UNIVERSITY OF CALIFORNIA, SAN DIEGO

Meshfree Image-Based Reduced Order Modeling of Multiple Muscle Components with Connective Tissue and Fat

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy

in

Structural Engineering

by

Ramya Rao Basava

Committee in charge:

Professor Jiun-Shyan Chen, Chair
Professor Yuri Bazilevs
Professor Li-Tien Cheng
Professor Petr Krysl
Professor Shantanu Sinha

2015

This Dissertation of Ramya Rao Basava is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2015

DEDICATION

To my late Grandfather, Dr. Kandula Veera Raghava Rao,

my Mom, Dad & Brother

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank and express my gratitude to my advisor Professor J. S. Chen for giving me this invaluable opportunity to work in his research group. His in-depth knowledge, approach towards research, originality of ideas and way of thinking helped me gain immense knowledge and right aptitude for research.

I would like to thank Professor Shantanu Sinha, Professor Yuri Bazilevs, Professor Petr Krysl, and Professor Li-Tien Cheng for serving on my committee.

Chapter 4 in part has been published in the following book chapter and journal paper respectively: **Book Chapter**: R. R. Basava, J.-S. Chen, Y. Zhang, S. Sinha, U. Sinha, J. Hodgson, R. Csapo, and V. Malis, "Pixel based meshfree modeling of skeletal muscles", in Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications (Y. Zhang and J. Tavares, eds.), vol. 8641 of Lecture Notes in Computer Science, pp. 316-327, Springer International Publishing, 2014. The dissertation author was the primary investigator and author of this book chapter. The contribution of all the co-authors is greatly acknowledged, particularly the second co-author for guiding the research. **Journal paper**: J.-S. Chen, R. R. Basava, Y. Zhang, R. Csapo, V. Malis, U. Sinha, J. Hodgson, and S. Sinha, "Pixel-based meshfree modelling of skeletal muscles", Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, no. (ahead-of-print), pp. 1-13, 2015. The contribution of all the co-authors is greatly acknowledged, particularly the first co-author for guiding the fundamental development of the research.

I would like to thank Professor Shantanu Sinha, Professor Usha Sinha, Dr. John Hodgson, Dr. Robert Csapo and Vadim Malis for their valuable discussions on muscle physiology and for providing me with the medical images necessary for my research

work. The support for this research by National Institute of Health is greatly acknowledged.

I would also like to thank Professor Stanley B. Dong, Professor Ertugrul Taciroglu and Professor Joseph M. Teran for serving on my committee during the part of my study at University of California Los Angeles.

I would like to thank all my current and previous colleagues in my research group for their productive discussions.

Lastly I would like to thank my father Dr. B. V. Appa Rao, my mother B. Padmaja Rao for their support and love. I would specially like to thank my mom for guiding me to pursue my dreams and career I wanted, and always answering my numerous phone calls anytime of the day. Exceptional thanks to my brother Rahul Rao Basava, my friend, critic and guide. They equally deserve every bit of my success. I would like to also thank my grandmother Radha Lakshmi Kandula, my uncles Srinivas Kandula and Dr. Jai Krishna Kandula for their prayers and blessings. A very special thanks to my uncle Dr. Vittal Kumar Kandula for his support.

VITA

2004 – 2008	Bachelor of Engineering, Major: Civil Engineering
	Andhra University College of Engineering
	Visakhapatnam, Andhra Pradesh, India
2009 – 2010	Master of Science, Major: Civil Engineering, Structural Mechanics
	University of California, Los Angeles
	Los Angeles, CA, USA
2010 – 2013	Graduate program, Major: Civil Engineering, Structural Mechanics
	University of California, Los Angeles
	Los Angeles, CA, USA
2013 – 2015	Doctor of Philosophy, Major: Structural Engineering
	University of California, San Diego
	San Diego, CA, USA

Publications

1. J.-S. Chen, R. R. Basava, Y. Zhang, R. Csapo, V. Malis, U. Sinha, J. Hodgson, and S. Sinha, "Pixel-based meshfree modelling of skeletal muscles", Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, no. ahead-of-print, pp. 1-13, 2015.

2. R. R. Basava, J.-S. Chen, Y. Zhang, S. Sinha, U. Sinha, J. Hodgson, R. Csapo and V. Malis, "Pixel based meshfree modeling of skeletal muscles", Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications (Y. Zhang and J. Tavares, eds.), vol. 8641 of Lecture Notes in Computer Science, pp. 316-327, Springer International Publishing, 2014.

Conference Presentations

1. R. R. Basava, J.-S. Chen, "Image-based modeling of skeletal muscles based on strong form reproducing kernel collocation method for hyperelasticity", USNCCM 13, San Diego, CA, USA, July 26 – 30, 2015.

2. R. R. Basava, J.-S. Chen, Y. Zhang, S. Sinha, U. Sinha, J. Hodgson, R. Csapo and V. Malis, "Pixel based meshfree modeling of skeletal muscles", Paper and presentation, CompIMAGE 2014, Pittsburgh, Pennsylvania, USA, September 3 – 5, 2014.

Research experience

Spring 2013, Summer 2013, January 2014 – December 2015:

Graduate Student Researcher

Research Project: National Institute of Health (NIH) Project

Topic: Segmentation and image based meshfree modeling of skeletal muscles.

Advisor: Professor J.-S. Chen

Teaching experience

Fall 2012 and 2013:

Teaching Associate for Finite Element Methods course and lab instructor for Abaqus. Department of Civil and Environmental Engineering, UCLA.

Winter 2013:

Teaching Associate for Intermediate structural analysis course. Department of Civil and Environmental Engineering, UCLA

Summer 2015:

Teaching Assistant for Algorithms and Programming for Structural Engineering. Department of Structural Engineering, UCSD.

Winter 2011 – Spring 2012:

Teaching Assistant/Associate for lab sessions. Department of Physics and Astronomy, UCLA. Lead TA for Mechanics Lab in Winter and Spring, 2012

Summer 2010 and 2011:

Graduate student facilitator. UCLA Engineering, Center for Excellence in Engineering and Diversity. Course: Introduction to mechanics of deformable solids

ABSTRACT OF THE DISSERTATION

Meshfree Image-Based Reduced Order Modeling of Multiple Muscle Components with Connective Tissue and Fat

by

Ramya Rao Basava

Doctor of Philosophy in Structural Engineering

University of California, San Diego, 2015

Professor Jiun-Shyan Chen, Chair

Modeling of anatomically accurate skeletal muscle models is still a challenging area of research to date. In general, muscles have complex architectures with spatially varying fiber orientations. Using the conventional Finite Element analysis, the generated mesh needs to be conformed to the muscle geometry and material interfaces to obtain accurate simulation models. Poorly built meshes can also lead to significant errors in

analysis. To alleviate these issues and to provide effective transformation from images to simulation models, this work introduces the meshfree strong form Reproducing Kernel Collocation Method (RKCM) in context of nonlinear hyperelasticity. Further, Reduced Order Modeling (ROM) for nonlinear RKCM is developed to achieve simultaneous computational efficiency and controllable accuracy for large scale problems. The proposed methods are applied for modeling of skeletal muscles.

A segmentation framework is first developed for three dimensional model construction from Magnetic Resonance (MR) images using level set based technique, which incorporates multiple materials and muscle fiber orientations specified from Diffusion Tensor (DT) images. Further, a semi-automatic method of segmentation is proposed for segmenting individual muscles from images. A strong form RKCM is proposed to allow discretization of problem domain using MR and DT imaging data directly for effective image-based modeling, and to avoid the issues associated with domain integration and essential boundary imposition that typically exist in the Galerkin meshfree methods. In this work, nonlinear solution procedures and algorithms for RKCM analysis of hyperelasticity problems is formulated. It is shown that RKCM for nonlinear analysis provides more accurate results compared to Galerkin meshfree methods with quadratic bases using Gauss integration.

ROM for RKCM is further developed for nonlinear analysis, where a Least Squares Galerkin projection is introduced to project the over-determined system onto a discrete system with relatively lower dimension. For nonlinear analysis using RKCM, the construction of the stiffness matrix and force vector in each iteration is relatively less time consuming than that for Galerkin meshfree method using Gauss integration, making it a robust method for nonlinear model reduction. Sufficient accuracy can be achieved in the proposed method even by using only 1-2% of degrees of freedom of the full model in skeletal muscle modeling.

Chapter 1

Introduction

1.1 Motivation and scope

1.1.1 Image-based skeletal muscle modeling

The skeletal muscular system forms one of the major components of the human body mainly responsible for movement. Skeletal muscles are composed of different materials like muscle, tendon, connective tissues, fat, etc. which form an inter-dependent system to work together and perform different kinds of activities. Considerable efforts have been made in the past few decades, related to the modeling of the skeletal muscles represented by various suitable mechanical models, but development of mechanically accurate and computationally effective models is still an open area of research.

Segmentation plays a very important role in transforming the in vivo morphological data from images into structured information that can be used for physiological investigation or numerical simulation. The most commonly used images for segmentation are the Magnetic Resonance (MR) images. However, automatically segmenting individual muscle components from MR images poses a difficult problem since identification

of interfaces between muscle components is challenging, as the boundaries between different muscles are not usually distinguishable. Traditionally knowledge based methods are used to accomplish this segmentation. The first aim of this work is to develop a semi-automatic method to segment the individual muscles of the human lower leg from medical images, using the muscle fiber direction (obtained from Diffusion Tensor (DT) images) as an additional input along with MR image for segmentation. The segmented 3D multi-component images of skeletal muscles are then used as input to the meshfree modeling.

FEM has been successfully applied to modeling simplified models of the muscles, but modeling of complicated 3D muscle geometries increases the complexity of mesh generation for FE analysis. Poorly built meshes lead to mesh distortion and inaccurate results. In 3D subject specific models, fiber direction is measured at each of the muscle pixel points, which need to be interpolated at the integration points in FE model, which introduces additional approximation errors in the FE analysis. In the present work, image based modeling is introduced under the meshfree framework, which can be formulated using the weak form based Reproducing Kernel Particle Method (RKPM) and the strong form based Reproducing Kernel Collocation Method (RKCM), where the pixel points from images are directly used as nodes for discretization in the meshfree models. Under this meshfree framework, the muscle fiber directions and material properties are specified at the nodes, without additional interpolation. A smooth transition between different materials in the muscle is obtained by interpolating the material properties using the Reproducing Kernel (RK) shape functions and this smooth transition in material properties can be controlled by changing the support size of the RK kernel. This is suitable for modeling bio-materials without sharp discontinuities at material interfaces.

1.1.2 Meshfree methods based on strong form collocation

Meshfree methods like RKPM based on the weak form Galerkin governing equations have been successfully applied to linear and nonlinear problems, but certain issues like domain integration and imposition of essential boundary conditions remain complex. These complexities can be greatly reduced by the strong form methods which have been proposed for linear problems. The approximation functions used in strong form methods include Radial Basis (RB) functions or Reproducing Kernel (RK) functions, and the corresponding methods are called the Radial Basis Collocation Method (RBCM) and Reproducing Kernel Collocation Method (RKCM), respectively. RBCM gives highly accurate solutions with exponential rates of convergence, but the matrices are very ill-conditioned due to the non-local character of the RB functions. This makes RBCM limited to small scale simple geometry problems. On the other hand, RKCM yields a well-conditioned system due to its local character, and it converges algebraically. In this work, the RKCM framework is developed for nonlinear analysis in the context of hyperelasticity. RKCM is particularly suitable for modeling complex geometries like muscles, and the issues related to domain integration and imposition of essential boundary conditions in RKPM can be alleviated. In this research, RKCM is applied for image based modeling of skeletal muscles where the pixel points are chosen as the source and collocation points, and muscle properties like fiber direction are assigned at these pixel points.

1.1.3 Reduced order modeling of skeletal muscles

Reduced order models intend to significantly reduce discretization dimensions while maintaining sufficient accuracy through effective projection operators. Proper Orthogonal Decomposition (POD) is a popular technique to construct the projection op-

erator. In POD method for model reduction, suitable snapshots of the full scale solution are collected and used for construction of the bases for the reduced order model. For linear dynamic analysis, POD modes are obtained by collecting solution snapshots at different instances of time. In nonlinear static and dynamic problems snapshots can be collected from the selected applied load or time steps. In general for nonlinear analysis, the tangent stiffness matrix and residual force vector need to be constructed in every load step and iteration. The reduced order method can be effective only if the time taken to construct the tangent stiffness matrix and residual force vector is an insignificant portion of the total CPU. For solving nonlinear problem using the weak form type method such as RKPM where Gauss integration is used for domain integration, the tangent stiffness matrix and residual force vector construction in each time step consumes significant amount of time. Whereas in RKCM for nonlinear analysis, the time taken to construct these matrices is relatively insignificant compared to the solver time, hence its solution by reduced order modeling becomes more effective. In view of this, in this work, a Reduced Order Modeling (ROM) approach is proposed for strong form collocation method based on a Least Squares Galerkin projection to project the over-determined system of Gauss Newton incremental equations in RKCM. The proposed method provides an efficient reduced order computational model for nonlinear analysis with sufficient accuracy. This method has also been applied to static hyperelastic analysis of skeletal muscles to demonstrate its effectiveness.

1.2 Objectives

The objectives of this research are to develop effective computational frameworks to allow seamless transformation from images to simulation models and to provide efficient and accurate modeling of skeletal muscles. This is achieved by developing

a meshfree strong form Reproducing Kernel Collocation Method in context of nonlinear hyperelasticity where the simulation model can be constructed directly from MR and DT image data without the tedious procedures in mesh generation, domain integration, and imposition of essential boundary conditions. Further, a reduced order modeling approach for nonlinear RKCM is developed to achieve simultaneous computational efficiency and controllable accuracy for large scale modeling of skeletal muscles. The main tasks of this dissertation can be outlined as follows:

- *Model construction from medical images:*

  Each 2D MR image is segmented using active contours without edges method and are stacked to form the 3D model. For incorporating multiple materials, different stacks of images of the same model containing pre-segmented passive material (fat, Intra-Muscular Connective Tissue (IMCT)) information are used. A multiphase multichannel framework based on the active contours without edges framework is introduced using the fiber direction and MR intensity data in different channels to segment different regions in an image. The segmented pixel points associated with different muscle tissue materials as well as the DT image data at each pixel point are used directly as the discrete model for meshfree modeling of skeletal muscles.

- *Image based modeling using the weak form based Reproducing Kernel Particle Method (RKPM) and application to predict force decrease in aging muscles:*

  RKPM is first introduced for numerical simulation of skeletal muscles. The method of stabilized conforming nodal integration is used for integrating the matrix equations and the penalty method is used to impose the essential boundary conditions. The model consists of different materials and muscle fiber direction is input at each pixel point. The RK shape function is used to represent material hetero-

geneity with smooth transition across material interfaces. A study is performed on young and old medial gastrocnemius muscles of the human lower leg to predict the muscle force production. It can be observed from MR images of young and elderly people that there is considerable increase in passive materials (which include fat and IMCT) in the aging muscle. From the parametric study, a disproportionate decrease in force production is observed due to decrease in muscle volume between young and old muscles, which clearly shows that both the total volume of passive materials and their distributions contribute to the loss in force generation in aging muscles.

- *RKCM framework for nonlinear hyperelasticity and its application for skeletal muscle modeling:*

  In conventional meshfree methods based on the weak form, background meshes are required for the purpose of domain integration. Additionally, imposing essential boundary conditions requires special treatment since the meshfree approximation functions are usually not interpolating functions. This work introduces the meshfree strong form Reproducing Kernel Collocation Method for solving boundary value problems in context of nonlinear elasticity (hyperelasticity), and is applied for modeling of skeletal muscles using image data. The method is derived from least-squares form of nonlinear collocation equations followed by Gauss Newton linearization. The method can be equivalently derived from the minimization and linearization of Least Squares functional with quadrature. It is also shown that linearization of nonlinear strong form equations results in the Gauss Newton method of solving the least-squares form of nonlinear collocation equations. For convergence of the nonlinear iterations, it is show that the norm of the projection of the residual vector needs to be converging. The proposed

method is applied for image based modeling of skeletal muscles, which shows its effectiveness in modeling complex geometries.

- *ROM for RKCM and its application for skeletal muscle modeling:*

  ROM for the strong form RKCM is proposed, where a Least Squares Galerkin projection is used to project the over-determined system of equations onto a lower dimensional space. The proposed ROM can be applied to both linear and nonlinear problems. In nonlinear static analysis the method is applied to ROM of rubber elasticity examples and skeletal muscle models. Considerable reduction in computational time with marginal loss of accuracy in the ROM-RKCM modeling of skeletal muscles can be achieved.

All medical images used in this work are obtained from Department of Radiology, UCSD School of Medicine [1].

1.3 Outline of the dissertation

The remaining chapters of this dissertation are arranged as follows. In Chapter 2 literature review is given. In Chapter 3, a review of the image segmentation and bias correction methods used in this work are given. The proposed medical image segmentation methods and image based modeling of skeletal muscles using the weak form based RKPM are introduced in Chapter 4. Chapter 5 introduces the RKCM for nonlinear analysis of hyperelastic materials and its application to numerical simulation of skeletal muscles. Chapter 6 introduces the proposed model order reduction method based on Least Squares Galerkin projection, for RKCM in linear and nonlinear analysis. Conclusions and recommendations for future research are given in Chapter 7.

Chapter 2

Literature Review

2.1 Image segmentation methods and diffusion tensor imaging

2.1.1 Level set based methods for image segmentation

One of the first and the most influential work on using Level set methods was introduced by Osher and Sethian [2], which gives numerical solutions to solving problems associated with fronts moving with a curvature dependent speed, using level set functions for curve evolution. The level set function is an implicit function which is one dimension higher than the interface which is an evolving curve [3]. The motion of this interface is governed by the evolution of the level set function. Due to its implicit nature, the level set function can easily identify topological changes, like splitting and merging of the interface, which makes its application in segmentation problems very efficient. In the past few years a number of image segmentation methods have been developed based on variational principles incorporating the level set function which evolves a closed contour for detection of discontinuities in images. This evolving contour is called an 'Active

contour'.

Image segmentation based on active contours was first introduced by Kass et al. [4]. The concept was to evolve a contour called Snake by minimizing an energy functional which had terms involving internal energy which controls the smoothness of the contour; and an edge detection term which is based on the gradient of the image and forces the contour towards the edges. Later in the work of Caselles et al. [5], a geometric active contour model was presented for detecting edges by curve evolution rather than minimization of energy functional. Further Caselles et al. [6] used the geodesic approach to find a connection between the Snakes model and the geometric active contour model. Level set function was introduced into these curve evolution models, which could efficiently track topological changes. The above methods can be categorized as edge based methods and work precisely only in the presence of sharp edges in the images, that is the gradient of the image near the edges should be large. They may fail to detect weak boundaries or segment images with intensity inhomogeneity.

To overcome the difficulty of edge based segmentation, Chan and Vese [7, 8] proposed an active contour method without edges, famously called as the 'Chan Vese' model. This is a seminal work in the field of image segmentation and in the last decade many extensions and variations of this method were developed based on this same underlying concept. In this method the minimization of the Mumford-Shah functional [9] with active contours in a level set framework is introduced. The stopping term is based on the Mumford-Shah segmentation techniques, and the method can detect edges which are not necessarily based on change in image gradient. Due to this reason this method can be considered as a region based method, opposed to edge based methods. This Active Contour Without Edges (ACWE) model was extended by Vese and Chan to a generalized multiphase level set framework in [10] to segment images having multiple connected regions. In this method multiple level set curves are used to segment multiple

regions (or phases) and also triple junctions, without creating any overlap or vacuum in the segmented image. The union of all the zero isocontours of the level set functions represents the final boundaries of segmentation. Application of ACWE model to vector-valued images has been proposed by Chan et al. [11]. In this method the fitting term derived from the Mumford-Shah functional is averaged over all the channels of the vector-valued image for segmentation. This model can detect edges in at least one of the channels and not necessarily in all the channels. It can also detect edges even if some features of the image are missing in one channel or if channels are complete but are noisy.

Although the Chan-Vese model can efficiently segment images with piece-wise constant or homogeneous intensities in different regions of the image, it may fail to segment images correctly in the presence of intensity inhomogeneity or weak and blurred boundaries or to segment textures or patterns in the images. The region scalable fitting energy model proposed by Li et al. [12] was successfully applied in the segmentation of images in the presence of intensity inhomogeneity. This model takes into account a region scalable fitting energy at each point in the domain that locally approximate the image intensities on the two sides of the contour. This is an extension to the local binary fitting energy model [13] where the scalability of the local kernel could be of any size. Other methods for segmenting images with intensity inhomogeneity include local fitting energy models [14] and combination of local and global energy models [15, 16].

The Active Contours Without Edges based models can efficiently handle most of the image segmentation problems but are sometimes sensitive to initialization, since the energy function that is minimized is non-convex and has local minima. A completely convex formulation for the Chan-Vese segmentation model is proposed in [17].

2.1.2 Bias correction in medical images

MRI's are often corrupted by noise or intensity inhomogeneity due to the presence of non-uniform magnetic fields produced by the MRI machinery, radio frequency transmission and reception inhomogeneity, bandwidth filtering of the data of the frequencies transmitted by the RF coil and other patient specific disturbances such as distortions due to shape, position and orientation of the object inside the magnet, specific magnetic permeability and dielectric properties of the imaged object. Patient specific intensity inhomogeneity is more difficult to deal with. These artifacts may produce significant errors in medical image analysis like segmentation, and can also effect clinical interpretation. To overcome this to a certain extent shim coils are used in the MRI machine, but these do not eliminate inhomogeneity which is patient specific. Together this inhomogeneity in medical images is often called as a 'Bias field' which is commonly assumed to be a multiplicative field to the true image. A review of sources of intensity inhomogeneity in MRI and the various methods used to estimate this bias field in medical images can be found in [18] and the references therein. In this work bias correction of the MRI is done using the method proposed by Li et al. in [19], where the bias field is assumed to be a multiplicative field and is estimated using a region scalable local Gaussian kernel level set based formulation, similar to the region scalable method proposed by Li et al. in [12].

2.1.3 Prior knowledge based methods for segmentation of skeletal muscles

In the MRI of the human lower leg, the boundaries of different muscles are not distinct and sometimes only sparse or partial. This makes the problem of segmenting different muscles challenging as edges or boundaries are not distinguishable even visu-

ally. To overcome this segmentation problem, the most popular method currently used is to introduce prior knowledge into the segmentation process. This prior knowledge is usually a pre-segmented image with labels for different regions of segmentation and is called an Atlas image. The atlas image can be constructed in various ways, for example it can be a single image, or they can be a set of manually segmented images from experts, or a population based atlas (also called probabilistic atlas) and this atlas gives the ground-truth segmentation. Based on this atlas image as prior knowledge, different methods have been proposed for segmenting a new image, also called Target image or Reference image. Usually the segmentation is carried out in two steps: First the atlas image and the target image are spatially aligned and in this process, finding a suitable transformation between the images is called 'Registration' [20]. Once the images are registered, the segmented regions (labels) of the atlas image are propagated to the target image. The error in segmentation is dependent on the atlas chosen and the registration procedure. If the anatomical differences between the atlas and the target image are greater, the error in the registration step will be greater. A review of atlas-based segmentation methods for magnetic resonance brain images is given in [21]. Limited amount of literature is available which deals with segmentation of the skeletal muscles of the lower limb. They are briefly described in the following paragraph.

In [22] Baudin et al. proposed an automatic method for segmenting the thigh muscles using prior knowledge. Initial seeds for region based segmentation algorithm were generated by a sampling process where the partial edges of the muscles and the local variances were considered. The clustering of these seeds with respect to different muscle classes in the thigh MRI was achieved by a registration step between segmented atlas and the seeds in target image, using a Markov Random Field energy optimization formulation by comparing certain similarities. The outcome of this optimization process (clustered seeds) is fed to a segmentation algorithm (random walker) which

gives the final segmented regions. Similar works by Baudin et al. include [23, 24]. In [25, 26] Prescott et al. developed a semi-automatic method to segment the four quadriceps muscles in the thigh. A template (atlas image) was selected by finding the similarity between the normalized histograms of the target image and the template, by using the Kullback-Leibler divergence, which gives a measure of similarity between two probability distributions. The contours of the template are registered on to the target image semi-automatically. The segmentation is further refined using a multi-phase level set method. Segmenting DTI data was attempted in [27] by Neji et al. based on a prior knowledge based tensor clustering algorithm. They demonstrated the potential of DTI to discriminate different muscle groups. The prior knowledge is introduced by a Support Vector Machine (SVM) classifier which is trained to learn a certain number of diffusion tensors of the different muscle groups. A Markov Random Field (MRF) formulation is used for classifying the diffusion tensors, which takes into account the spatial information of the tensors and the SVM results. Other related works include; wavelet driven knowledge based segmentation of muscles [28, 29], segmentation and registration using multi-resolution simplex meshes [30], probabilistic multi-shape segmentation methods [31], segmentation using pose-invariant higher order graphs [32].

Although the prior knowledge based methods give reasonably good segmentation results with area overlaps ranging between 60%-90% for different muscles of the lower limb, the segmentation results are highly dependent on the prior introduced and the registration method used. Some of these methods require robust statistical shape learning methods which require huge number of manually segmented images as prior. Even after learning, there might still be considerable variability in the shape of the muscle between different individuals which has to be dealt with during segmentation.

2.1.4 Diffusion Tensor Imaging (DTI)

Water molecules in skeletal muscles move or diffuse mainly along the direction of the muscle fibers, which is called the principal direction. This movement of water is random translational motion (Brownian motion), which is due to thermal energy present in the molecules. In Diffusion Tensor Imaging (DTI) [33], typically a 3×3 diffusion tensor is obtained at each voxel (volumetric pixel) of the image which gives the diffusion coefficient in each direction and correlation between these directions. This diffusion tensor quantifies the random movement of water molecules in muscles (or tissues) and gives a good measure of the diffusion anisotropy in the muscles. DTI gives us important information regarding diffusion taking place (in vivo) at a microscopic scale, which is well below the scale of the actual MR image resolution. To evaluate or compare the diffusion properties at each voxel, commonly invariants of the diffusion tensor are computed, such as Fractional anisotropy or Mean Diffusivity which is given by the trace of the diffusion tensor. Currently diffusion tensor imaging is used widely to observe differences in various diseases and normal parts of brain, organs or muscles. Diffusion tensor images can be used to compare changes in water diffusion due to injury, differences due to exercise and other pathologies like myopathy (muscular diseases like muscle atrophy), Edema (elevation of water content in tissues), etc. and also to monitor disease progression. Some of the applications of DTI include; study of brain architecture and pathologies [34], fiber tracking in muscles [35, 36, 37, 38, 39], and segmentation using DTI data [40, 41, 42, 43, 44, 45, 46].

2.2 Meshfree methods

Meshfree methods are numerical solution procedures to approximate the solution to differential equations without the use of a mesh; compared to other numerical methods like the Finite Element Method (FEM), which requires a mesh in the domain or the finite difference methods which require a grid. In meshfree methods the domain is discretized with a set of nodes or points and an approximation to the solution is constructed using these nodes.

2.2.1 Background

Smoothed particle hydrodynamics (SPH) [47, 48] was one of the first implementation of a meshfree method used for modeling problems in astrophysics like motion of fluids in unbounded 3D space. In this method approximation to the solution is constructed using a set of particles and a kernel function. The kernel function provides certain smoothing at each particle depending upon the kernel width, which defines the locality of the kernel support. One of the major drawback of SPH is it exhibits tensile instability which occurs in the form of clustering of nodes and creation of artificial voids in regions subjected to large tension. Stability analysis for SPH was done by Swegle et al. in [49]. Another drawback is the lack of consistency in the SPH method. Due to this, treatment of boundary conditions becomes difficult and the accuracy of the SPH method deteriorates at the boundaries. This can be improved by modifying the kernel to satisfy consistency conditions.

Later Nayroles et al. [50] introduced the Diffuse Element Method (DEM) in which the standard finite element shape functions in the FEM were replaced with meshfree shape functions constructed using the Moving Least Squares (MLS) methods [51]. They also showed the calculation of derivatives can be done simply by taking the deriva-

tives of the basis functions. Further Belytschko et al. developed the Element Free Galerkin Method (EFGM) [52] which were based on the same MLS approximation but had improved accuracy over the DEM method, by taking exact derivatives which were more accurate and by imposing the essential boundary conditions using Lagrange multipliers as the meshfree shape functions did not obey Kronecker delta properties like FEM shape functions. They also showed that EFGM gave good results for nearly incompressible elasticity problems without introducing any additional constraints, compared to the results given by FEM. To overcome the inconsistency of the SPH method, Liu et al. [53] developed the Reproducing Kernel Particle Method (RKPM) where a correction term was included in the kernel functions which enforced consistency conditions and increased the accuracy of the method.

2.2.2 Reproducing Kernel Particle Method (RKPM) for large deformation analysis

The RKPM method was further extended by Chen et al. [54] for large deformation analysis of non-linear structures. In this work a Lagrangian material kernel was introduced which covers the same set of particles during material deformation and avoids tension instability due to large deformation. The RKPM shape functions were built using this kernel. High solution accuracy was observed in hyperelastic, elasto-plastic and also in nearly incompressible problems compared to solutions obtained using FEM. The application of the RKPM formulation for rubber hyperelasticity, involving large deformations, is presented in [55]. For nearly incompressible problems, in the limit of incompressibility, usage of higher order quadrature rules for required domain integration accuracy in RKPM, leads to volumetric locking and pressure oscillation. A pressure projection method was introduced by Chen et al. in [56] to elevate these problems. In

this method, the pressure is locally projected onto a lower order space to reduce the number of independent discrete constraint equations.

2.2.3 Domain integration

Domain integration poses a problem when the weak formulation of the governing equations is used for meshfree approximations of PDE solutions. In general for the meshfree methods in the Galerkin framework, Moving Least Squares (MLS) or the Reproducing Kernel (RK) shape functions are used. The RK shape functions are rational functions and have overlapping supports. In the earlier literature, the Gauss integration method [52, 57, 54] was used and later the node based methods [58, 59] were typically used for integrating the weak form Galerkin equations.

In the Gauss integration method, a background integration mesh is required for integrating the weak form matrices. The main source of error using gauss integration in meshfree methods is due to the misalignment of shape function supports with the integration cells which was first investigated in [60]. High order quadrature rules or low quadrature with a high number of quadrature cells are required to obtain good accuracy in numerical solutions [60, 59, 61]. Gauss integration is convergent and stable, but is prohibitively expensive for large scale problems.

On the other hand, node based methods are much simpler and more efficient, and maintain the true meshfree character for solving Galerkin equations, but it is well known that these methods lead to poor accuracy and numerical instability due to under integration [58, 62, 59]. The first derivatives of the shape functions are nearly zero at the nodes for displacement modes of short wavelengths, resulting in spurious oscillatory modes in the solution. Various methods have been proposed to alleviate the stability in nodal integration. A least squares stabilization was proposed by Beissel et al. in [58]. A stress point method was proposed in [63, 64] to alleviate the problem of tension instabil-

ity in SPH, which is based on avoiding derivatives of the shape functions at the nodes. A Stabilized Conforming Nodal Integration (SCNI) for Galerkin meshfree methods was proposed by Chen et al. in [59], which has improved accuracy and stability over direct nodal integration. In this method a strain smoothing approach is used which avoids taking derivatives at the nodes, and additionally, Galerkin linear exactness is satisfied, and optimal convergence rates can be achieved for linear basis. SCNI has also been applied to solving plate and shell problems in [65, 66, 67]. SCNI was further extended to solving nonlinear problems in [68]. A generalization of linear exactness in Galerkin meshfree methods to arbitrary order of exactness was proposed by Chen et al. in [61]. Here the requirements in domain integration to achieve arbitrary order exactness are obtained, and a method was proposed to satisfy these requirements using a Petrov-Galerkin approach. For problems with material separation like fragment-impact, reconstruction of SCNI representative nodal domains is extremely tedious and prohibitively expensive. For this class of problems a Stabilized Non-Conforming Nodal Integration (SNNI) method was proposed in [69], where this conforming condition is relaxed and suitable nodal domains for integration are chosen for each node, which do not necessarily conform. The smoothing based methods avoid taking direct derivatives of the shape functions at the nodes, which diminishes the spurious oscillatory modes.

In more recent studies, it has been shown that SCNI generates near zero low energy modes which are predominant in problems where the boundary influence is small, that is where the surface to volume ratio is small, or in problems with very fine discretization or with mesh refinement; due to loss of coercivity as detailed in [70]. Only the small amount of energy from the boundary nodes, contribute to the nonzero values at these near zero modes. A correction has been proposed to enhance the stability of SCNI and SNNI in [71, 70] which results in the modified SCNI (MSCNI) and SNNI (MSNNI) methods. Due to the addition of extra terms in the modified methods, there is consider-

able increase in CPU time for computation. An accelerated stabilized nodal integration has been proposed by Hillman et al. in [72] which is based on Taylor expansion using implicit gradients [73] of the meshfree RK shape functions.

In general for weak form based meshfree methods, integration is a very important consideration which affects the stability, efficiency and accuracy of the Galerkin-based meshfree method.

2.2.4 Imposition of essential boundary conditions

The meshfree approximation does not obey Kronecker delta properties like the FEM interpolation which makes the imposition of essential boundary conditions (EBC) difficult in Galerkin based meshfree methods. In the recent years many methods have been developed to enforce the EBC in meshfree methods. A good overview of these methods is given in [74]. Mainly, the Lagrange multiplier method was introduced in the EFG method [52], where EBC are imposed using Lagrange multipliers which are addition unknowns to be solved for. The Transformation method was introduced by Chen et al. in [54], where the meshfree shape functions are transformed to obey Kronecker delta property enabling the straight forward imposition of EBC. Further the Mixed Transformation method and the Boundary Singular Kernel method were proposed by Chen et al. in [75] for imposing EBC with more computational efficiency. The Penalty method was proposed by Zhu and Atluri [76] where an additional penalty term is added to the variational functional which enforces the EBC. A coupled finite element - element free Galerkin method was proposed by Belytschko et al. [77] in which a coupled shape function using FEM and EFG interpolation was constructed with a ramping function in the interface zone. By using this method FEM interpolation can be used for the essential boundary points so that EBC can be directly imposed at these points, as FEM shape functions obey Kronecker delta properties.

Instead of using the weak form Galerkin meshfree method, the strong form collocation methods [78, 79, 73] can also be used with a meshfree approximation, in which EBC can be directly imposed by enforcing the residuals to be zero at the collocation points on the essential boundary. This provides a very convenient way to directly impose the EBC without the use of additional changes to the weak form or to the shape functions to make them obey the Kronecker delta properties.

2.2.5 Strong form collocation methods for linear problems

In the strong form collocation methods, partial differential equations can be solved directly using the strong form of the equations and associated boundary conditions. One of the most popular method used is the Radial Basis Collocation Method (RBCM) where the solution is approximated using Radial Basis (RB) functions and more recently the Reproducing Kernel Collocation Method (RKCM) where the solution is approximated using the Reproducing Kernel (RK) shape functions. In collocation methods the domain is discretized using a set of source points, which are same as the nodes for discretizing the domain, and the solution is approximated using these points. Also a set of collocation points are chosen which are generally greater in number than the source points. The approximated solution is substituted in the strong form of the governing equations, and boundary conditions; and the residual is enforced to be zero at the collocation points.

The multiquadric RB functions were first applied for interpolating scattered data and for solving partial differential equations by Kansa et al. in [80, 81]. Usually the partial derivatives of the radial basis functions can be evaluated very easily and it also offers exponential convergence in numerical approximations. This has made the RB functions very popular, but one of the major drawbacks of RBCM is that the system of equations obtained is very ill-conditioned. Also standard RBCM gives larger solution

errors near the boundaries. Error analysis was done by Hu et al. [78], to show the unbalanced error between the domain, natural boundaries and the essential boundaries. A weighted RBCM was proposed where appropriate weights are added to the boundary condition equations to minimize this error. The weights should be properly chosen for a given problem to enhance numerical accuracy and convergence of solution.

A point collocation method based on RK approximation was proposed in [82]. The RK shape functions provide compact support and the system of equations obtained are well conditioned, but due to the RK approximation algebraic convergence is obtained rather than exponential convergence as obtained from RBCM. The convergence and computational complexity of RKCM was analyzed in [83], where the operation counts of RKPM and RKCM were compared and also showed that RKCM gives the same rate of convergence as RKPM. Error analysis for the RKCM method was given by Hu et al. in [79]. It was derived that for the RKCM method to converge the degree of monomial basis in the RK shape function should be greater than one. Further a reproducing kernel enhanced local radial basis collocation method was proposed by Chen et al. in [84], which combined the advantages of the RB and the RK shape functions by introducing a modified approximation called localized RB function, where the locality was due to the RK shape function. This method gave a better conditioned system and a convergence rate greater than that of the RKCM.

The main disadvantage of using the RKCM can be due to the increase in computational complexity for calculating the second order derivatives of the RK shape functions which are needed to approximate the strong form of the governing equations. A gradient reproducing kernel particle method was proposed by Chi et al. in [73], where the order of differentiation of the RK shape functions was reduced to first order and the partial differential equation is still solved using strong form collocation method.

In this work the RKCM is extended to solve nonlinear hyperelasticity problems

using the strong form of the governing equations in the Total Lagrangian framework.

2.3 Modeling of muscle tissues

The skeletal muscle is a complicated composite structure, which consists of different regions with different material components and forms a very efficient system to carry out the task of locomotion in mammals. In the past few decades, modeling of skeletal muscles has been a primary topic of investigation, as it provides a virtual framework to predict and solve complicated muscular disorders by giving an insight into the behavior of muscles under different conditions, for patient specific models which incorporate both material as well as geometrical data.

Skeletal muscles are typically modeled as nearly incompressible or fully incompressible hyperelastic materials. The Finite Element (FE) method has been widely applied for the modeling of biological tissues and skeletal muscles in the past few decades [85, 86, 87, 88, 89, 90]. FE method works accurately and provides good results for modeling simplified muscle geometries like the lumped parameter model [91]. Although 2D models provide insight into the working behavior of muscle mechanics, they fail to provide the true representation of the complex muscle architecture which may result in inaccurate predictions of muscle properties like fiber length and resulting force-length behavior [92]. Muscle geometries are complex in nature and representation of the actual geometry in modeling is essential to obtain accurate results. Despite its success, FE analysis poses some problems in accurately modeling 3D muscle geometries, with actual anatomical muscle fiber directions specified in the model. Usually 3D models of muscles are constructed from medical images through segmentation of the objects to be modeled. For FE modeling the geometries need to be meshed carefully using CAD tools before running the numerical analysis. In this process, if complex topologies ex-

ist in the muscle models, it leads to poorly built meshes which will provide inaccurate results due to excessive mesh distortion. Moreover, the representation of the actual muscle fiber direction by interpolation through templates in FE models introduces additional interpolation error as described in [93]. If fiber direction is input from diffusion tensor imaging data, it needs to be re-interpolated at the quadrature points which introduces some approximation error into the model. In the present work numerical modeling of skeletal muscles is carried out using the meshfree methods which eliminate issues related to loss in accuracy due to poor mesh construction and representation of the true muscle fiber direction.

2.4 Reduced Order Modeling (ROM)

Numerical modeling for problems with a large number of degrees of freedom is computationally very expensive and even if small changes are made in the model parameters or loading conditions the entire simulation has to be re-run. Model order reduction deals with constructing a reduced order model for a particular problem, which has very less number of degrees of freedom and still captures the essential features of the full scale model. In model order reduction a lower dimensional approximation of a full model solution is obtained by projecting the full scale solution onto a lower dimensional space. The various methods of reduction differ by taking in to account the choice of the projection basis used. Reduced order modeling is popularly used for design optimization problems or real time simulations. The procedure is carried out in two phases: on-line and off-line phases. In the off-line phase the projection basis are calculated by running full scale simulations as required. In the on-line phase the basis saved during the off-line phase are utilized to generate approximate real time solutions for the problem at hand.

2.4.1 ROM for nonlinear structural analysis

For nonlinear problems, the success of obtaining a good reduced solution depends on selecting the correct choice of the reduced basis, which can accurately reproduce the nonlinear structural behavior at any instant of time. Additionally as mentioned in [94]; the basis should be easy to generate, selection criteria for choosing number of basis to use should be simple, basis should have good approximation properties and be able to capture the solution for a large interval on the nonlinear solution path. Also, they should fully characterize the nonlinear response of the structure at least locally. Many methods for approximate solution of nonlinear mechanics were proposed in the 1970s to 1990s, which focused on reducing the number of degrees of freedom, by choosing appropriate basis or other formulations, such that the analysis is computationally less expensive. These methods worked efficiently for mild or moderately nonlinear problems. One of the first applications of finding approximate solution to nonlinear dynamic problems was done in [95]. In this method the principle of mode superposition is extended to nonlinear analysis. A local mode superposition technique is employed for solving the nonlinear dynamics problem, which states that small harmonic motions may be superimposed upon large static motion and that small forced motion maybe represented in terms of the nonlinear tangent stiffness frequency spectrum. Here the incremental motion in each time step is decomposed as a linear combination of a set of basis vectors and modal coordinates. The basis vectors are given by the eigenvectors corresponding to the lowest modes of the nonlinear tangent stiffness matrix at the beginning of the time step. This procedure requires solving an eigenvalue problem in each time step. Also the basis vectors need to be updated if the nonlinear behavior changes within a time step, making it computationally expensive. To reduce the computational cost, Stricklin and Haisler [96] proposed a pseudo-force approach where the nonlinearities in the problem

are included as a pseudo-force vector and the tangent is based on the linear stiffness matrix. This avoids construction of the nonlinear tangent stiffness matrix in every iteration within a load step. The method is suitable for moderately nonlinear problems or problems with local nonlinearities [97]. The reduced basis technique was proposed by Noor and Peters in [94] for nonlinear static response of structures. The method uses the finite element discretization and then the Rayleigh-Ritz basis functions to reduce the number of degrees of freedom. These basis functions are constructed using the static perturbation technique [98, 99] and consist of the nonlinear solution and its various order path derivatives. The reduced basis method works efficiently for the case of problems with geometric nonlinearities but is difficult to extend for problems with material nonlinearities. In [100], the use of reduced basis method was extended to nonlinear problems with prescribed non-zero edge displacements and mixed finite element formulations. In [101], Wilson et al. proposed a new set of orthogonal mass normalized Ritz vector basis for dynamic analysis of linear dynamic problems by direct superposition. The Ritz vectors are generated using a recursive relationship and the first vector is obtained from the solution of a static problem. This method does not require solving an eigenvalue problem and also the basis generated account for the loading on the structure. Idelsohn and Cardona proposed a computational algorithm for predicting the nonlinear dynamic response of a structure in [102]. In this method, the basis vectors for projecting the incremental generalized coordinates of the finite element solution are chosen to be the tangent eigenmodes together with some of the modal derivatives. Lesser number of basis updating was required compared to modal superposition method, but each change of basis resulted in an accumulation of truncation error in the velocities and accelerations. Further Idelsohn et al. extended the method proposed by Wilson et al. in [101] for reduced solutions of linear structural dynamic problems, to nonlinear problems by adding additional basis vectors, given by the derivatives of the ritz basis vectors with respect to

generalized displacement amplitudes of the reduced system in [103]. Updating the basis vectors is performed when required according to the characteristics of the nonlinear system. In [104] the reduced basis approach is applied for solving dynamical response of impulsively loaded structures. The initial Ritz vectors, derivatives of Ritz vectors and updated Ritz vectors are chosen as candidates for the reduced basis. The use of empirical eigenvectors, also known as Proper Orthogonal Decomposition (POD) modes, as basis vectors for problems in nonlinear solid dynamics was proposed by Krysl et al. in [105]. The POD basis are constructed by minimizing the average squared error between the vector to be projected and its projection on to a set of basis. These basis give a linear subspace which capture the response of the full nonlinear system. Using this method, frequent updating of the basis during the online phase of model reduction can be avoided. In this method the reduction is applied to the linearized nonlinear equations where the incremental displacement is projected on to the reduced space. This method works efficiently for nonlinear structural analysis problems and provides numerical results that compare well with the full scale model. In general, for solving reduced order models for nonlinear structural analysis, the saving of time is accomplished in solving the set of equations obtained. The time for assembling the stiffness and force vectors in each load step and iteration remains the same. A second level of system approximations have been proposed in the works [106, 107, 108], by constructing surrogates to the nonlinear residual and the tangent stiffness matrices by projection onto a different set of basis. This makes the reduced system independent of the dimension of the original full system and reduces computational complexity associated with stiffness and force assembly.

In this work, model order reduction method for linear and nonlinear Reproducing Kernel Collocation Method is developed. The basis for projection is chosen to be the POD basis which is reviewed in the next section.

2.4.2 ROM using Proper Orthogonal Decomposition (POD)

The method of Proper Orthogonal Decomposition (POD) is currently the most popular method used for forming the basis vectors for model reduction in both linear and nonlinear analysis and it has been widely applied in many fields of study including control theory (see chapter 9 in [109]), turbulent flows [110, 111], fluid dynamics [112], image analysis [113], weather prediction [114] and structural dynamics [105]. It was known by various names in different fields of study, for example: by Principal Component Analysis (PCA) in statistics, KarhunenLoeve Expansion (KLE) in stochastics and Empirical Orthogonal Eigenfunctions (EOE) in weather prediction.

In POD, a suitable approximation basis is found for the variable to be approximated say u, such that the average squared error between u and its orthogonal projection on the basis is minimized (mathematical statement of optimality). The detailed derivation of POD basis can be found in [115, 116], and has been reviewed in section 6.1 of chapter 6. The POD is optimal on an average. In order to carry out the reduction process, in the offline phase where the basis are constructed, the snapshots of the solution vector are collected when the full scale model is run. These solution vectors obtained at different time steps form the snapshot matrix. The orthogonal POD basis can be obtained either from the Eigenvalue Decomposition (EVD) of the correlation matrix of the snapshots or by Singular Value Decomposition (SVD) of the snapshots matrix. Both methods give equivalent results and are related to each other. Due to the optimality property of POD, the first r POD basis functions capture more energy than the first r functions of any other basis. That is the eigenvalues or singular values decay rapidly using POD. (Here the term 'energy' is used in context of dynamic problems where the variable to be approximated u is the velocity and the associated eigenvalues obtained from eigenvalue analysis represent twice the kinetic energy for the corresponding mode). This property

is most suitable and is a primary requirement for reduced order modeling, where the system characteristics need to be captured in the smallest number of basis.

In [113] Sirovich et al. introduced a snapshot POD method, where a smaller finite number of the snapshots of the full scale solution are used to construct the POD basis. This reduces the computational cost associated with carrying out the EVD of the snapshots matrix for large scale systems.

Chapter 3

Review of Image Segmentation and Bias Correction Methods

This chapter provides detailed review of the segmentation methods used in this work. First the level set function theory and standard two phase segmentation using active contours without edges is presented. This is followed by discussion of multiphase segmentation, segmentation of vector valued images and combined multichannel multiphase method of segmentation. A method of bias correction for medical images is also reviewed. Segmentation examples are given for all these methods. The multichannel multiphase method described will be used in the next chapter which deals with implementing segmentation of individual muscles in human lower leg using the MR and DT imaging data.

3.1 Level set function

One of the first and the most influential work on using Level set methods was introduced by Osher and Sethian [2] which gives numerical solutions to solving prob-

lems associated with fronts moving with a curvature dependent speed, using level set functions for curve evolution. The level set function is an implicit, sign-distance function in n^{th} dimension and its zeroth isocontour represents the moving interface, which is one dimension lower than the level set function dimension. Consider a closed moving interface $\Gamma(t)$ in the domain Ω and let $\phi(\boldsymbol{x}, t)$ be a continuous function defined as:

$$\phi(\boldsymbol{x}, t) > 0 \quad \text{if } \boldsymbol{x} \text{ is inside } \Gamma(t) \tag{3.1a}$$

$$\phi(\boldsymbol{x}, t) = 0 \quad \text{if } \boldsymbol{x} \text{ is on } \Gamma(t) \tag{3.1b}$$

$$\phi(\boldsymbol{x}, t) < 0 \quad \text{if } \boldsymbol{x} \text{ is outside } \Gamma(t) \tag{3.1c}$$

$\phi(\boldsymbol{x}, t)$ is said to be the level set function of the interface $\Gamma(t)$. The motion of this interface is governed by the evolution of the level set function. Some of the important parameters associated with the interface $\Gamma(t)$ can be defined in terms of the level set function as follows:

Unit outward normal:

$$\boldsymbol{n} = \frac{\nabla\phi(\boldsymbol{x}, t)}{|\nabla\phi(\boldsymbol{x}, t)|} \tag{3.2}$$

Curvature of $\Gamma(t)$:

$$\kappa = \nabla \cdot \boldsymbol{n} = \nabla \cdot \left(\frac{\nabla\phi(\boldsymbol{x}, t)}{|\bigtriangledown\phi(\boldsymbol{x}, t)|} \right) \tag{3.3}$$

Area inside $\Gamma(t)$:

$$A_{in} = \int_{\Omega} H(\phi(\boldsymbol{x}, t)) d\boldsymbol{x} \tag{3.4}$$

Area outside $\Gamma(t)$:

$$A_{out} = \int_{\Omega} [1 - H(\phi(\boldsymbol{x}, t))] d\boldsymbol{x} \tag{3.5}$$

Length of $\Gamma(t)$:

$$L = \int_{\Omega} \delta(\phi(\boldsymbol{x}, t)) \, |\bigtriangledown\phi(\boldsymbol{x}, t)| \, d\boldsymbol{x} \tag{3.6}$$

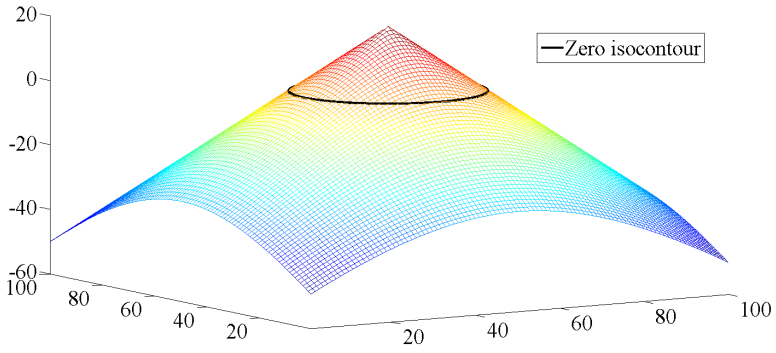


Figure 3.1: Signed distance level set function. Black contour represents the zero[th] isocontour of the level set function

where $H(\bullet)$ and $\delta(\bullet)$ represent the Heaviside function and Dirac delta function defined as follows:

$$H(z) = \begin{cases} 1 & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \tag{3.7}$$

$$\delta(z) = \frac{dH(z)}{dz} \tag{3.8}$$

For numerical purposes the regularized version of $H(\bullet)$ and $\delta(\bullet)$ are used in practice. For practical applications the level set function is chosen to be a signed distance function, which an implicit smooth function and is defined below [3]:

$$\phi(\boldsymbol{x}) = min(|\boldsymbol{x} - \boldsymbol{x}_I|) \quad \text{for all } \boldsymbol{x}_I \in \Gamma \tag{3.9}$$

The sign distance function also obeys the property of $|\nabla\phi(\boldsymbol{x})| = 1$. Example of a signed distance function is shown in figure 3.1, where the black contour indicates the zero[th] level set curve. During numerical time evolution the level set function tends to deviate from a signed distance function which may cause numerical instability. To avoid this, the level set function is re-initialized periodically to make it a signed distance function. Additional re-initialization equation given in (3.10) should be solved which

makes the procedure computationally expensive. In the present work re-initialization has not been required to be done in any of the examples given.

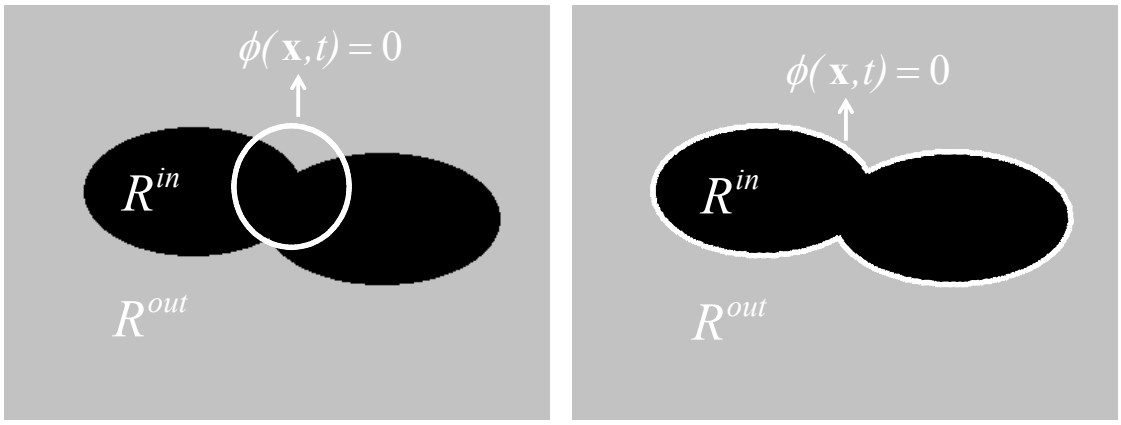
$$\frac{\partial \psi}{\partial t} = sign(\phi(\boldsymbol{x}, \tau))(1 - |\nabla \psi|)$$
$$\psi(\boldsymbol{x}, 0) = \phi(\boldsymbol{x}, \tau) \tag{3.10}$$

In the past two decades level set based methods have been very successfully applied in the field of image segmentation, where the level set function is used for boundary or interface identification. The main advantage of using the level set surfaces in image segmentation is their ability to easily identify topological changes in images, that is, splitting and merging of regions.

3.2 Active contours without edges (ACWE)

The ACWE method was proposed by Chan and Vese [7] and is derived from the piecewise constant Mumford-Shah functional in a level set framework for image segmentation. The following paragraph illustrates this method in the context of segmenting an image which has two distinct intensities - 2 phase segmentation.

Consider an evolving curve $\Gamma(t)$ in the domain Ω. Let the image be formed by two regions, inside region R^{in} surrounded by the outside region R^{out} as shown in figure 3.2(a). The two regions have approximately piece-wise constant intensities given by u_o^{in} in R^{in} and u_o^{out} in R^{out}, respectively. The aim is to obtain the final boundary $\Gamma(t)$ such that $u_o(\boldsymbol{x}) \in R^{in} \approx u_o^{in}$ and $u_o(\boldsymbol{x}) \in R^{out} \approx u_o^{out}$ as shown in figure 3.2(b), where $u_o(\boldsymbol{x})$ is the pixel intensity at point $\boldsymbol{x}$. To achieve this, following fitting functional is minimized:



(a) Initial 0^{th} isocontour of level set function (b) Final contour obtained after segmentation

Figure 3.2: Two phase (region) segmentation using ACWE method

$$\Pi(c_1, c_2, \Gamma) = \mu\, length(\Gamma) + \nu\, Area\, inside(\Gamma) + \lambda_1 \int_{inside\ \Gamma} (u_o(\boldsymbol{x}) - c_1)^2 d\boldsymbol{x} + \lambda_2 \int_{outside\ \Gamma} (u_o(\boldsymbol{x}) - c_2)^2 d\boldsymbol{x} \quad (3.11)$$

where μ, ν, λ_1, λ_2 are non-negative parameters which indicate weights for their respective terms in the above functional. Introducing the level set formulation into the above functional, equation (3.11) can be re-written as:

$$\begin{aligned}\Pi(c_1, c_2, \phi) = \mu \int_\Omega \delta(\phi(\boldsymbol{x}))|\nabla\phi(\boldsymbol{x})|d\boldsymbol{x} + \nu \int_\Omega H(\phi(\boldsymbol{x}))d\boldsymbol{x} \\ + \lambda_1 \int_\Omega (u_o(\boldsymbol{x}) - c_1)^2 H(\phi(\boldsymbol{x}))d\boldsymbol{x} \\ + \lambda_2 \int_\Omega (u_o(\boldsymbol{x}) - c_2)^2 [1 - H(\phi(\boldsymbol{x}))]d\boldsymbol{x} \quad (3.12)\end{aligned}$$

The expressions for the unknowns c_1 and c_2 can be obtained by minimizing (3.12) with

respect to c_1 and c_2, respectively, keeping all other variables constant:

$$c_1 = \frac{\int_\Omega u_o(\boldsymbol{x}) H(\phi(\boldsymbol{x})) d\boldsymbol{x}}{\int_\Omega H(\phi(\boldsymbol{x})) d\boldsymbol{x}} \tag{3.13}$$

$$c_2 = \frac{\int_\Omega u_o(\boldsymbol{x})[1 - H(\phi(\boldsymbol{x}))] d\boldsymbol{x}}{\int_\Omega [1 - H(\phi(\boldsymbol{x}))] d\boldsymbol{x}} \tag{3.14}$$

It can be seen from equations (3.13) and (3.14) that c_1 represents the average of $u_o(\boldsymbol{x})$ inside $\Gamma(t)$ and c_2 the represents the average of $u_o(\boldsymbol{x})$ outside $\Gamma(t)$. Minimizing the functional (3.12) with respect to ϕ keeping c_1 and c_2 constant, the Euler-Lagrange equations are deduced. In this process, the minimization is parameterized by artificial time $t \geq 0$, which gives the evolution of the level set function and associated boundary conditions as follows:

$$\begin{aligned}
&\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_o(\boldsymbol{x}) - c_1)^2 + \lambda_2 (u_o(\boldsymbol{x}) - c_2)^2 \right] \quad \text{in } \Omega \\
&\frac{\delta(\phi)}{|\nabla \phi|} \nabla \phi \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega \\
&\phi(\boldsymbol{x}, 0) = \phi^o(\boldsymbol{x}) \quad \text{(given)}
\end{aligned} \tag{3.15}$$

Here Ω represents the interior domain of the image and $\partial\Omega$ represents the boundary of the image. As mentioned in the previous section the regularized versions of $H(\bullet)$ and $\delta(\bullet)$ as given below are used for numerical implementation.

$$H_\varepsilon(z) = \frac{1}{2} \left[1 + \frac{2}{\pi} arctan \left(\frac{z}{\varepsilon} \right) \right] \tag{3.16}$$

$$\delta_\varepsilon(z) = \frac{1}{\pi} \left[\frac{\varepsilon}{\varepsilon^2 + x^2} \right] \tag{3.17}$$

where ε is a positive parameter. ε is taken to be equal to 0.5 in all the examples provided in further chapters unless specified. The boundary condition in equation (3.15)

which corresponds to the image boundary does not play a significant role in finding the segmented regions which are in the interior of the image. In all examples in this work a constant term is used for updating ϕ near the boundaries. Equation (3.15) is solved numerically using a semi-implicit scheme, which is detailed in Appendix A.

3.2.1 Example: Two phase ACWE segmentation

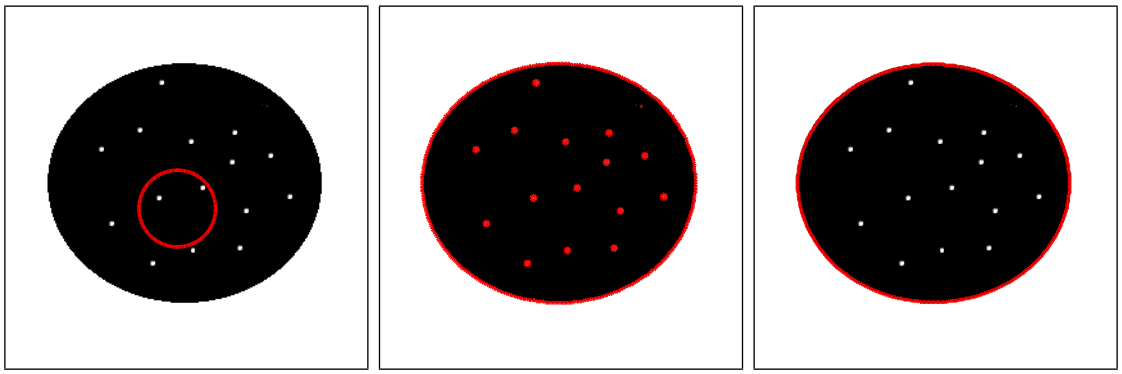
This example illustrates the two phase image segmentation using one level set function for the gray scale image shown in figure 3.2. The values of parameters used are $\mu = 100$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\nu = 0$ and $\triangle t = 1$. The final contour obtained which segments the 2 regions, is shown in figure 3.2(b).

3.2.2 Effect of weighting terms in the ACWE model

The weighting terms in the ACWE model should be suitably adjusted for different images to achieve the required segmentation. For this purpose it is very important to understand the effects produced in segmentation by changing these weighting terms.

3.2.2.1 Effect of μ:

From equation (3.12) it can be observed that μ behaves like a penalty term to the total length of the contour. If μ is increased and the functional in equation (3.12) is minimized, it strongly imposes the condition that the length of the contour should be a minimum, so only larger objects are detected in the process of segmentation. The effect of this parameter μ can be alternatively explained in terms of the smoothness of the boundary. If the boundary to be detected should be smooth, the value of μ should be increased. Thus complex winding contours which may have sharper curves are avoided, (that is lengthy contours are avoided). μ provides a means to regularize the length of



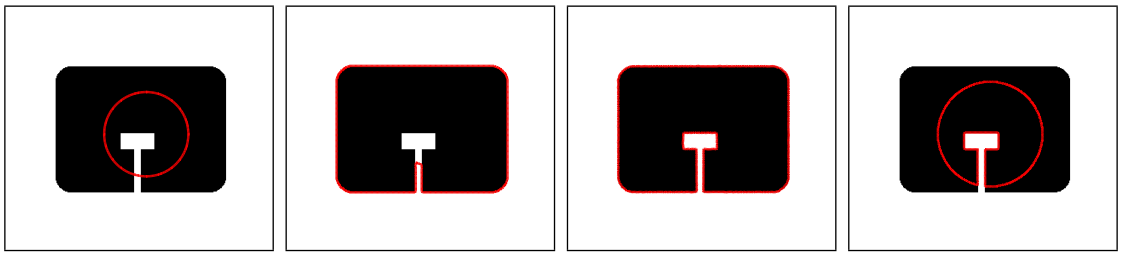
(a) Initial contour (b) $\mu = 1$ (c) $\mu = 10^5$

Figure 3.3: Effect of μ

the contour. This length regularization plays an important role especially in segmenting noisy images or images where grouping of objects is required.

This is illustrated in the following example. A noisy black and white image is selected, which has dotted regions in the interior. Only μ is varied and the remaining parameters are taken as $\lambda_1 = 1, \lambda_2 = 1, \nu = 0$ and $\triangle t = 1$. The effect of changing μ is illustrated in figure 3.3. As shown in figure 3.3(b), when $\mu = 1$ is taken, the contours for all the interior dots are identified. In figure 3.3(c), where the value of $\mu = 10^5$, only the larger boundary is identified, since the total length is forced to be a minimum.

Special case of $\mu = 0$: In this case there is no length regularization term. So every point is capable of being a boundary point, that is, it is possible to form a contour around each point in the domain if each pixel has a different color, as there is no restriction on the length of the detected boundary. This case cannot be used to segment noisy images or images where grouping of objects needs to be done. Only a limited set of images which have piece-wise constant colors can be segmented by ignoring the length regularization term. Although in this case, the level set curve evolution equation reduces to an ordinary differential equation and can be solved quickly, requiring very less computational time.



(a) Initial contour (b) $\nu = -10$ (c) $\nu = 0$ (d) $\nu = 10$

Figure 3.4: Effect of ν

3.2.2.2 Effect of ν:

Effect of ν is similar to μ except that the regularization is done on the inside area of the 2 phase segmentation. As the value of ν is increased the inside area is forced to be a minimum and vice versa. This trend is illustrated in figure 3.4. The values of other parameters used are $\lambda_1 = 1, \lambda_2 = 1, \mu = 1$ and $\triangle t = 1$.

3.2.2.3 Effect of λ:

In equation (3.12) the terms having λ as weight (λ_1 for the inside region and λ_2 for outside region) are proportional to the variance of intensities in their respective regions. As λ is increased and the functional is minimized, the condition that the variance should be a minimum is strongly imposed in that region, that is during segmentation the level set contour evolves such that the intensities in the inside region (or outside region) are more homogeneous. The effect of this λ parameter becomes prominent in images having gradients in intensities or inhomogeneity or bias.

3.3 Multiphase level set method

The multiphase level set method [10] is an extension to the 2 phase ACWE formulation described in the previous section. It is a generalized multiphase level set

framework to segment images having multiple connected regions (or phases). In this method multiple level set curves are used to segment multiple regions and also triple junctions, without creating any overlap or vacuum in the segmented image. k level set curves can segment up to 2^k regions in the image. The union of all the zero isocontours of the level set functions represents the final boundaries of segmentation.

In the following paragraphs multi-phase segmentation method using 3 level sets is described in detail which can segment up to 8 regions in an image. In general this method can be extended to any number of level set curves.

3.3.1 Segmentation using three level set functions

Consider an image I. Three level set curves partition the domain of the image into 8 regions as shown in figure 3.5. The boxes denote the region numbers. Each region is denoted by Ω_i where $i = 1, 2, \cdots, 8$ denotes the region number. Extending the 2 phase formulation, the functional for this multi-phase segmentation can be written as

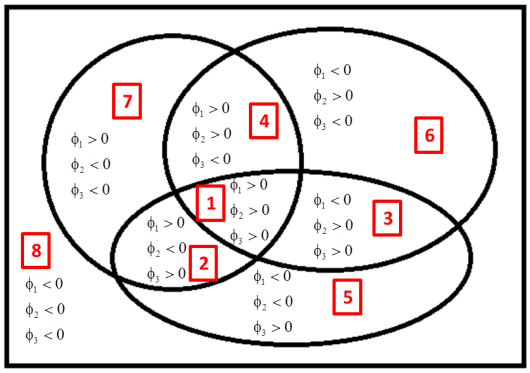


Figure 3.5: Image domain partition with 3 Level sets

follows in level set based formulation (taking $\nu = 0$ for convenience):

$$\Pi(c_1, c_2, \cdots, c_8, \phi_1, \phi_2, \phi_3) = \sum_{k=1}^{3} \mu_k \int_{\Omega} \delta(\phi_k(\boldsymbol{x}))|\nabla \phi_k(\boldsymbol{x})| d\boldsymbol{x} + \sum_{p=1}^{8} \left(\lambda_p \int_{\Omega} (u_o(\boldsymbol{x}) - c_p)^2 M_p d\boldsymbol{x} \right) \tag{3.18}$$

where

$$\begin{aligned} M_1 &= H(\phi_1)H(\phi_2)H(\phi_3) \\ M_2 &= H(\phi_1)(1 - H(\phi_2))H(\phi_3) \\ M_3 &= (1 - H(\phi_1))H(\phi_2)H(\phi_3) \\ M_4 &= H(\phi_1)H(\phi_2)(1 - H(\phi_3)) \\ M_5 &= (1 - H(\phi_1))(1 - H(\phi_2))H(\phi_3) \\ M_6 &= (1 - H(\phi_1))H(\phi_2)(1 - H(\phi_3)) \\ M_7 &= H(\phi_1)(1 - H(\phi_2))(1 - H(\phi_3)) \\ M_8 &= (1 - H(\phi_1))(1 - H(\phi_2))(1 - H(\phi_3)) \end{aligned} \tag{3.19}$$

Here p denotes the label for different regions ($p = 1, 2, \cdots, 8$). k denotes the label for the level set curve. M_p denotes the region in terms of the level sets. c_p is an unknown constant and gives the mean intensity in region Ω_p. The expressions for c_p's can be obtained by minimizing equation (3.18) with respect to each c_p and keeping ϕ_1, ϕ_2, ϕ_3 and other c's fixed and are given by the following equation:

$$c_p = \frac{\int_{\Omega} u_o(\boldsymbol{x}) M_p d\boldsymbol{x}}{\int_{\Omega} M_p d\boldsymbol{x}} \tag{3.20}$$

Three independent Euler-Lagrange equations are obtained by minimizing the functional with ϕ_1, ϕ_2, ϕ_3, respectively, each time keeping $c_1, c_2, \cdots, c_8$ constant. In the following

expressions for Euler-Lagrange equations, the weighting parameters are taken as $\lambda_p = \lambda$ for $p = 1, 2, \cdots, 8$ and $\mu_k = \mu$ for $k = 1, 2, 3$. Minimizing the functional (3.18) w.r.t ϕ_1, keeping $c_1, c_2, \cdots, c_8, \phi_2, \phi_3$ constant:

$$\begin{aligned}\frac{\partial \phi_1}{\partial t} &= \delta(\phi_1)\mu\nabla \cdot \left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) \\ &+ \delta(\phi_1)\lambda[-(u_o - c_1)^2 H(\phi_2)H(\phi_3) - (u_o - c_2)^2(1 - H(\phi_2))H(\phi_3) \\ &- (u_o - c_4)^2 H(\phi_2)(1 - H(\phi_3)) - (u_o - c_7)^2(1 - H(\phi_2))(1 - H(\phi_3)) \\ &+ (u_o - c_3)^2 H(\phi_2)H(\phi_3) + (u_o - c_5)^2(1 - H(\phi_2))H(\phi_3) \\ &+ (u_o - c_6)^2 H(\phi_2)(1 - H(\phi_3)) + (u_o - c_8)^2(1 - H(\phi_2))(1 - H(\phi_3))] \quad (3.21)\end{aligned}$$

Minimizing the functional (3.18) w.r.t ϕ_2, keeping $c_1, c_2, \cdots, c_8, \phi_1, \phi_3$ constant:

$$\begin{aligned}\frac{\partial \phi_2}{\partial t} &= \delta(\phi_2)\mu\nabla \cdot \left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) \\ &+ \delta(\phi_2)\lambda[-(u_o - c_1)^2 H(\phi_1)H(\phi_3) - (u_o - c_3)^2(1 - H(\phi_1))H(\phi_3) \\ &- (u_o - c_4)^2 H(\phi_1)(1 - H(\phi_3)) - (u_o - c_6)^2(1 - H(\phi_1))(1 - H(\phi_3)) \\ &+ (u_o - c_2)^2 H(\phi_1)H(\phi_3) + (u_o - c_5)^2(1 - H(\phi_1))H(\phi_3) \\ &+ (u_o - c_7)^2 H(\phi_1)(1 - H(\phi_3)) + (u_o - c_8)^2(1 - H(\phi_1))(1 - H(\phi_3))] \quad (3.22)\end{aligned}$$

Minimizing the functional (3.18) w.r.t ϕ_3, keeping $c_1, c_2, \cdots, c_8, \phi_1, \phi_2$ constant:

$$\begin{aligned}\frac{\partial \phi_3}{\partial t} &= \delta(\phi_3)\mu\nabla \cdot \left(\frac{\nabla \phi_3}{|\nabla \phi_3|}\right) \\ &+ \delta(\phi_3)\lambda[-(u_o - c_1)^2 H(\phi_1)H(\phi_2) - (u_o - c_3)^2(1 - H(\phi_1))H(\phi_2) \\ &- (u_o - c_2)^2 H(\phi_1)(1 - H(\phi_2)) - (u_o - c_5)^2(1 - H(\phi_1))(1 - H(\phi_2)) \\ &+ (u_o - c_4)^2 H(\phi_1)H(\phi_2) + (u_o - c_6)^2(1 - H(\phi_1))H(\phi_2) \\ &+ (u_o - c_7)^2 H(\phi_1)(1 - H(\phi_2)) + (u_o - c_8)^2(1 - H(\phi_1))(1 - H(\phi_2))] \quad (3.23)\end{aligned}$$

where $u_o = u_o(\boldsymbol{x})$. Equations (3.21), (3.22), (3.23) are the three uncoupled evolution equations and can be solved independently in each time iteration. After steady state is reached and final ϕ_1, ϕ_2, ϕ_3 are computed, pixels belonging to the final regions of segmentation can be obtained by taking the values of ϕ's for different regions as given in figure 3.5. The regularized versions of H and δ as given in (3.16), (3.17) respectively, are used for numerical implementation.

The multi-phase formulation can be extended to 'n' level set functions which can segment up to 2^n regions. The general form of the functional is given as:

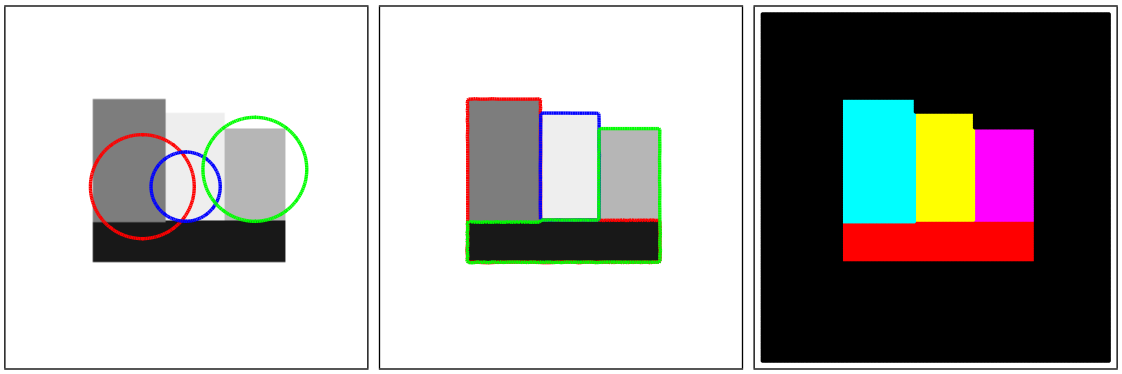
$$\Pi(c_p\text{'s}, \phi_k\text{'s}) = \sum_{k=1}^{n} \mu_k \int_{\Omega} \delta(\phi_k)|\nabla\phi_k| d\boldsymbol{x} + \sum_{p=1}^{2^n} \left(\lambda_p \int_{\Omega_p} (u_o(\boldsymbol{x}) - c_p)^2 d\boldsymbol{x} \right) \quad (3.24)$$

3.3.1.1 Example: Five region segmentation

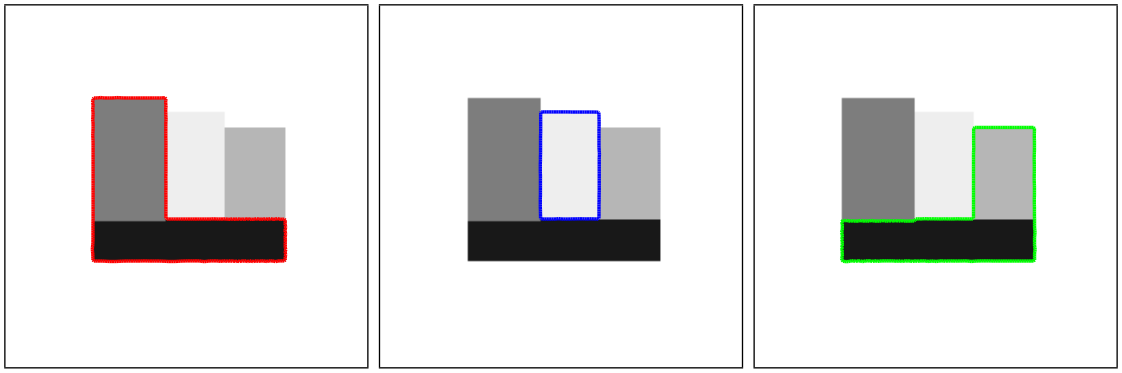
Figure 3.6 shows an image which has triple junctions and 5 regions to be segmented (including background). The parameters used in the segmentation are $\lambda_p = 1$, $\nu_k = 0$, $\mu_k = 100$ and $\triangle t = 1$. The initial contours taken for the 3 level set curves are shown in figure 3.6(a). The final contours obtained after segmentation and the segmented regions are shown in figures 3.6(b) and 3.6(c), respectively. Figures 3.6(d) to 3.6(f) shown the final 0^{th} isocontours of the 3 level set functions, the union of which gives the segmentation boundary.

3.4 Vector valued image segmentation

The ACWE model has been extended to the segmentation of multichannel images in [11]. A multi-channeled image can be split into multiple channels, which have to be input together for segmentation. Each pixel has a vector as input in a multichannel image. For example a RGB image consists of 3 channels for red, green, blue respectively

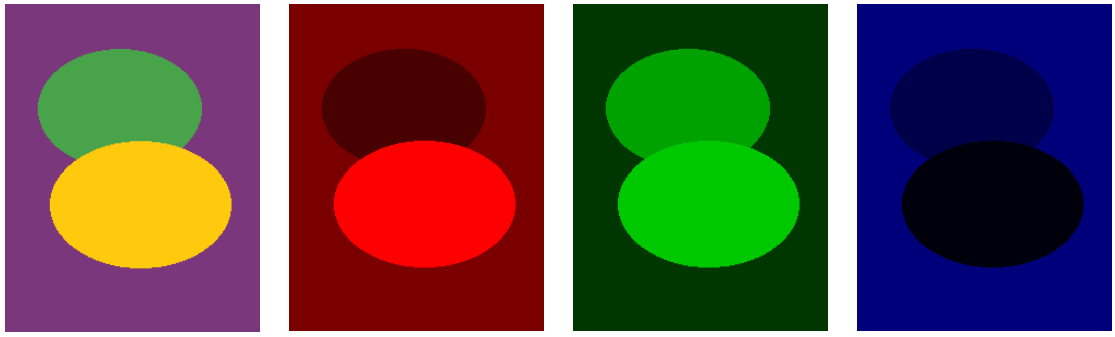


(a) Initial contours (b) Final contours (c) Segmented regions



(d) Final contour $\phi_1 = 0$ (e) Final contour $\phi_2 = 0$ (f) Final contour $\phi_3 = 0$

Figure 3.6: Multiphase Segmentation using 3 Level set functions



(a) RGB image (b) Red channel of RGB image (c) Green channel of RGB image (d) Blue channel of RGB image

Figure 3.7: RGB image channel splitting

as shown in figure 3.7. Each channel will be a grayscale image. Similarly a CMYK image can be split into 4 channels for cyan, magenta, yellow and black, respectively. The functional for the vector valued image segmentation for a 2 phase multi-channeled image is given by:

$$\begin{aligned}\Pi(c_1, c_2, \phi) = \mu \int_\Omega \delta(\phi)|\nabla\phi| d\boldsymbol{x} + \nu \int_\Omega H(\phi) d\boldsymbol{x} \\ + \int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_1^i (u_o^i(\boldsymbol{x}) - c_1^i)^2 H(\phi) d\boldsymbol{x} \\ + \int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_2^i (u_o^i(\boldsymbol{x}) - c_2^i)^2 [1 - H(\phi)] d\boldsymbol{x}\end{aligned} \tag{3.25}$$

where N is the number of channels in the image, superscript i denotes the channel number, $u_o^i(\boldsymbol{x})$ denotes the pixel intensity in the i^{th} channel at point $\boldsymbol{x}$ and c_1^i, c_2^i are the unknown constants which denote the mean values of the intensities inside and outside the level set contour respectively, in the i^{th} channel. As can be seen in equation (3.25) multichannel segmentation includes averaging the contributions from the forcing terms in each channel. The expressions for c_1^i, c_2^i can be obtained by minimizing the functional

in (3.25) with c_1^i and c_2^i respectively, for $i = 1, 2, \cdots, N$ and keeping ϕ constant.

$$c_1^i = \frac{\int_\Omega u_o^i(\boldsymbol{x}) H(\phi) d\boldsymbol{x}}{\int_\Omega H(\phi) d\boldsymbol{x}} \tag{3.26}$$

$$c_2^i = \frac{\int_\Omega u_o^i(\boldsymbol{x})[1 - H(\phi)] d\boldsymbol{x}}{\int_\Omega [1 - H(\phi)] d\boldsymbol{x}} \tag{3.27}$$

The Euler Lagrange equations can be obtained by minimizing the functional with respect to ϕ keeping c_1^i, c_2^i fixed.

$$\begin{aligned}
&\frac{\partial \phi}{\partial t} = \delta(\phi) \mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \\
&+ \delta(\phi) \left[-\frac{1}{N} \sum_{i=1}^{N} \lambda_1^i (u_o^i(\boldsymbol{x}) - c_1^i)^2 + \frac{1}{N} \sum_{i=1}^{N} \lambda_2^i (u_o^i(\boldsymbol{x}) - c_2^i)^2 \right] \quad \text{in } \Omega \\
&\frac{\delta(\phi)}{|\nabla \phi|} \nabla \phi \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega \\
&\phi(\boldsymbol{x}, 0) = \phi^o(\boldsymbol{x}) \quad \text{(given)}
\end{aligned} \tag{3.28}$$

where Ω represents the interior domain of the image and $\partial\Omega$ represents the boundary of the image. The regularized versions of H and δ as given in (3.16), (3.17) respectively, are used for numerical implementation.

3.5 Multiphase multichannel segmentation

The multiphase segmentation can be applied to multichannel images by combining the multiphase and multichannel segmentation methods. A general framework of

the functional for an image with N channels, using n level set curves is given by:

$$\Pi(c_p\text{'s}, \phi_k\text{'s}) = \sum_{k=1}^{n} \mu_k \int_{\Omega} \delta(\phi_k)|\nabla\phi_k| d\boldsymbol{x} + \sum_{p=1}^{2^n} \left(\int_{\Omega_p} \frac{1}{N} \sum_{i=1}^{N} \left(\lambda_p^i (u_o^i(\boldsymbol{x}) - c_p^i)^2\right) d\boldsymbol{x} \right) \tag{3.29}$$

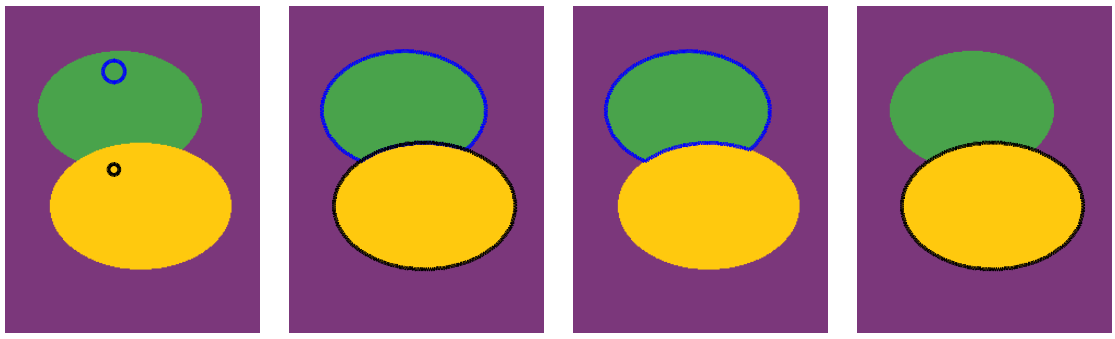
where $u_o^i(\boldsymbol{x})$ denotes the pixel intensity in the i^{th} channel at point $\boldsymbol{x}$ and c_p^i denotes the unknown constant which represents the mean value of intensity in the region Ω_p of channel i. Euler Lagrange equations and the expressions for obtaining c_p^i will be elaborated further in the next chapter.

3.5.1 Example: Three channel, three region segmentation

Figure 3.8(a) shows an RGB image made up of 3 channels and has 3 regions to be segmented. A multiphase multichannel segmentation is done using 2 level set functions for 3 channels ($k = 2$, $i = 3$ and $p = 4$ in equation (3.29)). The values of parameters used are: $\lambda_p^i = 1$, $\mu_k = 1$, $\triangle t = 1$ and $\varepsilon = 1$. The results of the segmentation are shown in figure 3.8(b) and the final contour of each level set function in figures 3.8(c) and 3.8(d), respectively.

3.6 Estimating and correcting bias field in MR images

Almost all magnetic resonance images usually have inhomogeneous intensities in the image (uneven shading in the image) as shown in figure 3.9(a) which are due to presence of inhomogeneous magnetic fields produced in the MR imaging machine, differences in frequency transmission and reception of the RF coils and many other factors like patient positioning, dielectric properties etc. In image segmentation the intensities of the MR image at each pixel point are input for segmentation. If the MR



(a) Initial contours (b) Final contours (c) Final contour $\phi_1 = 0$ (d) Final contour $\phi_2 = 0$

Figure 3.8: 2 Level set 3 channel segmentation of RGB image

image is having bias in the form of inhomogeneous intensities at certain regions in the image, this will result in giving incorrect segmentation results. As a pre-processing step for segmentation, the MR image has to be corrected for this bias.

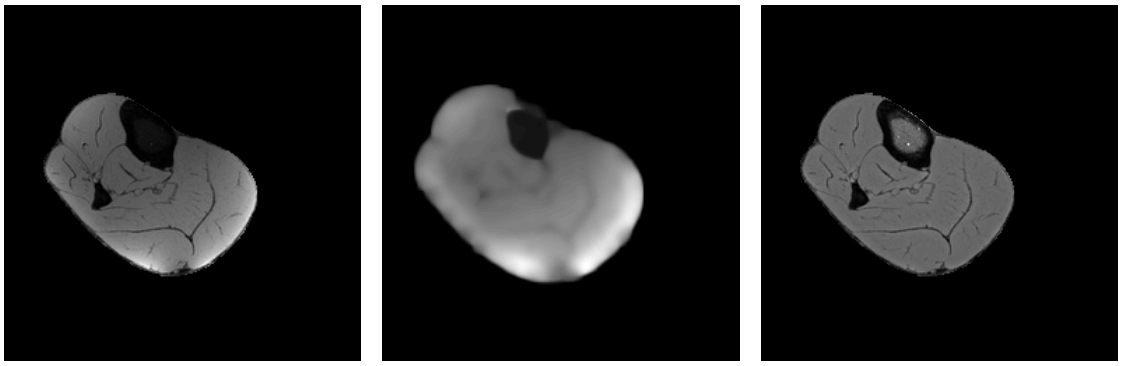
The bias correction method used in this work is proposed by Chunming Li et al. in [19], where a method was developed for combined segmentation and bias correction in images, particularly with application to MR images. In this work this method is used for bias correction of the MR images only. The following paragraphs outline the procedure of the method in the framework of two phase level set segmentation.

At each pixel point the observed image can be modeled as:

$$I(\boldsymbol{x}) = b(\boldsymbol{x})J(\boldsymbol{x}) + n(\boldsymbol{x}) \tag{3.30}$$

where I is the observed image which has to be corrected for bias, J is the true image which is assumed to have piecewise constant intensities, b is the unknown bias field which is approximately estimated in the method, n is the additive noise in the image which can be neglected for MR images.

The bias field is assumed to be slowly varying and approximated to be a constant in the local neighborhood of a pixel point. Consider a circular neighborhood $\mathcal{O}_y$ at each



(a) Original MR image with bias (b) Estimated bias field (c) Bias corrected MR image

Figure 3.9: Bias correction in MR images

point $\boldsymbol{y}$ in the image domain Ω. The bias field $b(\boldsymbol{x})$ at any point in the support $\mathcal{O}_y$ is assumed to be approximately equal to $b(\boldsymbol{y})$:

$$b(\boldsymbol{x}) \approx b(\boldsymbol{y}) \quad \text{for } \boldsymbol{x} \in \mathcal{O}_y \tag{3.31}$$

The image to be segmented has two regions given by Ω_1 and Ω_2. Using (3.31) the intensities in each region can be approximated to be:

$$\begin{aligned} I(\boldsymbol{x}) &\approx b(\boldsymbol{y})J(\boldsymbol{x}) + \underbrace{n(\boldsymbol{x})}_{=0}, \quad \text{for } \boldsymbol{x} \in \mathcal{O}_y \cap \Omega_i, \quad i = 1, 2 \\ I(\boldsymbol{x}) &\approx b(\boldsymbol{y})c_i \end{aligned} \tag{3.32}$$

where c_i are the unknown fitting values of the true image intensities in the region Ω_i. The energy formulation is based on a local intensity clustering property. A local fitting energy is defined at each point in the image given by:

$$E_y^{Fit}(f_1(\boldsymbol{y}), f_2(\boldsymbol{y})) = \sum_{i=1}^{2} \lambda_i \int_{\mathcal{O}_y \cap \Omega_i} |I(\boldsymbol{x}) - f_i(\boldsymbol{y})|^2 d\boldsymbol{x} \tag{3.33}$$

where λ_i are non-negative parameters and

$$f_i(\boldsymbol{y}) = b(\boldsymbol{y})c_i \tag{3.34}$$

Taking the local support $\mathcal{O}_y$ at each point as a Gaussian kernel, equation (3.33) can be re-written as:

$$E_y^{Fit}(f_1(\boldsymbol{y}), f_2(\boldsymbol{y})) = \sum_{i=1}^{2} \lambda_i \int_{\Omega_i} K_\sigma(\boldsymbol{y} - \boldsymbol{x})|I(\boldsymbol{x}) - b(\boldsymbol{y})c_i)|^2 d\boldsymbol{x} \tag{3.35}$$

where K_σ is the Gaussian kernel function with standard deviation (or scale parameter) $\sigma > 0$ defined below:

$$\begin{aligned} K_\sigma(\boldsymbol{u}) &= \frac{1}{\sqrt{2\pi}\sigma} e^{(-|\boldsymbol{u}|^2/2\sigma^2)} \\ \int K_\sigma(\boldsymbol{u}) d\boldsymbol{u} &= 1 \end{aligned} \tag{3.36}$$

This implements the condition that $K_\sigma(\boldsymbol{y} - \boldsymbol{x}) = 0$ for $\boldsymbol{x} \notin \mathcal{O}_y$. The segmentation is governed by the intensities in the local support and the bias field is approximated better if the support size is smaller. For segmentation a contour Γ should be found, which minimizes all E_y^{Fit} in the entire image domain. Integrating E_y^{Fit} over the entire image domain Ω gives the functional:

$$E = \int_{\Omega} E_y^{Fit}(f_1(\boldsymbol{y}), f_2(\boldsymbol{y})) d\boldsymbol{y} \tag{3.37}$$

Re-writing the above functional in 2 phase level set formulation:

$$E(\phi, f_1(\boldsymbol{y}), f_2(\boldsymbol{y})) = \sum_{i=1}^{2} \lambda_i \int_\Omega \left[\int_\Omega K_\sigma(\boldsymbol{y}-\boldsymbol{x}) |I(\boldsymbol{x}) - f_i(\boldsymbol{y})|^2 M_i(\phi(\boldsymbol{x})) d\boldsymbol{x} \right] d\boldsymbol{y} + \mu L(\phi)\boldsymbol{x} + \gamma P(\phi) \tag{3.38}$$

where ϕ is the level set function, μ and γ are non-negative constants >0 and

$$\begin{aligned} M_1(\phi) &= H(\phi) \\ M_2(\phi) &= [1 - H(\phi)] \\ L(\phi) &= \int_\Omega |\nabla H(\phi(\boldsymbol{x}))| d\boldsymbol{x} = \int_\Omega \delta(\phi(\boldsymbol{x})) |\nabla \phi(\boldsymbol{x})| d\boldsymbol{x} \\ P(\phi) &= \int_\Omega \frac{1}{2} \left(|\nabla \phi(\boldsymbol{x})| - 1 \right)^2 d\boldsymbol{x} \end{aligned} \tag{3.39}$$

$L(\phi)$ is the length regularization term of the zero[th] isocontour of the level set function. $P(\phi)$ is an additional term added to the function to keep ϕ as a signed distance function. This eliminates the need to re-initialize the level set function. $b(\boldsymbol{y})$ and c_i are unknowns and are required to be evaluated. Minimizing the functional in equation (3.38) with respect to each c_i keeping b and ϕ constant gives:

$$c_i = \frac{\int_\Omega (b * K_\sigma) I M_i(\phi(\boldsymbol{y})) d\boldsymbol{y}}{\int_\Omega (b^2 * K_\sigma) M_i(\phi(\boldsymbol{y})) d\boldsymbol{y}} \quad \text{for } i = 1, 2 \tag{3.40}$$

where '$*$' is the convolution operator. Minimizing the functional in equation (3.38) with respect to b and keeping c_i and ϕ constant gives:

$$b = \frac{\left(I \sum_{i=1}^{2} c_i M_i \right) * K_\sigma}{\left(\sum_{i=1}^{2} c_i^2 M_i \right) * K_\sigma} \tag{3.41}$$

To obtain the Euler Lagrange equations for curve evolution the functional in equation

(3.38) is minimized with respect to ϕ keeping b and c_i constant:

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= -\delta(\phi(\boldsymbol{x}))(\lambda_1 e_1 - \lambda_2 e_2) \\ &+ \mu\delta(\phi(\boldsymbol{x}))\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \gamma\left(\nabla^2\phi - \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right) \\ \text{Given} \quad & \phi(\boldsymbol{x}, 0) = \phi^o(\boldsymbol{x}) \end{aligned} \tag{3.42}$$

where

$$e_i = \int_\Omega K_\sigma(\boldsymbol{y} - \boldsymbol{x})|I(\boldsymbol{x}) - b(\boldsymbol{y})c_i|^2 d\boldsymbol{y}; \quad i = 1, 2 \tag{3.43}$$

Equation (3.42) is solved numerically using finite difference methods where every gradient is discretized using the central difference scheme as follows:

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = \frac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}}{h^2} + \frac{\phi_{i,j+1} + \phi_{i,j-1} - 2\phi_{i,j}}{h^2} \tag{3.44}$$

where $h = 1$ is the pixel spacing in the image.

The estimated bias field is obtained at the end of the time evolution. Greater accuracy in the estimation is obtained if the local support is smaller and the time evolution is done for more number of steps. The bias corrected image (J), can be obtained by dividing the intensity at a pixel point with the corresponding bias at that point.

$$J(\boldsymbol{x}) = \frac{I(\boldsymbol{x})}{b(\boldsymbol{x})} \tag{3.45}$$

The original MR image, estimated bias field and the bias corrected MR image for an example cross sectional of human lower leg are shown in figure 3.9.

Chapter 4

Segmentation and Image Based Meshfree Modeling of Skeletal Muscles

In this chapter the meshfree Reproducing Kernel Particle Method (RKPM) for three-dimensional modeling of skeletal muscles is introduced. This approach allows for construction of simulation model based on pixel data obtained from medical images. The muscle fiber direction obtained from Diffusion Tensor (DT) imaging and material properties are input at each pixel point. The reproducing kernel (RK) approximation allows a representation of material heterogeneity with smooth transition. The application of the proposed framework for modeling skeletal muscles in the human lower leg is demonstrated. For the construction of 3D pixel based models from images, two segmentation frameworks are proposed, which are based on 2D segmentation of in-plane images and then stacking them to get the 3D models. Further, a method is proposed to obtain 3D normals for the boundary of the stacked 3D muscle model.

4.1 Proposed segmentation frameworks

Segmenting medical images plays a very important role in transforming the in vivo morphological data from images into structured information that can be used for physiological investigation or numerical simulation. This information obtained from images can be used to achieve improvements in the field of medical education, treatment and image guided surgery. In the present day, accurate segmentation and processing of medical images is essential for constructing 3D models of the objects of interest from 2D image slices.

4.1.1 Level set segmentation of muscles tissues using DT enriched MR images

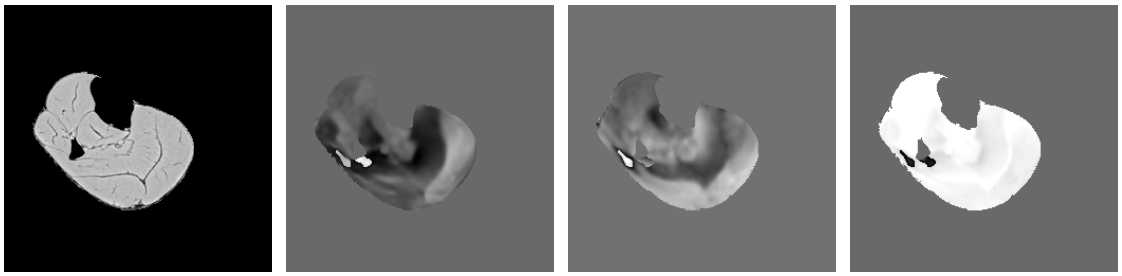
Automatic segmentation of all the muscles together from MR image can be easily done using standard segmentation methods after correcting the imaging artifacts, since all the muscles have the same intensity and this makes the boundary identification a relatively easy task. However, automatically segmenting individual muscle components from MR images poses a difficult problem since identification of interfaces between muscle components is challenging, as the boundaries between different muscles are not clearly distinguishable or are partial. In this work, a framework is developed for the segmentation of individual muscles which have different fiber orientations. The idea proposed in this work is to use the MR image intensities in combination with muscle fiber direction obtained from DT images, to semi-automatically segment individual muscles. A combined multiphase multichannel method of segmentation as described in Chapter 3 is used to implement this idea. A multichannel method is used to incorporate the MR and DT images in different channels for segmentation. A multiphase framework

is required to segment connected regions in an image. Sometimes in the final segmentation obtained, regions need to be combined in order to segment a particular muscle, due to which the method is termed as 'Semi-Automatic' since manual interaction is necessary in particular cases. The details of the segmentation procedure are described in the following paragraphs. Here the method is implemented to segment muscles of the human lower leg.

In DT imaging typically a 3×3 diffusion tensor as given in equation (4.1) is obtained at each voxel (volumetric pixel) of the image which gives the diffusion coefficient in each direction and correlation between these directions.

$$\overline{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \tag{4.1}$$

The lead Eigen Vector (EV) (eigen vector associated with largest eigenvalue) of this diffusion tensor gives the direction of maximum water diffusion, which is assumed to be collinear with the muscle fiber direction at that point. This EV data consists of an unit vector $[e_1, e_2, e_3]^T$ as input at each pixel point which is obtained as 3 images as shown in figure 4.1 where there is an image for each of the vector components e_i. For the purpose of segmentation the EV data is scaled suitably to match the range of intensities of the MR image. Each of the images, that is, MR image, e_1, e_2 and e_3 components of the EV data are taken as 4 channels for segmentation. As the final segmentation has 7 regions (6 major muscles and background) to be identified, 3 level set functions which can segment up to 8 regions in the image, are used for segmentation. The functional for the 3 level set 4 channel segmentation is given by:



(a) Bias corrected MR image (b) e_1 component of EV (c) e_2 component of EV (d) e_3 component of EV

Figure 4.1: MR image and grayscale images of EV data components

$$\Pi(c_p^i, \phi_1, \phi_2, \phi_3) = \sum_{k=1}^{3} \mu_k \int_{\Omega} \delta(\phi_k) |\nabla \phi_k| d\boldsymbol{x} + \sum_{p=1}^{8} \left(\int_{\Omega_p} \frac{1}{4} \sum_{i=1}^{4} \left(\lambda_p^i (u_o^i(\boldsymbol{x}) - c_p^i)^2 \right) d\boldsymbol{x} \right) \tag{4.2}$$

where ϕ_k are the level set functions, $u_o^i(\boldsymbol{x})$ denotes the pixel intensity in the i^{th} channel at point $\boldsymbol{x}$, p denotes the number of regions that can be segmented and c_p^i denotes the unknown constants which represents the mean value of intensity in the region Ω_p of the i^{th} channel. μ_k and λ_p^i denote the weights for their respective terms. In level set formulation equation (4.2) can be re-written as:

$$\Pi(c_p^i, \phi_1, \phi_2, \phi_3) = \sum_{k=1}^{3} \mu_k \int_{\Omega} \delta(\phi_k) |\nabla \phi_k| d\boldsymbol{x} + \sum_{p=1}^{8} \left(\int_{\Omega} \frac{1}{4} \sum_{i=1}^{4} \left(\lambda_p^i (u_o^i(\boldsymbol{x}) - c_p^i)^2 \right) M_p d\boldsymbol{x} \right) \tag{4.3}$$

where

$$
\begin{aligned}
M_1 &= H(\phi_1)H(\phi_2)H(\phi_3) \\
M_2 &= H(\phi_1)(1 - H(\phi_2))H(\phi_3) \\
M_3 &= (1 - H(\phi_1))H(\phi_2)H(\phi_3) \\
M_4 &= H(\phi_1)H(\phi_2)(1 - H(\phi_3)) \\
M_5 &= (1 - H(\phi_1))(1 - H(\phi_2))H(\phi_3) \\
M_6 &= (1 - H(\phi_1))H(\phi_2)(1 - H(\phi_3)) \\
M_7 &= H(\phi_1)(1 - H(\phi_2))(1 - H(\phi_3)) \\
M_8 &= (1 - H(\phi_1))(1 - H(\phi_2))(1 - H(\phi_3))
\end{aligned} \tag{4.4}
$$

The expression for c_p^i can be obtained by minimizing equation (4.3) with respect to each c_p^i; keeping ϕ_1, ϕ_2, ϕ_3 and other c's fixed:

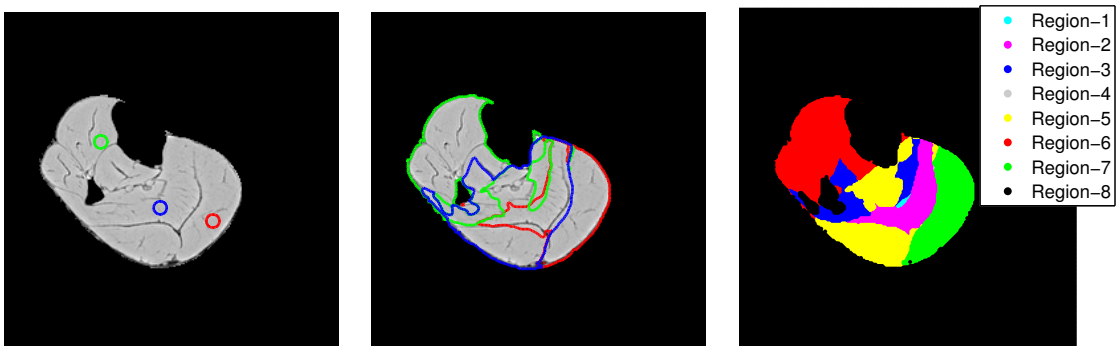
$$
c_p^i = \frac{\int_\Omega u_o^i(\boldsymbol{x}) M_p d\boldsymbol{x}}{\int_\Omega M_p d\boldsymbol{x}} \tag{4.5}
$$

The 3 Euler Lagrange equations are obtained by minimizing the functional given in equation (4.3) with ϕ_1, ϕ_2, ϕ_3 respectively each time keeping c_p^i fixed. The derived equations are given in Appendix B. They are solved using a semi-implicit finite difference scheme. Since the muscle fibers in each muscle more or less point in similar direction, it is important to account for this variability (for example due to curvature in muscle geometry) within each muscle to obtain accurate segmentation results. In the multiphase multichannel ACWE model above, the parameter to consider for this is the λ terms which are the weights for the forcing terms in formulation (4.3). As mentioned in section 3.2.2.3, if the value of λ is decreased it allows for slightly more variance of intensities in the region to be segmented. And if λ is increased, more homogeneous re-

gions are obtained after segmentation. If a particular muscle has multiple homogeneous regions, these regions can be grouped to form the semi-automatically segmented muscle. Depending on the muscle to be segmented and considering the amount of variance in each muscle from images, this parameter can be adjusted appropriately to obtain the best segmentation results.

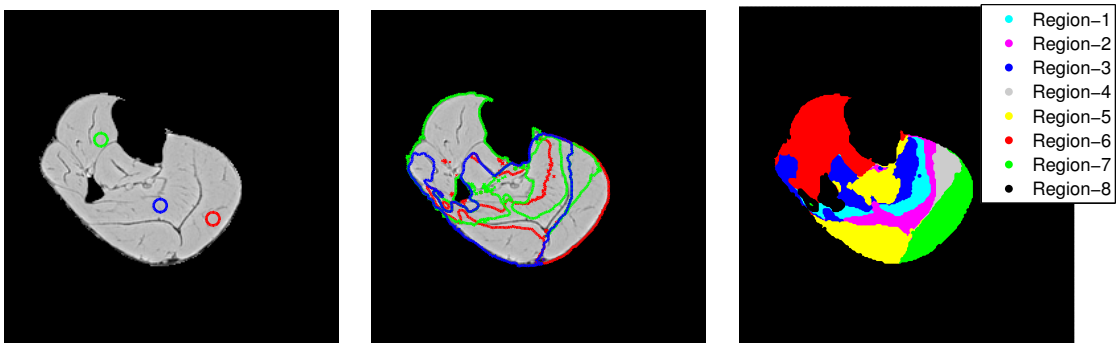
4.1.1.1 Example 1: Multiphase multichannel DT enriched MR image segmentation

This example illustrates the segmentation obtained using the proposed formulation and the effect of the λ parameter. The 4 channels as given in figure 4.1 are used. Parameter values of $\lambda_p^i = \lambda$, $\nu_k = \nu = 0$ and $\mu_k = \mu = 1$ are taken. A time step increment of $\triangle t = 1$ is used. The eigenvector data is scaled suitably so that it has the similar range of intensities as of the MR image data. This gives equivalent level of contribution from the MR image and the EV data. Two different values of λ are considered and the results of segmentation are given in figures 4.2 and 4.3, for $\lambda = 0.02$ and $\lambda = 2$ respectively. The manual segmentation for the MR image slice is given in figure 4.4 for comparison. As can be seen in figures 4.2 and 4.3, as the value of λ is increased, more number of homogeneous regions are formed in the segmentation. These regions can be grouped together to form segmented individual muscles. This process of grouping makes the method semi-automatic. Unipennate (unidirectional) muscles like the medial gastrocnemius in the lower leg can be automatically or semi-automatically segmented depending upon the value of the λ used. Some muscles (like the soleus) which are multipennate can be segmented by grouping regions together.



(a) Initial contours (b) Final contours (c) Segmented regions

Figure 4.2: Multiphase multichannel DT enriched MR image segmentation for $\lambda = 0.02$



(a) Initial contours (b) Final contours (c) Segmented regions

Figure 4.3: Multiphase multichannel DT enriched MR image segmentation for $\lambda = 2$

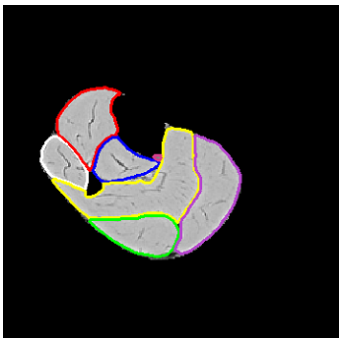


Figure 4.4: Manual segmentation of the 6 major muscles in MR image

4.1.1.2 Semi-automatic segmentation of medial gastrocnemius muscle

It can be observed from the segmentation results in the previous section, that the medial gastrocnemius (MG) muscle has been segmented very well, almost automatically using the proposed method. Figures 4.5 and 4.6 show the comparison between the segmented MG muscle using the semi-automatic multiphase multichannel segmentation and the manual segmentation for MR image slices 15 and 30 respectively. The common points between the semi-automatic and manual segmentation are also shown. To quantitatively assess the segmentation results, the amount of overlap or similarity between the manual and the semi-automatic segmentation is calculated from each image using a similarity index called Dice Coefficient (DC), [23, 117] which is defined as follows:

$$DC = 2\frac{n\{A \cap M\}}{n\{A\} + n\{M\}} \tag{4.6}$$

where A and M are the set of pixels in semi-automatic and manual segmentation respectively. $n\{X\}$ denotes the number of pixels in set X. The DC is sensitive to differences in both size of the segmented regions and location of pixels in these regions. A value of $DC > 0.7$ is considered to be excellent agreement between the two sets of pixels A and M, [117]. Additionally the Pearson's Correlation Coefficient (PCC) [118], which is used to compare two images can be calculated as follows:

$$PCC = \frac{\sum_i (f_i - f_m)(g_i - g_m)}{\sqrt{\sum_i (f_i - f_m)^2}\sqrt{\sum_i (g_i - g_m)^2}} \tag{4.7}$$

where f_i is the intensity of the i^{th} pixel in the first image, g_i is the intensity of the i^{th} pixel in the second image, f_m is the mean intensity in the first image and g_m is the mean intensity in the second image.

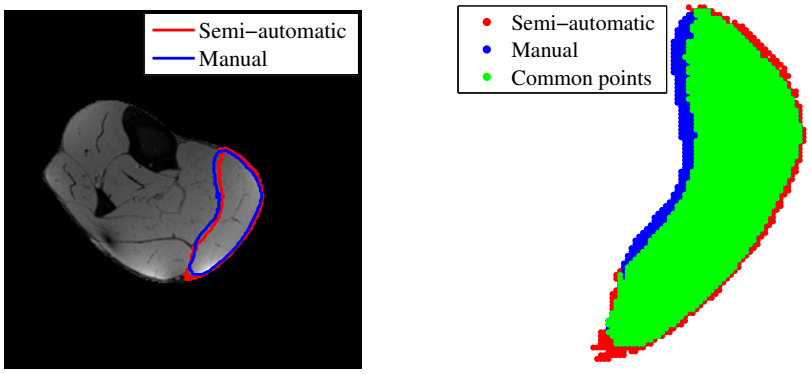


Figure 4.5: MR image slice 15: DC = 0.90, PCC = 0.89

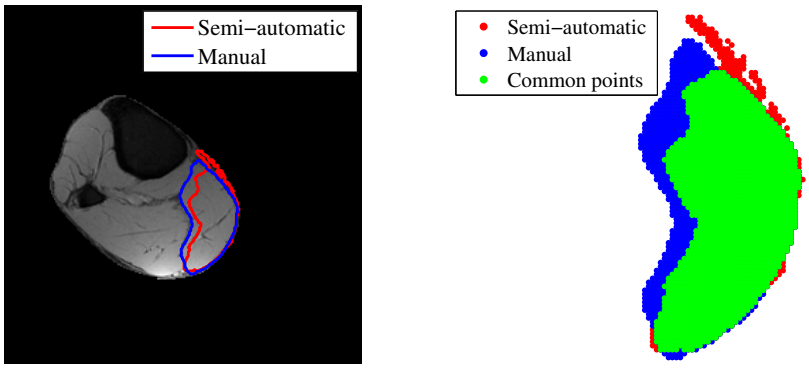


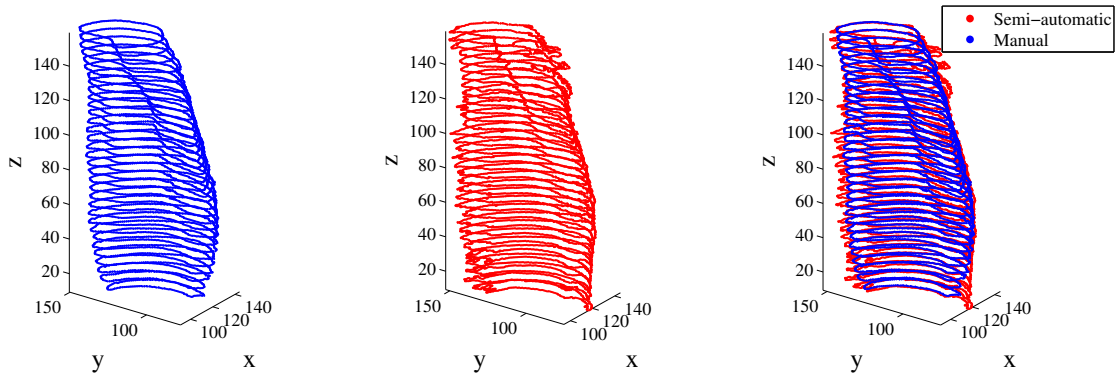
Figure 4.6: MR image slice 30: DC = 0.84, PCC = 0.84

4.1.1.3 Example 2: Segmentation of the medial gastrocnemius muscle

The MG muscle is segmented from 33 images and compared with manual segmentation as shown in figure 4.7. An average DC value of 0.8744 and an average PCC value of 0.8740 are obtained when comparing the semi-automatic and manual segmentations, which shows very good agreement between the two.

The following points should be noted for the implementation of the proposed method:

- The DT image giving the three components of the muscle fiber direction should be overlapping as closely as possible with the MR image for accurate segmentation.
- Each MR image should be bias corrected such that muscle regions have uniform intensity without shading.
- In the process of segmentation the number of regions that need to be segmented is predefined depending upon the number of individual muscles in the cross-section of the MR image. It is observed that more number of regions might be needed to segment each muscle and these regions should be grouped manually to form individual muscles.

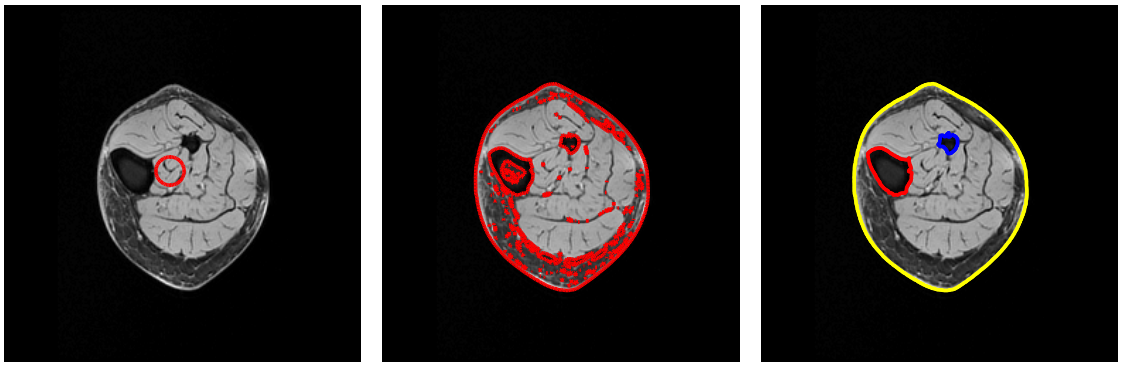


(a) Manual segmentation (b) Semi-automatic segmentation (c) Comparison

Figure 4.7: Semi-automatic and manual segmentation of medial gastrocnemius muscle, dimensions in mm

4.1.2 Full model construction of human lower leg muscles from images

The following methodology is developed for constructing the full model of the human lower leg muscles from medical images. The geometry is extracted through segmentation of the anatomical MR image. Muscle, Intra-Muscular Connective Tissue (IMCT), fat (including both intra-muscular fat (IMAT) and subcutaneous fat) are the materials included in the construction of the model. The Chan-Vese level set segmentation method [7] is used for extracting boundaries of the bones and the outer boundary of the lower leg from each MR image as shown in figure 4.8. The interior points of boundary contours are obtained as shown in figure 4.9. The IMCT and IMAT are segmented independently from a separate specialized sequences of MR images using the fuzzy C-means algorithm. The segmented pixels overlapped on the MR image are shown in figure 4.10. The segmented IMAT and IMCT points are subtracted from the interior points to obtain an image (say image1). From the DT images, the points which have EV data (at least one of the eigen values should be non-zero) are chosen as the muscle points in image1



(a) Initial contour

(b) Chan Vese segmentation result

(c) Outer contours and bone contours obtained from segmentation

Figure 4.8: Chan Vese segmentation of anatomical MR image

and all other interior points are assigned as fat since they constitute the outer subcutaneous fat layer. There might be a small number of additional points in the interior of the muscle regions with no fiber direction specified. These are labeled as intra-muscular fat. This final image obtained which has all the different material points labeled using different images, forms the fine scale model as shown in figure 4.11(a). The points of each material are tested using an algorithm to check that there are no common points between any two materials. The units of x, y and z coordinates are in pixel coordinates where 1 pixel = 0.078125 cm. A coarse scale model was generated for computational efficiency, by creating a coarse set of points with larger spacing between the pixels. The nodal spacing in the fine model is 1 pixel, whereas in the coarse model it is taken as 2 pixels. The muscle, fat and IMCT of the coarse image can be obtained by sorting common points between the coarse and the fine images for each slice. The coarser image generated is shown in figure 4.11(b). This procedure is repeated for every image (slice) and the 3D pixel based model is constructed by stacking the 2D slices. Figure 4.11(c) shows the 3D coarse model with an in-plane pixel spacing of 2, constructed by stacking 25 segmented images of the human lower leg. The model consists of $73,659$ nodes.

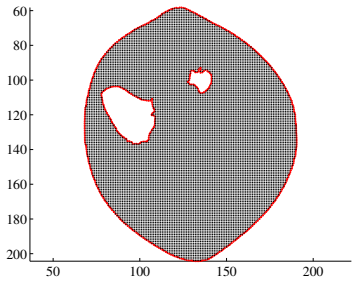


Figure 4.9: Interior points obtained from segmentation

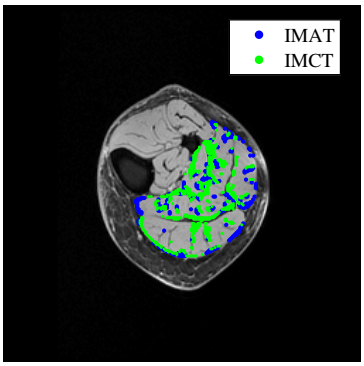
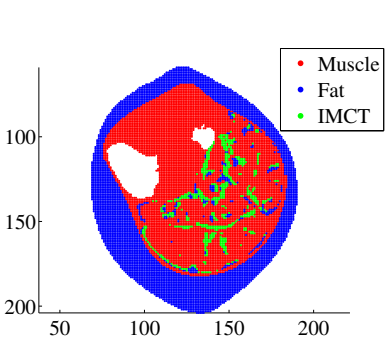
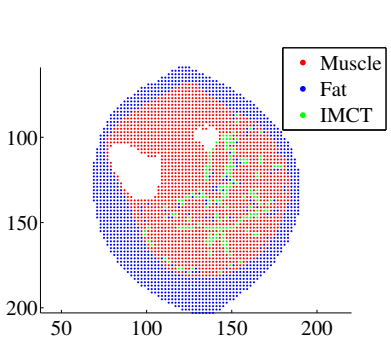


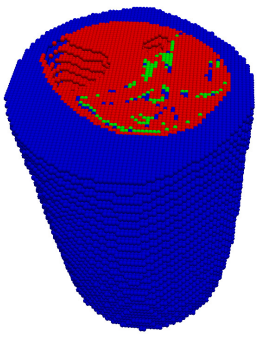
Figure 4.10: Segmented IMAT and IMCT pixels



(a) Fine model



(b) Coarse model



(c) 3D model constructed from 25 images of the coarse model, has 73,659 nodes

Figure 4.11: Final segmentation results with different materials

4.1.3 3D surface normal approximation from stacked 2D contours

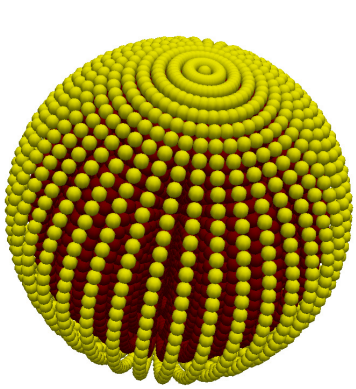
As illustrated in the previous 2 sections, the 3D model geometries are obtained by stacking the 2D segmented images to form the 3D models. In numerical simulation, the surface normals on the natural boundaries are required in order to impose prescribed tractions on these surfaces. The 2D level set functions obtained for each of the medical images can only give the 2D normals to the contour obtained in that image plane. When the 3D model is constructed by stacking the 2D contours, the 3D surface normals cannot be directly obtained. In this section the following method is used for approximating the 3D surface normals from the 2D level set contours. This is illustrated through an example. Consider a sphere model, for which the points on the boundary, interior and exterior are known a priori (say for example through 2D segmentation of in-plane slices of the sphere). The cross section of the sphere which shows the point distribution is shown in figure 4.12(b), where yellow, blue and red points denote the boundary, interior and exterior points, respectively. The 3D level set function $\phi(\boldsymbol{x})$ is re-constructed such that:

$$\phi(\boldsymbol{x}) > 0 \quad \text{if } \boldsymbol{x} \text{ is inside } \Gamma \tag{4.8a}$$

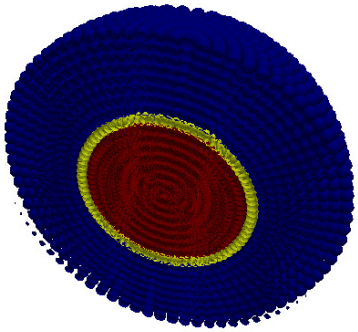
$$\phi(\boldsymbol{x}) = 0 \quad \text{if } \boldsymbol{x} \text{ is on } \Gamma \tag{4.8b}$$

$$\phi(\boldsymbol{x}) < 0 \quad \text{if } \boldsymbol{x} \text{ is outside } \Gamma \tag{4.8c}$$

where Γ represents the boundary of the model. The normal to the boundary surface at a point is constructed using this 3D level set function and is given in equation (3.2). Here $\phi(\boldsymbol{x})$ is approximated using the reciprocal Multiquadric Radial Basis (RB) functions as given in [78]. The approximated normals obtained on the boundary of the sphere are shown in figure 4.13.



(a) Model of the sphere for which surface normals need to be obtained



(b) Cross-section of the sphere with exterior points added

Figure 4.12: Sphere model (without exterior points) and its cross-section (with exterior points added) showing the points on the boundary (yellow color), inside the boundary (red color) and outside the boundary (blue color)

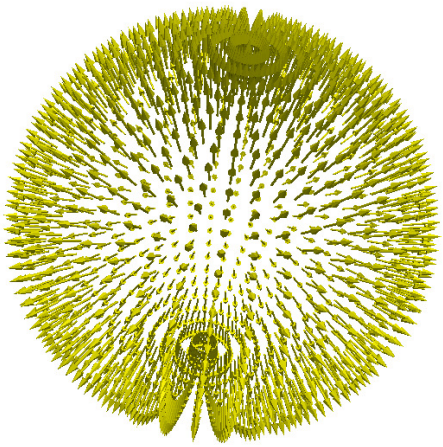
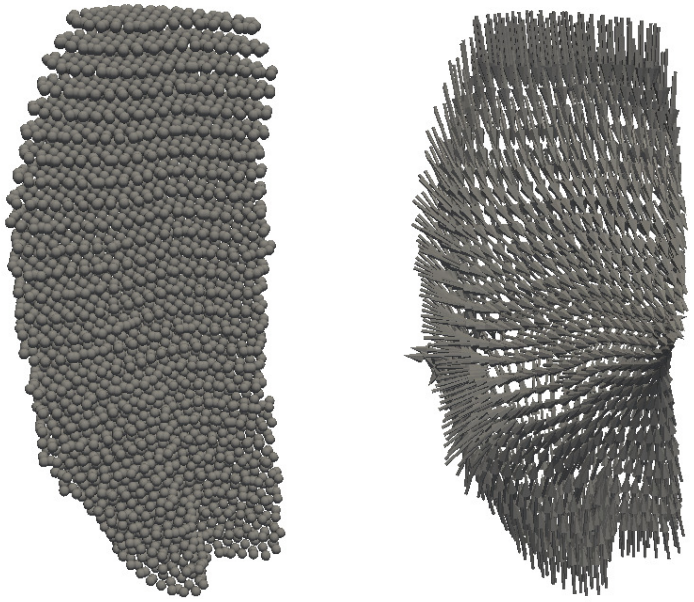


Figure 4.13: 3D normal approximation for the sphere model plotted at the boundary points



(a) Medial gastrocnemius muscle model

(b) Approximated normals obtained for the boundary points

Figure 4.14: 3D normal approximation for the medial gastrocnemius muscle model

The above method is implemented to find the 3D surface normals for a 3D medial gastrocnemius muscle model shown in figure 4.14(a). The approximated 3D normals obtained are given in figure 4.14(b).

4.2 Pixel based meshfree modeling of skeletal muscles

In the conventional Finite Element (FE) approach, the meshes need to be conformed to muscle geometry, which increases the complexity of mesh construction. Generally muscles have a complex architecture and poorly built meshes can easily lead to significant errors in FE analysis due to mesh distortion. Abrupt changes of topology in the muscle cross sections could also result in failure in FE mesh generation. Additionally, muscle material is anisotropic in nature due to the presence of muscle fibers. In FE modeling, one way to introduce the anisotropy is to approximate the fiber directions by

interpolating from some pre-defined templates as described in [93]. The fiber directions obtained in this way could generate noticeable discretization errors in the simulation models. To overcome these drawbacks, in this section the meshfree Reproducing Kernel Particle Method (RKPM) is used for numerical modeling of heterogeneous skeletal muscles. The fiber direction obtained from DT imaging data [33, 36] and the material properties are defined at pixel points and are directly used as input into meshfree modeling, without the need for additional data processing. Since no mesh is required in meshfree methods, the complexity related to meshing in finite element method is avoided. The Reproducing Kernel (RK) approximation also allows for a smooth transition of material properties at the interfaces between different materials in the muscle, which is required in the context of biological materials where the material interfaces do not exhibit sharp discontinuities. The skeletal muscle is represented as a nearly incompressible hyperelastic material [89]. RKPM has been used to simulate extremely large deformation of rubber like hyperelastic material [54, 55]. The unique properties of RKPM, such as arbitrary order continuity (smoothness) in the approximation and the straightforward h-adaptivity, can also be applied effectively to the modeling of biomaterials.

4.2.1 Reproducing Kernel (RK) approximation

In meshfree modeling, the problem domain is discretized with a set of arbitrarily distributed points (nodes) as shown in figure 4.15. Each point I is associated with an open cover ω_I which defines the locality of the approximation defined on ω_I. In the present modeling, the pixel coordinates of the geometry are obtained from the images and are used as nodes for discretizing the domain for the meshfree modeling. The RK shape functions are constructed based on a set of points and are used to approximate the displacement field governed by the equilibrium equation or equation of motion of a

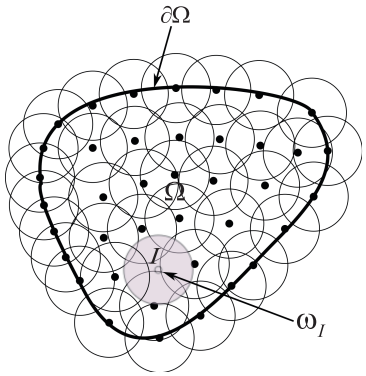


Figure 4.15: Meshfree domain discretization

solid [53, 54].

Let $\alpha = \alpha_1\alpha_2\cdots\alpha_d$ represent the multi-dimensional index notation where 'd' is the number of spatial dimensions. The following notation is used:

$$\boldsymbol{p}_\alpha = \boldsymbol{p}_{\alpha_1\alpha_2\cdots\alpha_d} \tag{4.9a}$$

$$|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_d \tag{4.9b}$$

$$\boldsymbol{x}^\alpha = \boldsymbol{x}_1^{\alpha_1}\boldsymbol{x}_2^{\alpha_2}\cdots\boldsymbol{x}_d^{\alpha_d} \tag{4.9c}$$

$$(\boldsymbol{x} - \boldsymbol{x}_I)^\alpha = (\boldsymbol{x}_1 - \boldsymbol{x}_{1I})^{\alpha_1}(\boldsymbol{x}_2 - \boldsymbol{x}_{2I})^{\alpha_2}\cdots(\boldsymbol{x}_d - \boldsymbol{x}_{dI})^{\alpha_d} \tag{4.9d}$$

Consider a domain Ω in 'd' dimensional space $\mathbb{R}^d$ which is discretized by a set of nodes '*NP*' given by $\{\boldsymbol{x}_I | \boldsymbol{x}_I \in \Omega\}_{I=1}^{NP}$. The Reproducing Kernel (RK) approximation of a function 'u' denoted as 'u^h' is given by:

$$u^h(\boldsymbol{x}) = \sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) d_I \tag{4.10}$$

where $\Psi_I(\boldsymbol{x})$ is the RK shape function at node 'I' and d_I are the nodal coordinates at node 'I'. The RK shape function is expressed as:

$$\Psi_I(\boldsymbol{x}) = C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)\phi_a(\boldsymbol{x} - \boldsymbol{x}_I) \tag{4.11}$$

where $\phi_a(\boldsymbol{x}-\boldsymbol{x}_I)$ is a kernel function that provides the required smoothness in the RK approximation and determines the locality of the approximation by its compact support measured by 'a'. The multi-dimensional kernel function can be constructed as follows using either a circular or rectangular support.

Kernel with a circular support:

$$\phi_a(\boldsymbol{x}-\boldsymbol{x}_I)=\phi_a\left(z\right) \tag{4.12}$$

where $z=\frac{\|\boldsymbol{x}-\boldsymbol{x}_I\|}{a}$.

Kernel with a rectangular support: Obtained using a product of one dimensional kernels.

$$\phi_a(\boldsymbol{x}-\boldsymbol{x}_I)=\prod_{i=1}^{d}\phi_a\left(\frac{\boldsymbol{x}_i-\boldsymbol{x}_{iI}}{a_i}\right) \tag{4.13}$$

where 'd' is the number of spatial dimensions and 'a_i' is the support size in the i^{th} direction. The term $C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)$ is the correction function which is constructed using a set of basis functions and satisfies the consistency requirements of the shape function. It is expressed as:

$$C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)=\sum_{|\alpha|=0}^{n}(\boldsymbol{x}-\boldsymbol{x}_I)^{\alpha}\boldsymbol{b}_{\alpha}(\boldsymbol{x})=\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}_I)\boldsymbol{b}(\boldsymbol{x})\qquad\text{in }\mathbb{R}^d \tag{4.14}$$

where 'n' is the order of the basis functions, $\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}_I)=\{(\boldsymbol{x}-\boldsymbol{x}_I)^{\alpha}\}_{|\alpha|=0}^{n}$ is a vector containing all the monomial basis functions, and $\boldsymbol{b}_{\alpha}(\boldsymbol{x})$ is an unknown vector which is determined by enforcing the following n^{th} order reproducing conditions (n^{th} order completeness requirement):

$$\sum_{I=1}^{NP}\Psi_I(\boldsymbol{x})\boldsymbol{x}_I^{\alpha}=\boldsymbol{x}^{\alpha};\quad|\alpha|=0\text{ to }n \tag{4.15}$$

This is equivalent to:

$$\sum_{I=1}^{NP} \Psi_I(\boldsymbol{x})(\boldsymbol{x}-\boldsymbol{x}_I)^{\alpha} = \delta_{0|\alpha|}; \quad |\alpha| = 0 \text{ to } n \tag{4.16}$$

where δ is the Kronecker delta. Expressing equation (4.16) in matrix notation:

$$\sum_{I=1}^{NP} \Psi_I(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{0}) \tag{4.17}$$

From equation (4.11) RK shape function can be re-written as:

$$\Psi_I(\boldsymbol{x}) = \boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}_I)\boldsymbol{b}(\boldsymbol{x})\phi_a(\boldsymbol{x}-\boldsymbol{x}_I) \tag{4.18}$$

Substituting the expression for the shape function from equation (4.18) in equation (4.17) gives the expression for the vector $\boldsymbol{b}(\boldsymbol{x})$.

$$\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{0}) \tag{4.19}$$

where $\boldsymbol{M}$ is called the moment matrix and is given by:

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I=1}^{NP} \boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}_I)\phi_a(\boldsymbol{x}-\boldsymbol{x}_I) \tag{4.20}$$

Using the expression for the RK shape function in equation (4.18) and substituting the $\boldsymbol{b}$ vector from equation (4.19) the RK shape function can be written as:

$$\Psi_I(\boldsymbol{x}) = \boldsymbol{H}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)\phi_a(\boldsymbol{x}-\boldsymbol{x}_I) \tag{4.21}$$

Example of the two dimensional RK shape function, constructed using the quintic B-spline kernel function, is shown in figure 4.16.

For the moment matrix to be invertible at any point '$\boldsymbol{x}$' in the domain, '$\boldsymbol{x}$' has

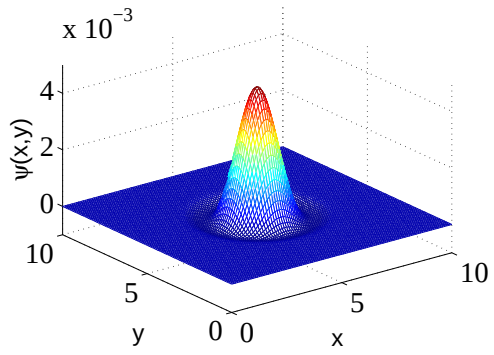


Figure 4.16: 2D Reproducing Kernel shape function

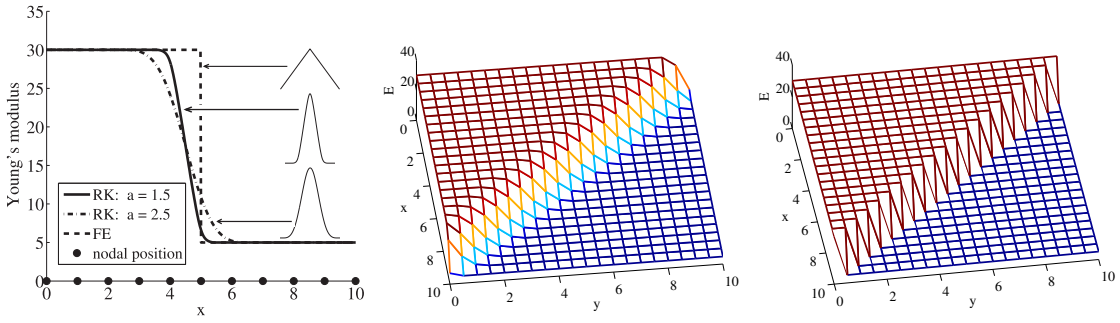
to be covered by at least 'k' non-coplanar shape functions where 'k' is the number of terms in the basis vector. The RK approximation satisfies the reproducing conditions everywhere in the domain and on the boundary in the discrete form [119]. The order of completeness of the RK approximation is determined by the order of consistency, that is, the order of basis functions 'n'. The RK approximation functions do not possess Kronecker Delta property, that is, $\Psi_I(\boldsymbol{x}_J) \neq \delta_{IJ}$. Due to this condition, numerous methods have been proposed to impose the essential boundary conditions, including the Lagrange multiplier method [52, 55], transformation method [75, 54], Nitsches method and penalty method [74]. In the present formulation, the penalty method is used for imposing the essential boundary conditions.

As can be seen in the full lower leg model given in figure 4.11(c), the pixel resolution in the longitudinal z-direction is lower than the resolution in the transverse xy-direction. This is due to the fact that in the MR image acquisition, the scanning in the vertical direction is only taken at certain intervals. In the meshfree modeling, this is taken into account by using a suitable support size in the RK approximation function, which is adjusted according to the nodal spacing in different directions in the model. For this modeling, RK shape function with rectangular support is used, with the length in the longitudinal direction about 3 times of that in the transverse direction.

In the following examples for muscle simulation in this chapter, the reproducing kernel approximation functions are constructed using a cubic spline kernel function and linear basis. A normalized support size of $a/h = 1.65$ is used, where a is the support size of the kernel function and h is the nodal (pixel) spacing.

4.2.2 Muscle material properties interpolation using RK approximation

It has been shown in many studies [120] that the change of material properties from skeletal muscle to tendon is a smooth transition. Heterogeneous material modeling using FE representation results in jumps in material properties at the material interfaces, and the computed stresses and strains are also discontinuous across the element boundaries. On the contrary, in the proposed meshfree RKPM analysis the material properties are assigned to the nodes and the material properties transition represented by the smooth RK approximation can be made with the desired smooth transition, which avoids the abrupt jumps of stresses and strains. This RK representation of the smooth transition in material properties is illustrated in the following example. Consider a region $x \in [0, 10]$ with the interface at $x = 5$ between two materials, where the Young's moduli (E) are given by: $E = 30$ for $x \in [0, 5)$ and $E = 5$ for $x \in [5, 10]$. As can be seen in figure 4.17 finite element approximation exhibits a sharp discontinuity at $x = 5$ while the RK approximation shows smooth transition in representing the material property transition across the material interface by choosing appropriate kernel function support size 'a' in the RK approximation.



(a) 1D comparison (b) 2D RK approximation (c) 2D FE interpolation

Figure 4.17: Comparison of RK and FE approximations of the Young's modulus

4.2.3 Meshfree RKPM formulation for hyperelasticity

In this section, a 3D RKPM formulation is used to solve the hyperelastic problem, in which the penalty method is used to impose the essential boundary conditions. The formulation is based on the total Lagrangian framework, where the implementation is carried out in the reference or undeformed configuration. Let the problem domain in the undeformed configuration be denoted by Ω_X, with the corresponding essential boundary and natural boundary denoted by $\partial\Omega_X^g$ and $\partial\Omega_X^h$, respectively. The energy functional for this total Lagrangian formulation can be written as follows:

$$U = \int_{\Omega_X} W(\boldsymbol{u}) d\Omega - \int_{\Omega_X} u_i b_i d\Omega - \int_{\partial\Omega_X^h} u_i h_i d\Gamma + \frac{\beta}{2} \int_{\partial\Omega_X^g} (u_i - g_i)(u_i - g_i) d\Gamma \quad (4.22)$$

where $W(\boldsymbol{u})$ is the strain energy density function, $\boldsymbol{b}$ is the body force per unit undeformed volume, $\boldsymbol{h}$ is the prescribed surface force per unit undeformed area on the natural boundary $\partial\Omega_X^h$, β is the penalty number and $\boldsymbol{g}$ is the prescribed displacement on the essential boundary $\partial\Omega_X^g$. The stationary condition is obtained by taking variation of

equation (4.22), which yields:

$$\begin{aligned}&\int_{\Omega_X} \delta F_{ij} P_{ji} d\Omega + \beta \int_{\partial\Omega_X^g} \delta u_i u_i d\Gamma \\ &= \int_{\Omega_X} \delta u_i b_i d\Omega + \int_{\partial\Omega_X^h} \delta u_i h_i d\Gamma + \beta \int_{\partial\Omega_X^g} \delta u_i g_i d\Gamma \end{aligned} \tag{4.23}$$

where $\boldsymbol{F}$ is the deformation gradient and $\boldsymbol{P}$ is the first Piola-Kirchhoff stress. Due to the geometric and material nonlinearities, Newton's method is used to solve the nonlinear equations and the linearization of equation (4.23) is required. Let n and ν denote the current load step counter and iteration step counter, respectively, the linearized equation is given by:

$$\begin{aligned}&\int_{\Omega_X} \delta F_{ij} (C_{ijkl})_{n+1}^{\nu} \Delta F_{kl} d\Omega + \beta \int_{\partial\Omega_X^g} \delta u_i \Delta u_i d\Gamma \\ &= \int_{\Omega_X} \delta u_i (b_i)_{n+1} d\Omega + \int_{\partial\Omega_X^h} \delta u_i (h_i)_{n+1} d\Gamma \\ &+ \beta \int_{\partial\Omega_X^g} \delta u_i [(g_i)_{n+1} - (u_i)_{n+1}^{\nu}] d\Gamma - \int_{\Omega_X} \delta F_{ij} (P_{ji})_{n+1}^{\nu} d\Omega \end{aligned} \tag{4.24}$$

where C_{ijkl} is the first elasticity tensor. The displacement vector, its variation and the incremental displacement vector are approximated by the RK shape functions as follows:

$$\boldsymbol{u} = \sum_{I=1}^{NP} \boldsymbol{N}_I \boldsymbol{d}_I, \quad \delta\boldsymbol{u} = \sum_{I=1}^{NP} \boldsymbol{N}_I \delta\boldsymbol{d}_I, \quad \Delta\boldsymbol{u} = \sum_{I=1}^{NP} \boldsymbol{N}_I \Delta\boldsymbol{d}_I \tag{4.25}$$

where

$$\boldsymbol{u} = [u_1 \;\; u_2 \;\; u_3], \quad \boldsymbol{d}_I^T = [d_{1I} \;\; d_{2I} \;\; d_{3I}], \quad \boldsymbol{N}_I = \begin{bmatrix} \Psi_I & 0 & 0 \\ 0 & \Psi_I & 0 \\ 0 & 0 & \Psi_I \end{bmatrix} \tag{4.26}$$

Here Ψ_I is the Lagrangian RK shape function defined as:

$$\Psi_I(\boldsymbol{X}) = C(\boldsymbol{X}; \boldsymbol{X} - \boldsymbol{X}_I)\phi_a^X(\boldsymbol{X} - \boldsymbol{X}_I) \tag{4.27}$$

and the kernel ϕ_a^X is called the 'material kernel' given by:

$$\phi_a^X(\boldsymbol{X} - \boldsymbol{X}_I) = \phi_a^X\left(\frac{\|\boldsymbol{X} - \boldsymbol{X}_I\|}{a}\right) \tag{4.28}$$

As shown in the above equation, the material kernel is defined using the distance measured by the material coordinate in the reference configuration. The support of the material kernel function covers the same set of nodes throughout the history of deformation and hence the associated shape function is called the 'Lagrangian' shape function.

The incremental deformation gradient in vector form is given by:

$$\Delta\boldsymbol{F} = \sum_I \boldsymbol{B}_I \Delta\boldsymbol{d}_I \tag{4.29}$$

where

$$\begin{aligned}
\boldsymbol{F}_I^T &= [F_{11}\ \ F_{22}\ \ F_{33}\ \ F_{12}\ \ F_{21}\ \ F_{13}\ \ F_{31}\ \ F_{23}\ \ F_{32}] \\
\boldsymbol{B}_I^T &= \begin{bmatrix}
\Psi_{I,1} & 0 & 0 & \Psi_{I,2} & 0 & \Psi_{I,3} & 0 & 0 & 0 \\
0 & \Psi_{I,2} & 0 & 0 & \Psi_{I,1} & 0 & 0 & \Psi_{I,3} & 0 \\
0 & 0 & \Psi_{I,3} & 0 & 0 & 0 & \Psi_{I,1} & 0 & \Psi_{I,2}
\end{bmatrix}
\end{aligned} \tag{4.30}$$

Here $\Psi_{I,j} = \partial\Psi_I/\partial X_j$. Taking the material derivative of the Lagrangian RK shape function is straightforward and is detailed in section **??**. By introducing the RK approximation for displacements given in equation (4.25) and the deformation gradient given in equation (4.29) into the linearized equation (4.24) the following matrix equation is

obtained:

$$\boldsymbol{K}_{IJ}\Delta\boldsymbol{d}_J = \boldsymbol{f}_I^{\text{ext}} - \boldsymbol{f}_I^{\text{int}} \tag{4.31}$$

where $I, J = 1, \cdots, NP$. The dimension of stiffness $\boldsymbol{K}_{IJ}$ is 3×3 and that of $\Delta\boldsymbol{d}_J$ is 3×1 and

$$\begin{aligned}
\boldsymbol{K}_{IJ} &= \int_{\Omega_X} \boldsymbol{B}_I^T \tilde{\boldsymbol{C}} \boldsymbol{B}_J d\Omega + \beta \int_{\partial\Omega_X^g} \boldsymbol{N}_I^T \boldsymbol{N}_J d\Gamma \\
\boldsymbol{f}_I^{\text{int}} &= \int_{\Omega_X} \boldsymbol{B}_I^T \boldsymbol{P} d\Omega \\
\boldsymbol{f}_I^{\text{ext}} &= \int_{\Omega_X} \boldsymbol{N}_I^T \boldsymbol{b} d\Omega + \int_{\partial\Omega_X^h} \boldsymbol{N}_I^T \boldsymbol{h} d\Gamma + \beta \int_{\partial\Omega_X^g} \boldsymbol{N}_I^T (\boldsymbol{g} - \boldsymbol{u}_{n+1}^{\nu}) d\Gamma
\end{aligned} \tag{4.32}$$

and

$$\begin{aligned}
\boldsymbol{P} &= [P_{11} \;\; P_{22} \;\; P_{33} \;\; P_{12} \;\; P_{21} \;\; P_{13} \;\; P_{31} \;\; P_{23} \;\; P_{32}]^T \\
\boldsymbol{b} &= [b_1 \;\; b_2 \;\; b_3]^T \\
\boldsymbol{h} &= [h_1 \;\; h_2 \;\; h_3]^T
\end{aligned} \tag{4.33}$$

where $\tilde{\boldsymbol{C}}$ is a 9×9 matrix and each element is given by $\tilde{\boldsymbol{C}}_{AB} = \partial^2 W / \partial \boldsymbol{F}_A \partial \boldsymbol{F}_B$.

The method of Stabilized Conforming Nodal Integration (SCNI) [59, 68, 71] is used for integrating the discrete equations in (4.31) where a smoothed derivative of the approximation function is used in conjunction with a nodal integration in equation (4.30) to achieve computational efficiency and accuracy. SCNI ensures a quadratic rate of convergence in Galerkin meshfree approximation of the equilibrium equation, and it achieves greater computational efficiency and accuracy for the meshfree method compared to conventional Gauss quadrature rules. In the proposed image based modeling, using SCNI in RKPM muscle modeling, the pixel points which constitute the nodes in the domain are also used as integration points for integrating the variational equation

(weak form). RKPM with SCNI has been shown to be accurate for nearly incompressible problems [68] and hence is suitable for modeling skeletal muscles.

4.2.4 Material laws for components of skeletal muscle

A transversely isotropic hyperelastic model is employed to represent the mechanical behavior of skeletal muscle. The strain energy density function for the muscle is decoupled into isotropic and anisotropic parts as defined below:

$$W_{muscle} = W_{matrix}(\bar{I}_1, \bar{I}_2, J) + W_{fiber}(\hat{\lambda}) \tag{4.34}$$

where W_{matrix} is the strain energy stored within the isotropic muscle matrix, W_{fiber} is the strain energy stored within muscle fiber and this introduces anisotropy to the model. $\bar{I}_1 = I_1 I_3^{-1/3}$, $\bar{I}_2 = I_1 I_3^{-2/3}$ where I_1, I_2 and I_3 are the three invariants of right Cauchy-Green strain tensor $\boldsymbol{C}$, $\hat{\lambda} = \sqrt{\hat{\boldsymbol{N}} \cdot \bar{\boldsymbol{C}} \cdot \hat{\boldsymbol{N}}}$ is the stretch ratio along the fiber direction $\hat{\boldsymbol{N}}$ and $\bar{\boldsymbol{C}} = J^{-2/3}\boldsymbol{C}$, where $J = det(\boldsymbol{F})$. A quadratic polynomial type strain energy density function for W_{matrix} as employed in [89] is used, which is given below:

$$W_{matrix} = \sum_{i+j=1}^{2} A_{ij}(\bar{I}_1 - 3)^i(\bar{I}_2 - 3)^j + \frac{K}{2}(J-1)^2 \tag{4.35}$$

Here, A_{ij} and K are the material constants and bulk modulus of muscle respectively. The fiber stress is related to the energy density function W_{fiber} as follows:

$$\hat{\lambda}\frac{\partial W_{fiber}}{\partial \hat{\lambda}} = \sigma_{max}\frac{\hat{\lambda}}{\hat{\lambda}_o}(\alpha f_{active} + f_{passive}) \tag{4.36}$$

where $\hat{\lambda}_o$ is the stretch ratio at optimal length, σ_{max} is the maximum isometric stress, α is the activation factor which represents the level of activation in the muscle fiber, f_{active}

Table 4.1: Material constants for muscle matrix (in N/cm^2)

A_{10}	A_{01}	A_{20}	A_{02}	A_{11}	K
6.43	-3.8	0.94	0.0005	-0.0043	5000

and $f_{passive}$ are the normalized active and passive fiber forces, respectively, given by:

$$\begin{cases} f_{passive} = 0 & \text{for } \lambda^* \leq 1 \\ f_{passive} = \gamma_1(e^{\gamma_2(\lambda^*-1)} - 1) & \text{for } 1 < \lambda^* \leq 1.4 \\ f_{passive} = (\gamma_1\gamma_2 e^{0.4\gamma_2})\lambda^* + \gamma_1(e^{0.4\gamma_2} - 1) & \text{for } \lambda^* > 1.4 \end{cases} \tag{4.37}$$

$$\begin{cases} f_{active} = 9(\lambda^* - 0.4)^2 & \text{for } \lambda^* \leq 0.6 \\ f_{active} = 1 - 4(1 - \lambda^*)^2 & \text{for } 0.6 < \lambda^* \leq 1.4 \\ f_{active} = 9(\lambda^* - 1.6)^2 & \text{for } \lambda^* > 1.4 \end{cases} \tag{4.38}$$

where $\lambda^* = \hat{\lambda}/\hat{\lambda}_o$ is the normalized stretch ratio, and $\gamma_1 = 0.05$, $\gamma_2 = 6.6$.

Connective tissue and fat are modeled by an isotropic cubic hyperelastic model with the strain energy density function defined as [89]:

$$W = A_{10}(\bar{I}_1 - 3) + A_{20}(\bar{I}_1 - 3)^2 A_{30}(\bar{I}_1 - 3)^3 + \frac{K}{2}(J - 1)^2 \tag{4.39}$$

The calibrated values of material constants for muscle and connective tissue are adopted from [89]. The material constants for the fat are 10 times softer than those of the connective tissue. The material constants used in this work are summarized in table 4.1 for muscle and table 4.2 for connective tissue and fat.

Table 4.2: Material constants for connective tissue and fat (in N/cm^2)

	A_{10}	A_{20}	A_{30}	K
Connective tissue	30	80	800	50,000
Fat	3	8	80	5000

4.2.5 Numerical analysis of isometric contraction of the lower leg using RKPM

In this example, isometric contraction of the 3D muscle model of the lower leg shown in figure 4.11(c), which is constructed from slices of the coarse image (figure 4.11(b)), is simulated using the RKPM formulation. Isometric contraction is achieved by fixing the top and bottom boundaries. For the points near the interior bone boundaries, the displacements in the cross-sectional plane are fixed, but are free to move along the longitudinal direction of the muscle. The activation factor α of the muscle material is linearly increased from 0 to 0.95. In the simulation, all muscle points are activated simultaneously. The generated force due to muscle contraction is calculated at the fixed end. The reaction force generated at different levels of muscle activation is shown in figure 4.18. A convergence study is performed to ensure that the discretization resolution of muscle model (figure 4.11(c)) is sufficient for desired accuracy. A refined RKPM model is generated from finer cross-sectional images (shown in figure 4.11(a)). The spacing in the longitudinal direction (vertical z-direction) is not refined due to the limitation in the image acquisition. This is also justified due to the fact that the spatial variations in the stress and strain fields in the longitudinal direction are much smaller than those in the transverse direction. The force generation results obtained from the coarse and refined models are shown in figure 4.18, which confirm convergence of the RKPM solution. In figure 4.19 the nodal displacement vectors are plotted on sample

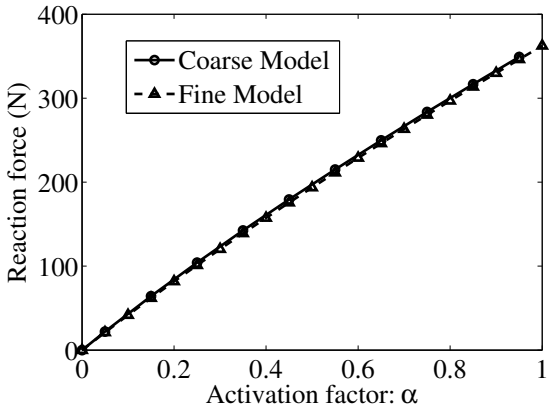


Figure 4.18: Reaction force generated at different levels of muscle activation predicted by the coarse and refined models

planes (transverse, coronal, sagittal) on the reference (undeformed) configuration. The muscle fiber directions in the undeformed configuration are also plotted for comparison. The results show that while in certain areas the nodal displacement vectors are approximately in the direction of the muscle fibers, no particular correlation between them is observed. As can be seen in the sagittal cross-section in figure 4.19(e), the points near the posterior region displace downwards towards the distal end and the points near the anterior region move upwards toward the proximal end due to muscle contraction. Figure 4.20 shows the maximum principal Cauchy stress in the full model and on sample planes at the final configuration. It is important to note that the stress fields shown in figure 4.20 are calculated using the RK approximation functions which are continuous in its first order derivatives. Thus, the computed stress field is continuous, in contrast to finite element method which yields discontinuous stress field across element boundaries.

4.2.6 Comparison of force production in young and old muscles

Two Medial Gastrocnemius (MG) muscle models from young and old subjects were selected to investigate the effect of aging on force generation. Sample MR image slices of these young and old subjects are given in figure 4.21(a) and 4.21(b), respec-

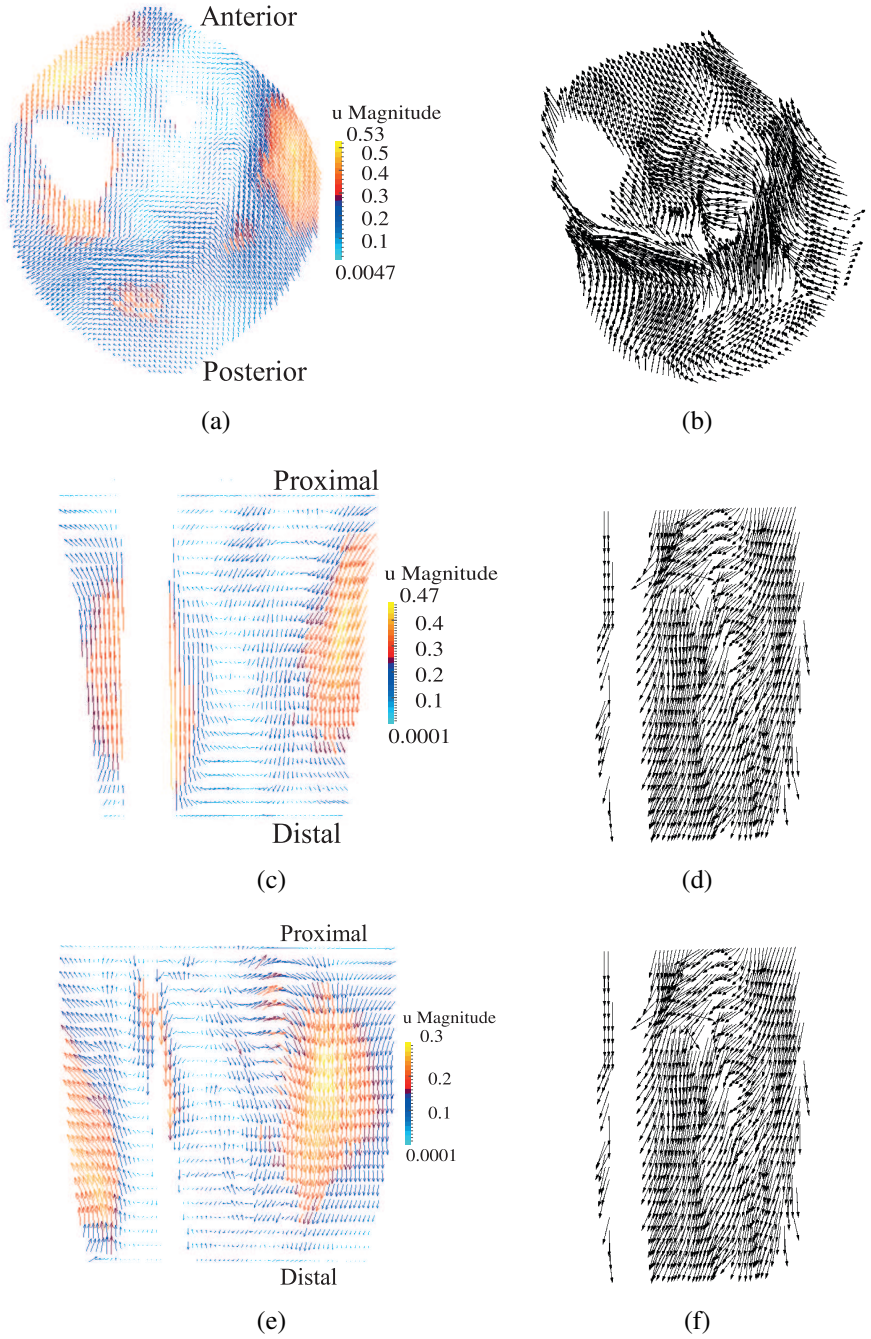


Figure 4.19: Displacement vectors (left) and corresponding fiber directions (right) plotted on reference configuration on (a)-(b) transverse plane at $z = 5\ cm$; (c)-(d) coronal plane at $y = 10.1\ cm$; (e)-(f) sagittal plane at $x = 10\ cm$. (units of u magnitude: cm). Note: The fiber directions are plotted only at the muscle points, whereas displacements are shown at all material points.

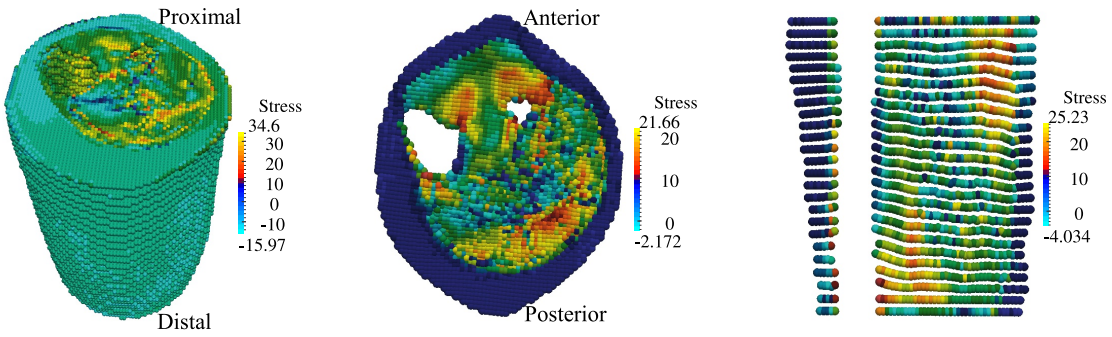
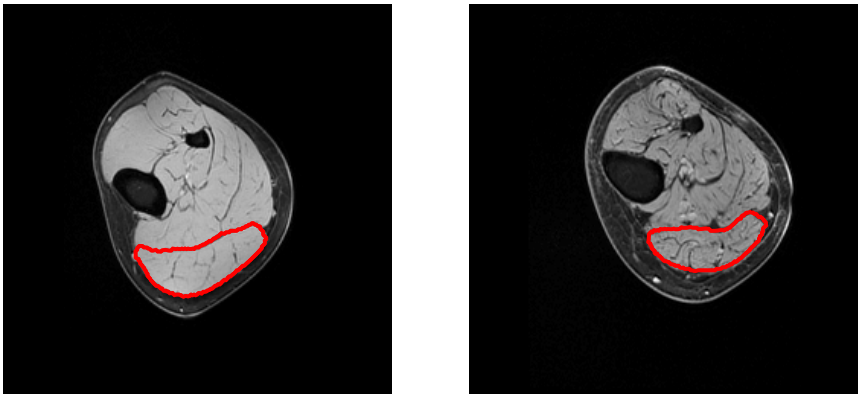


Figure 4.20: Distribution of the maximum principal Cauchy stress (N/cm^2) at the final deformed configuration for full model (left), at transverse plane $z = 5.0\ cm$ (center), and at coronal plane $y = 10.1\ cm$ (right)

tively, which clearly show the increased amount of connective tissue in the older subject. The MG muscle is outlined in red in these figures. The segmented MG models of the young and old subjects with the same axial length are shown in figure 4.22(a) and 4.22(b), respectively. Here green points are the connective tissue, red points are muscle and blue points are fat (IMAT). The MG is surrounded by an outer layer of connective tissue which forms the aponeurosis. The volumes of the different material components in the young and the old models are given in table 4.3. It should be noted that IMCT here does not include the aponeurosis.

In the numerical simulation, both ends of the muscle are fixed and the activation of the muscle is increased to simulate isometric contraction. The muscle force activation characteristics for both the young and the old muscles are taken to be the same. The force generation at maximum activation shows that the younger model generates much bigger force than the older model (figure 4.22(c)). The effective force generated per unit total volume (including the outer layer of connective tissue) for the younger model is 0.430 N/cm^3 and for the old model is 0.294 N/cm^3. This shows that the amount of non-contractile tissue strongly affects the force production. It is worth noting that the muscle volume of the older model is about 37% smaller than that of the younger model but the force generation in the older model drops by around 45% compared to the young model.



(a) Young subject (b) Old subject

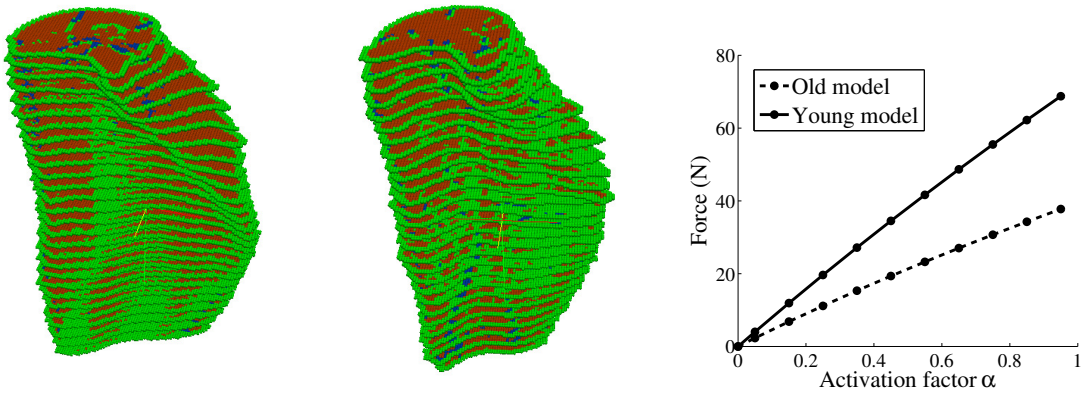
Figure 4.21: Connective tissue visualization from MR image of (a) young and (b) old models. The MG muscle is outlined in red.

Table 4.3: Young versus old models results comparison

	Young model	Old model	Old normalized by young
Muscle (cm^3)	98.93	61.89	62.5%
IMAT (cm^3)	2.83	6.15	217.3%
IMCT (cm^3)	20.86	29.15	139.7%
F_{max} (N)	68.76	37.74	54.8%

There is a disproportionate decrease in force production due to the decrease in muscle volume. These results from numerical modeling support the hypothesis that the amount of connective tissue in the muscle affects the force production considerably [121].

In this chapter, an image based meshfree RKPM computational framework is developed for skeletal muscle modeling. A method for RKPM simulation model construction with different material components is developed by extracting pixel data from medical images. The pixel points from these images can be directly used as nodes for domain discretization in the meshfree modeling, and the fiber direction obtained from the DT imaging data is input directly at each pixel point, without the need for addi-



(a) Young subject (b) Old subject (c) Reaction force generated

Figure 4.22: MG muscles of young and old models. Comparison of force generation versus activation level between young and old models

tional processing or interpolation. The employed reproducing kernel approximation can represent the smooth transition of material properties in heterogeneous materials. The variation of material properties in the transition zone can be controlled by adjusting the support size of the kernel function in the RK approximation. These properties of image based meshfree methods render it suitable for subject specific modeling.

A multiphase multichannel level set approach for segmenting individual muscles using both MR and DT imaging data is also proposed. Using this method, adjacent muscles in the image which have different fiber orientations can be segmented. The multichannel method enables the incorporation of MR image and EV image data in different channels for segmentation, and a multiphase framework is required to segment connected regions in an image. Additionally a method is proposed for approximating the 3D surface normals for the 3D muscle models which are constructed by stacking 2D segmented images.

Numerical examples are given to demonstrate the effectiveness of the proposed image based meshfree method for modeling the mechanical behavior of skeletal muscles. A preliminary study of force production between young and old skeletal muscles

having different amounts of non-contractile tissue volumes is performed. A disproportionate decrease in force production is observed due to the decrease in muscle volume between the young and the old MG muscles.

This chapter in part has been published in the following book chapter and journal paper respectively: **Book Chapter**: R. R. Basava, J.-S. Chen, Y. Zhang, S. Sinha, U. Sinha, J. Hodgson, R. Csapo, and V. Malis, "Pixel based meshfree modeling of skeletal muscles", in Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications (Y. Zhang and J. Tavares, eds.), vol. 8641 of Lecture Notes in Computer Science, pp. 316-327, Springer International Publishing, 2014. The dissertation author was the primary investigator and author of this book chapter. The contribution of all the co-authors is greatly acknowledged, particularly the second co-author for guiding the research. **Journal paper**: J.-S. Chen, R. R. Basava, Y. Zhang, R. Csapo, V. Malis, U. Sinha, J. Hodgson, and S. Sinha, "Pixel-based meshfree modelling of skeletal muscles", Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, no. (ahead-of-print), pp. 1-13, 2015. The contribution of all the co-authors is greatly acknowledged, particularly the first co-author for guiding the fundamental development of the research.

Chapter 5

Reproducing Kernel Collocation Method For Nonlinear Hyperelasticity

5.1 Derivatives of the RK shape function

The expression for the RK shape function is given in equation (4.21). The first derivative of the RK shape function is given by:

$$\Psi_{I,i}(\boldsymbol{x}) = C_{,i}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)\phi_a(\boldsymbol{x}-\boldsymbol{x}_I) + C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)\phi_{a,i}(\boldsymbol{x}-\boldsymbol{x}_I) \tag{5.1}$$

where $A_{,i} = \partial A/\partial x_i$, $i = 1, 2, \cdots, d$ and $C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)$ is the correction function. The derivative of the correction function is derived to be:

$$C_{,i}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}^T(\boldsymbol{0})\left[\boldsymbol{M}^{-1}_{,i}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I) + \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}_{,i}(\boldsymbol{x}-\boldsymbol{x}_I)\right] \tag{5.2}$$

When the strong form collocation methods are used for solving differential equations, the strong form of the differential equation is used directly which contains second order derivatives of the unknown variable. To approximate this, the second order deriva-

tives of the shape functions are required. The second derivative of the RK shape function is given by:

$$\begin{aligned}\Psi_{I,ij}(\boldsymbol{x}) = C_{,ij}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)\phi_a(\boldsymbol{x}-\boldsymbol{x}_I) + C_{,i}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)\phi_{a,j}(\boldsymbol{x}-\boldsymbol{x}_I) \\ + C_{,j}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)\phi_{a,i}(\boldsymbol{x}-\boldsymbol{x}_I) + C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)\phi_{a,ij}(\boldsymbol{x}-\boldsymbol{x}_I)\end{aligned} \tag{5.3}$$

where $A_{,ij} = \partial^2 A/\partial x_i \partial x_j$ and $i, j = 1, 2, \cdots, d$. The second derivative of the correction function is given by:

$$\begin{aligned}C_{,ij}(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}^T(\boldsymbol{0})[&\boldsymbol{M}^{-1}_{,ij}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I) \\ &+\boldsymbol{M}^{-1}_{,i}(\boldsymbol{x})\boldsymbol{H}_{,j}(\boldsymbol{x}-\boldsymbol{x}_I) \\ &+\boldsymbol{M}^{-1}_{,j}(\boldsymbol{x})\boldsymbol{H}_{,i}(\boldsymbol{x}-\boldsymbol{x}_I) \\ &+\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}_{,ij}(\boldsymbol{x}-\boldsymbol{x}_I)]\end{aligned} \tag{5.4}$$

In the above expressions the derivatives of the inverse of the moment matrix can be evaluated as follows:

$$\boldsymbol{M}^{-1}_{,i}(\boldsymbol{x}) = -\boldsymbol{M}^{-1}\boldsymbol{M}_{,i}\boldsymbol{M}^{-1} \tag{5.5a}$$

$$\boldsymbol{M}^{-1}_{,ij}(\boldsymbol{x}) = -\boldsymbol{M}^{-1}\left[\boldsymbol{M}_{,ij}\boldsymbol{M}^{-1} + \boldsymbol{M}_{,i}\boldsymbol{M}^{-1}_{,j} + \boldsymbol{M}_{,j}\boldsymbol{M}^{-1}_{,i}\right] \tag{5.5b}$$

In this work the quintic B-spline function given below is used as the kernel function.

$$\phi_a(z) = \begin{cases} \frac{11}{20} - \frac{9z^2}{2} + \frac{81z^4}{4} - \frac{81z^5}{4} & \text{for } 0 \le z < \frac{1}{3} \\ \frac{17}{40} + \frac{15z}{8} - \frac{63z^2}{4} + \frac{135z^3}{4} - \frac{243z^4}{8} + \frac{81z^5}{8} & \text{for } \frac{1}{3} \le z < \frac{2}{3} \\ \frac{81}{40} - \frac{81z}{8} + \frac{81z^2}{4} - \frac{81z^3}{4} + \frac{81z^4}{8} - \frac{81z^5}{40} & \text{for } \frac{2}{3} \le z < 1 \\ 0 & \text{for } z \ge 1 \end{cases} \tag{5.6}$$

where

$$z = \frac{\|\boldsymbol{x} - \boldsymbol{x_I}\|}{a}$$

5.2 Review of Reproducing Kernel Collocation Method (RKCM) for linear problems

For solving differential equations (DE) using the strong form collocation methods, the strong form of the DE is discretized directly for numerical implementation. The following procedure illustrates the implementation of the strong form collocation method using RK approximation for a general Boundary Value Problem (BVP). Typically a set of source points $\boldsymbol{S}$ with N_S number of points and set of collocation points $\boldsymbol{C}$ with N_C number of points are selected in the problem domain and usually $N_C > N_S$ to obtain accurate results. The source points and collocation points may or may not have common points. The numerical approximation is substituted in the strong form of the DE of the BVP and the residuals are enforced to be zero at the collocation points. This results in a system of equations which can be solved for the unknowns. The numerical procedure is presented below:

Consider the BVP in general form:

$$\boldsymbol{L}\boldsymbol{u} = \boldsymbol{f} \quad \text{in} \quad \Omega \in \mathbb{R}^d \tag{5.7a}$$

$$\boldsymbol{B}^h\boldsymbol{u} = \boldsymbol{h} \quad \text{on} \quad \partial\Omega^h \tag{5.7b}$$

$$\boldsymbol{B}^g\boldsymbol{u} = \boldsymbol{g} \quad \text{on} \quad \partial\Omega^g \tag{5.7c}$$

where Ω is the problem domain (excluding the boundaries), $\partial\Omega^h$ is the Natural boundary and $\partial\Omega^g$ is the Essential boundary. Here $\partial\Omega^h \cap \partial\Omega^g = \partial\Omega$ and $\partial\Omega \cap \Omega = \bar{\Omega}$ where $\bar{\Omega}$ is the entire problem domain. $\boldsymbol{L}$, $\boldsymbol{B}^h$ and $\boldsymbol{B}^g$ are the differential operators on Ω, $\partial\Omega^h$

and $\partial\Omega^g$, respectively. $\boldsymbol{f}$, $\boldsymbol{h}$ and $\boldsymbol{g}$ are the prescribed values for the forcing term, natural boundary condition and essential boundary condition on Ω, $\partial\Omega^h$ and $\partial\Omega^g$ respectively. The unknown variable '$\boldsymbol{u}$' can be a scalar like in heat conduction problems or a vector like in linear elasticity problems.

Introducing the approximation of $\boldsymbol{u}$ based on RK approximation:

$$\boldsymbol{u}^h(\boldsymbol{x}) = \boldsymbol{\Psi}^T \boldsymbol{a} \tag{5.8}$$

where $\boldsymbol{\Psi}^T = [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \cdots, \boldsymbol{\Psi}_{N_S}]$ with $\boldsymbol{\Psi}_I = \Psi_I \boldsymbol{I}_{\hat{d}\times\hat{d}}$, and $\boldsymbol{a}^T = \left\{\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \cdots, \boldsymbol{a}_{N_S}^T\right\}$ with $\boldsymbol{a}_I^T = \{a_{1I}, a_{2I}, \cdots, a_{\hat{d}I}\}$. Here Ψ_I represents the RK shape function at node 'I'. It can be seen that in the collocation method, the discretized points for constructing the approximation of $\boldsymbol{u}$ are the source points N_S. $\boldsymbol{I}_{\hat{d}\times\hat{d}}$ is the identity matrix of size $\hat{d}\times\hat{d}$ where $\boldsymbol{u} \in \mathbb{R}^{\hat{d}}$.

Let $\boldsymbol{P}$ be the set of N_P collocation points in Ω, $\boldsymbol{Q}$ be the set of N_Q collocation points on $\partial\Omega^h$ and $\boldsymbol{R}$ be the set of N_R collocation points on $\partial\Omega^g$.

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{N_P}] \in \Omega \tag{5.9a}$$

$$\boldsymbol{Q} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{N_Q}\right] \in \partial\Omega^h \tag{5.9b}$$

$$\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_{N_R}] \in \partial\Omega^g \tag{5.9c}$$

The total number of collocation points will be $N_C = N_P + N_Q + N_R$ denoted previously by set $\boldsymbol{C}$.

$$\boldsymbol{C} = \left[\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{N_P}, \boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{N_Q}, \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_{N_R}\right] \in \bar{\Omega} \tag{5.10}$$

The approximation of $\boldsymbol{u}$ given in equation (5.8) is substituted in the strong form

of the BVP (5.7) and the residual is forced to be zero at the collocation points which leads to a system of equations given by:

$$\boldsymbol{A}\boldsymbol{a} = \boldsymbol{b} \tag{5.11}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \boldsymbol{A}_3 \end{bmatrix}; \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \end{bmatrix} \tag{5.12}$$

and

$$\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{L}(\Psi^T(\boldsymbol{p}_1)) \\ \boldsymbol{L}(\Psi^T(\boldsymbol{p}_2)) \\ \vdots \\ \boldsymbol{L}(\Psi^T(\boldsymbol{p}_{N_P})) \end{bmatrix}; \boldsymbol{A}_2 = \begin{bmatrix} \boldsymbol{B}^{\boldsymbol{h}}(\Psi^T(\boldsymbol{q}_1)) \\ \boldsymbol{B}^{\boldsymbol{h}}(\Psi^T(\boldsymbol{q}_2)) \\ \vdots \\ \boldsymbol{B}^{\boldsymbol{h}}(\Psi^T(\boldsymbol{q}_{N_Q})) \end{bmatrix}; \boldsymbol{A}_3 = \begin{bmatrix} \boldsymbol{B}^{\boldsymbol{g}}(\Psi^T(\boldsymbol{r}_1)) \\ \boldsymbol{B}^{\boldsymbol{g}}(\Psi^T(\boldsymbol{r}_2)) \\ \vdots \\ \boldsymbol{B}^{\boldsymbol{g}}(\Psi^T(\boldsymbol{r}_{N_R})) \end{bmatrix} \tag{5.13}$$

$$\boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{p}_1) \\ \boldsymbol{f}(\boldsymbol{p}_2) \\ \vdots \\ \boldsymbol{f}(\boldsymbol{p}_{N_P}) \end{bmatrix}; \boldsymbol{b}_2 = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{q}_1) \\ \boldsymbol{h}(\boldsymbol{q}_2) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{q}_{N_Q}) \end{bmatrix}; \boldsymbol{b}_3 = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{r}_1) \\ \boldsymbol{g}(\boldsymbol{r}_2) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{r}_{N_R}) \end{bmatrix} \tag{5.14}$$

It was shown in [78] that there is unbalanced error between the domain, natural boundary and essential boundary terms given in equation (5.11). This unbalanced error can be balanced by properly weighting the boundary terms which gives the Weighted Collocation Method (WCM) and the system of equations can be re-written as follows

with the weight terms included:

$$\begin{bmatrix} \boldsymbol{A}_1 \\ \sqrt{\alpha^h}\boldsymbol{A}_2 \\ \sqrt{\alpha^g}\boldsymbol{A}_3 \end{bmatrix} \boldsymbol{a} = \begin{bmatrix} \boldsymbol{b}_1 \\ \sqrt{\alpha^h}\boldsymbol{b}_2 \\ \sqrt{\alpha^g}\boldsymbol{b}_3 \end{bmatrix} \tag{5.15}$$

where $\sqrt{\alpha^h}$ and $\sqrt{\alpha^g}$ are the weights for the natural and essential boundary terms respectively. For the Poisson problem, the weights of $\sqrt{\alpha^h} \approx \mathcal{O}(1)$ and $\sqrt{\alpha^g} \approx \mathcal{O}(N_S)$ should be used and for the elasticity problem weights are derived to be $\sqrt{\alpha^h} \approx \mathcal{O}(1)$ and $\sqrt{\alpha^g} \approx \mathcal{O}(max\{\lambda, \mu\}N_S)$ where λ and μ are the Lame's constants [78].

Since the number of collocation points are taken to be greater than the number of source points, the above system of equations is over determined as there are more number of equations than unknowns. This linear system (5.11) can be solved for the unknown nodal coordinates '$\boldsymbol{a}$' using the method of Least Squares where the square of the Euclidean norm of the residual $\boldsymbol{e} = \boldsymbol{A}\boldsymbol{a} - \boldsymbol{b}$ is minimized with respect to $\boldsymbol{a}$, which leads to the equation:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{a} = \boldsymbol{A}^T\boldsymbol{b} \tag{5.16}$$

where $\boldsymbol{A}^T\boldsymbol{A}$ is a square invertible matrix.

5.2.1 Convergence of RK approximation and RKCM error

Let $\boldsymbol{V} = \text{span}\{\Psi_1, \Psi_2, \cdots, \Psi_{N_S}\}$ where $\Psi_I \in H_1^m$ is the RK shape function of degree 'p' and the support size satisfies minimum requirement. Let $\boldsymbol{u}^h \in \boldsymbol{V}$ be the RK approximation of $\boldsymbol{u}$. The following property is given for the RK approximation error estimation [122]:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_{L,\Omega} \leq Ch^r|\boldsymbol{u}|_{p+1,\Omega} \tag{5.17}$$

where $r = p + 1 - L$ is the rate of convergence and 'h' is the nodal spacing. The following estimates can be obtained from equation (5.17) [78, 79]:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_{0,\Omega} \approx \mathcal{O}h^{p+1} \tag{5.18a}$$

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_{1,\Omega} \approx \mathcal{O}h^{p} \tag{5.18b}$$

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_{2,\Omega} \approx \mathcal{O}h^{p-1} \tag{5.18c}$$

The RKCM method converges in the following norm [78, 79, 73]:

$$\left|\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|\right| = \inf_{\boldsymbol{v}\in\boldsymbol{V}} \left|\left\|\boldsymbol{u} - \boldsymbol{v}\right\|\right| \tag{5.19a}$$

$$\begin{aligned}\left|\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|\right| &\leq C\left\{\|\boldsymbol{u} - \boldsymbol{v}\|_{2,\Omega} + \|(\boldsymbol{u} - \boldsymbol{v})_n\|_{0,\partial\Omega^h} + \|\boldsymbol{u} - \boldsymbol{v}\|_{0,\partial\Omega^g}\right\} \\ &\leq Ch^{p-1}|\boldsymbol{u}|_{p+1,\Omega}\end{aligned} \tag{5.19b}$$

It can be observed from the above equation (5.19) that the RKCM error converges only if $p \geq 2$. So RK shape functions should be constructed using at least quadratic basis for the solution to converge. In all the examples that follow, quadratic bases are used in the RK shape functions. The numerical convergence rates in the following examples are measured by computing the L_2 and H_1 norm of the error in the solution, as defined below.

L_2 norm:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_0 = \left(\int_\Omega (u_i^h - u_i)(u_i^h - u_i)d\Omega\right)^{1/2} \tag{5.20}$$

H_1 norm:

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_1 = \left(\int_\Omega (u_i^h - u_i)(u_i^h - u_i)d\Omega + \int_\Omega (u_{i,j}^h - u_{i,j})(u_{i,j}^h - u_{i,j})d\Omega\right)^{1/2} \tag{5.21}$$

where $\boldsymbol{u}$ is the analytical solution and $\boldsymbol{u}^h$ is the numerical solution obtained from RKCM.

5.2.2 Numerical examples

For all the examples in this section, quadratic basis and a support size of $3h$ is used for the RK shape function where 'h' is the nodal spacing of the source points.

5.2.2.1 1D Poisson problem

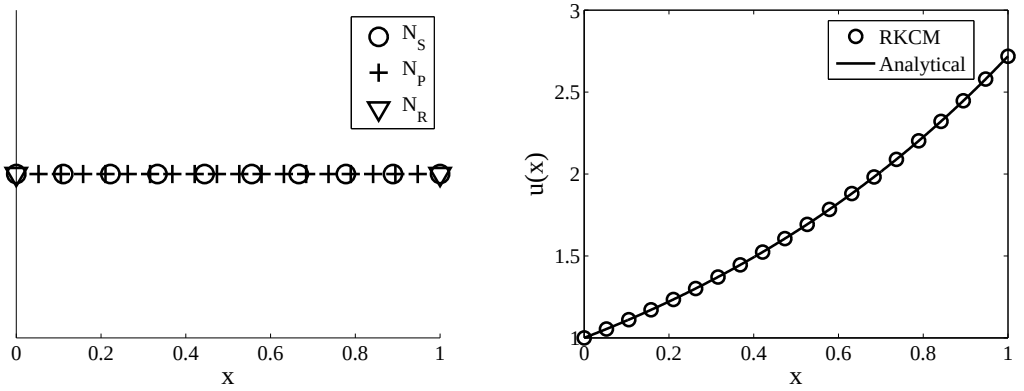
Consider the 1D Poisson problem with essential boundary conditions as given below:

$$\Delta u(x) = e^x \quad \text{in} \quad \Omega = \{x|0 < x < 1\} \tag{5.22a}$$

$$u(x) = e^x \quad \text{on} \quad \partial\Omega = \{x|x = 0, x = 1\} \tag{5.22b}$$

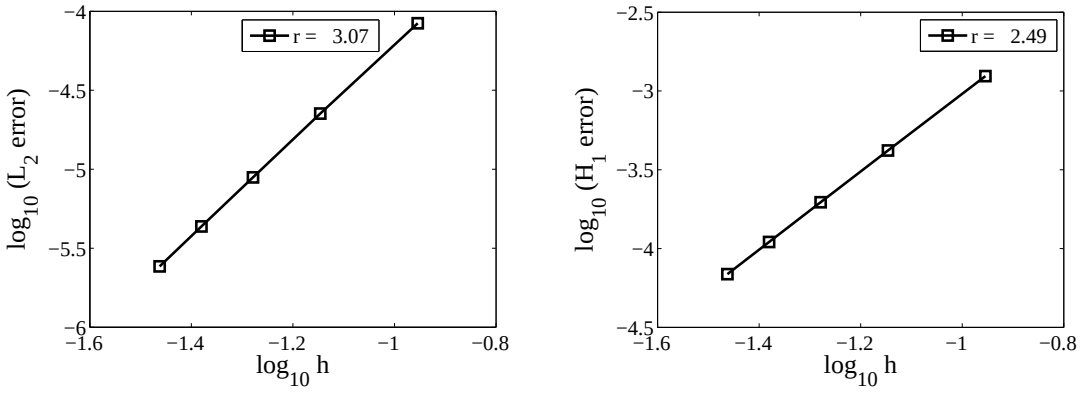
The analytical solution to the problem is $u(x) = e^x$. The domain is discretized using 10 source points and 20 collocation points which do not overlap in the interior domain as shown in figure 5.1(a) and the problem is solved using RKCM. The weight on the essential boundary is taken to be $\sqrt{\alpha^g} = N_S$. The numerical solution obtained in the domain using RKCM is plotted in figure 5.1(b) and compared with the analytical solution. The RKCM result is very close to the analytical solution.

The convergence of the solution in L_2 and H_1 norm using RKCM was studied using a discretization of $\{10, 15, 20, 25, 30\}$ source points and $\{20, 30, 40, 50, 60\}$ collocation points. Gauss integration was used for integrating the error. The convergence plots are shown in figure 5.2. The rate of convergence obtained in L_2 norm is 3.07 which corresponds well to the theoretical prediction of 3. The rate of convergence obtained in H_1 norm is 2.49 which is better than the theoretical prediction of 2.



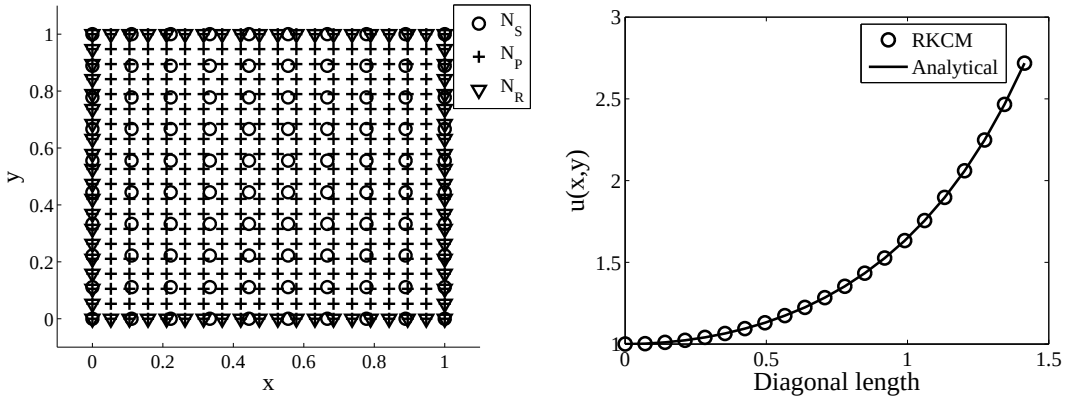
(a) Domain discretization (b) Comparison of solution

Figure 5.1: Domain discretization and RKCM solution comparison for 1D Poisson problem



(a) L_2 norm (b) H_1 norm

Figure 5.2: Convergence of RKCM for 1D Poisson problem



(a) Domain discretization (b) Comparison of solution

Figure 5.3: Domain discretization and RKCM solution comparison for 2D Poisson problem

5.2.2.2 2D Poisson problem

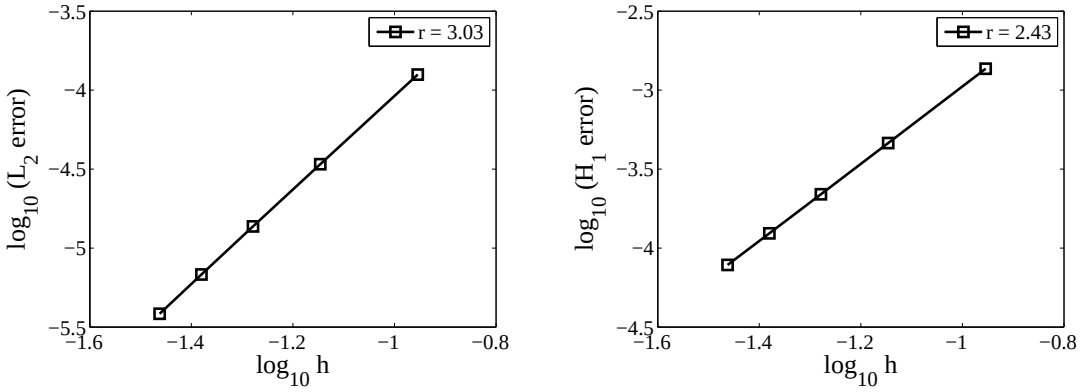
Consider the 2D Poisson problem with essential boundary conditions as given below:

$$\Delta u(x,y) = (x^2+y^2)e^{xy} \quad \text{in} \quad \Omega = \{(x,y)|0<x<1, 0<y<1\} \tag{5.23a}$$

$$u(x,y) = e^{xy} \quad \text{on} \quad \partial\Omega = \{(x,y)|x=0,1 \text{ and } y=0,1\} \tag{5.23b}$$

The analytical solution to the problem is $u(x,y) = e^{xy}$. The domain is discretized using 10×10 source points and 20×20 collocation points which do not overlap in the problem domain as shown in figure 5.3(a) and the problem is solved using RKCM. The weight on the essential boundary is taken to be $\sqrt{\alpha^g} = N_S$. The numerical solution using RKCM obtained along the diagonal line passing through the points $(0,0)$ and $(1,1)$ is plotted in figure 5.3(b) and compared with the analytical solution. The RKCM result is very close to the analytical solution.

The convergence of the solution in L_2 and H_1 norm using RKCM was studied using a discretization of $\{10\times10, 15\times15, 20\times20, 25\times25, 30\times30\}$ source points and



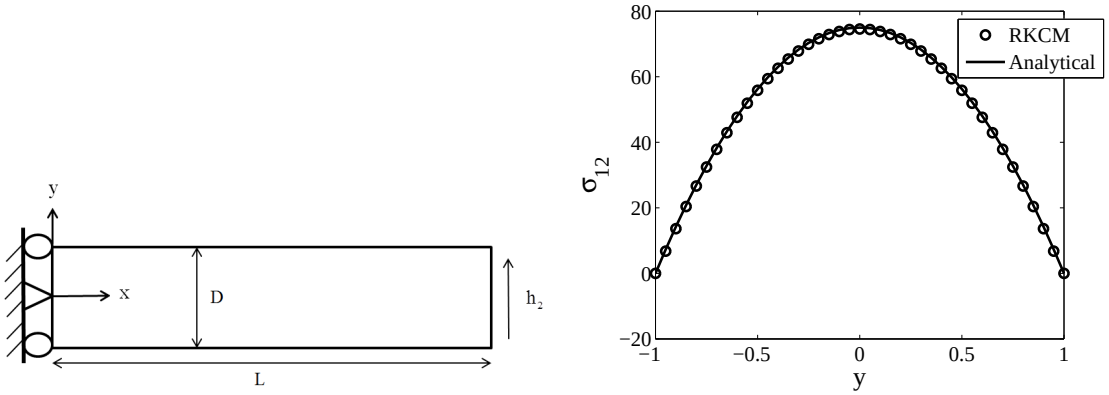
(a) L_2 norm (b) H_1 norm

Figure 5.4: Convergence of RKCM for 2D Poisson problem

$\{20 \times 20, 30 \times 30, 40 \times 40, 50 \times 50, 60 \times 60\}$ collocation points. Gauss integration was used for integrating the error. The convergence plots are shown in figure 5.4. The rate of convergence obtained in L_2 norm is 3.03 which corresponds well to the theoretical prediction of 3. The rate of convergence obtained in H_1 norm is 2.43 which better than the theoretical prediction of 2.

5.2.2.3 2D Linear elastic cantilever beam

Consider a plane strain elastic beam subjected to traction on the right edge as shown in figure 5.5(a). The equilibrium equations and associated boundary conditions



(a) Cantilever beam geometry and boundary conditions

(b) Comparison of shear stress along $x = L/2$

Figure 5.5: Domain discretization and RKCM solution comparison for 2D linear elastic cantilever beam

are given by:

$$\sigma_{ij,j} = 0 \quad \text{in} \quad \Omega = \{(x,y)|0 < x < L, \quad -D/2 < y < D/2\} \tag{5.24a}$$

$$\text{at} \quad x = 0, y = 0, \quad u_1 = u_2 = 0$$

$$\text{at} \quad x = 0, y = \pm D/2, \quad u_1 = h_2 = 0$$

$$\text{on} \quad x = L, -D/2 \leq y \leq D/2, \quad h_1 = 0, h_2 = \frac{6P}{D^3}\left(\frac{D^2}{4} - y^2\right)$$

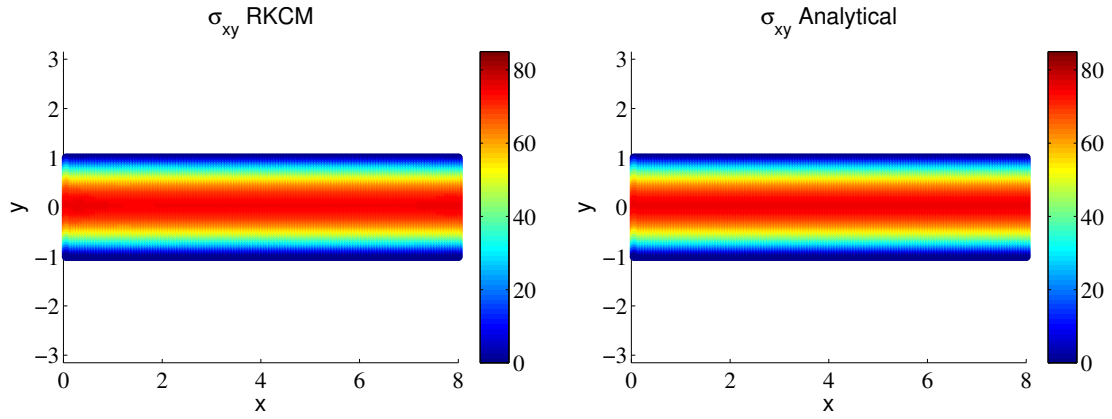
$$\text{on} \quad x = 0, -D/2 < y < 0, 0 < y < D/2, \quad h_1 = \frac{12PL}{D^3}y, h_2 = -\frac{6P}{D^3}\left(\frac{D^2}{4} - y^2\right)$$

$$\text{on} \quad 0 < x < L, y = \pm D/2, \quad h_1 = h_2 = 0 \tag{5.24b}$$

The analytical solution to the problem is given by:

$$u_1(x,y) = -\frac{Py}{6\bar{E}I}\left[(6L - 3x)x + (2 + \bar{\nu})\left(y^2 - \frac{D^2}{4}\right)\right] \tag{5.25a}$$

$$u_2(x,y) = \frac{P}{6\bar{E}I}\left[(3L - x)x^2 + 3\bar{\nu}y^2(L - x) + (4 + 5\bar{\nu})\frac{D^2 x}{4}\right] \tag{5.25b}$$



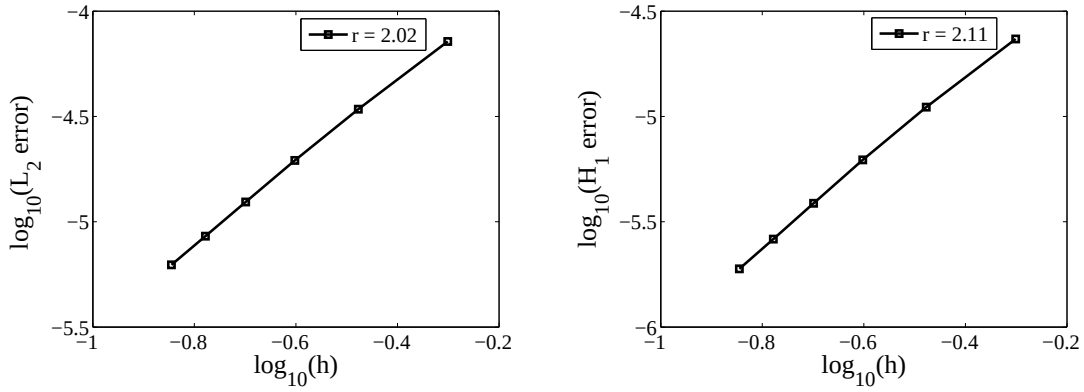
(a) RKCM solution for shear stress

(b) Analytical solution for shear stress

Figure 5.6: Comparison of RKCM and analytical shear stress for cantilever beam problem

Here $I = D^3/12$, $\bar{E} = E/(1-\nu^2)$ and $\bar{\nu} = \nu/(1-\nu)$, where E, ν are the Young's modulus and Poisson's ratio respectively. The values of parameters used are: $L = 8m$, $D = 2m$, $P = 100N/m$, $E = 3 \times 10^7 Pa$ and $\nu = 0.25$. The domain of the beam is discretized using 73×19 source points and 147×39 collocation points and the problem is solved using RKCM. The weight on the essential boundary is taken to be $\sqrt{\alpha^g} = max\{\lambda, \mu\}N_S$ and weight on the natural boundary is taken to be $\sqrt{\alpha^h} = 1$. The comparison of RKCM solution with exact solution for the shear stress along line $x = L/2$ is shown in figure 5.5(b) and the comparison of the solution for shear stress in the entire domain is shown in figure 5.6.

The convergence of the solution in L_2 norm and H_1 norm using RKCM was studied using the discretization of $\{17 \times 5, 25 \times 7, 33 \times 9, 41 \times 11, 49 \times 13, 57 \times 15\}$ source points and $\{35 \times 11, 51 \times 15, 67 \times 19, 83 \times 23, 99 \times 27, 115 \times 31\}$ collocation points. Gauss integration was used for integrating the error. The convergence plots are shown in figure 5.7. The rates of convergence obtained are shown on the figure.



(a) L_2 norm (b) H_1 norm

Figure 5.7: Convergence of RKCM for cantilever beam problem

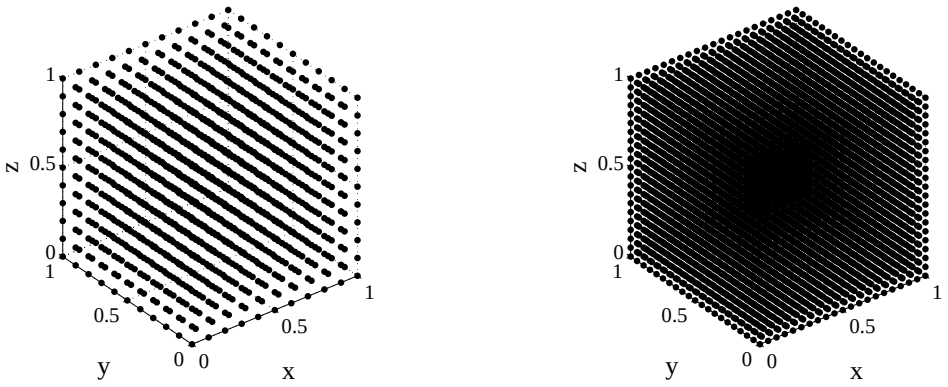
5.2.2.4 3D Poisson problem

Consider the following 3D Poisson problem with given essential boundary conditions:

$$\Delta u(x,y,z) = f \quad \text{in} \quad \Omega = \{(x,y,z)|0 < x < 1, 0 < y < 1, 0 < z < 1\} \tag{5.26a}$$

$$u(x,y,z) = 0 \quad \text{on} \quad \partial\Omega = \{(x,y,z)|x = 0,1; y = 0,1 \text{ and } z = 0,1\} \tag{5.26b}$$

Here $f = -3\pi^2 sin(\pi x)sin(\pi y)sin(\pi z)$. The analytical solution to the problem is $u(x,y,z) = sin(\pi x)sin(\pi y)sin(\pi z)$ as given in [123]. The domain is discretized using $11 \times 11 \times 11$ source points and $21 \times 21 \times 21$ collocation points as shown in figure 5.8. The weight on the essential boundary is taken to be $\sqrt{\alpha^g} = N_S$. The numerical solution using RKCM obtained along the diagonal line passing through the points $(0,0,0)$ and $(1,1,1)$ is plotted in figure 5.9 and compared with the analytical solution. The RKCM result is very close to the analytical solution.



(a) Source points (b) Collocation points

Figure 5.8: Domain discretization for 3D Poisson problem

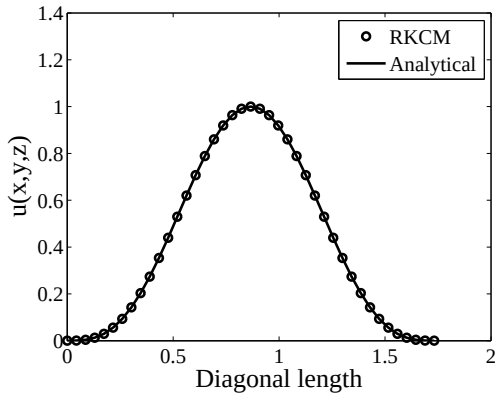


Figure 5.9: Comparison of RKCM and analytical solution for 3D Poisson problem

5.3 RKCM for nonlinear hyperelasticity

In conventional meshfree methods based on the weak form, background meshes are required for the purpose of domain integration. Additionally, imposing essential boundary conditions requires special treatment since the meshfree approximation functions are usually not interpolating functions. In this chapter the meshfree strong form Reproducing Kernel Collocation Method (RKCM) for solving boundary value problems in context of nonlinear elasticity (in particular hyperelasticity) is introduced. The RK approximation function with a material kernel [54] that deforms with material deformation is used. Several examples are provided in the context of deformation of rubber, which is classified as an isotropic hyperelastic material. The method is further implemented for modeling human skeletal muscles using image data. The muscle models are constructed from segmentation of anatomical Magnetic Resonance (MR) images using level set based techniques as described in chapter 4. The pixel points obtained from these segmented MR images are used directly as nodes (source and collocation points) for domain discretization in the meshfree RKCM formulation. This method fully eliminates the need for complex mesh construction and domain integration for complex geometries. Also both essential and natural boundary conditions can be enforced directly. The method is derived by least squares minimization and linearization (using Newton's method) of the strong form nonlinear collocation equations. Additionally it can be derived by variation of the weighted least squares functional followed by linearization and collocation. Both the above derivations give the exact tangent operators for the incremental equations. Further it is shown that direct linearization of the strong form governing equations followed by least squares solution to solve the over-determined system results in the Gauss Newton method for the incremental iterative solution, where the tangent operator is approximate and not exact. The detailed derivations for the three

cases, computational algorithm and examples are given in the following sections.

5.3.1 Strong form of the governing equations for nonlinear elasticity

For nonlinear elasticity the strong form of the governing equations related to the undeformed configuration are considered and are given by:

$$\sigma_{Ai}(\boldsymbol{u})_{,A} + b_i = 0 \quad \text{in} \quad \Omega_X \tag{5.27a}$$

$$\sigma_{Ai}(\boldsymbol{u})N_A = h_i \quad \text{on} \quad \partial\Omega_X^h \tag{5.27b}$$

$$u_i = g_i \quad \text{on} \quad \partial\Omega_X^g \tag{5.27c}$$

where $\boldsymbol{\sigma}$ is the first Piola-Kirchhoff stress (PK1) and is a nonlinear function of $\boldsymbol{u}$, $(\bullet)_{,A} = \partial(\bullet)/\partial X_A$ is the derivative with respect to the material coordinate $\boldsymbol{X}$, $\boldsymbol{b}$ is the body force per unit undeformed volume, $\boldsymbol{h}$ is the surface force per unit undeformed area on the natural boundary $\partial\Omega_X^h$, $\boldsymbol{N}$ is the outward normal vector to the surface $\partial\Omega_X^h$, $\boldsymbol{u}$ is the displacement and $\boldsymbol{g}$ is the prescribed displacement on the essential boundary $\partial\Omega_X^g$. Additionally Ω_X represents the problem domain excluding the boundary, $\partial\Omega_X = \partial\Omega_X^h \cap \partial\Omega_X^g$ represents the entire boundary and $\bar{\Omega}_X = \Omega_X \cap \partial\Omega_X$ represents the full problem domain.

The deformation of a material particle $\boldsymbol{X} \in \bar{\Omega}_X$ at time t is described by the one-to-one mapping function given by $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t)$ and the displacement of particle $\boldsymbol{X}$ is defined by:

$$\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{\phi}(\boldsymbol{X}, t) - \boldsymbol{X} = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X} \tag{5.28}$$

The deformation gradient $\boldsymbol{F}$, Green-Lagrangian strain $\boldsymbol{E}$ and right Cauchy Green strain

tensor $\boldsymbol{C}$, are defined as follows:

$$F_{ij} = \frac{\partial x_i}{\partial X_j} = \frac{\partial u_i}{\partial X_j} + \delta_{ij} \tag{5.29a}$$

$$E_{ij} = \frac{1}{2}(F_{ki}F_{kj} - \delta_{ij}) \tag{5.29b}$$

$$C_{ij} = 2E_{ij} + \delta_{ij} = F_{ki}F_{kj} \tag{5.29c}$$

For hyperelastic material, PK1 is calculated from the strain energy density function W using the relationship:

$$\sigma_{Ai} = \frac{\partial W}{\partial F_{iA}} \tag{5.30}$$

The second Piola-Kirchhoff stress (PK2), $\boldsymbol{S}$ and Cauchy stress $\boldsymbol{\tau}$ can be obtained from PK1 using the following relationships:

$$\boldsymbol{S} = \boldsymbol{\sigma}\boldsymbol{F}^{-T} \tag{5.31a}$$

$$\boldsymbol{\tau} = J^{-1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^{T} \tag{5.31b}$$

where $J = det(\boldsymbol{F})$.

5.3.2 Review of hyperelasticity for nearly incompressible materials

The strain energy density for elastic materials can be expressed in terms of the three invariants (I_1, I_2, I_3) of the right Cauchy Green strain tensor [124, 125]:

$$W(I_1, I_2, I_3) = \sum_{i+j+k=1}^{\infty} A_{ijk}(I_1 - 3)^i(I_2 - 3)^j(I_3 - 1)^k \tag{5.32}$$

where

$$I_1 = tr(\boldsymbol{C}) \tag{5.33a}$$

$$I_2 = \frac{1}{2}\left[(tr(\boldsymbol{C}))^2 - tr(\boldsymbol{C}^2)\right] \tag{5.33b}$$

$$I_3 = det(\boldsymbol{C}) \tag{5.33c}$$

and A_{ijk} are the material constants. If the material behavior is incompressible, $I_3 = 1$ and the strain energy density reduces to:

$$W(I_1, I_2) = \sum_{i+j=1}^{\infty} A_{ij}(I_1 - 3)^i (I_2 - 3)^j \tag{5.34}$$

For nearly incompressible materials, a volume conservation condition is required to be imposed (using Penalty or Lagrange multiplier method) and the strain energy density is modified as:

$$W(I_1, I_2, I_3) = \bar{W}(I_1, I_2) + \tilde{W}(I_3) \tag{5.35}$$

where $\tilde{W}(I_3)$ is the part of the strain energy density which accounts for the small change in volume. Additionally for nearly incompressible materials, the invariants I_1 and I_2 are not measures of pure distortion alone and contain a certain amount of dilatational strain energy. A fully decoupled strain energy density function can be written as follows [124, 125]:

$$W(\bar{I}_1, \bar{I}_2, J) = \bar{W}(\bar{I}_1, \bar{I}_2) + \tilde{W}(J) \tag{5.36}$$

where $\bar{W}(\bar{I}_1, \bar{I}_2)$ and $\tilde{W}(J)$ are the distortional and dilatational strain energy density functions, respectively, given by:

$$\bar{W}(\bar{I}_1, \bar{I}_2) = \sum_{i+j=1}^{\infty} A_{ij}(\bar{I}_1 - 3)^i(\bar{I}_2 - 3)^j \tag{5.37a}$$

$$\tilde{W}(J) = \frac{K}{2}(J-1)^2 \tag{5.37b}$$

Here A_{ij} are the material constants and K is the bulk modulus. $\bar{I}_1, \bar{I}_2$ are the reduced invariants which separate the distortional and dilatational deformation:

$$\bar{I}_1 = I_1 I_3^{-1/3} \tag{5.38a}$$

$$\bar{I}_2 = I_2 I_3^{-2/3} \tag{5.38b}$$

The hydrostatic pressure P is related to $\tilde{W}(J)$ by:

$$P = \frac{\partial \tilde{W}}{\partial J} \tag{5.39}$$

5.3.3 Derivation from strong form of the nonlinear governing equations

Consider the strong form of the governing equations in the undeformed configuration as given in equation (5.27). These equations can be written in operator form as given below:

$$\boldsymbol{l}(\boldsymbol{u}) + \boldsymbol{f} = \boldsymbol{0} \quad \text{in} \quad \Omega_X \tag{5.40a}$$

$$\boldsymbol{b}^h(\boldsymbol{u}) = \boldsymbol{h} \quad \text{on} \quad \partial\Omega_X^h \tag{5.40b}$$

$$\boldsymbol{b}^g(\boldsymbol{u}) = \boldsymbol{g} \quad \text{on} \quad \partial\Omega_X^g \tag{5.40c}$$

where the nonlinear operators are given by the following matrices:

$$\boldsymbol{l}(\boldsymbol{u}) = \begin{bmatrix} \sigma_{A1}(\boldsymbol{u})_{,A} \\ \sigma_{A2}(\boldsymbol{u})_{,A} \\ \sigma_{A3}(\boldsymbol{u})_{,A} \end{bmatrix} ; \quad \boldsymbol{b}^h(\boldsymbol{u}) = \begin{bmatrix} \sigma_{A1}(\boldsymbol{u})N_A \\ \sigma_{A2}(\boldsymbol{u})N_A \\ \sigma_{A3}(\boldsymbol{u})N_A \end{bmatrix} ; \quad \boldsymbol{b}^g(\boldsymbol{u}) = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{5.41}$$

and the forcing terms are given by:

$$\boldsymbol{f} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} ; \quad \boldsymbol{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} ; \quad \boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{5.42}$$

Let $\boldsymbol{P}$ be the set of N_P collocation points in domain Ω_X, $\boldsymbol{Q}$ be the set of N_Q collocation points on $\partial\Omega_X^h$ and $\boldsymbol{R}$ be the set of N_R collocation points on $\partial\Omega_X^g$ defined as:

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{N_P}] \in \Omega_X \tag{5.43a}$$

$$\boldsymbol{Q} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{N_Q}\right] \in \partial\Omega_X^h \tag{5.43b}$$

$$\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_{N_R}] \in \partial\Omega_X^g \tag{5.43c}$$

The total number of collocation points will be $N_C = N_P + N_Q + N_R$ denoted by set $\boldsymbol{C}$.

$$\boldsymbol{C} = \left[\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{N_P}, \boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{N_Q}, \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_{N_R}\right] \in \bar{\Omega} \tag{5.44}$$

The approximation of $\boldsymbol{u}$ denoted by $\boldsymbol{u}^h$and based on the RK approximation functions is given by:

$$\boldsymbol{u}^h(\boldsymbol{x}) = \boldsymbol{\Psi}^T \boldsymbol{a} \tag{5.45}$$

where $\boldsymbol{\Psi}^T = [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \cdots, \boldsymbol{\Psi}_{N_S}]$ with $\boldsymbol{\Psi}_I = \Psi_I \boldsymbol{I}_{\hat{d}\times\hat{d}}$, and $\boldsymbol{a}^T = \left\{\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \cdots, \boldsymbol{a}_{N_S}^T\right\}$ with $\boldsymbol{a}_I^T = \{a_{1I}, a_{2I}, \cdots, a_{\hat{d}I}\}$. Here Ψ_I represents the RK shape function at node 'I'

and N_S denote a set of source points on which the RK approximation of $\boldsymbol{u}$ is constructed. $\boldsymbol{I}_{\hat{d}\times\hat{d}}$ is the identity matrix of size $\hat{d}\times\hat{d}$ where $\boldsymbol{u}\in\mathbb{R}^{\hat{d}}$.

Typically the number of collocation points N_C selected are greater than the number of source points N_S to obtain accurate results. Substituting $\boldsymbol{u}^h$ in the strong form governing equations (5.40), and enforcing the residual to be zero at the collocation points N_C results in the following set of over-determined system of nonlinear equations in $\boldsymbol{a}$:

$$\begin{aligned}
&\boldsymbol{l}(\boldsymbol{u}^h(\boldsymbol{p}_1))+\boldsymbol{f}(\boldsymbol{p}_1)=\boldsymbol{0}\\
&\vdots\\
&\boldsymbol{l}(\boldsymbol{u}^h(\boldsymbol{p}_{N_P}))+\boldsymbol{f}(\boldsymbol{p}_{N_P})=\boldsymbol{0}\\
&\boldsymbol{b}^h(\boldsymbol{u}^h(\boldsymbol{q}_1))-\boldsymbol{h}(\boldsymbol{q}_1)=\boldsymbol{0}\\
&\vdots\\
&\boldsymbol{b}^h(\boldsymbol{u}^h(\boldsymbol{q}_{N_Q}))-\boldsymbol{h}(\boldsymbol{q}_{N_Q})=\boldsymbol{0}\\
&\boldsymbol{b}^g(\boldsymbol{u}^h(\boldsymbol{r}_1))-\boldsymbol{g}(\boldsymbol{r}_1)=\boldsymbol{0}\\
&\vdots\\
&\boldsymbol{b}^g(\boldsymbol{u}^h(\boldsymbol{r}_{N_R}))-\boldsymbol{g}(\boldsymbol{r}_{N_R})=\boldsymbol{0}
\end{aligned}\tag{5.46}$$

Here it is assumed that $\boldsymbol{f}$ and $\boldsymbol{h}$ are independent of deformation. The above equation can be re-written as:

$$\boldsymbol{\xi}(\boldsymbol{a})-\boldsymbol{\eta}=\boldsymbol{0}\tag{5.47}$$

where

$$\boldsymbol{\xi}(\boldsymbol{a})=\begin{bmatrix}\boldsymbol{\xi}_1(\boldsymbol{a})\\ \boldsymbol{\xi}_2(\boldsymbol{a})\\ \boldsymbol{\xi}_3(\boldsymbol{a})\end{bmatrix};\quad \boldsymbol{\eta}=\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\\ \boldsymbol{\eta}_3\end{bmatrix}\tag{5.48}$$

with

$$\boldsymbol{\xi}_1(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{l}(\boldsymbol{u}^h(\boldsymbol{p}_1)) \\ \vdots \\ \boldsymbol{l}(\boldsymbol{u}^h(\boldsymbol{p}_{N_P})) \end{bmatrix}; \boldsymbol{\xi}_2(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{b}^h(\boldsymbol{u}^h(\boldsymbol{q}_1)) \\ \vdots \\ \boldsymbol{b}^h(\boldsymbol{u}^h(\boldsymbol{q}_{N_Q})) \end{bmatrix}; \boldsymbol{\xi}_3(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{b}^g(\boldsymbol{u}^h(\boldsymbol{r}_1)) \\ \vdots \\ \boldsymbol{b}^g(\boldsymbol{u}^h(\boldsymbol{r}_{N_R})) \end{bmatrix} \tag{5.49}$$

$$\boldsymbol{\eta}_1 = \begin{bmatrix} -\boldsymbol{f}(\boldsymbol{p}_1) \\ \vdots \\ -\boldsymbol{f}(\boldsymbol{p}_{N_P}) \end{bmatrix}; \boldsymbol{\eta}_2 = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{q}_1) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{q}_{N_Q}) \end{bmatrix}; \boldsymbol{\eta}_3 = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{r}_1) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{r}_{N_R}) \end{bmatrix} \tag{5.50}$$

A weighted equation is considered analogous to linear problems, to balance the error between the domain and the boundary terms. This results in the following system of equations:

$$\boldsymbol{\xi}(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{\xi}_1(\boldsymbol{a}) \\ \sqrt{\alpha^h}\boldsymbol{\xi}_2(\boldsymbol{a}) \\ \sqrt{\alpha^g}\boldsymbol{\xi}_3(\boldsymbol{a}) \end{bmatrix}; \quad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \sqrt{\alpha^h}\boldsymbol{\eta}_2 \\ \sqrt{\alpha^g}\boldsymbol{\eta}_3 \end{bmatrix} \tag{5.51}$$

where $\sqrt{\alpha^h}$ is the weight on the natural boundary and $\sqrt{\alpha^g}$ is the weight on the essential boundary. Following the results given in [78, 79], the following weights are used for hyperelasticity:

$$\sqrt{\alpha^h} \approx \mathcal{O}(1) \tag{5.52a}$$

$$\sqrt{\alpha^g} \approx \mathcal{O}(\kappa N_S) \tag{5.52b}$$

where κ is taken to be the maximum value of strain energy density at that boundary collocation point.

The above over-determined system is solved by minimizing the least squares

error of these nonlinear equations, as follows:

$$\Pi(\boldsymbol{a}) = \boldsymbol{e} \tag{5.53}$$

with $\boldsymbol{e} = \frac{1}{2}(\boldsymbol{\xi}(\boldsymbol{a}) - \boldsymbol{\eta})^T(\boldsymbol{\xi}(\boldsymbol{a}) - \boldsymbol{\eta})$. Minimization of this functional with respect to $\boldsymbol{a}$ gives:

$$\frac{\partial \boldsymbol{e}}{\partial a_i}\delta a_i = 0 \quad \Rightarrow \frac{\partial \boldsymbol{e}}{\partial a_i} = 0, \qquad \forall \delta a_i \neq 0 \tag{5.54}$$

where $a_i = \boldsymbol{a}(i)$. This leads to the following nonlinear equations in $\boldsymbol{a}$:

$$\boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{\xi}(\boldsymbol{a}) - \boldsymbol{\eta}) = \boldsymbol{0} \tag{5.55}$$

where $\boldsymbol{A}'$ is the transpose of the Jacobian matrix $\boldsymbol{J}$ and is given by:

$$\boldsymbol{A}'(\boldsymbol{a}) = \boldsymbol{J}^T(\boldsymbol{a}) = \begin{bmatrix} \frac{\partial \boldsymbol{\xi}^T(\boldsymbol{a})}{\partial a_1} \\ \vdots \\ \frac{\partial \boldsymbol{\xi}^T(\boldsymbol{a})}{\partial a_N} \end{bmatrix} \tag{5.56}$$

Here, N is the length of the vector $\boldsymbol{a}$. For example in 3D, $N = 3N_S$. The above nonlinear equations are solved using the Newton's method, where an incremental equation is considered. This requires linearization of equation (5.55). Let n denote the load step number and ν denote the iteration number. The body force, surface traction, prescribed displacements and deformed state of the body at the $n+1^{\text{th}}$ load step and ν^{th} iteration are known from previous iterations. The linearization is carried out as follows to obtain the incremental nodal coordinates $\Delta\boldsymbol{a}$:

$$\left(\frac{\partial \boldsymbol{A}'(\boldsymbol{a})}{\partial a_1}\Delta a_1 + \cdots + \frac{\partial \boldsymbol{A}'(\boldsymbol{a})}{\partial a_N}\Delta a_N\right)\underbrace{(\boldsymbol{\xi}(\boldsymbol{a}) - \boldsymbol{\eta})}_{\boldsymbol{r}(\boldsymbol{a})} + \boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{A}'(\boldsymbol{a}))^T\Delta\boldsymbol{a}$$

$$= -\boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{\xi}(\boldsymbol{a}) - \boldsymbol{\eta})]_{n+1}^{\nu} \tag{5.57}$$

where $\Delta \boldsymbol{a} = [\Delta a_1, \cdots, \Delta a_N]^T$ and $\boldsymbol{r}(\boldsymbol{a})$ is the residual. The above equation can be re-written as follows:

$$\underbrace{\left(\frac{\partial \boldsymbol{A}'(\boldsymbol{a})}{\partial a_1}\boldsymbol{r}(\boldsymbol{a}), \cdots, \frac{\partial \boldsymbol{A}'(\boldsymbol{a})}{\partial a_N}\boldsymbol{r}(\boldsymbol{a})\right)}_{\bar{\boldsymbol{A}}''(\boldsymbol{a})} \Delta \boldsymbol{a} + \boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{A}'(\boldsymbol{a}))^T \Delta \boldsymbol{a}$$

$$= -\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})]_{n+1}^{\nu} \tag{5.58}$$

which gives:

$$\left(\boldsymbol{A}''(\boldsymbol{a}) + \boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{A}'(\boldsymbol{a}))^T\right) \Delta \boldsymbol{a} = -\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})]_{n+1}^{\nu} \tag{5.59}$$

The above equation (5.59) can be solved for $\Delta \boldsymbol{a}$ in each iteration of the load step. The tangent operator obtained here is the exact tangent.

The expression for the Jacobian matrix $\boldsymbol{J}$ is given by:

$$(\boldsymbol{A}'(\boldsymbol{a}))^T = \boldsymbol{J}(\boldsymbol{a}) = \left[\tfrac{\partial \boldsymbol{\xi}(\boldsymbol{a})}{\partial a_1}, \ldots, \tfrac{\partial \boldsymbol{\xi}(\boldsymbol{a})}{\partial a_N}\right] \equiv \frac{\partial \boldsymbol{\xi}(\boldsymbol{a})}{\partial \boldsymbol{a}} \tag{5.60}$$

Here

$$\frac{\partial \boldsymbol{\xi}(\boldsymbol{a})}{\partial \boldsymbol{a}} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{u}^h}\frac{\partial \boldsymbol{u}^h}{\partial \boldsymbol{a}} \tag{5.61}$$

here

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{u}^h} = \left[\tfrac{\partial \boldsymbol{\xi}}{\partial u_1^h}, \ldots, \tfrac{\partial \boldsymbol{\xi}}{\partial u_3^h}\right] \tag{5.62}$$

$$\frac{\partial \boldsymbol{u}^h}{\partial \boldsymbol{a}} = \left[\tfrac{\partial \boldsymbol{u}^h}{\partial a_1}, \ldots, \tfrac{\partial \boldsymbol{u}^h}{\partial a_N}\right] = \begin{bmatrix} \frac{\partial u_1^h}{\partial a_1}, \ldots, \frac{\partial u_1^h}{\partial a_N} \\ \vdots \\ \frac{\partial u_3^h}{\partial a_1}, \ldots, \frac{\partial u_3^h}{\partial a_N} \end{bmatrix} = \boldsymbol{\Psi}^T \tag{5.63}$$

This gives:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{u}^h} = \frac{\partial}{\partial \boldsymbol{u}^h} \begin{bmatrix} \boldsymbol{\xi}_1(\boldsymbol{u}^h) \\ \sqrt{\alpha^h}\boldsymbol{\xi}_2(\boldsymbol{u}^h) \\ \sqrt{\alpha^g}\boldsymbol{\xi}_3(\boldsymbol{u}^h) \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{A}}_1(\boldsymbol{u}^h) \\ \sqrt{\alpha^h}\hat{\boldsymbol{A}}_2(\boldsymbol{u}^h) \\ \sqrt{\alpha^g}\hat{\boldsymbol{A}}_3(\boldsymbol{u}^h) \end{bmatrix} = \hat{\boldsymbol{A}}(\boldsymbol{u}^h) \tag{5.64}$$

with

$$\hat{\boldsymbol{A}}_1(\boldsymbol{u}^h) = \begin{bmatrix} \hat{\boldsymbol{L}}(\boldsymbol{u}^h(\boldsymbol{p}_1)) \\ \vdots \\ \hat{\boldsymbol{L}}(\boldsymbol{u}^h(\boldsymbol{p}_{N_P})) \end{bmatrix}; \hat{\boldsymbol{A}}_2(\boldsymbol{u}^h) = \begin{bmatrix} \hat{\boldsymbol{B}}^h(\boldsymbol{u}^h(\boldsymbol{q}_1)) \\ \vdots \\ \hat{\boldsymbol{B}}^h(\boldsymbol{u}^h(\boldsymbol{q}_{N_Q})) \end{bmatrix}; \hat{\boldsymbol{A}}_3(\boldsymbol{u}^h) = \begin{bmatrix} \hat{\boldsymbol{B}}^g(\boldsymbol{u}^h(\boldsymbol{r}_1)) \\ \vdots \\ \hat{\boldsymbol{B}}^g(\boldsymbol{u}^h(\boldsymbol{r}_{N_R})) \end{bmatrix} \tag{5.65}$$

The the nonlinear matrix operators $\hat{\boldsymbol{L}}(\boldsymbol{u}^h)$, $\hat{\boldsymbol{B}}^{\boldsymbol{h}}(\boldsymbol{u}^h)$ and $\hat{\boldsymbol{B}}^{\boldsymbol{g}}(\boldsymbol{u}^h)$ are given by:

$$\hat{\boldsymbol{L}}(\boldsymbol{u}^h) = \hat{\boldsymbol{L}}^1(\boldsymbol{u}^h) + \hat{\boldsymbol{L}}^2(\boldsymbol{u}^h) \tag{5.66}$$

with

$$\hat{\boldsymbol{L}}^1(\boldsymbol{u}^h) = \begin{bmatrix} \mathbb{C}^6_{1AjB1C}u^h_{j,BA}D_C & \mathbb{C}^6_{1AjB2C}u^h_{j,BA}D_C & \mathbb{C}^6_{1AjB3C}u^h_{j,BA}D_C \\ \mathbb{C}^6_{2AjB1C}u^h_{j,BA}D_C & \mathbb{C}^6_{2AjB2C}u^h_{j,BA}D_C & \mathbb{C}^6_{2AjB3C}u^h_{j,BA}D_C \\ \mathbb{C}^6_{3AjB1C}u^h_{j,BA}D_C & \mathbb{C}^6_{3AjB2C}u^h_{j,BA}D_C & \mathbb{C}^6_{3AjB3C}u^h_{j,BA}D_C \end{bmatrix} \tag{5.67a}$$

$$\hat{\boldsymbol{L}}^2(\boldsymbol{u}^h) = \begin{bmatrix} \mathbb{C}^4_{1A1B}D^2_{BA} & \mathbb{C}^4_{1A2B}D^2_{BA} & \mathbb{C}^4_{1A3B}D^2_{BA} \\ \mathbb{C}^4_{2A1B}D^2_{BA} & \mathbb{C}^4_{2A2B}D^2_{BA} & \mathbb{C}^4_{2A3B}D^2_{BA} \\ \mathbb{C}^4_{3A1B}D^2_{BA} & \mathbb{C}^4_{3A2B}D^2_{BA} & \mathbb{C}^4_{3A3B}D^2_{BA} \end{bmatrix} \tag{5.67b}$$

$$\hat{\boldsymbol{B}}^h(\boldsymbol{u}^h) = \begin{bmatrix} \mathbb{C}^4_{1A1B}N_AD_B & \mathbb{C}^4_{1A2B}N_AD_B & \mathbb{C}^4_{1A3B}N_AD_B \\ \mathbb{C}^4_{2A1B}N_AD_B & \mathbb{C}^4_{2A2B}N_AD_B & \mathbb{C}^4_{2A3B}N_AD_B \\ \mathbb{C}^4_{3A1B}N_AD_B & \mathbb{C}^4_{3A2B}N_AD_B & \mathbb{C}^4_{3A3B}N_AD_B \end{bmatrix} \tag{5.68}$$

$$\hat{\boldsymbol{B}}^g(\boldsymbol{u}^h) = \boldsymbol{I}_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.69}$$

$$D_A = \frac{\partial}{\partial X_A}; \quad D^2_{AB} = \frac{\partial^2}{\partial X_A \partial X_B} \tag{5.70}$$

The residual vector $\boldsymbol{r}(\boldsymbol{a})$ is given by:

$$\boldsymbol{r}(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{r}_1 \\ \sqrt{\alpha^h}\boldsymbol{r}_2 \\ \sqrt{\alpha^g}\boldsymbol{r}_3 \end{bmatrix} \tag{5.71}$$

with

$$\boldsymbol{r}_1 = \begin{bmatrix} \boldsymbol{R}_1(\boldsymbol{u}^h(\boldsymbol{p}_1)) \\ \vdots \\ \boldsymbol{R}_1(\boldsymbol{u}^h(\boldsymbol{p}_{N_P})) \end{bmatrix}; \boldsymbol{r}_2 = \begin{bmatrix} \boldsymbol{R}_2(\boldsymbol{u}^h(\boldsymbol{q}_1)) \\ \vdots \\ \boldsymbol{R}_2(\boldsymbol{u}^h(\boldsymbol{q}_{N_Q})) \end{bmatrix}; \boldsymbol{r}_3 = \begin{bmatrix} \boldsymbol{R}_3(\boldsymbol{u}^h(\boldsymbol{r}_1)) \\ \vdots \\ \boldsymbol{R}_3(\boldsymbol{u}^h(\boldsymbol{r}_{N_R})) \end{bmatrix} \tag{5.72}$$

and

$$\boldsymbol{R}_1(\boldsymbol{u}^h) = \begin{bmatrix} \sigma_{A1}(\boldsymbol{u}^h)_{,A} + b_1 \\ \sigma_{A2}(\boldsymbol{u}^h)_{,A} + b_2 \\ \sigma_{A3}(\boldsymbol{u}^h)_{,A} + b_3 \end{bmatrix}; \boldsymbol{R}_2(\boldsymbol{u}^h) = \begin{bmatrix} \sigma_{A1}(\boldsymbol{u}^h)N_A - h_1 \\ \sigma_{A2}(\boldsymbol{u}^h)N_A - h_2 \\ \sigma_{A3}(\boldsymbol{u}^h)N_A - h_3 \end{bmatrix}; \boldsymbol{R}_3(\boldsymbol{u}^h) = \begin{bmatrix} u^h_1 - g_1 \\ u^h_2 - g_2 \\ u^h_3 - g_3 \end{bmatrix} \tag{5.73}$$

Here,

$$\mathbb{C}^4_{iAjB} = \frac{\partial^2 W}{\partial F_{iA} \partial F_{jB}}, \tag{5.74a}$$

$$\mathbb{C}^6_{iAjBkC} = \frac{\partial^3 W}{\partial F_{iA} \partial F_{jB} \partial F_{kC}} \tag{5.74b}$$

and here $F_{iA} = u^h_{i,A} + \delta_{iA}$. The expressions for σ_{Ai}, $\mathbb{C}^4_{iAjB}$, $\mathbb{C}^6_{iAjBkC}$ are given in

Appendix C. From equations (5.60) – (5.64) the following equation is obtained:

$$(\boldsymbol{A}'(\boldsymbol{a}))^T = \hat{\boldsymbol{A}}\boldsymbol{\Psi}^T \tag{5.75}$$

5.3.4 RKCM using Gauss Newton method

The term $\boldsymbol{A}''(\boldsymbol{a})$ in equation (5.59) involves the computation of second order derivatives of the residual, which are complicated to derive and involve high computational expense in evaluating them. If the nonlinearity in the problem is not high, these second order derivatives of the residual terms can be omitted which results in the Gauss Newton method given by the following updated equation:

$$\boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{A}'(\boldsymbol{a}))^T\Delta\boldsymbol{a} = -\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})]_{n+1}^{\nu} \tag{5.76}$$

Substituting equation (5.75) in (5.76), the following equation is obtained:

$$\boldsymbol{\Psi}\hat{\boldsymbol{A}}^T\hat{\boldsymbol{A}}\boldsymbol{\Psi}^T\Delta\boldsymbol{a} = -\boldsymbol{\Psi}\hat{\boldsymbol{A}}^T\boldsymbol{r}]_{n+1}^{\nu} \tag{5.77}$$

The Gauss Newton method results in a loss of quadratic rate of convergence in the Newtons method. If the degree of nonlinearity is very high in the problem, the Gauss Newton method will fail to converge and in this case the full Newton method as discussed in previous section needs to be employed.

5.3.5 Convergence criteria

As can be seen from the nonlinear equations in equation (5.55) and the incremental equations in (5.59), as $\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})$ converges $\Delta\boldsymbol{a}$ also converges. Taking this

into account the following two convergence criteria can be used:

$$\|\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})\|_0 < \varepsilon(\|\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})\|_0)_{\nu=1} \tag{5.78a}$$

$$\|\Delta\boldsymbol{a}\|_0 < \varepsilon(\|\boldsymbol{a}\|_0)_{\nu=1} \tag{5.78b}$$

where ε is the specified tolerance.

5.3.6 Derivation from nonlinear least squares functional

The above RKCM formulation can be equivalently derived from the variation and linearization of the weighted least squares functional for the strong form of the governing equations given in (5.27):

$$\begin{aligned}\Pi(u_i; f_i, g_i, h_i) = &\frac{1}{2}\int_{\Omega_X}(\sigma_{Pi,P} + b_i)(\sigma_{Ai,A} + b_i)d\Omega_X \\ &+ \frac{\alpha^h}{2}\int_{\partial\Omega_X^h}(\sigma_{Pi}N_P - h_i)(\sigma_{Ai}N_A - h_i)d\Gamma \\ &+ \frac{\alpha^g}{2}\int_{\partial\Omega_X^g}(u_i - g_i)(u_i - g_i)d\Gamma\end{aligned} \tag{5.79}$$

where α^h and α^g are the square of the weights on the natural and essential boundary conditions, respectively. The variational equation (minimization) is obtained by the stationary condition of the functional (5.79) and is given by:

$$\begin{aligned}\delta\Pi \equiv &\int_{\Omega_X}\delta(\sigma_{Pi,P} + b_i)(\sigma_{Ai,A} + b_i)d\Omega_X + \alpha^h\int_{\partial\Omega_X^h}\delta(\sigma_{Pi}N_P - h_i)(\sigma_{Ai}N_A - h_i)d\Gamma \\ &+ \alpha^g\int_{\partial\Omega_X^g}\delta u_i(u_i - g_i)d\Gamma = 0\end{aligned} \tag{5.80}$$

The above nonlinear equation is solved using the Newton's method, where an incremental equation of motion is considered. This requires linearization of equation (5.80). Let n denote the load step number and ν denote the iteration number. The body force, sur-

face traction, prescribed displacements and deformed state of the body at the $n+1^{\text{th}}$ load step and ν^{th} iteration are known from previous iterations. The linearization is carried out as follows to obtain the incremental displacement, $\Delta \boldsymbol{u}$ defined by, $\boldsymbol{x}_{n+1}^{\nu+1} = \boldsymbol{x}_{n+1}^{\nu} + \Delta \boldsymbol{u}$:

$$\begin{aligned} \delta\Pi_{n+1}^{\nu} + \Delta[\delta\Pi] &= 0 \\ \Rightarrow \Delta[\delta\Pi] &= -\delta\Pi_{n+1}^{\nu} \end{aligned} \tag{5.81}$$

which gives:

$$\begin{aligned} &\int_{\Omega_X} \left[\Delta(\delta(\sigma_{Pi,P} + b_i))(\sigma_{Ai,A} + b_i) + \delta(\sigma_{Pi,P} + b_i)\Delta(\sigma_{Ai,A} + b_i)\right] d\Omega_X \\ &+ \alpha^h \int_{\partial\Omega_X^h} \left[\Delta(\delta(\sigma_{Pi}N_P - h_i))(\sigma_{Ai}N_A - h_i) + \delta(\sigma_{Pi}N_P - h_i)\Delta(\sigma_{Ai}N_A - h_i)\right] d\Gamma \\ &+ \alpha^g \int_{\partial\Omega_X^g} \delta u_i \Delta u_i d\Gamma \\ &= -\Bigg[\int_{\Omega_X} \delta(\sigma_{Pi,P} + b_i)(\sigma_{Ai,A} + b_i) d\Omega_X + \alpha^h \int_{\partial\Omega_X^h} \delta(\sigma_{Pi}N_P - h_i)(\sigma_{Ai}N_A - h_i) d\Gamma \\ &+ \alpha^g \int_{\partial\Omega_X^g} \delta u_i (u_i - g_i) d\Gamma\Bigg]_{n+1}^{\nu} \end{aligned} \tag{5.82}$$

Assuming that the body force and surface traction are independent of deformation, equation (5.82) can be simplified as:

$$\begin{aligned} &\int_{\Omega_X} \left[\Delta(\delta(\sigma_{Pi,P}))(\sigma_{Ai,A} + b_i) + \delta(\sigma_{Pi,P})\Delta(\sigma_{Ai,A})\right] d\Omega_X \\ &+ \alpha^h \int_{\partial\Omega_X^h} \left[\Delta(\delta(\sigma_{Pi}N_P))(\sigma_{Ai}N_A - h_i) + \delta(\sigma_{Pi}N_P)\Delta(\sigma_{Ai}N_A)\right] d\Gamma \\ &+ \alpha^g \int_{\partial\Omega_X^g} \delta u_i \Delta u_i d\Gamma \\ &= -\Bigg[\int_{\Omega_X} \delta(\sigma_{Pi,P})(\sigma_{Ai,A} + b_i) d\Omega_X + \alpha^h \int_{\partial\Omega_X^h} \delta(\sigma_{Pi}N_P)(\sigma_{Ai}N_A - h_i) d\Gamma \\ &+ \alpha^g \int_{\partial\Omega_X^g} \delta u_i (u_i - g_i) d\Gamma\Bigg]_{n+1}^{\nu} \end{aligned} \tag{5.83}$$

The above equation gives the exact tangent operator for the incremental equations. If the nonlinearity in the problem is not high, the Gauss Newton method is employed by omitting the second order derivatives of the residual terms, which results in the following equation:

$$\begin{aligned}
&\int_{\Omega_X} \delta(\sigma_{Pi,P}) \Delta(\sigma_{Ai,A}) d\Omega_X + \alpha^h \int_{\partial\Omega_X^h} \delta(\sigma_{Pi} N_P) \Delta(\sigma_{Ai} N_A) d\Gamma \\
&+ \alpha^g \int_{\partial\Omega_X^g} \delta u_i \Delta u_i d\Gamma \\
&= -\Bigg[\int_{\Omega_X} \delta(\sigma_{Pi,P})(\sigma_{Ai,A} + b_i) d\Omega_X + \alpha^h \int_{\partial\Omega_X^h} \delta(\sigma_{Pi} N_P)(\sigma_{Ai} N_A - h_i) d\Gamma \\
&+ \alpha^g \int_{\partial\Omega_X^g} \delta u_i (u_i - g_i) d\Gamma \Bigg]_{n+1}^{\nu}
\end{aligned} \tag{5.84}$$

The variation and increment terms are given by the following equations:

$$\sigma_{Ai} = \frac{\partial W}{\partial F_{iA}} \tag{5.85}$$

$$\sigma_{Ai,A} = \frac{\partial}{\partial X_A} \left(\frac{\partial W}{\partial F_{iA}} \right) = \mathbb{C}^4_{iAjB} \frac{\partial F_{jB}}{\partial X_A} = \mathbb{C}^4_{iAjB} F_{jB,A} \tag{5.86}$$

$$\delta(\sigma_{Pi,P}) = \mathbb{C}^6_{iP\hat{j}Q\hat{k}R} \delta F_{\hat{k}R} F_{\hat{j}Q,P} + \mathbb{C}^4_{iP\hat{j}Q} \delta F_{\hat{j}Q,P} \tag{5.87}$$

$$\Delta(\sigma_{Ai,A}) = \mathbb{C}^6_{iAjBkC} \Delta F_{kC} F_{jB,A} + \mathbb{C}^4_{iAjB} \Delta F_{jB,A} \tag{5.88}$$

$$\delta(\sigma_{Pi} N_P) = \mathbb{C}^4_{iP\hat{j}Q} N_P \delta F_{\hat{j}Q} \tag{5.89}$$

$$\Delta(\sigma_{Ai} N_A) = \mathbb{C}^4_{iAjB} N_A \Delta F_{jB} \tag{5.90}$$

$$\Delta(\delta(\sigma_{Pi,P})) = \mathbb{C}^8_{iP\hat{j}Q\hat{k}R\hat{l}S} F_{\hat{j}Q,P} \Delta F_{\hat{l}S} \delta F_{\hat{k}R} + \mathbb{C}^6_{iP\hat{j}Q\hat{k}R} \left(\delta F_{\hat{k}R} \Delta F_{\hat{j}Q,P} + \Delta F_{\hat{k}R} \delta F_{\hat{j}Q,P} \right) \tag{5.91}$$

$$\Delta(\delta(\sigma_{Pi} N_P)) = \delta F_{\hat{j}Q} \mathbb{C}^6_{iP\hat{j}Q\hat{k}R} N_P \Delta F_{\hat{k}R} \tag{5.92}$$

Here,

$$\mathbb{C}^8_{iAjBkClD} = \frac{\partial^4 W}{\partial F_{iA} \partial F_{jB} \partial F_{kC} \partial F_{lD}} \tag{5.93}$$

5.3.6.1 Discrete equations

Equation (5.84) can be re-written as follows:

$$\begin{aligned} &\equiv \int_{\Omega_X} \delta \boldsymbol{u}^T \tilde{\boldsymbol{L}}^T \tilde{\boldsymbol{L}} \Delta \boldsymbol{u} \, d\Omega_X + \alpha^h \int_{\partial \Omega^h_X} \delta \boldsymbol{u}^T (\tilde{\boldsymbol{B}}^h)^T \tilde{\boldsymbol{B}}^h \Delta \boldsymbol{u} \, d\Gamma \\ &+ \alpha^g \int_{\partial \Omega^g_X} \delta \boldsymbol{u}^T (\tilde{\boldsymbol{B}}^g)^T \tilde{\boldsymbol{B}}^g \Delta \boldsymbol{u} \, d\Gamma \\ &= -\Bigg[\int_{\Omega_X} \delta \boldsymbol{u}^T \tilde{\boldsymbol{L}}^T \tilde{\boldsymbol{R}}_1 \, d\Omega_X + \alpha^h \int_{\partial \Omega^h_X} \delta \boldsymbol{u}^T (\tilde{\boldsymbol{B}}^h)^T \tilde{\boldsymbol{R}}_2 \, d\Gamma \\ &+ \alpha^g \int_{\partial \Omega^g_X} \delta \boldsymbol{u}^T (\tilde{\boldsymbol{B}}^g)^T \tilde{\boldsymbol{R}}_3 \, d\Gamma \Bigg]^{\nu}_{n+1} \end{aligned} \tag{5.94}$$

where the nonlinear matrix operators $\tilde{\boldsymbol{L}}(\boldsymbol{u})$, $\tilde{\boldsymbol{B}}^h(\boldsymbol{u})$ and $\tilde{\boldsymbol{B}}^g(\boldsymbol{u})$, $\tilde{\boldsymbol{R}}_1(\boldsymbol{u})$, $\tilde{\boldsymbol{R}}_2(\boldsymbol{u})$ and $\tilde{\boldsymbol{R}}_3(\boldsymbol{u})$ are given by:

$$\tilde{\boldsymbol{L}}(\boldsymbol{u}) = \tilde{\boldsymbol{L}}^1(\boldsymbol{u}) + \tilde{\boldsymbol{L}}^2(\boldsymbol{u}) \tag{5.95}$$

with

$$\tilde{\boldsymbol{L}}^1(\boldsymbol{u}) = \begin{bmatrix} \mathbb{C}^6_{1AjB1C}u_{j,BA}D_C & \mathbb{C}^6_{1AjB2C}u_{j,BA}D_C & \mathbb{C}^6_{1AjB3C}u_{j,BA}D_C \\ \mathbb{C}^6_{2AjB1C}u_{j,BA}D_C & \mathbb{C}^6_{2AjB2C}u_{j,BA}D_C & \mathbb{C}^6_{2AjB3C}u_{j,BA}D_C \\ \mathbb{C}^6_{3AjB1C}u_{j,BA}D_C & \mathbb{C}^6_{3AjB2C}u_{j,BA}D_C & \mathbb{C}^6_{3AjB3C}u_{j,BA}D_C \end{bmatrix} \tag{5.96a}$$

$$\tilde{\boldsymbol{L}}^2(\boldsymbol{u}) = \begin{bmatrix} \mathbb{C}^4_{1A1B}D^2_{BA} & \mathbb{C}^4_{1A2B}D^2_{BA} & \mathbb{C}^4_{1A3B}D^2_{BA} \\ \mathbb{C}^4_{2A1B}D^2_{BA} & \mathbb{C}^4_{2A2B}D^2_{BA} & \mathbb{C}^4_{2A3B}D^2_{BA} \\ \mathbb{C}^4_{3A1B}D^2_{BA} & \mathbb{C}^4_{3A2B}D^2_{BA} & \mathbb{C}^4_{3A3B}D^2_{BA} \end{bmatrix} \tag{5.96b}$$

$$\tilde{\boldsymbol{B}}^h(\boldsymbol{u}) = \begin{bmatrix} \mathbb{C}^4_{1A1B}N_AD_B & \mathbb{C}^4_{1A2B}N_AD_B & \mathbb{C}^4_{1A3B}N_AD_B \\ \mathbb{C}^4_{2A1B}N_AD_B & \mathbb{C}^4_{2A2B}N_AD_B & \mathbb{C}^4_{2A3B}N_AD_B \\ \mathbb{C}^4_{3A1B}N_AD_B & \mathbb{C}^4_{3A2B}N_AD_B & \mathbb{C}^4_{3A3B}N_AD_B \end{bmatrix} \tag{5.97}$$

$$\tilde{\boldsymbol{B}}^g(\boldsymbol{u}) = \boldsymbol{I}_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.98}$$

$$\tilde{\boldsymbol{R}}_1(\boldsymbol{u}) = \begin{bmatrix} \sigma_{A1}(\boldsymbol{u})_{,A} + b_1 \\ \sigma_{A2}(\boldsymbol{u})_{,A} + b_2 \\ \sigma_{A3}(\boldsymbol{u})_{,A} + b_3 \end{bmatrix} ;\ \tilde{\boldsymbol{R}}_2(\boldsymbol{u}) = \begin{bmatrix} \sigma_{A1}(\boldsymbol{u})N_A - h_1 \\ \sigma_{A2}(\boldsymbol{u})N_A - h_2 \\ \sigma_{A3}(\boldsymbol{u})N_A - h_3 \end{bmatrix} ;\ \tilde{\boldsymbol{R}}_3(\boldsymbol{u}) = \begin{bmatrix} u_1 - g_1 \\ u_2 - g_2 \\ u_3 - g_3 \end{bmatrix} \tag{5.99}$$

For the strong form collocation method the integrals in equation (5.94) are carried out using quadrature rules where the quadrature points correspond to a set of collocation points. Consider a set of collocation points in the domain and on the essential and natural boundaries as defined in (5.43). Also consider the approximation of $\boldsymbol{u}$ as defined in equation (5.45). Similarly the approximation of the increment of displacement $\Delta\boldsymbol{u}$

denoted by $\Delta \boldsymbol{u}^h$ and the variation of displacement $\delta \boldsymbol{u}$ denoted by $\delta \boldsymbol{u}^h$ are given by:

$$\Delta \boldsymbol{u}^h(\boldsymbol{x}) = \boldsymbol{\Psi}^T \Delta \boldsymbol{a} \tag{5.100a}$$

$$\delta \boldsymbol{u}^h(\boldsymbol{x}) = \boldsymbol{\Psi}^T \delta \boldsymbol{a} \tag{5.100b}$$

where $\Delta \boldsymbol{a}^T = \left\{\Delta \boldsymbol{a}_1^T, \Delta \boldsymbol{a}_2^T, \cdots, \Delta \boldsymbol{a}_{N_S}^T\right\}$ with $\Delta \boldsymbol{a}_I^T = \{\Delta a_{1I}, \Delta a_{2I}, \cdots, \Delta a_{\hat{d}I}\}$ and $\delta \boldsymbol{a}^T = \left\{\delta \boldsymbol{a}_1^T, \delta \boldsymbol{a}_2^T, \cdots, \delta \boldsymbol{a}_{N_S}^T\right\}$ with $\delta \boldsymbol{a}_I^T = \{\delta a_{1I}, \delta a_{2I}, \cdots, \delta a_{\hat{d}I}\}$. Introducing the approximations for $\boldsymbol{u}$, $\delta \boldsymbol{u}$ and $\Delta \boldsymbol{u}$ in equation (5.94) and integrating using the collocation points, results in the following equation:

$$\begin{aligned}
&\sum_{I=1}^{N_P} (\delta \boldsymbol{a}^T \boldsymbol{\Psi}) \tilde{\boldsymbol{L}}^T(\boldsymbol{u}^h(\boldsymbol{p}_I)) \tilde{\boldsymbol{L}}(\boldsymbol{u}^h(\boldsymbol{p}_I)) (\boldsymbol{\Psi}^T \Delta \boldsymbol{a}) \\
&+ \sum_{I=1}^{N_Q} \alpha_I^h (\delta \boldsymbol{a}^T \boldsymbol{\Psi}) \left(\tilde{\boldsymbol{B}}^h\right)^T (\boldsymbol{u}^h(\boldsymbol{q}_I)) \tilde{\boldsymbol{B}}^h(\boldsymbol{u}^h(\boldsymbol{q}_I)) (\boldsymbol{\Psi}^T \Delta \boldsymbol{a}) \\
&+ \sum_{I=1}^{N_R} \alpha_I^g (\delta \boldsymbol{a}^T \boldsymbol{\Psi}) \left(\tilde{\boldsymbol{B}}^g\right)^T (\boldsymbol{u}^h(\boldsymbol{r}_I)) \tilde{\boldsymbol{B}}^g(\boldsymbol{u}^h(\boldsymbol{r}_I)) (\boldsymbol{\Psi}^T \Delta \boldsymbol{a}) \\
&= -\Bigg[\sum_{I=1}^{N_P} (\delta \boldsymbol{a}^T \boldsymbol{\Psi}) \tilde{\boldsymbol{L}}^T(\boldsymbol{u}^h(\boldsymbol{p}_I)) \tilde{\boldsymbol{R}}_1(\boldsymbol{u}^h(\boldsymbol{p}_I)) \\
&+ \sum_{I=1}^{N_Q} \alpha_I^h (\delta \boldsymbol{a}^T \boldsymbol{\Psi}) \left(\tilde{\boldsymbol{B}}^h\right)^T (\boldsymbol{u}^h(\boldsymbol{q}_I) \tilde{\boldsymbol{R}}_2(\boldsymbol{u}^h(\boldsymbol{q}_I)) \\
&+ \sum_{I=1}^{N_R} \alpha_I^g (\delta \boldsymbol{a}^T \boldsymbol{\Psi}) \left(\tilde{\boldsymbol{B}}^g\right)^T (\boldsymbol{u}^h(\boldsymbol{r}_I)) \tilde{\boldsymbol{R}}_3(\boldsymbol{u}^h(\boldsymbol{r}_I)) \Bigg]_{n+1}^{\nu}
\end{aligned} \tag{5.101}$$

where $\boldsymbol{a} = \boldsymbol{a}_{n+1}^{\nu}$. It should be noted that $\tilde{\boldsymbol{L}}(\boldsymbol{u}^h) = \hat{\boldsymbol{L}}(\boldsymbol{u}^h)$, $\tilde{\boldsymbol{B}}^h(\boldsymbol{u}^h) = \hat{\boldsymbol{B}}^h(\boldsymbol{u}^h)$, $\tilde{\boldsymbol{B}}^g(\boldsymbol{u}^h) = \hat{\boldsymbol{B}}^g(\boldsymbol{u}^h)$, $\tilde{\boldsymbol{R}}_1(\boldsymbol{u}^h) = \boldsymbol{R}_1(\boldsymbol{u}^h)$, $\tilde{\boldsymbol{R}}_2(\boldsymbol{u}^h) = \boldsymbol{R}_2(\boldsymbol{u}^h)$, $\tilde{\boldsymbol{R}}_3(\boldsymbol{u}^h) = \boldsymbol{R}_3(\boldsymbol{u}^h)$. Re-arranging in matrix form:

$$\begin{aligned}
&\delta \boldsymbol{a}^T \boldsymbol{\Psi} \left[\hat{\boldsymbol{A}}_1^T \hat{\boldsymbol{A}}_1 + \hat{\boldsymbol{A}}_2^T \hat{\boldsymbol{A}}_2 + \hat{\boldsymbol{A}}_3^T \hat{\boldsymbol{A}}_3\right] \boldsymbol{\Psi}^T \Delta \boldsymbol{a} \\
&= -\delta \boldsymbol{a}^T \boldsymbol{\Psi} \left[\hat{\boldsymbol{A}}_1^T \boldsymbol{r}_1 + \hat{\boldsymbol{A}}_2^T \boldsymbol{r}_2 + \hat{\boldsymbol{A}}_3^T \boldsymbol{r}_3\right]
\end{aligned} \tag{5.102}$$

which gives:

$$\delta \boldsymbol{a}^T \boldsymbol{\Psi} \hat{\boldsymbol{A}}^T \hat{\boldsymbol{A}} \boldsymbol{\Psi}^T \Delta \boldsymbol{a} = -\delta \boldsymbol{a}^T \boldsymbol{\Psi} \hat{\boldsymbol{A}}^T \boldsymbol{r}, \qquad \forall \delta \boldsymbol{a}^T \neq \boldsymbol{0} \tag{5.103}$$

which results in the final set of equations:

$$\boldsymbol{\Psi} \hat{\boldsymbol{A}}^T \hat{\boldsymbol{A}} \boldsymbol{\Psi}^T \Delta \boldsymbol{a} = -\boldsymbol{\Psi} \hat{\boldsymbol{A}}^T \boldsymbol{r} \tag{5.104}$$

where the matrix $\hat{\boldsymbol{A}}$ and vector $\boldsymbol{r}$ are defined in equations (5.64) – (5.65) and (5.71) – (5.72), respectively.

It can be seen that equations (5.104) and (5.77), which give derivations from the nonlinear least squares functional and the nonlinear strong form governing equations, respectively, result in exactly the same system of Gauss Newton equations.

5.3.7 Note on linearization followed by minimization of governing equations

Direct linearization of the strong form equilibrium equations and using least squares method to solve the over-determined system is equivalent to solving nonlinear least squares collocation using Gauss Newton method. This is derived as follows.

Consider the strong form of the governing equations as given in equation (5.27). Linearization of these equations using Newton's method gives:

$$\Delta \sigma_{Ai,A} = -(\sigma_{Ai,A} + b_i)^{\nu}_{n+1} \quad \text{in} \quad \Omega_X \tag{5.105a}$$

$$\Delta (\sigma_{Ai} N_A) = -(\sigma_{Ai} N_A - h_i)^{\nu}_{n+1} \quad \text{on} \quad \partial\Omega^h_X \tag{5.105b}$$

$$\Delta u_i = -(u_i - g_i)^{\nu}_{n+1} \quad \text{on} \quad \partial\Omega^g_X \tag{5.105c}$$

assuming b_i, h_i and g_i are independent of deformation. This gives:

$$\mathbb{C}^6_{iAjBkC} u_{j,BA} \Delta u_{k,C} + \mathbb{C}^4_{iAjB} \Delta u_{j,BA} = -\left(\sigma_{Ai,A} + b_i\right)^{\nu}_{n+1} \quad \text{in} \quad \Omega_X \tag{5.106a}$$

$$\mathbb{C}^4_{iAjB} N_A \Delta u_{j,B} = -\left(\sigma_{Ai} N_A - h_i\right)^{\nu}_{n+1} \quad \text{on} \quad \partial\Omega^h_X \tag{5.106b}$$

$$\Delta u_i = -(u_i - g_i)^{\nu}_{n+1} \quad \text{on} \quad \partial\Omega^g_X \tag{5.106c}$$

The above equations can be re-written as:

$$\begin{bmatrix} \tilde{\boldsymbol{L}}(\boldsymbol{u}) \\ \tilde{\boldsymbol{B}}^h(\boldsymbol{u}) \\ \tilde{\boldsymbol{B}}^g(\boldsymbol{u}) \end{bmatrix} \Delta \boldsymbol{u} = - \begin{bmatrix} \tilde{\boldsymbol{R}}_1(\boldsymbol{u}) \\ \tilde{\boldsymbol{R}}_2(\boldsymbol{u}) \\ \tilde{\boldsymbol{R}}_3(\boldsymbol{u}) \end{bmatrix} \tag{5.107}$$

where the nonlinear matrix operators $\tilde{\boldsymbol{L}}(\boldsymbol{u})$, $\tilde{\boldsymbol{B}}^h(\boldsymbol{u})$, $\tilde{\boldsymbol{B}}^g(\boldsymbol{u})$, $\tilde{\boldsymbol{R}}_1$, $\tilde{\boldsymbol{R}}_2$ and $\tilde{\boldsymbol{R}}_3$ are given in equations (5.95) – (5.99). Consider a set of collocation points as given in equation (5.43). Substituting the approximations for $\boldsymbol{u}$ and $\Delta\boldsymbol{u}$ given in equations (5.45) and (5.100a), respectively, in the above linearized strong form equilibrium equations (5.106) and enforcing the residual to be equal to zero at the collocation points results in the following over-determined system of equations:

$$\hat{\boldsymbol{A}} \boldsymbol{\Psi}^T \Delta \boldsymbol{a} = -\boldsymbol{r} \tag{5.108}$$

where weights are added on the essential and natural boundaries to balance the error between the domain and boundary terms, as discussed in section 5.3.3. The $\hat{\boldsymbol{A}}$ and vector $\boldsymbol{r}$ are defined in equations (5.64) – (5.65) and (5.71) – (5.72), respectively. Solving the above over-determined system of equations using the method of least squares results in equation (5.104) which was derived from the nonlinear least squares functional using Gauss Newton method. This procedure directly gives the approximate tangent in the

incremental equations.

5.3.8 Algorithm for implementing RKCM using the Gauss Newton method

- Input N_S, N_C, material parameters, load increment
- Loop over the collocation points $\boldsymbol{X}_A$
  - Compute the shape functions $\Psi(\boldsymbol{X}_A)$ and the first and second order material derivatives of the shape function at the collocation points $\boldsymbol{X}_A$.
- End loop over collocation points
- Initiate the nodal coordinates $\boldsymbol{a} = \boldsymbol{0}$
- In $(n+1)^{\text{th}}$ load step and $(\nu+1)^{\text{th}}$ iteration: Loop over the collocation points $\boldsymbol{X}_A$
  - Compute $\boldsymbol{u}^h(\boldsymbol{X}_A) = \boldsymbol{\Psi}^T \boldsymbol{a}_{n+1}^{\nu}$ and derivatives of displacement using $\boldsymbol{a}_{n+1}^{\nu}$ from the previous iteration.
  - Using $\boldsymbol{a}_{n+1}^{\nu}$ compute the deformation gradient $\boldsymbol{F}$, right Cauchy Green tensor $\boldsymbol{G}$, invariants of right Cauchy Green tensor I_1, I_2, I_3, reduced invariants $\bar{I}_1$, $\bar{I}_2$ and J at the collocation point.
  - Compute the matrices $\hat{\boldsymbol{A}}_1$, $\hat{\boldsymbol{A}}_2$, $\hat{\boldsymbol{A}}_3$ and $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, $\boldsymbol{r}_3$
  - Estimate the weight on the essential boundary $\sqrt{\alpha^g}$
- End loop over collocation points
- Assemble the matrices to form the over-determined system of equations

  $\hat{\boldsymbol{A}} \boldsymbol{\Psi}^T \Delta \boldsymbol{a} = -\boldsymbol{r}$

- Solve the above equations for $\Delta \boldsymbol{a}$ using the method of least squares or QR least squares and update $\boldsymbol{a}_{n+1}^{\nu+1} = \boldsymbol{a}_{n+1}^{\nu} + \Delta \boldsymbol{a}$.

- Check convergence

- If convergence criteria is satisfied, go to next load step else go to next iteration

5.3.9 Remarks

- Least squares solution of strong form collocation equations followed by Newton's linearization results in giving the exact tangent in the incremental equations.

- Equivalently, variation minimizing of the least squares functional followed with quadrature by Newton's linearization gives the exact tangent in the incremental equations.

- For derivations using both the above methods, the second order derivatives of the residual terms can be omitted if the nonlinearity in the problem is not high. This results in an approximate tangent for the incremental equations and gives the Gauss Newton method of solution.

- Alternatively, collocation of incremental strong form equations followed by least squares method to solve the over-determined system results in an approximate tangent in the Gauss Newton incremental equations. The exact tangent is not obtained in this process.

- For most problems if significant nonlinearity does not exist, the Gauss Newton method can be used for solution. If there exists significant nonlinearity and the Gauss Newton method converges slowly or ceases to converge, the omitted terms need to be added to get the exact tangent in the nonlinear iterations.

Table 5.1: Material constants (in N/cm^2) for cubic hyperelastic model of rubber

A_{10}	A_{20}	A_{30}
37.3	-3.1	0.5

- For image-based modeling using RKCM, the source points and collocation points are chosen from the segmented images. A coarser discretization is chosen for the source points compared to the collocation points which can have the same fine scale discretization as the image pixels. The material and fiber properties are specified at these collocation pixel points.

5.3.10 Numerical examples

In all examples provided in sections 5.3.10.1 and 5.3.10.2, the body force is taken to be zero. RK approximation function with quadratic basis and a rectangular support is used. A support size of $3h$ is used in all directions where h is the nodal spacing of the source points in each of the directions, unless specified otherwise. The weight on the natural boundary $\sqrt{\alpha^h} = 1$ is used in all examples. A cubic polynomial type strain energy density function as given in equation (5.37) is used, with material parameters for rubber model adopted from [125]. They are listed in table 5.1 for convenience. The bulk modulus (K) and the weight on the essential boundary $\sqrt{\alpha^g}$, is specified for each example. The units of Newton (N) and centimeter (cm) are used for force and length, respectively.

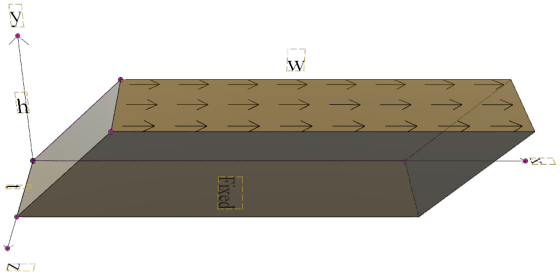


Figure 5.10: 'Near' simple shear deformation of rubber

5.3.10.1 Fundamental test problems

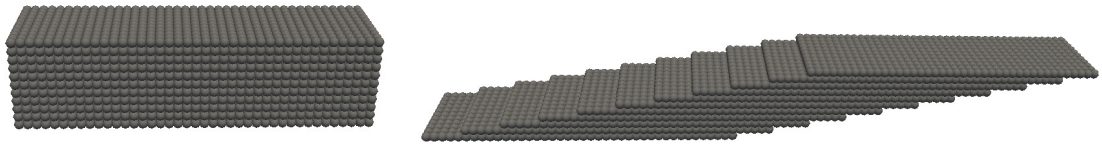
5.3.10.1.1 'Near' simple shear deformation

Simple shear deformation of rubber block in 3D is considered as shown in figure 5.10, with width $w = 4$, height $h = 1$ and thickness $t = 1$. When the width to height ratio (w/h) approaches infinity, rubber deforms in simple shear. In this example the width to height ratio of the rubber block is 4 which generates a 'near' simple shear deformation. The value of bulk modulus used is $K = 1\text{E}8\ N/cm^2$ which makes the material very nearly incompressible. The problem domain is defined by $\bar{\Omega}_X = \{(x, y, z)|0 \leq x \leq 4, 0 \leq y \leq 1, 0 \leq z \leq 1\}$ where x, y and z correspond to w, h and t respectively. The domain is discretized using $9 \times 3 \times 3$ source points and $17 \times 5 \times 5$ collocation points. A total of 425 collocation points in the domain and 290 collocation points on the boundary are used. The essential boundary conditions are prescribed as follows:

$$y = 0: \quad u_x = u_y = u_z = 0 \tag{5.109a}$$

$$y = 1: \quad u_x = \gamma y,\ u_y = u_z = 0 \tag{5.109b}$$

$$z = 0,\ z = 1: \quad u_z = 0 \tag{5.109c}$$



(a) Undeformed configuration (b) Deformed configuration

Figure 5.11: Undeformed and deformed configuration for simple shear problem using RKCM

Here γ is prescribed and is equal to the tangent of the shearing angle. Remaining boundaries are treated as traction free natural boundaries. The out-of-plane displacement on $z = 0$ and $z = 1$ is restricted to simulate plane strain condition. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = N_S\ max(C_{iAjB}(\boldsymbol{X}_A))$. The rubber block is sheared up to 500% engineering shear strain as shown in figure 5.11. The analytical solution for Cauchy shear stress in the xy plane is given by [126]:

$$\sigma_{xy} = 2\gamma \left(\frac{\partial W}{\partial I_1} + \frac{\partial W}{\partial I_2} \right) \tag{5.110}$$

with

$$I_1 = I_2 = 3 + \gamma^2 \tag{5.111}$$

The RKCM Cauchy shear stress at the midpoint of domain is compared with analytical solution in figure 5.12 which shows excellent agreement. In figure 5.13 the Cauchy shear stress distribution in the final deformed state is plotted in the domain and is compared using two methods, RKCM and Reproducing Kernel Particle Method (RKPM). For RKPM which is a weak form based method, 2 point gauss integration (reduced integration for nearly incompressible material) was used to integrate the stiffness matrix and force vector. Additionally for RKPM, the transformation method [54] is used to impose essential boundary conditions. The exact solution in the final deformed state is close to $8198 N/cm^2$. It can be seen that RKCM gives a more accurate solution throughout the

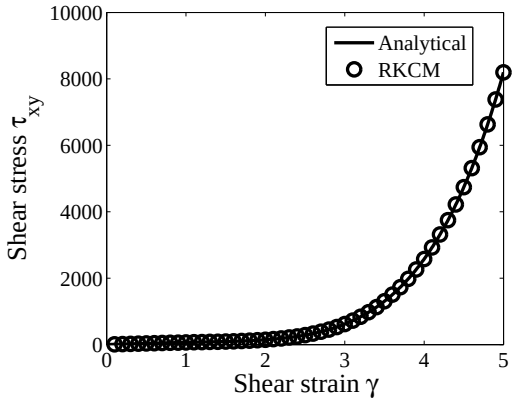
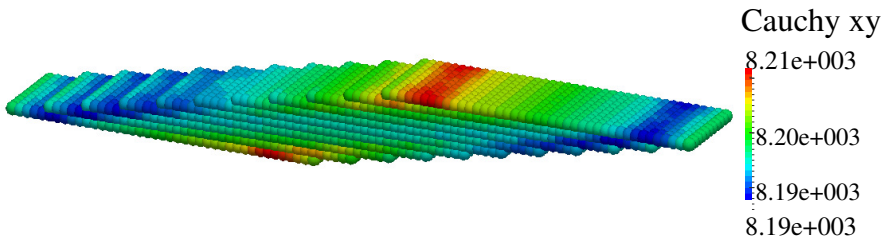


Figure 5.12: Cauchy shear stress (in N/cm^2) comparison for simple shear problem using RKCM

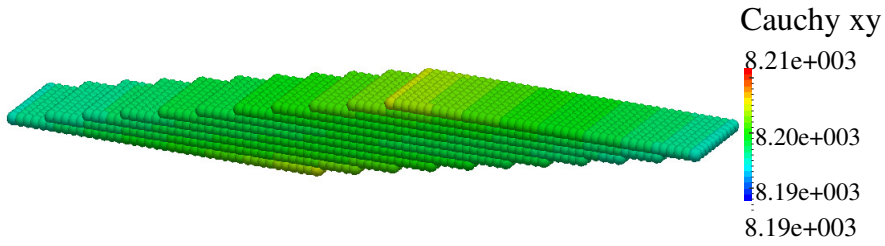
domain and a smoother stress distribution with less oscillations, compared to RKPM.

5.3.10.1.2 Uniaxial tension and compression

Uniaxial tension and compression of a rubber block is considered. A symmetric part of the 3D model is used, with width $w = 4$, height $h = 1$ and thickness $t = 1$. The problem domain is defined by $\bar{\Omega}_X = \{(x, y, z)|0 \leq x \leq 4, 0 \leq y \leq 1, 0 \leq z \leq 1\}$ where x, y and z correspond to w, h and t respectively. A displacement is prescribed on the plane perpendicular to the x–axis which produces an extension or compression of the rubber block as shown in figure 5.14. The value of bulk modulus used is $K =$ 1E8 N/cm^2 which makes the material very nearly incompressible. The same domain discretization for source and collocation points as given in the simple shear problem is used. The essential boundary conditions are prescribed as follows:

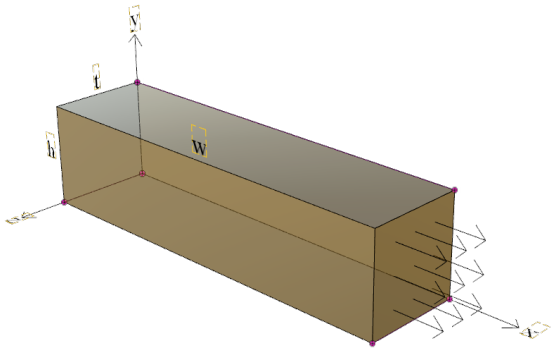


(a) σ_{xy} plotted in domain using RKPM

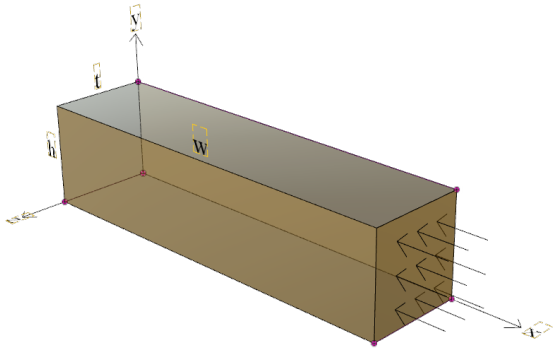


(b) σ_{xy} plotted in domain using RKCM

Figure 5.13: Comparison of Cauchy shear stress distribution using RKPM and RKCM for simple shear problem



(a) Uniaxial tension



(b) Uniaxial compression

Figure 5.14: Uniaxial tension and compression of rubber (symmetric part of the model)

Symmetry boundary conditions:

$$x = 0: \quad u_x = 0 \tag{5.112a}$$

$$y = 0: \quad u_y = 0 \tag{5.112b}$$

$$z = 0: \quad u_z = 0 \tag{5.112c}$$

Loading boundary conditions:

$$x = 4: \quad u_x = g_1; \quad \text{(for Uniaxial tension)} \tag{5.113a}$$

$$x = 4: \quad u_x = -g_2; \quad \text{(for Uniaxial compression)} \tag{5.113b}$$

where g_1 and g_2 are the prescribed displacements in tension and compression respectively. Remaining boundaries are treated as traction free natural boundaries. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = N_S\, max(C_{iAjB}(\boldsymbol{X}_A))$ for the uniaxial tension and $\sqrt{\alpha^g}(\boldsymbol{X}_A) = max(C_{iAjB}(\boldsymbol{X}_A))$ for uniaxial compression. The rubber block is stretched up to 500% for uniaxial tension case and compressed to 90% for the uniaxial compression case as shown in figure 5.15. The analytical solution for Cauchy stress σ_{xx} is given by [127]:

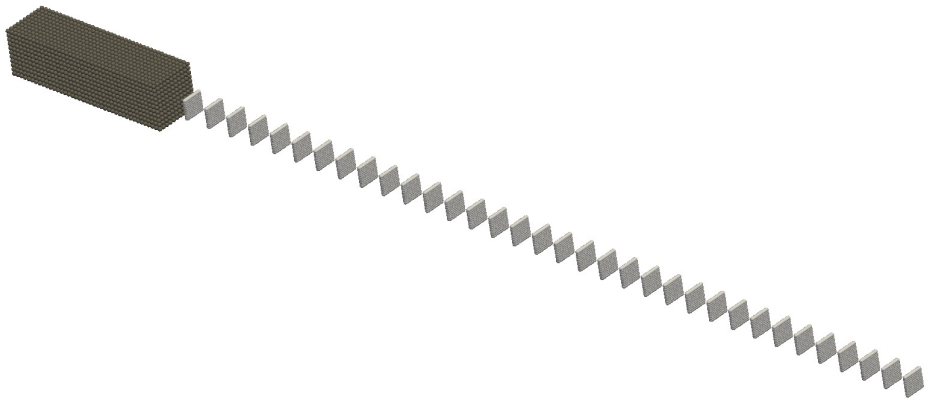
$$\sigma_{xx} = 2\left(\lambda^2 - \lambda^{-1}\right)\left(\frac{\partial W}{\partial I_1} + \lambda^{-1}\frac{\partial W}{\partial I_2}\right) \tag{5.114}$$

with

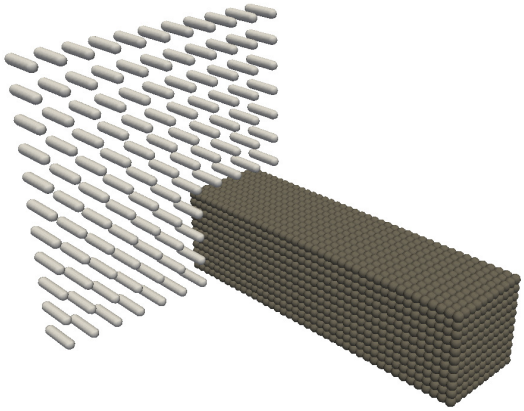
$$I_1 = \lambda^2 + 2\lambda^{-1} \tag{5.115a}$$

$$I_2 = 2\lambda + \lambda^{-2} \tag{5.115b}$$

where λ is the stretch ratio in the x direction. It should be noted that the exact solution

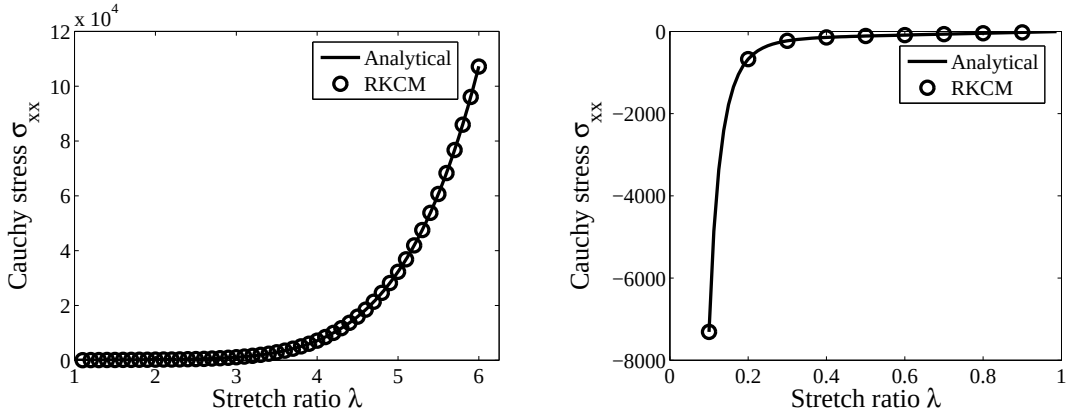


(a) Uniaxial tension



(b) Uniaxial compression

Figure 5.15: Undeformed (dark grey color) and deformed (light grey color) configurations for uniaxial tension and compression using RKCM



(a) Uniaxial tension

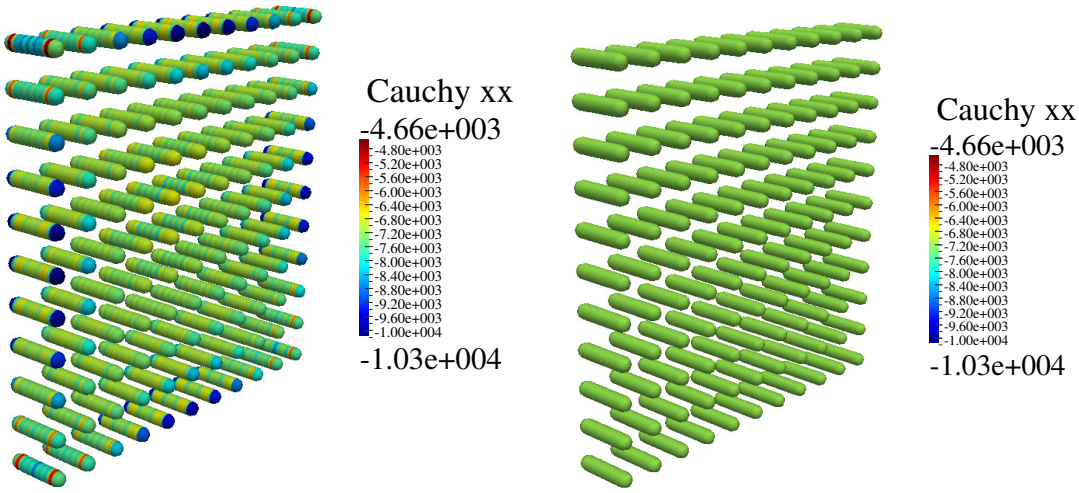
(b) Uniaxial compression

Figure 5.16: Cauchy stress σ_{xx} (in N/cm^2) comparison for uniaxial tension and compression problem using RKCM

corresponds to a plane stress case where a thin sheet of rubber material is considered and the thickness is close to zero. The RKCM Cauchy stress σ_{xx} at the midpoint of domain is compared with analytical solution in figure 5.16 for both the tension and compression cases. Very close agreement between the RKCM and analytical solution is obtained. In figure 5.17 the Cauchy stress distribution in the x-direction in the final deformed state is plotted in the domain for the uniaxial compression case only, for demonstration purpose. It is compared using two methods, RKCM and RKPM. For RKPM, 4 point gauss integration was used to integrate the stiffness matrix and force vector. Additionally for RKPM, the transformation method is used to impose essential boundary conditions. The exact solution in the final deformed state is close to $-7309N/cm^2$. It can be seen that RKCM gives a very accurate solution throughout the domain and also a smoother stress distribution with less oscillations compared to RKPM.

5.3.10.1.3 Equibiaxial deformation

Equibiaxial deformation of a rubber block is considered. A symmetric part of the 3D model is used, with width $w = 4$, height $h = 4$ and thickness $t = 1$. The



(a) σ_{xx} plotted in domain using RKPM

(b) σ_{xx} plotted in domain using RKCM

Figure 5.17: Comparison of Cauchy stress (σ_{xx}) distribution using RKPM and RKCM for uniaxial compression problem

problem domain is defined by $\bar{\Omega}_X = \{(x, y, z)|0 \leq x \leq 4, 0 \leq y \leq 4, 0 \leq z \leq 1\}$ where x, y and z correspond to w, h and t respectively. Displacements are prescribed on planes perpendicular to the x and y axis which produce an extension of the rubber block in these directions as shown in figure 5.18. The value of bulk modulus used is $K = 1\text{E}8\ N/cm^2$ which makes the material very nearly incompressible. The domain is discretized using $9 \times 9 \times 3$ source points and $17 \times 17 \times 5$ collocation points. A total of 1445 collocation points in the domain and 770 collocation points on the boundary are used. The essential boundary conditions are prescribed as follows:

Symmetry boundary conditions:

$$x = 0: \quad u_x = 0 \tag{5.116a}$$

$$y = 0: \quad u_y = 0 \tag{5.116b}$$

$$z = 0: \quad u_z = 0 \tag{5.116c}$$

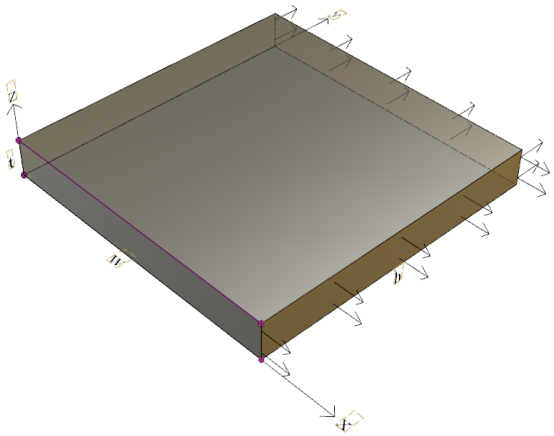


Figure 5.18: Equibiaxial deformation of rubber (symmetric part of the model)

Loading boundary conditions:

$$x = 4: \quad u_x = g \tag{5.117a}$$

$$y = 4: \quad u_y = g \tag{5.117b}$$

where g is the prescribed displacement. Remaining boundaries are treated as traction free natural boundaries. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = max(C_{iAjB}(\boldsymbol{X}_A))$. The rubber block is stretched up to 100% in each of the two directions x and y as shown in figure 5.19. The analytical solution for Cauchy stress is given by [128]:

$$\sigma_{xx} = \sigma_{yy} = \sigma = 2\left(\lambda^2 - \frac{1}{\lambda^4}\right)\left(\frac{\partial W}{\partial I_1} + \lambda^2 \frac{\partial W}{\partial I_2}\right) \tag{5.118}$$

with

$$I_1 = 2\lambda^2 + \lambda^{-4} \tag{5.119a}$$

$$I_2 = \lambda^4 + 2\lambda^{-2} \tag{5.119b}$$

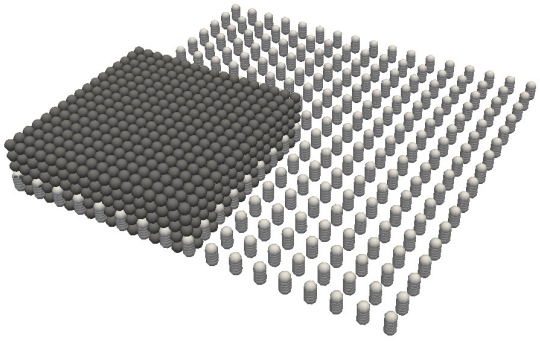


Figure 5.19: Undeformed (dark grey color) and deformed (light grey color) configurations for equibiaxial deformation problem using RKCM

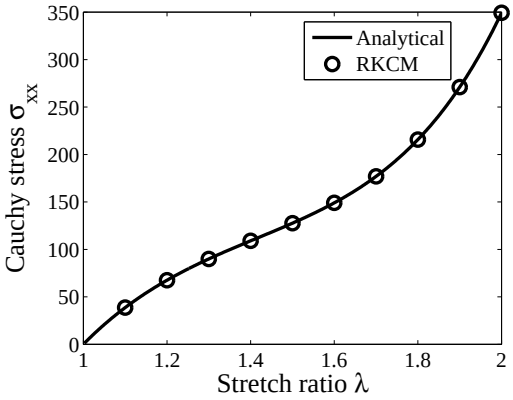


Figure 5.20: Cauchy stress σ_{xx} (in N/cm^2) comparison for equibiaxial deformation problem using RKCM

where $\lambda_1 = \lambda_2 = \lambda$ is the stretch ratio in the x and y directions. It should be noted that the exact solution corresponds to a plane stress case where a thin sheet of rubber material is considered and the thickness is close to zero. The RKCM Cauchy stress σ_{xx} at the midpoint of domain is compared with analytical solution in figure 5.20. Very close agreement between the RKCM and analytical solution is obtained.

5.3.10.1.4 Pure dilation

Pure dilation of a compressible rubber block is considered. The strain energy density considered is decoupled in to distortional and dilatational parts as given in

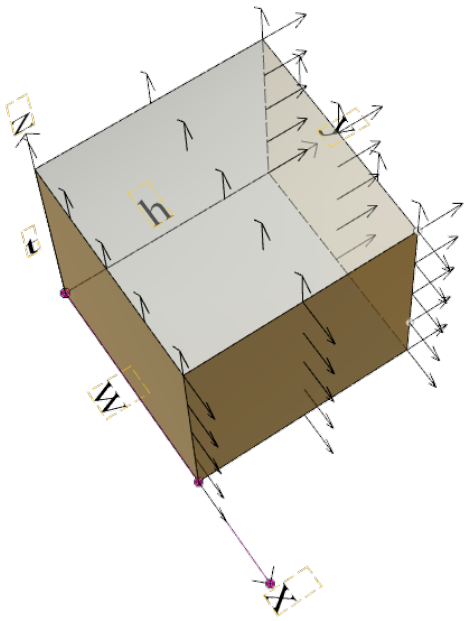


Figure 5.21: Pure dilation of compressible rubber (symmetric part of the model)

section 5.3.2. Under pure dilation, $\bar{I}_1$ and $\bar{I}_2$ are constant and the distortional part of strain energy $\bar{W}(\bar{I}_1, \bar{I}_2) = 0$. The deformation is described only by the dilatational strain energy $\tilde{W}(J)$. A symmetric part of the 3D model is used with width $w = 1$, height $h = 1$ and thickness $t = 1$. The problem domain is defined by $\bar{\Omega}_X = \{(x, y, z)|0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1\}$ where x, y and z correspond to w, h and t respectively. Displacements are prescribed on planes perpendicular to the x, y and z axis which produce an extension of the rubber block in these directions as shown in figure 5.21. The value of bulk modulus used is $K = 250N/cm^2$. The domain is discretized using $6 \times 6 \times 6$ source points and $11 \times 11 \times 11$ collocation points. A total of 1331 collocation points in the domain and 602 collocation points on the boundary are used. The essential boundary conditions are prescribed as follows:

Symmetry boundary conditions:

$$x = 0: \quad u_x = 0 \tag{5.120a}$$

$$y = 0: \quad u_y = 0 \tag{5.120b}$$

$$z = 0: \quad u_z = 0 \tag{5.120c}$$

Loading boundary conditions:

$$x = 1: \quad u_x = g \tag{5.121a}$$

$$y = 1: \quad u_y = g \tag{5.121b}$$

$$z = 1: \quad u_z = g \tag{5.121c}$$

where g is the prescribed displacement. Remaining boundaries are treated as traction free natural boundaries. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = max(C_{iAjB}(\boldsymbol{X}_A))$. The rubber block is stretched up to 500% in each of the three directions x, y and z and the deformed configuration is shown in figure 5.22. The analytical solution for Cauchy stress is given by:

$$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \sigma = K(J-1) \tag{5.122}$$

The RKCM Cauchy stress σ_{xx} at the midpoint of the domain is compared with analytical solution in figure 5.23. Excellent agreement between the RKCM and analytical solution is obtained.

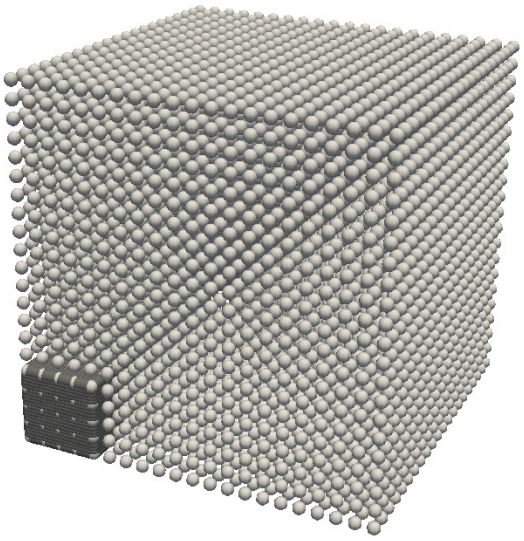


Figure 5.22: Undeformed (dark grey color) and deformed (light grey color) configurations for pure dilation problem using RKCM

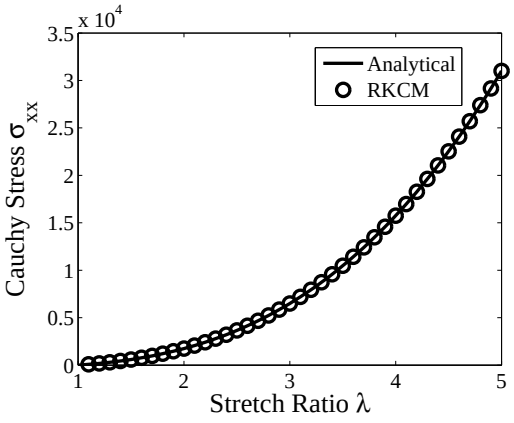


Figure 5.23: Cauchy stress σ_{xx} (in N/cm^2) comparison for pure dilation problem using RKCM

5.3.10.2 Inflation of a rubber tube

Inflation of a rubber tube is considered using 2D plane strain analysis. For this problem the pressure displacement behavior is highly nonlinear. An infinitely long rubber cylinder with inner radius of $R_i = 6$ and outer radius of $R_o = 8$ is subjected to an internal pressure as shown in figure 5.24(a). Using symmetry, a quarter of the geometry

is modeled as shown in figure 5.24(b). Bulk modulus of $K = 1E3N/cm^2$ is used. For this analysis an incremental radial displacement is prescribed and radial Cauchy stress is calculated in each load step. A support size of $4h$ is used for the RK shape function. The domain is discretized using 459 source points, 1717 collocation points in the domain and 232 collocation points on the boundary as shown in figure 5.25. The essential boundary conditions are prescribed as follows:

Symmetry boundary conditions:

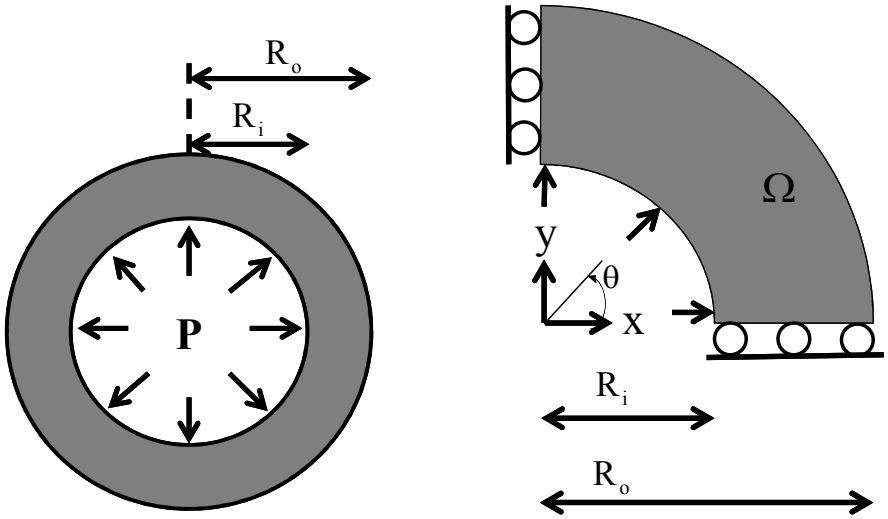
$$x = 0: \quad u_x = 0 \tag{5.123a}$$

$$y = 0: \quad u_y = 0 \tag{5.123b}$$

Loading boundary conditions:

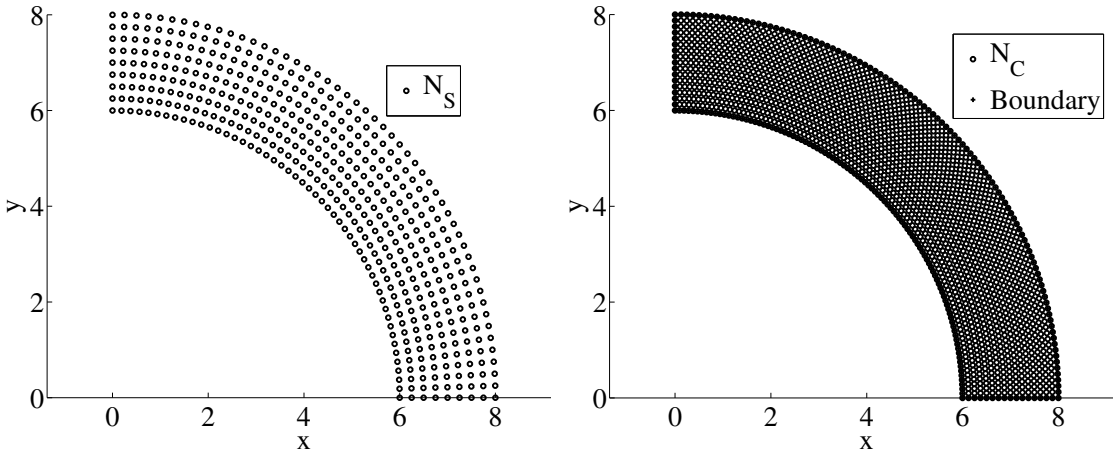
$$r = R_i: \quad u_x = \gamma cos(\theta), \quad u_y = \gamma sin(\theta) \tag{5.124}$$

Here γ is the prescribed radial displacement. Remaining boundaries are treated as traction free natural boundaries. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = N_S\, max(C_{iAjB}(\boldsymbol{X}_A))$. The analytical solution can be found in [125]. The tube is inflated till the inner radius is doubled. The RKCM radial Cauchy stress σ_{rr} at the midpoint of the inner circumference is compared with analytical solution in figure 5.26. Good agreement between the RKCM and analytical solution is obtained. The undeformed and deformed geometry along with the distribution of the radial Cauchy stress is given in figure 5.27.



(a) Problem definition (b) Symmetric model geometry

Figure 5.24: Inflation of a rubber tube problem geometry



(a) Source points (b) Collocation points

Figure 5.25: Inflation of a rubber tube problem discretization

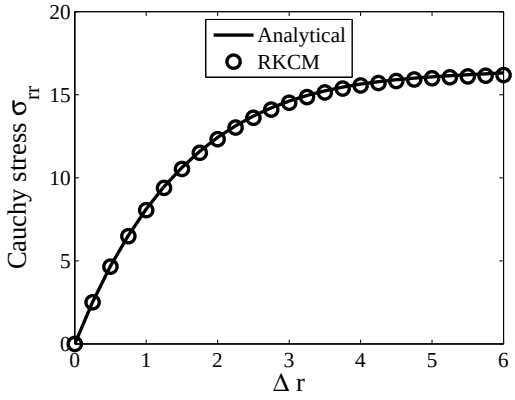


Figure 5.26: Radial Cauchy stress σ_{rr} (in N/cm^2) comparison for inflation of a rubber tube problem using RKCM

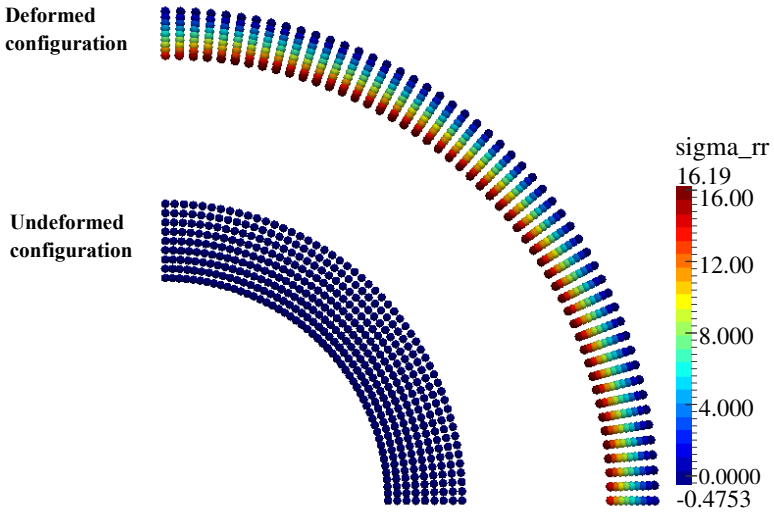


Figure 5.27: Undeformed and deformed configuration with radial Cauchy stress σ_{rr} (in N/cm^2) distribution for inflation of a rubber tube problem

5.3.10.3 Isometric contraction of skeletal muscle

5.3.10.3.1 Validation with RKPM using simple geometry

Isometric contraction of a simple geometry composed of muscle material is considered. A 3D model is used, with width $w = 1$, height $h = 1$ and thickness $t = 2$. The problem domain is defined by $\bar{\Omega}_X = \{(x, y, z)|0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 2\}$ where x, y and z correspond to w, h and t respectively as shown in figure 5.28. The strain energy density for muscle material as described in section 4.2.4 is used for modeling, with material parameters given in table 4.1. The fiber direction at all the collocation points in the muscle is taken to be vertically pointing in the z direction. The ends of the muscle are fixed and the activation factor is incremented to a maximum level in the load steps to simulate isometric contraction of the muscle. The domain is discretized using $9 \times 9 \times 17$ source points and $17 \times 17 \times 33$ collocation points. A total of 9537 collocation points in the domain and 2562 collocation points on the boundary are used. The essential boundary conditions are prescribed as follows:

$$z = 0: \quad u_x = u_y = u_z = 0 \tag{5.125a}$$

$$z = 2: \quad u_x = u_y = u_z = 0 \tag{5.125b}$$

Remaining boundaries are treated as traction free natural boundaries. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = N_S \, max(C_{iAjB}(\boldsymbol{X}_A))$. The undeformed and deformed (scaled 500 times) configurations are shown in figure 5.29. For the purpose of validation RKPM solution was obtained for the same discretization using 8 point gauss quadrature rule for background integration. The transformation method is used for imposing the essential boundary conditions. The RKCM solution for the Cauchy stress σ_{zz} at the midpoint of the domain is compared with the results obtained

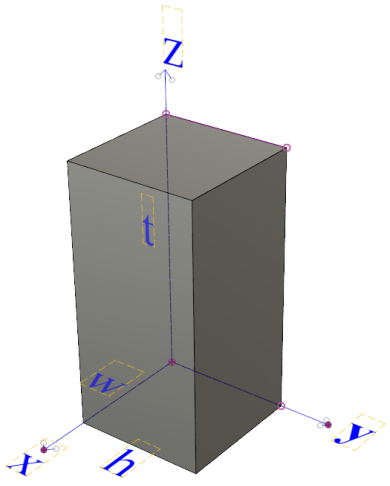
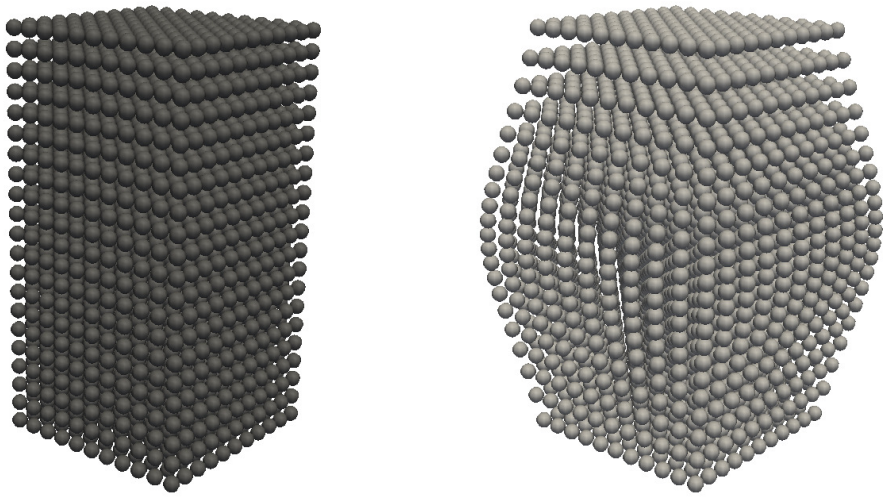


Figure 5.28: Isometric contraction of muscle, simple geometry

from RKPM as shown in figure 5.30(a). The reaction force generated at the fixed end is compared in figure 5.30(b) which shows very good agreement. For RKCM the reaction force is obtained by integrating the traction in the z-direction at the fixed end and for RKPM the reaction force is obtained by summing the internal force in the z-direction at the fixed end.

5.3.10.3.2 Modeling of Medial Gastrocnemius (MG) muscle

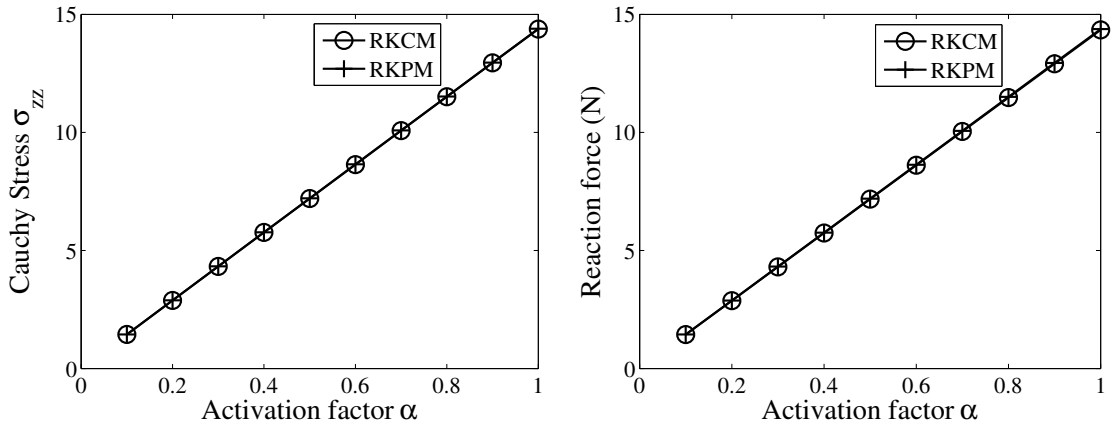
The medial gastrocnemius muscle under isometric contraction is simulated using RKCM. The ends of the muscle are fixed and the activation factor is incremented to a maximum level in the load steps to simulate isometric contraction of the muscle. The entire muscle is considered as muscle material and the same material law as described in the previous example is used. The actual muscle fiber directions as obtained from the DT images are specified at the collocation points. The domain is discretized using 1837 source points, 3727 collocation points in the domain and 1320 collocation points on the boundary as shown in figure 5.31. Sample image slice, showing the source and collocation points is shown in figure 5.32, where coarse models for both source and



(a) Undeformed configuration

(b) Deformed configuration (deformation scaled by 500 times)

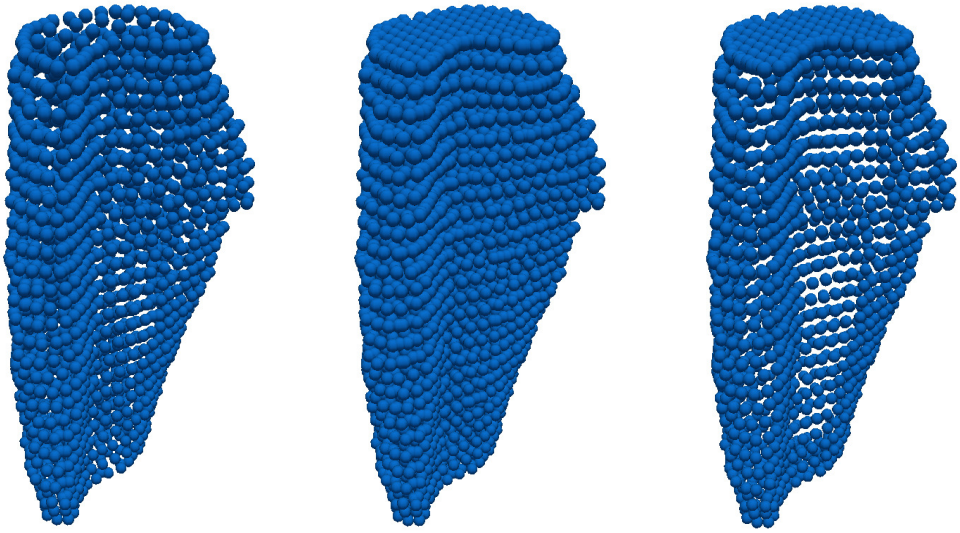
Figure 5.29: Undeformed and deformed configurations for the muscle validation problem



(a) Comparison of Cauchy stress σ_{zz} in N/cm^2 at the mid point of the domain

(b) Comparison of Reaction force at the fixed end

Figure 5.30: Comparison of Cauchy stress σ_{zz} in N/cm^2 and reaction force generated using RKCM and RKPM



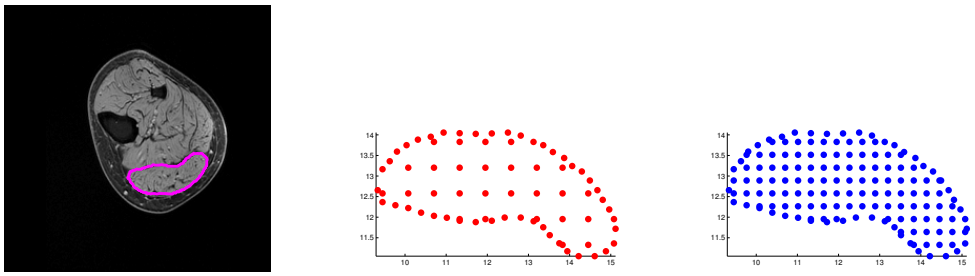
(a) Source points (b) Collocation points (c) Boundary collocation points

Figure 5.31: Source and collocation points for MG muscle under isometric contraction

collocation points are taken from the fine scale MR image. The weight on the essential boundary is taken to be $\sqrt{\alpha^g}(\boldsymbol{X}_A) = N_S\, max(C_{iAjB}(\boldsymbol{X}_A))$. The muscle deformation and distribution of maximum principle Cauchy stress obtained are shown in figure 5.33(a), 5.33(b) and 5.34, respectively. This example demonstrates the ability of RKCM to simulate complicated muscle geometries where the points discretization for domain and boundary is directly input from the segmentation results.

5.3.10.3.3 Effect on muscle force production due to fiber direction

This example demonstrates the effect in muscle force generation under isometric contraction for using anatomically accurate fiber direction from DT images compared with a case where fiber direction is assumed to be in a particular direction. The same GM model as given in the previous example is taken. Two cases are considered as shown in figure 5.35 corresponding to two different fiber orientations. For case 1, the fiber direction at each pixel point is fixed at an inclination close to the orientation assumed

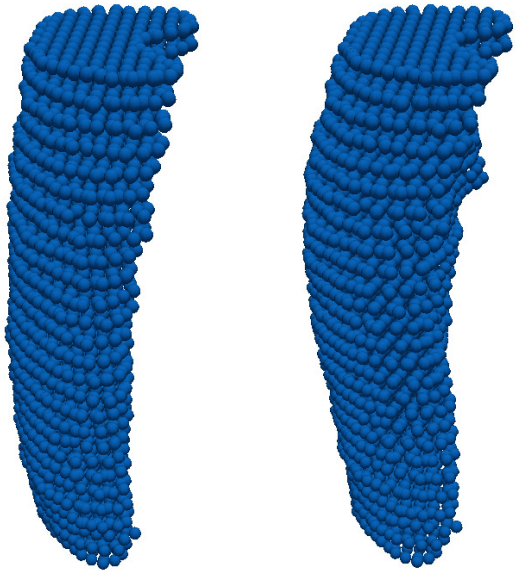


(a) Sample MR image slice with the GM muscle outlined in magenta

(b) Source points for the slice

(c) Collocation points for the slice

Figure 5.32: Source and collocation points on sample slice for MG muscle under isometric contraction



(a) Undeformed configuration

(b) Deformed configuration

Figure 5.33: Deformation of MG muscle under isometric contraction

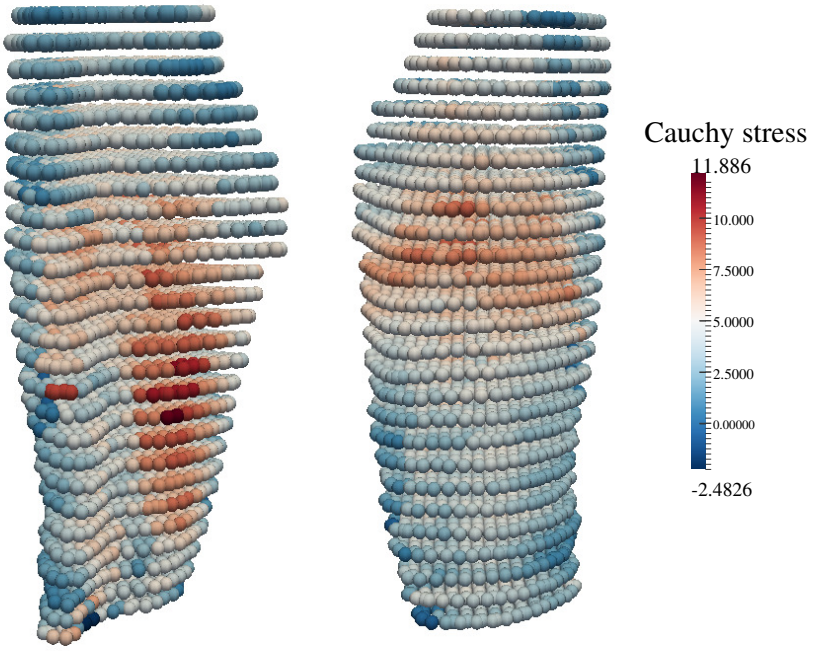
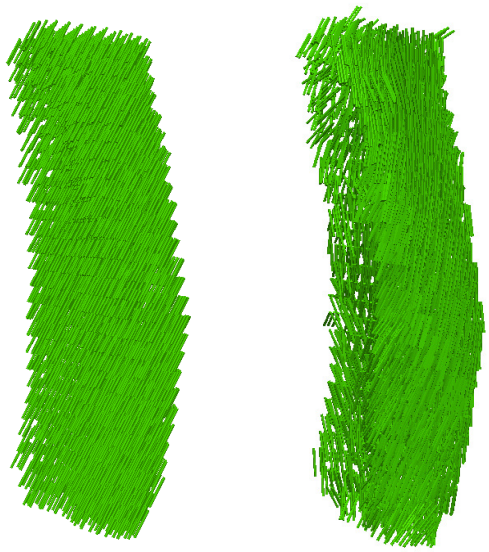


Figure 5.34: Distribution of maximum principal Cauchy stress (in N/cm^2) at maximum isometric contraction, plotted on the deformed configuration

from muscle physiology of GM [129]. Case 2 corresponds to the actual fiber direction at each pixel point obtained from DT imaging. Figure 5.36 shows the comparison of reaction force generated by the muscle for the two cases. Case 1 orientation predicts an increased reaction force, which is 1.69 times greater than the force generated in case 2, at 50% muscle activation. This shows that considerable errors in force production are generated when the fiber direction specified is approximated in modeling. This error could increase as model size increases when multiple muscles are considered.

5.3.11 Performance analysis of RKCM for hyperelasticity

The computational time taken for RKCM for nonlinear analysis in two iterations in a load step is compared with RKPM using the Gauss integration method. For both the methods, quadratic basis is used with a support size of $3h$ for the RK shape function. Consider 2D plane strain analysis of the 'near' simple shear deformation problem as



(a) Case1: Inclined fiber direction

(b) Case2: Actual fiber direction from DT images

Figure 5.35: Fiber directions for the two cases

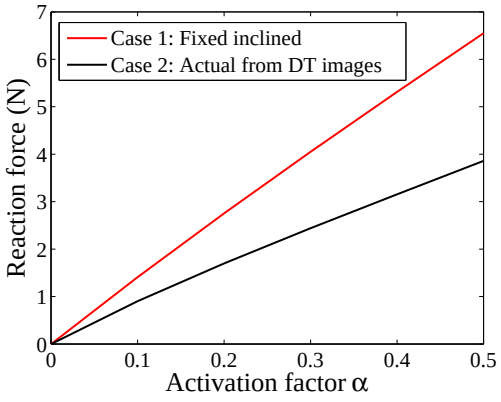


Figure 5.36: Reaction force at the fixed end for MG muscle under isometric contraction for the two cases of fiber orientation

shown in figure 5.37, with width $w = 4$ and height $h = 1$. The value of bulk modulus used is $K = 1E3\ N/cm^2$. The problem domain is defined by $\bar{\Omega}_X = \{(x, y)|0 \leq x \leq 4, 0 \leq y \leq 1\}$ where x and y correspond to the directions of w and h respectively. The essential boundary conditions are prescribed as follows:

$$y = 0: \quad u_x = u_y = 0 \tag{5.126a}$$

$$y = 1: \quad u_x = \gamma y,\ u_y = 0 \tag{5.126b}$$

For RKCM, the domain is discretized using 17×5 source points and 33×9 collocation points. A total of 85 source points, 297 collocation points in the domain and 80 collocation points on the boundary are used. The over-determined system is solved using QR with least squares (LAPACK subroutine). For RKPM the domain is discretized using 17×5 points. 5 point Gauss quadrature rule is used for integrating the stiffness and force matrices accurately and the transformation method is used for imposing the essential boundary conditions. The determined system is solved using LU factorization (LAPACK subroutine). The total CPU time taken to run the 2 iterations (including solver time), CPU time taken for computing the shape functions and its derivatives, CPU time taken for assembling the stiffness and force matrices, and the CPU time taken for solving the systems is given in table 5.2. Additionally a much finer model with 1105 source points, 4257 collocation points in the domain and 320 collocation points on the boundary for RKCM, and 1105 points for RKPM with 5 point Gauss quadrature rule, is analyzed and the results for timings are given in table 5.3. The solution at the mid point of the domain is compared for the 2 methods in figure 5.38 for 100% shear strain, for both the coarse and fine models. It can be seen from both these results that, computationally RKCM requires lesser CPU time for stiffness and force assembly compared to RKPM with Gauss quadrature. Also the result shows that the solver time taken for

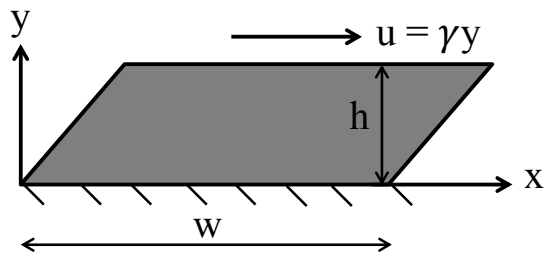


Figure 5.37: 2D 'near' simple shear deformation problem geometry, RKCM performance analysis problem

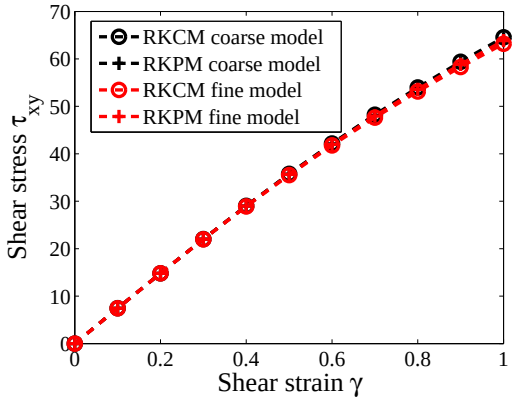


Figure 5.38: Comparison of Cauchy shear stress at the mid point of the domain using RKCM and RKPM for performance analysis problem

RKCM is greater. In the next chapter, reduced order modeling in the RKCM framework is introduced, where this solver time is reduced substantially, giving a very computationally efficient reduced order model for RKCM.

Table 5.2: Performance analysis for RKCM compared to RKPM, for 2D plane strain 'near' simple shear deformation problem, for 2 iterations, degrees of freedom = 170

	RKCM: CPU time (secs)	RKPM: CPU time (secs)
Total CPU time	5.17E-1	5.247E+0
CPU time taken for computing the shape functions and its derivatives	2.16E-1	4.579E-1
CPU time taken for assembling the stiffness matrices and force vectors	1.439E-1	4.675E+0
CPU time taken for solving the system of equations	1.319E-1	9.997E-3

Table 5.3: Performance analysis for RKCM compared to RKPM, for 2D plane strain 'near' simple shear deformation problem, for 2 iterations, degrees of freedom = 2210

	RKCM: CPU time (secs)	RKPM: CPU time (secs)
Total CPU time	8.65E+1	1.148E+3
CPU time taken for computing the shape functions and its derivatives	1.797E+0	5.063E+0
CPU time taken for assembling the stiffness matrices and force vectors	1.455E+0	1.027E+3
CPU time taken for solving the system of equations	8.279E+1	7.597E+0

Chapter 6

Model Order Reduction (MOR) for Linear and Nonlinear RKCM

The main aim of model order reduction is to find a lower dimension approximation of a full model solution by projecting it onto a lower dimensional space. Proper Orthogonal Decomposition (POD) is one of the most popular methods used to construct this projection operator. In this method the proper orthogonal modes of a system of equations are constructed and then these modes are truncated as required to construct the lower dimensional approximation. In this chapter, model reduction for the strong form collocation method (RKCM) is proposed where a Least Squares Galerkin projection is used to project the over-determined system of equations. Examples are provided for both linear and nonlinear problems for the RKCM framework. In the nonlinear case, the method is applied for solving static problems, with incremental loading and where the Gauss Newton method is used to solve the incremental equations. Error analysis is carried out to compare the accuracy of the reduced solution with respect to the full scale solution. As demonstrated in this chapter, the proposed framework provides accurate reduced order models with substantial reduction in time taken for solving.

6.1 Reduced bases from Proper Orthogonal Decomposition (POD)

In this section the detailed derivation for obtaining the reduced order POD bases is given [111, 116]. Consider a function $u(\boldsymbol{x}, t)$ defined on the domain $\boldsymbol{x} \in \mathbb{R}^n$, for which a approximate reduced order function is to be obtained by projection on to a reduced bases (here POD bases). The POD bases are obtained under the following statement of optimality:

Find a bases ϕ such that; the averaged squared error between $u(\boldsymbol{x}, t)$ and its orthogonal projection on to ϕ is minimized, [111]. This is given by the following expression:

$$\underset{\phi \in L_2}{min} \left\langle \left\| u - \frac{(u, \phi)}{\|\phi\|^2} \phi \right\|^2 \right\rangle \tag{6.1}$$

where $(\bullet, \bullet)$ is the inner product defined by $(f, g)_\Omega = \int_\Omega f(\boldsymbol{x}) g^*(\boldsymbol{x}) d\boldsymbol{x}$. Here '*' denotes the complex conjugate. $\| \bullet \| = (\bullet, \bullet)^{\frac{1}{2}}$ is the induced norm and $\langle \bullet \rangle$ is the averaging operator. This is equivalent to maximizing the averaged inner product between u and ϕ suitably normalized [111, 116]:

$$\underset{\phi \in L_2}{max} \frac{\langle |(u, \phi)|^2 \rangle}{\|\phi\|^2} \tag{6.2}$$

subject to the condition $\|\phi\|^2 = 1$. Here $| \bullet |$ denotes the absolute value. The proof for

equivalence of equation (6.1) and (6.2) is given as follows:

$$\begin{aligned}\left\langle \left\| u - \frac{(u,\phi)}{\|\phi\|^2}\phi \right\|^2 \right\rangle &= \left\langle \left(u - \frac{(u,\phi)}{\|\phi\|^2}\phi, u - \frac{(u,\phi)}{\|\phi\|^2}\phi \right) \right\rangle \\ &= \left\langle (u,u) - 2\frac{(u,\phi)}{\|\phi\|^2}(u,\phi) + \frac{(u,\phi)}{\|\phi\|^2}\frac{(u,\phi)}{\|\phi\|^2}\|\phi\|^2 \right\rangle \\ &= \left\langle \|u\|^2 - \frac{|(u,\phi)|^2}{\|\phi\|^2} \right\rangle \\ &= \left\langle \|u\|^2 \right\rangle - \left\langle \frac{|(u,\phi)|^2}{\|\phi\|^2} \right\rangle \end{aligned} \tag{6.3}$$

From the above proof, it can be seen that:

$$\underset{\phi \in L_2}{min} \left\langle \left\| u - \frac{(u,\phi)}{\|\phi\|^2}\phi \right\|^2 \right\rangle \Rightarrow \underset{\phi \in L_2}{max} \frac{\langle |(u,\phi)|^2 \rangle}{\|\phi\|^2}; \quad \|\phi\|^2 = 1 \tag{6.4}$$

For example, consider the case when u represents velocity. Equation (6.2) means that if u is projected along ϕ, the average energy content (kinetic energy) is greater than if u is projected along any other bases function. The maximization problem in equation (6.2) can be recast into a constrained minimization problem with the functional defined by:

$$J[\phi] = \left\langle |(u,\phi)|^2 \right\rangle - \lambda \left(\|\phi\|^2 - 1 \right) \tag{6.5}$$

where λ is the Lagrange multiplier. Using calculus of variations, the necessary condition for extrema is reached when the functional derivative vanishes for all variations of $\phi + \varepsilon\eta \in L_2$. Here $\eta \neq 0$ is an auxillary function and $\varepsilon \in \mathbb{R}$ is a small parameter.

$$\frac{d}{d\varepsilon} J[\phi + \varepsilon\eta] \bigg|_{\varepsilon=0} = 0 \tag{6.6}$$

Using the inner product property; $(f,g) = (g,f)^*$, the above equation can be written as:

$$\begin{aligned}
&\frac{d}{d\varepsilon}\left[\langle (u, \phi+\varepsilon\eta)(\phi+\varepsilon\eta, u)\rangle - \lambda(\phi+\varepsilon\eta, \phi+\varepsilon\eta)\right]\Big|_{\varepsilon=0} = 0 \\
\Rightarrow &2\left[\langle (u,\eta)(\phi,u)\rangle - \lambda(\phi,\eta)\right] = 0 \\
\Rightarrow &\left\langle \int_\Omega u(\boldsymbol{x})\eta^*(\boldsymbol{x})d\boldsymbol{x} \int_\Omega \phi(\boldsymbol{x}')u^*(\boldsymbol{x}')d\boldsymbol{x}' \right\rangle - \lambda \int_\Omega \phi(\boldsymbol{x})\eta^*(\boldsymbol{x})d\boldsymbol{x} = 0
\end{aligned} \tag{6.7}$$

Interchanging the order of the averaging operator and inner product (averaging operator commutes with the integral):

$$\int_\Omega \left[\int_\Omega \langle u(\boldsymbol{x})u^*(\boldsymbol{x}')\rangle \, \phi(\boldsymbol{x}')d\boldsymbol{x}' - \lambda\phi(\boldsymbol{x}) \right] \eta^*(\boldsymbol{x})d\boldsymbol{x} = 0 \tag{6.8}$$

This results in the following Euler–Lagrange equation:

$$\int_\Omega \langle u(\boldsymbol{x})u^*(\boldsymbol{x}')\rangle \, \phi(\boldsymbol{x}')d\boldsymbol{x}' = \lambda\phi(\boldsymbol{x}); \quad \forall \eta^*(\boldsymbol{x}) \neq 0 \tag{6.9}$$

The above equation gives the integral eigenvalue problem. Here $R(\boldsymbol{x},\boldsymbol{x}') = \langle u(\boldsymbol{x})u^*(\boldsymbol{x}')\rangle$ is the averaged auto correlation function. The solution to the integral eigenvalue problem given in equation (6.9), gives a series of eigenvector or eigenfuctions which are called Proper Orthogonal Modes $\phi_i(x)_{i=1}^{\infty}$ (POM's or POD modes) and the corresponding eigenvalues $\lambda_{i=1}^{\infty}$ are called Proper Orthogonal Values (POV's). The energy contained is defined as:

$$E = \sum_{i=1}^{\infty} \lambda_i \tag{6.10}$$

and the energy captured by the i^{th} POM is given by λ_i/E. The reduced order solution to u is obtained by truncating the bases and considering only the first r modes which capture say around 99% of energy. From the mathematical statement of optimality used for constructing the POD bases, it can be seen that on an average, the reduced POD bases

form the 'best' bases as the POD eigenvalues decay rapidly and capture more amount of energy in the first r POD modes, than any other bases.

Finally the approximation of u denoted by u^h is obtained by projecting u on to the reduced POD bases as follows:

$$u(\boldsymbol{x}) \approx u^h(\boldsymbol{x}) = \sum_{i=1}^{r} \phi(\boldsymbol{x}) a_i = \boldsymbol{U}\boldsymbol{a} \tag{6.11}$$

where $\boldsymbol{U} = [\phi_1, \cdots, \phi_r]$ are the reduced bases functions and $\boldsymbol{a} = \{a_1, \cdots, a_r\}$ are unknown coefficients.

6.1.1 POD computation in the discrete case using eigenvalue analysis

In general the continuous governing equations are discretized in space when solving using finite difference, finite element or meshfree methods. For example consider solving a linear elastodynamics problem using the meshfree Reproducing Kernel Particle Method (RKPM), where $\boldsymbol{u}(\boldsymbol{x}, t)$ is the field variable which is the displacement at point $\boldsymbol{x}$ at time t. The Galerkin weak form in $\boldsymbol{u}$ is discretized in space using the meshfree RK approximation to obtain the following matrix equation:

$$\boldsymbol{M}\ddot{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{d} = \boldsymbol{F} \tag{6.12}$$

where, $\boldsymbol{M}$ is the mass matrix of size $n \times n$ where n is the number of degrees of freedom, $\boldsymbol{K}$ is the stiffness matrix of size $n \times n$, $\boldsymbol{F}$ is the force vector of size n and $\boldsymbol{d}$ is the vector of nodal coordinates of size n.

The method of snapshots is used to find the reduced order solution to the above equation [111, 116]. In the off-line phase where the full model simulations are run, a set

of 'S' snapshots of the nodal coordinate vector $\boldsymbol{d} \in \mathbb{R}^n$, at different time intervals are collected in a matrix $\boldsymbol{D}$, which is called the Response matrix and is of size $n \times S$.

$$\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_S] = \begin{bmatrix} d_{11} & \cdots & d_{1S} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nS} \end{bmatrix}_{n \times S} \tag{6.13}$$

In this chapter, when a subscript is used for matrices and vectors, this denotes their size. The discrete averaged auto-correlation matrix $\hat{\boldsymbol{R}}$ is given by:

$$\hat{\boldsymbol{R}} = \frac{1}{S}\boldsymbol{D}\boldsymbol{D}^T = \frac{1}{S}\begin{bmatrix} \sum_{j=1}^{S}(d_{1j})^2 & \cdots & \sum_{j=1}^{S}(d_{1j})(d_{nj}) \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{S}(d_{nj})(d_{1j}) & \cdots & \sum_{j=1}^{S}(d_{nj})^2 \end{bmatrix} \tag{6.14}$$

The eigenvalue problem in the discrete form is given by:

$$\hat{\boldsymbol{R}}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Lambda} \tag{6.15}$$

where

$$\boldsymbol{U} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_n] \tag{6.16}$$

is the matrix containing the POD bases, and

$$\boldsymbol{\Lambda} = diag(\lambda_1, \cdots, \lambda_n) \tag{6.17}$$

is the matrix having the eigenvalues as the diagonal entries. This discrete eigenvalue problem is analogous to the continuous version given in equation (6.9). The above discrete eigenvalue problem given in equation (6.15) can be solved to obtained the POD bases. In equation (6.15), the discrete averaged auto-correlation matrix $\hat{\boldsymbol{R}}$, is a self-

adjoint matrix. The Hilbert-Schmidt theorem states that: 'The eigenvalues of a self-adjoint matrix are real and the eigenvectors corresponding to distinct eigenvalues are orthogonal'. According to this theorem the POD bases vectors are orthogonal.

It can be seen from (6.15) that a $n \times n$ eigenvalue problem needs to be solved to obtain the POD bases, but as this part of the process can be carried out off-line, this computational expense need not be considered. In general if $n >> S$ and if it is required to reduce this off-line computational cost, the method of snapshots as proposed by Sirovich et al. in [113] can be used. In this method the $n \times n$ eigenvalue problem is transformed to a $S \times S$ eigenvalue problem as follows. Since the POD bases and the snapshots span the same space, each eigenvector ϕ can be expressed as a linear combination of the snapshots.

$$\phi = \boldsymbol{D}_{(n\times S)}\, \hat{\boldsymbol{a}}_{(S\times 1)} \tag{6.18}$$

where $\hat{\boldsymbol{a}}^T = \{\hat{a}_1, \cdots, \hat{a}_S\}$ are the unknown coefficients. Substituting this in the discrete eigenvalue problem given in equation (6.15) gives:

$$\begin{aligned}
&\hat{\boldsymbol{R}}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Lambda} \\
\Rightarrow &\frac{1}{S}\boldsymbol{D}\boldsymbol{D}^T\phi = \lambda\phi; \quad (n \times n \text{ eigenvalue problem}) \\
\Rightarrow &\frac{1}{S}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{D}\hat{\boldsymbol{a}} = \lambda\boldsymbol{D}\hat{\boldsymbol{a}} \\
\Rightarrow &\frac{1}{S}\boldsymbol{D}^T\boldsymbol{D}\hat{\boldsymbol{a}} = \lambda\hat{\boldsymbol{a}}; \quad (S \times S \text{ eigenvalue problem})
\end{aligned} \tag{6.19}$$

In order for the above equation to be a necessary condition, the snapshots need to be linearly independent [111]. Provided $\hat{\boldsymbol{a}}$ is scaled to be orthonormal, that is, $\hat{\boldsymbol{a}}^T\hat{\boldsymbol{a}} = \boldsymbol{I}$:

$$\phi^T\phi = \hat{\boldsymbol{a}}^T\boldsymbol{D}^T\boldsymbol{D}\hat{\boldsymbol{a}} = \lambda S \tag{6.20}$$

To get a set of orthonormal bases ϕ divide by:

$$\phi_i = \frac{1}{\sqrt{S\lambda_i}} \boldsymbol{D}\hat{\boldsymbol{a}}_i; \quad i = 1, \cdots, S \tag{6.21}$$

This results in $\phi^T\phi = \boldsymbol{I}$. In this way the first 'S' POD modes can be obtained.

6.1.2 POD computation in the discrete case using Singular Value Decomposition (SVD)

Alternately the POD bases can be equivalently obtained by SVD of the response matrix as follows:

$$\boldsymbol{D}_{(n\times S)} = \boldsymbol{U}_{(n\times n)}\, \boldsymbol{\Sigma}_{(n\times S)}\, \boldsymbol{V}^T_{(S\times S)} \tag{6.22}$$

here $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{S\times S}$ are the orthogonal matrices containing the left singular vectors and right singular vectors of $\boldsymbol{D}$, respectively. The matrix $\boldsymbol{U}$ gives the POD bases. $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times S}$ is a semi positive definite psuedo-diagonal matrix and contains diagonal entries with singular values of $\boldsymbol{D}$, given by σ_i.

$$\boldsymbol{\Sigma}(i,i) = \sigma_i; \quad i = min(n, S) \tag{6.23}$$

and $\sigma_1 > \sigma_2 > \cdots > \sigma_{min(n,S)} \geq 0$.

6.1.3 Equivalence of EVD and SVD

The equivalence between EVD and SVD and the relationship between the eigenvalues and singular values can be obtained from the following equations:

$$\frac{1}{S}\boldsymbol{D}\boldsymbol{D}^T = \frac{1}{S}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\boldsymbol{U}^T = \boldsymbol{U}\ diag(\frac{\sigma_1^2}{S}, \cdots, \frac{\sigma_n^2}{S})\ \boldsymbol{U}^T \tag{6.24}$$

$$\frac{1}{S}\boldsymbol{D}^T\boldsymbol{D} = \frac{1}{S}\boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\ diag(\frac{\sigma_1^2}{S}, \cdots, \frac{\sigma_S^2}{S})\ \boldsymbol{V}^T \tag{6.25}$$

From the above equations, the relationship between the eigenvalues and singular values is given by:

$$\lambda_i = \frac{\sigma_i^2}{S} \tag{6.26}$$

6.2 Galerkin and Petrov-Galerkin projections

Consider the general case of the matrix equation for a linear elastodynamics problem as given in equation (6.12). The reduced order approximation for $\boldsymbol{d}$ denoted by $\boldsymbol{d}^h$ is given by the expression:

$$\boldsymbol{d}^h_{(n\times 1)} = \bar{\boldsymbol{U}}_{(n\times r)}\ \boldsymbol{d}^r_{(r\times 1)} \tag{6.27}$$

where r is the order of reduction with $r << n$, $\bar{\boldsymbol{U}} = [\phi_1, \cdots, \phi_r]$ is the projection matrix containing the reduced set of POD bases and $\boldsymbol{d}^r$ are the reduced nodal coordinates. Substituting equation (6.27) in the system of equations (6.12) gives:

$$\boldsymbol{M}\bar{\boldsymbol{U}}\ddot{\boldsymbol{d}}^r + \boldsymbol{K}\bar{\boldsymbol{U}}\boldsymbol{d}^r - \boldsymbol{F} = \boldsymbol{e} \tag{6.28}$$

where '$\boldsymbol{e}$' is the residual error as the approximation of $\boldsymbol{d}$ is substituted in the matrix equations (6.12). In order to minimize this error, the residual error is constrained to be

orthogonal to a subspace $\mathbb{W}$, defined by the bases $\boldsymbol{W} \in \mathbb{R}^{n \times r}$, which gives:

$$\begin{aligned} \boldsymbol{W}^T \boldsymbol{e} &= \boldsymbol{0} \\ \boldsymbol{W}^T \boldsymbol{M} \bar{\boldsymbol{U}} \ddot{\boldsymbol{d}}^r + \boldsymbol{W}^T \boldsymbol{K} \bar{\boldsymbol{U}} \boldsymbol{d}^r - \boldsymbol{W}^T \boldsymbol{F} &= \boldsymbol{0} \end{aligned} \tag{6.29}$$

For the case of Galerkin projection, $\boldsymbol{W} = \bar{\boldsymbol{U}}$ and in the case of a Petrov-Galerkin projection, $\boldsymbol{W} \neq \bar{\boldsymbol{U}}$.

Finally the reduced system of equations are given by:

$$\hat{\boldsymbol{M}} \ddot{\boldsymbol{d}}^r + \hat{\boldsymbol{K}} \boldsymbol{d}^r = \hat{\boldsymbol{F}} \tag{6.30}$$

where

$$\hat{\boldsymbol{M}}_{(r \times r)} = \boldsymbol{W}^T_{(r \times n)} \, \boldsymbol{M}_{(n \times n)} \, \bar{\boldsymbol{U}}_{(n \times r)} \tag{6.31a}$$

$$\hat{\boldsymbol{K}}_{(r \times r)} = \boldsymbol{W}^T_{(r \times n)} \, \boldsymbol{K}_{(n \times n)} \, \bar{\boldsymbol{U}}_{(n \times r)} \tag{6.31b}$$

$$\hat{\boldsymbol{F}}_{(r \times 1)} = \boldsymbol{W}^T_{(r \times n)} \, \boldsymbol{F}_{(n \times 1)} \tag{6.31c}$$

The matrices $\hat{\boldsymbol{M}}_{(r \times r)}$ and $\hat{\boldsymbol{K}}_{(r \times r)}$ are usually fully populated. POD transforms large sparse matrices to small dense systems. Equation (6.30) gives the reduced system with r degrees of freedom and can be solved for $\boldsymbol{d}^r$. The reduced solution $\boldsymbol{d}^h$ can be obtained from equation (6.27). It can be seen that when $r << n$, the computational cost for solving the system is greatly reduced. For discrete systems, the number of POD modes to be considered for the reduced system can be chosen such that the energy given by:

$$E = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{6.32}$$

is greater close to 1. The critical part in model order reduction using POD is choosing

the correct snapshots which capture the response of the full model which is essential to give results close to the full scale model.

6.3 Model order reduction using least squares Galerkin projection for RKCM

In the meshfree Reproducing Kernel Collocation Method (RKCM) for linear problems and for nonlinear problems using Gauss Newton method as described in section 5.3.4, result in solving an over-determined system of equations. In case of nonlinear problems an over-determined system has to be solved in every iteration of a load step. This over-determined system of equations cannot be projected using a standard Galerkin projection, as this projection is not compatible. This will be illustrated in sections 6.3.1 and 6.3.2 for linear and nonlinear problems, respectively. To overcome this, the Least Squares Galerkin (LSG) projection is proposed to be used in conjunction with RKCM to form the reduced set of equations in the RKCM framework. Using the LSG projection the bases $\boldsymbol{W}$ is chosen to be $\boldsymbol{K}_{(n\times n)}\ \bar{\boldsymbol{U}}_{(n\times r)}$. This method is detailed in the following two sections for linear and nonlinear problems, respectively, with examples and error estimates to demonstrate the effectiveness of the method.

6.3.1 Linear problems

Consider the following general linear elastodynamics problem for RKCM.

$$\alpha\ddot{\boldsymbol{u}}(\boldsymbol{x},t) + \boldsymbol{L}\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{f}(\boldsymbol{x},t); \quad \boldsymbol{x}\in\Omega,\ t\in(0,T] \tag{6.33}$$

with boundary conditions,

$$\boldsymbol{B}^h\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{h}(\boldsymbol{x},t); \quad \boldsymbol{x} \in \partial\Omega^h,\ t \in (0,T] \tag{6.34a}$$

$$\boldsymbol{B}^g\boldsymbol{u}(\boldsymbol{x},t) = \bar{\boldsymbol{u}}(\boldsymbol{x},t); \quad \boldsymbol{x} \in \partial\Omega^g,\ t \in (0,T] \tag{6.34b}$$

and initial conditions

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}^0(\boldsymbol{x}); \quad \boldsymbol{x} \in \bar{\Omega} \tag{6.35a}$$

$$\dot{\boldsymbol{u}}(\boldsymbol{x},0) = \boldsymbol{v}^0(\boldsymbol{x}); \quad \boldsymbol{x} \in \bar{\Omega} \tag{6.35b}$$

where $\boldsymbol{L}$, $\boldsymbol{B}^h$ and $\boldsymbol{B}^g$ are the spacial differential operators in the domain, on the natural and essential boundary, respectively. $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{f}$, $\boldsymbol{h}$ and $\bar{\boldsymbol{u}}$ are the body force, applied boundary traction vector and prescribed displacement vector, respectively. α is the density. $\boldsymbol{u}^0$ is the initial displacement vector, $\boldsymbol{v}^0$ is the initial velocity vector and $\bar{\Omega} = \Omega \cup \partial\Omega^h \cup \partial\Omega^g$ represents the entire problem domain.

The temporal discretization is carried out using the central difference scheme as follows:

$$\alpha(\boldsymbol{u}_{n+1}(\boldsymbol{x}) - 2\boldsymbol{u}_n(\boldsymbol{x}) + \boldsymbol{u}_{n-1}(\boldsymbol{x})) + \Delta t^2 \boldsymbol{L}\boldsymbol{u}_n(\boldsymbol{x}) = \Delta t^2 \underbrace{\boldsymbol{f}(\boldsymbol{x},t_n)}_{\boldsymbol{f}_n(\boldsymbol{x})}; \quad \boldsymbol{x} \in \Omega \tag{6.36a}$$

$$\boldsymbol{B}^h\boldsymbol{u}_{n+1}(\boldsymbol{x}) = \underbrace{\boldsymbol{h}(\boldsymbol{x},t_{n+1})}_{\boldsymbol{h}_{n+1}(\boldsymbol{x})}; \quad \boldsymbol{x} \in \partial\Omega^h \tag{6.36b}$$

$$\boldsymbol{B}^g\boldsymbol{u}_{n+1}(\boldsymbol{x}) = \underbrace{\bar{\boldsymbol{u}}(\boldsymbol{x},t_{n+1})}_{\bar{\boldsymbol{u}}_{n+1}(\boldsymbol{x})}; \quad \boldsymbol{x} \in \partial\Omega^g \tag{6.36c}$$

$$\boldsymbol{u}_0(\boldsymbol{x}) = \boldsymbol{u}^0(\boldsymbol{x}); \quad \boldsymbol{x} \in \bar{\Omega} \tag{6.36d}$$

$$\boldsymbol{u}_1(\boldsymbol{x}) - \boldsymbol{u}_0(\boldsymbol{x}) = \Delta t \boldsymbol{v}^0(\boldsymbol{x}); \quad \boldsymbol{x} \in \bar{\Omega} \tag{6.36e}$$

Here n is the time step counter and $\boldsymbol{u}_n(x) = \boldsymbol{u}(\boldsymbol{x},t_n)$. For the spatial discretization,

$\boldsymbol{u}_n(\boldsymbol{x})$ is discretized using the Reproducing Kernel (RK) shape function as follows:

$$\boldsymbol{u}_n(\boldsymbol{x}) \approx \boldsymbol{u}_n^h(\boldsymbol{x}) = \boldsymbol{\Psi}^T(\boldsymbol{x})\boldsymbol{a}_n \tag{6.37}$$

where $\boldsymbol{\Psi}^T = [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \cdots, \boldsymbol{\Psi}_{N_S}]$ with $\boldsymbol{\Psi}_I = \Psi_I \boldsymbol{I}_{\hat{d}\times\hat{d}}$ and $\boldsymbol{a}_n^T = \left\{(\boldsymbol{a}_1^T)_n, (\boldsymbol{a}_2^T)_n, \cdots, (\boldsymbol{a}_{N_S}^T)_n\right\}$ with $(\boldsymbol{a}_I^T)_n = \{(a_{1I})_n, (a_{2I})_n, \cdots, (a_{\hat{d}I})_n\}$. Here Ψ_I represents the RK shape function at node 'I'. The discretization points for constructing the approximation of $\boldsymbol{u}_n$ are the source points N_S. $\boldsymbol{I}_{\hat{d}\times\hat{d}}$ is the identity matrix of size $\hat{d} \times \hat{d}$ where $\boldsymbol{u}_n \in \mathbb{R}^{\hat{d}}$. Consider a set of collocation points as given in equations (5.9) and (5.10). The approximation of $\boldsymbol{u}_n$ given in equation (6.37) is substituted in the strong form of the governing equations (6.36) and the residual is enforced to be zero at the collocation points which leads to an over-determined system of equations given by:

$$\boldsymbol{M}\boldsymbol{a}_{n+1} = \underbrace{2\boldsymbol{M}\boldsymbol{a}_n - \boldsymbol{M}\boldsymbol{a}_{n-1} - \Delta t^2 \boldsymbol{A}^1 \boldsymbol{a}_n + \Delta t^2 \boldsymbol{F}_n}_{\boldsymbol{b}^1_{n,n-1}} \tag{6.38a}$$

$$\boldsymbol{A}^2 \boldsymbol{a}_{n+1} = \boldsymbol{b}^2_{n+1} \tag{6.38b}$$

$$\boldsymbol{A}^3 \boldsymbol{a}_{n+1} = \boldsymbol{b}^3_{n+1} \tag{6.38c}$$

where

$$\boldsymbol{M} = \begin{bmatrix} \alpha\boldsymbol{\Psi}(\boldsymbol{p}_1) \\ \vdots \\ \alpha\boldsymbol{\Psi}(\boldsymbol{p}_{N_P}) \end{bmatrix}; \quad \boldsymbol{A}^1 = \begin{bmatrix} \boldsymbol{L}\boldsymbol{\Psi}(\boldsymbol{p}_1) \\ \vdots \\ \boldsymbol{L}\boldsymbol{\Psi}(\boldsymbol{p}_{N_P}) \end{bmatrix}; \quad \boldsymbol{F}_n(t) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{p}_1, t_n) \\ \vdots \\ \boldsymbol{f}(\boldsymbol{p}_{N_P}, t_n) \end{bmatrix} \tag{6.39a}$$

$$\boldsymbol{A}^2 = \begin{bmatrix} \boldsymbol{B}^h\boldsymbol{\Psi}(\boldsymbol{q}_1) \\ \vdots \\ \boldsymbol{B}^h\boldsymbol{\Psi}(\boldsymbol{q}_{N_Q}) \end{bmatrix}; \quad \boldsymbol{b}^2_{n+1} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{q}_1, t_{n+1}) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{q}_{N_Q}, t_{n+1}) \end{bmatrix} \tag{6.39b}$$

$$\boldsymbol{A}^3 = \begin{bmatrix} \boldsymbol{B}^g\boldsymbol{\Psi}(\boldsymbol{r}_1) \\ \vdots \\ \boldsymbol{B}^g\boldsymbol{\Psi}(\boldsymbol{r}_{N_R}) \end{bmatrix}; \quad \boldsymbol{b}^3_{n+1} = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{r}_1, t_{n+1}) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{r}_{N_R}, t_{n+1}) \end{bmatrix} \tag{6.39c}$$

Re-writing equations (6.38) with added weights on the essential and natural boundaries:

$$\underbrace{\begin{bmatrix} \boldsymbol{M} \\ \sqrt{\beta^h}\boldsymbol{A}^2 \\ \sqrt{\beta^g}\boldsymbol{A}^3 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{a}_{n+1} = \underbrace{\begin{bmatrix} \boldsymbol{b}^1_{n,n-1} \\ \sqrt{\beta^h}\boldsymbol{b}^2_{n+1} \\ \sqrt{\beta^g}\boldsymbol{b}^3_{n+1} \end{bmatrix}}_{\boldsymbol{b}} \tag{6.40}$$

here, $\sqrt{\beta^h}$ and $\sqrt{\beta^g}$ are the weights to be applied on the natural and essential boundaries, respectively, to balance the error between the domain and the boundary terms. They are given by [130]:

$$\sqrt{\beta^h} = \mathcal{O}(\kappa^{-1}N_S^{-2}\sigma) \tag{6.41a}$$

$$\sqrt{\beta^g} = \mathcal{O}(N_S^{-1}\sigma) \tag{6.41b}$$

where κ is the material constant and $\sigma = max(\alpha, \Delta t^2, N_S^2)$. The initial conditions are

given by:

$$\boldsymbol{H}\boldsymbol{a}_0 = \bar{\boldsymbol{u}}_0 \tag{6.42a}$$

$$\boldsymbol{H}(\boldsymbol{a}_1 - \boldsymbol{a}_0) = \Delta t \bar{\boldsymbol{v}}_0 \Rightarrow \boldsymbol{H}\boldsymbol{a}_1 = \Delta t \bar{\boldsymbol{v}} + \boldsymbol{H}\boldsymbol{a}_0 \tag{6.42b}$$

where the matrices are given by:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{\Psi}^T(\boldsymbol{\chi}_1) \\ \vdots \\ \boldsymbol{\Psi}^T(\boldsymbol{\chi}_{N_C}) \end{bmatrix}; \quad \bar{\boldsymbol{u}}_0 = \begin{bmatrix} \boldsymbol{u}_0(\boldsymbol{\chi}_1) \\ \vdots \\ \boldsymbol{u}_0(\boldsymbol{\chi}_{N_C}) \end{bmatrix}; \quad \bar{\boldsymbol{v}}_0 = \begin{bmatrix} \boldsymbol{v}_0(\boldsymbol{\chi}_1) \\ \vdots \\ \boldsymbol{v}_0(\boldsymbol{\chi}_{N_C}) \end{bmatrix} \tag{6.43}$$

where $\boldsymbol{\chi} = [\boldsymbol{\chi}_1, \cdots, \boldsymbol{\chi}_{N_C}] = \left[\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{N_P}, \boldsymbol{q}_1, \cdots, \boldsymbol{q}_{N_Q}, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{N_R}\right]$ are the collocation points and $N_C = N_P + N_Q + N_R$. The over-determined system of equations (6.40) can be solved using the method of least squares.

Model order reduction using Least Squares Galerkin projection:

The over-determined system of equations $\boldsymbol{A}\boldsymbol{a}_{n+1} = \boldsymbol{b}$ are reduced as follows. Consider a one-dimensional case where the over-determined system matrices have the following size:

$$\boldsymbol{A}_{(N_C \times N_S)} \, (\boldsymbol{a}_{n+1})_{(N_S,1)} = (\boldsymbol{b})_{(N_C,1)} \tag{6.44}$$

The reduced approximation of $\boldsymbol{a}$ is given by:

$$\boldsymbol{a} \approx \boldsymbol{a}^h_{(N_S \times 1)} = \bar{\boldsymbol{U}}_{(N_S \times r)} \boldsymbol{a}^r_{(r \times 1)} \tag{6.45}$$

where $\bar{\boldsymbol{U}} = [\phi_1, \cdots, \phi_r]$ is the reduced bases obtained by truncating POD modes of the response matrix and r is the number of reduced modes or the reduced degrees of freedom for solving the reduced system. $\boldsymbol{a}^r$ are the unknown coefficients in the reduced

system. Substituting the reduced approximation (6.45) in (6.44) results in the equation:

$$\boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)} \boldsymbol{a}^r_{(r \times 1)} - (\boldsymbol{b})_{(N_C,1)} = \boldsymbol{e} \tag{6.46}$$

where $\boldsymbol{e}$ is the residual error. To minimize this error, the residual error is constrained to be orthogonal to a subspace $\mathbb{W}$, defined by the bases $\boldsymbol{W} \in \mathbb{R}^{N_C \times r}$, which gives:

$$\begin{aligned} &\boldsymbol{W}^T \boldsymbol{e} = \boldsymbol{0} \\ \Rightarrow &\boldsymbol{W}^T_{(r \times N_C)} \boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)} \boldsymbol{a}^r_{(r \times 1)} = \boldsymbol{W}^T_{(r \times N_C)} (\boldsymbol{b})_{(N_C,1)} \end{aligned} \tag{6.47}$$

It can be seen that the bases $\boldsymbol{W}$ should belong to $\mathbb{R}^{N_C \times r}$ for projecting the RKCM equations. A projection using $\boldsymbol{W} = \bar{\boldsymbol{U}}$ is not possible since in this case $\boldsymbol{W} \in \mathbb{R}^{N_S \times r}$. Hence a Least Squares Galerkin projection is used with:

$$\boldsymbol{W} = \boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)} \tag{6.48}$$

This results in the reduced equation:

$$(\boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)})^T \boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)} \boldsymbol{a}^r_{(r \times 1)} = (\boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)})^T (\boldsymbol{b})_{(N_C,1)} \tag{6.49}$$

which is re-written as:

$$\bar{\boldsymbol{U}}^T_{(r \times N_S)} \boldsymbol{A}^T_{(N_S \times N_C)} \boldsymbol{A}_{(N_C \times N_S)}\, \bar{\boldsymbol{U}}_{(N_S \times r)} \boldsymbol{a}^r_{(r \times 1)} = \bar{\boldsymbol{U}}^T_{(r \times N_S)} \boldsymbol{A}^T_{(N_S \times N_C)} (\boldsymbol{b})_{(N_C,1)} \tag{6.50}$$

As the above system of equations (6.50) has a symmetric matrix $\bar{\boldsymbol{U}}^T \boldsymbol{A}^T \boldsymbol{A}\, \bar{\boldsymbol{U}}$ on the left hand side, Cholesky decomposition can be used to solve these reduced determined system of equations.

6.3.1.1 Error analysis for linear dynamic problems

The error between the full and reduced solution is determined using the following error estimate:

$$e_1 = \sum_{t_s=1}^{n} \left(\frac{|||\boldsymbol{u}_{\text{full}} - \boldsymbol{u}_{\text{red}}|||}{|||\boldsymbol{u}_{\text{full}}|||} \right) \tag{6.51}$$

where t_s denotes the time step counter and n is the total number of time steps, $\boldsymbol{u}_{\text{full}}$ is the full scale solution evaluated at a set of $\hat{N}$ points in the problem domain and obtained off-line, $\boldsymbol{u}_{\text{red}}$ is the reduced order solution evaluated at the same set of $\hat{N}$ points. The norm $|||\bullet|||$ is defined as follows:

$$|||\boldsymbol{x}||| = \frac{\sqrt{\sum_{i=1}^{\hat{N}} (x_i)^2}}{\sqrt{\hat{N}}} \tag{6.52}$$

where $\hat{N}$ is the length of the vector $\boldsymbol{x}$.

6.3.1.2 Numerical examples

6.3.1.2.1 1D Wave equation

Consider a 1D string clamped at both ends. The governing equation of motion is given by:

$$\alpha \ddot{u} = E u_{,xx}; \quad x \in [0, L],\ t \in [0, T] \tag{6.53}$$

with boundary conditions:

$$u(0, t) = 0 \tag{6.54a}$$

$$u(L, t) = 0 \tag{6.54b}$$

Table 6.1: Error e_1 for 1D wave equation problem

r	% DOF of full model	e_1
1	1.96%	3.202E-003
5	9.8%	4.791E-004
25	49.02%	5.597E-009

and with initial conditions:

$$u(x,0) = 0 \tag{6.55a}$$

$$\dot{u}(x,0) = \omega sin(kx) \tag{6.55b}$$

The following values for the parameters are used: density $\alpha = 1$, length $L = 16$, Young's modulus $E = 1$, wavenumber $k = 5\pi/L$ and the total time $T = 6$. The angular frequency is given by $\omega = kc$, where $c = \sqrt{(}E/\alpha)$. A time step of $\Delta t = 0.2$ is used. For RKCM the domain is discretized using $N_S = 51$ and $N_C = 2N_S - 1$ collocation points. A support size of $3h$ is used in the RK approximation function where h is the nodal spacing of the source points. A total of 31 snapshots are collected. The decay of the first 31 POV's is shown in figure 6.1. As the eigenvalues decay very rapidly after the first eigenvalue, for the reduced order solution the number of POD modes is chosen to be $r = 1$ which is 1.96% of degrees of freedom (DOF) of the full model. Figure 6.2 shows the comparison of the full solution for u and the reduced solution for u at the mid point of the domain at each time step. Figure 6.3 shows the comparison of the full solution and the reduced solution at all the points in the domain at the final time $T = 6$. It can be seen that the reduced solution is very close to the full order solution. Table 6.1 gives the error e_1 for different values of r. It can be seen that the error reduces as the number of modes r are increased, as expected.

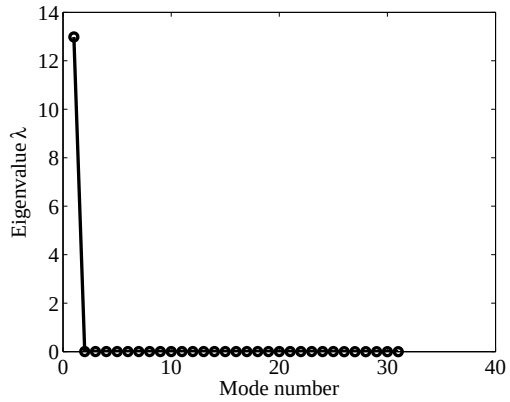


Figure 6.1: Decay of Proper Orthogonal Values for the 1D wave equation problem

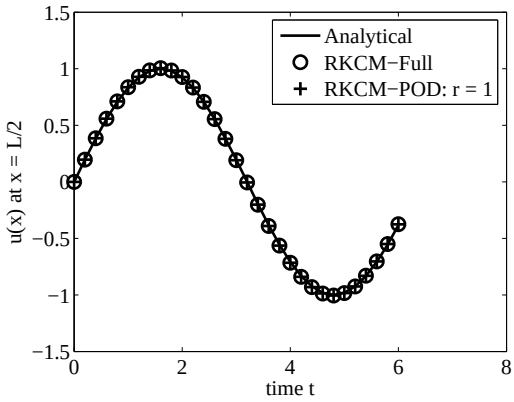


Figure 6.2: Comparison of the solution at the mid point of the domain, plotted for $t \in [0, T]$ for 1D wave equation problem

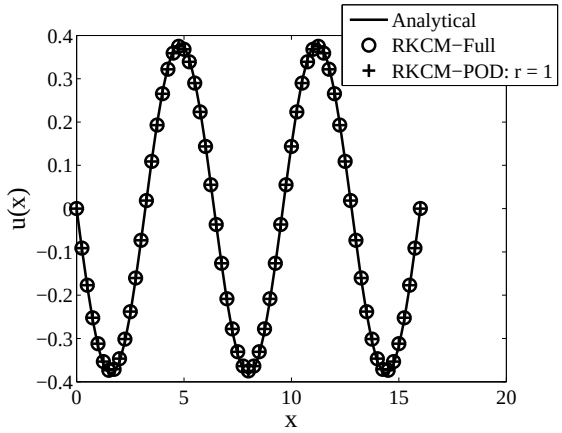


Figure 6.3: Comparison of the solution at the final time $T = 6$ plotted over the domain, for 1D wave equation problem

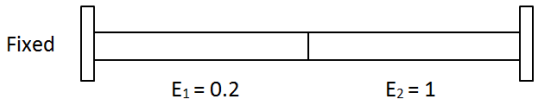


Figure 6.4: 1D bimaterial bar with clamped ends

6.3.1.2.2 1D Bar with heterogeneous material

Consider a 1D heterogeneous bar clamped at both ends as shown figure 6.4. The governing equation of motion is given by:

$$\alpha\ddot{u} = (E(x)u_{,x})_{,x}; \quad x \in [0, L],\ t \in [0, T] \tag{6.56}$$

with boundary conditions:

$$u(0, t) = 0 \tag{6.57a}$$

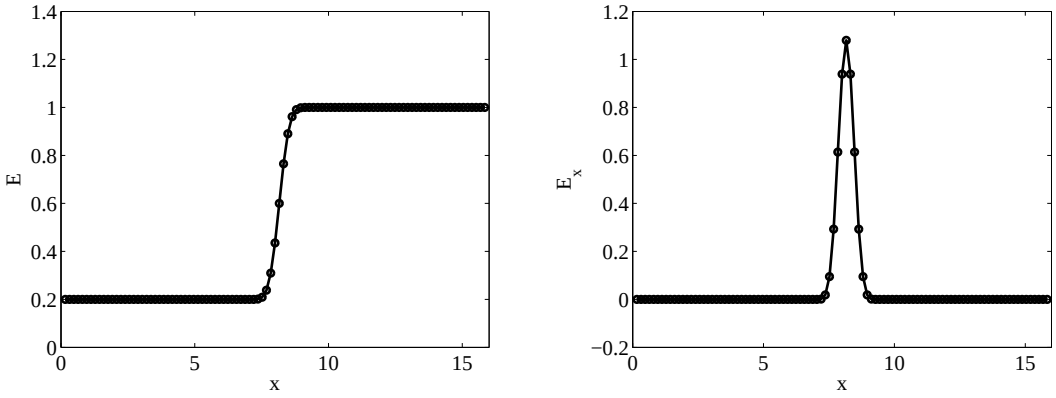
$$u(L, t) = 0 \tag{6.57b}$$

and with initial conditions:

$$u(x, 0) = 0 \tag{6.58a}$$

$$\dot{u}(x, 0) = sin(kx) \tag{6.58b}$$

A smooth transition of material properties at the interface is considered where the Young's modulus is interpolated using the RK approximation functions with linear bases and a support size of $4h$. The interpolated E values and $E_{,x}$ values plotted over the domain are shown in figure 6.5. The following values for the parameters are used: density $\alpha = 1$, length $L = 16$, wavenumber $k = 2\pi/L$ and the total time $T = 6$. A time step of $\Delta t = 0.2$ is used. For RKCM the domain is discretized using $N_S = 51$ and $N_C = 2N_S - 1$ collocation points. A support size of $3h$ is used in the RK approximation



(a) Interpolation of E (b) Interpolation of $E_{,x}$

Figure 6.5: Interpolation of E and $E_{,x}$ in the domain using RK approximation function

Table 6.2: Error e_1 for 1D heterogeneous bar problem

r	% DOF of full model	e_1
1	1.96%	5.776E+000
5	9.8%	3.547E-002
25	49.02%	3.018E-007

function where h is the nodal spacing of the source points. A total of 31 snapshots are collected. The decay of the first 31 POV's is shown in figure 6.6. Considering the eigenvalues decay, for the reduced order solution the number of POD modes is chosen to be $r = 5$ which is 9.8% of degrees of freedom (DOF) of the full model. Figure 6.7 shows the comparison of the full solution for u and the reduced solution for u at the mid point of the domain at each time step. Figure 6.8 shows the comparison of the full solution and the reduced solution at all the points in the domain at the final time $T = 6$. It can be seen that the reduced solution is very close to the full order solution. Table 6.2 gives the error e_1 for different values of r. It can be seen that the error reduces as the number of modes r are increased, as expected.

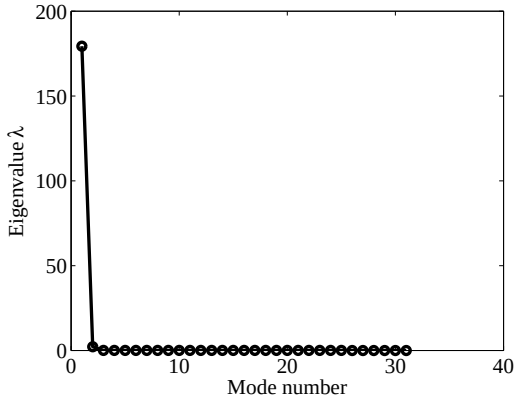


Figure 6.6: Decay of Proper Orthogonal Values for the 1D heterogeneous bar problem

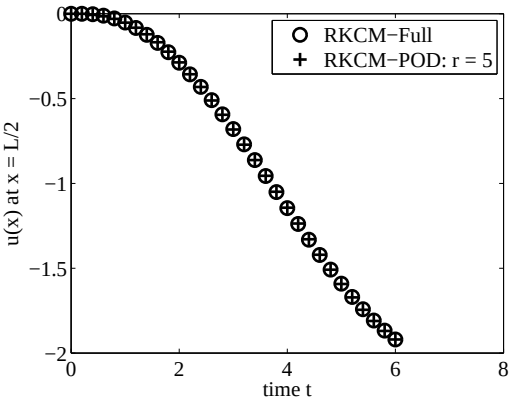


Figure 6.7: Comparison of the solution at the mid point of the domain, plotted for $t \in [0, T]$ for 1D heterogeneous bar problem

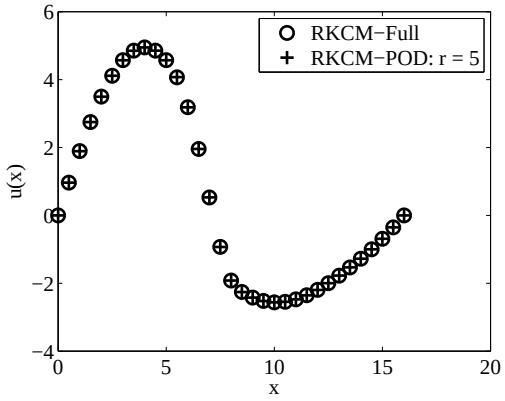


Figure 6.8: Comparison of the solution at the final time $T = 6$ plotted over the domain, for 1D heterogeneous bar problem

6.3.2 Nonlinear problems

Consider the Gauss Newton method for solving the nonlinear strong form equations for static hyperelastic problems using RKCM. The incremental equation is given in (5.76) and re-written again below:

$$\boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{A}'(\boldsymbol{a}))^T\Delta\boldsymbol{a} = -\boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})]_{n+1}^{\nu} \tag{6.59}$$

here n is the load step number and ν is the iteration counter for convergence of the Gauss Newton iterations. $\boldsymbol{A}'$ is the transpose of the Jacobian matrix and is defined in equation (5.56). $\Delta\boldsymbol{a}$ is the vector of incremental nodal coordinates which needs to be obtained in $\nu+1^{\text{th}}$ iteration of $n+1^{\text{th}}$ load step. $\boldsymbol{r}$ is the residual force vector. In the off-line phase when the full model solution is run, the snapshots matrix consists of the collection of nodal coordinates vectors, $\boldsymbol{a}$, obtained at the end of every load step. The reduced bases $\bar{\boldsymbol{U}}$ are obtained by truncation of the POD bases, which are obtained from singular value decomposition of the snapshots matrix.

In order to carry out the Least Squares Galerkin projection for getting the reduced order model, consider the equations in (6.59) without the least squares projection, that is the overdetermined system given below:

$$(\boldsymbol{A}'(\boldsymbol{a}))^T\Delta\boldsymbol{a} = -\boldsymbol{r}(\boldsymbol{a}) \tag{6.60}$$

The subscript and superscript $(]_{n+1}^{\nu})$ on the right hand side of equation (6.60) have been ignored for convenience. Consider a 1D case where N_S source points and N_C collocation points are used to discretize the domain. The dimensions of the matrices given in

equation (6.60) will be as follows:

$$(\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\Delta\boldsymbol{a}_{(N_S\times 1)} = -\boldsymbol{r}(\boldsymbol{a})_{(N_C\times 1)} \tag{6.61}$$

$\Delta\boldsymbol{a}$ is approximated by projecting on to the reduced bases as follows:

$$\Delta\boldsymbol{a} \approx \Delta\boldsymbol{a}^h_{(N_S\times 1)} = \bar{\boldsymbol{U}}_{(N_S\times r)}\Delta\boldsymbol{a}^r_{(r\times 1)} \tag{6.62}$$

where $\bar{\boldsymbol{U}} = [\phi_1, \cdots, \phi_r]$ is the reduced bases obtained from truncating POD modes of the response matrix and r is the number of reduced modes or the reduced degrees of freedom for solving the reduced system. $\Delta\boldsymbol{a}^r$ are the unknown coefficients in the reduced system. Substituting the reduced approximation given in (6.62) in the matrix equation (6.61) results in:

$$(\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)}\Delta\boldsymbol{a}^r_{(r\times 1)} + \boldsymbol{r}(\boldsymbol{a})_{(N_C\times 1)} = \boldsymbol{e} \tag{6.63}$$

where $\boldsymbol{e}$ is the residual error. To minimize this error, the residual error is constrained to be orthogonal to a subspace $\mathbb{W}$, defined by the bases $\boldsymbol{W} \in \mathbb{R}^{N_C\times r}$, which gives:

$$\begin{aligned} &\boldsymbol{W}^T\boldsymbol{e} = \boldsymbol{0} \\ \Rightarrow &\boldsymbol{W}^T_{(r\times N_C)}(\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)}\Delta\boldsymbol{a}^r_{(r\times 1)} = -\boldsymbol{W}^T_{(r\times N_C)}\boldsymbol{r}(\boldsymbol{a})_{(N_C\times 1)} \end{aligned} \tag{6.64}$$

It can be seen that the bases $\boldsymbol{W}$ should belong to $\mathbb{R}^{N_C\times r}$ for projecting the RKCM equations. A projection using $\boldsymbol{W} = \bar{\boldsymbol{U}}$ is not possible since in this case $\boldsymbol{W} \in \mathbb{R}^{N_S\times r}$. Hence a Least Squares Galerkin projection is used with:

$$\boldsymbol{W} = (\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)} \tag{6.65}$$

This results in the reduced equation:

$$
\begin{aligned}
&((\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)})^T(\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)}\Delta\boldsymbol{a}^r_{(r\times 1)}\\
&=-((\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)})^T\boldsymbol{r}(\boldsymbol{a})_{(N_C\times 1)}
\end{aligned}
\tag{6.66}
$$

which is re-written as:

$$
\begin{aligned}
&\bar{\boldsymbol{U}}^T_{(r\times N_S)}\boldsymbol{A}'(\boldsymbol{a})_{(N_S\times N_C)}(\boldsymbol{A}'(\boldsymbol{a}))^T_{(N_C\times N_S)}\bar{\boldsymbol{U}}_{(N_S\times r)}\Delta\boldsymbol{a}^r_{(r\times 1)}\\
&=-\bar{\boldsymbol{U}}^T_{(r\times N_S)}\boldsymbol{A}'(\boldsymbol{a})_{(N_S\times N_C)}\boldsymbol{r}(\boldsymbol{a})_{(N_C\times 1)}
\end{aligned}
\tag{6.67}
$$

As the above system of equations (6.67) has a symmetric matrix $\bar{\boldsymbol{U}}^T\boldsymbol{A}'(\boldsymbol{a})(\boldsymbol{A}'(\boldsymbol{a}))^T\bar{\boldsymbol{U}}$ on the left hand side, Cholesky decomposition can be used to solve these reduced determined system of equations.

As it can be seen from the above procedure, the saving in time using the reduced order model is achieved for solving the system of equations in every iteration of a load step, in case of the static nonlinear analysis. The cost associated with constructing the tangent stiffness and residual force matrices remains the same. As shown in the last chapter, compared to RKPM with Gauss integration, for RKCM the time taken for assembling the stiffness and force vectors in each iteration is much lesser, which makes the reduced order modeling very efficient. In general for nonlinear analysis using a reduced order projected solution, it should be noted that for the reduced order model, the method is effective only if the time for projecting the matrices and solving the reduced system is greater than the time taken to solve the full system in each iteration. This is true only when the size of the full scale model is large enough and in fact for smaller systems, the time for projecting and solving the reduced system might be comparable with the full scale system.

6.3.2.1 Convergence check for nonlinear iterations

For the reduced order model, where the reduced system of equations are solved in each iteration, the convergence in each load step is checked in the reduced system. That is, norm of the projected residual force vector should converge. From the reduced system of equations given in (6.67), the condition for convergence in the Newton iterations should be:

$$\bar{\boldsymbol{U}}^T \boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a}) = \boldsymbol{0} \tag{6.68}$$

Taking this into account the following convergence criteria is used:

$$\|\bar{\boldsymbol{U}}^T \boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})\|_0 < \varepsilon(\|\bar{\boldsymbol{U}}^{\boldsymbol{T}} \boldsymbol{A}'(\boldsymbol{a})\boldsymbol{r}(\boldsymbol{a})\|_0)_{\nu=1} \tag{6.69}$$

where ε is the specified tolerance.

6.3.2.2 Error analysis for nonlinear static problems

The error between the full and reduced solution is determined using the following two error estimates. The first one is used to compare the error between the full and the reduced solutions, and is given by:

$$e_1 = \sum_{n=1}^{N} \left(\frac{|||\boldsymbol{u}_{\text{full}} - \boldsymbol{u}_{\text{red}}|||}{|||\boldsymbol{u}_{\text{full}}|||} \right) \tag{6.70}$$

where n denotes the load step counter and N is the total number of load steps, $\boldsymbol{u}_{\text{full}}$ is the full scale solution evaluated at a set of $\hat{N}$ points in the problem domain and obtained off-line, $\boldsymbol{u}_{\text{red}}$ is the reduced order solution evaluated at the same set of $\hat{N}$ points. The norm $|||\bullet|||$ is defined as follows:

$$|||\boldsymbol{x}||| = \frac{\sqrt{\sum_{i=1}^{\hat{N}} (x_i)^2}}{\sqrt{\hat{N}}} \tag{6.71}$$

where $\hat{N}$ is the length of the vector $\boldsymbol{x}$.

The second error estimate is used to compare the error between the derivatives of the full and the reduced solutions, and is given by:

$$e_2 = \sum_{n=1}^{N} \left(\frac{||| \sum_{i=1}^{3} \sum_{j=1}^{3} (u_{i,j(full)} - u_{i,j(red)}) |||}{||| \sum_{i=1}^{3} \sum_{j=1}^{3} (u_{i,j(full)}) |||} \right) \tag{6.72}$$

6.3.2.3 Numerical examples

Using the proposed method, examples for nonlinear static problems for hyperelasticity are given. Details of reduction in CPU time using the reduced model and error analysis results are also presented.

Note about the linear solver used:
For the present study standard subroutines from LAPACK are used for solving the linear system of equations in every iteration, in order to be consistent for the different methods used to solve the full and reduced systems. For the full scale model the system of overdetermined equations is solved using QR factorization. For the reduced model, the determined system is solved using Cholesky factorization as discussed in sections 6.3.1 and 6.3.2.

6.3.2.3.1 'Near' simple shear deformation

Near simple shear deformation of a rubber block as given in section 5.3.10.1.1 is considered again. The geometry and material parameters are the same as taken previously. The domain is discretized using $17 \times 5 \times 5$ source points and $33 \times 9 \times 9$ collocation points. A total of 2673 collocation points in the domain and 1154 collocation points on the boundary are used. The rubber block is sheared up to 500% engineering shear strain as shown in figure 5.11. A total of 50 snapshots are collected which is equal to the number of load steps applied. The decay of the first 50 POV's is shown in figure 6.9. The

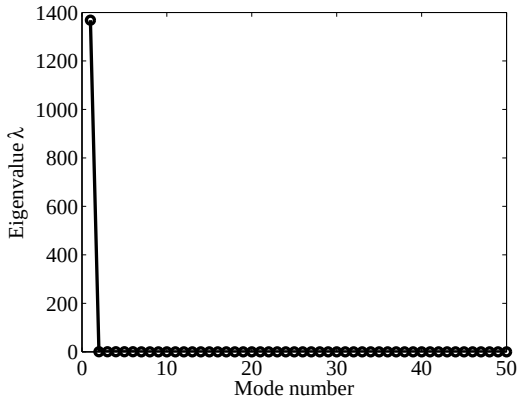


Figure 6.9: Decay of Proper Orthogonal Values for the 3D simple shear problem

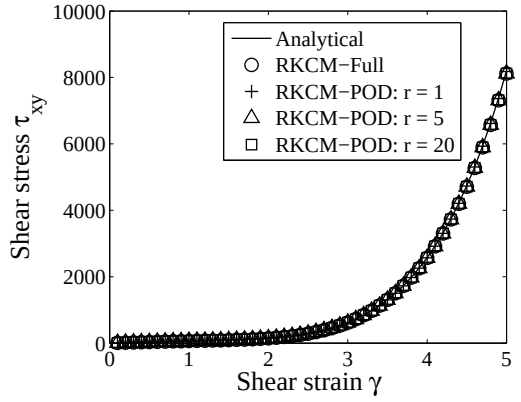


Figure 6.10: Cauchy shear stress (in N/cm^2) comparison for full and reduced models for the 3D simple shear problem using RKCM

total number of degrees of freedom (DOF) in the full model are 1275. Figure 6.10 shows the comparison of the Cauchy shear stress at the mid point of the domain for full and reduced models in every load step. As the POV's decay rapidly, for the reduced solutions three cases are chosen where the number of POD modes are taken to be $r = 1$ which is 0.078% of DOF of the full model, $r = 5$ which is 0.392% of DOF of the full model and $r = 20$ which is 1.568% of DOF of the full model. It can be seen that very good agreement between the full and reduced models is obtained. The total CPU time taken for solving the full and reduced systems is given in table 6.3. Considerable reduction of CPU time is obtained for the reduced model compared to the full model. Table 6.4 gives

Table 6.3: Timings comparison for 3D simple shear problem

Model	CPU time (mins)	% Reduction of time compared to full model
RKCM-Full	98.504	–
RKCM-POD: $r = 1$	10.681	89.156
RKCM-POD: $r = 5$	15.109	84.661
RKCM-POD: $r = 20$	18.445	81.274

Table 6.4: Errors e_1 and e_2 for 3D simple shear problem

r	% DOF of full model	e_1	e_2
1	0.078%	9.271E-03	4.076E-02
5	0.392%	1.188E-03	5.267E-03
20	1.568%	2.393E-06	1.060E-05

the errors e_1 and e_2 for different values of r. It can be seen that the errors reduce as the number of modes r is increased. Also for this problem, even with r being only 1.568% of DOF of the full model, very less error values are obtained.

6.3.2.3.2 Inflation of a rubber tube

Inflation of a rubber tube using 2D plane strain analysis as given in section 5.3.10.2 is considered again. The geometry and material parameters are the same as taken previously. The domain is discretized using 1717 source points, 6633 collocation points in the domain and 464 collocation points on the boundary. The tube is inflated till the inner radius is doubled. A total of 24 snapshots are collected which is equal to the number of load steps applied. The decay of the first 24 POV's is shown in figure 6.11. The total number of degrees of freedom (DOF) in the full model are 3434. Figure

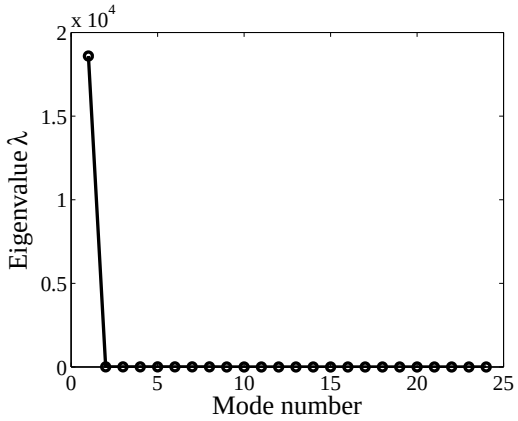


Figure 6.11: Decay of Proper Orthogonal Values for inflation of rubber tube problem

6.12 shows the comparison of the radial Cauchy stress σ_{rr} at the midpoint of the inner circumference for full and reduced models in every load step. The POV's decay rapidly after the first two modes. For the reduced solutions three cases are chosen where the number of POD modes are taken to be $r = 2$ which is 0.058% of DOF of the full model, $r = 5$ which is 0.145% of DOF of the full model and $r = 10$ which is 0.291% of DOF of the full model. Additionally the comparison of radial Cauchy stress σ_{rr} in the final deformed configuration for the full and the reduced model with $r = 5$ is shown in figure 6.13. It can be seen that very good agreement between the full and reduced models is obtained for the cases where $r \geq 5$. The total CPU time taken for solving the full and reduced systems is given in table 6.5. Considerable reduction of CPU time is obtained for the reduced model compared to the full model. Table 6.6 gives the errors e_1 and e_2 for different values of r. It can be seen that the errors reduce as the number of modes r is increased. Also for this problem, even with r being only 0.291% of DOF of the full model, very less error values are obtained.

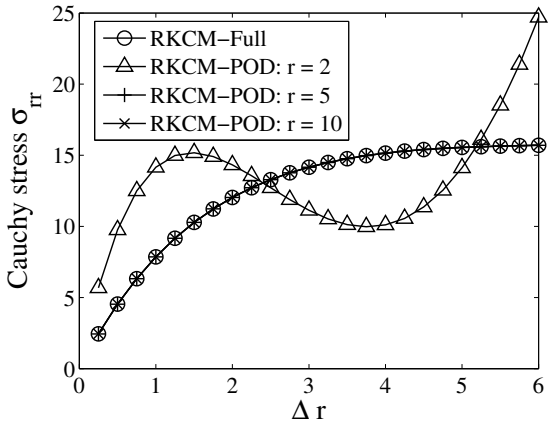
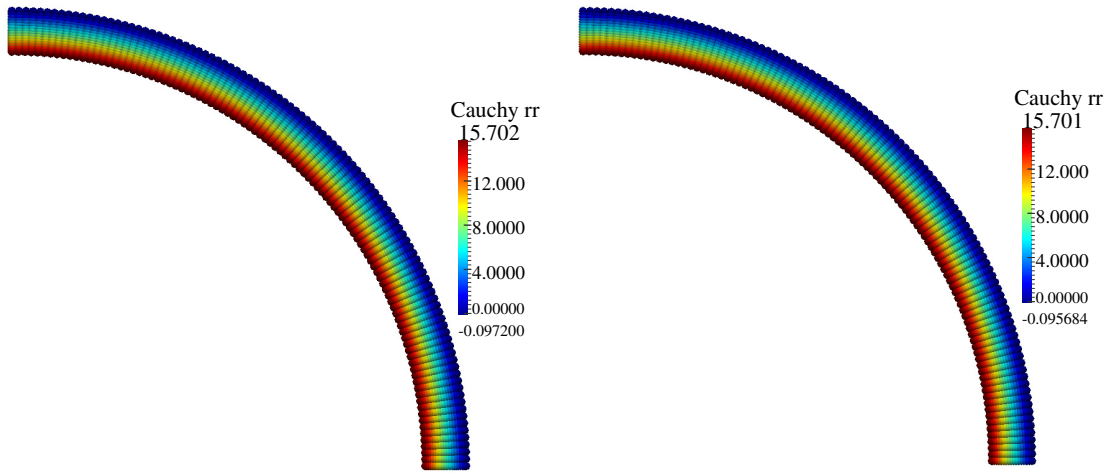


Figure 6.12: Radial Cauchy stress σ_{rr} (in N/cm^2) comparison for full and reduced models for inflation of rubber tube problem using RKCM



(a) Full model

(b) Reduced model with $r = 5$

Figure 6.13: Radial Cauchy stress σ_{rr} in the final deformed configuration for the full and reduced models for inflation of rubber tube problem

Table 6.5: Timings comparison for inflation of rubber tube problem

Model	CPU time (mins)	% Reduction of time compared to full model
RKCM-Full	196.983	–
RKCM-POD: $r = 2$	2.491	98.735
RKCM-POD: $r = 5$	2.684	98.637
RKCM-POD: $r = 10$	2.968	98.493

Table 6.6: Errors e_1 and e_2 for inflation of rubber tube problem

r	% DOF of full model	e_1	e_2
2	0.058%	6.356E-02	2.917E-01
5	0.145%	5.021E-05	2.777E-04
10	0.291%	5.305E-06	2.933E-05

6.3.2.3.3 Isometric contraction of muscle using standard geometry

Isometric contraction of skeletal muscle using standard geometry in 3D as given in section 5.3.10.3.1 is considered again. The model geometry and material parameters are the same as taken previously, with fiber direction vertically pointing in the z direction at all collocation points. The domain is discretized using $9 \times 9 \times 17$ source points and $17 \times 17 \times 33$ collocation points. A total of 9537 collocation points in the domain and 2562 collocation points on the boundary are used. The ends of the muscle are fixed and the activation factor is increased to a maximum in 10 load steps and a total of 10 snapshots are collected, one in each load step. The decay of the first 10 POV's is shown in figure 6.14. The total number of degrees of freedom (DOF) in the full model are 4131. Figure 6.15(a) shows the comparison of the Cauchy stress σ_{zz} at the mid point of the domain for full and reduced models in every load step. The comparison of reaction force

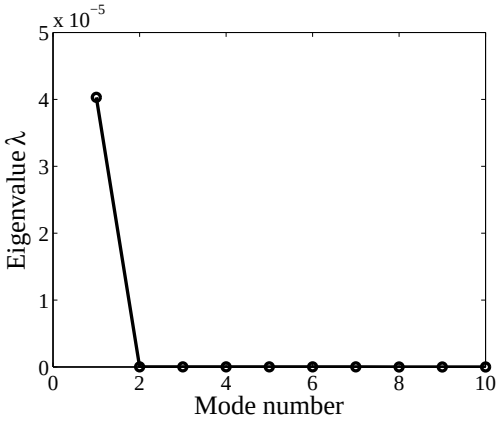
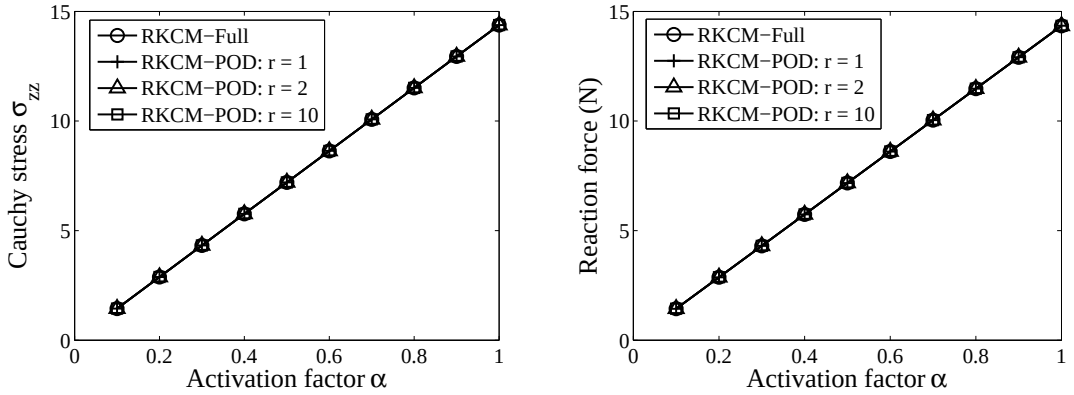


Figure 6.14: Decay of Proper Orthogonal Values for isometric contraction of muscle problem using standard geometry

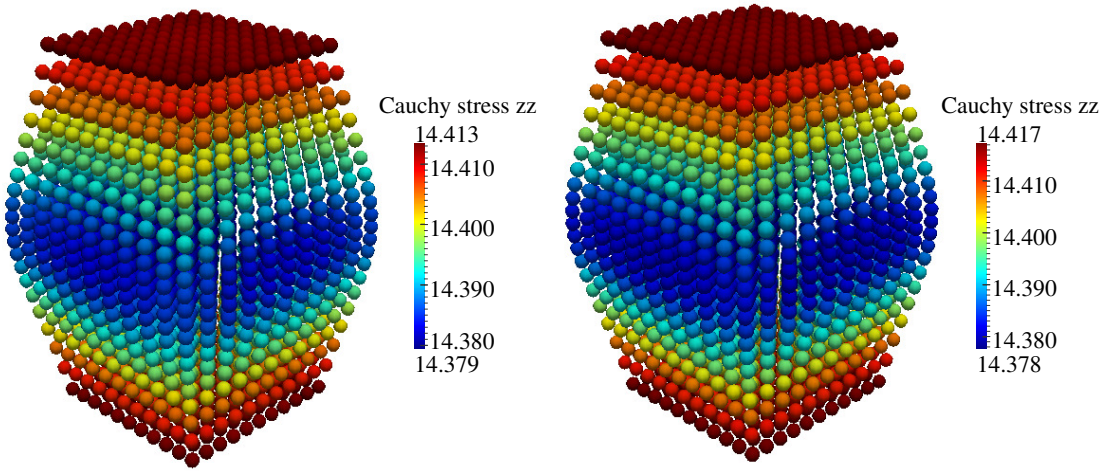
generated at the fixed end for the full and reduced models is shown in figure 6.15(b). As the POV's decay rapidly, for the reduced solutions three cases are chosen where the number of POD modes are taken to be $r = 1$ which is 0.0242% of DOF of the full model, $r = 2$ which is 0.0484% of DOF of the full model and $r = 10$ which is 0.242% of DOF of the full model. Additionally the comparison of Cauchy stress σ_{zz} in the final deformed configuration for the full and the reduced model with $r = 1$ is shown in figure 6.16. It can be seen that very good agreement between the full and reduced models is obtained. The total CPU time taken for solving the full and reduced systems is given in table 6.7. 32 cores parallel processing was used for both the full and reduced models. The parallelization was done only for assembling the stiffness and force matrices and not for solving the linear system in each iteration. Considerable reduction of CPU time is obtained for the reduced model compared to the full model. Table 6.8 gives the errors e_1 and e_2 for different values of r. It can be seen that the errors reduce as the number of modes r is increased. Also for this problem, even with r being only 0.242% of DOF of the full model, very less error values are obtained.



(a) Comparison of Cauchy stress σ_{zz} at the mid point of the domain

(b) Comparison of reaction force generated at the fixed end

Figure 6.15: Comparison of Cauchy stress σ_{zz} (in N/cm^2) at the mid point of the domain and muscle reaction force (in N) at the fixed end for full and reduced models for isometric contraction of 3D muscle problem using standard geometry using RKCM



(a) Full model

(b) Reduced model with $r = 1$

Figure 6.16: Cauchy stress σ_{zz} in the final deformed configuration (deformation scaled 500 times) for the full and reduced models for the 3D muscle problem using standard geometry

Table 6.7: Timings comparison for 3D muscle problem using standard geometry

Model	CPU time (mins)	% Reduction of time compared to full model
RKCM-Full	372.567	–
RKCM-POD: $r = 1$	23.843	93.600
RKCM-POD: $r = 2$	23.961	93.568
RKCM-POD: $r = 10$	24.420	93.445

Table 6.8: Errors e_1 and e_2 for 3D muscle problem using standard geometry

r	% DOF of full model	e_1	e_2
1	0.0242%	4.975E-02	5.622E-02
2	0.0484%	5.685E-03	6.973E-03
10	0.242%	2.277E-06	2.402E-06

6.3.2.3.4 Isometric contraction of medial gastrocnemius muscle

Isometric contraction of the medial gastorcnemius muscle as given in section 5.3.10.3.2 is considered again. The model geometry and material parameters are the same as taken previously. The actual muscle fiber directions as obtained from the DT images are specified at the collocation points. The domain is discretized using 1837 source points, 3727 collocation points in the domain and 1320 collocation points on the boundary. The ends of the muscle are fixed and the activation factor is increased to a maximum in 10 load steps and a total of 10 snapshots are collected, one in each load step. The decay of the first 10 POV's is shown in figure 6.17. The total number of degrees of freedom (DOF) in the full model are 5511. The comparison of reaction force generated at the fixed end for the full and reduced models is shown in figure 6.18. As the POV's decay rapidly, for the reduced solutions two cases are chosen where the

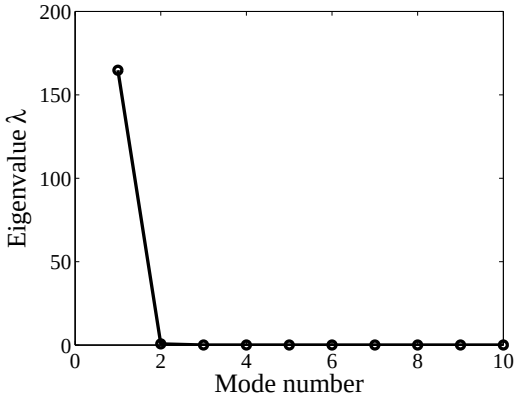


Figure 6.17: Decay of Proper Orthogonal Values for isometric contraction of medial gastorcnemius muscle problem

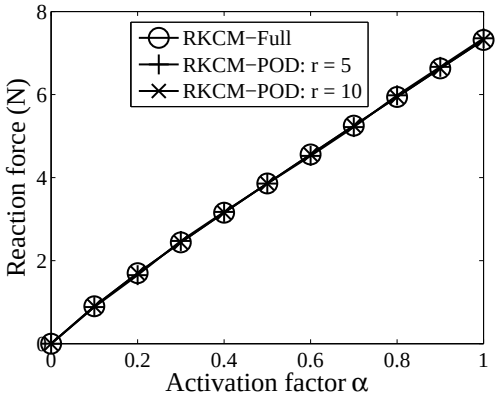
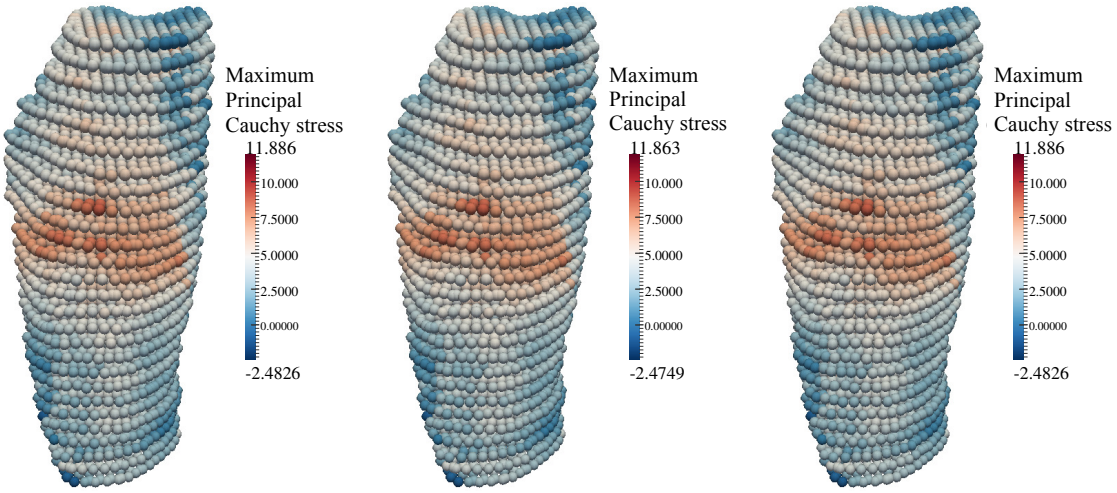


Figure 6.18: Comparison of reaction force generated at the fixed end for isometric contraction of medial gastrocnemius muscle

number of POD modes are taken to be $r = 5$ which is 0.0907% of DOF of the full model and $r = 10$ which is 0.1814% of DOF of the full model. The comparison of maximum principal Cauchy stress in the final deformed configuration for the full and the reduced models is shown in figure 6.19. It can be seen that very good agreement between the full and reduced models is obtained. The total CPU time taken for solving the full and reduced systems is given in table 6.9. 32 cores parallel processing was used for both the full and reduced models. The parallelization was done only for assembling the stiffness and force matrices and not for solving the linear system in each iteration. Considerable



(a) Full model (b) Reduced model with $r = 5$ (c) Reduced model with $r = 10$

Figure 6.19: Comparison of maximum principal Cauchy stress (in N/cm^2), in the final deformed configuration for the full and reduced models for isometric contraction of medial gastrocnemius muscle problem

reduction of CPU time is obtained for the reduced model compared to the full model. Table 6.10 gives the errors e_1 and e_2 for different values of r. It can be seen that the errors reduce as the number of modes r is increased. Also for this problem, even with r being only 0.1814% of DOF of the full model, very less error values are obtained.

Table 6.9: Timings comparison for isometric contraction of medial gastrocnemius muscle problem

Model	CPU time (mins)	% Reduction of time compared to full model
RKCM-Full	1868.9	–
RKCM-POD: $r = 5$	19.9	98.935
RKCM-POD: $r = 10$	21.932	98.826

Table 6.10: Errors e_1 and e_2 for isometric contraction of medial gastrocnemius muscle problem

r	% DOF of full model	e_1	e_2
5	0.0907%	1.365E-02	1.376E-02
10	0.1814%	3.999E-05	3.983E-05

Chapter 7

Conclusion

7.1 Conclusions

To obtain accurate results for skeletal muscle simulation, anatomically accurate models and efficient computational methods are essential. The objective of this work is to develop the effective segmentation methods, image-based computational framework and model order reduction method to enable efficient and accurate modeling of skeletal muscles.

Two segmentation schemes have been proposed and implemented. One for construction of full 3D skeletal models by stacking 2D segmented images with multiple materials and the other is a semi-automatic method for segmenting adjacent muscles with different fiber directions, using MR and DT images. The actual muscle fiber directions obtained from DT imaging are also used in the models for simulation. Meshfree methods are introduced to allow for using the image pixels directly as nodes for domain discretization, avoiding the construction of meshes for complicated geometries. A smooth transition of material properties across material interfaces in the muscles is achieved by approximating the material properties using the reproducing kernel shape

functions. Numerical simulation using weak form based RKPM for medial gastrocnemius muscle of young and old subjects shows a non-proportional decrease in muscle force generation associated with the decrease in muscle volumes. In meshfree methods based on the weak form, special treatment is necessary to deal with domain integration and imposition of essential boundary conditions. To alleviate this problem, the strong form RKCM is proposed for nonlinear analysis of skeletal muscles. In strong form collocation methods, the need for domain integration is eliminated and the essential boundary conditions are imposed directly in the collocation equations. The Gauss Newton method is employed for solving the nonlinear iterations and suitable convergence criteria are proposed. Numerical results show that RKCM gives more accurate results compared to RKPM.

Reduced order modeling (ROM) has been proposed for RKCM in solving both linear and nonlinear problems. In this approach, the POD bases are constructed from snapshots, and a Least Squares Galerkin projection is proposed to project the system of over-determined equations to a lower dimensional discrete system. In nonlinear analysis using RKCM, the construction of stiffness and force vectors is less time consuming compared to weak form based methods such as RKPM using Gauss integration. Using the proposed ROM approach, significant reduction in computational time can be achieved while maintaining sufficient accuracy. Error analysis is also performed to show the accuracy of the proposed method. It is shown that high accuracy can be obtained even for reduced models with only 1-2% of degrees of freedom of the full model.

7.2 Recommendations for future research

Directions for future research include:

1. Extension of RKCM for nonlinear analysis to nearly incompressible materials

without locking and pressure oscillation requires further investigation. Mixed formulation for RBCM in solving nearly incompressible linear problems has been proposed in [131], and this approach can be extended to RKCM for hyperelasticity.

2. Extend ROM for RKCM for patient specific modeling of skeletal muscles with multiple materials.

Appendix A

Finite difference scheme for ACWE method

Equation (3.15) for the ACWE method, is solved numerically using a semi-implicit scheme as given below:

$$\begin{aligned}\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\triangle t} &= \delta_{\varepsilon}(\phi_{i,j}^{n})\frac{\mu}{h^2}\left[\left(\frac{\phi_{i+1,j}^{n}-\phi_{i,j}^{n+1}}{\mathbb{A}}\right)-\left(\frac{\phi_{i,j}^{n+1}-\phi_{i-1,j}^{n}}{\mathbb{B}}\right)\right] \\ &+ \delta_{\varepsilon}(\phi_{i,j}^{n})\frac{\mu}{h^2}\left[\left(\frac{\phi_{i,j+1}^{n}-\phi_{i,j}^{n+1}}{\mathbb{C}}\right)-\left(\frac{\phi_{i,j}^{n+1}-\phi_{i,j-1}^{n}}{\mathbb{D}}\right)\right] \\ &+ \delta_{\varepsilon}(\phi_{i,j}^{n})\left[-\nu-\lambda_1(u_{o,i,j}-c_1(\phi_n))^2+\lambda_2(u_{o,i,j}-c_2(\phi_n))^2\right] \qquad \text{(A.1)}\end{aligned}$$

where

$$\mathbb{A} = \sqrt{\left(\frac{\phi^n_{i+1,j} - \phi^n_{i,j}}{h}\right)^2 + \left(\frac{\phi^n_{i,j+1} - \phi^n_{i,j-1}}{2h}\right)^2} \tag{A.2a}$$

$$\mathbb{B} = \sqrt{\left(\frac{\phi^n_{i,j} - \phi^n_{i-1,j}}{h}\right)^2 + \left(\frac{\phi^n_{i-1,j+1} - \phi^n_{i-1,j-1}}{2h}\right)^2} \tag{A.2b}$$

$$\mathbb{C} = \sqrt{\left(\frac{\phi^n_{i+1,j} - \phi^n_{i-1,j}}{2h}\right)^2 + \left(\frac{\phi^n_{i,j+1} - \phi^n_{i,j}}{h}\right)^2} \tag{A.2c}$$

$$\mathbb{D} = \sqrt{\left(\frac{\phi^n_{i+1,j-1} - \phi^n_{i-1,j-1}}{2h}\right)^2 + \left(\frac{\phi^n_{i,j} - \phi^n_{i,j-1}}{h}\right)^2} \tag{A.2d}$$

h is the spacing between the pixels in the image, which is the same in both x and y directions and $\triangle t$ is the time step increment.

Appendix B

Euler Lagrange equations for multichannel multiphase segmentation

For the multiphase multichannel segmentation the 3 Euler Lagrange equations are obtained by minimizing the functional given in equation (4.3) with ϕ_1, ϕ_2, ϕ_3 respectively each time keeping c_p^i fixed. Assuming for convenience $\lambda_i = \lambda$ for $i = 1, \cdots, 8$ and $\mu_k = \mu$ for $k = 1, \cdots, 3$ in all channels.

Minimizing the functional in equation (4.3) with respect to ϕ_1 keeping c_p^i, ϕ_2, ϕ_3

constant:

$$
\begin{aligned}
\frac{\partial \phi_1}{\partial t} &= \delta(\phi_1)\mu\nabla \cdot \left(\frac{\nabla\phi_1}{|\nabla\phi_1|}\right) \\
&\quad - \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_1^i\left(u_o^i(\boldsymbol{x}) - c_1^i\right)^2\right)H(\phi_2)H(\phi_3) \\
&\quad - \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_2^i\left(u_o^i(\boldsymbol{x}) - c_2^i\right)^2\right)(1-H(\phi_2))H(\phi_3) \\
&\quad - \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_4^i\left(u_o^i(\boldsymbol{x}) - c_4^i\right)^2\right)H(\phi_2)(1-H(\phi_3)) \\
&\quad - \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_7^i\left(u_o^i(\boldsymbol{x}) - c_7^i\right)^2\right)(1-H(\phi_2))(1-H(\phi_3)) \\
&\quad + \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_3^i\left(u_o^i(\boldsymbol{x}) - c_3^i\right)^2\right)H(\phi_2)H(\phi_3) \\
&\quad + \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_5^i\left(u_o^i(\boldsymbol{x}) - c_5^i\right)^2\right)(1-H(\phi_2))H(\phi_3) \\
&\quad + \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_6^i\left(u_o^i(\boldsymbol{x}) - c_6^i\right)^2\right)H(\phi_2)(1-H(\phi_3)) \\
&\quad + \delta(\phi_1)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_8^i\left(u_o^i(\boldsymbol{x}) - c_8^i\right)^2\right)(1-H(\phi_2))(1-H(\phi_3)) \qquad \text{(B.1)}
\end{aligned}
$$

Minimizing the functional in equation (4.3) with respect to ϕ_2 keeping c_p^i, ϕ_1, ϕ_3

constant:

$$
\begin{aligned}
\frac{\partial \phi_2}{\partial t} &= \delta(\phi_2)\mu \nabla \cdot \left(\frac{\nabla \phi_2}{|\nabla \phi_2|} \right) \\
&- \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_1^i \left(u_o^i(\boldsymbol{x}) - c_1^i \right)^2 \right) H(\phi_1) H(\phi_3) \\
&- \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_3^i \left(u_o^i(\boldsymbol{x}) - c_3^i \right)^2 \right) (1 - H(\phi_1)) H(\phi_3) \\
&- \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_4^i \left(u_o^i(\boldsymbol{x}) - c_4^i \right)^2 \right) H(\phi_1) (1 - H(\phi_3)) \\
&- \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_6^i \left(u_o^i(\boldsymbol{x}) - c_6^i \right)^2 \right) (1 - H(\phi_1))(1 - H(\phi_3)) \\
&+ \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_2^i \left(u_o^i(\boldsymbol{x}) - c_2^i \right)^2 \right) H(\phi_1) H(\phi_3) \\
&+ \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_5^i \left(u_o^i(\boldsymbol{x}) - c_5^i \right)^2 \right) (1 - H(\phi_1)) H(\phi_3) \\
&+ \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_7^i \left(u_o^i(\boldsymbol{x}) - c_7^i \right)^2 \right) H(\phi_1) (1 - H(\phi_3)) \\
&+ \delta(\phi_2) \left(\frac{1}{4} \sum_{i=1}^{4} \lambda_8^i \left(u_o^i(\boldsymbol{x}) - c_8^i \right)^2 \right) (1 - H(\phi_1))(1 - H(\phi_3)) \qquad \text{(B.2)}
\end{aligned}
$$

Minimizing the functional in equation (4.3) with respect to ϕ_3 keeping c_p^i, ϕ_1, ϕ_2

constant:

$$
\begin{aligned}
\frac{\partial \phi_3}{\partial t} &= \delta(\phi_3)\mu\nabla \cdot \left(\frac{\nabla \phi_3}{|\nabla \phi_3|}\right) \\
&- \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_1^i\left(u_o^i(\boldsymbol{x}) - c_1^i\right)^2\right) H(\phi_1)H(\phi_2) \\
&- \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_3^i\left(u_o^i(\boldsymbol{x}) - c_3^i\right)^2\right) (1 - H(\phi_1))H(\phi_2) \\
&- \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_2^i\left(u_o^i(\boldsymbol{x}) - c_2^i\right)^2\right) H(\phi_1)(1 - H(\phi_2)) \\
&- \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_5^i\left(u_o^i(\boldsymbol{x}) - c_5^i\right)^2\right) (1 - H(\phi_1))(1 - H(\phi_2)) \\
&+ \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_4^i\left(u_o^i(\boldsymbol{x}) - c_4^i\right)^2\right) H(\phi_1)H(\phi_2) \\
&+ \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_6^i\left(u_o^i(\boldsymbol{x}) - c_6^i\right)^2\right) (1 - H(\phi_1))H(\phi_2) \\
&+ \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_7^i\left(u_o^i(\boldsymbol{x}) - c_7^i\right)^2\right) H(\phi_1)(1 - H(\phi_2)) \\
&+ \delta(\phi_3)\left(\frac{1}{4}\sum_{i=1}^{4}\lambda_8^i\left(u_o^i(\boldsymbol{x}) - c_8^i\right)^2\right) (1 - H(\phi_1))(1 - H(\phi_2)) \qquad \text{(B.3)}
\end{aligned}
$$

Appendix C

Hyperelasticity material tensors for RKCM

The expressions for the first PK stress σ_{Ai} and the material tensors $\mathbb{C}^4_{iAjB}$, $\mathbb{C}^6_{iAjBkC}$ and $\mathbb{C}^8_{iAjBkClD}$ are given as follows:

$$\sigma_{Ai} = \frac{\partial W}{\partial F_{iA}} = \frac{\partial \bar{W}}{\partial F_{iA}} + \frac{\partial \tilde{W}}{\partial F_{iA}} \tag{C.1}$$

where

$$\frac{\partial \bar{W}}{\partial F_{iA}} = \frac{\partial \bar{W}}{\partial \bar{I}_1}\frac{\partial \bar{I}_1}{\partial F_{iA}} + \frac{\partial \bar{W}}{\partial \bar{I}_2}\frac{\partial \bar{I}_2}{\partial F_{iA}} \tag{C.2}$$

here

$$\frac{\partial \bar{I}_1}{\partial F_{iA}} = 2J^{-2/3}\left(F_{iA} - \frac{1}{3}I_1 F^{-1}_{Ai}\right) \tag{C.3a}$$

$$\frac{\partial \bar{I}_2}{\partial F_{iA}} = 2J^{-4/3}\left(I_1 F_{iA} - F_{ip}G_{pA} - \frac{2}{3}I_2 F^{-1}_{Ai}\right) \tag{C.3b}$$

and

$$\frac{\partial \tilde{W}}{\partial F_{iA}} = PJF^{-1}_{Ai} \tag{C.4}$$

The tensor $\mathbb{C}^4_{iAjB}$ is given by:

$$\mathbb{C}^4_{iAjB} = \frac{\partial^2 W}{\partial F_{jB}\partial F_{iA}} = \frac{\partial^2 \bar{W}}{\partial F_{jB}\partial F_{iA}} + \frac{\partial^2 \tilde{W}}{\partial F_{jB}\partial F_{iA}} \tag{C.5}$$

where

$$\begin{aligned}\frac{\partial^2 \bar{W}}{\partial F_{jB}\partial F_{iA}} = \bar{K}_{11}\frac{\partial \bar{I}_1}{\partial F_{jB}}\frac{\partial \bar{I}_1}{\partial F_{iA}} + \bar{K}_{12}\frac{\partial \bar{I}_2}{\partial F_{jB}}\frac{\partial \bar{I}_1}{\partial F_{iA}} + \bar{K}_1\frac{\partial^2 \bar{I}_1}{\partial F_{iA}\partial F_{jB}} \\ + \bar{K}_{21}\frac{\partial \bar{I}_1}{\partial F_{jB}}\frac{\partial \bar{I}_2}{\partial F_{iA}} + \bar{K}_{22}\frac{\partial \bar{I}_2}{\partial F_{jB}}\frac{\partial \bar{I}_2}{\partial F_{iA}} + \bar{K}_2\frac{\partial^2 \bar{I}_2}{\partial F_{iA}\partial F_{jB}}\end{aligned} \tag{C.6}$$

$$\frac{\partial^2 \tilde{W}}{\partial F_{jB}\partial F_{iA}} = \frac{\partial^2 \tilde{W}}{\partial J^2}J^2 F^{-1}_{Bj}F^{-1}_{Ai} + \frac{\partial \tilde{W}}{\partial J}J\left(F^{-1}_{Bj}F^{-1}_{Ai} - F^{-1}_{Aj}F^{-1}_{Bi}\right) \tag{C.7}$$

and

$$\begin{aligned}\frac{\partial^2 \bar{I}_1}{\partial F_{iA}\partial F_{jB}} = J^{-2/3}\Bigg[&-\frac{4}{3}F^{-1}_{Bj}F_{iA} + 2\delta_{ij}\delta_{AB} + \frac{4}{9}I_1 F^{-1}_{Ai}F^{-1}_{Bj} \\ &- \frac{4}{3}F_{jB}F^{-1}_{Ai} + \frac{2}{3}I_1 F^{-1}_{Aj}F^{-1}_{Bi}\Bigg]\end{aligned} \tag{C.8}$$

$$\begin{aligned}\frac{\partial^2 \bar{I}_2}{\partial F_{iA}\partial F_{jB}} = J^{-4/3}\Bigg[&-\frac{8}{3}F^{-1}_{Bj}I_1 F_{iA} + \frac{8}{3}F^{-1}_{Bj}F_{ip}G_{pA} \\ &+ \frac{16}{9}F^{-1}_{Bj}I_2 F^{-1}_{Ai} + 4F_{iA}F_{jB} + 2I_1\delta_{ij}\delta_{AB} \\ &- 2\delta_{ij}G_{BA} - 2F_{iB}F_{jA} - 2F_{ip}F_{jp}\delta_{AB} \\ &- \frac{8}{3}I_1 F_{jB}F^{-1}_{Ai} + \frac{8}{3}F_{jp}G_{pB}F^{-1}_{Ai} + \frac{4}{3}I_2 F^{-1}_{Aj}F^{-1}_{Bi}\Bigg]\end{aligned} \tag{C.9}$$

The tensor $\mathbb{C}^6_{iAjBkC}$ is given by:

$$\mathbb{C}^6_{iAjBkC} = \frac{\partial^3 W}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} = \frac{\partial^3 \bar{W}}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} + \frac{\partial^3 \tilde{W}}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} \tag{C.10}$$

where

$$\begin{aligned}\frac{\partial^3 \bar{W}}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} &= \sum_{\alpha=1}^{2}\sum_{\beta=1}^{2}\sum_{\gamma=1}^{2}\left(\frac{\partial^3 \bar{W}}{\partial \bar{I}_\alpha \partial \bar{I}_\beta \partial \bar{I}_\gamma}\frac{\partial \bar{I}_\alpha}{\partial F_{iA}}\frac{\partial \bar{I}_\beta}{\partial F_{jB}}\frac{\partial \bar{I}_\gamma}{\partial F_{kC}}\right) \\ &\quad + \sum_{\alpha=1}^{2}\sum_{\beta=1}^{2}\left(\frac{\partial^2 \bar{W}}{\partial \bar{I}_\alpha \partial \bar{I}_\beta}\frac{\partial \bar{I}_\alpha}{\partial F_{iA}}\frac{\partial^2 \bar{I}_\beta}{\partial F_{jB}\partial F_{kC}}\right. \\ &\qquad + \frac{\partial^2 \bar{W}}{\partial \bar{I}_\alpha \partial \bar{I}_\beta}\frac{\partial \bar{I}_\alpha}{\partial F_{jB}}\frac{\partial^2 \bar{I}_\beta}{\partial F_{iA}\partial F_{kC}} \\ &\qquad \left. + \frac{\partial^2 \bar{W}}{\partial \bar{I}_\alpha \partial \bar{I}_\beta}\frac{\partial \bar{I}_\alpha}{\partial F_{kC}}\frac{\partial^2 \bar{I}_\beta}{\partial F_{iA}\partial F_{jB}}\right) \\ &\quad + \sum_{\alpha=1}^{2}\frac{\partial \bar{W}}{\partial \bar{I}_\alpha}\frac{\partial^3 \bar{I}_\alpha}{\partial F_{kC}\partial F_{jB}\partial F_{iA}}\end{aligned} \tag{C.11}$$

$$\begin{aligned}\frac{\partial^3 \tilde{W}}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} &= \frac{\partial^3 \tilde{W}}{\partial J^3}J^3 F^{-1}_{Ai}F^{-1}_{Bj}F^{-1}_{Ck} \\ &\quad + \frac{\partial^2 \tilde{W}}{\partial J^2}J^2\left(3F^{-1}_{Ai}F^{-1}_{Bj}F^{-1}_{Ck} - F^{-1}_{Ai}F^{-1}_{Bk}F^{-1}_{Cj}\right. \\ &\qquad \left. - F^{-1}_{Ak}F^{-1}_{Bj}F^{-1}_{Ci} - F^{-1}_{Aj}F^{-1}_{Bi}F^{-1}_{Ck}\right) \\ &\quad + \frac{\partial \tilde{W}}{\partial J}J\left(F^{-1}_{Ai}F^{-1}_{Bj}F^{-1}_{Ck} + F^{-1}_{Ak}F^{-1}_{Bi}F^{-1}_{Cj} + F^{-1}_{Aj}F^{-1}_{Bk}F^{-1}_{Ci}\right. \\ &\qquad \left. - F^{-1}_{Aj}F^{-1}_{Bi}F^{-1}_{Ck} - F^{-1}_{Ai}F^{-1}_{Bk}F^{-1}_{Cj} - F^{-1}_{Ak}F^{-1}_{Bj}F^{-1}_{Ci}\right)\end{aligned} \tag{C.12}$$

and

$$\begin{aligned}\frac{\partial^3 \bar{I}_1}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} = -\frac{2}{3}J^{-2/3}F^{-1}_{Ck}\Bigg(&-\frac{4}{3}F^{-1}_{Bj}F_{iA} + 2\delta_{ij}\delta_{AB} + \frac{4}{9}I_1F^{-1}_{Ai}F^{-1}_{Bj} \\ &- \frac{4}{3}F_{jB}F^{-1}_{Ai} + \frac{2}{3}I_1F^{-1}_{Aj}F^{-1}_{Bi}\Bigg) \\ + J^{-2/3}\Bigg[&-\frac{4}{3}\Bigg(- F^{-1}_{Bk}F^{-1}_{Cj}F_{iA} + F^{-1}_{Bj}\delta_{ik}\delta_{AC}\Bigg) \\ &+ \frac{4}{9}\Bigg(2F^{-1}_{Ai}F^{-1}_{Bj}F_{kC} - I_1F^{-1}_{Ak}F^{-1}_{Bj}F^{-1}_{Ci} - I_1F^{-1}_{Ai}F^{-1}_{Bk}F^{-1}_{Cj}\Bigg) \\ &- \frac{4}{3}\Bigg(\delta_{jk}\delta_{BC}F^{-1}_{Ai} - F_{jB}F^{-1}_{Ak}F^{-1}_{Ci}\Bigg) \\ &+ \frac{2}{3}\Bigg(2F_{kC}F^{-1}_{Aj}F^{-1}_{Bi} - I_1F^{-1}_{Ak}F^{-1}_{Bi}F^{-1}_{Cj} - I_1F^{-1}_{Aj}F^{-1}_{Bk}F^{-1}_{Ci}\Bigg)\Bigg]\end{aligned} \tag{C.13}$$

$$\frac{\partial^3 \bar{I}_2}{\partial F_{kC}\partial F_{jB}\partial F_{iA}} = -\frac{4}{3}J^{-4/3}F^{-1}_{Ck}\mathbb{B} + J^{-4/3}\frac{\partial \mathbb{B}}{\partial F_{kC}} \tag{C.14}$$

here

$$\begin{aligned}\mathbb{B} = \Bigg[&-\frac{8}{3}F^{-1}_{Bj}I_1F_{iA} + \frac{8}{3}F^{-1}_{Bj}F_{ip}G_{pA} \\ &+ \frac{16}{9}F^{-1}_{Bj}I_2F^{-1}_{Ai} + 4F_{iA}F_{jB} + 2I_1\delta_{ij}\delta_{AB} \\ &- 2\delta_{ij}G_{BA} - 2F_{iB}F_{jA} - 2F_{ip}F_{jp}\delta_{AB} \\ &- \frac{8}{3}I_1F_{jB}F^{-1}_{Ai} + \frac{8}{3}F_{jp}G_{pB}F^{-1}_{Ai} + \frac{4}{3}I_2F^{-1}_{Aj}F^{-1}_{Bi}\Bigg]\end{aligned} \tag{C.15}$$

and

$$\frac{\partial \mathbb{B}}{\partial F_{kC}} = (i) + (ii) + (iii) + (iv) + (v) + (vi) + (vii) + (viii) + (ix) + (x) + (xi) \tag{C.16}$$

with:

$$(i) = -\frac{8}{3}\Big[- F^{-1}_{Bk}F^{-1}_{Cj}I_1F_{iA} + 2F_{kC}F^{-1}_{Bj}F_{iA} + F^{-1}_{Bj}I_1\delta_{ik}\delta_{AC}\Big] \tag{C.17a}$$

$$(ii) = \frac{8}{3}\Big[- F^{-1}_{Bk}F^{-1}_{Cj}F_{ip}G_{pA} + F^{-1}_{Bj}\delta_{ik}G_{CA} + F^{-1}_{Bj}F_{iC}F_{kA} + F^{-1}_{Bj}F_{ip}F_{kp}\delta_{AC}\Big] \tag{C.17b}$$

$$(iii) = \frac{16}{9}\Big[- I_2F^{-1}_{Bk}F^{-1}_{Cj}F^{-1}_{Ai} + 2F^{-1}_{Bj}I_1F_{kC}F^{-1}_{Ai} - 2F^{-1}_{Bj}F_{kp}G_{pC}F^{-1}_{Ai} - F^{-1}_{Bj}I_2F^{-1}_{Ak}F^{-1}_{Ci}\Big] \tag{C.17c}$$

$$(iv) = 4[\delta_{ik}\delta_{AC}F_{jB} + F_{iA}\delta_{jk}\delta_{BC}] \tag{C.17d}$$

$$(v) = 4F_{kC}\delta_{ij}\delta_{AB} \tag{C.17e}$$

$$(vi) = -2[\delta_{ij}\delta_{BC}F_{kA} + \delta_{ij}F_{kB}\delta_{AC}] \tag{C.17f}$$

$$(vii) = -2[\delta_{ik}\delta_{BC}F_{jA} + F_{iB}\delta_{jk}\delta_{AC}] \tag{C.17g}$$

$$(viii) = -2[F_{jC}\delta_{ik}\delta_{AB} + F_{iC}\delta_{jk}\delta_{AB}] \tag{C.17h}$$

$$(ix) = -\frac{8}{3}\Big[2F_{kC}F_{jB}F^{-1}_{Ai} + I_1\delta_{jk}\delta_{BC}F^{-1}_{Ai} - I_1F_{jB}F^{-1}_{Ak}F^{-1}_{Ci}\Big] \tag{C.17i}$$

$$(x) = \frac{8}{3}\Big[\delta_{jk}G_{CB}F^{-1}_{Ai} + F_{jC}F_{kB}F^{-1}_{Ai} + F_{jp}F_{kp}\delta_{BC}F^{-1}_{Ai} - F_{jp}G_{pB}F^{-1}_{Ak}F^{-1}_{Ci}\Big] \tag{C.17j}$$

$$(xi) = \frac{4}{3}\Big[2I_1F_{kC}F^{-1}_{Aj}F^{-1}_{Bi} - 2F_{kp}G_{pC}F^{-1}_{Aj}F^{-1}_{Bi} - I_2F^{-1}_{Ak}F^{-1}_{Cj}F^{-1}_{Bi} - I_2F^{-1}_{Aj}F^{-1}_{Bk}F^{-1}_{Ci}\Big] \tag{C.17k}$$

Similarly, the expression for the tensor $\mathbb{C}^8_{iAjBkClD}$ can be derived by taking the next consecutive differential of $\mathbb{C}^6_{iAjBkC}$:

$$\mathbb{C}^8_{iAjBkClD} = \frac{\partial \mathbb{C}^6_{iAjBkC}}{\partial F_{lD}} \tag{C.18}$$

References

[1] "Professor Shantanu Sinha, Department of Radiology, UCSD School of Medicine,"

[2] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[3] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Springer-Verlag New York Inc., 2003.

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[5] V. Caselles, F. Catté, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische mathematik*, vol. 66, no. 1, pp. 1–31, 1993.

[6] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[7] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[8] T. Chan and L. Vese, "An active contour model without edges," in *Scale-Space Theories in Computer Vision, Lecture notes in computer science 1682*, pp. 141–151, Springer, 1999.

[9] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[10] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

[11] T. F. Chan, B. Y. Sandberg, and L. A. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 130–141, 2000.

[12] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1940–1949, 2008.

[13] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Implicit active contours driven by local binary fitting energy," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pp. 1–7, IEEE, 2007.

[14] K. Zhang, H. Song, and L. Zhang, "Active contours driven by local image fitting energy," *Pattern Recognition*, vol. 43, no. 4, pp. 1199–1206, 2010.

[15] L. Wang, C. Li, Q. Sun, D. Xia, and C.-Y. Kao, "Active contours driven by local and global intensity fitting energy with application to brain MR image segmentation," *Computerized Medical Imaging and Graphics*, vol. 33, no. 7, pp. 520–531, 2009.

[16] D. Li, W. Li, and Q. Liao, "Active contours driven by local and global probability distributions," *Journal of Visual Communication and Image Representation*, vol. 24, no. 5, pp. 522–533, 2013.

[17] E. S. Brown, T. F. Chan, and X. Bresson, "Completely convex formulation of the chan-vese image segmentation model," *International Journal of Computer Vision*, vol. 98, no. 1, pp. 103–121, 2012.

[18] U. Vovk, F. Pernus, and B. Likar, "A review of methods for correction of intensity inhomogeneity in MRI," *IEEE Transactions on Medical Imaging*, vol. 26, no. 3, pp. 405–421, 2007.

[19] C. Li, R. Huang, Z. Ding, J. Gatenby, D. N. Metaxas, and J. C. Gore, "A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI," *Image Processing, IEEE Transactions on*, vol. 20, no. 7, pp. 2007–2016, 2011.

[20] J. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[21] M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, and M. B. Cuadra, "A review of atlas-based segmentation for magnetic resonance brain images," *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, pp. e158–e177, 2011.

[22] P.-Y. Baudin, N. Azzabou, P. G. Carlier, and N. Paragios, "Automatic skeletal muscle segmentation through random walks and graph-based seed placement," in *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*, pp. 1036–1039, IEEE, 2012.

[23] P.-Y. Baudin, N. Azzabou, P. G. Carlier, and N. Paragios, "Prior knowledge, random walks and human skeletal muscle segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 569–576, Springer, 2012.

[24] P.-Y. Baudin, N. Azzabou, P. G. Carlier, N. Paragios, *et al.*, "Manifold-enhanced segmentation through random walks on linear subspace priors," in *Proceedings of the British Machine Vision Conference*, 2012.

[25] J. W. Prescott, M. S. Swanson, K. Powell, M. N. Gurcan, F. Haq, T. M. Best, and R. D. Jackson, "Template-based level set segmentation using anatomical information," in *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*, pp. 24–29, IEEE, 2009.

[26] J. W. Prescott, T. M. Best, M. S. Swanson, F. Haq, R. D. Jackson, and M. N. Gurcan, "Anatomically anchored template-based level set segmentation: application to quadriceps muscles in MR images from the osteoarthritis initiative," *Journal of digital imaging*, vol. 24, no. 1, pp. 28–43, 2011.

[27] R. Neji, G. Fleury, J.-F. Deux, A. Rahmouni, G. Bassez, A. Vignaud, and N. Paragios, "Support vector driven markov random fields towards DTI segmentation of the human skeletal muscle," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pp. 923–926, IEEE, 2008.

[28] S. Essafi, G. Langs, J.-F. Deux, A. Rahmouni, G. Bassez, and N. Paragios, "Wavelet-driven knowledge-based mri calf muscle segmentation," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pp. 225–228, IEEE, 2009.

[29] S. Essafi, G. Langs, and N. Paragios, "Hierarchical 3D diffusion wavelet shape priors," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1717–1724, IEEE, 2009.

[30] B. Gilles and N. Magnenat-Thalmann, "Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations," *Medical Image Analysis*, vol. 14, no. 3, pp. 291–302, 2010.

[31] S. Andrews, G. Hamarneh, A. Yazdanpanah, B. HajGhanbari, and W. D. Reid, "Probabilistic multi-shape segmentation of knee extensor and flexor muscles," in

Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011, pp. 651–658, Springer, 2011.

[32] C. Wang, O. Teboul, F. Michel, S. Essafi, and N. Paragios, "3D knowledge-based segmentation using pose-invariant higher-order graphs," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pp. 189–196, Springer, 2010.

[33] D. Le Bihan, J.-F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriat, "Diffusion tensor imaging: concepts and applications," *Journal of Magnetic Resonance Imaging*, vol. 13, no. 4, pp. 534–546, 2001.

[34] Y. Assaf and O. Pasternak, "Diffusion tensor imaging (DTI) based white matter mapping in brain research: a review," *Journal of Molecular Neuroscience*, vol. 34, no. 1, pp. 51–61, 2008.

[35] A. M. Heemskerk, G. J. Strijkers, A. Vilanova, M. R. Drost, and K. Nicolay, "Determination of mouse skeletal muscle architecture using three-dimensional diffusion tensor imaging," *Magnetic Resonance in Medicine*, vol. 53, no. 6, pp. 1333–1340, 2005.

[36] S. Sinha, U. Sinha, and V. R. Edgerton, "In vivo diffusion tensor imaging of the human calf muscle," *Journal of Magnetic Resonance Imaging*, vol. 24, no. 1, pp. 182–190, 2006.

[37] M. Froeling, A. J. Nederveen, D. F. Heijtel, A. Lataster, C. Bos, K. Nicolay, M. Maas, M. R. Drost, and G. J. Strijkers, "Diffusion-tensor MRI reveals the complex muscle architecture of the human forearm," *Journal of Magnetic Resonance Imaging*, vol. 36, no. 1, pp. 237–248, 2012.

[38] M. Drost, M. Froeling, P. Oomen, A. Nederveen, and G. Strijkers, "Diffusion tensor MRI as tool to determine subject specific muscle architecture," in *IUTAM symposium on human movement analysis and simulation*, 2010.

[39] S. Blemker, A. Sherbondy, D. Akers, R. Bammer, S. Delp, and G. Gold, "Characterization of skeletal muscle fascicle arrangements using diffusion tensor tractography," in *Proceedings of the 13th Annual Meeting of ISMRM, Miami Beach, FL, USA*, 2005.

[40] L. Zhukov, R. Whitaker, K. Museth, D. Breen, and A. H. Barr, "Level set modeling and segmentation of diffusion tensor magnetic resonance imaging brain data," *Journal of Electronic Imaging*, vol. 12, no. 1, pp. 125–133, 2003.

[41] T. Liu, H. Li, K. Wong, A. Tarokh, L. Guo, and S. T. Wong, "Brain tissue segmentation based on DTI data," *NeuroImage*, vol. 38, no. 1, pp. 114–123, 2007.

[42] Z. Wang and B. C. Vemuri, "Tensor field segmentation using region based active contour model," in *Computer Vision-ECCV 2004*, pp. 304–315, Springer, 2004.

[43] Z. Wang and B. C. Vemuri, "DTI segmentation using an information theoretic tensor dissimilarity measure," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 10, pp. 1267–1277, 2005.

[44] Z. Wang and B. C. Vemuri, "An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–228, IEEE, 2004.

[45] C. Feddern, J. Weickert, and B. Burgeth, "Level-set methods for tensor-valued images," in *Proc. Second IEEE Workshop on Geometric and Level Set Methods in Computer Vision*, pp. 65–72, 2003.

[46] M. Rousson, C. Lenglet, and R. Deriche, "Level set and region based surface propagation for diffusion tensor MRI segmentation," in *Computer vision and mathematical methods in medical and biomedical image analysis*, pp. 123–134, Springer, 2004.

[47] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly notices of the royal astronomical society*, vol. 181, no. 3, pp. 375–389, 1977.

[48] J. J. Monaghan, "Why particle methods work," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 4, pp. 422–433, 1982.

[49] J. W. Swegle, D. L. Hicks, and S. W. Attaway, "Smoothed particle hydrodynamics stability analysis," *Journal of Computational Physics*, vol. 116, no. 1, pp. 123–134, 1995.

[50] B. Nayroles, G. Touzot, and P. Villon, "Generalizing the finite element method: diffuse approximation and diffuse elements," *Computational Mechanics*, vol. 10, no. 5, pp. 307–318, 1992.

[51] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," *Mathematics of Computation*, vol. 37, no. 155, pp. 141–158, 1981.

[52] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994.

[53] W. K. Liu, S. Jun, and Y. F. Zhang, "Reproducing kernel particle methods," *International Journal for Numerical Methods in Fluids*, vol. 20, no. 8-9, pp. 1081–1106, 1995.

[54] J.-S. Chen, C. Pan, C.-T. Wu, and W. K. Liu, "Reproducing kernel particle methods for large deformation analysis of non-linear structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1, pp. 195–227, 1996.

[55] J.-S. Chen, C. Pan, and C.-T. Wu, "Large deformation analysis of rubber based on a reproducing kernel particle method," *Computational Mechanics*, vol. 19, no. 3, pp. 211–227, 1997.

[56] J.-S. Chen, S. Yoon, H.-P. Wang, and W. K. Liu, "An improved reproducing kernel particle method for nearly incompressible finite elasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 181, no. 1, pp. 117–145, 2000.

[57] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless methods: an overview and recent developments," *Computer methods in applied mechanics and engineering*, vol. 139, no. 1, pp. 3–47, 1996.

[58] S. Beissel and T. Belytschko, "Nodal integration of the element-free galerkin method," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1, pp. 49–74, 1996.

[59] J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You, "A stabilized conforming nodal integration for galerkin mesh-free methods," *International Journal for Numerical Methods in Engineering*, vol. 50, no. 2, pp. 435–466, 2001.

[60] J. Dolbow and T. Belytschko, "Numerical integration of the Galerkin weak form in meshfree methods," *Computational mechanics*, vol. 23, no. 3, pp. 219–230, 1999.

[61] J.-S. Chen, M. Hillman, and M. Rüter, "An arbitrary order variationally consistent integration for Galerkin meshfree methods," *International Journal for Numerical Methods in Engineering*, vol. 95, no. 5, pp. 387–418, 2013.

[62] T. Belytschko, Y. Guo, W. K. Liu, and S. P. Xiao, "A unified stability analysis of meshless particle methods," *International Journal for Numerical Methods in Engineering*, vol. 48, no. 9, pp. 1359–1400, 2000.

[63] C. Dyka, P. Randles, and R. Ingel, "Stress points for tension instability in SPH," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 13, pp. 2325–2341, 1997.

[64] P. Randles and L. Libersky, "Normalized SPH with stress points," *International Journal for Numerical Methods in Engineering*, vol. 48, no. 10, pp. 1445–1462, 2000.

[65] D. Wang and J.-S. Chen, "Locking-free stabilized conforming nodal integration for meshfree Mindlin-Reissner plate formulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 12, pp. 1065–1083, 2004.

[66] J.-S. Chen and D. Wang, "A constrained reproducing kernel particle formulation for shear deformable shell in Cartesian coordinates," *International Journal for Numerical Methods in Engineering*, vol. 68, no. 2, pp. 151–172, 2006.

[67] D. Wang and J.-S. Chen, "A hermite reproducing kernel approximation for thin-plate analysis with sub-domain stabilized conforming integration," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 3, pp. 368–390, 2008.

[68] J.-S. Chen, S. Yoon, and C.-T. Wu, "Non-linear version of stabilized conforming nodal integration for galerkin mesh-free methods," *International Journal for Numerical Methods in Engineering*, vol. 53, no. 12, pp. 2587–2615, 2002.

[69] P. C. Guan, S. W. Chi, J. S. Chen, T. R. Slawson, and M. J. Roth, "Semi-lagrangian reproducing kernel particle method for fragment-impact problems," *International Journal of Impact Engineering*, vol. 38, no. 12, pp. 1033–1047, 2011.

[70] J. Chen, W. Hu, M. Puso, Y. Wu, and X. Zhang, "Strain smoothing for stabilization and regularization of Galerkin meshfree methods," in *Meshfree Methods for Partial Differential Equations III*, pp. 57–75, Springer, 2007.

[71] M. Puso, J.-S. Chen, E. Zywicz, and W. Elmer, "Meshfree and finite element nodal integration methods," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 3, pp. 416–446, 2008.

[72] M. Hillman and J. S. Chen, "An accelerated, convergent and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics," *Article submitted to: International Journal of Numerical Methods in Engineering*, 2015.

[73] S.-W. Chi, J.-S. Chen, H.-Y. Hu, and J. P. Yang, "A gradient reproducing kernel collocation method for boundary value problems," *International Journal for Numerical Methods in Engineering*, vol. 93, no. 13, pp. 1381–1402, 2013.

[74] S. Fernández-Méndez and A. Huerta, "Imposing essential boundary conditions in mesh-free methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 12, pp. 1257–1275, 2004.

[75] J.-S. Chen and H.-P. Wang, "New boundary condition treatments in meshfree computation of contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 187, no. 3, pp. 441–468, 2000.

[76] T. Zhu and S. Atluri, "A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free galerkin method," *Computational Mechanics*, vol. 21, no. 3, pp. 211–222, 1998.

[77] T. Belytschko, D. Organ, and Y. Krongauz, "A coupled finite element – element free galerkin method," *Computational Mechanics*, vol. 17, no. 3, pp. 186–195, 1995.

[78] H.-Y. Hu, J.-S. Chen, and W. Hu, "Weighted radial basis collocation method for boundary value problems," *International Journal for Numerical Methods in Engineering*, vol. 69, no. 13, pp. 2736–2757, 2007.

[79] H.-Y. Hu, J.-S. Chen, and W. Hu, "Error analysis of collocation method based on reproducing kernel approximation," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 3, pp. 554–580, 2011.

[80] E. J. Kansa, "Multiquadrics – A scattered data approximation scheme with applications to computational fluid-dynamics – I surface approximations and partial derivative estimates," *Computers & Mathematics with applications*, vol. 19, no. 8, pp. 127–145, 1990.

[81] E. J. Kansa, "Multiquadrics – A scattered data approximation scheme with applications to computational fluid-dynamics – II solutions to parabolic, hyperbolic and elliptic partial differential equations," *Computers & Mathematics with Applications*, vol. 19, no. 8, pp. 147–161, 1990.

[82] N. Aluru, "A point collocation method based on reproducing kernel approximations," *International Journal for Numerical Methods in Engineering*, vol. 47, no. 6, pp. 1083–1121, 2000.

[83] H.-Y. Hu, C.-K. Lai, and J.-S. Chen, "A study on convergence and complexity of reproducing kernel collocation method," *National Science Council Tunghai University Endowment Fund for Academic Advancement Mathematics Research Promotion Center*, pp. 189–212, 2009.

[84] J.-S. Chen, W. Hu, and H.-Y. Hu, "Reproducing kernel enhanced local radial basis collocation method," *International Journal for Numerical Methods in Engineering*, vol. 75, no. 5, pp. 600–627, 2008.

[85] J. A. Weiss, B. N. Maker, and S. Govindjee, "Finite element implementation of incompressible, transversely isotropic hyperelasticity," *Computer methods in applied mechanics and engineering*, vol. 135, no. 1, pp. 107–128, 1996.

[86] T. Johansson, P. Meier, and R. Blickhan, "A finite-element model for the mechanical analysis of skeletal muscles," *Journal of Theoretical Biology*, vol. 206, no. 1, pp. 131–149, 2000.

[87] C. Oomens, M. Maenhout, C. Van Oijen, M. Drost, and F. Baaijens, "Finite element modelling of contracting skeletal muscle," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1437, pp. 1453–1460, 2003.

[88] S. S. Blemker, P. M. Pinsky, and S. L. Delp, "A 3D model of muscle reveals the causes of nonuniform strains in the biceps brachii," *Journal of biomechanics*, vol. 38, no. 4, pp. 657–665, 2005.

[89] S.-W. Chi, J. Hodgson, J.-S. Chen, V. R. Edgerton, D. D. Shin, R. A. Roiz, and S. Sinha, "Finite element modeling reveals complex strain mechanics in the aponeuroses of contracting skeletal muscle," *Journal of Biomechanics*, vol. 43, no. 7, pp. 1243–1250, 2010.

[90] J. A. Hodgson, S.-W. Chi, J. P. Yang, J.-S. Chen, V. R. Edgerton, and S. Sinha, "Finite element modeling of passive material influence on the deformation and force output of skeletal muscle," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 9, pp. 163–183, 2012.

[91] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control.," *Critical reviews in biomedical engineering*, vol. 17, no. 4, pp. 359–411, 1988.

[92] S. S. Blemker and S. L. Delp, "Rectus femoris and vastus intermedius fiber excursions predicted by three-dimensional muscle models," *Journal of biomechanics*, vol. 39, no. 8, pp. 1383–1391, 2006.

[93] S. S. Blemker and S. L. Delp, "Three-dimensional representation of complex muscle architectures and geometries," *Annals of biomedical engineering*, vol. 33, no. 5, pp. 661–673, 2005.

[94] A. K. Noor and J. M. Peters, "Reduced basis technique for nonlinear analysis of structures," *AIAA Journal*, vol. 18, no. 4, pp. 455–462, 1980.

[95] R. E. Nickell, "Nonlinear dynamics by mode superposition," *Computer Methods in Applied Mechanics and Engineering*, vol. 7, no. 1, pp. 107–129, 1976.

[96] J. A. Stricklin and W. E. Haisler, "Formulations and solution procedures for nonlinear structural analysis," *Computers & Structures*, vol. 7, no. 1, pp. 125–136, 1977.

[97] V. Shah, G. Bohm, and A. Nahavandi, "Modal superposition method for computationally economical nonlinear structural analysis," *Journal of Pressure Vessel Technology*, vol. 101, no. 2, pp. 134–141, 1979.

[98] J. Thompson and A. Walker, "The non-linear perturbation analysis of discrete structural systems," *International Journal of Solids and Structures*, vol. 4, no. 8, pp. 757–768, 1968.

[99] A. Walker, "A non-linear finite element analysis of shallow circular arches," *International Journal of Solids and Structures*, vol. 5, no. 2, pp. 97–107, 1969.

[100] A. K. Noor, "Recent advances in reduction methods for nonlinear problems," *Computers & Structures*, vol. 13, no. 1, pp. 31–44, 1981.

[101] E. L. Wilson, M.-W. Yuan, and J. M. Dickens, "Dynamic analysis by direct superposition of Ritz vectors," *Earthquake Engineering & Structural Dynamics*, vol. 10, no. 6, pp. 813–821, 1982.

[102] S. R. Idelsohn and A. Cardona, "A reduction method for nonlinear structural dynamic analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 49, no. 3, pp. 253–279, 1985.

[103] S. R. Idelsohn and A. Cardona, "A load-dependent basis for reduced nonlinear structural dynamics," *Computers & Structures*, vol. 20, no. 1, pp. 203–210, 1985.

[104] J. Engblom and C. Chang, "Nonlinear dynamical response of impulsively loaded structures - a reduced basis approach," *AIAA Journal*, vol. 29, no. 4, pp. 613–618, 1991.

[105] P. Krysl, S. Lall, and J. Marsden, "Dimensional model reduction in non-linear finite element dynamics of solids and structures," *International Journal for Numerical Methods in Engineering*, vol. 51, no. 4, pp. 479–504, 2001.

[106] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, 2004.

[107] S. Chaturantabut and D. C. Sorensen, "Discrete empirical interpolation for nonlinear model reduction," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 4316–4321, IEEE, 2009.

[108] K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 2, pp. 155–181, 2011.

[109] A. C. Antoulas, *Approximation of large-scale dynamical systems*, vol. 6. Siam, 2005.

[110] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.

[111] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Springer-Verlag New York Inc., 2003.

[112] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM Journal on Numerical analysis*, vol. 40, no. 2, pp. 492–515, 2002.

[113] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *JOSA A*, vol. 4, no. 3, pp. 519–524, 1987.

[114] E. N. Lorenz, "Empirical orthogonal functions and statistical weather prediction," 1956.

[115] R. Pinnau, "Model reduction via proper orthogonal decomposition," *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13, pp. 95–109.

[116] G. Kerschen, J.-c. Golinval, A. F. Vakakis, and L. A. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview," *Nonlinear dynamics*, vol. 41, no. 1-3, pp. 147–169, 2005.

[117] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in MR images: method and validation," *Medical Imaging, IEEE Transactions on*, vol. 13, no. 4, pp. 716–724, 1994.

[118] A. M. Neto, A. C. Victorino, I. Fantoni, D. E. Zampieri, J. V. Ferreira, and D. A. Lima, "Image processing using pearson's correlation coefficient: Applications on autonomous robotics," in *13th International Conference on Mobile Robots and Competitions (Robotica)*, pp. 14–19, 2013.

[119] J.-S. Chen, C. Pan, C. Roque, and H.-P. Wang, "A lagrangian reproducing kernel particle method for metal forming analysis," *Computational Mechanics*, vol. 22, no. 3, pp. 289–307, 1998.

[120] J. G. Tidball, "Myotendinous junction: morphological changes and mechanical failure associated with muscle cell atrophy," *Experimental and Molecular Pathology*, vol. 40, no. 1, pp. 1–12, 1984.

[121] R. Csapo, V. Malis, U. Sinha, J. Du, and S. Sinha, "Age-associated differences in triceps surae muscle composition and strength – an MRI-based cross-sectional comparison of contractile, adipose and connective tissue," *BMC musculoskeletal disorders*, vol. 15, no. 1, p. 209, 2014.

[122] W. Han and X. Meng, "Error analysis of the reproducing kernel particle method," *Computer methods in applied mechanics and engineering*, vol. 190, no. 46, pp. 6157–6181, 2001.

[123] S. K. Abd-El-Hafiz, G. A. Ismail, B. S. Matit, and A. F. S. Heliopolis, "A numerical technique for the 3-D poisson equation," *International Journal of Pure and Applied Mathematics*, vol. 7, pp. 263–270, 2003.

[124] J.-S. Chen and C. Pan, "A pressure projection method for nearly incompressible rubber hyperelasticity, Part I: Theory," *Journal of Applied Mechanics*, vol. 63, no. 4, pp. 862–868, 1996.

[125] J.-S. Chen, C.-T. Wu, and C. Pan, "A pressure projection method for nearly incompressible rubber hyperelasticity, Part II: Applications," *Journal of Applied Mechanics*, vol. 63, no. 4, pp. 869–876, 1996.

[126] R. S. Rivlin, "Large elastic deformations of isotropic materials. VI. further results in the theory of torsion, shear and flexure," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 242, no. 845, pp. 173–195, 1949.

[127] R. S. Rivlin and D. Saunders, "Large elastic deformations of isotropic materials. VII. experiments on the deformation of rubber," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 243, no. 865, pp. 251–288, 1951.

[128] G. A. Holzapfel, *Nonlinear solid mechanics*, vol. 24. 2000.

[129] U. Sinha, V. Malis, R. Csapo, A. Moghadasi, R. Kinugasa, and S. Sinha, "Age-related differences in strain rate tensor of the medial gastrocnemius muscle during passive plantarflexion and active isometric contraction using velocity encoded mr imaging: Potential index of lateral force transmission," *Magnetic Resonance in Medicine*, vol. 73, no. 5, pp. 1852–1863, 2015.

[130] S. W. Chi, J. S. Chen, H. Luo, H. Y. Hu, and L. Wang, "Dispersion and stability properties of radial basis collocation method for elastodynamics," *Numerical Methods for Partial Differential Equations*, vol. 29, no. 3, pp. 818–842, 2013.

[131] S.-W. Chi, J.-S. Chen, and H.-Y. Hu, "A weighted collocation on the strong form with mixed radial basis approximations for incompressible linear elasticity," *Computational Mechanics*, vol. 53, no. 2, pp. 309–324, 2014.