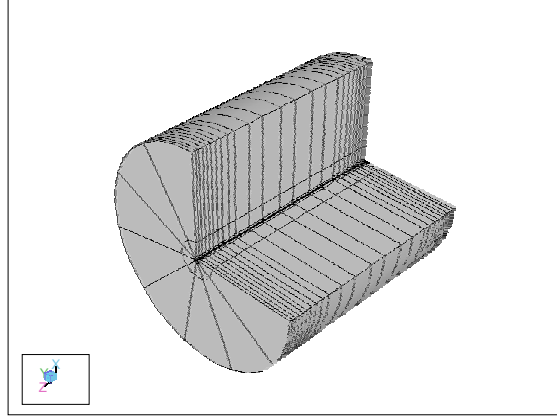


Fig. 6.1: The p-FEM model for the problem having a non-polynomial EFIF tending to large values at  $x_3 = \pm 1$ .

To evaluate the accuracy of the extracted EFIFs, one has first to examine the numerical results,  $\tau_{FE}$  and its derivatives, especially for solutions having large gradients. The graphs in Figure 6.2 present the relative error in  $\tau_{FE}$  and  $\partial_r \tau_{FE}$  in percentage, extracted from the finite element solution at  $p = 8$  for  $d = 2, 1.5, 1.05$ . These graphs are along the line  $R = 0.05$ ,  $\theta = 135^\circ$  and  $-1 \leq x_3 \leq 1$ . The FE results have a relative error of about 3% for  $-0.8 \leq x_3 \leq 0.8$ , and around 17% for  $0.8 < |x_3| < 1$  for the case when  $d = 1.05$ . This in turn will perturb the extraction of the EFIF by that order of magnitude when using the quasi-dual extraction technique, as we show in the sequel. We will also observe that the EFIFs are computed with similar accuracy and the extraction technique does not magnify the numerical error but the opposite. For  $d = 2, 1.5$ , the relative error in the function and its derivatives is very small (less than 0.7%) in all the range. Therefore, the extraction of the EFIFs

is expected to provide excellent results.

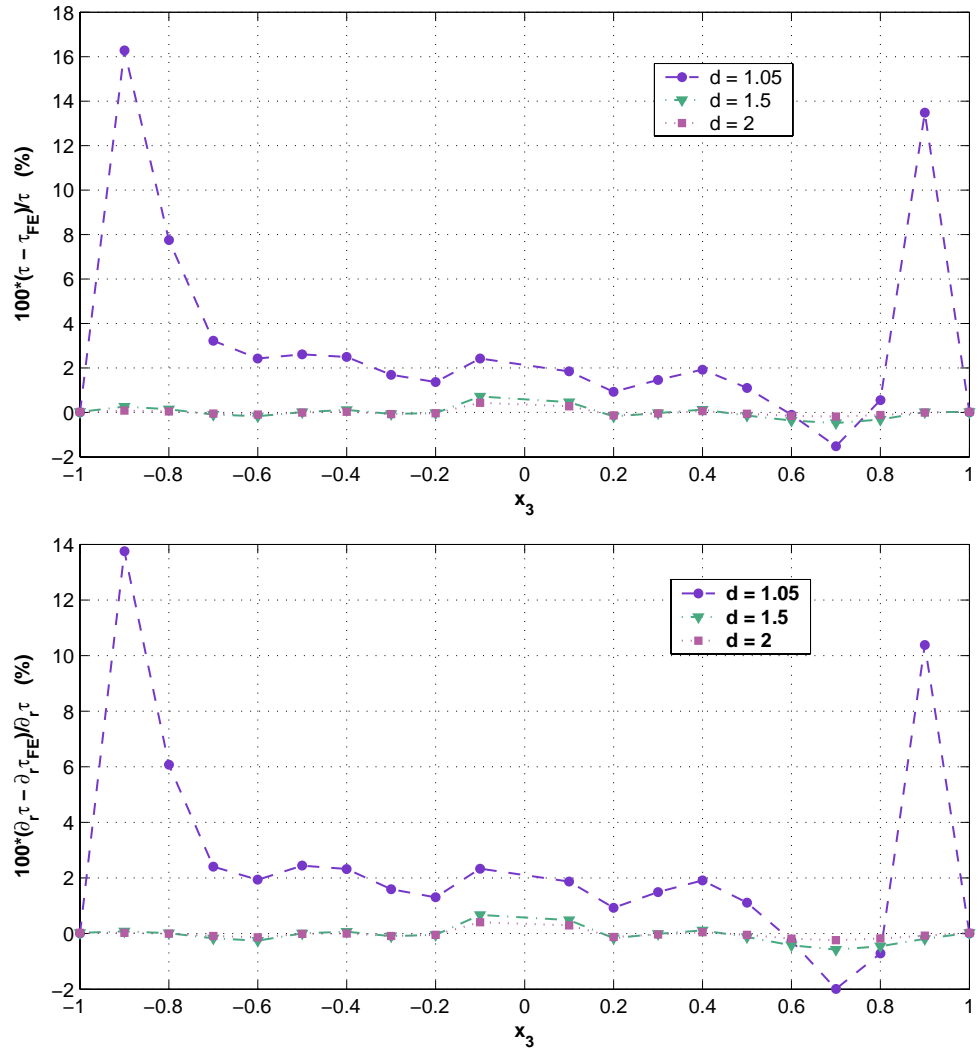


Fig. 6.2: Relative error in FE solution and its derivatives (%) at  $p = 8$  for  $(r, \theta, x_3) = (0.05, \theta = 135^\circ, [-1, 1])$  for the three problems defined by  $d = 2, 1.5, 1.05$ .

### 6.1 Computing EFIFs Using an Hierarchical Family of Extraction Polynomials

Using  $K_{2k}^{(\alpha_1)}$  and the hierarchical family  $BJ_2^{(k)}(x_3)$ , we extracted the EFIFs at  $R = 0.05$  using the solution at  $p = 8$  and 54 Gauss integration points (due to the strong gradients of the solutions we used a higher integration scheme). We also checked with 94 Gauss integration points that the integration error in evaluating  $J[R]$  is negligible.

Figure 6.3 presents the exact EFIF and the extracted EFIF using  $BJ_2^{(k)}(x_3)$  of increasing order obtained at  $R = 0.05$ . One may easily observe the strong gradients of the EFIF at  $x_3 = \pm 1$ , especially for the case where  $d = 1.05$ .

Relative errors between the extracted EFIF and the exact value are presented in Figure 6.4. For all cases of  $d$ , the EFIF is progressively better approximated away from the large gradients ( $\approx -0.85 \leq x_3 \leq 0.85$ ) as the order of the extraction polynomials ( $k$ ) is increased.

To better observe the convergence of the EFIF in this region we present in Figure 6.5 the data presented in Figure 6.4 zoomed around 0% relative error. At  $k = 19$  one may observe that the extracted EFIF has less than 3% relative error for the case when  $d = 1.05$  and less than 0.5% relative error for the cases  $d = 1.5$  and  $d = 2$ .

The large pointwise errors in the close neighborhood of the high gradients is expected.

We use the problem characterized by  $d = 1.05$  to present the convergence of the extracted  $a_j^{(\alpha_1)}$  as  $R$  approaches zero, when the hierarchical family of polynomials  $BJ_2^{(k)}(x_3)$ ,  $k \leq 19$  is used. The coefficients of the polynomial representation  $A^{(\alpha_1)} = a_0^{(\alpha_1)} + a_1^{(\alpha_1)}x_3 + \cdots + a_{19}^{(\alpha_1)}x_3^{19}$  extracted at  $R = 0.9, 0.5, 0.2, 0.05$  and these extrapolated to  $R = 0$  are presented in Table 6.1. We use Richardson's extrapolation, with the error behaving as  $\mathcal{O}(R^4)$ , and the coefficients at  $R = 0.9, 0.5, 0.2, 0.05$  to extrapolate to  $R = 0$ .

We illustrated the relative error in the extrapolated EFIF using the data at  $R = 0.9, 0.5, 0.2, 0.05$  in Figure 6.6. For these cases where the EFIF has very strong gradients, and the finite element solution is of low accuracy, the Richardson's extrapolation does not improve the accuracy beyond this given at the smallest extraction radius.

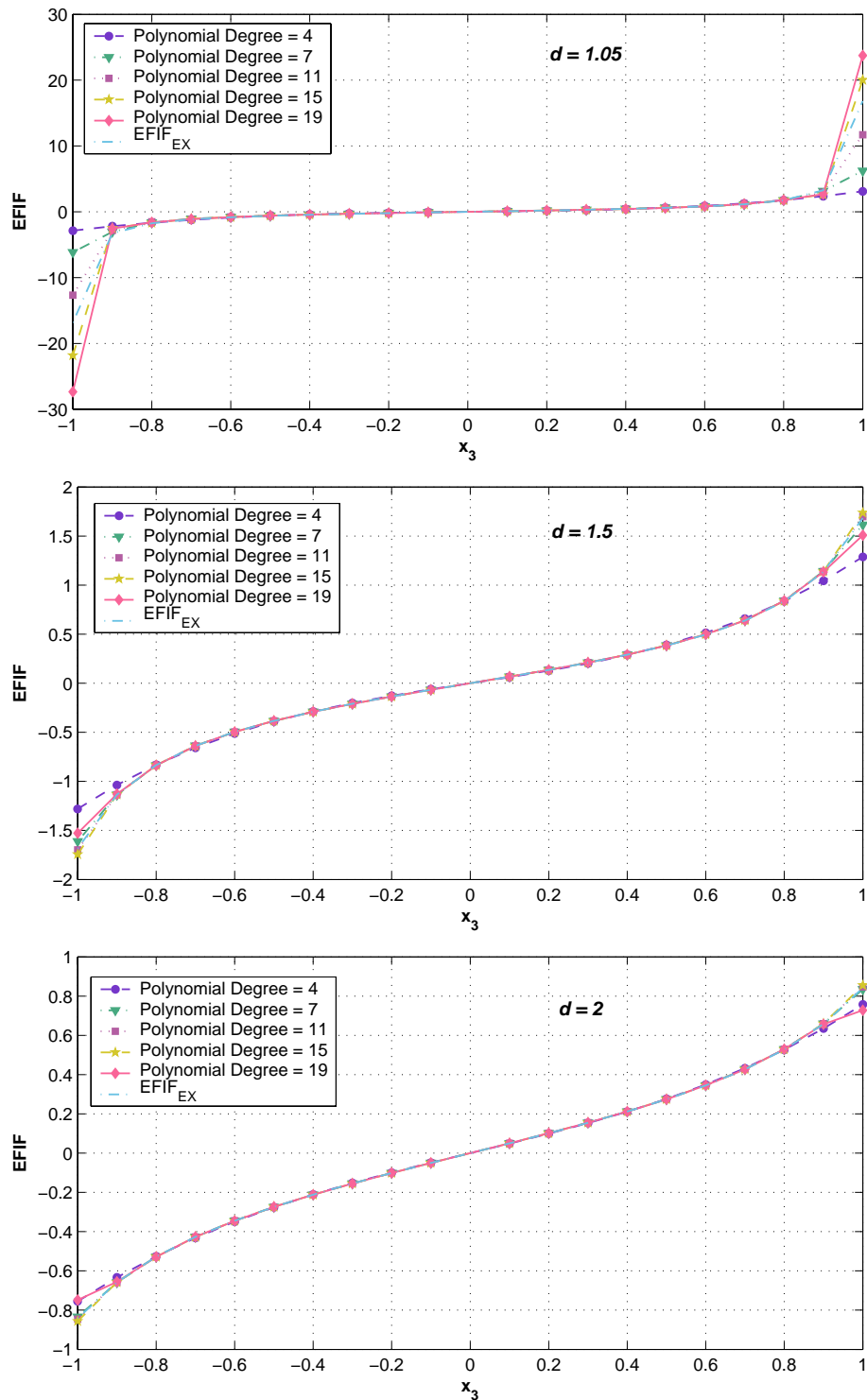


Fig. 6.3: The exact and extracted EFIF, using the hierarchical extraction polynomials of degree  $k$ ,  $K_{2k}^{(\alpha_1)}$ ,  $p = 8$ , at  $R = 0.05$  and 54 Gauss points.

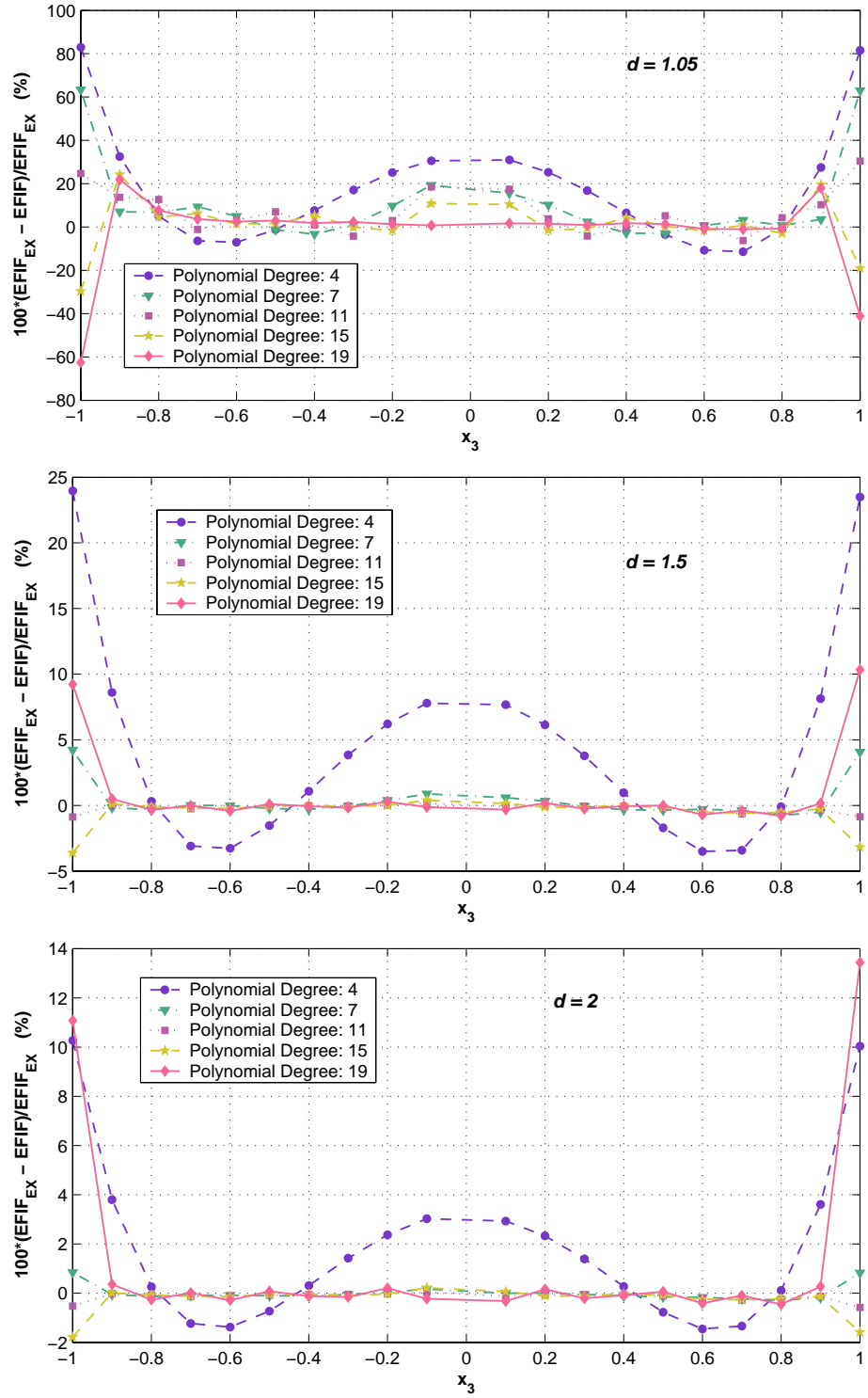


Fig. 6.4: Relative error (%) of extracted EFIF, using the hierarchical extraction polynomials of degree  $k$ ,  $K_{2k}^{(\alpha_1)}$ ,  $p = 8$ , at  $R = 0.05$  and 54 Gauss points.

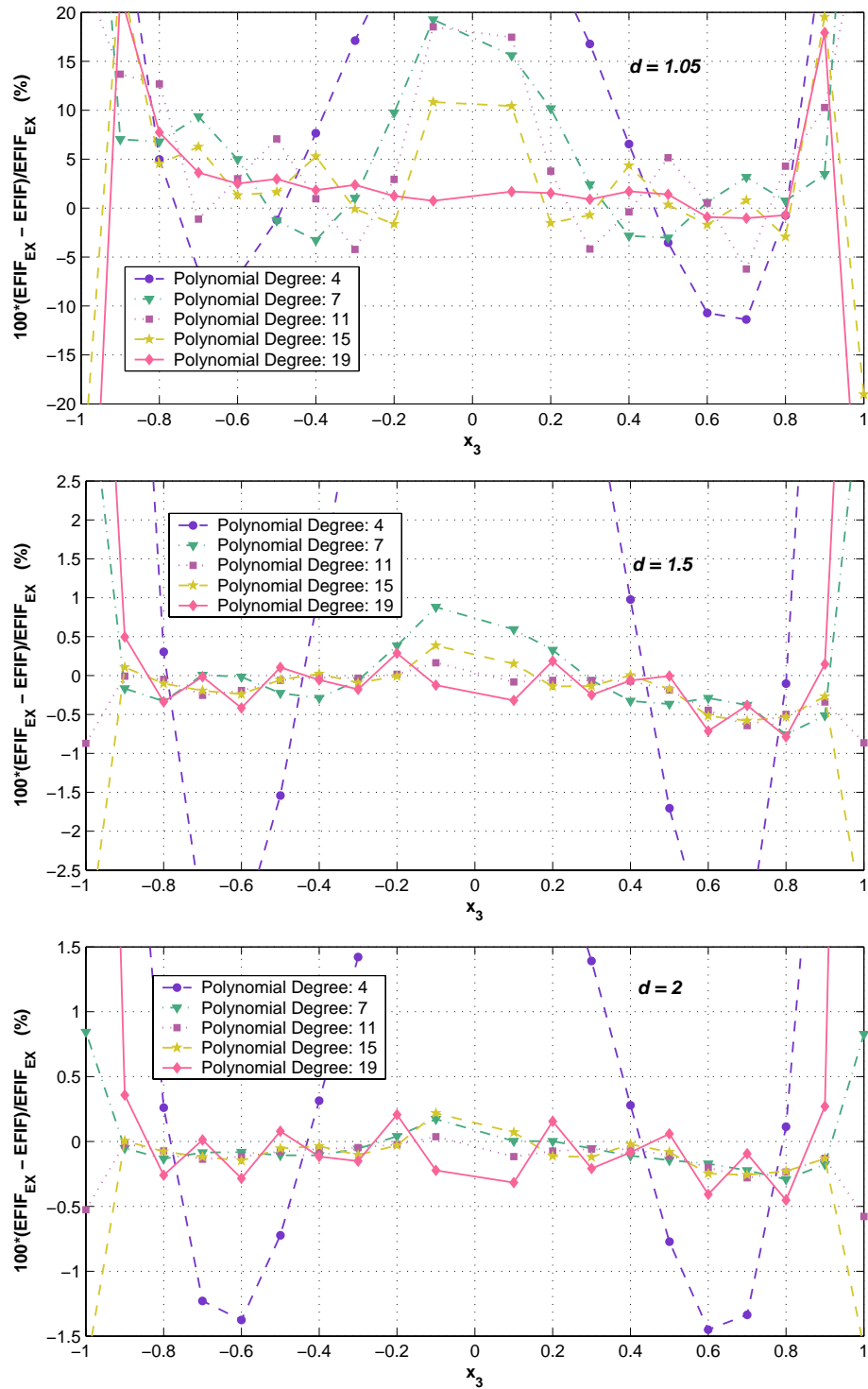


Fig. 6.5: Relative error zoomed around 0% of extracted EFIF, using the hierarchical extraction polynomials of degree  $k$ ,  $K_{2k}^{(\alpha_1)}$ ,  $p = 8$ , at  $R = 0.05$  and 54 Gauss points.

Tab. 6.1: Values of extracted  $a_i^{(\alpha_1)}$  for the case with  $d = 1.05$ , using  $K_{2k}^{(\alpha_1)}$  and  $BJ_2^{(k)}(x_3)$ ,  $k \leq 19$  at different  $R$ s, and extrapolated.

	$R = 0.9$	$R = 0.5$	$R = 0.2$	$R = 0.05$	Extrapolated Using $R =$ 0.9, 0.5, 0.2, 0.05
$a_0^{(\alpha_1)}$	8.1856E-01	1.8735E-02	6.6996E-04	1.6269E-03	1.6339E-03
$a_1^{(\alpha_1)}$	-2.0061E+00	-4.8343E+01	-1.8363E+00	9.3962E-01	9.5067E-01
$a_2^{(\alpha_1)}$	-1.8507E+02	-3.8411E+00	-3.0800E-01	-3.0275E-01	-3.0274E-01
$a_3^{(\alpha_1)}$	5.4053E+02	4.0292E+03	2.2935E+02	9.5502E-01	4.5259E-02
$a_4^{(\alpha_1)}$	6.7513E+03	1.2693E+02	1.9325E+01	1.0813E+01	1.0779E+01
$a_5^{(\alpha_1)}$	-1.7477E+04	-9.4900E+04	-5.3861E+03	-1.0273E+01	1.1156E+01
$a_6^{(\alpha_1)}$	-9.2084E+04	-1.7008E+03	-3.7931E+02	-1.4364E+02	-1.4269E+02
$a_7^{(\alpha_1)}$	2.1390E+05	9.9567E+05	5.6550E+04	1.8130E+02	-4.3023E+01
$a_8^{(\alpha_1)}$	6.1561E+05	1.1484E+04	3.3063E+03	9.3950E+02	9.2998E+02
$a_9^{(\alpha_1)}$	-1.2988E+06	-5.5587E+06	-3.1590E+05	-1.4556E+03	-2.0319E+02
$a_{10}^{(\alpha_1)}$	-2.2718E+06	-4.2583E+04	-1.4905E+04	-3.3868E+03	-3.3404E+03
$a_{11}^{(\alpha_1)}$	4.3893E+06	1.7944E+07	1.0204E+06	6.3883E+03	2.3493E+03
$a_{12}^{(\alpha_1)}$	4.8506E+06	9.0710E+04	3.7264E+04	7.0647E+03	6.9431E+03
$a_{13}^{(\alpha_1)}$	-8.6400E+06	-3.4523E+07	-1.9648E+06	-1.5899E+04	-8.1363E+03
$a_{14}^{(\alpha_1)}$	-5.9556E+06	-1.1037E+05	-5.1989E+04	-8.4782E+03	-8.3030E+03
$a_{15}^{(\alpha_1)}$	9.8390E+06	3.8986E+07	2.2207E+06	2.2383E+04	1.3627E+04
$a_{16}^{(\alpha_1)}$	3.8958E+06	7.1035E+04	3.7793E+04	5.4274E+03	5.2970E+03
$a_{17}^{(\alpha_1)}$	-6.0060E+06	-2.3812E+07	-1.3577E+06	-1.6580E+04	-1.1239E+04
$a_{18}^{(\alpha_1)}$	-1.0500E+06	-1.8707E+04	-1.1131E+04	-1.4353E+03	-1.3963E+03
$a_{19}^{(\alpha_1)}$	1.5215E+06	6.0636E+06	3.4618E+05	5.0161E+03	3.6573E+03

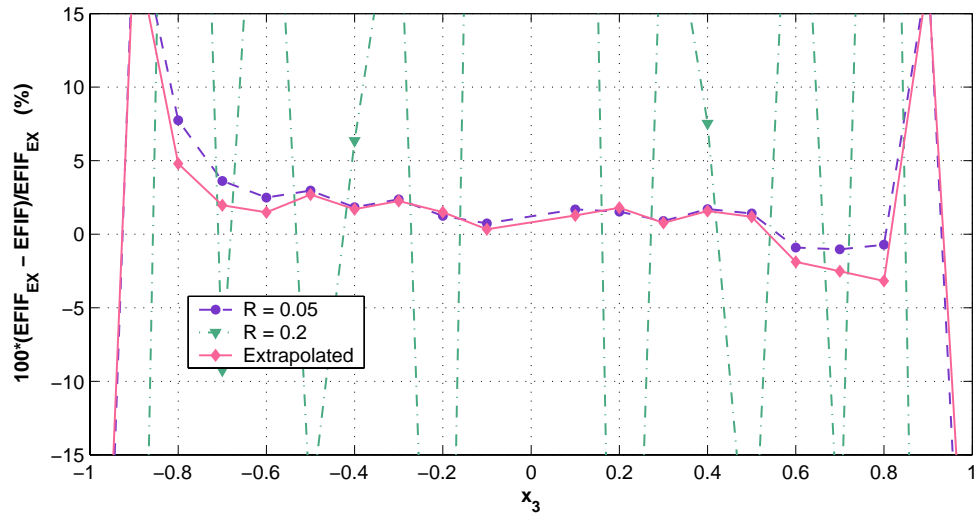


Fig. 6.6: Relative error (%) of the extracted EFIF at  $R = 0.2, 0.05$  and using Richardson's extrapolation from data at  $R = 0.9, 0.5, 0.2, 0.05$ . EFIF computed using  $K_{2k}^{(\alpha_1)}$  and the hierarchical family  $BJ_2^{(k)}(x_3)$ , with  $k \leq 19$ . The exact EFIF is  $A^{(\alpha_1)}(x_3) = \frac{\sin x_3}{(d-x_3^2)}$ , with  $d = 1.05$ .

## 6.2 Computing EFIFs Using a "Simple" Family of Extraction Polynomials

As the EFIFs have been extracted by the hierarchical family of extraction polynomials, it is also interesting to visualize the results when extracted by "simple" extraction polynomials at different radii. Of course we expect identical results to those obtained by the hierarchical family of order 9.

For the problem with  $d = 1.05$ , using 54 Gauss points, and the FE solution at  $p = 8$  and the dual extraction function  $K_{2k}^{(\alpha_1)}$  we obtain the data summarized in Table 6.2.

Tab. 6.2: Values of extracted  $a_i^{(\alpha_1)}$ , for  $d = 1.05$  using the "simple" extraction polynomials,  $K_{2k}^{(\alpha_1)}$  at different  $R$ s

	$R = 0.9$	$R = 0.5$	$R = 0.2$	$R = 0.05$
$a_0^{(\alpha_1)}$	0.180381895	0.005718293	0.000874814	-0.000479043
$a_1^{(\alpha_1)}$	10.50863143	4.128950762	1.540068032	0.996381364
$a_2^{(\alpha_1)}$	-12.05365926	-0.568294488	-0.045248079	0.05594091
$a_3^{(\alpha_1)}$	-268.6434399	-93.29068638	-15.61594767	-0.762082317
$a_4^{(\alpha_1)}$	128.9941269	3.791042131	0.298540606	-0.545939736
$a_5^{(\alpha_1)}$	1796.590434	637.5863189	113.5008291	10.08201703
$a_6^{(\alpha_1)}$	-417.8461538	3.48650068	0.303480116	1.879851558
$a_7^{(\alpha_1)}$	-4268.241849	-1556.342814	-270.544577	-20.06337682
$a_8^{(\alpha_1)}$	373.8230211	-12.49822012	-0.543907697	-1.438630681
$a_9^{(\alpha_1)}$	3237.491443	1148.928745	203.3701282	16.02264003

The polynomial coefficients of Table 6.2 are used to compute the extracted EFIF as presented in Figure 6.2, and the relative error in percentage for different extraction radii  $R$ .

As we can see from the results presented in this Chapter, the EFIF extraction method provides an accurate approximation of the EFIF even when the exact EFIF is not a polynomial. Two families of polynomials were examined, the "simple" family and the hierarchical family of polynomials. Both extraction families provide good results away from the vertices and by using Richardson extrapolation one can produce even more accurate results.

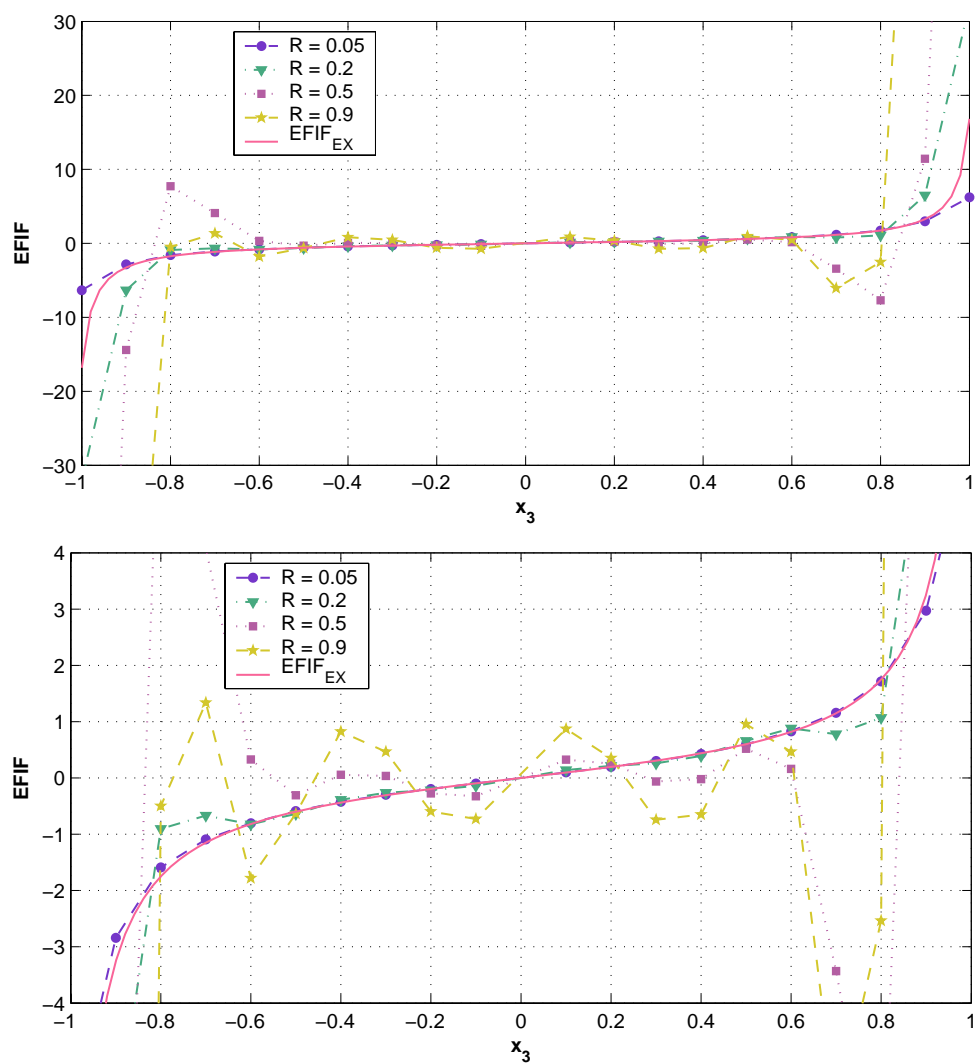


Fig. 6.7: The exact and extracted EFIF for the problem with  $d = 1.05$ , using “simple” extraction polynomials  $K_{2k}^{(\alpha_1)}$ ,  $p = 8$ , 54 Gauss points, at different  $R$ s. Bottom graph is zoomed.

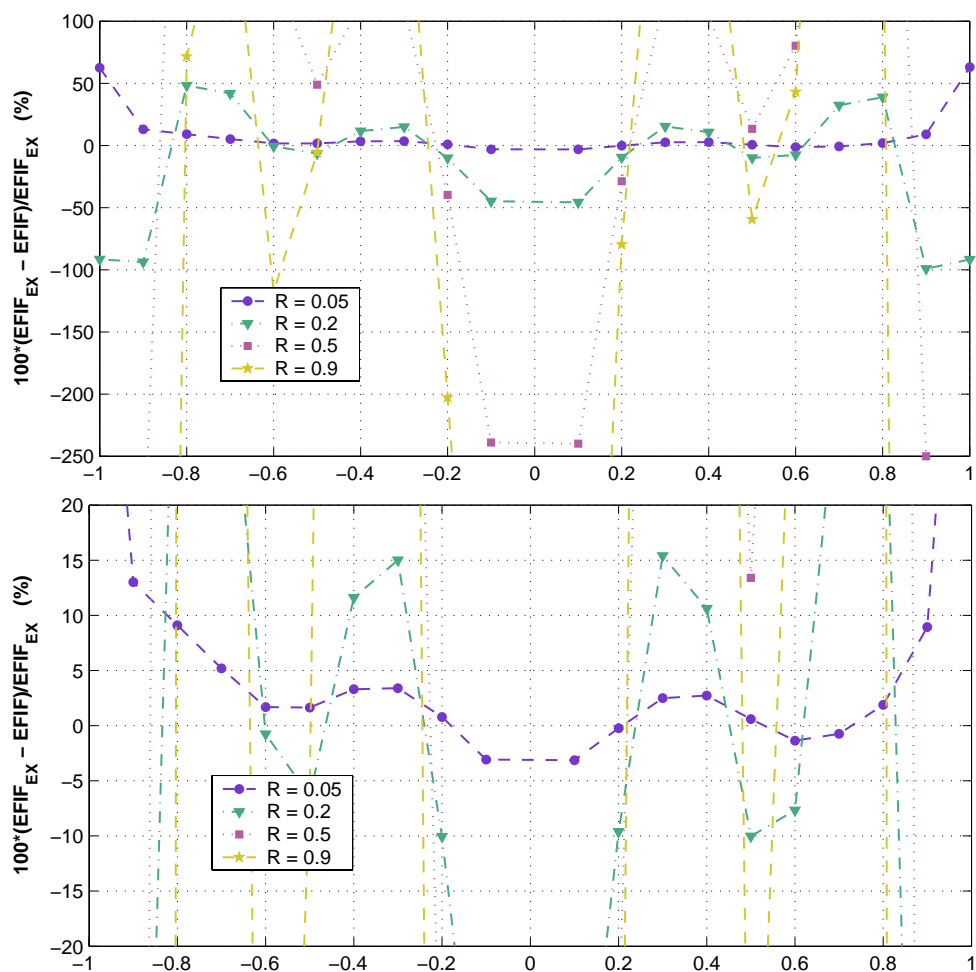


Fig. 6.8: Relative error (%) of extracted EFIF for the problem with  $d = 1.05$ , using “simple” extraction polynomials  $K_{2k}^{(\alpha_1)}$ ,  $p = 8$ , 54 Gauss points, at different  $R$ s. Bottom graph is zoomed.

## 7. SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

### 7.1 *Summary and Conclusions*

The solution of the general scalar elliptic equation in three dimensional domains, in the neighborhood of an edge is of great importance from an engineering point of view. It is characterized by the eigen-pairs, their associated shadow functions and the EFIFs. The edge stress intensity functions (ESIFs), similar to the EFIFs, are associated with failure theories and therefore are of interest in fracture mechanics for evaluating the life of structures. Since the  $3 - D$  scalar elliptic problem has the same characteristics as the  $3 - D$  elastic solution and is simpler to consider, we first investigated it.

Due to the complexity of the three-dimensional scalar elliptic solution in the vicinity of an edge, many researches have focused on the two dimensional domains containing a reentrant corners. Methods used for two dimensional problems as the semi analytical approach (the path independent integral) and numerical approaches (finite element methods), cannot be expanded to the three dimensional domains easily. The methods for EFIF extraction, which have been reported in the past, focused on two dimensional approximation of the three dimensional domain under plane stress, plane strain or the point-wise method. These methods provide an estimation of the EFIF at a specific point on the edge, but do not provide a functional representation of the EFIF.

Herein we presented an extension of the contour integral method for extracting EFIF, based on the mathematical framework which is presented in [10]. The method provides a functional (polynomial) representation of the EFIF along the edge. This accurate and efficient method is implemented as a post-solution operation in conjunction with the p-version finite element method.

Two families of extraction polynomials were examined in this work. The first is a "simple family of polynomials". Each "simple family of polynomials" is produced to extract a specific polynomial approximation to the EFIF of a given degree. The advantage of the "simple family of polynomials" is that each coefficient in the EFIF polynomial can be separately computed, independent of the other coefficients. The disadvantage of this family of polynomials is that when one wants to increase the order of EFIF polynomial approximation, one has to calculate a new set of "simple family of polynomials". The second family of polynomials is the "hierarchical family of polynomials". It is based on Jacobi orthogonal polynomials. The quasi-dual function method, with the use of the "hierarchical family of polynomials" becomes adaptive in the sense that it uses a simple procedure to increase the degree of the extracted EFIF polynomial. Both families of polynomials were shown to provide good results when computing EFIFs.

Analytical solutions have been constructed against which the extracted EFIFs were compared. As shown, the relative errors of the extracted EFIF were less than 1%, when the degree of the extracted EFIF polynomials is determined by an adaptive method, and Richardson extrapolation was used.

The extraction method uses a finite element solution and Gauss quadrature for numerical integration. Both the finite element solution and the Gauss quadratures involve numerical errors. The errors were monitored and the results presented in this work show that both the FEM errors and the numerical quadrature influence the accuracy of the extracted EFIF very little, when a high polynomial degree is used in

the FEM approximation and the quadrate order is at least 10.

The results presented herein indicate that the method proposed for EFIF extraction is accurate and efficient.

## 7.2 *Suggestions for Future Research*

Since near vertecies the EFIF may tend to infinity and have a large gradient, one may wish to extract EFIF in a given interval - i.e. a piecewise extraction method has to be developed.

Of course the  $3 - D$  singularity problem is of engineering importance, for which the solution near an edge has to be derived in terms of  $2 - D$  eigen-pairs and their shadows. Then, the quasi-dual function has to be extended to extract ESIFs.

## APPENDIX

## A. COMPUTER PROGRAM

The  $J[R]$  computation requires a finite element solution and its derivatives evaluated at different  $r, \theta, z$  points. Furthermore, we wish to be able to specify different equation coefficients ( $k_{11}, k_{12}, k_{22}, k_{13}, \dots$ ), domain opening angles  $\omega$  and boundary conditions so that the FE models/analysis changes accordingly. For this purpose we first construct a parametric FE model using StressCheck, such that  $k_{11}, k_{12}, k_{22}, k_{13}, \dots, \omega, a_1, a_2, \dots$  are defined parameters, and the models mesh and analysis changes according to them.

Via a COM interface, using Visual Basic, a computer code has been developed which activate StressCheck, loads the parametric FE model, changes the parameters, runs the FE analysis and then extracts the required solution and its derivatives at points required for  $J[R]$  computation. Finally the computer program computes  $J[R]$  and thus  $a_i$ .

In this appendix the program's user interface (input and output) and algorithm are provided.

The user input parameters for the selected problem is provided through a Visual Basic form. There are two similar VB forms: The first form, as presented in Figure A.1, is for cases 1-4 and the second form, as presented in Figure A.2, is for case 5.

In each form there are several parameters for the user to choose from. The first group of parameters are associated with the problem of interest, such as material properties ( $k_{11}, k_{12}$  and  $k_{22}$ ) and the open angle of the model ( $\omega$ ). The second group

of parameters is for the boundary conditions. The user can choose whether the BCs are such that the EFIF is a polynomial ( $A_{exact}^{\alpha_1}(x_3) = a_0 + a_1x_3 + a_2x_3^2 + a_3x_3^3 + a_4x_3^4$ ) or a non polynomial function ( $A_{exact}^{\alpha_1}(x_3) = \frac{\sin(x_3)}{a-x_3}$ ). For each case the parameters ( $a_0, a_1, a_2, a_3, a_4$  or  $a$ ) can be selected. Another parameter that determines the boundary conditions is the number of shadow functions to include. In cases 1-4 the user can choose between 0, 2 or 4 where in case 5 the user can choose any number of shadow functions between 0 to 3. The third group of parameters is related to the required numerical accuracy where the user can choose the polynomial degree of finite element approximation, the number of points in the Gauss quadrature integration and also the number of dual shadow functions to use for  $J[R]$  integration. Two more parameters are  $R$  (radius of the circular surface integration) and the degree of the extraction polynomial.

Once all parameters are provided, the bottom "Compute  $A(x_3)$ " is pressed. At this point, the Visual Basic program starts the computation process. A diagram of the computer program is presented in Figure A.3. The program inputs the the user values, computes the main coefficients of the considered case ( $k_1, k_2, \omega_1, \omega_2, \alpha_1$  and  $c_0^{(\alpha_1)}$ ) and loads the Gauss quadrature points. Then it activates StressCheck, loads the required input file and updates the necessary parameters of StressCheck input file (boundary conditions, material properties, open angle of the model and polynomial degree). StressCheck then performs a FE analysis and provides an estimation of the error in  $\tau_{FE}(r, \theta, x_3)$ . After the FE solution is obtained, the first hierarchical polynomials coefficients ( $B(x_3)$ ) are loaded and the first  $J[R]$  computation starts. Since  $J[R]$  is computed numerically, as presented in equation (3.14), the VB computer program extracts the solution  $\tau_{FE}$  at the selected Gauss quadrature points. When  $J[R]$  is computed, the program turns to the next extracting polynomial and computes its related  $J[R]$ . This process repeats itself according to the number of  $a_i$  coefficients specified by the user (the degree of requested polynomial +1). Once all the  $J[R]$ 's

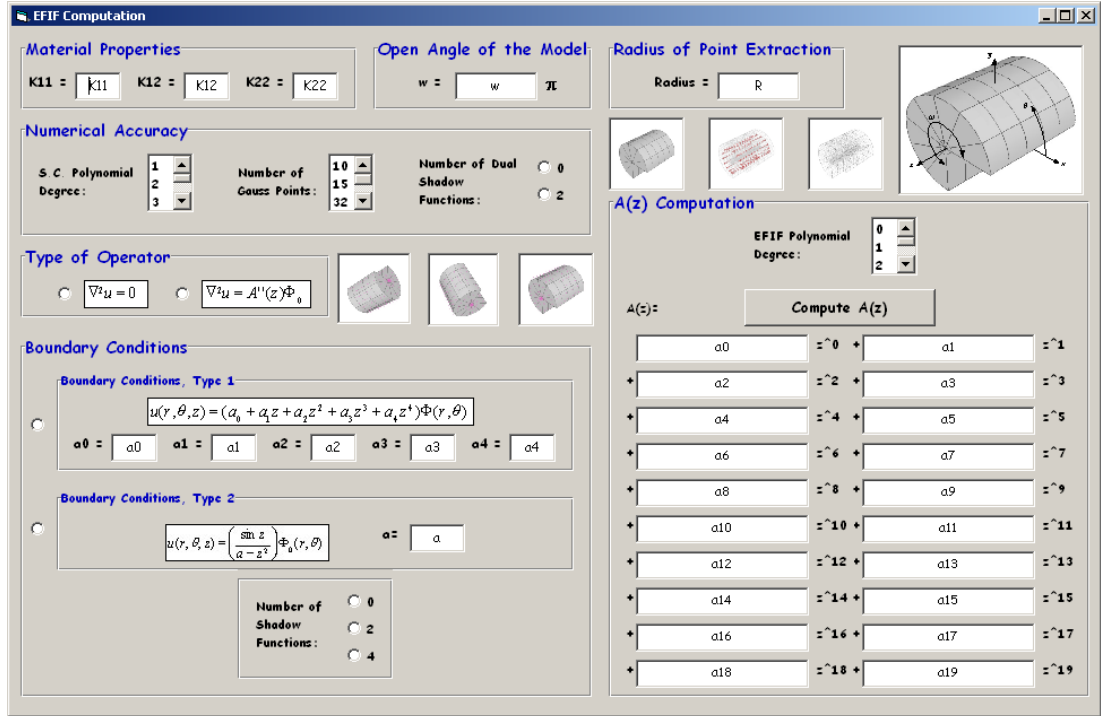


Fig. A.1: Visual Basic Form for cases 1-4, EFIF Computational.

are computed, the coefficients of the EFIF are known. The polynomial of EFIF is a linear combination of the Jacobi polynomials:

$$A(x_3) = \tilde{a}_0 J_0 + \tilde{a}_1 J_1(x_3) + \dots + \tilde{a}_n J_n(x_3)$$

As a last step, the program output the EFIF's coefficients to the VB form.

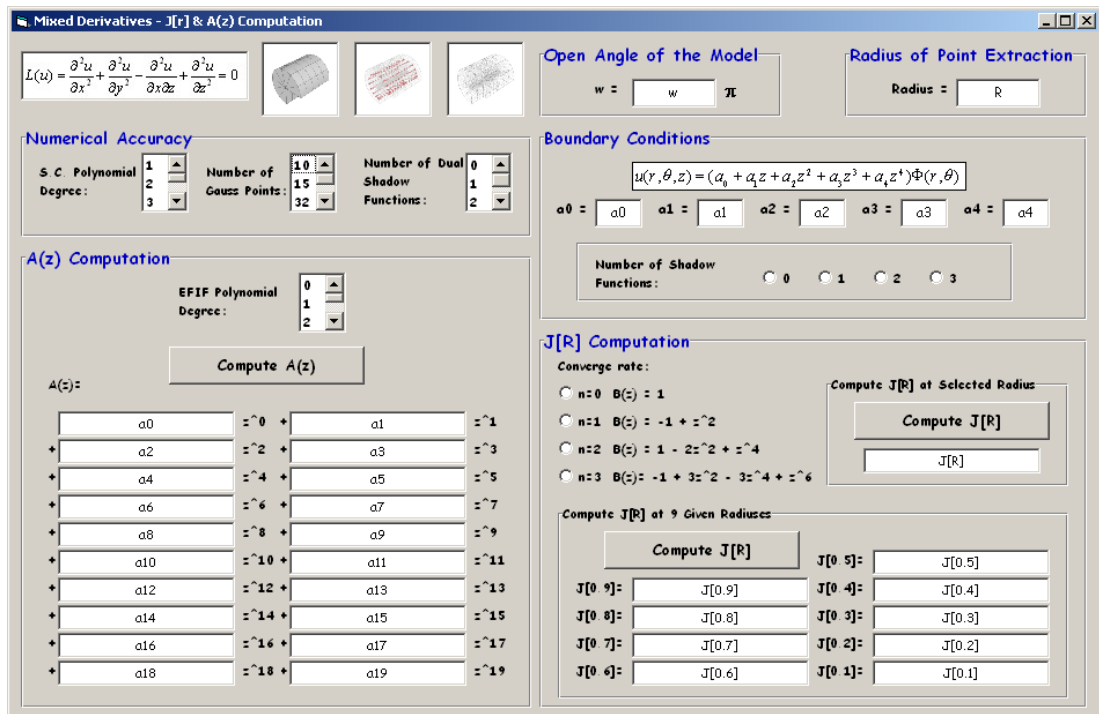


Fig. A.2: Visual Basic Form for case 5, EFIF and  $J[R]$  Computational.

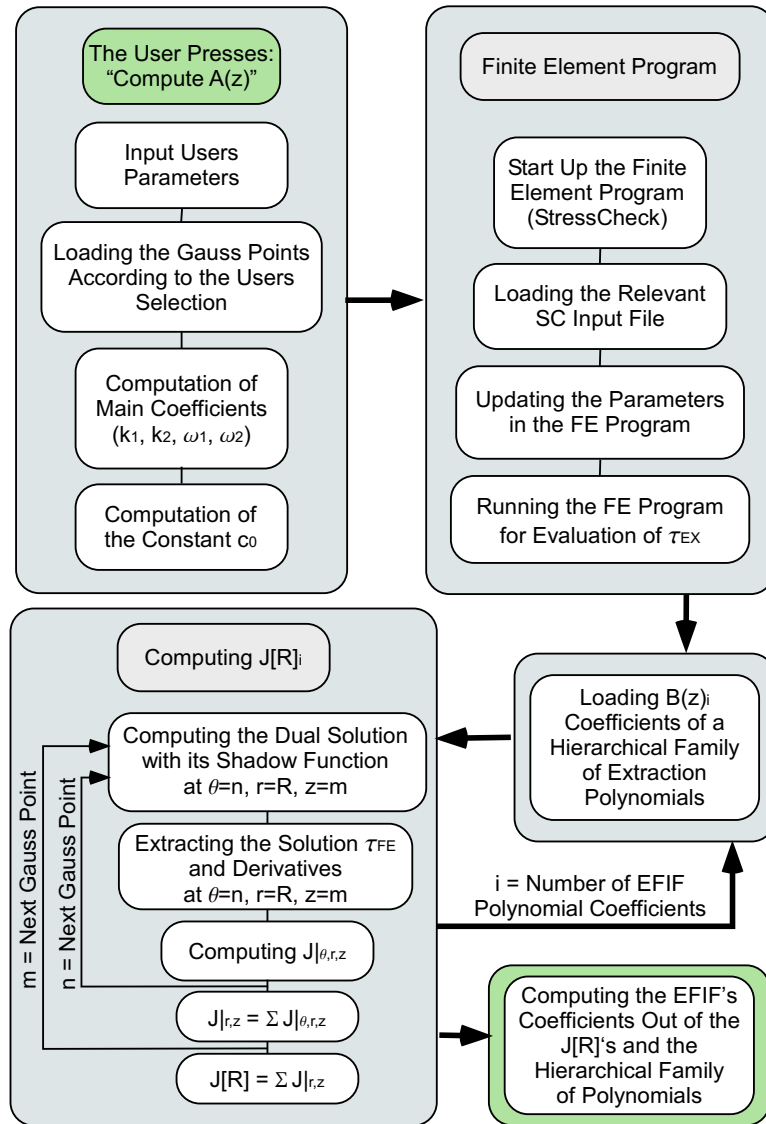


Fig. A.3: Diagram of the Computer Program

## BIBLIOGRAPHY

- [1] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs and mathematical tables*. National Bureau of Standards Applied Mathematics Series, 1964.
- [2] I. Babuška and A. Miller. The post-processing approach in the finite element method - part 2: The calculation of stress intensity factors. *International Journal for Numerical Methods in Engineering*, 20:1111–1129, 1984.
- [3] I. Babuška and M. Suri. The p and h-p versions of the finite element method - an overview. *Computer Methods in Applied Mechanics and Engineering*, 80(1-3):2–26, June 1990.
- [4] L. Banks Sills and Sherer A. A conservative integral for determining stress intensity factors of a bimaterial notch. *International Journal of Fracture*, 115:1–26, November 2002.
- [5] M. Bochniak and A.M. Sändig. Computation of generalized stress intensity factors for bonded elastic structures. *Rairo-Mathematical Modelling and Numerical Analysis*, 33(4):853–878, Jul-Aug 1999.
- [6] W.C. Carpenter. Calculation of fracture machanics parameters for general corner. *International Journal of Fracture*, 24:45–58, 1984.
- [7] W.C. Carpenter. Insensitivity of the reciprocal work contour integral method

- 
- to higher order eigenvectors. *International Journal of Fracture*, 73:93–108, May 1995.
- [8] W.C. Carpenter and C. Byers. A path independent integral for computing stress intensities for v-notched cracks in a bi-material. *International Journal of Fracture*, 35:245–268, June 1987.
- [9] M. Costabel, M. Dauge, and Y. Lafranche. Fast semi-analytic computation of elastic edge singularities. *Computer Methods in Applied Mechanics and Engineering*, 190:2111–2134, 2001.
- [10] M. Costabel, M. Dauge, and Z. Yosibash. A quasidual function method for extracting edge stress intensity functions. *SIAM Journal of Mathematical Analysis*, in Press, 2003.
- [11] M. Dauge. Elliptic boundary value problem in corner domains - smoothness and asymptotics of solutions. *Lecture Notes in Mathematics 1341*, Springer-Verlag, Heidelberg, 1988.
- [12] ESRD. *User's guide, stresscheck by esrd*. ESRD Inc., St. Louis, USA, first edition, January 2000.
- [13] C.A. Felippa. *Introduction to finite element methods*. University of Colorado, 2001.
- [14] M.J. Forray. *Variational calculus in science and engineering*. McGraw-Hill Inc., New York, 1968.
- [15] P. Grisvard. *Elliptic problems in nonsmooth domains*. Pitman Publishing, London, 1985.

- 
- [16] P. Grisvard. *Singularities in boundary value problems*. Masson Springer-Verlag, France, 1992.
- [17] P. Hunter and A. Pullan. *FEM/BEM notes*. University of Auckland, Apr 2003.
- [18] P.E.W. Labossiere and M.L. Dunn. Calculation of stress intensities at sharp notches in anisotropic media. *Engineering Fracture Mechanics*, 61:635–654, 1998.
- [19] P.E.W. Labossiere and M.L. Dunn. Stress intensities at interface corners in anisotropic bimaterials. *Engineering Fracture Mechanics*, 62:555–575, 1999.
- [20] C. Lanczos. *The variational principles of mechanics*. University of Toronto Press, Canada, Toronto, third edition, 1966.
- [21] D. Leguillon and E. Sanchez-Palencia. *Computation of singular solutions in elliptic problems and elasticity*. John Wiley & Sons, France, Paris, 1987.
- [22] V. Mantic, F. Paris, and J. Berger. Singularities in 2D anisotropic potential problems in multi-material corners. *To appear in International Journal of Solids Structures*, 2003.
- [23] Karp S. N. and Karal F. C. The elastic-field behavior in the neighborhood of a crack of arbitrary angle. *Communications in Pure and Applied Mathematics*, XV:413–421, 1962.
- [24] S.S. Pageau and S.B. Biggers. A finite element approach to three-dimensional singular stress states in anisotropic multi-material wedges and junctions. *International Journal of Solids and Structures*, 33(1):33–47, 1996.
- [25] A. Rössle. Corner singularity and regularity of weak solutions for the two-dimensional Lamé equations on domains with angular corners. *Journal of Elasticity*, 60:57–75, 2000.

- 
- [26] B. Szabo and I. Babuška. *Finite Element Analysis*. John Wiley & Sons, New York, 1991.
- [27] B.A. Szabo and Z. Yosibash. Numerical analysis of singularities in two dimensions. Part 2: Computation of generalized flux/stress intensity factors. *International Journal for Numerical Methods in Engineering*, 39:409–434, 1996.
- [28] M. L. Williams. Stress singularities resulting from various boundary conditions in angular corner of plates in extension. *Journal of Applied Mechanics*, 19:526–528, 1952.
- [29] L. Yijun. *Lecture notes: Introduction to the finite element method*. University of Cincinnati, 2003.
- [30] Z. Yosibash. Computing edge singularities in elastic anisotropic three-dimensional domains. *International Journal of Fracture*, 86:221–245, 1997.
- [31] Z. Yosibash. Numerical analysis of edge singularities in three-dimensional elasticity. *International Journal for Numerical Methods in Engineering*, 40:4611–4632, 1997.
- [32] Z. Yosibash. Computing singular solution of elliptic boundary value problems in polyhedral domains using the p-fem. *Applied Numerical Mathematics*, 33:71–93, 2000.
- [33] Z. Yosibash, R. Actis, and B.A. Szabo. Extracting edge flux intensity functions for the Laplacian. *International Journal for Numerical Methods in Engineering*, 53(1):225–242, Jan 2002.
- [34] Z. Yosibash and B.A. Szabo. Generalized stress intensity factors in linear elastostatics. *International Journal of Fracture*, 72:223–240, 1995.

- 
- [35] Z. Yosibash and B.A. Szabo. Numerical analysis of singularities in two-dimensions. Part 1: Computation of eigenpairs. *International Journal for Numerical Methods in Engineering*, 38:2055–2082, 1995.