Abstract

LANKALAPALLI, SHRINIVAS. Optimal Pick-up Locations for Transport and Handling of Limp Materials. (Under the direction of Dr. Jeffrey W. Eischen)

Optimization procedures are developed to solve for pick-up locations that minimize a measure of the deformation of flat limp parts. The limp parts are modeled as shells undergoing large deformations and rotations using a geometrically exact nonlinear shell finite element formulation. The strain energy is the objective function minimized and is computed from a finite element solution for the deformation of the limp part. The optimal locations are found by solving the corresponding bound constrained minimization problem using gradient based algorithms. The ability to automatically mesh the domain of the limp part determines whether the problem can be solved as a continuous problem. Meshes for one dimensional and square/rectangular two dimensional domains are automatically generated and the optimal locations are solved using continuous optimization algorithms based on the Broyden, Fletcher, Goldfarb and Shanno (BFGS) update of the Hessian. Optimal locations for all other two dimensional domains are obtained by solving a discrete optimization problem on a fixed mesh by a new gradient based algorithm. Results are obtained for one dimensional strips and various two dimensional shapes of limp material. These include shapes obtained from apparel and datawear products.

Optimal Pick-up Locations for Transport and Handling of Limp Materials

by
Shrinivas Lankalapalli

A Thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Department of Mechanical and Aerospace Engineering

Raleigh

September 2001

APPROVED BY:

Prof. Larry M. Silverberg

Prof. Mohammad A. Zikry

Prof. Timothy G. Clapp

Prof. Jeffrey W. Eischen
Chair of Advisory Committee

Dedication

This work is dedicated to my father, Dr. L. V. Gangadhara Rao and mother L. Sakuntala.

Acknowledgements

I would like to thank my advisor, Dr. Eischen for his guidance, encouragement and patience during the course of this work. I would also like to thank my committee members Dr. Clapp, Dr. Silverberg and Dr. Zikry for the many useful discussions during the course of this work. A special thanks to Dr. Tim Kelley, Department of Mathematics for useful discussions related to optimization methods.

I am grateful to the National Textile Center for supporting this research through a grant and the Department of Mechanical & Aerospace Engineering for supporting me initially through a Teaching Assistantship. I would like to acknowledge the excellent computing, support and training facilities at the North Carolina Supercomputing Center (NCSC). The results in this thesis could not have been obtained without the computing resources at NCSC.

I thank Claudia Deaton, Apparel Lab, College of Textiles for helping me with the measurement of fabric material properties and Shelly Cooper, Medcovers for providing fabric samples.

Finally, I thank my parents, sister and brother for thier support and encouragement throughout my graduate studies.

Contents

List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Literature Review	2
1.2.1 Automated Handling Systems	2
1.2.2 Modeling and Simulation Related to Limp Material Handling .	3
1.2.3 Optimal Locations	3
1.3 Summary of Work Done	4
1.4 Organization of Thesis	5
2 Modeling of Limp Materials	9
2.1 Introduction	9
2.2 Nonlinear Shell Formulation	10
2.3 Adaptive Arc-length Control Algorithm	15
2.3.1 Derivation	15
2.3.2 Modification for fixed loading	19
2.3.3 Adapting the Arc-length	20
2.3.4 Implementation	21
2.4 Strain Energy Computation	22
2.4.1 Validation of Method	22
2.4.1.1 Analytical Expressions	22
2.4.1.2 Comparison of Numerically Computed Values	24
2.5 Summary	25

3	Algorithms	27
3.1	Introduction	27
3.2	Minimization Problem	28
3.3	Meshing Issues	28
3.3.1	Automatic Meshing for a One Dimensional Domain	29
3.3.2	Automatic Meshing for Square/Rectangular domains	31
3.4	Gradient Computation	36
3.5	Iterative Algorithms for Minimization	36
3.5.1	Unconstrained Minimization by BFGS Method	36
3.5.2	Minimization with Simple Bound Constraints	38
3.5.3	Gradient Based Method for a Fixed Mesh	40
3.6	Summary	42
4	Measurement of Fabric Material Properties	43
4.1	Introduction	43
4.2	Measurements by the FAST System	44
4.3	Fabric Material Properties	45
4.4	Summary	45
5	Optimal Locations for 1 Dimensional Domains	47
5.1	Introduction	47
5.2	Beams Undergoing Small Deformations	48
5.2.1	Numerical Solution Procedure	49
5.2.2	Results	50
5.3	Beams Undergoing Large Deformations	57
5.3.1	Numerical Solution Procedure	57
5.3.2	Results	60
5.4	Fabric Strips	66
5.5	Summary	66
6	Optimal Locations for 2 Dimensional Domains	70
6.1	Introduction	70
6.2	4 Pick-up Locations for Square/Rectangular Domains	71
6.2.1	Small deformations	72
6.2.1.1	Solution as a Continuous Problem	72

6.2.1.2	Solution on a Fixed Mesh with Iterative Mesh Refinement	76
6.2.2	Effect of γ and l/t ratio on optimal locations for a square domain	80
6.2.3	Effect of Aspect Ratio on Optimal Locations	83
6.2.4	Optimal Locations for Square/Rectangular Fabric Parts	84
6.3	3 Pick-up Locations for Square domain	86
6.4	4 Pick-up Locations for a Circular Domain	88
6.5	3 Pick-up Locations for a Circular Domain	91
6.6	Irregular Domains	93
6.6.1	Shirt Front	93
6.6.2	Short Front	93
6.6.3	4079back	102
6.6.4	4079frfc	102
7	Conclusions	110
7.1	Scope for future work	111
	References	113

List of Figures

1.1	One dimensional and two dimensional limp parts picked up at different locations	7
1.2	Pick-up locations for limp parts	8
2.1	Configuration of the shell	10
2.2	Effective Stress Resultants Acting on the Shell Mid-Surface	14
2.3	Solution by Arc-length Control	18
2.4	Area under a nodal load deflection curve	21
2.5	Plate Meshes (Symmetry employed)	26
3.1	Meshes generated by automatic meshing procedure for a 1-D domain. Dots indicate the nodes where boundary conditions are to be applied.	30
3.2	4 pick-up locations on a rectangular domain	31
3.3	Symmetry used to solve for 1 location on quarter rectangle	32
3.4	Meshes generated for rectangular domains	35
3.5	Node 3 is the closest node in the steepest descent direction	42
4.1	Cantilever Bending Test	44
5.1	Beam supported at n positions	48
5.2	Strain energy for 2 Supports	52
5.3	Deflection curves at optimal locations (small deformations)	54
5.4	Deflection curves at optimal locations (small deformations)	55
5.5	Plot of objective function v/s number of iterations	56
5.6	Plot of gradient norm v/s number of iterations	56
5.7	Deflection curves for 2 supports for different values of α (large deformations)	63

5.8	Deflection curves for 3 supports for different values of α (large deformations)	64
5.9	Deflection curves for 4 supports for different values of α (large deformations)	65
5.10	2 Pick-up locations for a Blue Denim strip	67
5.11	3 Pick-up locations for a Cotton Twill strip	68
5.12	4 Pick-up locations for a Cotton Polyester strip	69
6.1	4 pick-up locations on a rectangular domain	72
6.2	Iteration histories of different initial guesses when solving for 1 pick-up location on a quarter square.	76
6.3	Symmetry used to solve for one location on a quarter square	78
6.4	Iteration histories for 4 Pick-up locations for a square domain (coarse mesh)	78
6.5	Iteration histories for 4 Pick-up locations for a square domain (fine mesh)	79
6.6	Deformed shape at Optimal Locations	79
6.7	Effect of γ and l/t ratio on x_1/l	82
6.8	Effect of γ and l/t ratio on y_1/l	82
6.9	Effect of the aspect ratio on x_1/a and y_1/b for blue denim fabric	83
6.10	Deformed shape of 30cm x 30cm cotton twill fabric part	85
6.11	Deformed shape of 40cm x 24cm blue denim fabric part	85
6.12	Iteration histories for 3 Pick-up locations for a square domain	87
6.13	Deformed shape at Optimal Locations	87
6.14	Iteration histories for 4 Pick-up locations for a circular domain	89
6.15	Deformed shape at Optimal Locations	89
6.16	Iteration histories for 4 Pick-up locations for a circular blue denim part	90
6.17	Deformed shape at Optimal Locations	90
6.18	Iteration histories for 3 Pick-up locations for a circular domain	92
6.19	Deformed shape at Optimal Locations	92
6.20	Shapes of Shirt and Short parts	94
6.21	Mesh with iteration histories for SHIRT part	96
6.22	Deformed shape of SHIRT part at Optimal Locations	97
6.23	Mesh with iteration histories for SHORT part	98
6.24	Deformed shape of SHORT part at Optimal Locations	99
6.25	Symbol 4340 hand held computer	100

6.26	Shapes of Two Parts	101
6.27	Mesh with iteration histories for 4079back	104
6.28	Deformed shape of 4079back at Optimal Locations	105
6.29	Mesh with iteration histories for 4079frfc	106
6.30	Deformed shape of 4079frfc at Optimal Locations	107
6.31	Mesh with iteration histories for 4079frfc	108
6.32	Deformed shape of 4079frfc at Optimal Locations	109

List of Tables

2.1	Numerically and Analytically computed Plate Strain Energies	25
4.1	Fabric Material Properties	46
5.1	Optimal Locations for Small Deformations	53
5.2	Optimal Locations for 2 Supports (Large Deformations)	61
5.3	Optimal Locations for 3 Supports (Large Deformations)	61
5.4	Optimal Locations for 4 Supports (Large Deformations)	62
6.1	Coordinates to which different initial guesses converge for the mesh in Figure 1.5	77
6.2	Solution with different initial guesses and discretizations for $r=500$ and $l/t = 500$	80
6.3	Coordinates to which different initial guesses converge	86
6.4	Coordinates to which different initial guesses converge for shirt front .	95
6.5	Coordinates to which different initial guesses converge for short front	95
6.6	Coordinates to which different initial guesses converge for 4079back .	103
6.7	Coordinates to which different initial guesses converge for 4079frfc . .	103

Chapter 1

Introduction

1.1 Motivation and Objectives

Limp materials are used in many economically important industries such as textiles, aerospace, automobiles and leather. Most of the tasks involving the handling of limp materials are done manually which make them labor intensive and time consuming. Automatic handling systems that can be reprogrammed to perform a different task in relation to rigid objects are readily available. These are often robot systems and the reprogramming involves defining a new set of end-effector trajectories than can be repeated. Application of similar systems for handling limp materials will reduce the time associated with manual handling, lower cost and increase quality thus resulting in increased productivity. This, however is a challenging task primarily due to the low bending stiffness of limp materials which makes them easily susceptible to large deformations and rotations. As a result, limp parts can easily change shape during handling. In addition, there are huge variations in bending, friction and tensile properties of different limp materials which are further influenced by environmental conditions. The aforementioned unique properties and behavior of limp materials presents numerous problems when applying sensory robotics to automate their handling.

A large percentage of limp materials used in industries are flat, like fabrics, composites, metal foils, polymer sheets and leather. This work addresses issues related to transport and handling of such flat limp materials. An example is in the apparel manufacturing industry where most products start as various stacks of cloth that are first layed up for cutting which is followed by ply separation. Then, individual parts are transported and presented to suitable equipment for joining to make the product. Although these tasks are repetitive, they are mostly done manually. It has been noted by Taylor [1, 2] that about 80 % of production time in a sewing room is

spent handling the workpieces as opposed to joining. Automation of operations such as ply separation, transport, orientation, pick and place and joining is anticipated to increase competitiveness by increasing flexibility and decreasing response times. The above mentioned transport and handling tasks are common to manufacturing operations involving most limp materials.

The pick and place operation is one of the most basic and necessary task. Automation of this operation can be done using robots equipped with suitable end effectors and sensors. Limp parts undergo large deformations with the amount of deformation depending on where they are picked (see Figure 1.1). It is essential that the picked up parts undergo minimum deformation when being manipulated by robots. In general, limp parts are of different shapes and their behavior is greatly influenced by their physical properties. Hence, the objective of this research is to determine optimal locations of pick-up points that facilitate ease of handling of arbitrarily shaped limp parts (see Figure 1.2). The limp parts are modeled as shells undergoing large deformations and rotations and the optimal locations are found by solving a optimization problem in which a measure of the deformation as a function of pick-up locations is minimized. A priori knowledge of pick-up locations for parts of different shapes is anticipated to reduce the sensory requirements and facilitate offline programming of robots for limp material handling.

1.2 Literature Review

A review of literature was conducted in three key areas of relevance to this research : Automated handling systems, modeling and simulation related to limp material handling and determination of optimal locations.

1.2.1 Automated Handling Systems

Early research focussed on the development of retrieval devices for destacking operations in apparel manufacturing where a single ply of fabric is to be separated from a stack. Various robotic end-effectors to grip fabric parts were developed. These include a gripper based on the pin and adhesive concept by Parker et al [4], an electrostatic gripper by Taylor et al [5] and a flat-surface gripper based on the operational principle of suction and pressure differential by Kolluru et al [6]. Karakerezi et al [7] provide a literature survey of various gripping mechanisms and principles that have

been employed for gripping flat non-rigid materials.

Vision-guided robotic fabric manipulation was developed and implemented by Torgerson et al [8]. They generated robot motion paths from visual information of fabric edges and demonstrated manipulation of polygonal and nonpolygonal fabric parts. More recently, some automated robotic handling systems have been developed. Czarnecki [9] developed a robotic handling cell for garment manufacture in which garment piece parts are separated from a multi-ply stack and loaded onto a hanger. Fahantidis et al [10] developed a robotic system incorporating vision and force/torque sensing for handling flat textile materials. They presented experimental results for the tasks of grasping, folding, laying and sweeping of fabric parts.

1.2.2 Modeling and Simulation Related to Limp Material Handling

Modeling and simulation of limp material handling operations is of great use in the design optimization of manufacturing lines to adapt to limp materials of different physical properties and shapes. In addition, simulation results can facilitate offline programming of robots thus leading to flexible automation. Eischen et al [11] optimized fabric manipulation during pick and place operations using large displacement beam theory and finite elements to simulate fabric drape, manipulation and contact. In [12], they also developed software based on nonlinear shell theory to simulate 3D motions related to real fabric-manufacturing processes. Numerical simulations of fabric draping and folding were presented. Cugini et al [13, 14] developed a software environment to model and simulate non-rigid materials behavior during handling operations. The system's three main components include one which geometrically models both the workspace and the handling machinery, a physically based particle model that simulates the behavior of flat elastic and visco-plastic bulk types of non-rigid materials materials, and a robot offline programming system.

1.2.3 Optimal Locations

In mechanics literature, there exists some work on determination of optimal support locations for beams, columns and plates. Mroz et al [15] derived conditions on support location for minimum compliance of elastic beams, maximum safety factor of plastic collapse for plastic beams, and optimal design with varying cross section and

support position. Prager et al [16] established criteria for optimal location of supports and steps in yield moment for plastic design of beams. Rozvany et al [17] derived conditions for the optimal location of segment boundaries and internal supports for column design. Olhoff et al [18] designed continuous columns for minimum total cost of material and interior supports. In all these works classical optimization tools for finding maxima and minima of functions and functionals were used.

More recently, Narita [19] used a gradient technique to find support locations that maximize the fundamental natural frequency of beam and plate structures. Wang et al [20] used genetic algorithms to find optimal rigid and elastic support locations for beams with different boundary conditions. Xiang et al [21] used the simplex method of Nelder and Mead to solve for optimal locations of point supports that maximize the fundamental frequency of vibrating plates of different shapes. Wang et al [20] used the same optimization method to find optimal support points that maximize fundamental frequency of laminated rectangular plates. In all the above vibration problems, the objective function was computed using the Rayleigh-Ritz method. Pitarresi et al [22] presented a simple technique that uses a two-dimensional nonlinear least-squares fit of natural frequency versus support location data for rapid estimation of optimal support locations for vibrating plates. Roschke [23] used an iterative method based on Powell's conjugate directions to find optimal pick-up locations that minimize the absolute value of principle stresses in beams and plates. The objective functions were computed from a finite element solution.

1.3 Summary of Work Done

In this work, we determine pick-up locations for arbitrarily shaped flat limp parts that minimize a measure of their deformation. In particular, we develop procedures to solve the associated bound constrained optimization problem to minimize the strain energy (a measure of deformation) of the limp part as a function of pick-up locations. The limp parts are modeled as shells undergoing large deformations and rotations by a geometrically exact nonlinear shell finite element formulation. The strain energy is computed from the finite element solution for the deformation and requires the discretization of the limp part by a mesh. The pick-up locations correspond to enforcing boundary conditions in the mesh, and in the finite element method, boundary conditions can be enforced only at nodes in the mesh. Optimization procedures require

the evaluation of the objective function for different pick-up locations. Hence, an ability to automatically generate a mesh such that nodes are located at desired coordinates in the domain is required to solve the problem as a continuous optimization problem. When this is difficult, we solve a discrete optimization problem on a fixed mesh in which the pick-up locations can only be at nodes in the mesh. Optimization algorithms that use the Broyden, Fletcher, Goldfarb and Shanno (BFGS) update of the Hessian have been used to solve the continuous optimization problem. For the discrete optimization problem, a new gradient based algorithm to solve for optimal locations on a fixed mesh has been developed.

A strip of limp material can be considered one dimensional and modeled as a flexible beam. The pick-up locations correspond to the support locations of a continuous beam with uniformly distributed load. We solve the problem of optimal support locations for continuous beams that minimize the strain energy using an automatic meshing procedure and the BFGS methods for unconstrained and bound constrained minimization. The nonlinear shell finite element formulation is used to model beams by setting the Poisson's ratio to zero. Optimal locations upto 4 supports have been obtained for beams with different flexibility which is characterized by a non-dimensional number. Pick-up locations for different fabric strips have also been obtained.

Automatic meshing procedures are easy to implement for square/rectangular two dimensional domains, whereas, it is a difficult task for arbitrarily shaped two dimensional domains. We solve the problem of 4 pick-up locations for square/rectangular domains as a continuous optimization problem with automatic meshing using the BFGS method for bound constrained optimization. Results have been obtained in terms of nondimensional numbers involving the bending rigidity, length and thickness of the domain. For all other two dimensional domains, the optimal locations are solved on a fixed mesh using a gradient based algorithm and iterative mesh refinement. Optimal pick-up locations have been obtained for regular shapes such as rectangular and circular as well as irregular shapes obtained from apparel and datawear products.

1.4 Organization of Thesis

In Chapter 2, we give an overview of the nonlinear shell finite element formulation. The adaptive arc-length control algorithm that is used to solve the finite element

equations is derived and a suitable modification to stop the solution at a desired load level is incorporated. A method to compute the strain energy from the finite element solution is developed and validated. In Chapter 3, the minimization problem to be solved is defined and problems arising from meshing are discussed. Automatic mesh generation procedures for one dimensional domains and square/rectangular two dimensional domains are outlined. Optimization algorithms used to solve for optimal locations are described in detail. In Chapter 4, the method used to measure material properties of fabrics is described. Material properties of five different fabrics are listed. In Chapter 5 optimal locations for one dimensional domains are presented. In Chapter 6 optimal locations for two dimensional domains are presented. In Chapter 7, we draw conclusions from this research and suggest possible avenues for future work.

2 pick-up locations for a strip

4 pick-up locations for a square part

Figure 1.1: One dimensional and two dimensional limp parts picked up at different locations

Figure 1.2: Pick-up locations for limp parts

Chapter 2

Modeling of Limp Materials

2.1 Introduction

Various methods have been used to model limp materials such as fabrics in the literature. Particle based methods have been popular amongst the computer graphics community where the objective is visual realism and animation. These methods enforce equilibrium of external and internal forces at grid points on a particle mesh. Finite element methods have been used for modeling limp materials involved in engineering applications where physical accuracy is important. The approach follows from the well established discipline of continuum mechanics. As opposed to particle methods, there are no ad-hoc assumptions regarding calculation of stress and strain quantities. Equilibrium of the externally applied forces and internal elastic forces are enforced globally, not locally as in the particle method. A comparison of the two methods for the modeling of cloth was done by Eischen and Bigliani [30]. They found that the two methods compare relatively well with the finite element method exhibiting a slight edge in benchmarks against theoretical solutions. They recommended the particle method where computational speed and qualitative correct results are required and the finite element method where high accuracy is desired.

Since the present work is directed towards engineering applications, a finite element method based on nonlinear continuum shell theory is used to model limp materials. The formulation is based on the geometrically exact shell theory conceived by Simo et al [24, 25, 26]. Limp parts are modeled as flexible, doubly curved shells that can accommodate stretching, bending, and transverse shear deformations. Shell theory based on large deformations and rotations is used to formulate a finite element solution strategy. The resulting coupled system of nonlinear equations are solved by an adaptive arc-length control algorithm to take care of limit points on the load

In this chapter, an overview of the nonlinear shell formulation is given. Complete details can be found in Deng[32]. This is followed by the derivation and implementation details of the adaptive arc-length control algorithm. Finally, a method to compute the strain energy from the finite element solution is developed and validated.

The formulation is based on classical nonlinear shell theory in which the shell is regarded as an inextensible *one director Cosserat surface*. Key features of the formulation are :

- Use of the exponential mapping theory to derive an orthogonal transformation matrix which plays an important role in the rotational kinematics of the shell.
- Reduction to resultant form is carried out analytically
- Assumed strain and mixed interpolation methods are used to ameliorate shear and membrane locking respectively

The resulting geometrically exact stress resultant approach leads to an efficient numerical implementation free from mathematical complexities and suitable for large scale computations.

Shell Kinematics

The kinematic description of the shell starts by parameterizing the position of points within the shell, both on and off the mid-surface. Referring to Figure 2.1, points off the mid-surface are located by a position vector $\underline{\Phi}$,

$$\underline{\Phi}(\xi^1, \xi^2, \xi) = \underline{\phi}(\xi^1, \xi^2) + \xi \underline{t}(\xi^1, \xi^2) \quad (2.1)$$

where $\underline{\phi}$ is a position vector locating points on the mid-surface (reference surface) of the shell. The vector $\underline{t}$ is called the director and is a unit vector directed along fibers in the shell that are initially perpendicular to the reference surface. The coordinate ξ measures distance between the mid-surface and points off the mid-surface along $\underline{t}$. ξ^1 and ξ^2 are dimensionless (non-arc-length) curvilinear coordinates. In the resulting finite element implementation these coordinates are selected to be the “parent element” coordinates for the standard isoparametric quadrilateral element. Note that the unit vector $\underline{t}$ is not necessarily normal to the mid-surface of the deformed shell, thus admitting the possibility of transverse shear strain. The thru-thickness coordinate ξ is in the range $-h/2 \leq \xi \leq h/2$, where h is the shell thickness. The undeformed configuration of the shell is given by

$$\underline{\Phi}_0(\xi^1, \xi^2, \xi) = \underline{\phi}_0(\xi^1, \xi^2) + \xi \underline{t}_0(\xi^1, \xi^2) \quad (2.2)$$

where $\underline{\phi}_0$ is a vector locating points on the undeformed mid-surface and $\underline{t}_0$ is the director in the initial configuration, assumed normal to the mid-surface. The essential

problem is to determine the evolution of $\underline{\phi}$ and $\underline{t}$ as the shell deforms under its own weight or is manipulated in some way.

The evolution of the unit director vectors $\underline{t}(\xi^1, \xi^2)$ during a motion of the shell depends on an orthogonal transformation matrix $\underline{\Lambda}$. Let $\underline{t} = \underline{\Lambda}\underline{E}$, where $\underline{\Lambda}$ is an orthogonal matrix such that $\underline{\Lambda}\underline{\Lambda}^T = \underline{I}$ and $\underline{E}$ is an inertially fixed unit vector. Thus, to determine $\underline{t}$ during the shell motion, it is sufficient (or equivalent) to determine the matrix $\underline{\Lambda}$. At any point on the shell, this matrix is related to the unit director vector according to

$$\underline{\Lambda} = (\underline{E} \cdot \underline{t})\underline{I} + \widehat{\underline{E} \times \underline{t}} + \frac{1}{1 + \underline{E} \cdot \underline{t}}(\underline{E} \times \underline{t}) \otimes (\underline{E} \times \underline{t}) \quad (2.3)$$

where the $\widehat{}$ indicates the skew symmetric matrix associated with the indicated vector, $\otimes$ represents a tensor outer product operator.

Strain Measures

For the shell model there are three types of deformation (strain) considered. First membrane strains measure the extension and/or compression of fibers in the plane of mid-surface as well as shear distortion of fibers in the plane of the midsurface that are initially perpendicular. Transverse shear strain is a measure of the change in the angle between fibers tangent to the mid-surface and fibers initially perpendicular to the mid-surface. Bending strains are a measure of the curvature and twist of the mid-surface. The strain measures are defined as follows and depend on the derivatives of position vector $\underline{\phi}$, together with the normal $\underline{t}$ and its derivative,

- Membrane strain

$$\varepsilon_{\alpha\beta} \equiv \frac{1}{2}(\underline{\phi}_{,\alpha} \cdot \underline{\phi}_{,\beta} - \underline{\phi}_{0,\alpha} \cdot \underline{\phi}_{0,\beta}) \quad (2.4)$$

- Transverse shear strain

$$\gamma_\alpha \equiv \underline{\phi}_{,\alpha} \cdot \underline{t} - \underline{\phi}_{0,\alpha} \cdot \underline{t}_0 \quad (2.5)$$

- Bending Strain

$$\kappa_{\alpha\beta} \equiv \underline{\phi}_{,\alpha} \cdot \underline{t}_{,\beta} - \underline{\phi}_{0,\alpha} \cdot \underline{t}_{0,\beta} \quad (2.6)$$

where,

$$\underline{\phi}_{,\alpha} = \frac{\partial \underline{\phi}(\xi^1, \xi^2)}{\partial \xi^\alpha} \text{ and } \underline{t}_{,\alpha} = \frac{\partial \underline{t}(\xi^1, \xi^2)}{\partial \xi^\alpha} \text{ for } \alpha = 1, 2 \quad (2.7)$$

are the covariant tangent base vectors and director derivatives on the mid-surface. These definitions are consistent with classical shell theory. Linear elastic isotropic constitutive law is used for the relationship between the effective stress resultants and the strains.

Constitutive Equations

The relationship between the forces and moments acting on the shell and the strains is governed by constitutive equations. For shells, the internal forces and moments are referred to as stress and moment “resultants”. Figure 2.2 shows the directions and notations of these quantities. The constitutive equations for linear elastic isotropic materials in a non-orthogonal, non arc-length curvilinear coordinate system are given by

Membrane :

$$\tilde{n}^{\alpha\beta} = hC^{\alpha\beta\gamma\delta}\varepsilon_{\gamma\delta} \quad (2.8)$$

Bending :

$$\tilde{m}^{\alpha\beta} = \frac{h^3}{12}C^{\alpha\beta\gamma\delta}\kappa_{\gamma\delta} \quad (2.9)$$

Transverse Shear :

$$\tilde{q}^\alpha = \kappa h C^{\alpha 3 \gamma 3} \gamma_\gamma \quad (2.10)$$

where $\alpha, \beta, \gamma, \delta = 1, 2$. The elastic constants $C^{\alpha\beta\gamma\delta}$ are

$$C^{\alpha\beta\gamma\delta} = \frac{E}{1-\nu^2} \left[\nu a^{0\alpha\beta} a^{0\gamma\delta} + \frac{1-\nu}{2} (a^{0\alpha\gamma} a^{0\beta\delta} + a^{0\alpha\delta} a^{0\gamma\beta}) \right] \quad (2.11)$$

$$C^{\alpha 3 \gamma 3} = \kappa G_s \begin{bmatrix} a^{011} & a^{012} \\ a^{012} & a^{022} \end{bmatrix} \quad (2.12)$$

where

$$a^{0\alpha\beta} = \underline{a}^{0\alpha} \cdot \underline{a}^{0\beta} = \frac{\partial \xi^\alpha}{\partial x^a} \frac{\partial \xi^\beta}{\partial x^a} \quad (2.13)$$

$$G_s = \frac{E}{2(1+\nu)} \quad (2.14)$$

E is the modulus of elasticity, ν is Poisson’s ratio, and κ is the shear correction factor.

Figure 2.2: Effective Stress Resultants Acting on the Shell Mid-Surface

Finite Element Formulation

The principles of balance of linear and angular momentum are used to derive partial differential equations of motion of the shell. The static weak form is obtained using standard variational techniques. This results in a highly nonlinear system of integral equations which are linearized using the concept of directional derivatives. The linearized weak form is then discretized by interpolating the position vector $\underline{\phi}$ and unit director $\underline{t}$ between nodal values by standard bilinear isoparametric shape functions. This yields a system of linear algebraic equations of the form

$$\underline{K}(\underline{\phi}, \underline{t}) \left\{ \begin{array}{c} \Delta \underline{\phi} \\ \Delta \underline{t} \end{array} \right\} = \underline{F}_{ext} - \underline{P}(\underline{\phi}, \underline{t}) \quad (2.15)$$

where $\underline{K}(\underline{\phi}, \underline{t})$ is the tangent stiffness matrix, $\underline{P}(\underline{\phi}, \underline{t})$ is the internal force vector, and $\underline{F}_{ext}$ is the external force vector. Iterative solution of this matrix equation by an arc-length control algorithm generates an incremental displacement vector $\Delta \underline{\phi}$ that is used to update the position of the shell mid-surface, together with an incremental director vector $\Delta \underline{t}$ that is used to update the directors as the shell deforms.

2.3 Adaptive Arc-length Control Algorithm

The finite element equations (Equation 2.15) must be solved using a nonlinear iterative algorithm, such as Newton-Raphson. The basic Newton-Raphson algorithm fails when limit points or bifurcation points are encountered on the load deflection curve. In the literature, various methods have been proposed to trace the nonlinear response from the pre-limit into the post-limit range [33]. Of these, the arc-length methods (Riks[34], Crisfield[35, 36], Schweizerhof et al.[27], etc) are the most versatile as they can trace the nonlinear response over the entire range. In these methods, a constraint equation is added to the equilibrium equations and satisfied at every iteration. The most common implementation keeps a specific arc-length, defined as a weighted euclidean norm of the displacement and loads constant. We use Schweizerhof and Wriggers [27] method in which the arc-length constraint is consistently linearized along with the equilibrium equations. In the following, we first derive the iterative updates for the configuration and load factor and a new unique modification required to stop at a desired load level. This is followed by discussions on arc-length adaptation and implementation aspects.

2.3.1 Derivation

The finite element equations for an equilibrium configuration at load step n are

$$\underline{P}(\underline{\Phi}_n) - \underline{F}_{ext}(\lambda_n) = \underline{0} \quad (2.16)$$

where $\underline{P}$ and $\underline{F}_{ext}$ are the internal and external forces respectively. Henceforth, $\underline{F}_{ext}$ will be referred to as $\underline{F}$. The external force may be viewed as a function of a load factor λ_n , i.e.

$$\underline{F}(\lambda_n) = \lambda_n \underline{F}_{ref} \quad (2.17)$$

where $\underline{F}_{ref}$ is the reference external force which in our case is the weight/area of the part being modeled. It is assumed that the current value of the displacements (and rotations) stored in $\underline{\Phi}_n$ and the load factor λ_n satisfy Equation 2.16. The computational task is to advance the solution to a new equilibrium configuration $\underline{\Phi}_{n+1}, \lambda_{n+1}$. The arc-length method limits the excursion along the load-deflection curve by a distance Δs . The iteration occurs on spherical hyperplanes and the constraint equation is only iteratively satisfied. This is shown in Figure 2.3 in which the vector $\underline{p}$ represents the excursion.

The equilibrium equations that must be satisfied at load step $n + 1$ are,

$$\underline{P}(\underline{\Phi}_{n+1}) - \underline{F}(\lambda_{n+1}) = \underline{0} \quad (2.18)$$

The following constraint equation is introduced

$$f = [(\underline{\Phi}_{n+1}^{(i)} - \underline{\Phi}_n) \cdot (\underline{\Phi}_{n+1}^{(i)} - \underline{\Phi}_n) + \beta^2(\lambda_{n+1}^{(i)} - \lambda_n)^2]^{1/2} - \Delta s = 0 \quad (2.19)$$

where Δs and β are the arc-length control constant and scaling factor, respectively. In the following, the algorithm iteration counter is indicated by index i .

Upon linearizing both the equilibrium and the constraint equations, the following equation is obtained,

$$\underline{P}(\underline{\Phi}_{n+1}^{(i)}) + \left. \frac{\partial \underline{P}}{\partial \underline{\Phi}_{n+1}} \right|^{(i)} \Delta \underline{\Phi}_{n+1}^{(i)} - \underline{F}(\lambda_{n+1}^{(i)}) - \left. \frac{\partial \underline{F}}{\partial \lambda_{n+1}} \right|^{(i)} \Delta \lambda_{n+1}^{(i)} \approx \underline{0} \quad (2.20)$$

$$f(\underline{\Phi}_{n+1}^{(i)}, \lambda_{n+1}^{(i)}) + \left(\frac{\partial f}{\partial \underline{\Phi}_{n+1}} \right)^T \Big|^{(i)} \Delta \underline{\Phi}_{n+1}^{(i)} + \left. \frac{\partial f}{\partial \lambda_{n+1}} \right|^{(i)} \Delta \lambda_{n+1}^{(i)} \approx 0 \quad (2.21)$$

Let

$$\begin{aligned} \underline{P}^{(i)} &\equiv \underline{P}(\underline{\Phi}_{n+1}^{(i)}) \\ f^{(i)} &\equiv f(\underline{\Phi}_{n+1}^{(i)}, \lambda_{n+1}^{(i)}) \\ \underline{K}^{(i)} &\equiv \left. \frac{\partial \underline{P}}{\partial \underline{\Phi}_{n+1}} \right|^{(i)} \\ \underline{v}^{(i)T} &\equiv \left(\left. \frac{\partial f}{\partial \underline{\Phi}_{n+1}} \right|^T \right)^{(i)} \\ \underline{F}^{(i)} &\equiv \underline{F}(\lambda_{n+1}^{(i)}) \\ \alpha^{(i)} &\equiv \left. \frac{\partial f}{\partial \lambda_{n+1}} \right|^{(i)} \\ \underline{F}_0 &\equiv \left. \frac{\partial \underline{F}}{\partial \lambda_{n+1}} \right|^{(i)} \end{aligned}$$

The linearized Equations 2.20 - 2.21 can be put in matrix form,

$$\begin{bmatrix} \underline{K}^{(i)} & -\underline{F}_0 \\ \underline{v}^{(i)T} & \alpha^{(i)} \end{bmatrix} \begin{Bmatrix} \Delta \underline{\Phi}_{n+1}^{(i)} \\ \Delta \lambda_{n+1}^{(i)} \end{Bmatrix} = \begin{Bmatrix} \underline{F}^{(i)} - \underline{P}^{(i)} \\ -f^{(i)} \end{Bmatrix} \quad (2.22)$$

Solving for $\Delta\Phi_{n+1}^{(i)}$, yields

$$\Delta\Phi_{n+1}^{(i)} = \left(\underline{K}^{(i)} + \frac{1}{\alpha^{(i)}} \underline{F}_0 \right)^{-1} \left(\underline{F}^{(i)} - \underline{P}^{(i)} - \frac{f^{(i)}}{\alpha^{(i)} \underline{F}_0} \right) \quad (2.23)$$

The Sherman-Morrison [37] formula can be employed to express the required matrix inverse in terms of $\underline{K}^{(i)-1}$ explicitly.

$$\left(\underline{K}^{(i)} + \frac{1}{\alpha^{(i)}} \underline{F}_0 \underline{v}^{(i)T} \right)^{-1} = \underline{K}^{(i)-1} - \frac{1}{\alpha^{(i)} + \underline{F}_0^T \underline{K}^{(i)-1} \underline{v}^{(i)}} \left(\underline{K}^{(i)-1} \underline{F}_0 \underline{v}^{(i)T} \underline{K}^{(i)-1} \right) \quad (2.24)$$

This is significant, because it is this matrix, $\underline{K}^{(i)-1}$, that is directly available in the finite element code. Define

$$\Delta\mathbf{u}^I \equiv \underline{K}^{(i)-1} \underline{F}_0 \quad (2.25)$$

$$\Delta\mathbf{u}^{II} \equiv \underline{K}^{(i)-1} (\underline{F}^{(i)} - \underline{P}^{(i)}) \quad (2.26)$$

Substitute Equations 2.24 - 2.26 back into Equation 2.23, then

$$\Delta\Phi_{n+1}^{(i)} = - \left[\frac{f^{(i)} + \underline{v}^{(i)T} \Delta\mathbf{u}^{II}}{\alpha^{(i)} + \underline{v}^{(i)T} \Delta\mathbf{u}^I} \right] \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} \quad (2.27)$$

Substituting Equation 2.27 back into Equation 2.22 gives the expression for $\Delta\lambda_{n+1}^i$ as

$$\Delta\lambda_{n+1}^{(i)} = - \frac{f^{(i)} + \underline{v}^{(i)T} \Delta\mathbf{u}^{II}}{\alpha^{(i)} + \underline{v}^{(i)T} \Delta\mathbf{u}^I} \quad (2.28)$$

From the definitions of $\underline{v}^{(i)T}$ and $\alpha^{(i)}$, the explicit expressions for $\underline{v}^{(i)T}$ and $\alpha^{(i)}$ are given by

$$\underline{v}^{(i)T} = (f^{(i)} + \Delta s)^{-1} (\Phi_{n+1}^{(i)} - \Phi_n)^T \quad (2.29)$$

$$\alpha^{(i)} = (f^{(i)} + \Delta s)^{-1} \beta^2 (\lambda_{n+1}^{(i)} - \lambda_n) \quad (2.30)$$

Therefore, the explicit expressions for $\Delta\lambda_{n+1}^i$ and $\Delta\Phi_{n+1}^{(i)}$ are given by

$$\Delta\lambda_{n+1}^{(i)} = - \frac{f^{(i)}(f^{(i)} + \Delta s) + (\Phi_{n+1}^{(i)} - \Phi_n)^T \Delta\mathbf{u}^{II}}{\beta^2(\lambda_{n+1}^{(i)} - \lambda_n) + (\Phi_{n+1}^{(i)} - \Phi_n)^T \Delta\mathbf{u}^I} \quad (2.31)$$

$$\Delta\Phi_{n+1}^{(i)} = \Delta\lambda_{n+1}^{(i)} \Delta\mathbf{u}^I + \Delta\mathbf{u}^{II} \quad (2.32)$$

Figure 2.3: Solution by Arc-length Control

2.3.2 Modification for fixed loading

The arc-length procedure has no provision to stop at a desired load level as the increments in the load factor ($\Delta\lambda_{n+1}^{(i)}$) are solved for in each iteration i . In this work, we require the solution at the load level corresponding precisely to the self weight of the limp part (i.e. $\lambda = 1.0$). This was achieved by solving for the configuration corresponding to the load factor of $\lambda = 1.0$ using the ordinary Newton-Raphson method (Cook et al. [38]) once the load factor exceeded $\lambda = 1.0$. During this process, the arc-length constraint is not active. The iterative procedure can be derived as follows :

Let $\underline{\Phi}_1$ be the configuration corresponding to a load factor $\lambda_1 > 1.0$, $\underline{F}_1 = \underline{F}(\lambda_1)$ and $\underline{F}_t = \underline{F}(\lambda = 1.0)$. Then,

$$\underline{P}(\underline{\Phi}_1) = \underline{F}_1 \quad (2.33)$$

Upon linearizing we get

$$\begin{aligned} \underline{P}(\underline{\Phi}_1^{(i)} + \Delta\underline{\Phi}_1^{(i)}) &\approx \underline{P}(\underline{\Phi}_1^{(i)}) + \left. \frac{\partial \underline{P}}{\partial \underline{\Phi}_1} \right|^{(i)} \Delta\underline{\Phi}_1^{(i)} \\ &\approx \underline{F}_1^{(i)} + \underline{K}^{(i)} \Delta\underline{\Phi}_1^{(i)} \end{aligned} \quad (2.34)$$

We seek $\Delta\underline{\Phi}_1^{(i)}$ such that

$$\underline{P}(\underline{\Phi}_1^{(i)} + \Delta\underline{\Phi}_1^{(i)}) = \underline{F}_t$$

Therefore, we solve for

$$\Delta\underline{\Phi}_1^{(i)} = \underline{K}^{-1}(\underline{F}_t - \underline{F}_1^{(i)}) \quad (2.35)$$

and update the configuration by

$$\underline{\Phi}_1^{(i+1)} = \underline{\Phi}_1^{(i)} + \Delta\underline{\Phi}_1^{(i)} \quad (2.36)$$

This process (Equations 2.35 - 2.36) is continued until the load imbalance $\|\underline{F}_t - \underline{F}_1^{(i)}\|$ is sufficiently small. The following relative convergence test is used to stop the iteration,

$$\frac{\|\underline{F}_t - \underline{F}_1^{(i)}\|}{\|\underline{F}_t - \underline{F}_1^{(0)}\|} \leq tol \quad (2.37)$$

where, a stopping tolerance $tol = 10^{-9}$ was typically used.

2.3.3 Adapting the Arc-length

The arc-length constraint restricts the solution to lie on a spherical hyperplane as shown in Figure (2.3). In the original method of Wriggers et. al. [27], the arc-length factor (Δs) is fixed in the first load step based on an initial load factor λ_0 and remains constant for all subsequent load steps. As is typical with nonlinear problems, a suitable λ_0 that does not result in divergence is problem dependent. Different λ_0 's may be required for different boundary conditions or shapes of the domain. These are typically found by numerical experiments for each case.

At this point it is important to look at the problem to be solved. In this work, optimization is to be done with the objective function computed from a nonlinear finite element solution. In implementation, the optimization algorithm would require the subroutine returning the objective function to function with no intervention and the objective function will be evaluated at different boundary conditions. It is easy to anticipate that divergence is bound to occur if a constant initial load factor λ_0 is used for different boundary conditions. A solution to this problem is to reduce the arc-length when divergence occurs and restart the load step. This has been implemented and our numerical experience shows that such a procedure solves the above mentioned problem to a large extent.

The arc-length procedure cannot detect bifurcation points. Hence, divergence can occur when a bifurcation point is reached. Methods are available in literature to detect the point of branching of solution and follow a particular branch (Wagner et al [39], Wriggers et al [40, 41]). These typically require solution of the associated eigenvalue problem to determine the number of existing branches and perturbation of the solution using the associated eigenvector. This would require a priori knowledge of where the bifurcation point occurs as an extended system of equations needs to be solved. Again, with regard to the optimization problem to be solved, it is impractical to implement detection of bifurcation points and tracing of post-critical branches with no user intervention.

Our numerical experience suggests that thin materials (low bending rigidity) are prone to bifurcation with the geometry of the limp part and the location of boundary conditions also playing a major role. In the solution procedure, convergence failure is seen in the form of perpetual oscillation between two load factors. The solution seems to flip from one branch to another and hence the desired final load factor of $\lambda = 1.0$ is not attained. It was discovered that increasing the arc-length at this point, in some

Figure 2.4: Area under a nodal load deflection curve

cases, helped the solution to escape this oscillatory behavior and reach the final load factor of $\lambda = 1.0$ via one of the post-critical branches. A procedure to detect the oscillatory behavior and increase the arc-length has been incorporated.

2.3.4 Implementation

The implementation of the arc-length procedure follows that of Deng[32] except that no acceleration factor is used and no contact problem is solved. The definiteness of the stiffness matrix K is observed at the beginning of each step and unloading is initiated whenever it is negative definite. This is done by multiplying the increment in the load factor by the sign of the determinant of K . Arc-length reduction on divergence and expansion on detection of oscillatory behavior have been added. Our numerical experience indicates that the added features improve the convergence properties.

2.4 Strain Energy Computation

The computation of strain energy of the deformed limp part is required. This is computed from the finite element solution by summing the areas under load deflection curves of all the nodes in the mesh. This is conveniently done by approximating the area due to each load step by that of a trapezoid as shown in Figure 2.4. The following formula for strain energy (SE) was used.

$$SE = \sum_{nodes} \sum_{i=1}^{nsteps} \frac{1}{2} (f_i + f_{i-1}) (d_i - d_{i-1}) \quad (2.38)$$

where, $nsteps$ is the total number of load steps, f_i and d_i are the nodal force and the nodal displacement, respectively, at the end of load step i . It is to be noted that the only nodal forces are those due to the self weight of the fabric. Nodal forces developed at hard reaction points do not contribute to strain energy.

2.4.1 Validation of Method

The method was validated by comparing the computed values of strain energies with those obtained from analytical expressions for plates undergoing small deformations. The book by Timoshenko et al [42] provides expressions for deflections of square and circular plates with various boundary conditions. These were used to derive analytical expressions for strain energies.

2.4.1.1 Analytical Expressions

The calculation of strain energy in rectangular and polar coordinates of an isotropic plate is as follows :

Rectangular coordinates :

$$V = \frac{1}{2} D \int \int \left\{ \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)^2 - 2(1 - \nu) \left[\frac{\partial^2 w}{\partial x^2} \frac{\partial^2 w}{\partial y^2} - \left(\frac{\partial^2 w}{\partial x \partial y} \right)^2 \right] \right\} dx dy \quad (2.39)$$

Polar coordinates :

$$V = \frac{1}{2} D \int \int \left\{ \left(\frac{\partial^2 w}{\partial r^2} + \frac{1}{r} \frac{\partial w}{\partial r} \right)^2 - \frac{2(1 - \nu)}{r} \frac{\partial w}{\partial r} \frac{\partial^2 w}{\partial r^2} \right\} r dr d\theta \quad (2.40)$$

where,

$$D = \frac{Eh^3}{12(1-\nu^2)} \text{ is the bending rigidity,}$$

E is Young's modulus, ν is Poisson's ratio, h is thickness of plate and w is the transverse plate deflection.

Simply Supported Rectangular Plate with Uniformly Distributed Load

Navier's solution for a rectangular plate of dimensions $a \times b$ subjected to a uniformly distributed load (q) is

$$w_{rect} = \frac{16q}{\pi^6 D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\sin(\frac{m\pi x}{a}) \sin(\frac{n\pi y}{b})}{mn(\frac{m^2}{a^2} + \frac{n^2}{b^2})^2} \quad (2.41)$$

(m, n are odd)

Substituting Equation 2.41 into Equation 2.39 and simplifying results in the following expression for strain energy

$$V_{rect} = \frac{32abq^2}{\pi^8 D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{m^2 n^2 (\frac{m^2}{a^2} + \frac{n^2}{b^2})^2} \quad (2.42)$$

Simply Supported Circular Plate with Uniformly Distributed Load

The deflection (w_{ss}) of a circular plate of radius a is given by

$$w_{ss} = \frac{q(a^2 - r^2)}{64D} (Ca^2 - r^2), \text{ where } C = \left(\frac{5+\nu}{1+\nu}\right) \quad (2.43)$$

Substitution of Equation 2.43 into Equation 2.42 and simplifying results in the following expression for strain energy :

$$V_{ss} = \frac{\pi q^2 a^5}{3072D} \{11 + 3\nu + 3(1+\nu)(C^2 - 2C)\} \quad (2.44)$$

Clamped Circular Plate with Uniformly Distributed Load

The deflection (w_{clamp}) of a circular plate of radius a is given by

$$w_{clamp} = \frac{q}{64D} (a^2 - r^2) \quad (2.45)$$

Substitution of Equation 2.44 into Equation 2.40 and simplifying results in the following expressions for strain energy :

$$V_{clamp} = \frac{\pi q^2 a^6}{384D} \quad (2.46)$$

2.4.1.2 Comparison of Numerically Computed Values

The following data was used to compute strain energies using the FE program for comparison.

Simply Supported Square Plate

$$\begin{aligned} a = b &= 2.0 \text{ m} \\ h &= 0.1 \text{ m} \\ q &= 1.0 \text{ N/m}^2 \\ E &= 726400 \text{ N/m}^2 \\ \nu &= 0.3 \end{aligned}$$

Simply Supported Circular Plate

$$\begin{aligned} a &= 0.5 \text{ m} \\ h &= 0.01 \text{ m} \\ q &= 1.0 \text{ N/m}^2 \\ E &= 43476562.5 \text{ N/m}^2 \\ \nu &= 0.3 \end{aligned}$$

Clamped Circular Plate

$$\begin{aligned} a &= 0.5 \text{ m} \\ h &= 0.01 \text{ m} \\ q &= 1.0 \text{ N/m}^2 \\ E &= 10664062.5 \text{ N/m}^2 \\ \nu &= 0.3 \end{aligned}$$

Symmetry was used and only a quarter of the domain is modeled (See Figure 2.5 for the meshes). Table 2.1 compares the numerical and analytical values. The numerically computed values are within a 2% error. It is to be noted that the solution is obtained in one step as a linear problem is solved. When solving a nonlinear problem, the accuracy of the strain energy computed will depend on how well the load deflection curve is traced. This can be ensured by choosing suitable initial arc-length λ_0 and scaling factor β .

Table 2.1: Numerically and Analytically computed Plate Strain Energies

Plate Type	Strain Energy		% error
	Analytical	Numerical	
Square Simply supported	8.1896e-4	8.1507e-4	0.47
Circular Simply supported	1.8034e-4	1.7732e-4	1.66
Circular Clamped	1.309e-4	1.2883e-4	1.58

2.5 Summary

In this chapter, we presented an overview of the geometrically exact nonlinear shell finite element formulation used to model limp materials. Modifications to the arc-length control procedure and a method to compute the strain energy have also been presented. In the next chapter, we describe the various optimization algorithms that have been used to solve for optimal locations.

Figure 2.5: Plate Meshes (Symmetry employed)

Chapter 3

Algorithms

3.1 Introduction

The objective of this research is to determine optimal pick-up locations for limp parts of different shapes such that they undergo minimum deformation when picked up. The limp parts are modeled using the geometrically exact nonlinear shell finite element formulation described in Chapter 2. The optimal locations are determined by minimizing a measure of the deformation of the limp part as a function of pick-up locations. We choose to minimize the strain energy as it is a measure of average curvature of the deformed part and in some sense the average deflection. The strain energy is computed from a finite element solution for the deformation of the limp part by the method described in Chapter 2.

The problem to be solved is a *bound constrained optimization* problem as the pick-up locations are restricted to lie within the domain bounded by the boundary of the limp part. Different methods are available in literature to solve such optimization problems. In this work, we have been successful in solving the problem using *gradient based methods*. These methods have a strong theoretical basis and require the evaluation of both function and gradient. The use of a finite element mesh for computing the objective function presents a unique problem while implementing these methods. As a result, the problem can be solved as a continuous optimization problem only for a one dimensional domain corresponding to a strip of limp material and square/rectangular two dimensional domains. Optimal locations for arbitrary shaped two dimensional domains have to be solved as discrete optimization problems.

In this chapter, we first define the minimization problem. Next, problems arising from meshing are discussed followed by the automatic meshing procedures developed. The method of computing the gradient is described next. In the remainder of the chapter, optimization algorithms are outlined.

3.2 Minimization Problem

The optimal locations are found by solving the following minimization problem :

$$\begin{aligned} \min \quad & f(\underline{x}) : \mathbb{R}^m \rightarrow \mathbb{R} \\ & \underline{x} \in D \end{aligned} \quad (3.1)$$

where, $f(\underline{x})$ is the strain energy and objective function, D is the domain bounded by the boundary of the limp material part and $\underline{x}$ is a $(m \times 1)$ vector of coordinates of pick-up locations. For one dimensional domains $m = n$, whereas for two dimensional domains $m = 2n$, where n is the number of pick-up locations.

3.3 Meshing Issues

Numerical solution of the minimization problem in Equation 3.1 requires computation of the strain energy for different pick-up locations. The pick-up locations correspond to boundary conditions in the mesh. In the finite element method, boundary conditions can be applied only at nodes in the mesh. Hence, for any $\underline{x}$, computation of $f(\underline{x})$ requires that the domain D be discretized by a mesh of quadrilateral elements with nodes located at the coordinates of pick-up locations contained in the vector $\underline{x}$.

In the implementation of optimization algorithms, computation of objective function is to be done with no user intervention. Hence a mesh needs to be automatically generated whenever the objective function is computed. This can be easily done for a one dimensional domain. Also, for the specific problem of 4 pick-up locations on a square/rectangular domain, an automatic meshing procedure can be developed when symmetry is used and only one pick-up location is solved for on a quarter portion of the domain. However, automatic meshing of arbitrarily shaped two dimensional domains is a difficult task and has not been addressed in this work.

The ability to automatically generate a mesh enables us to solve the problem as a continuous optimization problem. In the case where automatic meshing is not possible, the optimization problem is solved as a discrete optimization problem on a fixed mesh. The pick-up locations are restricted to lie on the nodes of the fixed mesh. For a more accurate solution, the mesh is refined in selected regions and the problem is resolved. The automatic meshing procedure for one dimensional and square/rectangular domains are outlined in the next two sections.

3.3.1 Automatic Meshing for a One Dimensional Domain

The meshing problem is as follows :

Given length of the domain (l), number of locations (n), vector of coordinates of locations ($\underline{x}$) of dimension $n \times 1$, desired number of elements ($Numeld$), a mesh is to be generated with nodes located at the coordinates of the pick-up locations contained in $\underline{x}$.

The procedure is as follows :

1. The interval $[0, l]$ is divided into the following sub-intervals :

$$[0, x_1], [x_1, x_2], \dots, [x_i, x_{i+1}], \dots, [x_n, l]$$

If $x_1 = 0$ or $x_n = L$, the first and last intervals are absent. In the general case when all $x_i \in (0, l)$, there will be $(n+1)$ intervals.

2. The number of elements in each interval are obtained from the ratio of the length of the interval to l as follows :

$$\text{No. of elements in interval} = \text{INT} \left[\left(\frac{\text{length of interval}}{l} \right) Numeld \right]$$

where, INT refers to rounding off to an integer number. At least one element is assigned to an interval no matter how small it is. The actual number of elements which is the sum of the elements in all intervals may be different from $Numeld$.

3. The length of elements in each interval are determined by dividing the interval length with the number of elements in the interval.
4. Nodal coordinates are generated using the element lengths and node numbers are assigned successively starting from the left end. For a mesh of quadrilateral elements (to model a strip with plate elements) an additional set of nodes offset by the width of the element are generated. Element numbers and element connectivity are generated from node numbers.

The above procedure ensures that a node is located at every pick-up location x_i . The specific node numbers at which boundary conditions are to be applied can be determined from the node numbers and number of elements in each section. Some examples of the meshes generated along with input data are shown in Figure 3.1.

Figure 3.1: Meshes generated by automatic meshing procedure for a 1-D domain. Dots indicate the nodes where boundary conditions are to be applied.

Figure 3.2: 4 pick-up locations on a rectangular domain

3.3.2 Automatic Meshing for Square/Rectangular domains

The problem of 4 pick-up locations on a rectangular domain can be simplified to 1 pick-up location on a quarter domain by exploiting symmetry as shown in Figure 3.2. It is sufficient to solve for optimal location (x_1, y_1) on $OABC$ by enforcing symmetric boundary conditions along edges OA and OC . This greatly simplifies the automatic meshing task with the meshing problem reducing to the following :

Given dimensions a and b of the rectangle, desired level of discretization in terms of number of elements in the x -direction (Nx) or y -direction (Ny), coordinates of pick-up location (x_1, y_1) , a mesh of quadrilateral elements is to be generated with a node located at (x_1, y_1) . In addition, we require the node number corresponding to (x_1, y_1) and node numbers where symmetric boundary conditions are to be enforced.

Figure 3.3: Symmetry used to solve for 1 location on quarter rectangle

The algorithm is as follows :

1. The desired number of elements in the x and y directions are determined from the specified level of discretization and the aspect ratio of the rectangle. For example, if Nx has been specified, Ny is determined as follows :

$$Ny = INT\left(\frac{b}{a}Nx\right)$$

2. The *actual* number of elements in the x and y directions is determined. For this, $OABC$ is divided into 4 regions as shown in Figure 3.3.

- Regions 1 and 2 will have the same number of elements (Nx_1) in the x -direction. Similarly, regions 3 and 4 will have the same number of elements

(Nx_2) in the x -direction. Nx_1 and Nx_2 are determined as follows :

$$Nx_1 = INT\left(\frac{a - x_1}{a}\right)Nx$$

$$Nx_2 = INT\left(\frac{x_1}{a}\right)Nx$$

- Regions 1 and 4 will have the same number of elements (Ny_1) in the y -direction. Similarly, regions 3 and 2 will have the same number of elements (Ny_2) in the y -direction. Ny_1 and Ny_2 are determined as follows :

$$Ny_1 = INT\left(\frac{y_1}{b}\right)Ny$$

$$Ny_2 = INT\left(\frac{b - y_1}{b}\right)Ny$$

- The actual number of elements in the x and y direction are

$$Nx = Nx_1 + Nx_2$$

$$Ny = Ny_1 + Ny_2$$

3. Length of element edges in the x -direction (dx_i) and y -direction (dy_i) for each region are determined as follows:

$$\text{Region 1 : } dx_1 = \frac{a - x_1}{Nx_1}; \quad dy_1 = \frac{b - y_1}{Ny_1}$$

$$\text{Region 2 : } dx_1; \quad dy_2 = \frac{y_1}{Ny_2}$$

$$\text{Region 3 : } dx_2 = \frac{x_1}{Nx_2}; \quad dy_2$$

$$\text{Region 4 : } dx_2; \quad dy_2$$

Note : When $x_1 = a$ or $y_1 = b$, $OABC$ is divided into only two regions. $Nx_2 = Ny_2 = 0$ and dx_2 and dy_2 do not exist.

4. The coordinates of the nodes are generated using Nx_i , Ny_i , dx_i and dy_i . A columnwise numbering scheme starting from the node located at point A in figure 3.3 is used for numbering the nodes. This enables us to generate the element numbers and element connectivity from node numbers. In addition, the node numbers corresponding to (x_1, y_1) and the symmetric boundary conditions along lines OA and OC are determined.

Sample meshes generated by the above procedure are shown in Figure 3.4 along with the input parameters. It can be seen that the procedure generates good meshes except when (x_1, y_1) is located close to the boundaries as shown in Figure 3.4(d). This is prevented from happening by choosing initial guesses not close to the boundaries and by perturbing the coordinates away from the the boundaries during the optimization. More details can be found in Chapter 6.

Figure 3.4: Meshes generated for rectangular domains

3.4 Gradient Computation

The optimization methods used in this work require the computation of gradient of the objective function ($\nabla f(\underline{x})$). $\nabla f(\underline{x})$ is an $(m \times 1)$ vector and the following forward difference or central difference approximations are used to compute the gradient:

$$\text{Forward difference : } \nabla f(\underline{x})_i = \frac{f(\underline{x} + \epsilon \underline{e}_i) - f(\underline{x})}{\epsilon} \text{ for } i = 1, \dots, m \quad (3.2)$$

$$\text{Central difference : } \nabla f(\underline{x})_i = \frac{f(\underline{x} + \epsilon \underline{e}_i) - f(\underline{x} - \epsilon \underline{e}_i)}{2\epsilon} \text{ for } i = 1, \dots, m \quad (3.3)$$

where, ϵ is a perturbation and $\underline{e}_i$ is the i^{th} unit vector. $f(\underline{x} \pm \epsilon \underline{e}_i)$ is computed after perturbing the coordinate of the nodes where boundary conditions corresponding to pick-up locations are applied by ϵ .

3.5 Iterative Algorithms for Minimization

Although the minimization problem in Equation 3.1 is bound constrained, it was possible to solve for optimal locations on one dimensional domains using an unconstrained minimization algorithm. In the case of rectangular domains, upper and lower bounds of the coordinates are readily available. Hence, a bound constrained minimization algorithm for simple bounds was used to solve for optimal locations. Both these cases were solved as continuous optimization problems with automatic meshing and using existing optimization codes. For arbitrarily shaped domains, a gradient based algorithm that solves for optimal locations on a fixed mesh has been developed.

In the following, we give a brief overview of the unconstrained and bound constrained minimization methods used. The gradient based algorithm for arbitrarily domains is described in detail.

3.5.1 Unconstrained Minimization by BFGS Method

Solution to the unconstrained minimization problem requires solving the necessary conditions Equation $\nabla f(\underline{x}) = 0$. Newtons method yields the following iterative scheme

$$\underline{x}_{k+1} = \underline{x}_k + \underline{s}_k^N \quad (3.4)$$

where, $\underline{x}_k$ is the current iterate, $\underline{x}_{k+1}$ is the next iterate and $\underline{s}_k^N = -\underline{H}_k^{-1} \nabla f(\underline{x}_k)$ is the newton step with $\underline{H}_k = \nabla^2 f(\underline{x}_k)$ being the $(n \times n)$ Hessian matrix. It is well known

that Newton's method is only *locally* quadratically convergent and hence requires initial guesses close to the solution. Outside the convergence region of Newton's method a global strategy is required to provide solution estimates. A natural global strategy for minimization problems is to make sure that each step results in a decrease in the value of the objective function f . This can be achieved by taking steps in the descent direction of f . Mathematically, $\underline{p}$ is a descent direction from $\underline{x}_k$ if the directional derivative from $\underline{x}_k$ in the direction of $\underline{p}$ is negative, i.e.,

$$\nabla f(\underline{x}_k)^T \underline{p} < 0 \quad (3.5)$$

The Newton direction, $\underline{s}_k^N$ is a descent direction provided $\underline{H}_k$ is positive definite. However, far from the minimizer, $\underline{s}_k^N$ may not be a descent direction because $\underline{H}_k$ may not be positive definite. In addition, the step size $\underline{s}_k^N$ may be too large for the local convergence theory to apply. This leads to the failure of the ordinary Newton method when the initial iterate is far from the minimizer. Globally convergent modifications to the Newton method attempt to solve the step-size problem by employing a line search procedure to backtrack systematically and reduce the step-size. Line searches solve for a λ_k that minimizes $\hat{f}(\lambda) = f(\underline{x}_k + \lambda \underline{s}_k^N)$, which is the one-dimensional restriction of f to the line through $\underline{x}_k$ in the descent direction $\underline{s}_k^N$. A polynomial model for $\hat{f}(\lambda)$ is generally used. In addition, the step-size is required to satisfy a sufficient decrease condition or *Armijo's rule* [29] which is

$$f(\underline{x}_k + \lambda \underline{s}_k^N) - f(\underline{x}_k) < \alpha \lambda \nabla f(\underline{x}_k)^T \underline{s}_k^N$$

where, $\alpha \in (0, 1)$ is an algorithmic parameter and is typically 10^{-4} . The scheme for updating the Hessian such that it remains positive definite as the iteration progresses determines the optimization method. The BFGS or the positive definite secant update has been the most successful update for the Hessian and is used in this work. The complete iterative procedure with the BFGS update is as follows :

1. $\underline{x}_0$ = initial guess; $\underline{H}_0 = \underline{I}$
2. do while ($\|\nabla f(\underline{x}_k)\| > \text{tolerance}$.and. $k \leq \text{maxit}$)
 - (a) $\underline{x}_{k+1} = \underline{x}_k - \lambda_k \underline{H}_k^{-1} \nabla f(\underline{x}_k)$ (λ_k computed by line search)
 - (b) $\underline{s}_k = \underline{x}_{k+1} - \underline{x}_k$; $\underline{y}_k = \nabla f(\underline{x}_{k+1}) - \nabla f(\underline{x}_k)$
 - (c) $\underline{H}_{k+1} = \underline{H}_k + \frac{\underline{y}_k \underline{y}_k^T}{\underline{y}_k^T \underline{s}_k} - \frac{\underline{H}_k \underline{s}_k \underline{s}_k^T \underline{H}_k}{\underline{s}_k^T \underline{H}_k \underline{s}_k}$

where k is the iteration index and $maxit$ is the maximum number of iterations permissible. The iteration is started with the initial guess and the identity matrix $\underline{I}$ as the initial Hessian. In each iteration a full Newton step ($\lambda_k = 1$) is always tried first. If this fails, the step size is reduced by employing a line search which solves for a λ_k that insures sufficient decrease. The iteration is terminated when $\|\nabla f(\underline{x}_k)\| \leq$ stopping tolerance or when k exceeds $maxit$. More detailed analysis of this method can be found in [28] and [29].

3.5.2 Minimization with Simple Bound Constraints

When simple bounds on the variables are available, the minimization problem in Equation 3.1 can be written as

$$\min f(\underline{x}) \quad (3.6)$$

$$\text{subject to } \underline{l} \leq \underline{x} \leq \underline{u} \quad (3.7)$$

where, the vectors $\underline{l}$ and $\underline{u}$ represent lower and upper bounds on the variables.

The optimization problem is solved using the L-BFGS-B code developed by Zhu et. al. [43]. L-BFGS-B implements a limited memory algorithm for large scaled bound constrained optimization which is described in Byrd et. al. [44]. We give below an outline of the algorithm.

At the beginning of each iteration, we have the current iterate $\underline{x}_k$, the function value f_k , the gradient $\nabla f(\underline{x}_k)$, and a positive definite limited memory approximation to the hessian H_k . From this the following quadratic model of f at x_k is formed.

$$m_k(\underline{x}) = f(\underline{x}_k) + \nabla f(\underline{x}_k)^T (\underline{x} - \underline{x}_k) + \frac{1}{2} (\underline{x} - \underline{x}_k)^T H_k (\underline{x} - \underline{x}_k) \quad (3.8)$$

Next, the algorithm approximately minimizes $m_k(\underline{x})$ subject to the bounds given by Equation 3.7. This is done by first using the gradient projection method to find a set of active bounds, followed by minimization of m_k treating this bounds as equality constraints. To do this, first a piecewise linear path

$$\underline{x}(t) = P(\underline{x}_k - t \nabla f(\underline{x}_k), \underline{l}, \underline{u}) \quad (3.9)$$

obtained by projecting the steepest descent direction onto the feasible region, where

$$P(\underline{x}, \underline{l}, \underline{u})_i = \begin{cases} l_i & \text{if } x_i < l_i, \\ x_i & \text{if } x_i \in [l_i, u_i], \\ u_i & \text{if } x_i > u_i. \end{cases}$$

is considered. The generalized Cauchy point $\underline{x}^c$, which is defined as the first local minimizer of the univariate, piecewise quadratic

$$q_k(t) = m_k(\underline{x}(t))$$

is computed.

The variables whose value at $\underline{x}^c$ is at lower or upper bound, comprise the active set $\mathcal{A}(\underline{x}^c)$ and are held fixed. Next, the following quadratic problem over the subspace of free variables is considered,

$$\min \{m_k(\underline{x}) : x_i = x_i^c \forall i \in \mathcal{A}(\underline{x}^c)\} \quad (3.10)$$

$$\text{subject to } l_i \leq x_i \leq u_i \forall i \notin \mathcal{A}(\underline{x}^c). \quad (3.11)$$

Equation 3.10 is solved approximately by ignoring the bounds on the free variables. With the approximate solution $\underline{x}'_{k+1}$, the new iterate $\underline{x}_{k+1}$ is computed by a line search subject to the bounds on the problem along $\underline{d}_k = \underline{x}'_{k+1} - \underline{x}_k$ that satisfies the sufficient decrease condition

$$f(\underline{x}_{k+1}) \leq f(\underline{x}_k) + \alpha \lambda_k \nabla f(\underline{x}_k)^T \underline{d}_k$$

and the curvature condition

$$|\nabla f(\underline{x}_k)^T \underline{d}_k| \leq \beta |\nabla f(\underline{x}_k)^T \underline{d}_k|$$

where λ_k is the steplength and α, β are parameters that have values 10^{-4} and 0.9 in the code. The gradient at $\underline{x}_{k+1}$ is evaluated and a new limited memory Hessian approximation H_{k+1} is computed to begin a new iteration. The Hessian approximations are limited memory BFGS matrices (see Byrd et. al. [45]) that require small amount of storage that permit efficient subspace minimization and computation of the generalized Cauchy point.

Two stopping tests are built into the code. The first stopping test is

$$\frac{(f_k - f_{k+1})}{\max(|f_{k+1}, f_k, 1|)} \leq factr * epsmch$$

where *epsmch* is the machine precision and *factr* is a parameter controlled by the user. The test is designed to terminate the run when the change in the objective function f is sufficiently small. Typical values for *factr* on a computer with 15 digits of accuracy in double precision are : *factr* = 1.d+12 for low accuracy; *factr* = 1.d+7

for moderate accuracy; $factr = 1.d+1$ for extremely high accuracy. If $factr = 0$, the test will stop the algorithm only if the objective function remains unchanged after one iteration.

The second built-in stopping test is based on the projected gradient. This is the projection of the gradient onto the space tangent to the active bounds, and it must equal zero at a local minimizer of the bound constrained problem. The test is designed to terminate the run when the infinity norm of the projected gradient becomes sufficiently small,

$$\|proj \nabla f(\underline{x}_k)\|_{\inf} \leq pgtol$$

where, the parameter $pgtol$ is controlled by the user. The test will be hard to satisfy if $pgtol$ is set smaller than the square root of machine precision.

3.5.3 Gradient Based Method for a Fixed Mesh

This method is used to solve for optimal locations on domains of arbitrary shape for which automatic mesh generation is extremely difficult. Optimal locations are solved on a fixed mesh and the solution space is a discrete space consisting of the coordinates of nodes in the mesh. The method consists of a systematic search procedure using the steepest descent direction to find the next set of nodes amongst the closest nodes to the current nodes that result in a decrease in objective function. For a more accurate solution, the mesh is refined in selective regions and problem resolved.

The inputs are the mesh of the limp part, material properties and initial guesses (xy coordinates) of the pick-up locations. To describe the algorithm, we use the following notation :

$\underline{N}$ - vector of node numbers of dimension n = number of pick-up locations

$f(\underline{N})$ - strain energy computed from finite element solution with boundary conditions applied at nodes $\underline{N}$

$\nabla f(\underline{N})$ - gradient of strain energy with boundary conditions applied at nodes $\underline{N}$.

This is a $(2n \times 1)$ vector and is computed by finite differences. For this purpose, the x and y coordinates of each node of $\underline{N}$ is perturbed and a new finite element solution is computed with each perturbation.

The algorithm is as follows :

-
1. Read mesh data, physical properties and initial guess
 2. Find $\underline{N}$ = nodes in the mesh closest to initial guess (xy coordinates) of pick-up locations
 3. Compute $f_{old} = f(\underline{N})$ and $\nabla f(\underline{N})$
 4. While $k < \text{maximum number of iterations}$
 - a. Find $\underline{N}_s$ = nodes closest to $\underline{N}$ in the steepest descent (negative gradient) direction
 - b. $\underline{N}_t = \underline{N}_s$; compute $f_{new} = f(\underline{N}_t)$
 - c. If ($f_{new} < f_{old}$) then
 - i) $\underline{N} = \underline{N}_t$; $f_{old} = f_{new}$
 - ii) Compute $\nabla f(\underline{N})$
 - iii) $k = k + 1$; Go to 4 a.
 - else
 - i) Find a new trial set of nodes $\underline{N}_t$ (explained below)
If no. of tries is $> maxtry$ then EXIT
 - ii) Compute $f_{new} = f(\underline{N}_t)$
 - iii) Go to 4 c.
 5. Refine mesh in selective areas and rerun to obtain a more accurate solution.

The nodes of $\underline{N}_s$ are found by computing the angles between the unit vectors in the steepest descent direction at the corresponding nodes of $\underline{N}$ and the unit vectors in the direction of the immediate surrounding nodes in the mesh. The closest node is the node which gives the smallest angle. This procedure is illustrated in figure 3.5.

The first trial set of nodes ($\underline{N}_t$) are the nodes in the steepest descent direction ($\underline{N}_s$). If this does not give a decrease in function value, then $\underline{N}_t$ is chosen to be a combination of nodes from $\underline{N}$ and $\underline{N}_s$. For example, one choice for $n = 4$ would be $\underline{N}_t = [N_s(1), N(2), N(3), N(4)]$, i.e. only the first node of $\underline{N}$ is replaced by the node in the steepest descent direction. The other tries involve replacing 2 nodes and 3 nodes of $\underline{N}$ by corresponding nodes in the steepest descent direction. The total number

Figure 3.5: Node 3 is the closest node in the steepest descent direction

of such combinations (*maxtry*) including the first trial set is given by $\sum_{k=1}^n C_k^n$. For $n = 4$ and $n = 3$, *maxtry* is 15 and 7 respectively. When $n = 1$, *maxtry* = 1 and the algorithm simplifies to trying only the closest node in the steepest descent direction.

Initially, the limp part is discretized by a coarse mesh and the algorithm is run for different initial guesses. Based on the results, the mesh is refined in selective regions and the algorithm is again run for different initial guesses. This process of iterative refinement and solution is carried on until an accurate enough solution is obtained. Symmetry in the problem is exploited whenever possible as it reduces the size of the problem and solution time.

3.6 Summary

In this chapter we have described the algorithms used in this work. Results obtained by applying these algorithms will be presented in chapters 5 and 6. The next chapter deals with the measurement of material properties of fabrics.

Chapter 4

Measurement of Fabric Material Properties

4.1 Introduction

One of the applications of this work, as mentioned in Chapter 1, is in the automation of manufacturing operations in the textile and apparel industries. Fabric parts of different shapes are required to be transported and handled by robotic devices. Picking up fabric parts such that they undergo minimum deformation is essential to the successful automation of these operations. By obtaining optimal locations for fabric parts of different shapes, we intend to demonstrate the practical use of this work. With this in mind, material properties of different fabrics were measured.

Although fabrics are nonhomogeneous materials, the continuum based model described in Chapter 2 is sufficient to predict their deformation or drape. This was demonstrated in the work of Deng[32] in which numerically predicted fabric drape was found to compare well with the experimental result. Fabrics also exhibit nonlinear material response. This is well known from the relationship between the moment and curvature. The slope of the moment-curvature curve is the bending rigidity. Deng [32] implemented the nonlinear response and found that both linear and nonlinear material responses matched experimental results almost equally. In the present work, linear isotropic material response with constant bending rigidity is used to model the fabrics. Sample fabrics were obtained from the Apparel Lab in the College of Textiles and Medcovers, a company that manufactures equipment covers. The FAST (Fabric Assurance by Simple Testing) [46] system was used to measure the material properties.

In this chapter, the procedures for measurement of thickness and bending rigidity are first described. Next, the method for calculating elastic properties from the

Figure 4.1: Cantilever Bending Test

weight, thickness and bending rigidity is described. Finally, properties of various fabrics are listed.

4.2 Measurements by the FAST System

The thickness and bending rigidity were measured using the compression meter and bending meter of the FAST system respectively.

FAST-1 Compression Meter

The compression meter measures the thickness of the fabric under loads of 2 gmf/cm^2 and 100 gmf/cm^2 . A sample is placed on the compression meter reference surface and the weight is lowered on to the sample. Since we are modeling fabric drape under self weight, the measurement under the lighter load of 2 gmf/cm^2 was taken as the fabric thickness. Three $100 \text{ mm} \times 100 \text{ mm}$ samples of each fabric were used for the measurement. The thickness was measured at 5 different places on each sample and the readings were averaged.

FAST-2 Bending Meter

The bending meter is a setup to perform the ASTM D 1388 cantilever fabric bending rigidity test. In this test, a strip of fabric is allowed to bend under its own weight by pushing it off a horizontal surface until the leading edge intercepts the plane at an angle of 41.5° from the horizontal surface. The length of the overhang portion

is the bending length and the procedure is depicted in Figure 4.1. In the FAST-2 system, the edge of the fabric is detected using a photocell and the length is measured in millimeters.

The bending rigidity per unit width B is calculated from the bending length and weight/area W of the fabric as follows :

$$B (\mu\text{Nm}) = W (\text{gm/m}^2) \times [\text{Bending Length (mm)}]^3 \times 9.807 \times 10^{-6}$$

Three strips (130mm x 50mm) each in the warp and weft directions were used for the measurements. For each strip, the bending length was measured on both the face and back of the fabric. The measurements were averaged to determine the bending lengths in the warp and weft direction. The weight/area of the fabric was determined by weighing a 100mm x 100mm sample.

4.3 Fabric Material Properties

The second moment of area I is calculated from the thickness t and the width b of the strip used for measuring the bending rigidity as follows :

$$I = \frac{1}{12}bt^3$$

Next, the Young's modulus E is calculated as follows :

$$E = \frac{Bb}{I}$$

Since isotropic material properties are assumed, the shear modulus G is given by

$$G = \frac{E}{2(1 + \nu)}$$

Following the work of Kim [31], we assumed the Poisson's ratio $\nu = 0.3$. Material properties of 5 different fabrics are listed in Table 4.3.

4.4 Summary

In this chapter, we described the method used to determine the material properties of fabrics. These properties were used to obtain results for fabric parts of different shapes.

Table 4.1: Fabric Material Properties

Fabric Type	Thickness (mm)	Weight/area (gmf/cm ²)	Youngs Modulus (gmf/cm ²)	
			Warp	Weft
Blue Denim	1.142	0.0431	44499.27	15646.2
White Denim	1.199	0.0429	24369.97	15428.19
Cotton Twill Style 423	0.601	0.0264	75507.65	34091.23
Cotton Polyester Style 7436	0.402	0.0156	21089.3	19206.36
1000 Denier Cordura	0.594	0.0312	83760.48	78623.95

Chapter 5

Optimal Locations for 1 Dimensional Domains

5.1 Introduction

A strip of limp material can be considered one dimensional and modeled as a flexible beam. Hence, the problem of optimal pick-up locations that minimize the strain energy of a strip is the same as that of optimal support locations that minimize the strain energy of continuous beams with uniformly distributed load. Figure 5.1 shows a continuous beam of length l supported at n points. x_i 's are the support locations measured from the left end and w is the distributed load per unit length. The support locations correspond to the pick-up locations and the distributed load is the self weight of the strip. Optimal locations are found by solving the minimization problem in Equation 3.1 on the one dimensional domain $[0, l]$.

We first consider the problem for beams undergoing small deformations. For this, an analytical solution exists for two support locations. But, when the number of supports is greater than two, the problem has to be solved numerically. For beams undergoing large deformations, the problem can only be solved numerically. Optimization algorithms from Chapter 3 have been used with the objective function computed from a finite element solution to solve for optimal locations. Beams undergoing small deformations are modeled by standard beam elements whereas the geometrically exact finite element formulation described in Chapter 2 is used to model beams undergoing large deformations. In Chapter 1, we reviewed some of the literature available on beam optimization problems. To the best of our knowledge, optimal support locations that minimize the strain energy of continuous beams has not been solved in literature.

In this chapter, we first present results for beams undergoing small deforma-

Figure 5.1: Beam supported at n positions

tions. This is followed by results for beams undergoing large deformations. Finally, results for fabric strips are presented.

5.2 Beams Undergoing Small Deformations

The strain energy (U_m) by small deflection beam theory is given by the well known formula

$$U_m = \int_0^l \frac{M^2(x)}{2EI} dx \quad (5.1)$$

where $M(x)$ is the bending moment, I the second moment of area and E Young's modulus. When $n = 2$, the problem is statically determinate and the following expression for strain energy can be written down by evaluating Eqn. (5.1)

$$\begin{aligned} U_m = & \frac{w^2 l^5}{240EI} \left\{ 10\left(\frac{x_1}{l}\right)^3 \left(\frac{x_2}{l}\right) - 5\left(\frac{x_1}{l}\right)^3 - 5\left(\frac{x_1}{l}\right)^2 \left(\frac{x_2}{l}\right) + 10\left(\frac{x_1}{l}\right)^2 \left(\frac{x_2}{l}\right)^2 \right. \\ & + 40\left(\frac{x_1}{l}\right) \left(\frac{x_2}{l}\right) - 45\left(\frac{x_1}{l}\right) \left(\frac{x_2}{l}\right)^2 - 10\left(\frac{x_1}{l}\right) + 10\left(\frac{x_1}{l}\right) \left(\frac{x_2}{l}\right)^3 + 20\left(\frac{x_2}{l}\right)^2 \\ & \left. - 5\left(\frac{x_2}{l}\right)^3 + 6 - 20\left(\frac{x_2}{l}\right) \right\} \end{aligned} \quad (5.2)$$

The necessary condition for a local minimizer yields the following two nonlinear simultaneous equations.

$$\begin{aligned} \frac{\partial U_m}{\partial(\frac{x_1}{l})} = & 30(\frac{x_2}{l})(\frac{x_1}{l})^2 - 15(\frac{x_1}{l})^2 - 10(\frac{x_1}{l})(\frac{x_2}{l}) + 20(\frac{x_1}{l})(\frac{x_2}{l})^2 + 40(\frac{x_2}{l}) - 45(\frac{x_2}{l})^2 \\ & - 10 + 10(\frac{x_2}{l})^3 = 0 \end{aligned} \quad (5.3)$$

$$\begin{aligned} \frac{\partial U_m}{\partial(\frac{x_2}{l})} = & 10(\frac{x_1}{l})^3 - 5(\frac{x_1}{l})^2 + 20(\frac{x_2}{l})(\frac{x_1}{l})^2 + 40(\frac{x_1}{l}) - 90(\frac{x_1}{l})(\frac{x_2}{l}) + 30(\frac{x_1}{l})(\frac{x_2}{l})^2 \\ & + 40(\frac{x_2}{l}) - 15(\frac{x_2}{l})^2 - 20 = 0 \end{aligned} \quad (5.4)$$

The roots of Eqns. (5.3) and (5.4) obtained using the *fsolve* function of MAPLE with $\frac{x_1}{l} \in [0, 0.5]$ and $\frac{x_2}{l} \in [0.5, 1.0]$ are

$$\left(\frac{x_1}{l}\right)_m = 0.2247 \quad \left(\frac{x_2}{l}\right)_m = 0.7753$$

Figure 5.2 shows a plot of U_m (with $\frac{w^2 l^5}{240EI} = 1.0$) as a function of the support locations. It can be verified from the plot that $(\frac{x_1}{l})_m$ and $(\frac{x_2}{l})_m$ are indeed the global minimizers.

When $n \geq 3$, the problem is statically indeterminate and to apply the same procedure as above we need to determine the bending moments in each of the spans. This can be done by first solving for the moments at the supports using the *three moment equation* (Gere and Timoshenko, 1990), then solving for the support reactions and then obtaining expressions for the bending moments in each of the spans. The strain energy can then be obtained by evaluating Eqn. (5.1) as before and the resulting n equations from the necessary conditions have to be solved numerically. It is easy to realize that the procedure is cumbersome and will lead to long expressions with increasing complexity as n increases and ultimately the solution is to be obtained numerically. However, by discretizing the beam by finite elements, the strain energy can be computed in a straightforward manner. The optimal locations can then be solved by using a minimization algorithm from chapter 3 with the objective function computed from a finite element solution. As will be seen, this is a much simpler procedure and yields quick results.

5.2.1 Numerical Solution Procedure

Standard beam elements are used to discretize the beam and the strain energy (U_m) is computed by the following formula

$$U_m = \frac{1}{2} \underline{d}^T \underline{K} \underline{d} \quad (5.5)$$

where, $\underline{K}$ is the stiffness matrix and $\underline{d}$ is the vector of nodal displacements and rotations obtained after solving the finite element equations. The gradient is obtained by finite differences by the method described in chapter 3. The automatic meshing procedure described in chapter 3 enables us to solve the problem as a continuous optimization problem.

It was possible to solve for optimal locations using an unconstrained optimization method. The bound constraints were never violated during the optimization process. All the numerical computations were done in MATLAB. For this purpose, a MATLAB function that computes the strain energy of a beam by the finite element method given a vector of support locations was developed. The BFGS method for unconstrained minimization was used to solve for the optimal locations. MATLAB code `bfgswopt` from Kelley [29] implements the algorithm and was used. The gradient was computed by central differences by equation 3.3 and the stopping tolerance for the optimization routine was taken such that the norm of the gradient is reduced by at least 5 orders of magnitude.

A level of discretization of at least 10 elements per span of the beam was used. In some cases, a finer mesh was required to satisfy the stopping criteria. The initial guess was chosen to be the centers of each of the p equal divisions of the beam where p is the number of supports. For example, when $p = 2$, the initial guess was $[0.25l, 0.75l]$ where l is the length of the beam. Any other initial guess resulted in a greater number of iterations to converge to the solution.

5.2.2 Results

Optimal support locations for minimum strain energy up to 8 supports have been obtained. Table 5.2.2 lists numerical values. Solution for the $n = 2$ statically determinate case can be verified to be the same as the one obtained previously by analytical methods. Deflection curves corresponding to the optimal locations are shown in Figures 5.3 and 5.4. The optimal locations are symmetrically placed about the centerline, as expected. Spacings between adjacent internal support locations are equal and as n increases the outer two support locations move closer to the ends. Figures 5.5 and 5.6 show the objective function and gradient norm respectively during the optimization. The number of support locations are indicated besides the corresponding lines. It is to be noted that the magnitude of both the objective function and the gradient norm are small and hence logarithmic scale has been used along the y axis. In general,

as the number of support locations increase the algorithm takes more iterations to achieve the desired reduction in the gradient norm.

Figure 5.2: Strain energy for 2 Supports

Table 5.1: Optimal Locations for Small Deformations

Location	Number of supports						
	2	3	4	5	6	7	8
$\frac{x_1}{l}$	0.2247	0.1450	0.1070	0.0848	0.0702	0.0599	0.0522
$\frac{x_2}{l}$	0.7753	0.5	0.3690	0.2924	0.2421	0.2066	0.1802
$\frac{x_3}{l}$	-	0.8550	0.6310	0.5	0.4140	0.3533	0.3081
$\frac{x_4}{l}$	-	-	0.8930	0.7076	0.5860	0.5	0.4360
$\frac{x_5}{l}$	-	-	-	0.9152	0.7579	0.6467	0.5640
$\frac{x_6}{l}$	-	-	-	-	0.9298	0.7934	0.6919
$\frac{x_7}{l}$	-	-	-	-	-	0.9401	0.8198
$\frac{x_8}{l}$	-	-	-	-	-	-	0.9478

Figure 5.3: Deflection curves at optimal locations (small deformations)

Figure 5.4: Deflection curves at optimal locations (small deformations)

Figure 5.5: Plot of objective function v/s number of iterations

Figure 5.6: Plot of gradient norm v/s number of iterations

5.3 Beams Undergoing Large Deformations

It is not possible to write down closed form expressions for strain energy for beams undergoing large deformations. Hence, the optimal locations are numerically solved for using the bound constrained optimization described in Chapter 3 with the objective functions computed from a finite element solution.

5.3.1 Numerical Solution Procedure

The geometrically exact nonlinear shell finite element formulation described in Chapter 2 is used to model beams undergoing large deformations. Beam theory is recovered by setting Poissons ratio $\nu = 0$ in the shell formulation and the beam is discretized by quadrilateral shell elements. Boundary conditions at the support points restrain both transverse and axial displacements. The strain energy is computed by the method described in Chapter 2. A nondimensional parameter $\alpha = \frac{wl^3}{EI}$ is used to characterize flexibility of the beam. The greater the value of α , the larger the deformation the beam undergoes.

The automatic meshing algorithm described in Chapter 3 enables us to solve the problem as a continuous optimization problem. L-BFGS-B code for bound constrained optimization described in Chapter 3 was used to solve for the optimal locations. All variables were bounded between 0 and l (i.e. $0 \leq x_i \leq l$) and *factr* = 1.0 and *pgtol* = $1e^{-8}$ were used for specifying the stopping tests. The gradient was computed by forward differences using equation 3.2. A level of discretization of at least 10 elements per span of the beam was used. Different initial guesses were tried and all of them resulted in the same solution. A sample output of the optimization process is shown below for the problem of 4 locations and $\alpha = 1200$.

INPUT:

Arc-length parameters : $\lambda = 0.01$; $\beta = 8.0$

No. of Elements : 60

Initial guess : [0.125 0.375 0.625 0.875]

RUNNING THE L-BFGS-B CODE

* * *

Machine precision = 2.220D-16

N = 4 M = 5

At X0 0 variables are exactly at the bounds

At iterate	0	f=	3.57752D-04	proj g =	3.17459D-03
At iterate	1	f=	3.38046D-04	proj g =	3.04440D-03
At iterate	2	f=	2.40247D-04	proj g =	8.67040D-04
At iterate	3	f=	2.13471D-04	proj g =	5.80420D-04
At iterate	4	f=	2.03237D-04	proj g =	5.59609D-05
At iterate	5	f=	2.03169D-04	proj g =	1.73712D-05
At iterate	6	f=	2.03152D-04	proj g =	1.63964D-05
At iterate	7	f=	2.03116D-04	proj g =	1.60157D-05
At iterate	8	f=	2.03042D-04	proj g =	3.99233D-05
At iterate	9	f=	2.02906D-04	proj g =	3.29067D-05
At iterate	10	f=	2.02885D-04	proj g =	5.34347D-06
At iterate	11	f=	2.02885D-04	proj g =	1.11137D-07
At iterate	12	f=	2.02885D-04	proj g =	9.06953D-08
At iterate	13	f=	2.02885D-04	proj g =	9.07052D-08

* * *

Tit = total number of iterations

Tnf = total number of function evaluations

Tnint = total number of segments explored during Cauchy searches

Skip = number of BFGS updates skipped

Nact = number of active bounds at final generalized Cauchy point

Projg = norm of the final projected gradient

F = final function value

```

      * * *
      N   Tit   Tnf   Tnint   Skip   Nact       Projg       F
      4    13    28     13     0      0    9.071D-08    2.029D-04
      F = 2.0288502958946432E-04

```

CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH

Warning: more than 10 function and gradient
evaluations in the last line search. Termination
may possibly be caused by a bad search direction.

Cauchy time 7.401E-04 seconds.
Subspace minimization time 1.007E-03 seconds.
Line search time 1.868E+03 seconds.
Total User time 1.985E+03 seconds.

```

it      = iteration number
nf      = number of function evaluations
nint    = number of segments explored during the Cauchy search
nact    = number of active bounds at the generalized Cauchy point
sub     = manner in which the subspace minimization terminated:
          con = converged, bnd = a bound was reached
itls    = number of iterations performed in the line search
stepl   = step length used
tstep   = norm of the displacement (total step)
projg   = norm of the projected gradient
f       = function value

```

```

      * * *
Machine precision = 2.220D-16
      N =          4      M =          5
it   nf   nint   nact   sub   itls   stepl   tstep   projg       f
0    1    -     -     -     -     -     -     3.175D-03  3.578D-04
1    2    1     0    ---    0    1.0D+00  4.5D-03  3.044D-03  3.380D-04
2    3    1     0   con    0    1.0D+00  1.1D-01  8.670D-04  2.402D-04
3    4    1     0   con    0    1.0D+00  2.6D-02  5.804D-04  2.135D-04

```

4	6	1	0	con	1	4.2D-01	2.1D-02	5.596D-05	2.032D-04
5	7	1	0	con	0	1.0D+00	2.2D-03	1.737D-05	2.032D-04
6	8	1	0	con	0	1.0D+00	6.0D-04	1.640D-05	2.032D-04
7	9	1	0	con	0	1.0D+00	1.9D-03	1.602D-05	2.031D-04
8	10	1	0	con	0	1.0D+00	5.6D-03	3.992D-05	2.030D-04
9	11	1	0	con	0	1.0D+00	4.2D-03	3.291D-05	2.029D-04
10	12	1	0	con	0	1.0D+00	8.2D-04	5.343D-06	2.029D-04
11	13	1	0	con	0	1.0D+00	1.6D-04	1.111D-07	2.029D-04
12	17	1	0	con	3	1.3D-01	6.1D-06	9.070D-08	2.029D-04
13	28	1	0	con	10	5.9D-05	1.6D-09	9.071D-08	2.029D-04

5.3.2 Results

Optimal support locations for minimum strain energy up to 4 supports have been obtained. Tables 1.2, 1.3 and 1.4 show optimal locations for different values of α for $n = 2, 3$ and 4 supports respectively. The locations are symmetrically placed and as the value of α increases, the end support locations move towards the two ends of the beam. The spacings between adjacent supports need not be equal as is the case with beams undergoing small deflections. Typical deflection curves for $n = 2, 3$ and 4 supports are shown in Figures 5.7, 5.8 and 5.9 respectively. An interesting feature is that the deflection can be up or down along the length of the beam for large deformations whereas the deflection is always down for small deformations.

Table 5.2: Optimal Locations for 2 Supports (Large Deformations)

Location	α						
	7.68	61.44	120.0	491.53	1000.0	1200.0	1500.0
$\frac{x_1}{l}$	0.2246	0.1879	0.1639	0.1255	0.1098	0.1071	0.1038
$\frac{x_2}{l}$	0.7754	0.8121	0.8361	0.8745	0.8902	0.8929	0.8962

Table 5.3: Optimal Locations for 3 Supports (Large Deformations)

Location	α						
	7.68	61.44	120.0	491.53	1000.0	1200.0	1500.0
$\frac{x_1}{l}$	0.1451	0.1440	0.1400	0.1022	0.0880	0.0855	0.0827
$\frac{x_2}{l}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$\frac{x_3}{l}$	0.8549	0.8560	0.8600	0.8978	0.9120	0.9145	0.9173

Table 5.4: Optimal Locations for 4 Supports (Large Deformations)

Location	α						
	7.68	61.44	120.0	491.53	1000.0	1200.0	1500.0
$\frac{x_1}{l}$	0.1074	0.1068	0.1064	0.0937	0.0800	0.0777	0.0738
$\frac{x_2}{l}$	0.3694	0.3690	0.3688	0.3802	0.3937	0.3929	0.3710
$\frac{x_3}{l}$	0.6306	0.6310	0.6312	0.6198	0.6023	0.6071	0.6290
$\frac{x_4}{l}$	0.8926	0.8932	0.8936	0.9063	0.9100	0.9223	0.9262

Figure 5.7: Deflection curves for 2 supports for different values of α (large deformations)

Figure 5.8: Deflection curves for 3 supports for different values of α (large deformations)

Figure 5.9: Deflection curves for 4 supports for different values of α (large deformations)

5.4 Fabric Strips

Optimal locations for fabric strips were obtained by using the material properties from Chapter 4. Young's modulus in the warp direction was used in all the results. Figures 5.10, 5.11 and 5.12 show deformed shapes of different fabric strips when picked at optimal locations. Dimensions of the strip and numerical values of optimal locations are provided below each figure.

5.5 Summary

In this chapter the algorithms described in Chapter 3 were used to obtain optimal locations for one dimensional domains. In the next chapter we present results for two dimensional domains.

Figure 5.10: 2 Pick-up locations for a Blue Denim strip

Strip dimensions :

$$l = 30.5 \text{ cm}$$

$$b = 3.0 \text{ cm}$$

$$\alpha = 221.4$$

Optimal locations :

$$x_1 = 3.62 \text{ cm}$$

$$x_2 = 26.88 \text{ cm}$$

Figure 5.11: 3 Pick-up locations for a Cotton Twill strip

Strip dimensions :

$$l = 40.0 \text{ cm}$$

$$b = 3.0 \text{ cm}$$

$$\alpha = 1236.9$$

Optimal locations :

$$x_1 = 2.56 \text{ cm}$$

$$x_2 = 20.0 \text{ cm}$$

$$x_3 = 37.44 \text{ cm}$$

Figure 5.12: 4 Pick-up locations for a Cotton Polyester strip

Strip dimensions :

$$l = 40.0 \text{ cm}$$

$$b = 5.0 \text{ cm}$$

$$\alpha = 8744.7$$

Optimal locations :

$$x_1 = 1.54 \text{ cm}$$

$$x_2 = 13.87 \text{ cm}$$

$$x_3 = 26.13 \text{ cm}$$

$$x_4 = 38.46 \text{ cm}$$

Chapter 6

Optimal Locations for 2 Dimensional Domains

6.1 Introduction

In this work, two dimensional parts are modeled as shells undergoing large deformations. Hence, while solving for optimal locations, we will encounter the typical nonlinear phenomena (limit points and bifurcation points) associated with shells. We mentioned in Chapter 2 that the arc-length solution procedure can diverge when a bifurcation instability is encountered. This typically occurs in materials with low bending rigidity and is also dependent on the shape of the domain and the pick-up location. We also said that it is impractical to implement methods to detect the point of bifurcation and trace a particular branch in an optimization procedure. Thus, the results that can be obtained will be limited by the convergence properties of the arc-length procedure.

Since our numerical experience indicated that divergence occurs mostly in materials of low bending rigidity, we started by solving for optimal locations for relatively stiff parts. In this case, the deformations are small and the objective function is computed by solving a linear problem. This allowed us to quickly test the optimization algorithms described in Chapter 3. Then, we systematically lowered the bending rigidity and solved for optimal locations while the objective function was computed from the solution to a nonlinear problem. We lowered the bending rigidity until divergence in the arc-length procedure occurred frequently and stalled the optimization procedure. Whenever possible, symmetry in the problem was exploited to reduce the size of the problem. The problem of 4 pick-up locations for a square/rectangular domain is solved as a continuous optimization problem using symmetry and automatic meshing. We use, this problem to test the gradient based optimization algorithm for a

fixed mesh described in Chapter 4. Optimal locations for all other domains are solved on a fixed mesh with iterative mesh refinement by the gradient based algorithm.

In this chapter, we first present results for square/ rectangular domains. This is followed by results for circular domains. Finally, results for some arbitrarily shaped domains obtained from practical applications are presented.

6.2 4 Pick-up Locations for Square/Rectangular Domains

The problem of 4 pick-up locations on square/rectangular domains can be simplified to 1 pick-up location on a quarter domain by exploiting symmetry as shown in Figure 6.1. It is sufficient to solve for the optimal location (x_1, y_1) on OABC by enforcing symmetric boundary conditions along edges OA and OC. The problem is solved as a continuous optimization problem using the L-BFGS-B and automatic meshing algorithms described in Chapter 3.

We first solve the problem for a square domain with material properties corresponding to high bending rigidity. Since the deformations are small, the objective function is computed from a single load step and this allows us to quickly check our algorithms. Then, we gradually lower the bending rigidity and solve for optimal locations. The effect of bending rigidity on optimal locations is presented in terms of the ratio between the length and thickness and the following nondimensional parameter,

$$\gamma = \frac{wl^3}{D} \quad (6.1)$$

where,

w = weight per unit area

l = length of the square ($2a = 2b = l$)

$D = \frac{Et^3}{12(1-\nu^2)}$ is the bending rigidity, t = thickness, E = Young's modulus

The higher the value of γ , the lower is the bending rigidity and greater the deformations. We also study the effect of different aspect ratios on optimal locations.

Figure 6.1: 4 pick-up locations on a rectangular domain

6.2.1 Small deformations

We consider a square domain of length $l = 2.0\text{m}$ and thickness $t = 0.1\text{m}$. Material properties are chosen such that $\gamma = 0.12$. We solve the problem by three different methods in order to test the algorithms developed in Chapter 3. First, we solve the problem as a continuous optimization problem. Next, the problem is solved on a fixed mesh with iterative mesh refinement. In this case, we solve for 1 location using symmetry and for all 4 locations on the entire domain. The results from all three methods are then compared.

6.2.1.1 Solution as a Continuous Problem

The L-BFGS-B code and automatic meshing procedure described in Chapter 3 are used to solve the problem as a continuous optimization problem. A desired level

of discretization of 10 elements along the length of the square was specified in the automatic meshing procedure. The gradient was computed by finite differences using Eqn. 3.2. Solutions from different initial guesses converged to any accuracy within 4 decimal places. The optimal pick-up location is at

$$x_1 = 0.54505 \text{ m}; y_1 = 0.54505 \text{ m}$$

Figure 6.2 shows the paths of the iterations starting from different initial guesses represented by different colors plotted on the square domain. A sample output from a initial guess of (1.0,0.75) corresponding to the red line in Figure 6.2 is shown below. Note that the x-coordinate of the initial guess, $x(1)$, is on the boundary. In a subsequent function evaluation, $x(1)$ is found close enough to the boundary to cause a bad mesh to be generated by the automatic meshing procedure (see Figure 3.4 (d)). This is prevented by perturbing $x(1)$ away from the boundary in the direction of search. The clause “ modifying $x(1)$ ” in the output indicates the perturbation.

INPUT:

Arc-length parameters : $\lambda = 1.0$; $\beta = 0.0$

Desired no. of elements along length of square : 10

Bounds on variables : $0.0 \leq x_i \leq 1.0$; $i = 1, 2$

Initial guess : [1.0 0.75]

OUTPUT:

1.0000000000000000 0.7500000000000000

RUNNING THE L-BFGS-B CODE

* * *

Machine precision = 2.220D-16

N = 2 M = 5

At X0 1 variables are exactly at the bounds

At iterate 0 f= 1.27583D-03 |proj g|= 6.03118D-03
modifying x(1)

At iterate	1	f=	8.73076D-04	proj g =	4.42920D-03
At iterate	2	f=	4.77359D-04	proj g =	2.91699D-03
At iterate	3	f=	1.09476D-04	proj g =	9.31745D-04
At iterate	4	f=	5.37807D-05	proj g =	2.63827D-04
At iterate	5	f=	4.79032D-05	proj g =	4.35156D-05
At iterate	6	f=	4.77061D-05	proj g =	3.54324D-06
At iterate	7	f=	4.77051D-05	proj g =	3.71925D-07
At iterate	8	f=	4.77051D-05	proj g =	1.51825D-08
At iterate	9	f=	4.77051D-05	proj g =	7.13988D-10

* * *

Tit = total number of iterations

Tnf = total number of function evaluations

Tnint = total number of segments explored during Cauchy searches

Skip = number of BFGS updates skipped

Nact = number of active bounds at final generalized Cauchy point

Projg = norm of the final projected gradient

F = final function value

* * *

N	Tit	Tnf	Tnint	Skip	Nact	Projg	F
2	9	11	9	0	0	7.140D-10	4.771D-05

F = 4.7705109969954D-05

CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL

Cauchy time 3.383E-04 seconds.

Subspace minimization time 7.534E-04 seconds.

Line search time 1.970E+01 seconds.

Total User time 2.259E+01 seconds.

RUNNING THE L-BFGS-B CODE

it = iteration number

nf = number of function evaluations
 nint = number of segments explored during the Cauchy search
 nact = number of active bounds at the generalized Cauchy point
 sub = manner in which the subspace minimization terminated:
 con = converged, bnd = a bound was reached
 itls = number of iterations performed in the line search
 stepl = step length used
 tstep = norm of the displacement (total step)
 projg = norm of the projected gradient
 f = function value

* * *

Machine precision = 2.220D-16

N = 2 M = 5

it	nf	nint	nact	sub	itls	stepl	tstep	projg	f
0	1	-	-	-	-	-	-	6.031D-03	1.276D-03
1	2	1	0	---	0	1.0D+00	6.3D-03	4.429D-03	8.731D-04
2	4	1	0	con	1	5.0D+00	1.1D-01	2.917D-03	4.774D-04
3	5	1	0	con	0	1.0D+00	1.9D-01	9.317D-04	1.095D-04
4	6	1	0	con	0	1.0D+00	9.1D-02	2.638D-04	5.378D-05
5	7	1	0	con	0	1.0D+00	3.7D-02	4.352D-05	4.790D-05
6	8	1	0	con	0	1.0D+00	6.3D-03	3.543D-06	4.771D-05
7	9	1	0	con	0	1.0D+00	4.4D-04	3.719D-07	4.771D-05
8	10	1	0	con	0	1.0D+00	6.3D-05	1.518D-08	4.771D-05
9	11	1	0	con	0	1.0D+00	2.8D-06	7.140D-10	4.771D-05

CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL

Total User time 2.259E+01 seconds.

Figure 6.2: Iteration histories of different initial guesses when solving for 1 pick-up location on a quarter square.

6.2.1.2 Solution on a Fixed Mesh with Iterative Mesh Refinement

The gradient based method for a fixed mesh described in Chapter 3 was used to solve for the optimal locations. Figure 6.3 shows the iteration histories plotted on the mesh when we solve for 1 location using symmetry on the lower right quadrant of the square domain. Lines of different colors corresponding to different initial guesses trace the movement of the pick-up locations during the optimization procedure. Note that the lines pass through nodes as the pick-up locations can be enforced only at the nodes in the mesh. The solution from different initial guesses converges to the same node. The optimal location is

$$x_4 = 0.5418 \text{ m}; y_4 = -0.5456 \text{ m}$$

Table 6.1: Coordinates to which different initial guesses converge for the mesh in Figure 1.5

Color	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	Objective Function
Red	-0.5413	-0.5584	0.5525	-0.5371	0.5445	0.5505	-0.5454	0.5382	2.201E-04
Blue	-0.5399	-0.5444	0.5513	-0.5524	0.5445	0.5505	-0.5454	0.5382	2.206E-04
Green	-0.5413	-0.5584	0.5525	-0.5371	0.5445	0.5505	-0.5454	0.5382	2.201E-04

Figure 6.4 shows the iteration histories for different initial guesses on a coarse mesh when solving for all 4 locations. Since we are solving for 4 locations, 4 lines of the same color trace the movement of the 4 pick-up locations starting from an initial guess. Based on the result, the mesh was refined in selective regions and the problem resolved. Figure 6.5 shows the iteration histories on a fine mesh. It can be seen that the solution from different initial guesses converges to within an element. The final locations from different initial guesses along with the value of the objective function are shown in Table 6.1. There is a less than 1 percent difference in objective function between the solutions obtained from different initial guesses. Figure 6.6 shows the deformed shape at optimal locations.

The above exercise, besides providing the optimal locations demonstrates the success of our optimization procedures. Comparing the results from the gradient based method to that obtained using L-BFGS-B method, we note that the optimal locations match only up to 2 decimal places. This is to be expected because in the fixed mesh method, the solution is constrained to lie on a node in the mesh. Also, there is a slight difference between the solution with symmetry and without symmetry. This is because the two meshes are different.

Figure 6.3: Symmetry used to solve for one location on a quarter square

Figure 6.4: Iteration histories for 4 Pick-up locations for a square domain (coarse mesh)

Figure 6.5: Iteration histories for 4 Pick-up locations for a square domain (fine mesh)

Figure 6.6: Deformed shape at Optimal Locations

Table 6.2: Solution with different initial guesses and discretizations for $r=500$ and $l/t = 500$

Initial guess	numel	λ	β	x_1/l	x_2/l	f	$ g $
1.0,1.0	10	0.001	20.0	0.7453	0.7453	3.1330E-05	1.0821E-05
	12	0.001	20.0	0.7578	0.7578	3.2120E-05	5.4330E-06
1.0,0.7	10	0.001	20.0	0.7678	0.7757	3.2860E-05	1.8364E-05
	12	0.001	20.0	0.7666	0.7666	3.2333E-05	8.4385E-09
0.6,0.8	10	0.001	20.0	0.7614	0.7600	3.2794E-05	1.9623E-05
	12	0.001	20.0	0.7666	0.7663	3.2331E-05	3.6636E-07

6.2.2 Effect of γ and l/t ratio on optimal locations for a square domain

In order to study the effect of bending rigidity on optimal locations for a square domain, we systematically lowered the bending rigidity and solved for optimal locations for different l/t ratios. Different initial guesses and discretizations were used for each set of γ and l/t values. Table 6.2 shows typical results for the case of $\gamma = 500$ and $l/t = 500$ in which numel is the specified number of elements along the edge of the domain, f is the objective function, $|g|$ is the projected gradient and λ and β are the arc-length parameters. It is to be noted that the optimal locations converge at most up to 2 decimal places. This is due to inaccuracies in the computation of the objective function. The inaccuracies are due to discretization by finite elements and the computation of the strain energy by summing the areas under the load deflection curves of every node in the mesh. We select the solution with the lowest objective function and highest level of discretization. Results could be obtained only up to $\gamma=1000$ and $l/t=800$ beyond which divergence in the arc-length procedure occurred frequently.

Figures 6.7 and 6.8 show the variation of x and y coordinates of optimal locations

as γ and $1/t$ vary. The locations lie along the diagonal and as the bending rigidity is lowered (higher the value of γ), the locations tend to move towards the corner. When the size of the domain is increased (higher the $1/t$ ratio) the locations tend to move towards the corner.

Figure 6.7: Effect of γ and l/t ratio on x_1/l Figure 6.8: Effect of γ and l/t ratio on y_1/l

Figure 6.9: Effect of the aspect ratio on x_1/a and y_1/b for blue denim fabric

6.2.3 Effect of Aspect Ratio on Optimal Locations

The effect of aspect ratio on optimal locations was studied for blue denim fabric. Material properties for blue denim fabric from Chapter 4 were used and the optimal locations were obtained for different aspect ratios. Figure 6.9 shows the variation of x_1/a and y_1/b as the aspect ratio is varied. It can be noted that the optimal locations do not lie along the diagonal for aspect ratios other than 1.0. Also, the change in the location of y_1/b is more compared to that in the x_1/a .

6.2.4 Optimal Locations for Square/Rectangular Fabric Parts

Optimal locations for a square cotton twill fabric part and rectangular blue denim fabric part were obtained. Figures 6.10 and 6.11 show the deformed shapes when the parts are picked up at optimal locations. Dimensions and optimal locations are listed below :

Cotton Twill

$$\text{Dimensions} = 30\text{cm} \times 30\text{cm}$$

$$\gamma = 474.87$$

$$l/t = 499.16$$

$$\text{Optimal locations} : x_l = 11.16 \text{ cm}$$

$$y_l = 11.16 \text{ cm}$$

Blue Denim

$$\text{Dimensions} = 40\text{cm} \times 24\text{cm}$$

$$\text{Optimal locations} : x_l = 15.11 \text{ cm}$$

$$y_l = 6.79 \text{ cm}$$

Figure 6.10: Deformed shape of 30cm x 30cm cotton twill fabric part

Figure 6.11: Deformed shape of 40cm x 24cm blue denim fabric part

Table 6.3: Coordinates to which different initial guesses converge

Color	x_1	y_1	x_2	y_2	x_3	y_3	Objective Function
Red	-0.2980	-0.6309	-0.4316	0.5968	0.6742	0.0954	0.5903E-03
Blue	-0.1634	-0.6660	-0.5471	0.5008	0.6556	0.2230	0.6108E-03
Green	-0.0723	-0.6730	-0.5934	0.4324	0.6239	0.3023	0.6046E-03
Pink	-0.1777	-0.6656	-0.5311	0.5150	0.6525	0.2092	0.6107E-03

6.3 3 Pick-up Locations for Square domain

Three pick-up locations for a square domain of length $l=2.0\text{m}$ with material properties corresponding to $\gamma = 0.12$ were solved using the gradient based algorithm. Figure 6.12 shows the iteration histories plotted on the mesh. The coordinates to which different initial guesses converge are listed in Table 6.3. It can be seen that there exist multiple solutions with very small difference in final objective function values. We select the solution with the lowest objective function as the optimal solution. Figure 6.13 shows the deformed shape at optimal locations.

We attempted to solve the problem for materials with lower bending rigidity. During the solution process, we encountered frequent divergence in the arc-length procedures and this prevented us from obtaining results.

Figure 6.12: Iteration histories for 3 Pick-up locations for a square domain

Figure 6.13: Deformed shape at Optimal Locations

6.4 4 Pick-up Locations for a Circular Domain

The problem of four pick-up locations on a circular domain of radius $R = 1.0$ m with material properties corresponding to $\gamma = 0.015$ (computed using $l = 2R$ in Eqn. 6.1) was solved using the gradient based algorithm. Symmetry was used to solve for one location on the lower right quarter of the domain. Figure 6.14 shows the iteration histories plotted on the mesh. Solutions from different initial guesses converged to the same node in the mesh. The coordinates of the optimal locations with reference to Figure 6.14 are

$$x = 0.4629 \text{ and } y = -0.4736$$

with

$$r/R = 0.6622$$

where, r is the radial distance of the optimal locations. Figure 6.15 shows the deformed shape at optimal locations.

Circular Blue Denim Part

The problem was solved for a circular part of Blue denim material of radius $r=25.0$ cm. The corresponding nondimensional number $\gamma = 887.69$. Figure 6.16 shows the iteration histories plotted on the mesh. The optimal locations are at

$$x = 15.185 \text{ and } y = 15.285$$

with

$$r/R = 0.8618$$

where, r is the radial distance of the optimal locations. Note that the effect of flexibility is to move the locations towards the outer edge. Figure 6.17 shows the deformed shape at optimal locations.

Figure 6.14: Iteration histories for 4 Pick-up locations for a circular domain

Figure 6.15: Deformed shape at Optimal Locations

Figure 6.16: Iteration histories for 4 Pick-up locations for a circular blue denim part

Figure 6.17: Deformed shape at Optimal Locations

6.5 3 Pick-up Locations for a Circular Domain

The problem of 3 pick-up locations on a circular domain of radius $R = 1.0$ m with material properties corresponding to $\gamma = 0.015$ (computed using $l = 2R$ in Eqn. 6.1) was solved using the gradient based algorithm. Different initial guesses converged to solutions which differed by an angle of rotation about an axis perpendicular to the plane of the domain. Figure 6.18 shows the iteration history from one initial guess plotted on the mesh. The coordinates of the optimal locations with reference to Figure 6.18 are

$$x_1 = 0.5276; \quad y_1 = -0.3250; \quad r/R = 0.6197$$

$$x_2 = 0.0210; \quad y_2 = 0.6139; \quad r/R = 0.6143$$

$$x_3 = -0.5350; \quad y_3 = -0.2917; \quad r/R = 0.6094$$

where, r is the radial distance of the location.

We attempted to solve the problem for materials with lower bending rigidity. During the solution process, we encountered frequent divergence in the arc-length procedures and this prevented us from obtaining results.

Figure 6.18: Iteration histories for 3 Pick-up locations for a circular domain

Figure 6.19: Deformed shape at Optimal Locations

6.6 Irregular Domains

Examples of irregular domains were obtained from apparel and datawear products. Dimensions and shapes of typical shirt and shorts were obtained from the Apparel Lab, College of Textiles. Figure 6.20 shows actual shapes and dimensions of parts that are assembled to make a shirt and shorts. A datawear product was obtained from Medcovers, a company that manufactures equipment covers. Figure 6.25 shows a picture of Symbol 4340 hand held computer which is used for wireless access of information from a server. Covers for the equipment are made of laminated 1000 Denier Cordura material. Two components that are assembled in the making of the cover are shown in Figure 6.26. All dimensions in Figures 6.20 and 6.26 are in centimeters.

6.6.1 Shirt Front

Four pick-up locations for a blue denim shirt front were solved for using the gradient based algorithm. Iteration histories of different initial guesses are plotted on the mesh in Figure 6.21. Only the initial node and final node have been connected by lines of different colors. Table 6.4 lists the coordinates to which different initial guesses converge. Three of the four initial guesses converge to within an element. A more accurate solution can be obtained by using a refined mesh. This will take a lot of computational time because with more number of elements, the problem size increases. We note that there is small difference in the values of the final objective function for three of the four initial guesses. Hence, we select the solution that has the lowest objective function as the optimal solution. Figure 6.22 shows the deformed shape of the shirt part when it is picked up at optimal locations.

6.6.2 Short Front

Four pick-up locations for a blue denim short front were solved for using the gradient based algorithm. Iteration histories of different initial guesses are plotted on the mesh in Figure 6.23. Only the initial node and final node have been connected by lines of different colors. Table 6.5 lists the coordinates to which different initial guesses converge. The optimal pick-up locations are the those which have the lowest value of objective function. Figure 6.24 shows the deformed shape of the short part when it is picked up at optimal locations.

Figure 6.20: Shapes of Shirt and Short parts

Table 6.4: Coordinates to which different initial guesses converge for shirt front

Color	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	Objective Function
Red	4.858	4.900	6.984	25.583	68.726	14.786	65.598	2.295	28.094
Blue	7.226	2.418	4.716	23.091	41.821	19.306	73.633	6.928	22.114
Green	7.226	2.418	4.716	23.091	44.757	18.948	73.633	6.928	23.607
Orange	7.226	2.418	4.716	23.091	44.757	18.948	73.633	6.928	23.607

Table 6.5: Coordinates to which different initial guesses converge for short front

Color	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	Objective Function
Red	4.627	3.246	5.670	23.521	60.770	28.094	60.614	5.483	28.067
Blue	4.627	3.246	5.670	23.521	55.564	28.891	63.331	5.417	28.577
Purple	0.0	12.150	33.474	29.344	63.306	25.035	57.912	2.819	38.168
Green	4.627	3.246	5.670	23.521	60.770	28.094	60.716	2.806	28.537

Figure 6.21: Mesh with iteration histories for SHIRT part

Figure 6.22: Deformed shape of SHIRT part at Optimal Locations

Figure 6.23: Mesh with iteration histories for SHORT part

Figure 6.24: Deformed shape of SHORT part at Optimal Locations

Figure 6.25: Symbol 4340 hand held computer

Figure 6.26: Shapes of Two Parts

6.6.3 4079back

The 4079back part is symmetric about the x-axis as shown in Figure 6.26. Hence, when solving for 4 pick-up locations, it is sufficient to solve for 2 pick-up locations on the domain bounded by ABCDEF by enforcing symmetric boundary conditions along edge BC. Figure 6.27 shows the iteration histories of different initial guesses plotted on the mesh when 2 locations are solved on half the domain. Table 6.6 shows the coordinates to which different initial guesses converged along with the final objective function values. It can be noted that the solution for one of the location converges to within an element. However, there is a very small difference in the final objective function values. We pick the solution with the lowest objective function as the optimal solution.

6.6.4 4079frfc

The 4079frfc part is symmetric about the y-axis as shown in Figure 6.26. We obtained 4 and 6 pick-up locations by solving for 2 and 3 locations respectively on half the domain bounded by ABCDEFGH. Symmetric boundary conditions were enforced along edges CD, EF and GH. Figure 6.29 shows the iteration histories of different initial guesses plotted on the mesh for 2 pick-up locations. In this case all the initial guesses converged to the same nodes. The optimal locations are at

$$x_1 = 0.8793; \quad y_1 = 2.3885$$

$$x_2 = 1.5835; \quad y_2 = 26.567$$

Figure 6.31 shows the iteration histories of different initial guesses plotted on the mesh for 3 pick-up locations. Table 6.7 shows the coordinates to which different initial guesses converge along with the final objective function values. In this case these exist two local minima. We select the solution with the lower objective function as the optimal solution. Figure 6.32 shows the deformed shape at optimal locations.

Table 6.6: Coordinates to which different initial guesses converge for 4079back

Color	x_1	y_1	x_2	y_2	Objective Function
Red	3.541	2.526	27.569	2.503	0.68268
Green	3.541	2.526	27.569	2.503	0.68268
Blue	3.082	3.107	27.534	2.757	0.67715
Orange	3.541	2.526	27.569	2.503	0.68268

Table 6.7: Coordinates to which different initial guesses converge for 4079frfc

Color	x_1	y_1	x_2	y_2	x_3	y_3	Objective Function
Red	1.699	1.752	2.028	13.491	2.207	26.90	0.34860
Green	1.699	1.752	2.067	13.968	2.207	26.90	0.34625
Blue	1.30	1.783	5.347	13.695	1.209	26.518	0.42791
Purple	1.30	1.783	5.347	13.695	0.790	26.537	0.42831

Figure 6.27: Mesh with iteration histories for 4079back

Figure 6.28: Deformed shape of 4079back at Optimal Locations

Figure 6.29: Mesh with iteration histories for 4079frfc

Figure 6.30: Deformed shape of 4079frfc at Optimal Locations

Figure 6.31: Mesh with iteration histories for 4079frfc

Figure 6.32: Deformed shape of 4079frfc at Optimal Locations

Chapter 7

Conclusions

This research deals with the determination of optimal pick-up locations that minimize a measure of deformation of flat limp parts. In Chapter 1 of the dissertation, we reviewed existing work in the relevant areas of automated handling systems for limp materials, modeling and simulation related to limp material handling and determination of optimal locations. In Chapter 2, we gave an overview of the nonlinear shell finite element formulation used to model limp parts. The method of computing the objective function was also described. In Chapter 3, meshing issues were discussed and various algorithms developed to solve for optimal locations were described. In Chapter 4, the method used to measure material properties of fabrics was described. Material properties of 5 different fabrics were measured. In Chapter 5, results were presented for one dimensional domains or strips of limp material. Some results for fabric strips were also presented. In Chapter 6, results were presented for two dimensional domains. These consist of results for regular shapes like square/rectangular, circular and irregular shapes obtained from apparel and datawear applications. The following are the conclusions of this research :

- Gradient based optimization methods have been developed to solve for pick-up locations that minimize the deformation of flat limp parts. An essential difference when compared to existing methods for determination of optimal locations (reviewed in Chapter 1) is the computation of the objective function and gradient from a nonlinear finite element solution. The methods have been successfully applied to obtain results for one dimensional and two dimensional domains of different shapes.
- The ability to automatically generate a mesh with nodes located at specified coordinates in the domain of the limp part determines whether the problem can

be solved as a continuous or discrete optimization problem. Automatic meshing procedures were developed for one dimensional domains and square/rectangular domains with four pick-up locations. These problems were solved as continuous optimization problems. For arbitrarily shaped two dimensional domains, optimal locations are solved on a fixed mesh using a novel gradient based algorithm.

- The shell model for limp parts has limitations. Convergence failures in the solution process are encountered frequently when modeling thin materials of low bending rigidity thus stalling the optimization procedures. The results in this research have been limited by the convergence properties of the model.
- For one dimensional domains or strips of limp material, the optimal locations are symmetrically placed and as the flexibility of the limp part increases, the end locations move towards the ends of the domain. Also, the spacings between adjacent locations can be non uniform.
- Results for square and circular two dimensional domains indicate that more the flexibility, the closer the pick-up locations are to the boundary of the domain.

7.1 Scope for future work

- The results obtained were limited by the convergence properties of the shell model which is solved as a quasi-static problem. Divergence in the solution process occurs due to instability phenomena such as limit and bifurcation points. It is possible that by including dynamics in the model, instability phenomena may be eliminated by inertia. The resulting model can then be used to obtain results for materials with lower bending rigidities.
- A fixed mesh was used when solving for optimal locations on arbitrarily shaped domains. Incorporation of automatic meshing algorithms should be investigated. This will allow the solution as a continuous optimization problem and will be computational faster as smaller mesh sizes will be used.
- The optimization procedure is time consuming since a nonlinear problem is solved whenever the objective function is computed. It was found that around 80% of the time is spent for factoring the stiffness matrix. Incorporation of parallel solvers will reduce computational time and should be explored.

-
- Experiments using actual limp parts should be done to validate the computed optimal locations.

References

- [1] Taylor P.M., and Koudis S.G., 1987, "Automated Handling of Fabrics," *Sci. Prog., Oxf.*, 71, pp. 351-363.
- [2] Taylor P.M., Wilkinson G.E., Taylor G.E., Gunner M.B., and Palmer G.S., 1990, "Automated Fabric Handling Problems and Techniques," *Proceedings of IEEE International Conference in Systems Engineering*, pp. 367-370.
- [3] Taylor P.M., and Pollet D.M., 1997, "Why is automated garment manufacture so difficult ?," *ICAR 97*, pp. 39-44.
- [4] Parker J. K., Dubey R., Paul F. W., and Becker R. J., 1982, "Robotic Fabric Handling for Automating Garment Manufacturing," *ASME Journal of Engineering for Industry*, pp. 1-6.
- [5] Taylor P. M., Monkman G. J., and Taylor G.E., 1988, "Electrostatic Grippers for Fabric Handling," *Proceedings of IEEE International Conference on Robotics & Automation*, pp. 431-433.
- [6] Kolluru R., Valavanis K. P., Steward A., and Sonnier M. J., September 1995, "A Flat-Surface Robotic Gripper for Handling Limp Material," *IEEE Robotics & Automation Magazine*, pp. 19-26.
- [7] Karakerezis A., Ippolito M., Doulgeri Z., Rizz C., Cugini C., and Petridis V., 1994, "Robotic handling of Flat Non-Rigid Materials," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 937-946.
- [8] Torgerson E., and Paul F. W., February 1988, "Vision-Guided Robotic Fabric Manipulation for Apparel Manufacturing," *IEEE Control Systems Magazine*, pp. 14-20.

-
- [9] Czarnecki C., June 1995, "Automated Stripping: A Robotic Handling Cell for Garment Manufacture," *IEEE Robotics & Automation Magazine*, pp. 4-8.
 - [10] Fahantidis N., Paraschidis K., Petridis V., Doulgeri Z., Petrou G. and Hasapis G., March 1997, "Robot Handling of Flat Textile Materials," *IEEE Robotics & Automation Magazine*, pp. 34-41.
 - [11] Eischen J. W. and Kim Y. G., 1993, "Optimization of Fabric Manipulation during Pick/Place Operations," *International Journal of Clothing Science and Technology*, Vol. 5 No. 3/4, pp. 68-76.
 - [12] Eischen J. W., Deng S., and Clapp T. G., September 1996, "Finite-Element Modeling and Control of Flexible Fabric Parts," *IEEE Computer Graphics and Applications*, pp. 71-80.
 - [13] Cugini, U., Denti, P., and Rizzi, C., 1996, "Design and Simulation of Non-rigid Materials Handling Systems," *Mathematics and Computers in Simulation*, 41, pp. 587-593.
 - [14] Cugini, U., Denti, P., Ippolito, M., and Rizzi, C., March 1998, "Modeling and Simulation of Handling Machinery with Dynamic and Static Behavior of Non-rigid Materials," *IEEE Robotics & Automation Magazine*, pp. 48-56.
 - [15] Mroz, Z. and Rozvany, G. I. N., 1975, "Optimal Design of Structures with Variable Support Conditions," *Journal of Optimization Theory and Applications*, Vol. 15, No. 1, pp. 85-101.
 - [16] Prager, W. and Rozvany, G. I. N., 1975, "Plastic Design of Beams : Optimal Locations of Supports and Steps in Yield Moment," *International Journal of Mechanical Sciences*, Vol. 17, pp. 627-631.
 - [17] Rozvany, G. I. N. and Mroz, Z., 1977, "Column Design : Optimization of Support Conditions and Segmentation," *Journal of Structural Mechanics*, Vol. 5(3), pp. 279-290.
 - [18] Olhoff, N. and Taylor, J. E., 1978, "Designing Continuous Columns for Minimum Total Cost of Material and Interior Supports," *Journal of Structural Mechanics*, Vol. 6(4), pp. 367-382.

-
- [19] Narita Y., 1989, "Optimal Design of Support Location for Vibration of Structure and its Components," *Current Topics in Structural Mechanics, ASME PVP-179*, pp. 169-173.
- [20] Wang B. P. and Chen C. L., 1996, "Application of Genetic Algorithm for the Support Location Optimization of Beams," *Computers and Structures*, 58(4), pp. 797-800.
- [21] Xiang Y., Wang C. M., and Kitipornchai S., 1996, "Optimal Locations of Point Supports in Plates for Maximum Fundamental Frequency," *Structural Optimization*, 11, pp. 170-177.
- [22] Pitarresi J. M. and Kunz R. J., 1992, "A Simple Technique for the Rapid Estimation of the Optimal Support Locations for a Vibrating Plate," *ASME Journal of Vibration and Acoustics*, Vol. 114, pp. 112-118.
- [23] Roschke P. N., 1989, "Optimal Pick-up locations for Irregular Concrete Panels," *Microcomputers in Civil Engineering*, 4, pp. 267-273.
- [24] Simo, J. C. and Fox, D. D., 1989, "On a Stress Resultant Geometrically Exact Shell Model. Part I: Formulation and Optimal Parameterization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 72, pp. 267-304.
- [25] Simo, J. C., Fox, D. D. and Rifai, M. S., 1989, "On a Stress Resultant Geometrically Exact Shell Model. Part II: The Linear Theory; Computational Aspects," *Computer Methods in Applied Mechanics and Engineering*, Vol. 73, pp. 53-92.
- [26] Simo, J. C., Fox, D. D. and Rifai, M. S., 1990, "On a Stress Resultant Geometrically Exact Shell Model. Part III: Aspects of the Nonlinear Theory," *Computer Methods in Applied Mechanics and Engineering*, Vol. 79, pp. 21-70.
- [27] Schweizerhof K. H., Wriggers P., 1986. "Consistent Linearization for Path Following Methods in Nonlinear FE Analysis," *Computer Methods in Applied Mechanics and Engineering*, Vol. 59, pp. 261-279.
- [28] Dennis J. E. and Schnabel R. B., 1996, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, No. 16 in Classics in Applied Mathematics, SIAM, Philadelphia.

-
- [29] Kelley C. T., 1999, *Iterative Methods for Optimization*, No. 18 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1999.
- [30] Eischen J. W. and Bigliani R., 2000, "Continuum versus Particle Representations", Chapter in in *Cloth Modeling and Animation*, A. K. Peters.
- [31] Kim J. H., 1991, "Fabric Mechanics Analysis using Large Deformation Orthotropic Shell Theory," PhD Thesis, North Carolina State University.
- [32] Deng S., 1994, "Nonlinear Fabric Mechanics Including Material Nonlinearity, Contact, and an Adaptive Global Solution Algorithm," PhD Thesis, North Carolina State University.
- [33] Ramm, E., 1980, "Strategies for Tracing the Nonlinear Response Near Limit Points," *Europe-US Workshop on Nonlinear Finite Element Analysis in Structural Mechanics*, Editors : K. J. Bathe, Stein, E. and Wunderlich, W., pp. 63-89, Springer, Berlin.
- [34] Riks, E., 1979, "An Incremental Approach to the Solution of Snapping and Buckling Problems," *International Journal of Solids and Structures*, Vol. 15, pp 524-551.
- [35] Crisfield, M. A., 1981, "A Fast Incremental Iterative Solution Procedure that Handles Snap Through," *Computers & Structures*, Vol. 13, pp. 55-62.
- [36] Crisfield, M. A., 1983, "An Arc-length method including Line Searches and Accelerations," *International Journal for Numerical Methods in Engineering*, Vol. 19, pp. 1269-1289.
- [37] Sherman, J. and Morrison, W. J., 1950, "Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix," *Annals of Mathematical Statistics*, Vol. 21, pp. 124-127.
- [38] Cook, D. C., Malkus, D. S. and Plesha, M. E., 1989, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons.
- [39] Wagner, W. and Wriggers, P., 1988, "A simple method for the calculation of postcritical branches," *Engineering Computations*, Vol. 5, pp. 103-109.

-
- [40] Wriggers, P., Wagner, W. and Miehe, C., 1988, "A Quadratically Convergent Procedure for the Calculation of Stability Points in Finite Element Analysis," *Computer Methods in Applied Mechanics and Engineering*, 70, pp. 329-347.
- [41] Wriggers, P., and Simo, J. C., 1990, "A General Procedure for the Direct Computation of Turning and Bifurcation Points," *International Journal for Numerical Methods in Engineering*, 30, pp. 155-176.
- [42] Timoshenko, S., and Woinowsky-Krieger, S., 1959, *Theory of Plates and Shells*, McGraw-Hill Book Company.
- [43] Zhu, C., Byrd, R. H., Lu, P., Nocedal, J., December 1997. "Algorithm 778 : L-BFGS-B : Fortran Subroutines for Large-Scale Bound Constrained Optimization," *ACM Transactions on Mathematical Software*, 23(4), 550-560.
- [44] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. 1997. "A limited memory algorithm for bound constrained optimization," *SIAM Journal of Scientific Computing*, 16, pp. 1190-1208.
- [45] Byrd, Nocedal, J., and Schnabel, R. B., 1994. "Representation of quasi-Newton matrices and thier use in limited memory methods," *Math. Programming*, 63, pp. 129-156.
- [46] *The FAST System for the Objective Measurement of Fabric Properties : User's Manual*. CSIRO, Australia.