11-8-2010

Development of Meshfree method for Certain Engineering Analysis Problem

Sunil Kumar Pasupuleti
Florida International University, spasu001@fiu.edu

DOI: 10.25148/etd.FI10121002

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

DEVELOPMENT OF MESHFREE METHOD FOR CERTAIN ENGINEERING ANALYSIS PROBLEM

A thesis submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE

in

MECHANICAL ENGINEERING

by

Sunil Kumar Pasupuleti

2010

To: Dean Amir Mirmiran
College of Engineering and Computing

This thesis, written by Sunil Kumar Pasupuleti, and entitled Development of Meshfree method for Certain Engineering Analysis Problem, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

Sabri Tosunoglu

Chunlei Wang

Igor Tsukanov, Major Professor

Date of Defense: November 8, 2010

The thesis of Sunil Kumar Pasupuleti is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Interim Dean Kevin O' Shea
University Graduate School

Florida International University, 2010

DEDICATION

I dedicate this thesis to my parents and advisor. Without their patience, sacrifice and support the completion of this work would not have been possible.

ACKNOWLEDGMENTS

I would like to convey my gratitude and acknowledge my advisor Dr. Igor Tsukanov, whose valuable suggestions guided me through the thesis process. He has been a constant source of inspiration who provided me with confidence and enthusiasm throughout my work. With his continuous technical and financial support, it has been possible for me to complete my thesis. I highly appreciate the contributions of Dr.Tosunoglu and Dr. Chunlei Wang, for giving me valuable suggestions and for spending their valuable time on my thesis work.

I wish to individually thank all of my friends who helped me in my entire thesis progress. Some of them are Arunav Mokkapati, Sudhir Posireddy, Laxmi Ganesh Sunkara, Manoj Seetharama Reddy, Anil Nimmala and all my room mates, Who provided me with their strength and encouragement.

I am also thankful to the Mechanical and Material Science Engineering Department of Florida International University, which has provided me with a diverse course curriculum. I would like to thank the writing staff who helped me with their suggestions. I am deeply indebted to my family members for their ceaseless inspiration and constant encouragement.

ABSTRACT OF THE THESIS

DEVELOPMENT OF MESHFREE METHOD FOR CERTAIN ENGINEERING ANALYSIS PROBLEM

by

Sunil Kumar Pasupuleti

Florida International University, 2010

Miami, Florida

Professor Igor Tsukanov, Major Professor

This study presents a numerical technique that enables exact treatment of all boundary conditions including those that are given on the interface boundary of two distinct media. This interface boundary conditions for Poisson equation are formulated as equality of the physical field and fluxes across the interface boundary.

In this work first, the range of physical and geometric parameters which allow the applicability of the meshfree method with distance fields are tested and compared with analytical solution. Second, it investigates how the solution error depends on the ratio of B-spline support and thickness of the interface layer. Further, this study also concentrates on developing improved computational tools like 1D integration and modification of distance fields for analysis of diffusion concentration in heterogeneous material with high contrast of physical and geometrical properties. These improved computational tools for meshfree method with distance fields improves the accuracy of solution and decreases the computational time. Finally, these improved tools are used to solve a 2D problem for analysis of diffusion concentration and the results are compared to FEM solution to show that the improved tools yield computationally better results.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

FIGURE PAGE

Chapter 1

INTRODUCTION

1.1 Role of Heterogeneous Materials in Modern Engineering

Engineering analysis involves application of scientific and mathematical principles to know the behavior of an engineering system. In recent years engineering analysis has become a prominent part in the design process to ensure or determine the sustainability of design. This work describes the analysis of physical fields in piecewise homogeneous materials, which are special case of heterogeneous materials. Such materials are combinations of two or more materials, which are bonded together to increase the capability of the product for the desired application.

In modern engineering world, there are many applications that use combination of different materials. For example, Figure 1.1 illustrates a leading application in the field of manufacturing medical products. This application relates to the field of bio-engineering that uses implantable enzymatic biofuel cell (EBFC). EBFC is a implantable medical device (IMD), which is used to develop power for micro electromechanical systems (MEMS) and micro-electronic circuits for extended period of time within human body [1]. EBFC is composed of heterogeneous materials, where very thin layers of highly insulated materials (enzymes) are integrated to a carbon electrode using electro polymerization technique [17].

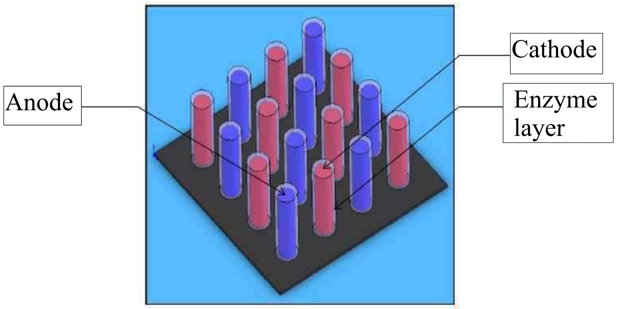


Fig. 1.1: Enzymatic biofuel cell (EBFC)

A prototype model of EBFC has high contrast of geometric parameters and material properties [21]. For example, a typical EBFC in Figure 1.2 has the following dimensions with the height of electrodes as $120\mu m$, diameter $20\mu m$, width $40\mu m$, and the enzyme-layer thickness $10\mu m$, which shows that the enzyme layer is several times thinner than the electrode. In this example, highly dense 3D carbon micro-electrode array chips in Figure 1.2 are used. The geometry of this prototype is assumed to be cylindrical. The bio fuel cell with “3D high aspect ratio carbon electrode arrays structures” can be fabricated by integrating lithography and pyrolysis processing approaches [10, 17]. Carbon is selected as an electrode material because of its attractive features such as low cost, good bio compatibility, a wide working potential window, chemical stability under strong acidic and alkaline conditions, and also easy strategies for surface modifications [30]. The base used is SiO2 layer. In this case, implantable EBFC utilizes the glucose from blood as fuel as it is assumed to be placed inside a blood artery of the human body. Glucose oxidase (GOx) is immobilized as enzyme on anode and laccase is immobilized as enzyme on cathode to catalyze oxidation and reduction reactions respectively. This type of combination of materials and high difference in geometric ratio increases the complexity to analyze physical fields. The current analysis techniques faces various challenges to analyze such combinations for example discontinuity of fluxes at inteface boundary .

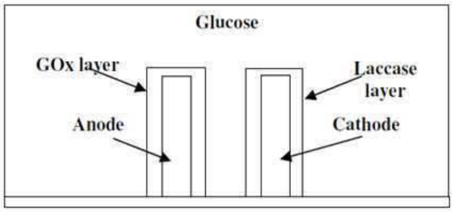
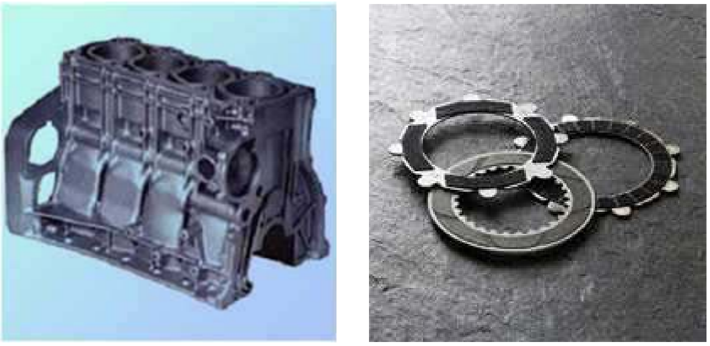


Fig. 1.2: Typical enzymatic biofuel cell (EBFC) with different sets of materials

The application of piecewise homogeneous materials is diverse in the engineering field. Other emerging fields are thermal spray coatings and plasma spray coatings [7]. These coatings are routinely used where there is extensive wear or high temperature damage like in the case of power plant turbines, clutch plates, automotive cylinder bores, on pulp rolls in the paper industry and in many other applications as shown in Figure 1.3. The function of these coatings vary based on the materials used. For example, materials used in coating of automotive cylindrical bore are Aluminium Silicon-Cast alloys, various cast irons, steel, Magnesium-Cast alloys. The functions of the coating for the automotive cylindrical bore are reduction of weight, low oil consumption, resistance to corrosion, increase in efficiency by friction reduction and to avoid bore polishing in heavy duty diesel engines with exhaust gas recirculation.



(a) (b)

Fig. 1.3: Engineering applications of heterogeneous materials [24]. (a) Cylindrical bore coated with Plasma spray coatings; (b) Clutch plates coated with thermal spray coatings

Other extensive applications of heterogeneous materials are in the domain of commercial aircraft. The use of heterogeneous materials increases manufacturers performance and economy. The characteristics of heterogeneous materials are:

- Weight saving leads to fuel saving, increase in payload, increase in range which improves performances.

- Good fatigue resistance leads to enhanced life which involves savings in the long-term cost of the product.

- Good corrosion resistance means fewer requirements for inspection which results in savings on maintenance cost.

Analysis of physical fields in any material is one of the key steps in design process and plays a major role in the study of engineering problems. Engineering analysis provides the solution for engineering problems by using mathematics and principles of science. Physical phenomenon can be studied by conducting physical experiments and by numerical simulation. Numerical simulation and physical experiments provide similar solutions, but cost of physical experiments can be very high.

In this work, numerical simulation technique is used for analysis of physical fields in piecewise homogenous materials with high contrast of geometric and physical parameters. Using numerical simulation, one can model the physical process on a computer without doing physical experiment. The evolution of the system also obeys the physical laws that govern the real physical processes in the simulated region. Hence, the result of such simulation can provide similar representation of the real environment and thus a conclusion can be drawn for a better understanding of the system. This simulation methods are divided into two categories — methods of continuum mechanics (mesh-based and mesh free techniques) and methods of molecular dynamics.

The mesh-based techniques are predominated by the Finite Element Method (FEM) [11] and Boundary Element Method (BEM) [6]. These methods are popularly used for finding approximate solution of partial differential equations as well as integral equations. The core of the finite element method is to discretize the complex domain into a number of reasonably "good" elements to approximate the solution of the problem. The discretization of the domain into small elements is usually referred to as a mesh generation. The solution approach is based either on eliminating the differential equations completely or rendering the partial differential equation into an approximating system of ordinary differential equation, which is then numerically integrated using standard techniques such as Euler's method, Runge-kutta method etc. Main problem encountered using finite element method in analysis of physical fields with high geometric contrast materials is that the mesh has to accommodate for thin structures as shown in Figure 1.4, which results in denser mesh. This may lead to billions of tiny finite elements that requires significant amount of computational resources and time to solve the problem. In addition to this, FEM can guarantee continuity of a physical fields on the interface boundary while fluxes might be discontinuous.

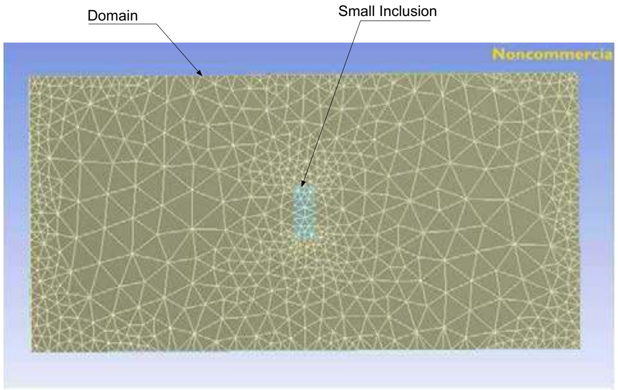


Fig. 1.4: Denser mesh in thin structure

Other successful techniques in numerical simulations of nano structures are ab initio methods. These methods are most accurate and precise of all the currently available methods in molecular modeling. ab intio methods are based on the concept of model chemistry, which has two components: the specific theory [Hartree-Fock (HF) Self-Consistent Field (SCF) Theory] being used, and the specific basis set that is being used as the starting point of calculation [6]. A few limitations make ab initio method less desirable. First, it is expensive, and these methods are typically limited to molecules of 50 atoms or less. Even for small molecules, the user must have access to some reasonably significant computing power.

To overcome problems with mesh generations, meshfree methods were developed. These methods does not necessarily require a spatial discretization that conforms to the shape of the geometric domain as shown in Figure 1.5. Instead, they discretizes underlaying functional space.

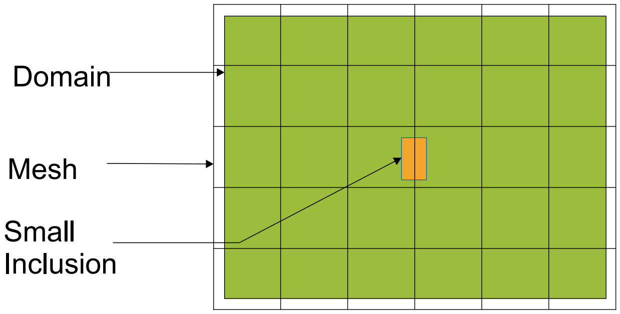


Fig. 1.5: Mesh generated using meshfree method

1.2 Introduction to meshfree methods

Over the last forty years, many research works were carried out and are still under progress in the field of mesh free analysis. The development of these methods took place in late 1970s and then after number of techniques with basis functions that do not have to confirm the geometry of the domain have been developed: Smooth particle hydrodynamics (SPH)[9, 20], the diffuse element method (DEM) [3], the reproducing kernel particle method (RKPM) [29, 14], the HP cloud method [5], the meshless local Petrov-Galerkin (MLPG) [23], and the partition of unity method (PUM) [19], extended partition of unity method (PUFEM) [18], interface element method (IEM) [16], boundary element free method (BEFM) [15], meshless local Petrove-Galerkin based on Rankine source solution method (MLPGR) [22] and many more. All these methods faced difficulties in satisfying boundary conditions. To overcome this problem, a meshfree method with distance fields [27, 26] has been proposed, which has the ability to treat boundary conditions exactly.

1.2.1 Meshfree method with distance fields

The idea of this method is based on the observation that the solution of a differential equation with boundary conditions $C|_{\partial\Omega} = 0$ can be represented in the form

$$C = \omega\Phi \tag{1.1}$$

where ω is a known function that takes on zero values on the boundary of the domain $\partial\Omega$, and is positive in the interior of Ω, and Φ is some unknown function. For example, consider a circular plate with geometry shown in Figure 1.6 to explain the construction of global solution structure for meshfree methods with distance fields. The temperature distribution is given by Laplace equation $-\nabla^2 C =$ *f(x,y).* The plots for this global function ω(x,y) is shown in Figure 1.7 on the left, which is identically zero on the boundary of a two dimensional domain in Figure 1.6 and is positive in the domain's interior. As such, ω completely describes all the geometric information for

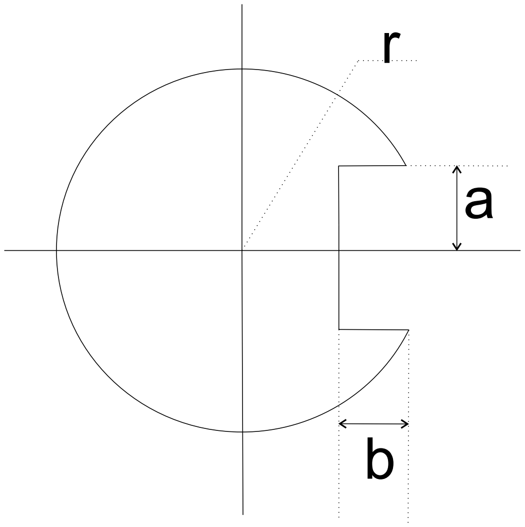


Fig. 1.6: Geometric domain defined for heat transfer analysis

the homogeneous Dirichlet boundary value problem, and in face any function u of the form in expression (1.1) will satisfy the zero boundary conditions exactly.

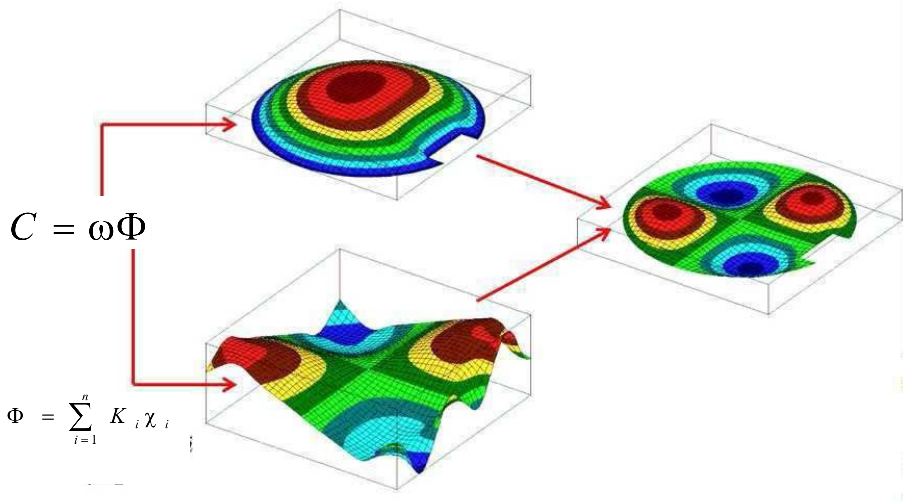


Fig. 1.7: Construction of solution structure for Dirichlet boundary conditions

Expression (1.1) contains no information about the differential equation of the boundary value problem. Rather, it represents the structure of any solution to a boundary value problem satisfying the given boundary conditions. For any given boundary value problem, determination of the unknown Φ immediately translates into solution to the boundary value problem. Since we usually cannot expect to determine such Φ exactly, we can approximate it by a finite (convergent) linearly- independent series.

$$\Phi = \sum_{i=1}^{n} K_j \chi_j \tag{1.2}$$

where K_j are scalar coefficients and χ_j are some basis functions, Kantorovich relied on the standard global polynomial basis, but shown in the center Figure 1.7 is the combination of the function ω with a two-dimensional uniform 30 × 30 rectangular grid of bicubic B-splines χ_i.

It is important that the structure shown in expression (1.1) does not place any constraints on the choice of the basis functions χ_j that approximate the function Φ. In particular, the choice of the basis functions does not depend on any particular spatial discretization of the domain. The grid of B-splines in our example is aligned with the space and not with the domain. For any given boundary value problem and a choice of the basis function χ_j, the approximate solution is obtained as:

$$C = \omega \sum_{i=1}^{n} K_j \chi_j \tag{1.3}$$

Using variational, projection or a variety of other numerical methods to solve for the numerical values of the coefficients K_j. For example, if we choose the coefficients to approximate the solution of the differential equation $\nabla^2 C = 1 - \mathit{sin(y)}$ in the least square sense, we obtain the function u as shown in the above Figure 1.7 on the right. Some other types of boundary conditions are shown in expressions (1.4, 1.5). First expression represents the physical field on the boundary $\partial\Omega$, which is equal to φ and the other represents the convective heat exchange on the boundary $\partial\Omega$, which is equal to ψ.

$$C|_{\partial\Omega} = \varphi \tag{1.4}$$

$$\left(\frac{\partial C}{\partial n} + hC\right)|_{\partial\Omega} = \psi \tag{1.5}$$

Using meshfree method with distance fields, the solution of differential equation with above mentioned boundary conditions can be represented in the form shown in expressions (1.6 and 1.7) respectively.

$$C = \omega\Phi + \varphi \tag{1.6}$$

$$C = \Phi - \omega D_1^{\omega}[\Phi] - h\omega\Phi - \omega\psi + \omega^2\Phi \tag{1.7}$$

1.3 Focus of the thesis

This work demonstrates feasibility of meshfree analysis in heterogeneous material with high contrast of geometric and physical parameters and also investigates how the solution error depends on the ratio of B-spline support and thickness of the interface layer. The primary focus is to develop a computational infrastructure to support meshfree analysis of micro and nano structures. Tsukanov et al, demonstrated applicability of this meshfree approach for structures with a ratio of physical and geometric parameters up to 1:100. Thus, this work focuses on testing bigger ratios and comparing them with analytic solution. This research is expected to portray the range of physical and geometric parameters which allow applicability of the meshfree method with and without computational tools for meshfree analysis in heterogeneous material.

Chapter 2

MESHFREE ANALYSIS OF PHYSICAL FIELDS IN HETEROGENEOUS MEDIA WITH HIGH CONTRAST OF GEOMETRIC AND PHYSICAL PARAMETERS

2.1 Solution structure satisfying both external and interface boundary conditions

Mathematical model describing the process of physical fields such as thermal conductivity, diffusion in piecewise heterogeneous materials is defined by Fourier equation. For convenience, consider Fourier equation for Diffusion

$$grad(\lambda_i div(C)) = [c\rho]_i \frac{\partial C}{\partial \tau}, \quad i = 1, 2, ..., n, \tag{2.1}$$

With the initial and the boundary conditions of the form [2]

$$C|_{\tau=0} = C_0; \quad \lambda_i \frac{\partial C}{\partial n}|_{\partial\Omega_1} = \alpha_i [C - C_s], \quad i = 1, 2, ..., n, \tag{2.2}$$

where λ_i is the diffusion coefficient, C_s is the medium concentration, α_i is the transfer coefficient. The expression (2.2) implies that the initial condition at time $\tau = 0$ is C_0 and the diffusion flux at particular boundary $\partial\Omega_1$ depends on the difference between concentration at that boundary and the medium concentration. Apart from, on the boundary of contact of the product the following interface conditions must be satisfied.

$$C_i|_{\partial\Omega} = C_j|_{\partial\Omega}; \quad \lambda_i \frac{\partial C}{\partial n}|_{\partial\Omega} = \lambda_j \frac{\partial C}{\partial n}|_{\partial\Omega}, \quad i \neq j. \tag{2.3}$$

The expression (2.3) implies that the concentration and there fluxes are equal at the interface boundary $\partial\Omega$.

2.1.1 Generalised solutionn structure for meshfree method with distance fields

To solve boundary value problems, meshfree method with distance fields technique is used. General idea of this approach consists in the solution of boundary value problem in sought in the form of the so-called solution structure:

$$C = B[\Phi, \omega_0, \varphi, \lambda] \tag{2.4}$$

which at any selected indefinite component $\Phi = [\Phi_i]_{i=1}^{m}$ takes given on values on the boundaries of the object (they are determined by function φ) and /or exactly conditions of a differential character. Here B is an operator depending on the operator of the boundary conditions defined on each separate sections $\partial\Omega_i$.
The sequence of coordinate function necessary for solving problem is obtained when approximate indefinite component of the structure by means of some full sequence $[\chi_j]_{j=1}^{n}$.

$$\Phi = \Sigma_{j=1}^{n} K_j \chi_j. \tag{2.5}$$

where K_j are unknown coefficients, which can be determined by using classic variational and projection methods, viz. Solution structure satisfying boundary conditions (2.3) is of the form [2]:

$$C_i = B[\Phi, \omega_0, \varphi, \lambda] + \Sigma_{j=1}^{n} (\omega_0^2 \bigcap \omega_j) D_1^{(i)} (B[\Phi, \omega_0, \varphi, \lambda]) \left(\frac{M_{ij}}{\lambda_i} - 1 \right), \quad i = 1, 2, 3...m, \tag{2.6}$$

where ω_j and M_{ij} are defined by
$\omega_j(M)|_{M\epsilon\Omega_j} \neq 0$; $\omega_j(M)\ |_{M\epsilon\neq\Omega_j} = 0$; $\frac{\partial\omega j}{\partial n}\ |_{\partial\Omega j} = 1$; $M_{ij} = \frac{2\lambda_i\lambda_j}{(\lambda_i + \lambda_j)}$

2.2 Exact solution for a benchmark problem

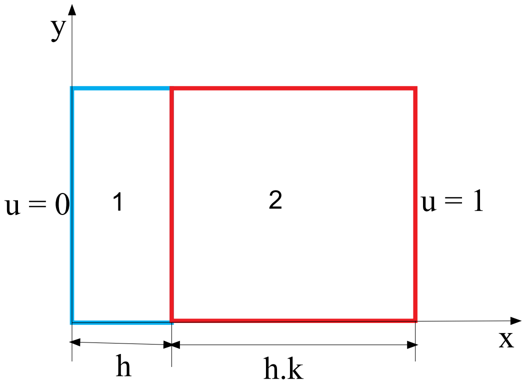


Fig. 2.1: Modeling of physical fields in two different materials

To demonstrate feasibility of meshfree methods in heterogenous material with high contrast of geometric and physical parameters, consider a geometric domain shown in Figure 2.1. This domain consists of two parts with different material properties. The basis of considering this domain is because of the exact solution, which helps to compare with the approximate solution of meshfree methods. Boundary conditions for the domain are prescribed as follows.

$$C_1|_{x=0} = 0 = \varphi_1, \tag{2.7}$$

$$C_2|_{x=h(k+1)} = 1 = \varphi_2, \tag{2.8}$$

The boundary conditions (2.7) and (2.8) define the two opposite sides of the domain. These boundary conditions prescribes concentration on left and right edges of the domain, which are equal to zero and one respectively. The boundary conditions prescribed on the interface boundary are concentration and flux, which are equal on either sides of the interface boundary:

$$C_1|_{x=h} = C_2|_{x=h}, \tag{2.9}$$

$$\lambda_1 \frac{\partial C}{\partial x}|_{x=h} = \lambda_2 \frac{\partial C}{\partial x}|_{x=h}, \tag{2.10}$$

In each sub domain, solution will be represented using piecewise linear combination, which is in the form of:

$$C_i = d_i^1 + d_i^2 x, \quad i = 1, 2; \tag{2.11}$$

where d_1^i and d_2^i are coefficients. Depending on the domain, different coefficients are chosen. This solution satisfies the differential equation exactly. By satisfying the boundary conditions and solving the system of algebraic equations following coefficients are obtained:

$d_1^1 = \varphi_1;$

$d_2^1 = -[\frac{(\lambda_2(\varphi_1-\varphi_2))}{(h(k\lambda_1+\lambda_2))}];$

$d_1^2 = [(\lambda_1\varphi_1 + k\lambda_1\varphi_1 - \lambda_1\varphi_2 + \lambda_2\varphi_2)/(k\lambda_1 + \varphi_2)];$

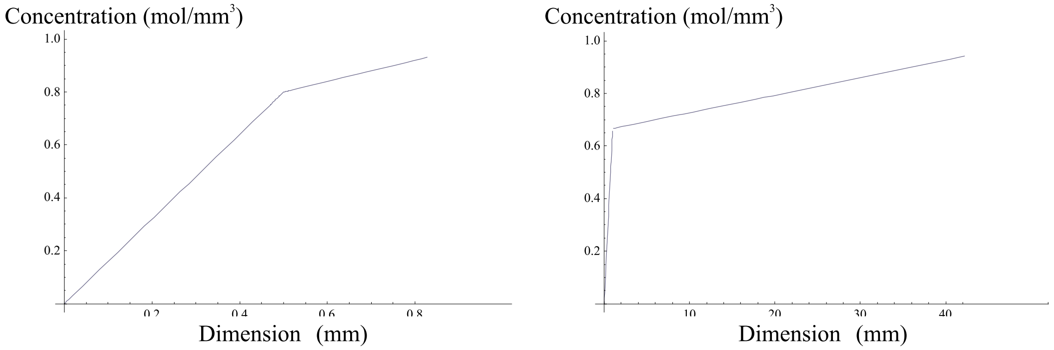
$d_2^2 = [(-\lambda_1\varphi_1 + \lambda_1\varphi_2)/(h(k\lambda_1 + \lambda_2))].$

For example consider modeling of concentration distribution in two different materials with following two sets of dimensions and physical parameters:

1) $h = 0.5, \quad k = 1$ and diffusion coefficients $\lambda_1 = 50 \ m^2/s, \quad \lambda_2 = 200 \ m^2/s;$

2) $h = 0.02, \quad k = 49$ and diffusion coefficients $\lambda_1 = 1 \ m^2/s, \quad \lambda_2 = 100 \ m^2/s$

Results obtained by using exact solution is show as a plot in Figure 2.2.



(a) (b)

Fig. 2.2: Plot for exact solution (a) Parameters h = 0.5, k = 1; $\lambda_i = 50, \lambda_j = 200$ (b) Parameters h = 1, k = 50; $\lambda_i = 1, \lambda_j = 100$

2.3 Preliminary numerical solution of a benchmark problem — comparing numerical solution and exact solution

Consider a detailed example as shown in Figure 2.1 for comparing numerical solution and exact solution. This example consists of two thin layers of different materials bonded together. Notations used in this example are h, λ_1 and $h \times k$, λ_2, which represents the thickness, diffusion coefficient of first and second layers respectively. For this example the boundary conditions prescribed on the domain are given by

$$\begin{aligned} C_1|_{x=0} &= 0 = \varphi_1 \\ C_2|_{x=h(k+1)} &= 1 = \varphi_2 \\ C_1|_{x=h} &= C_2|_{x=h} \\ \lambda_1 \frac{\partial C}{\partial x}|_{x=h} &= \lambda_2 \frac{\partial C}{\partial x}|_{x=h} \end{aligned} \tag{2.12}$$

To study diffusion (concentration gradient) in this example, meshfree methods with distance fields is used. The numerical solution is represented using new basis functions. These functions are created as the product of the function ω and the FEM basis functions Φ.

For any kind of given boundary value problem with inhomogenous Dirchelet boundary conditions (2.13) and with selected basis functions χ_i, the approximate solution is given by

$$C = \omega\Phi + \varphi \tag{2.13}$$

The function ω in the expression (2.13) is constructed by applying R-conjunctions to ω_1 and ω_2. Let $\omega_1 = x$ and $\omega_2 = h(k+1) - x$ be functions of distances to the boundary points $x = 0$ and $x = h(k+1)$ respectively. These are simply linear rays inclined at 45^0 to the x-axis, which is given by formula:

$$\omega = \omega_1 + \omega_2 - \sqrt{\omega_1^2 + \omega_2^2} \tag{2.14}$$

The resulting function behaves as an approximate distance point at $x = 0$ and $x = h(k+1)$ as shown in Figure 2.3.

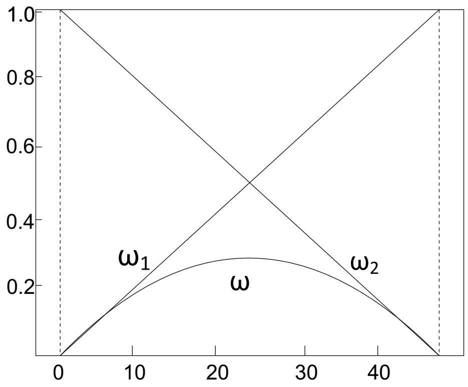
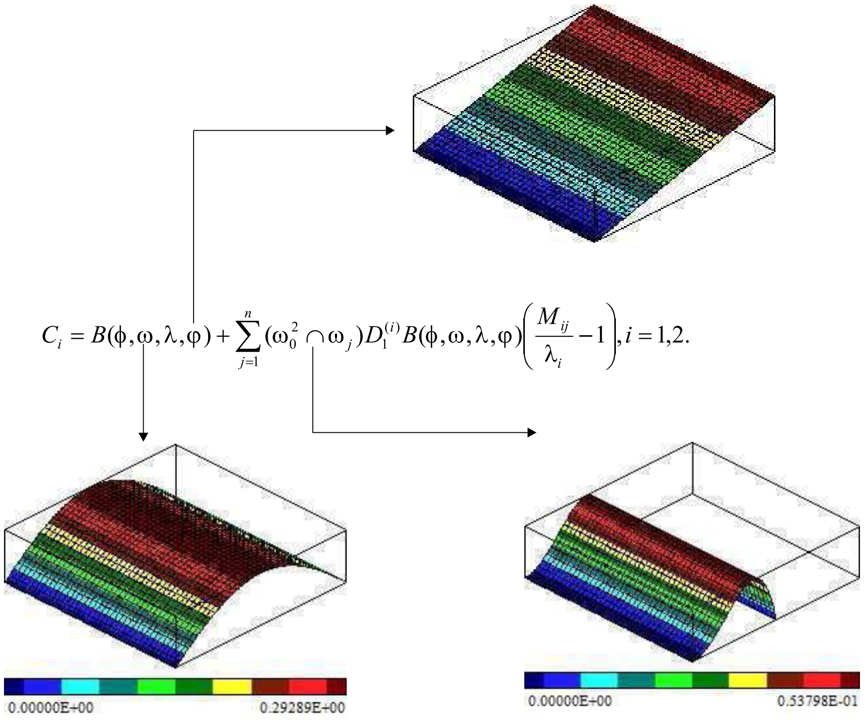


Fig. 2.3: Distance fields

The global boundary function φ is constructed using transfinite Lagrangian interpolation [25, 27]. Using distances $x = 0$, $x = h(k+1)$ and $x = h$, function φ is constructed, that interpolates the prescribed boundary conditions at the end points $x = 0$, $x = h(k+1)$ and $x = h$. This φ is added to the linear combination of the newly constructed basis functions, which results in a representation for diffusion concentration that satisfies the boundary conditions at the two edges of the domain, and is given by

$$\varphi = \frac{\omega_1 c_2 + \omega_2 c_1}{\omega_1 + \omega_2} \tag{2.15}$$

Figure 2.4 presents the plots of global functions ω and φ:

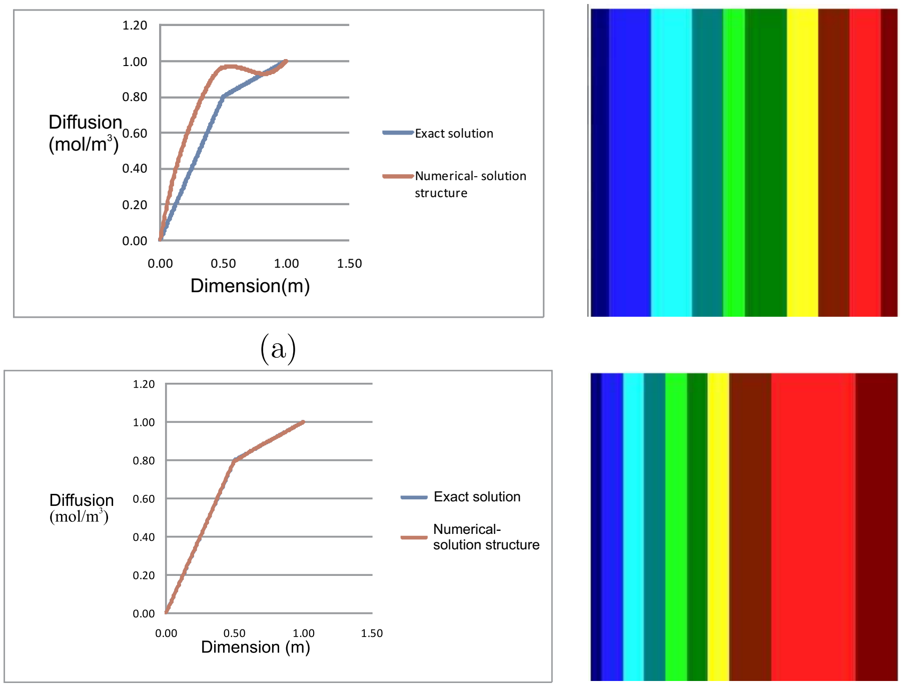


$$C_i = B(\phi,\omega,\lambda,\varphi) + \sum_{j=1}^{n} (\omega_0^2 \cap \omega_j) D_1^{(i)} B(\phi,\omega,\lambda,\varphi)\left(\frac{M_{ij}}{\lambda_i} - 1\right), i = 1,2.$$

Fig. 2.4: Construction of solution structure for Dirchlet inhomogeneous boundary conditions

In the introduction part it is concluded that FEM requires more number of supports to analyze physical fields in thin regions. In this section the results given by meshfree method are compared to exact solution (2.2). This type of comparison is selected to determine the number of supports that meshfree method with distance fields will require to yield better solution. This comparison is done using two sets of parameters, which are named as benchmark problem - 1 and benchmark problem - 2.

2.3.1 *Domain with equal thickness problem*



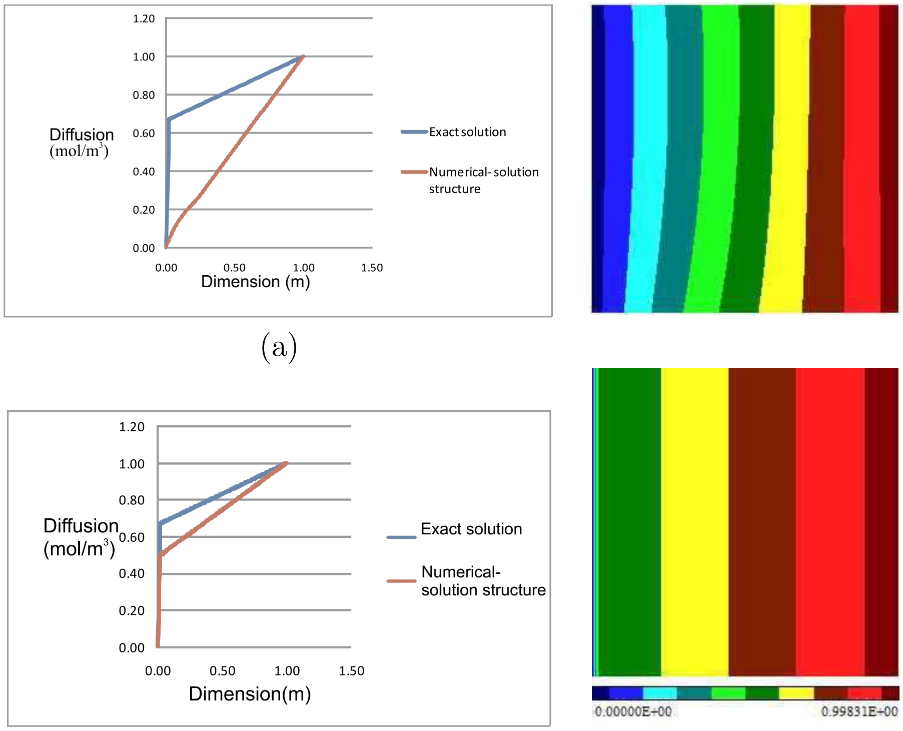
(a)

(b)

Fig. 2.5: Plots for parameters h = 0.5, k = 1; λ_i = 50, λ_j = 200 and effect of varying number of B-spline supports in the thin layer: (a) one (2×2 size grid) and its concentration distribution (b) two (5×5 size grid)and its concentration distribution

Using this benchmark problem, we modeled concentration distribution in a two layer domain. The thickness ratio between the two layers of the domain is one and diffusion coefficient ratio is four. The objective of this experiment is to find out whether this problem gives satisfying results by using just one B-spline in each layer. For better interpretation, this problem was solved twice. Formerly, the results were plotted for one B-spline in the thin layer as shown in Figure 2.5(a) and later for two B-splines in the thin layer as shown in Figure 2.5(b).

From Figure 2.5, the plot with more number of B-splines coincide with exact solution when compared to the plot with less number of B-splines. For the above mentioned parameters this numerical experiment proves that current implementation of meshfree method will require more than one B-spline in the layer to yield better results.



(a)

(b)

Fig. 2.6: Plots for parameters h = 0.02, k = 49; $\lambda_i = 1, \lambda_j = 100$ and effect of varying number of B-spline supports in the thin layer: (a) less than one (2×2 size grid) and its concentration distribution using isolines (b) one (40×40 size grid) and its concentration distribution using isolines

2.3.2 Domain with thin layer and high gradient problem

This experiment was conducted with a similar setup as that of the benchmark problem-1. The difference is that the ratio between layers for both physical and geometric parameters were increased. Due to this increase in geometric ratio between the layers, one of the layer's becomes thin. And in order to model diffusion profile in this kind of domains, it might require more than one B- splines in the thin region. The objective of this experiment is to test the current implementation of meshfree algorithm for increased ratio of geometric and physical parameters. To better understand the circumstances, this problem was solved twice, one with less than one B-spline and other with one B-spline in the thin region. The results of the experiments are as shown in Figure 2.6(a) and 2.6(b).

Figure 2.6 show major distortion from the exact solution because of less than one B-spline in thin layer. But whereas the plot in Figure 2.6(b) show that the solution improves comparatively due to finer grid, but still fails accommodate gradient in the thin region. So this infers that the number of B-splines involved must be further increased to yield better solution. This eventually results in increase of computational time and cost. By further contemplating through the plots for isolines, it depicts that the isolines are more distorted for plot with less number of basis functions (Figure 2.6(a)). This distortion in isolines is because of numerical integration, which fails to detect small geometric features.

The current integration algorithm is based on geometry adaptation [12]. With respect to the given geometric domain, cells fall into two categories: Internal cells completely situated inside the geometric domain, and boundary cells which enclose portions of the domain's boundary. The simple geometric shape of the internal cells allow direct application of the lattice rules. In contrast, integration in boundary cells uses different approach. The simplest way to determine boundary cells is to check whether the corner points of the cell are inside or outside the geometry. For example, if only one vertex is inside the domain, the boundary cell can be parameterized using any one of the appropriate coordinate systems. Based on the geometry, different coordinate systems are used: Polar coordinate system, spherical coordinate system and cartesian coordinate system. Then the vertex, which is inside the geometry becomes the solo starting for n rays. These rays are arranged radially around the initial edge, according to the chosen Gauss rule within the 90^0 span between the two adjacent cell faces. Each ray is intersected with the boundary and used to place the integration nodes between the cell's vertex and the intersection point as shown in Figure 2.8.

However, if a geometric feature is smaller than a cell, it may not be discovered. One of the way to detect small geometric features is by sampling random points. When totally interior or exterior cells are encountered, the characteristic function is compared at some randomly-placed points as shown in Figure 2.8, to ensure that cell does not contain any undetected voids or features. If such region is detected, the cell is subdivided and integration is retired on each of the subcells. Whenever there is a decrease in thickness of the thin layer, this approach requires increase in sampling points to detect

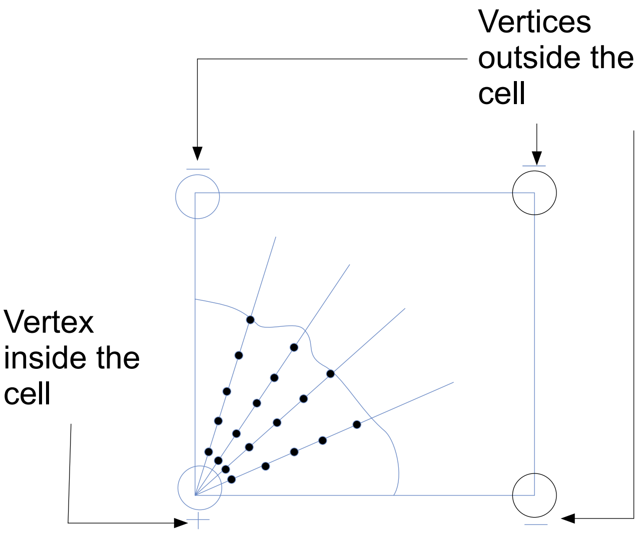


Fig. 2.7: Gauss integration

thin feature, Which again leads to increase in computational time and cost. A similar situation where the sampling of random points approach fails to detect the small layer is shown as distorted isolines in Figure 2.6(b). This failure is due to insignificant or less number of points sampled on the domain.

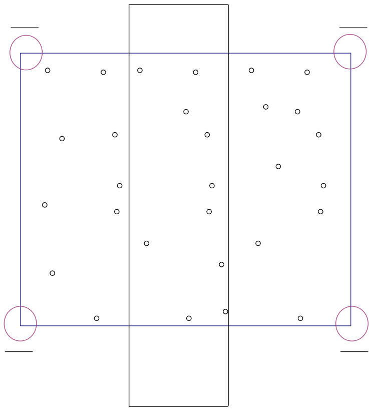


Fig. 2.8: Small features detection by sampling random points

2.4 Conclusions

This chapter was started with the introduction of basic concepts and theory of meshfree method with distance fields and then showed results of exact solution for a benchmark problem with two different set of parameters. In the section 2.3, we studied the results of preliminary numerical experiments for the same set of parameters used in exact solution. The synopsis of the preliminary numerical experiments resulted in following conclusions: Benchmark problem-1 is not considered as a serious problem because both the layers are of equal thickness. Moreover, this problem can be solved by using any commercial software with less computational time and cost. But where as the numerical experiments of benchmark problem- 2 pose a serious concern in solving this problem due to its bigger geometric and physical parametric ratio. The reason for the failure of meshfree algorithm might be due to some deficiency in current integration algorithm, which requires more number of B-splines in the thin layer to yield better solution. Thus, to improve current implementation of meshfree algorithm for bigger ratios, we focus on modifying current integration algorithm.

Chapter 3

IMPROVED COMPUTATIONAL TOOLS

3.1 1D integration

To increase the accuracy of current integration algorithm, there should be either a explicitly finer mesh on thin layer or nodes of the cell should be inside the domain. But if the geometry is complex it is very hard to keep the nodes of the cell inside the domain. Therefore current integration algorithm requires some modifications and improvements.

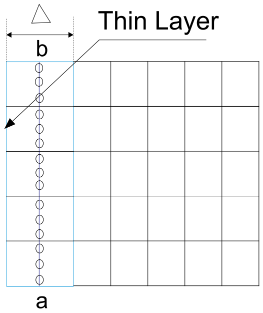


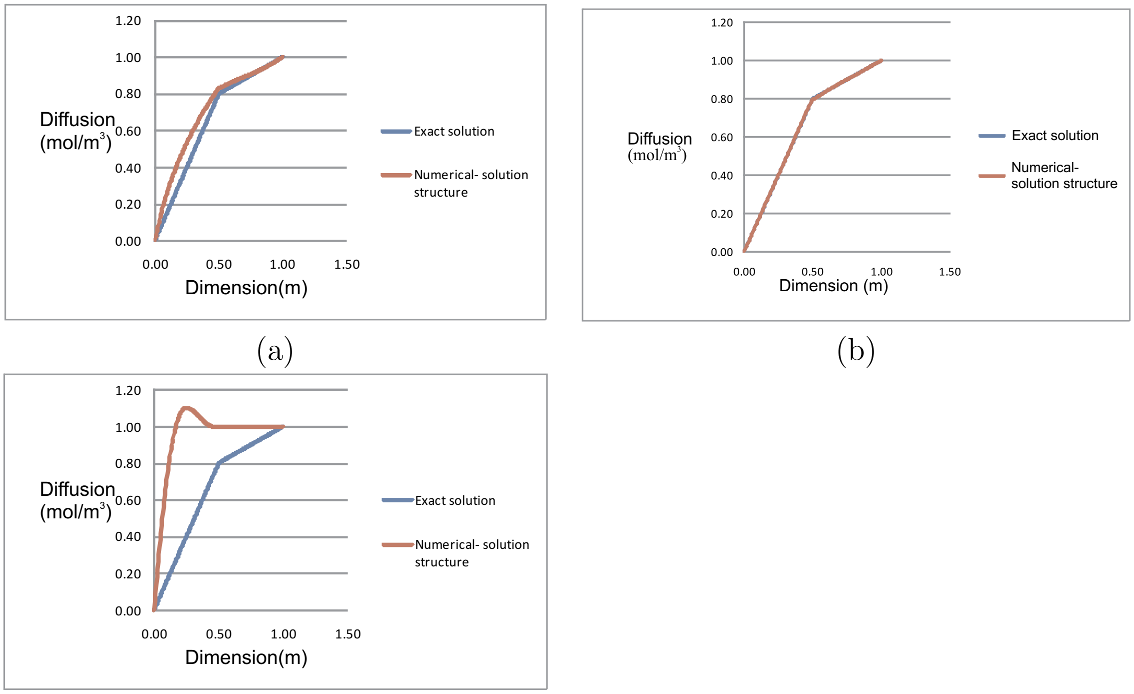
Fig. 3.1: Proposed integration algorithm

Assume that the feature (thin layer) is very thin, then the variations of physical fields are very small along the thickness of the feature. By using this assumption, 2D integration can be represented as integral on boundary, times thickness. So it is proposed to consider a middle line of the thin layer as shown in Figure 3.1 and allocate integration points on that line. Since the thickness of the thin layer (Δ) is known, integration over the the line a to b can be done by multiplying "Δ " with the one dimensional integral.

3.2 Results of solution structure with 1d integration

In this section, the results from meshfree approach using 1D integration algorithm are compared to current implementation of meshfree methods. This type of comparison was opted to demonstrate the feasibility of 1D integration algorithm. This comparison is done using the same set of parameters used in previous chapter 2.2.

3.2.1 Domain with equal thickness problem



(a)

(b)

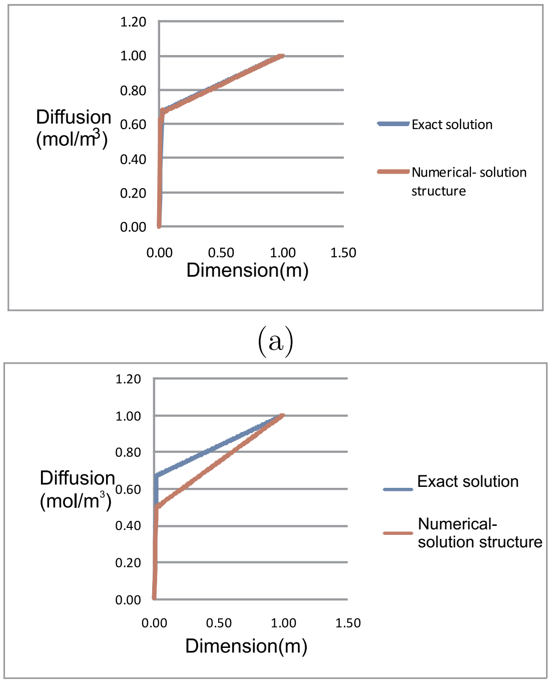
(c)

Fig. 3.2: Plots for parameters h = 0.5, k = 1; λ_i = 50, λ_j = 200 with (a) 2×2 size grid using 1D- integration algorithm (b) 5×5 size grid using current implementation of meshfree algorithm (c) 5×5 size grid using 1D- integration algorithm

This experiment focuses on evaluating and determining the new integration approach for a significantly thick layered domain. So for better analysis, this problem was solved using both 1D integration algorithm and current integration algorithm. The plots for the respective approaches are graphed as shown in Figures 3.2(a) and 3.2(b).

The plot in Figure 3.2 show that the numerical solution fails to coincide with the exact solution for 1D- integration algorithm when compared with current integration algorithm. By analysis, this distortion of numerical solution is due to adequate amount of thickness in the thin layer. This infers that there is a chance for change of physical fields along the horizontal direction of thin region, which conflicts with the assumptions of 1D integration approach. Also, the plot in Figure 3.2(c) show that solution is more deviated when the number of B-splines are more than one in the thin layer. This is due to the assumption used for construction of 1D integration algorithm. This assumption infers that when there are more than one B-spline in the thin region, the 1D integration algorithm allocates integration points on the middle line of the first B-spline of the thin layer and then implies solution structure on these integration points. The other B-splines in the thin layer are considered as insulted. This results in major distortion of solution structure from exact solution.

3.2.2 Domain with thin layer and high gradient problem



(a)

(b)

Fig. 3.3: Plots for parameters h = 0.02, k = 49; $\lambda_i = 1$, $\lambda_j = 100$ with 40 × 40 size grid using- (a) 1D- integration algorithm (b) Current solution structure algorithm

The objective of this experiment is to test the meshfree method with 1D integration algorithm for increased ratio of geometric and physical parameters ratio. For better perception, this problem is solved twice, one using 1D integration algorithm and other with current integration algorithm. Both approaches were performed by just using one B- spline in the thin layer.

The plot in Figure 3.3 infers that the numerical solution fails to coincide with the exact solution for current integration algorithm (Figure 3.3(b)) when compared with 1D- integration algorithm (Figure 3.3(a)). This results in a conclusion that the solution structure with 1D integration works better when the thin layer is very small and there is one B- spline in the thin region. The experimental results are as expected because it is satisfying the assumption that there is no change in physical fields across horizontal direction of the thin layer.

1D integration algorithm works better when the following conditions are satisfied: first, the layer should be sufficiently thin such that there is no change of physical fields in the horizontal direction of the thin layer. Second, the thin layer should be accompanied with in one cell or B-spline as shown in Figure 3.1. Different scenarios expected in 1D integration based on the above mentioned conditions are:

- Scenario 1: The thin layer is accompanied by two cells or B-splines as shown in Figure 3.4. When this type of condition exists then the above mentioned 1D integration fails. In order to overcome this condition, it is proposed to use 1D integration algorithm in each cell as shown in Figure 3.4, since the thickness of the thin layer in each cell is known.

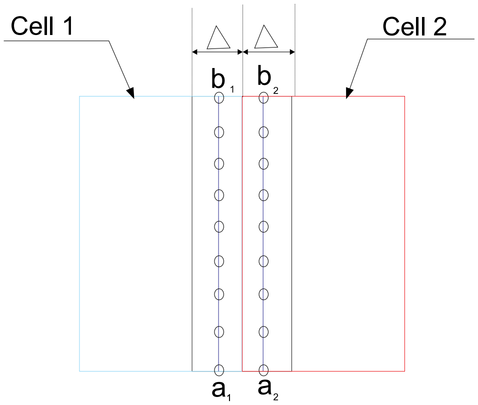


Fig. 3.4: Thin layer accompanied by two cells

- Scenario 2: The thin layer is accompanied by multiple number of cells as shown in Figure 3.5. In this case since the boundaries of the domain are explicitly known, it is proposed to divide the domain into sub domains in each cell and allocate integrations points within each sub domain instead of relying on sampling random points algorithm.

By contemplating through the results of benchmark problem - 2 in section 3.2, the plot shows that even to accommodate one B-spline in thin region it requires grid size

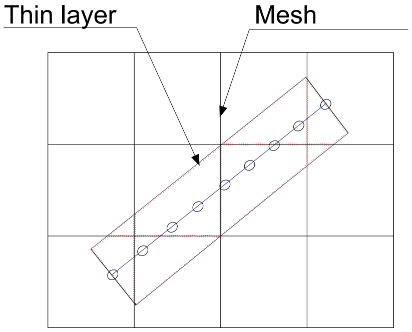


Fig. 3.5: Thin layer accompanied by multiple cells

of 40 × 40, which is obvious because of thin layer.

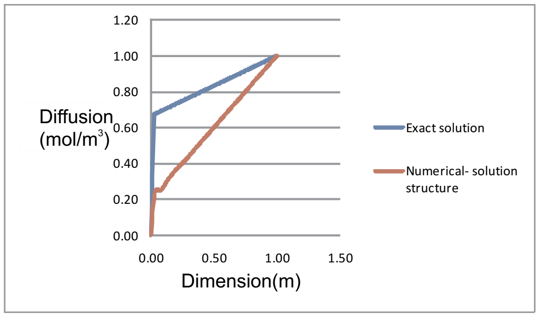


Fig. 3.6: 1D integration for parameters h = 0.02, k = 49; $\lambda_i = 1$, $\lambda_j = 100$ with 10×10 size grid

To better understand the current problems, let us consider the same benchmark problem-2 in section 3.2 with a coarser grid. The objective of this experiment is to test the applicability of 1D integration algorithm for less than one b-spline in the thin layer. By analysis, the plot in Figure 3.6 show that the solution structure fails to accommodate high gradient in thin region when there are no enough B-splines in the thin region. So this results in conclusion that the improved integration algorithm will work fine when there is strictly one B- spline in thin region. In order to accommodate even one B-spline in very thin layers there should be much finer grid, which results in increase of computational time. So this leaves room for further improvement. Since there are problems with high gradient in the thin region, the later section is focussed on accommodating high gradient in the thin region and reducing the computational time.

3.3 Construction of modified distance fields

$$C = \omega\Phi + \varphi \tag{3.1}$$

The approach proposed to accommodate high gradient in thin layer is modification of distance fields. The solution of a differential equation with boundary conditions $C_1|_{x=0} = 0 = \varphi_1$ and $C_2|_{x=h(k+1)} = 1 = \varphi_2$ can be represented in the form as shown in expression (3.1), where ω is a known function that takes on zero values on the boundary of the domain $\partial\Omega$, and is positive in the interior of Ω, and Φ is some unknown function. In order to accommodate high gradient at the thin region, either grid in the thin region must be denser or ω or Φ is to be modified. Opting to make denser grid results in increase of computational time and moreover modification of Φ is also difficult. So modifications of distance fields is the better option for the current scenario. Hence, the distance field ω is modified to look as shown in Figure 3.7. This type of modification in distance field ω is proposed to allow high gradient in the thin region.

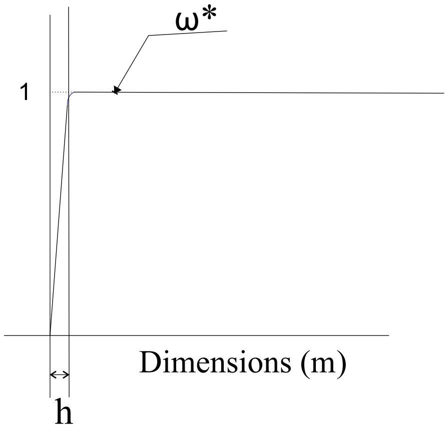


Fig. 3.7: modified distance field (ω^*)

And the expression for the modified ω is [11]:

$$\omega^*(x) = 1 - max(0, 1 - \frac{\omega(x)}{h})^\gamma; \gamma = 2, 3, 4... \tag{3.2}$$

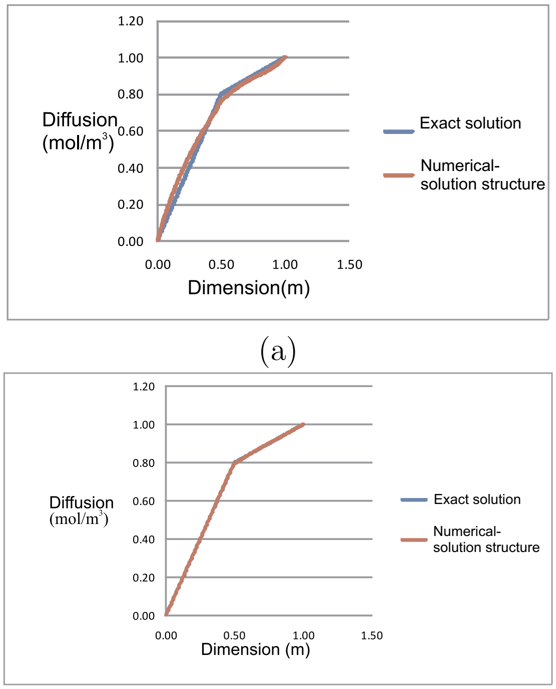
In the expression 3.2, when $\omega(x)$ is greater than or equal to h, then $\omega^*(x)$ takes the value of 1. Hence, when this new distance field ω^* is multiplied with B-splines, the resulting function acts normally at the horizontal line shown in Figure 3.7, but at high gradient zone it inherits the property of high gradient. Moreover, ω^* is going to be sharp at the interface boundary, so in order to make it smooth γ is introduced.

3.4 Results for modification of distance fields

This section is proposed to compare the results of numerical approach using modified distance fields with 1D integration algorithm to current implementation of meshfree algorithm. The goal of this numerical experiments is to demonstrate the feasibility of solution structure with modified distance fields algorithm over current implementation of meshfree algorithm 2.3. The parameters used in this section are same as of previous sections 2.3.

3.4.1 Domain with equal thickness problem

The objective of this experiment is to test the modified distance fields algorithm for smaller geometric ratios. By analysis, the plot in Figure 3.8 show the numerical approach almost satisfies the exact solution with small distortion for solution structures using modified distance fields algorithm. This small distortion from exact solution is due to two reasons: First factor is 1D integration approach, which is proposed for thin layers. This approach is based on the assumption that there is no change of physical fields (diffusion coefficient) along the horizontal direction of thin layer. The second factor is about modified distance fields, which is designed to accommodate high gradient in thin region.



(a)

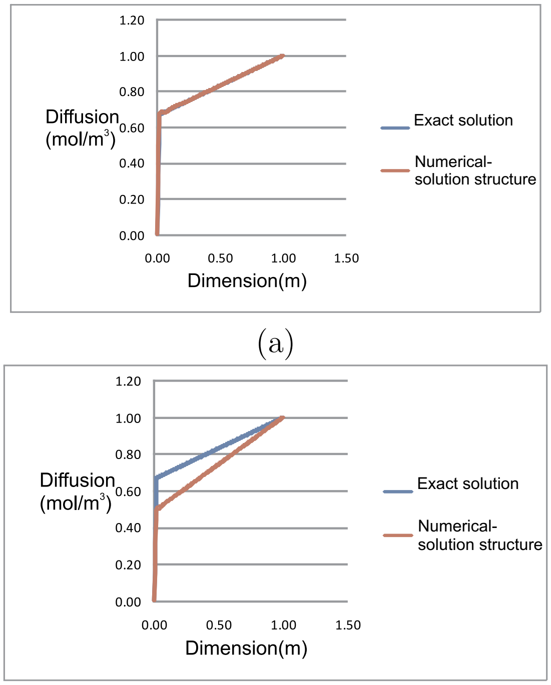
(b)

Fig. 3.8: Plots for parameters h = 0.5, k = 1; λ_i = 50, λ_j = 200 with (a)2 × 2 size grid using modified distance fields with 1D integration (b)5 × 5 size grid using current implementation of meshfree method

3.4.2 Domain with thin layer and high gradient problem

The key objective of implementing new modified distance fields algorithm in meshfree methods is to accommodate high gradient in the thin regions. This benchmark problem was designed in such a way that it has high gradient in thin region.

By analysis, the plot in Figure 3.9(a) show that the numerical solution coincide with the exact solution using modified distance fields with 1D integration algorithm. This is because the new solution structure successfully accommodates high gradient in the thin region. The plot in Figure 3.9(b) show current solution structure algorithm. The distortion in this plot is due to less number of B-splines. So from the results of modified distance fields, it clearly portrays that the resulting function of expression for (3.2) inherits to the property of high gradient. Moreover, the solution structure with modified distance fields (ω^*) using 1D integration works better for bigger geometric and physical parametric ratios with coarser grid.



(a)

(b)

Fig. 3.9: Plots for parameters h = 0.02, k = 49; $\lambda_i = 1$, $\lambda_j = 100$ with (a) 10×10 size grid using modified distance fields with 1D integration (b) 40×40 size grid using current implementation of meshfree method

3.5 Conclusions

In this chapter, we discussed implementation of meshfree method with distance fields using two computational tools. We started with construction of 1D integration algorithm and later studied the results of numerical experiments for the same set of parameters used in exact solution 2.2 and then stated limitations and conditions for 1D integration algorithm. Later in section 3.3, we proposed a new technique for accommodating high gradient in the thin region. For the proposed new technique, we compared the results with current solution structure algorithm and proved that the resulting function from this new technique inherits to the property of high gradient at high gradient zone with less computational time.

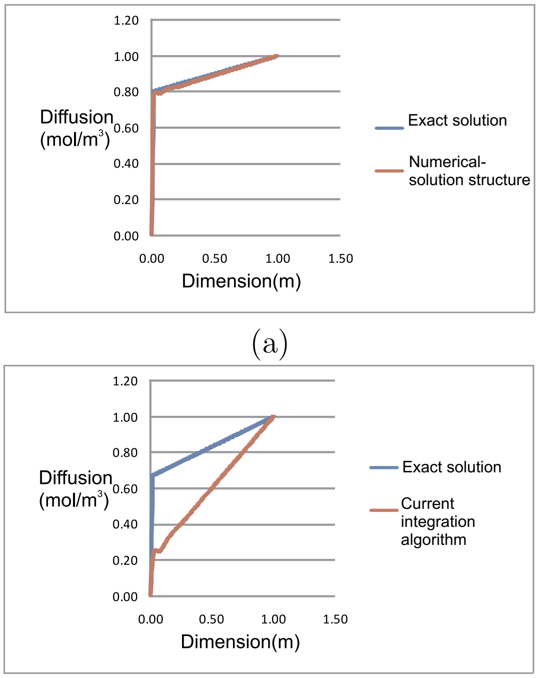
Chapter 4

NUMERICAL EXAMPLES

4.1 Numerical experiments for 1D example

This section of numerical experiments is introduced to determine the range of physical and geometric parameters that are applicable for improved computational tools. To evaluate the new improved computational tools, the results of current implementation of meshfree algorithm is compared with results of modified distance fields with 1D integration algorithm. For better understanding of the results, solution error has been calculated. Following subsections present the plots of numerical experiments for different geometric and physical parametric ratios.

4.1.1 Experiment - 1



(a)

(b)

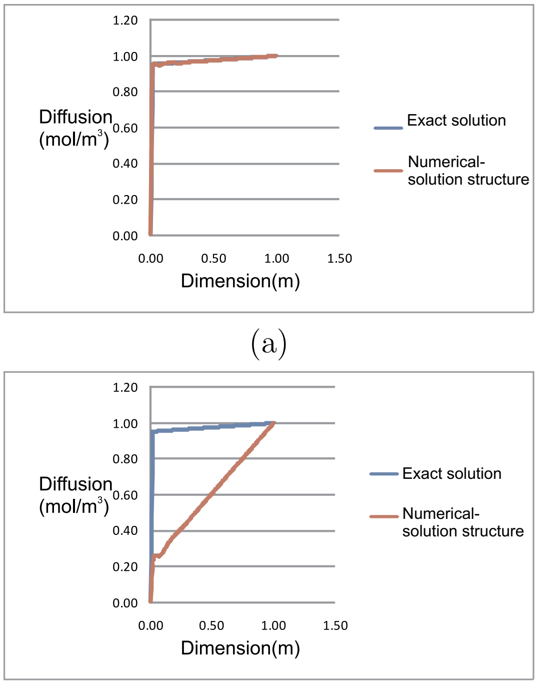
Fig. 4.1: Plots for parameters h = 0.02, k = 49; $\lambda_i = 1$, $\lambda_j = 200$ with 10×10 size grid using - (a) Modified distance fields with 1D integration (b) Current implementation of solution structure method

In this numerical experiment, concentration distribution is modeled in a two layer domain. The ratio between the thickness is same as of benchmark problem-2. But the physical parametric ratio is further increased. The objective of this experiment is to find whether the new improved computational tools for meshfree methods will work for the increased physical parameter ratio.

The plots in Figure 4.1 show concentration distribution for parameters with thickness ratio of fifty and diffusion coefficients ratio of two hundred between the two layers. Figure 3.2(b) show that the numerical solution for current implementation of meshfree algorithm fails to coincide with the exact solution. On the other hand, Figure 3.2(a) show plot for numerical solution using improved computational tools coincides with exact solution even after using coarser grid (10×10). The solution error for both current implementation and improved computational tools of meshfree slgorithm are $3.77E - 001$ and $2.35E - 002$, respectively. Analyzing of these results, the improved computational algorithm successfully accommodates high gradient in the thin region and models concentration distribution with less number of B- splines. Moreover the use of improved computational tools reduces 93 percent of error when compared with current algorithm.

4.1.2 Experiment - 2

This experiment resembles to a similar setup as of experiment- 1 in section 4.1. The only differnce is that the physical parametric ratio is further increased. The plots in Figure 4.2 show concentration distribution for parameters with thickness ratio of fifty and a diffusion coefficient ratio of thousand between the two layers. The solution error for both FEM and improved computational tools are $3.36E - 001$ and $2.76E - 002$, respectively. By analysis, it potrays that even though the ratio between the physical parameters is high, modified distance fields with 1D integration algorithm (Figure 3.2(b)) coincides with the exact solution using less number of B-splines. This results in the same conclusion as of Experiment - 1 in section 4.1. The improved computational tools reduce 91 percent of error when compared with current algorithm.



(a)

(b)

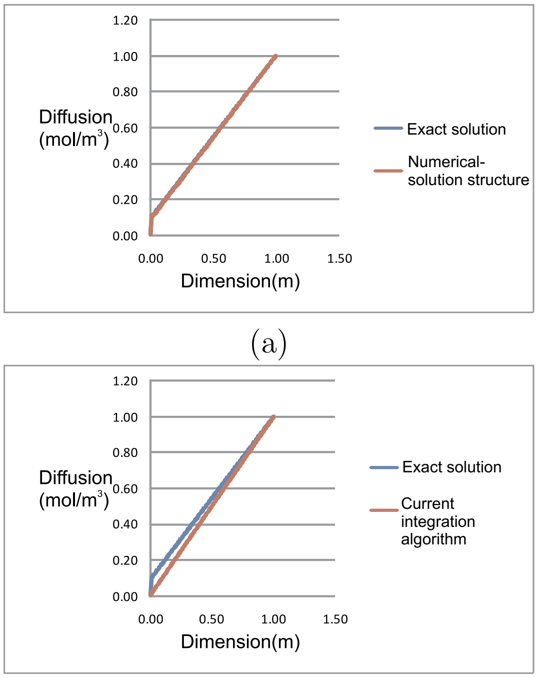
Fig. 4.2: Plots for parameters h = 0.02, k = 49; $\lambda_i = 1$, $\lambda_j = 1000$ with 10×10 size grid using - (a) Modified distance fields with 1D integration (b) Current implementation of solution structure method

4.1.3 Experiment - 3

The difference between this experiment and the other two is that there is increase in geometric ratio. The plots in Figure 4.3 show concentration distribution for parameters with thickness ratio of thousand and diffusion coefficients ratio of hundred between the two layers. The solution error for both FEM and improved computational tools are $8.46E-002$ and $2.71E-002$, respectively. This results in a similar conclusion as that of Experiment - 1 in section 4.1. The new improved computational tools reduces 68 percent of error compared with FEM solution for this experiment.

4.1.4 Experiment - 4

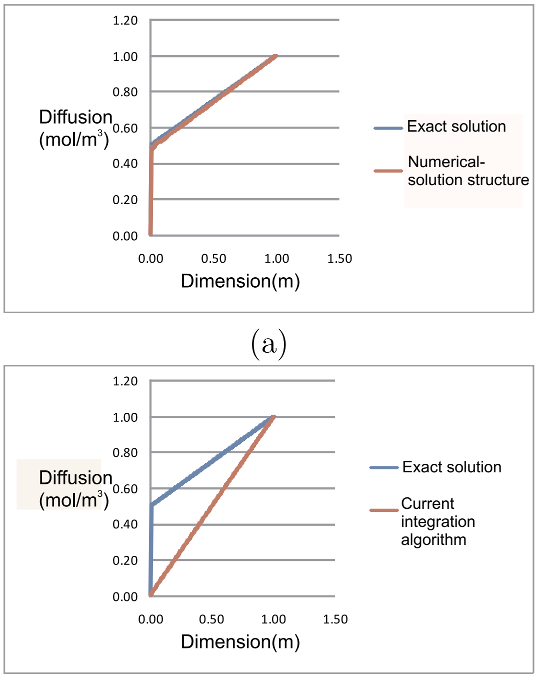
In this experiment, both ratios are increased to determine the applicability of meshfree methods with new computational tools. The plots in Figure 4.4 show concentration distribution for parameters with thickness ratio of thousand between layers and diffusion coefficient ratio of hundred between the two layers. By analysis it shows that



(a)

(b)

Fig. 4.3: Plots for parameters h = 0.001, k = 999; $\lambda_i = 1, \quad \lambda_j = 100$ with 15 × 15 size grid using - (a) Modified distance fields with 1D integration (b) Current implementation of solution structure method



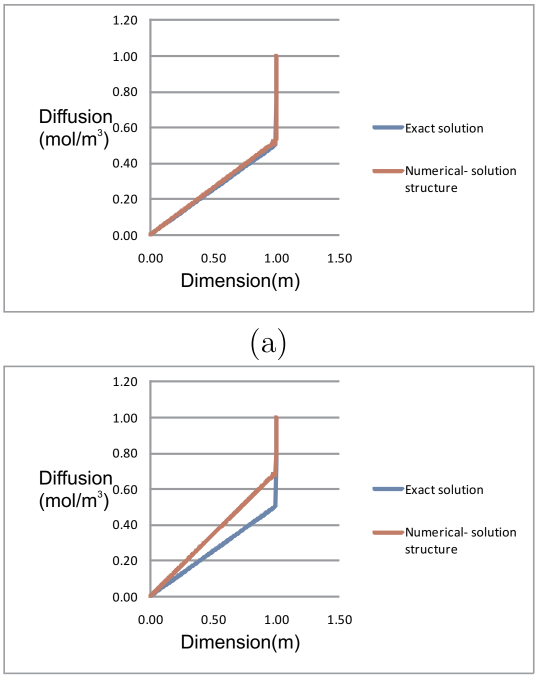
(a)

(b)

Fig. 4.4: Plots for parameters h = 0.002, k = 499; $\lambda_i = 1, \quad \lambda_j = 500$ with 20 × 20 size grid using - (a) Modified distance fields with 1D integration (b) Current implementation of solution structure method

solution structure for modified distance with 1D integration works better with both increase in physical parameters ratio and geometric parameters ratio. This results in the same conclusion as of Experiment - 1 in subsection 4.1.2. The solution error for both FEM and improved computational tools are $3.73E-001$ and $3.95E-002$, respectively. The new improved computational tools reduces 89 percent of error compared with FEM solution for this experiment.

4.1.5 Experiment - 5



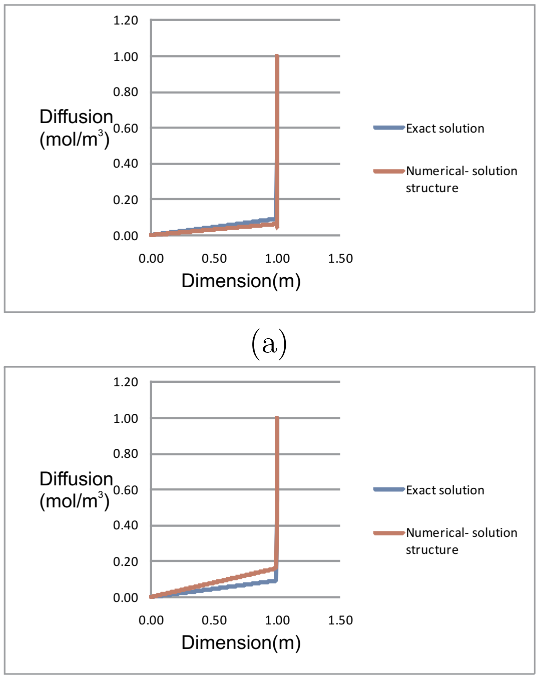
(a)

(b)

Fig. 4.5: Plots for parameters h = 0.99, k = 0.01; $\lambda_i = 100$, $\lambda_j = 1$ with 51 × 51 size grid using - (a) solution structure with 1D integration (b) Current implementation of solution structure method

In this experiment, the second layer was reduced in thickness instead of the first layer. The setup of this experiment is similar to all the previous experiments except that the implementation of 1D integration algorithm is applied to the second layer instead of first layer. The objective of this experiment is to test 1D integration algorithm for different geometric profiles. The results were graphed for both 1D integration algorithm and current implementation algorithm as shown in Figure 4.5. The plots show that the solution structure with improved computational tools coincides with exact solution compared to current implementation of solution structure. This experiment concludes that 1D integration algorithm yields better solution for different geometry profiles provided the conditions or assumptions for 1D integration algorithm are satisfied.

4.1.6 Experiment - 6



(a)

(b)

Fig. 4.6: Plots for parameters h = 0.99, k = 0.01; $\lambda_i = 1000$, $\lambda_j = 1$ with 104×104 size grid using - (a) solution structure with 1D integration (b) Current implementation of solution structure method

The difference between this experiment and the previous experiment was the high gradient in the second layer. The results of this experiment conclude that the 1D integration yields better solution by keeping one B- spline in the thin layer.

4.2 Analysis of glucose diffusion

In this example, a typical prototype model for only one pair of electrodes is taken into consideration. This model has dimensions with height and diameter of electrodes are $120\mu m$ and diameter $20\mu m$ respectively. The well width between electrodes is $40\mu m$ and the enzyme layer thickness is $10\mu m$. Model of the domain is as shown in Figure 4.7.

Glossy carbon anodes and cathodes are standing on SiO2 layer. They have different enzyme layers immobilized on as shown in Figure 4.7. Environment surrounding the electrodes is glucose substrate.

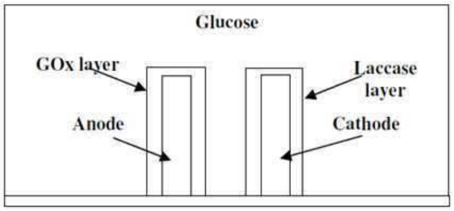


Fig. 4.7: Geometric domain

Using this 2D example, diffusion profile is studied using meshfree methods with distance fields. The boundary conditions prescribed for this 2D model are as follows: Anode - Enzyme layer interface and cathode layer interface is zero concentration, enzyme layer - bulk interface are continuity, outer bulk is maximum concentration and bottom layer is insulated.

Mathematical formulation of the problem is:

Partial differential equation-

$\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} = 0$

Boundary conditions-

$C|_{\partial\Omega_1} = 1;$

$C|_{\partial\Omega_2} = 0;$

$C|_{\partial\Omega_{3+}} = C|_{\partial\Omega_{3-}}$ (interface boundary)

The first three boundary conditions prescribes concentration on the respective domains. But the last two boundary conditions describe the interface boundary conditions, where the concentration is equal.

The domain is considered as a combination of two different materials. So the complete domain is sub divided into two different materials: Ω_1 and Ω_2. Ω_1 is the total bulk excluding the electrode and enzyme layer. Ω_2 consists of both the electrodes enzyme layer. Since the two domains are made up of different materials, each domain will have different diffusion coefficient. The physical properties of the materials are:

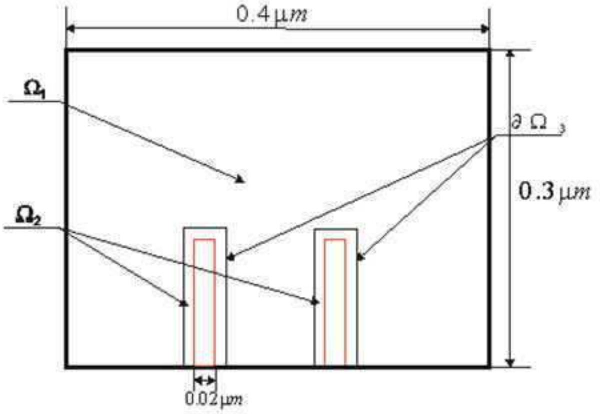


Fig. 4.8: Actual geometric domain generated by using meshfree methods

$\lambda_1 = 4.0e^{-7}$ (Diffusion coefficient for domain$_1$);

$\lambda_2 = 7.6e^{-7}$ (Diffusion coefficient for domain$_2$).

So the solution structure satisfying the above mentioned boundary conditions for mesh free methods with distance fields is shown in section in 2.1.1.

By solving the deferential equation using generalized Galerikin method results in values of matrix assembly (a_{ij}) and vector assembly (b_i) are calculated for finding unknown coefficient K_j.

$a_{ij} = \lambda_1[\int\int_{\Omega_1}\{(\frac{\partial\xi_i}{\partial x})(\frac{\partial\xi_j}{\partial x}) + (\frac{\partial\xi_i}{\partial y})(\frac{\partial\xi_j}{\partial y})\}\partial\Omega] + \lambda_2[\int\int_{\Omega_2}(\frac{\partial\xi_i}{\partial x})(\frac{\partial\xi_j}{\partial x}) + (\frac{\partial\xi_i}{\partial y})(\frac{\partial\xi_j}{\partial y})\}\partial\Omega]$

$b_i = \lambda_1[\int\int_{\Omega_1}(\nabla\varphi \cdot \nabla\xi_{1i})\partial\Omega] + \lambda_2[\int\int_{\Omega_2}(\nabla\varphi \cdot \nabla\xi_{2i})\partial\Omega]$

Where ξ represents the basis fuctions and φ prescribes the boundary conditions, by substituting this matrix and vector assembly in the equation 4.1, we obtain the system of linear algebraic equations that must be solved for the numerical values of the coefficients K_i.

$$AK = B \tag{4.1}$$

Solving the linear system and substituting the computed values of K_i into the assumed expression 2.6 of C produces an approximate solution $C(x)$ to the differential equation satisfying the specified boundary conditions.

The distance function ω completely describes all the geometric information for this two dimensional Dirchilet boundary value problem is shown in Figure 4.9.

Figure 4.10 shows the comparison of solution of both meshfree methods with distance

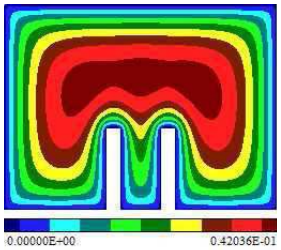
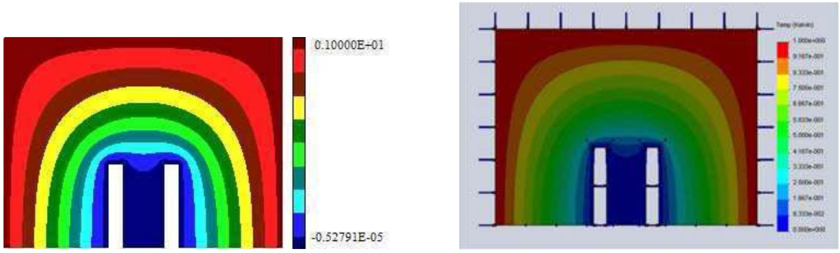
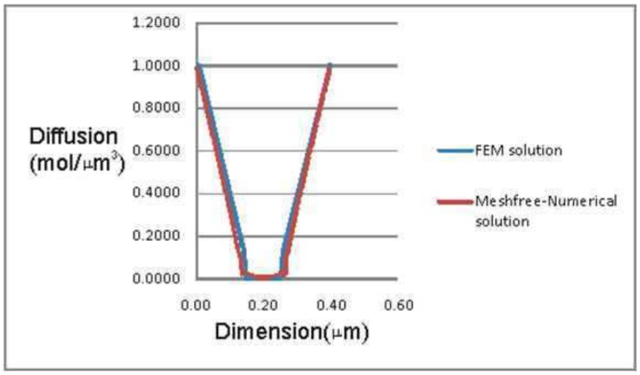


Fig. 4.9: Plot for distance function



(a) (b)

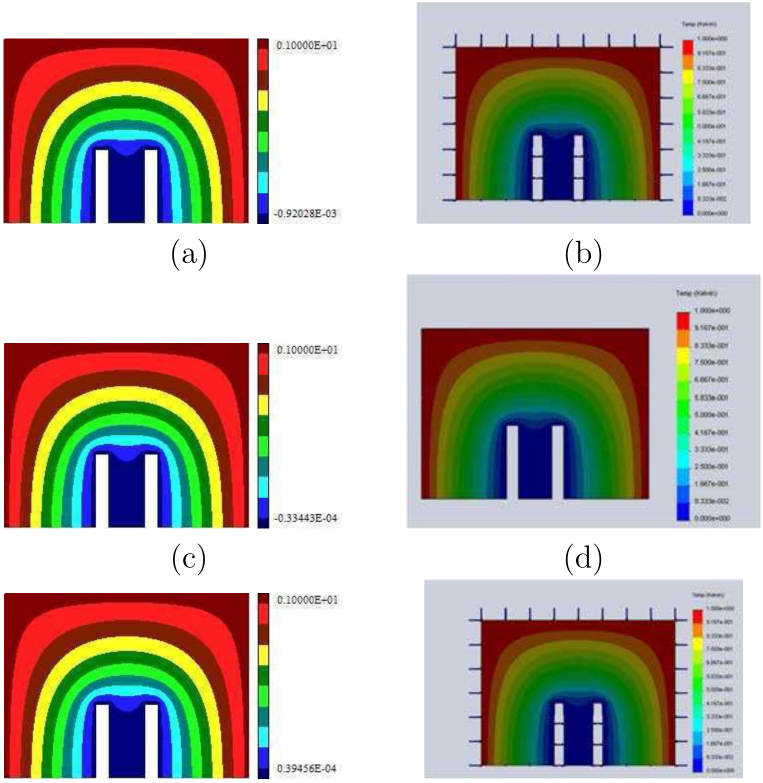
Fig. 4.10: Comparison of diffusion concentration solution and degrees of freedom for domain with enzyme layer thickness of 0.01 (a) meshfree methods with distance fields: Number of B-splines 2500 (b) FEM: Number of finite element 18684



(a)

Fig. 4.11: Comparison of diffusion concentration values along the bottom edge of the domain with enzyme layer thickness of 0.01: meshfree methods with distance fields-Number of B-splines 2500; FEM- number of finite elements 18684

fields and FEM. For FEM solution, Cosmosworks software was used. The same set of geometric and physical parameters were used for both methods. The degrees of freedom used in both approaches were optimum inorder to yield better diffusion concentration solution. For better comparative analysis the values on the bottom edge of the domain are plotted for both meshfree and FEM method. The plot in Figure 4.11 show that the solution of FEM coincides with mesffree methods solution. So these Figures 4.11 and 4.10 results in a conclusion that the number of supports required by meshfree methods to yield better solution are one third of the number of supports required by FEM.



(a) (b)

(c) (d)

(e) (f)

Fig. 4.12: Comparison of degrees of freedom for different thickness of the thin layer (a) meshfree methods with distance fields: Thickness- 0.01/10; Number of B-splines- 2500 (b) FEM: Thickness- 0.01/10; Number of finite elements- 30190 (c) meshfree methods with distance fields: Thickness- 0.01/50; Number of B-splines- 2500 (d) FEM: Thickness- 0.01/50; Number of finite elements- 56182 (e) meshfree methods with distance fields: Thickness- 0.01/100; Number of B-splines- 2500 (f) FEM: Thickness- 0.01/100; Number of finite elements- 56182

To extend the applicability of new developed computational tools further experiments are performed by varying the thickness of thin layer. From the experimental results in Figure 4.12, it is significant to note that the degrees of freedom used for FEM were more, when compared with meshfree methods with distance fields. This concludes that new computational tools for meshfree methods with distance fields yield accurate solution with less number of supports.

4.3 Summary

The results of the numerical experiments proves that new computational tools for meshfree methods reduces computational time compared to other commercial methods. Moreover, the new computational tools yield better solutions even for large geometric and parametric ratios.

Chapter 5

CONCLUSIONS

This research was started with a frame of idea of demonstrating the feasibility of meshfree analysis in heterogeneous materials with high contrast of geometric and physical parameters. For this purpose, the current implementation of meshfree method was used to solve two different benchmark problems. These experiments resulted in a conclusion that meshfree analysis requires allocation at least one B-spline in the thin region. If there is a large contrast among geometric dimensions of the layers, current implementation of meshfree method will result in a substantially large degrees of freedom. In this work we developed new computational tools for meshfree method with distance fields and our work concluded in very positive results such as:

- Developed 1D integration algorithm to analyze diffusion profile in thin regions. This approach requires just one B-spline in the thin layer to analyze the problem. The increase in speed of computations can be measured by calculating the ratio of number of B-splines used in current implementation of meshfree methods to the number of B-splines in 1D integration algorithm.

- Developed modified distance fields algorithm to accommodate high gradient in thin region. The combination of both modified distance fields algorithm and 1D integration algorithm resulted in more accurate numerical solution. The results of experiments in section 3.4 conclude that the number of B-splines required in the thin layer are less than one.

- The new computational computational tools for meshfree methods were tested for geometric ratio up to 1 : 1000 and for physical parameter ratio up to 1 : 1000. Solution error for both current implementation and improved computational tools for meshfree method are calculated. This new computational reduces up to 80 percent of relative error when compared with traditional approach.

- To further test the applicability of new improved computational tools for different domain, a 2D model is considered to analyze diffusion concentration. This experiment resulted in uniform distribution of gradient over the domain with less number of B-splines.

From this work, it is concluded that the improved computational tools reduce computational time and improve accuracy of the solution. Moreover from the 2D example, it portrays that the improved computational tools are applicable even for complex geometries with bigger geometric and physical parameters ratio between the materials. The same approach can be used to study other physical fields such as temperature, electric and magnetic fields, since all these physical fields obey the same equation.

BIBLIOGRAPHY

[1] D. O. Kimble A. T. Yahiro, S. M. Lee. Bio electrochemistry. i. enzyme utilizing bio-fuel cell study. *Biochimica et biophysica acta*, 1964.

[2] V.Rokityanska A.Shevchenko, I.Tsukanov. Simulation of temperature fields in forming products from composite materials. *EUROSIM*, 1995.

[3] P Villon B Nayroles, G Touzot. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 10:301–318, 1992.

[4] I.Tsukanov Brain Luft, Vadim Sharpiro. Geomentrically adaptive numerical integration. 2008.

[5] J. Tinsley Oden C. Armando Duarte. H-p clouds- an h-p meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1998.

[6] A. H. D. Cheng Carlos A. Brebbia. *Engineering analysis with boundary elements.* Elsevier, 2009.

[7] Xin-An Feng Feng Wang, Ke-Zhang Chen. Modeling of plasma spraying process to manufacture hybrid materials. *Computer aided design*, 2007.

[8] Wing Kam Liu Frank C. Gunter. Implementation of boundary conditions for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 163:205–230, 1998.

[9] M.B. Liu G.R. Liu. *Smoother particle hydrodynamics.* World Scientific, 2003.

[10] C. Wang-L. Kulinsky M. Song M. Madou H. Xu, K. Malladi. Carbon post micro arrays for glucose sensors. *Biosensors and Bioelectronics*, 23, 2008.

[11] K. Höllig. *Finite Element Methods with B-Splines.* Number 26 in Frontiers in Applied Mathematics. SIAM, 2003.

[12] V.Sharpiro I.Tsukanov. The architecture of sage — a meshfree system based on rfm. *Engineering with computers*, 18:295–311, 2002.

[13] V.Sharpiro I.Tsukanov. Meshfree modeling and analysis of physical fields in hereogeneous media. *Advances in computational mathematics*, 23:95–124, 2005.

[14] C. M. O. L. Roque Hui-Ping Wang Jiun-Shyan Chen, C. Pan. A lagrangian reproducing kernel particle method for metal forming analysis. *Computational Mechanics*, 22:289–307, 1998.

[15] S. Kitipornchai K. M. Liew, Yumin Cheng. Boundary element-free method (befm) for two-dimensional elastodynamic analysis using laplace transform. *International Journal for Numerical Methods in Engineering*, 64:1610–1627, 2005.

[16] Hyun-Gyu Kim. Arbitrary placement of local meshes in a global mesh by the interface-element method (iem). *International Journal for Numerical Methods in Engineering*, 56:2279–2312, 2003.

[17] K. Servat K. B. Kokoh C. Jolivalt-C. Innocent M. Cretin M.Rolland S. Tingry L. Brunel, J. Denele. Oxygen transport through laccase biocathodes for a membrane — less glucose/o2 biofuel cell. *Electrochemical Communications*, 9, 2007.

[18] Rodrigo Rossi Marcelo Krajnc Alves. A modified element-free galerkin method with essential boundary conditions enforced by an extended partition of unity finite element weight function. *International Journal for Numerical Methods in Engineering*, 57:1523–1552, 2003.

[19] J. M. Melenk and I. Babuska. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[20] L. D. Libersky P. W. Randles. Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:375–408, 1996.

[21] Yamini Parikh. Simulation of c-mems based biofuel cell.

[22] Q.W.Ma. A new meshless interpolation scheme for mlpgr method. *Computer modelling in engineering and sciences*, 2008.

[23] T. Zhu S.N. Atluri. A new meshless local petrov-galerkin (mlpg) approach in computational mechanics. *Computational Mechanics*, 22(2):117–127, 1998.

[24] 2010 Sulzer Metco Ltd. *Automotive.* www.sulzermetco.com, Accessed 10 February 2010.

[25] V. L. Rvachev T. I. Sheiko. R-functions in boundary value problems in mechanics. *Applied Mechanics*, 48:151–188, 1996.

[26] V. Shapiro I. Tsukanov V. L. Rvachev, T. I. Sheiko. On completeness of rfm solution structures. *Computational Mechanics*, 25:305–317, 2000.

[27] V. Shapiro I. Tsukanov V. L. Rvachev, T. I. Sheiko. *On Completeness of RFM solution structures, special issue on meshfree methods*, volume 25. 2005. In Russian.

[28] W. T. Flannery W. H. Teukolsky, S. A. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1982.

[29] Yi Fei Zhang Wing Kam Liu, Sukky Jun. Reproducing kernel particle methods. *International journal for numerical methods in fluids*, 20:439, 1995.

[30] M.D. Porter J. Zak Y. Dai, G.M. Swain. Opticaly trasparent electrodes for use in ir transmission spectroelectrochemical measurements. *Analytical chemistry*, 2008.

[31] G.P.Manko Yu.G.Stoian, V.S. Protsenko. Meshfree modeling analysis of physical fieldsin heterogeneous media, advances in computational mathematics. 23:95–124, 2005.

APPENDIX

PROGRAMME FOR MESHFREE METHODS WITH IMPROVED COMPUTATIONAL TOOLS

```
#include "rfms.h"

#include <stdio.h>

#include <conio.h>

#include <math.h>

int pre1(void);

int pre2(void);

tuple omega(void);

tuple omega1(void);

double fa(int flag, int i, int j);

double fa1(int flag, int i, int j);

double fa2(int flag, int i, int j);

double fb(int flag, int i);

double fb1(int flag, int i);

double fb2(int flag, int i);

double fu(void);

double fu1(void);

double fu_exact(void);

tuple omega_dstar(void);

double om_dstar(void);

double error(void);

double error1(void);

double om(void);
```

```
void u0(void);
    void u0_1(void);
    void u0_2(void);
    tuple u1(void);
    tuple u1_1(void);
    tuple u1_2(void);
    tuple *w1, *w2;
    tuple_seq *p;
    tuple_seq *p1;
    double ff1(void);
    double ff2(void);
    double h, c, k, fi1, fi2, lam1, lam2, m12,c1, c2, lammax;
    double lambda;
    void main(void)
    {       double xmin, ymin, xmax, ymax, x1, y1, x2, y2;
            double xmin_ext, ymin_ext, xmax_ext, ymax_ext;
            xmin = 0;
            ymin = 0;
            xmax = 1;
            ymax = 1;
            k = 99;
            fi1 = 0.0;
            fi2 = 1.0;
            lam1 = 1;
            lam2 = 100.0;
            h = xmin + (xmax-xmin)/(1.0+k);
```

```
 x1 = (xmin + h * 0.5);
        y1 = (ymin);
  x2 = (xmin + h * 0.5);
  y2 = (ymax);
        double L1, L2;
        L1 = 2.0/lam1; L2 = 2.0/lam2;
        m12 = L1 <= L2 ? L1 : L2;
        timerclass T;
        T.ShowTimeBegin();
// Allocation of the data structure for automatic differentiation
        SetTupleDimensionOrder(2,2); //2D, max order = 2;
// Defining the bounding box
        SetBoundingBox2D( xmin, ymin, xmax, ymax );
//      SetRfunctionsType( RFUNCTIONS_ALPHA_VAR, 12);
        w1 = new tuple;
        w2 = new tuple;
//      Defining information about B-splines: degree, grid
        int ns, nx,ny, i;
        ns = 3;       // degree of B-splines
        nx = 20;
        ny = 20; // grid nx x ny
//      Allocation of the matrix and vector
        matrixclass a;
        vectorclass b;
        a.CreateMatrix(M_BANDED_SYMMETRIC, ns, nx, ny);
       b.CreateVector(ns, nx, ny);
```

```
//  Defining information about basis functions
            tuple_seq P(1);  p  = &P; //An empty entry; (1) means the number of sequences
            tuple_seq P1(1); p1 = &P1; //An empty entry; (1) means the number of sequences
//  The first sequence is B-splines of degree ns, with grid nx*ny
            P.SetBsplines2DDegree (0, ns, nx, ny); // 0's sequence
            P1.SetBsplines2DDegree(0, ns, nx, ny); // 0's sequence
//  Define which matrix and vector will be assembled by the integration procedures
            SetMatrixPointer( &a );
            SetVectorPointer( &b );
//  Define data structure for integration: the first 2 parameters are the number of
//          B-splines; the last parameters are coordinates of the bounding box
            AdjustBoundingBox2D(xmin, ymin, xmax, ymax,   xmin_ext, ymin_ext, xmax_ext, 
ymax_ext);
            quadtreeclass tree1(nx, ny, xmin_ext, ymin_ext, xmax_ext, ymax_ext);
            quadtreeclass tree2(nx, ny, xmin_ext, ymin_ext, xmax_ext, ymax_ext);
            int n_gauss = 5;
            int n_random= 5;
            tree1.BuildTree(n_gauss, 0.5, n_random, pre1);
            tree2.BuildTree(n_gauss, 0.5, n_random, pre2);
            ContourIntegrateLineMV(nx,  ny, n_gauss, x1, y1, x2,y2, fa1, fb1 );
            lambda = lam1;
            tree1.IntegrateMV(fa1,fb1);
            lambda = lam2;
            tree2.IntegrateMV(fa2,fb2);
            a.Solve( b ); // on return b contains values of C_i
//  save results
            filexy("ritz",299,299, xmin,ymin,xmax,ymax, fu1, om);
```

```
        FILE *fp1, *fp2, *fp3;

        fp1 = fopen("numerical_solution.txt","w");

        fp2 = fopen("exact_solution.txt","w");

        fp3 = fopen("om_dstar.txt","w");

        int j,n;

        double dx, x, f,c1,c2,t1,t2,eps,Dx;

        n = 201;

        dx = (xmax - xmin)/(n-1);

        SetArgumentX(2,0.0); // y = 0;

        for( j = 0; j < n; j++ )

        {       x = xmin + j * dx;

                SetArgumentXR(1, x);

                f = fu1();

           fprintf(fp1,"%26.15e %26.15e\n", x, f );

                fprintf(fp2,"%26.15e %26.15e\n", x, fu_exact() );

                fprintf(fp3,"%26.15e %26.15e\n", x, om_dstar() );       }

        fclose(fp1);

        fclose(fp2);

        fclose(fp3);

// Calculation of Epsilon

        tree1.Integrate (error) ;  // int((fu_exact - fu1)^2)

        tree2.Integrate (error);  // int((fu_exact - fu1)^2)

  t1 = GetIntegralValue();

        tree1.Integrate (error1);   // int((fu_exact)^2)

        tree2.Integrate (error1);   // int((fu_exact)^2)

  t2 = GetIntegralValue();
```

```
        eps = sqrt(t1/t2);

        printf("eps=%f",eps);

// Calculation of Dx

        Dx = (xmax_ext-xmin_ext)/(nx-1);

        printf("\nDx = %26.15e   error = %26.15e",Dx, eps);

        printf("\nPress any key to exit . . .\n");

        getch();

        T.ShowTime();

        return;         }

//Point membership classification function

int pre1(void)

{       if( L_line( 0, 1, 0, -1 ) && L_line( h, -1, h, 1 ))

        {       return 1;

        }

        else

        {       return -1;

        }

}

int pre2(void)

{       if( L_line( h, 1, h, -1 ) && L_line( h*(k+1), -1, h*(k+1), 1 ) )

        {       return 1;

        }

        else

        {       return -1;

        }

}
```

```
double om(void)
{       return omega().GetValue();
};
// The omega function
tuple omega(void)
{       *w1 = line( 0,1,0,-1 );
        *w2 = line(h*(1+k),-1,h*(1+k),1);
        return ( *w1 ) & (*w2);         }
tuple omega_dstar(void)
{       static tuple w(GetTupleMaxOrder()), w_cut(GetTupleMaxOrder());

        *w1 = line( 0,1,0,-1 );
        *w2 = line(h*(1+k),-1,h*(1+k),1);
//      return ((*w1) & (*w2));
  // c = (0.15*k)+4.85;
        c = (0.1*k)- 2.35;
        w = *w1;
//      w = 1.0 - w/(c9*h);
//      w = 1.0 - w/(h * (6+ ((k+1)/100)*2));
        if ( k > 49)
        {       w = 1.0 - w/(h * (6+ ((k+1)/100)*2));
        }
        else
                w = 1.0 - w/(h * 6);
        if( w.GetValue() > 0.0 )
        {       w = (1.0 - ((square(w))));
```

```
            }
            else
                        w = (1.0 ) ;
                        if ( k > 49)
            {           w_cut = h * (6+ ((k+1)/100)*2);
            }
            else
                        w_cut = h * 6;
             return ((w) & (*w2)) & w_cut;                }
double om_dstar(void)
{           return omega_dstar().GetValue();
}
//Solution structure (external bc)
void u0(void)
{           p->Compute(0);
            p->Mult(omega_dstar());
            return;              }
tuple u1(void) // interpolation of BC
{           tuple f1, f2;
            f1 = fi1; f2 = fi2;
            omega();
            return paste((*w1),(*w2), f1,f2);           }
double ff1(void)
{           tuple om, w;
            double delta;
            om = omega();
```

```
        delta = m12 * lam2 - 1.0;
        w  = line( h, -1, h, 1 ); // interface boundary
        return (square(om) & w).GetValue() * delta;      }
double ff2(void)
{       tuple om, w;
        double delta;
        om = omega();
        delta = m12 * lam1 - 1.0;
        w  = line( h, 1, h, -1 );
        return (square(om) & w).GetValue() * delta;      }
void u0_1(void)
{       tuple om, w;
        double delta;
        om = omega_dstar();
        p->Compute(0);
        p->Mult(om);
        delta = m12 * lam2 - 1.0;
        w  = line( h, -1, h, 1 ); // interface boundary
        *p1 = *p + (square(om) & w) * delta * d1(w,*p);
// d1 -- differentiation operator in "internal" normal direction
        return;          }
void u0_2(void)
{       tuple om, w;
        double delta;
        om = omega_dstar();
        p->Compute(0);
```

```
          p->Mult(om);
          delta = m12 * lam1 - 1.0;
          w  = line( h, 1, h, -1 );
          *p1 = *p + (square(om) & w) * delta * d1(w,*p);
          return;          }
tuple u1_1(void)
{         tuple f1, f2;
          tuple om, w;
          double delta;
          f1 = fi1; f2 = fi2;
          om = omega_dstar();
          delta = m12 * lam2 - 1.0;
          f1 = paste((*w1),(*w2), f1,f2);
          w  = line( h, -1, h, 1 );
          return f1 + (square(om) & w) * delta * d1(w,f1);          }
tuple u1_2(void)
{         tuple f1, f2;
          tuple om, w;
          double delta;
          f1 = fi1; f2 = fi2;
          om = omega_dstar();
          f1 = paste((*w1),(*w2), f1,f2);
          w  = line( h, 1, h, -1 );
          delta = m12 * lam1 - 1.0;
          return f1 + (square(om) & w) * delta * d1(w,f1);          }
//Ritz functional
```

```
double fa(int flag, int i, int j)
{         double result1,result2;
//  Compute basis function at the point
          if( flag == 0 ) // new integration point
          {         u0();
          }
//  Computation of the dot product of the gradients
          result1 =   dx_direct( p->GetTuple(0,i), 1)*
                                        dx_direct( p->GetTuple(0,j), 1);
          result2 =   dx_direct( p->GetTuple(0,i), 2)*
                                        dx_direct( p->GetTuple(0,j), 2);
          return (result1 + result2) * lambda;        }
double fa1(int flag, int i, int j)
{         double result1,result2;
//  Compute basis function at the point
          if( flag == 0 ) // new integration point
          {         u0_1();
          }
//  Computation of the dot product of the gradients
          result1 =   dx_direct( p1->GetTuple(0,i), 1)*
                                        dx_direct( p1->GetTuple(0,j), 1);
          result2 =   dx_direct( p1->GetTuple(0,i), 2)*
                                        dx_direct( p1->GetTuple(0,j), 2);
          return (result1 + result2) * lam1 * h;      }
double fa2(int flag, int i, int j)
{         double result1,result2;
```

```
//  Compute basis function at the point
        if( flag == 0 ) // new integration point
        {       u0_2();
        }
//  Computation of the dot product of the gradients
        result1 =   dx_direct( p1->GetTuple(0,i), 1)*
                                dx_direct( p1->GetTuple(0,j), 1);
        result2 =   dx_direct( p1->GetTuple(0,i), 2)*
                                dx_direct( p1->GetTuple(0,j), 2);
        return (result1 + result2) * lam2;          }
double fb(int flag, int i)
{       double result1,result2,omm;
        static tuple uu1(GetTupleMaxOrder());
if( flag == 0 )
        {       u0();
                uu1 = u1();     }
        result1 =  dx_direct( p->GetTuple(0,i), 1) *
                    dx_direct( &uu1, 1 );
         result2 =  dx_direct( p->GetTuple(0,i), 2 )*
                          dx_direct( &uu1, 2 );
        return -(result1 + result2)*lambda;         }
double fb1(int flag, int i)
{       double result1,result2,omm;
        static tuple uu1(GetTupleMaxOrder());
        if( flag == 0 )
        {       u0_1();
```

```
                    uu1 = u1_1();    }
          result1 =  dx_direct( p1->GetTuple(0,i), 1) *
                              dx_direct( &uu1, 1 );
          result2 =  dx_direct( p1->GetTuple(0,i), 2 )*
                               dx_direct( &uu1, 2 );
          return -(result1 + result2)*lam1*h;         }
double fb2(int flag, int i)
{         double result1,result2,omm;
          static tuple uu1(GetTupleMaxOrder());
          if( flag == 0 )
          {         u0_2();
                    uu1 = u1_2();    }
          result1 =  dx_direct( p1->GetTuple(0,i), 1) *
                         dx_direct( &uu1, 1 );
          result2 =  dx_direct( p1->GetTuple(0,i), 2 )*
                               dx_direct( &uu1, 2 );
          return -(result1 + result2)*lam2;           }
// SOLUTION
double fu(void)
{         static vectorclass *c;
//  Get address of the coefficients
          c = GetVectorPointer();
//  Compute basis functions at the point
          u0();
//  Compute the sum
          return (sum( c, u1(), *p ).GetValue());    }
```

```
double fu1(void)
{           static vectorclass *c;
//  Get address of the coefficients
            c = GetVectorPointer();
//  Compute basis functions at the point
            if( pre1() > 0 )
            {           u0_1();
//  Compute the sum
                        return (sum( c, u1_1(), *p1 ).GetValue());
            }
            else
            {           u0_2();
//  Compute the sum
                        return (sum( c, u1_2(), *p1 ).GetValue());
            }
}
double fu_exact(void)
{           double x = GetArgumentX(1);
            if( x >= 0.0 && x < h )
            {
// first region
                        c1 = fi1;
                        c2 = (-lam2 *(fi1 - fi2) )/(h *(k *lam1 + lam2));
                        return c1 + c2*x;
            }
            else
```

```
        {
    // 2nd region
            c1 = (lam1* fi1 + k* lam1* fi1 - lam1* fi2 + lam2 *fi2)/(k *lam1 + lam2) ;
            c2 = ((-lam1* fi1 + lam1* fi2) )/(h *(k *lam1 + lam2));
            return c1 + c2*x;        }
    }
    double error(void)
    {       return (fu_exact()- fu1())*(fu_exact()- fu1());
    }
    double error1(void)
    {       return (fu_exact())*(fu_exact());
}
```