THREE DIMENSIONAL SIMULATION OF CLOTH DRAPE

by

ANNE BRICIS

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

of the Loughborough University of Technology

1995

DECLARATION

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained herein has been submitted to this or any other institution for a higher degree.

ANNE BRICIS

ACKNOWLEDGEMENTS

This research project has been jointly funded by the Engineering and Physical Sciences Research Council (EPSRC) and Coats Viyella plc.

My thanks go to: Prof Peter Jones, Department of Human Sciences, Loughborough University, for supplying the LASS body data; Dr Hugh Gong, Department of Textiles, UMIST, for supplying the material property measurements; Brian Negus for access to the Silicon Graphics and Jon Knight for sorting out various problems above and beyond the call of friendship.

I would like to thank my supervisors Dr Helmut Bez, Department of Computer Studies, and John Ascough, Department of Mechanical Engineering, for all their valuable help and advice. My thanks also go to Bob Moulsdale, Coats Viyella, for his keen interest in the research and his valuable time and comments.

Finally my thanks go to all my friends and colleagues who have helped in various ways during the past three years - in particular David and Ann, Jo and Ann and the rest of the Elvyn crew. Love to Dr Jackie Elliott, Rose and Linda - who supported from afar. Special love to Mum, Dad and Rose for their faith in me, and to Dr Andrew Hunt - who knew I'd get there eventually.

ABSTRACT

Research has been carried out in the study of cloth modelling over many decades. The more recent arrival of computers however has meant that the necessary complex calculations can be performed quicker and that visual display of the results is more realistic than for the earlier models.

Today's textile and garment designers are happy to use the latest two dimensional design and display technology to create designs and experiment with patterns and colours. The computer is seen as an additional tool that performs some of the more tedious jobs such as re-drawing, re-colouring and pattern sizing.

Designers have the ability and experience to visualise their ideas without the need for photo reality. However the real garment must be created when promoting these ideas to potential customers. Three dimensional computer visualisation of a garment can remove the need to create the garment until after the customer has placed an order.

As well as reducing costs in the fashion industry, realistic three dimensional cloth animation has benefits for the computer games and film industries.

This thesis describes the development of a realistic cloth drape model. The system uses the Finite Element Method for the draping equations and graphics routines to enhance the visual display. During the research the problem of collision detection and response involving dynamic models has been tackled and a unique collision detection method has been developed. This method has proved very accurate in the simulation of cloth drape over a body model and is also described in the thesis.

Three dimensional design and display are seen as the next logical steps to current two dimensional practices in the textiles industry. This thesis outlines current and previous cloth modelling studies carried out by other research groups. It goes on to provide a full description of the drape method that has been developed during this research period.

CONTENTS

1. INTRODUCTION 1.1 - 1.11

1.1 Computer Aided Design in the Textile Industry 1.1
1.2 3D CAD and the Textile Industry 1.5
1.3 Computer Simulation of Cloth Drape 1.8
1.4 Cloth Modelling Research 1.9
1.5 The Thesis 1.9

2. SURVEY OF CLOTH MODELLING 2.1 - 2.11

2.1 Introduction 2.1
2.2 Geometric Models 2.2
2.3 Physical Properties 2.3
2.4 Physical Models 2.4
2.4.1 Physical Modelling Using Finite Elements 2.4
2.4.2 Physical Modelling Using Other Methods 2.5
2.5 3D Display of Fabric Fit 2.7
2.6 3D Display of Fabric Drape 2.8
2.7 3D Design 2.9
2.8 Conclusion 2.10

PROJECT AIMS 3.1 - 3.11

3.1 Introduction 3.1
3.2 Advantages of 3D Cloth Display 3.3
3.3 Using the Finite Element Method 3.4
3.4 System Overview 3.5
3.4.1 Body Modelling 3.6
3.4.2 Cloth Mesh Generation 3.7
3.4.3 Draping Calculations 3.7
3.4.4 Display Routines 3.8
3.5 Project Stages 3.8
3.6 Current Project Status 3.10

4. BODY MODELLING 4.1 - 4.8

4.1 Introduction 4.1
4.2 Problems Associated with Body Modelling 4.3
4.3 The LASS System 4.3
4.4 From LASS Data to Draping System 4.5
4.4.1 Obtaining Full Body Model Data 4.5
4.4.2 Translating into Draping System Co-ordinates 4.7
4.4.3 Establishing the Waist Co-ordinates 4.8
4.5 Conclusion 4.8

5. CLOTH MESH GENERATION 5.1 - 5.14

5.1 Introduction 5.1
5.2 Mesh Generation Methods 5.1
5.2.1 Mesh Topology 5.2
5.2.2 Node Creation 5.2
5.2.3 Adapted Mesh Template 5.2
5.2.4 Geometry Decomposition 5.3
5.3 The Mesh Generation Program MSHGEN 5.3
5.3.1 Input Data 5.5
5.3.2 Establishing the Cloth Waist Nodes 5.5
5.3.3 Establishing the Remaining Cloth Mesh Nodes 5.8
5.3.4 Efficient Node Numbering 5.10
5.3.5 Adding the Elements 5.11
5.3.6 Output Data 5.13
5.4 Conclusion 5.13

6. DRAPE CALCULATIONS 6.1 - 6.31

6.1 Introduction 6.1
6.2 Drape Simulation Studies 6.1
6.3 The Draping Calculation Program DRAPE 6.3
6.3.1 Input Data 6.5
6.3.2 The Time Step Loop 6.5
6.3.3 System Matrices 6.9
6.3.3.1 Stiffness Matrix K 6.10
6.3.3.2 Mass Matrix M 6.25
6.3.3.3 Damping Matrix C 6.26
6.3.4 Applying the Load 6.27
6.3.5 Newmark's Method of Integration 6.28
6.3.6 Updating Nodal Co-ordinate Values 6.30
6.3.7 Output Data 6.30
6.4 Conclusion 6.31

7. COLLISION DETECTION AND RESPONSE 7.1 - 7.23

7.1 Introduction 7.1
7.2 Collision Detection 7.2
7.3 Development of a Collision Detection Method 7.4
7.3.1 Body Model Mapping 7.10
7.3.2 Cloth Model Mapping 7.12
7.3.2.1 Finding the Closest Body Slices 7.13
7.3.2.2 Finding the Four Closest Body Nodes 7.14
7.3.2.3 Calculating the Mapped Location 7.15
7.3.3 Checking for Element Collisions 7.16
7.4 Collision Response 7.18
7.5 Development of a Collision Response Method 7.19
7.5.1 Finding the Contact Point 7.19
7.5.2 Calculating the Time Interval for Step Rerun 7.21
7.6 Conclusion 7.22

8. DISPLAY SIMULATION 8.1 - 8.6

8.1 Introduction 8.1
8.2 Silicon Graphics 8.2
8.2.1 Mesh Description 8.2
8.2.2 Colour 8.3
8.2.3 Animation 8.3
8.2.4 Hidden Surface Removal 8.3
8.2.5 Lighting 8.4
8.3 The Display Simulation Program DISPLAY 8.4
8.3.1 Cloth Mesh and Body Model Data 8.5
8.3.2 Display Background 8.5
8.3.3 Displaying Each Frame 8.6
8.4 Conclusion 8.6

9. DISCUSSION AND CONCLUSIONS 9.1 - 9.20

9.1 Introduction 9.1
9.2 Results and Verification of the Modeller 9.2
9.3 Summary of the Modeller 9.17
9.4 Extensions to the Modeller 9.18
9.5 Conclusion 9.20

PUBLICATIONS RESULTING FROM THE RESEARCH P.1

REFERENCES R.1 - R.8

BIBLIOGRAPHY B.1 - B.2

APPENDIX I Body Modelling Translation Programs I.1 - I.5

APPENDIX II Mesh Generation Program II.1 - II.9

APPENDIX III Drape Calculation Program III.1 - III.24

APPENDIX IV Display Program IV.1 - IV.8

APPENDIX V Meetings and Conferences V.1 - V.2

1. INTRODUCTION

1.1 Computer Aided Design in the Textile Industry

Computer Aided Design (CAD) is an automated system that aims to reduce the time and costs involved with the design process of a product. These benefits are accomplished by using computers to perform the many tedious, repetitive and time consuming tasks traditionally carried out by hand.

The design process is comprised of four sub processes that overlap each other and can be performed many times during the design cycle. The first stage is the development of a specification in order to provide a description of the intended design. This is followed by the analysis stage where design principles are applied to the specification. Solutions are developed from this stage and an evaluation is made on their potential. The fourth stage is the presentation and appraisal of these solutions.

CAD aims to reduce the number of cycles in the design process of a product and to make each design stage easier and less tedious for the designer. This results in an increase in the designer's work output as feasible solutions are discovered earlier, and thus more design work can be achieved. This also means that the overall design time involved in a particular product can be reduced. CAD enables an accurate representation of a design and facilitates a clearer insight into complex problems arising from the design, thus providing for better decision making and the reduction of errors.

Computer Aided Design and Manufacture (CAD/CAM) provides powerful tools for today's markets. The benefits include faster responses to customer requests, increased quality of products, improved production flexibility and higher levels of customer service. If design and manufacture can become more integrated with the help of CAD/CAM systems, then turn around times will be shorter and profitability greater.

The fashion industry is a highly volatile one. Design and product development times must be kept short in order to remain competitive and to maintain or increase market share. The real bottleneck in the fashion industry is during the design stages. Once a design has been approved it can be in the high street store very quickly due to the advantages brought by CAM. It is the stages prior to garment approval that can take so long and this is where the use of CAD should be exploited.

Some CAD systems available commercially are for use in the design of fabrics, both woven and knitted. These packages can simulate actual warp and weft threads, knitted stitches, colour and yarn in two dimensions (2D). The representation is basically a more visual and realistic version than that produced by the traditional designers' tools of paper and pencils.

These computerised images can be shown directly to the customer. Alternatively they can be printed onto paper, transfers or even directly onto cloth. Some systems allow the output to be produced as colour separation on film or sent directly to a laser engraver. Knit designs can be interfaced directly to knitting machines and weave designs to looms. Pattern outlines can be connected to a plotter, automated cutter or laser cutter. Images can be transported electronically for use in desktop publishing and completed designs can be used in catalogues, brochures and publishing.

The use of electronic images is a fairly recent concept in marketing that is becoming more widespread. With the electronic presentation of a design idea, a team of buyers and designers can interact with the design system in order to make alterations to colours, styles and other details before a final decision is made. With the additional use of network communications the team do not necessarily have to be in the same place but can work from their own locations linked together by the system.

Human body simulation and animation is still a new area of research. Modelling the way in which clothes hang, drape and wrap on bodies is a very exciting concept. One of the main problems for cloth is matching descriptions, such as handle and drape that are difficult to define numerically, with actual properties that can be programmed.

The aim of this project has been to develop a three-dimensional (3D) computerised model that simulates the drape of cloth on a human body. The cloth model should be able to take into account the physical properties of various textile materials in order to display photo-realistic 3D images of garments. This removes the need to physically create either the garment or even the cloth.

Initial queries carried out at the beginning of this research seemed to show that the textile industry was not yet ready for this proposed 3D system and various sources displayed negative reactions. However since then interest in cloth modelling has increased and a 3D system should be easier to accept once 2D CAD systems are commonplace.

Most of the available CAD systems do not simulate the drape of an actual garment but allow colours, styles and shapes to be designed in 2D. Some systems have a texture mapping facility where a designed print or weave image is mapped onto an underlying 2D image. This image has been previously scanned into the system and the garment outline defined manually.

This texture mapping facility allows both the designer and the customer to visualise the intended garment idea. In order to improve realism, patterns and prints of the intended cloth are manipulated to fit around the folds and wrinkles of an underlying image that retains its original shading and highlighting.

The main problem with this 2D image is simply that it is only in two dimensions. The display cannot be rotated so that all views of the garment can be seen. Also this system gives only an idea of what the final image will look like. To give a realistic view of the intended garment, an image of that actual garment needs to be scanned into the computer, and this defeats the purpose of the system.

When CAD was first introduced into the textile industry it was very primitive. Screen resolution and quality of output were poor, options were limited and designers took a long time to accept the systems. CAD systems have been improving at great speed ever since and today's systems are much more sophisticated. The systems have reduced the overall time involved in the design of the final product and designers are able to react quickly to market changes in order to boost productivity and product ranges. When quality, flexibility and productivity are increased then market position is strengthened.

CAD meets the requirement of frequently changing fashion seasons where a short throughput time is essential. The systems stimulate experimentation and allow the designer the time to develop ideas at lower costs. Repetitive and time consuming tasks are automated, such as re-drawing of different colourways and repeats, and this allow the designer more time to concentrate on creative work.

CAD systems are intelligent and this allows mistakes in production to be viewed at an earlier stage in the design. What the designer sees on the screen can be transmitted electronically to the loom, knitting machine or printer. CAD offers the possibility of defining the exact styling and design requirements of a garment and this improves customer choice resulting in a much wider range of design ideas.

Designers are best at creating and adapting designs, colour and print ideas. These are functions of human creativity and inspiration and therefore cannot yet be replaced by a computer. However the design process incorporates much more than these functions between the initial concept and the final garment and it is these additional processes that are costly in terms of both time and money.

The industry sees time and cost as the worst enemies of design with punctuality and price being the main constraints that win and lose orders. CAD can help shorten design time and cut costs. This allows more opportunity not only to produce designs on time but to provide extra time to modify and enhance designs depending on customer requirements.

CAD is a tool. It does not aim to replace the textile designer but to relieve the burdens and constraints of traditional design and production methods. Traditionally textile designers have worked in two dimensions. A design would be developed, the fabric printed and the sample created. The aim is to show a visual image of the design to the customer. CAD systems started by developing tools to model this way of working.

Early CAD systems did not provide new creative tools for the designer, just imitations of their existing ones. This was partly due to a lack of available technology but also to a lack of imagination and communication on the part of both textile designers and CAD system designers.

Much of the design work within textile CAD systems consists of the re-colouring and modification of designs that have been previously scanned in. Gradually the software is improving and textile designers are becoming more confident in the use of these systems. New tools are being exploited and as the designers become aware of the potential advantages of CAD systems, new tools are being demanded.

1.2 3D CAD and the Textile Industry

CAD has developed through the evolution of computer graphics and computer aided drawing and drafting. Early CAD users and promoters were the military, aerospace, automobile and electronics sectors. Thirty years after the development of the first electronic digital computers, CAD was being established as a vital component within the expanding electronics industry in the design and manufacture of integrated circuits and printed circuit boards. Gradually improvements in hardware technology allowed CAD to become a feasible tool in other industries.

CAD has been widely applied more recently in industries where drawings are an important element for illustrating new scientific or design ideas. The aim of CAD is to apply computers to both the modelling and communication of designs. This has been achieved by automating or assisting in design tasks or by providing new techniques.

3D computer modelling has risen out of a need for methods of information processing related to the actual shape of the intended product. The physical prototype is the first 3D visualisation of a design idea and 3D computer modelling has been developed for this stage in the design process. 3D models enable the design in three dimensions of the intended product by providing complete visual information from any viewing angle. They reduce product costs and improve both product quality and reliability. The reduction in lead times also results in an overall productivity increase.

Early 3D models displayed the design as a wireframe model where 3D co-ordinates defined end points of lines in 3D space. This was later developed into 3D surface modelling to include the faces between the lines. These models were capable of calculating surface intersections and areas and some provided shaded images. Eventually solid modelling was developed. These models are capable of holding information about the design outline, surface, volume and mass.

Solid modelling techniques have been established over many years but soft object modelling is still a new area of research due to the increased complexity of the problem. Once a solid object has been defined, a single position in space can determine the location of that solid object. This is not the case for soft objects as the shape of a soft object changes in response to its surroundings.

Cloth objects were initially modelled as rigid surfaces. Gradually however computer graphics techniques have evolved from the display of rigid geometrical objects to display those that are more flexible by nature. As methods were developed to simulate flexible surfaces such as the human body, the problem of simulating cloth, in particular garments, has moved into focus.

Soft objects are characterised by flexible surfaces which are not easily defined geometrically. Techniques for modelling and animating flexible materials are needed in order to accurately represent a cloth garment while the effects of physical forces such as gravity must also be taken into account.

The existing application of standard computer graphics techniques in the textiles industry has concentrated the whole design process in 2D, from the design concept through to pattern generation and cut out. Textiles companies are already experiencing the benefits possible from the 2D CAD systems.

This leads to the question of the feasibility of 3D design tools. In 1987 Nisselson [NIS87] posed the question: can fashion benefit from computer graphics applications? Doubts had been raised that the potential cost benefits from these systems may not be enough incentive to use these techniques. The years since then have shown that there is a valid place in the fashion industry for computer graphics applications.

The current problem is that 3D design methods and tools are desperately under-researched and under-developed in the fashion industry. The questions being asked now are: how are garments to be designed in 3D and would new 3D methods be acceptable to the designers?

The main argument against 3D methods seems to be that the systems would require the designer to alter the way in which they design. However what seems to be forgotten is that many designers do actually work in 3D. Turning a design idea into a finished sample can be achieved in two ways. One is by working on a flat pattern (the 2D method) and the other by working directly onto a mannequin or live model (the 3D method). These methods are, of course, not exclusive and many designers incorporate features of both in their work.

Working with the flat pattern starts by adapting a basic pattern block for the intended garment. The block is a foundation pattern constructed to fit an average figure. The designer uses this block as a basis to develop the design idea. Style lines, gathers and pleats may be added but the basic fit of the pattern conforms to the block used. On the other hand working in three dimensions is achieved by draping, cutting and pinning fabric pieces onto a mannequin to obtain the necessary pattern pieces. The designer may work with the intended fabric or, if this is too costly, a cheaper substitute.

Current 2D CAD systems mimic the flat pattern method of design. The design ideas as well as the basic blocks can be saved electronically and many ideas can be developed in a shorter period of time. Once the design is complete the CAD system develops the resulting pattern pieces, including grading of sizes, and determines the optimum layout of the pieces on the fabric. The instructions can be sent directly to a laser cutter, so the patterns need not be printed onto paper at any stage.

The aim of 3D display is to improve on the visual images currently supplied by the 2D texture mapping systems. If pattern pieces can be "sewn" together and viewed on screen, there is less need to develop a full sample. An extension of 3D display is 3D design, which mimics the method of developing ideas directly onto a mannequin. Of course designers have their preferred method of working, but the ideal CAD system would enable both methods of design.

The main concern currently restraining 3D methods is a technological one. 3D design can only be approached successfully once real-time interaction has been achieved for soft modelling. Current hardware technology and software techniques do not allow for realistic, real-time 3D CAD for cloth drape. Some research groups have looked into 3D design for fashion garments, but the research described in this thesis concentrates on the area of 3D display. The model developed during the research period shows cloth draping over a body model while folds are generated due to the application of gravity forces. Even without the additional problems of 3D design, real-time modelling has not been achieved.

The application of 3D garment CAD will change the traditional garment design methods in some way. Designers have already accepted the new 2D CAD tools and are beginning to exploit many features that were initially seen as difficult to use. It is important that any system developed for 3D CAD is not only convenient but also acceptable to most garment designers.

The tools should be introduced at the learning stage in the design colleges so that new designers become familiar with the technology and are able to take advantage of the features these systems provide. The current lack of CAD systems in the design colleges is not due to lack of interest but to lack of money.

1.3 Computer Simulation of Cloth Drape

The definition of simulation for this research has been adopted from Mealing [MEA92] who defines simulation as a special case of computer animation. The aim of simulation is to model an occurrence dynamically using physical laws. Dynamics is defined to be the branch of mechanics concerned with the way in which masses move under the influence of forces.

A simulation is seen as a scientific experiment that is solved incrementally, where each successive frame is dependent on the calculation of the previous frame. As such it is an empirical method where the circumstances must be evaluated in order to determine the result at each stage.

The drape of cloth defines the folds that occur naturally when cloth is subjected to forces such as gravity. Cloth is seen as a soft object and falls into the general area of deformable models. These models are more difficult to represent than solid objects because they can be continually deformed into many different complex shapes.

1.4 Cloth Modelling Research

This thesis describes the cloth modelling research work carried out at Loughborough University of Technology. The research is a joint project between the Departments of Computer Studies and Mechanical Engineering with funding from the Engineering and Physical Sciences Research Council (EPSRC) and Coats Viyella plc.

The aim of the project has been to display in 3D the folds that occur naturally as cloth drapes over a body model due to the applied force of gravity. The cloth model incorporates physical property information taken from cloth samples in order to incorporate actual values within the computed display. This thesis covers the work carried out over three years in order to obtain the current cloth modelling system.

The research can be divided into three areas: the mesh generation, the draping calculations and the visual display. The most important part of the research has been to develop the draping calculations. This includes not only the actual equations involved in applying forces but tackles the problems associated with collision detection and response during interaction with an underlying solid body model. The mesh generation and display have been essential but secondary goals to the main draping equations.

1.5 The Thesis

Studies have been carried out into textile modelling over many years, in several countries and approached through several industries. The possibility of drape simulation has only become visual due to the latest technological advancements in computer graphics hardware and through the availability of the increased processing power required for this type of problem. Chapter 2 provides an extensive survey of this research and looks at the types of cloth models that have been developed.

While Chapter 2 shows that the 3D display of textiles is a rapidly growing research area, Chapter 3 explains some of the advantages of 3D display and defines the reasons for this particular research project. The project aims are introduced and an overview is given of the draping system that has been developed.

The human body model used in the draping system is not part of this research project. The body data has been obtained from Loughborough University's LASS system and incorporated into the main system. Information on how this has been achieved is provided in Chapter 4.

The three main programs in this system are the mesh generator, the draping equations and the display. These are covered in detail in Chapters 5 to 8 and an evaluation of each program is provided.

Chapter 5 covers the mesh generator. This is the pre-processor to the draping calculations and is required in order to describe the correct geometry and topology of the cloth mesh.

The main part of this research has concentrated on the drape equations. Physical property data has been included in the calculations in order to represent different cloth types. During the draping sequence, interaction with solid objects must be detected so that realism is retained in the resulting display. Chapter 6 describes the algorithms that have been established to develop the drape calculations and Chapter 7 describes the collision detection and response method that has been developed for this research.

The display program is the post-processor of the results calculated by the drape program. The display program provides the environment for the simulation of the cloth drape with the body model and this is covered in Chapter 8.

The main part of the thesis concludes with Chapter 9. This chapter provides several visual examples generated by the draping system and includes a discussion of the work carried out and ideas for the extension and expansion of the system.

The remainder of the thesis contains the Reference and Bibliographical listings and the Appendices.

The work carried out over the research period has resulted in two papers that have been submitted for publication. These papers are listed under the Publications section.

The References and Bibliography sections detail all the papers and texts used during the research period.

The Appendices contain additional material to support the main body of the thesis. Appendices I - IV describe the system programs and provide program listings and Appendix V is a list of meetings held with people vital to the successful completion of the research.

2. SURVEY OF CLOTH MODELLING

2.1 Introduction

The arrival of CAD/CAM in any industry allows Quick Response (QR) and Just In Time (JIT) techniques to be realised. These techniques allow product developers to respond to customer demands much quicker and with less by-product waste than traditional techniques. The aims of these techniques are to produce only the required items, at the required time and in the required quantities.

These approaches involve a continuous commitment to the pursuit of excellence in all phases of manufacturing systems design and operations. By doing this a company's position can be maintained and expansion can be facilitated in a competitive market place. CAD/CAM has made a significant impact in industries such as engineering and electronics because its benefits include a consistently high quality output in a shorter period of time.

In an industry such as textiles where a large amount of manual effort is involved in the design and manufacture of a product, CAD/CAM can only bring advantages. Large production runs have been the traditional method for product development aimed primarily at the mass market. However this demand is now changing and CAD/CAM can facilitate individual requirements that were once impractical and uneconomical.

In the fashion industry CAM is already used extensively in the production of garments and cloth products. Looms and knitting machines, pattern cutting and pattern grading are just some of the processes that are now commonly controlled by computers. CAM automates the monotonous processes involved in the manufacture of cloth products and this reduces the overall production time. This reduction in production time thus enables the product to reach the customer more quickly.

Today's textile designers are happy to use the latest 2D CAD technology to create and experiment with styles, prints and pattern ideas without the costs involved in producing actual fabric and garment samples. Customers are gradually accepting a computer generated image in place of the real sample and will make decisions based on the CAD output. 2D CAD may already be acceptable to the industry but there is still a long way to go before photo-realistic 3D fabric representation is achieved.

Cloth modelling is of interest to several industries as well as the obvious ones of textiles and computer graphics. Other interested parties include the areas of animation and structural component manufacturing. Studies into the general area of deformable object modelling have been published in a wide range of technical journals.

Some of the models aim to represent cloth precisely while other applications aim only for a near representation. Depending on the level of accuracy and visual realism required, this can reduce the overall processing time. Modelling soft objects can be more complex than solid modelling because in the former many parameters interact with each other in a complicated way thus result in dramatic shape changes.

This chapter aims to summarise past and present research of cloth modelling in order to provide a complete reference listing.

2.2 Geometric Models

The geometric approach to modelling deformable objects describes the shape of the object entirely by continuous functions. These models are simple to program and are economical in computation time. Geometric models work well for a fast prediction of the deformed state of the object.

Many of the earlier visual models of soft objects were usually based on geometry. Barr [BAR84] used Jacobians to deform objects locally and globally. Wyvill *et al.* [WYV87] used blended groups of spheres to represent a soft object. Other methods developed loosely draped cloth objects modelled as rigid surfaces with textures mapped onto them.

More recently Dhande *et al.* [DHA93] have looked at defining the shape of draped fabric using intrinsic geometry parameters such as curvature and arc length. They include fabric physical properties such as bending, shearing and fabric weight and look at how the support geometry also affects the draped fabric surface.

The main advantage of geometric models is that they provide quick results. Unfortunately dynamic parameters are difficult to incorporate into these models and so they can behave unrealistically. For improved realism and if dynamic modelling is to be considered, then physical models incorporating the cloth's physical property information should be adopted.

2.3 Physical Properties

Research into the physical properties of fabric has been carried out over many years. Peirce [PEI37] carried out some of the earliest studies that introduced qualities enabling a general comparison to be made between fabrics based on their geometric form. He looked at the relationships between threads at thread crossings and derived a set of equations to define these relationships.

Later studies emphasised the importance of bending and shear properties in the characterisation of fabric drape and handle. Behre and colleagues [BEH61, DAH61, LIN61] analysed shear and buckling behaviour. Cusick [CUS61] and Treloar [TRE65] also looked at the influence of shear in cloth behaviour.

Grosberg and colleagues [GRO66a-e, GRO68] investigated load-extension, bending, buckling and shear properties of woven fabrics. Kawabata *et al.* [KAW73a-c] looked at tensile properties as well as shear deformation. Abbott *et al.* [ABB73] looked at elastic resistance to bending. De Jong and Postle [DE77a and DE77b] used optimal control theory to look at tension and bending.

Hearle *et al.* [HEA72 and HEA80] expressed a need to extend engineering analysis into the areas of discontinuity, buckling and soft elastic yielding under pressure. Shanahan *et al.* [SHA78] felt that non-linear elasticity, time and friction effects needed to be considered as well. Lloyd [LLO80a and LL080b] looked at complex fabric deformation, buckling and drape. Amirbayat [AMI86 and

AMI89] looked into bending and buckling deformations. Ghosh *et al.* [GHO90] looked at the relationship between fabric and yarn bending rigidity.

2.4 Physical Models

In contrast with geometric models, physical models define the state of the cloth using differential equations. The general approach of physical deformable models uses equilibrium equations of continuum mechanics. The starting position of the cloth, the loading parameters, the support conditions and the physical properties of these elasto-dynamic surfaces are assumed to be known and the deformation of the surface can be predicted using the solution of these equations.

Either a discretisation strategy, such as the finite element method, or an optimisation strategy, using a minimum energy principle, can be adopted in order to solve these equations. Although these methods are much more time consuming than the methods used in geometric modelling, the results are more realistic.

The earlier studies were interested in the mathematical definition of textiles. Mack and Taylor [MAC56] derived differential equations for fitting fabrics to surfaces of revolution and Olofsson [OLO64] looked into the geometric-mechanical interdependence of the interlaced threads in woven fabric.

Once computer graphics techniques had improved considerably, cloth researchers could apply these mathematical models in order to develop realistic visual models.

2.4.1 Physical Modelling Using Finite Elements

Behre and colleagues [BEH61, DAH61, LIN61] treated fabric as a thin plate and Grosberg and Swani [GRO66b and GRO66c] included non-linear bending behaviour to improve the analysis. Hearle [HEA72] and Shanahan *et al.* [SHA78] also considered textile fabric to be a 2D sheet material.

A series of papers [BUL849] describe work carried out within the Japanese Research Institute for Polymers and Textiles that looks at the relationship between sewn fabric and how it drapes. Imaoka *et al.* [IMA849a] describe a model consisting of hanging rods connected by bending springs at the top and bottom of a skirt model. Finite element mechanical properties correspond to those of the cloth and potential energy is defined. A searching method is applied to minimise this energy.

Another paper [IMA849b] uses triangular elements and drapes a skirt model over body shapes. The incorporated physical properties are altered for different types of cloth and realistic results are achieved. Okabe and Akami [OKA849] obtain a rough estimate of the eventual garment shape from pattern pieces which have been divided into triangles. Again realistic results are obtained for these wireframe models.

Collier *et al.* [COL91] derived a model for an accurate prediction of the draping coefficient of a cloth sample. They used a geometric non-linear finite element method coupled with physical properties in order to predict the draping behaviour of fabric treated as a shell membrane.

2.4.2 Physical Modelling Using Other Methods

Weil [WEI86] used a minimum energy principle to represent cloth and incorporated the cloth's physical properties. His work is based on the use of catenaries which are the natural curves formed when a thread is suspended at two points. His models of cloth material hanging in three dimensions contain folds and are more realistic than simpler texture mapped models. In a later discussion [NIS87] Weil recommended a model for real time manipulation of cloth whereby the cloth is defined by parameters such as stiffness, elasticity and weight.

Terzopoulos *et al.* [TER87] used elasticity theory to construct their differential equations. The model was later refined to use non-linear elasticity which improved its results [TER88]. The system they created behaves like a set of points linked together by non-linear springs that take into account internal forces and global behaviour. Their realistic examples show a shrink wrapping effect, a flag waving in the wind, and a rug falling over some solid objects.

Scanlon [SCA90] extended this modelling method to develop a mesh where the threads are represented as a series of masses connected by springs. Although he discussed the possibility of introducing parameters to represent the characteristics of different types of cloth, his models were intended for animation purposes only. However his displays of cloth draping over spheres and tables are very impressive.

Aono [AON90] used an enhanced wave equation for his physically based model which simulates the behaviour of cloth with given forces and boundary conditions. The models he develops of wrinkling handkerchiefs and sleeves display very realistic results.

Possibilities of increased automation in the apparel industry prompted the work of Brown *et al.* [BRO90] who looked at predicting large deflection bending in typical apparel fabrics during an automated process. A similar study was carried out by Clapp and Peng [CLA91] who simulated laying fabric onto a table.

Breen *et al.* [BRE92] based their model on the idea that the macroscopic behaviour of a complex material such as cloth is based on the interactions of the microscopic structure of its threads. They represented the cloth weave by a grid of connected nodes and subjected this grid to a time stepping process where each step has two stages. The first is where all inter-grid constraints are removed and each node is allowed to free fall. The second stage is to regroup these nodes by applying an energy minimisation process to pull the grid together in order to produce buckling and folding. Although this process was found to be computationally expensive, very realistic results were achieved. Visual examples show cloth draping over more complex solid objects like an armchair as well as the usual square and round tables.

Some of the dynamic models mentioned so far are based on the equations used in vibration and structural analysis to study deformation. These models subject the cloth to concentrated forces at selected points on the cloth. Ling *et al.* [LIN93] feel that a more realistic approach would be to approximate aerodynamic forces which act on cloth immersed in air flow. This is because air flow past a moving piece of cloth is deemed to be unsteady, non-linear and turbulent. This paper describes the construction of a distributed force model based on low speed incompressible aerodynamic wing theories that can be applied to cloth surfaces such as flags and curtains. The model uses classical elasticity theory and rigid body mechanics.

Kunii and Gotoda [KUN90] used a model that combines both physical and geometric approaches. The former is used to describe local structures and the latter for global structures. Their results show a realistic animated model of wrinkles forming in a sleeve.

Early work by Magnenat-Thalmann and Thalmann [MAG87 and MAG90] simulated the clothes of their synthetic actors as part of the body with no autonomous motion. Later work [MAG91a] extended the cloth surface model to polygonal regions where geometric parameters were introduced to determine the dimension and shape of the cloth and dynamic parameters to influence motion. Their later examples show a skirt billowing in the wind, taking into account the underlying body for detecting collisions, which is a vast improvement on their earlier rigid surface cloth models.

2.5 3D Display of Fabric Fit

In contrast to fabric drape, fabric fit is described as the cloth being everywhere in contact with the underlying solid surface. In the manufacture of automotive components, flat layers of continuous fibre composite are assembled then moulded into the desired shape. Robertson *et al.* [ROB81] describe a procedure for computing the arrangement of fibres of a flat woven cloth after it has been placed round a spherical surface without wrinkling. A later study [ROB84] extends their work to shaping the cloth over a cone with a rounded top.

When a fabric is draped over a mould surface and consolidated under heat and pressure into rigid components, its resulting integrity may be compromised if the fabric wrinkles or buckles when trying to conform to the underlying surface. Van West *et al.* [VAN90] developed a graphical simulation tool that describes the draping of bi-directional fabrics over arbitrary surfaces. The tool is used to select suitable draped configurations for specific surfaces. In a similar study Van der Weeën [VAN91] compared three algorithms for draping biaxially woven fabrics on arbitrarily curved surfaces.

Heisey and Haller [HEI88a] feel that a better understanding of how fabrics fit to 3D objects will lead to the improved automation of custom produced garment patterns. Their work involved fitting woven fabric to non-applicable, non-algebraic 3D surfaces using numerical analysis techniques.

2.6 3D Display of Fabric Drape

Imaoka *et al.* [IMA85] developed a CAD system to simulate the garment sample-making process. They started with digitised 2D pattern pieces and obtained a 3D approximation of the final garment. The shape is refined by constrained non-linear finite element methods using the physical properties of the intended cloth.

CIMTEX is a collaborative venture based at Leicester's De Montfort University looking into the automation of the garment manufacturing industry. One of the projects involves a 3D garment dressing system based on finite element methods. Fozzard and Rawling [FOZ91] describe the development of this dressing system that combines a drape modeller with solid body modelling techniques. Garment panels are created using a traditional 2D design system and are then converted to a mesh representation. The required fabric characteristics are extracted from an objective measurement database and the pieces are "sewn" together and displayed wrapped around the solid body [FOZ92]. The current system does not have a complex body form but is under development to eventually allow a mannequin with arms and legs to be dressed.

Magnenat-Thalmann and colleagues [MAG91b and YAN92] extended their early dynamic work by developing a tool for the garment industry to interactively design garment panels in 2D. Key points on the body are defined and cloth panels are created by passing through these points. The panels are seamed together in 3D to simulate dynamically the resulting shape that is texture mapped onto a moving body. They use the cloth draping and body model algorithms achieved in their early work and are combining these into a complete design package.

Liu [LIU93] has also looked at the conversion of 2D flat patterns into 3D representations. The problem was approached in two ways: empirically and analytically. In the first method a database of patterns, physical properties and measured data on 3D garment profiles was analysed and the results correlated to generate a 3D computer representation of a garment. The second method considers the bending behaviour of a fabric panel modelled as a simple wireframe. A bending curve program then predicts the shape of the garment hem.

Gray [GRA92 and GRA93] explores the idea of a virtual catwalk where 3D computer generated models display 3D computer generated clothes. The work achieved so far has been to model a catwalk and simulate a human body walking a few steps. Work into developing draped garments is continuing.

2.7 3D Design

Computer Design Inc. (CDI) have probably the only commercial 3D apparel design system available in today's market. Their 3D body mapped garments such as underwear and swimwear are very realistic. Their system allows interactive design, directly onto digitised 3D mannequins, that can be viewed from any angle. Fabrics and textures can be applied and when the design is complete, 2D pattern pieces can be generated.

Although garment fit appears to have been achieved commercially by CDI, fabric drape had, until recently, proved to be more difficult. Their Design Concept 3D system currently incorporates a solid modelling 3D drape facility that is a vast improvement on their earlier system. When the Company initially carried out work on soft modelling they achieved reasonable results, but discovered that at the time the American textile industry was not ready for the application. This rejection of 3D is changing however as more recent discussions with designers seem to show that the initial reluctance shown towards 3D design was due to the limitations of the systems rather than the actual idea. CDI have come a long way since their first attempts, however it is interesting to note that they cannot yet achieve real-time drape. Even some of the simplest applications require lengthy processing times.

Yang and Zhu [YAN91] describe a method of 3D computer aided garment design. They consider the 3D human body as the construction of a motion mechanism. Parts of the body which are logically relational are chained through linked list. A hand for example is linked to the respective elbow information which in turn is linked to shoulder information. To generate the garment wireframe a multiple linked list structure is built. This wireframe can be modified to obtain different styles and structures.

Hinds and McCartney [HIN90 and MCC92] have also attempted the problem of designing in 3D for draped garments, rather than designing in 2D and mapping to 3D for the display as with most of the models mentioned earlier. In their work a garment is considered as an assembly of 3D surface panels which can be represented by mathematical models. By designing in 3D, the process of defining the style patterns first would be eliminated as once the design has been completed, a 3D to 2D mapping can take place to develop the necessary pattern pieces [HIN91].

Heisey and Haller [HEI88b] have also looked at 3D to 2D mappings. They extended their earlier work to model the physical process of manually draping a garment. A functional relationship was developed to project a known 3D fabric surface onto a 2D surface. This was then demonstrated by producing garment patterns [HEI90].

2.8 Conclusion

Consumers' preferences change frequently and as a result many people want their clothes to be different to those of the next person. To satisfy these demands the garment industry must change its basis to the multi-kind, small lot production system. CAD/CAM is seen as the tool to be adopted to make this possible.

Current practices involving CAD/CAM ensure that garment design and manufacture is fast and geared towards the customers' preferences. Retailers use the output generated by the CAD system in order to make decisions on the final product. CAM ensures that the garment is created to a high quality standard within a reasonable time scale.

3D design is a future extension of current 2D CAD systems. Here garments can be custom-designed for an individual depending on body data previously entered into the system. The design can then be displayed on a simulation of the customer's body model so that the customer can preview how the garment looks. Changes can be made interactively with the display and when the customer is satisfied with the image, the real garment can be created according to pattern pieces developed by the system, again tailored to the customer's body data.

However there are several issues to be tackled before this vision of successful 3D representation can be accomplished. Realistic and reliable models must be developed so that the designer has confidence in the final 3D result and the customer sees a photo-realistic image, rather than a general idea. These models must be incorporated into a user-friendly system that aids the designer as a useful tool. Difficulties in obtaining reasonable results will mean that a system, however technically advanced, will not be used. Also the fully draped image must be developed in real-time and should be interactive. Finally the 3D design must be transformed accurately into 2D pattern pieces.

Fabric modelling is not a new area of research. Studies have been looking into cloth physical properties for many years and many mathematical models of textiles have been developed. The arrival of computers has meant that complex calculations can be performed quicker and the resulting models are more easily and realistically visual.

Gradually the visual realism of the displays is improving. While yesterday's 2D design of fabric seems to be the latest technology, tomorrow could see the arrival of 3D as a useful and convenient design and display tool. With the improvement of hardware technology and software modelling, resulting in reduced processing time and realistic graphics, the vision described above of 3D garment design will not be too far in the future.

This research addresses the problem of 3D display. A realistic mathematical model has been developed that displays the dynamic drape of cloth around a static human body model. Physical methods have been used, together with physical property data taken from cloth samples, for the development of the draping system.

3. PROJECT AIMS

3.1 Introduction

3D CAD involves changing the way in which a designer has traditionally worked and this may be viewed as too radical and even an unnecessary step to take. When 2D CAD was first introduced, there was a lot of resistance from the industry that had to be overcome. The resistance came mainly from the customers, the actual CAD output receivers. The customers were not concerned with the methods used to create the design, they were interested only in the garment sample. This meant that early 2D CAD output was not realistic enough to be convincing.

Gradually however, the customers began to appreciate how CAD was helping with the design process. Alterations could be made more quickly and customer input could be incorporated immediately. As they began to trust these systems, the customers started to appreciate the benefits of 2D CAD and eventually accepted the 2D output. 2D CAD is now establishing a firm foothold in the industry and in many cases has become an essential, if not indispensable, tool.

The most dramatic change has already occurred and this was the actual introduction of computers into the design office. Many of the initial 2D CAD features were just imitations of traditional design tools, such as re-colouring drawings or the repetition of patterns. These tools helped to speed up the existing processes carried out by the designer. New features were introduced to provide additional tools within the 2D capacity, such as the texture mapping of a pattern over an underlying 2D image and the simulation of knitting stitches.

The techniques brought by 2D CAD are now being fully exploited and the next step is 3D. The physical 3D part of the design process is the generation of the garment sample and this is the area being tackled in this research.

This thesis is about the development of a tool that demonstrates the potential of 3D methods in computer aided garment design. This visualisation tool, incorporating drape simulation, is just one part of the huge problem of 3D CAD in the textiles industry.

The visualisation of a design is the important link between the designer and the customer. Traditionally this visualisation has been in the form of the garment sample. The arrival of 2D CAD meant realistic images could be achieved with the texture mapping facility. Many customers are now making decisions based on these 2D images.

The 3D visualisation tool described in this thesis has been developed to evaluate and compare designs that may or may not have been generated through 3D methods. The visual display aims to be more realistic than the current 2D texture mapped images because it can be viewed from any angle. Also the representation will be a closer image because the actual garment data, including pattern pieces and cloth type, will be taken into account.

The 3D modeller has been developed using finite element methods. One important issue tackled by this thesis is collision detection involving soft objects. Collision detection and response are important considerations for 3D CAD in garment design as clothes are designed around an underlying body. Soft objects can be more complex to model than solid ones and this increases the complexity of the collision detection problem as well.

A new approach to collision detection has been developed. This approach reduces the complexity of the collision check by mapping the irregular shaped body onto a regular shaped cylinder. The technique is currently applied during interaction of the dynamic cloth model with the static solid human model but has wider applications in many other collision detection situations.

The previous chapter outlined many cloth modelling studies which have been carried out and shows that cloth modelling is an active research topic. The initial aim of this project was to apply an existing program, adopted from structural analysis techniques, in order to develop draping algorithms for cloth while incorporating measured physical properties. This has been achieved and two minor goals, in the form of a mesh generator and a display simulator, have also been completed.

After outlining some of the advantages of 3D simulation, this chapter describes the initial plans determined at the start of the project and the methods that have been adopted during the research period. It goes on to provide an overview of the system programs that have been developed. A description of the achieved project milestones and the current status of the system concludes the chapter.

3.2 Advantages of 3D Cloth Display

This research is concerned mainly with the textiles industry, in particular the area of 3D garment display. 3D cloth modelling however is of interest to several other industries. In the area of computer games and film, for example, increased realism for human models is vital to the success of visual effects. Better cloth modelling will improve the overall 3D impact of these human body models and this can only be beneficial to the entertainment industries.

3D display can be used to visualise the layout of textiles over moulds in the area of component manufacture. When a textile is set over a mould the resulting integrity of the component could be compromised if buckling or stretching occurs. If 3D display can be used to discover the optimum layout for the textile, then less wastage occurs through defective components.

Combined with personal body model information, garments can be designed to suit each individual customer. If the individual likes a particular outfit then the 3D garment image can be dressed onto the individual's body model data. The overall look of the garment can be seen realistically as folds occur during the simulation. Modifications can be made to the garment in relation to fit, style and colours and the garment can be made up once all these virtual alterations have been completed.

This is obviously an idea for the future but one main reason for this research is to reduce the costs involved in the garment sampling process. Design ideas can be visualised using the current 2D texture mapping facilities on many of today's textile CAD systems.

However there are limitations to the suitability of the scanned-in underlying image. The advantage of 3D display would be that the required design information obtained from pattern pieces with the required print and colour data could be displayed. In other words 3D display shows a realistic version of the intended garment, not just a general idea, and the view can be rotated so that all angles of the image can be seen. An extension of this is dynamic display where a simulation of the intended garment in a real life environment will greatly enhance the image.

3.3 Using the Finite Element Method

The main goal at the start of this research project was to develop a dynamic draping cloth model. A dynamic problem is by definition time varying. There are two fundamental differences between static and dynamic problems. The first is that the dynamic problem does not have a single solution. A succession of solutions must be established which correspond to all periods of interest. As a result, the dynamic analysis is more complex and time consuming.

The second difference is that in the static problem, the internal moments and forces and the deflected shape depend directly on the given load. These can be computed directly from the load by established principles of force equilibrium. In the dynamic problem however the displacements are associated with accelerations which in turn produce inertia forces resisting the accelerations. These inertia forces must also be included in the analysis.

The inertial forces result from structural displacements which in turn are influenced by the magnitudes of the inertial forces. This closed loop of cause and effect can be analysed directly in terms of differential equations.

By concentrating the mass of a structure at a series of discrete points, inertia forces, displacements and accelerations can also be developed exclusively at these discrete points. This simplifies the whole analysis and is why the finite element method has been adopted in this project.

The finite element method is commonly used in the numerical modelling of physical phenomena and has been used to model cloth in many research projects. This method approximates a set of partial differential equations which govern the behaviour of a system by a set of algebraic equations.

It is a systematic procedure whereby a structure is approximated by a discrete model. First the nodes and elements are constructed and then local approximation over each element is carried out. This is achieved by interpolating the variable between the nodal values using continuous functions called "shape functions". Finite elements can be considered separately for the purpose of local approximation which is independent of all the other elements and their respective approximations. The elements become interdependent when each element is assembled into the complete system.

Routines from the Numerical Analysis Group (NAG) Library have been used for the draping calculations. This Library is in two levels: one level contains a suite of routines which perform standard numerical operations and the other contains example programs which can be modified to meet particular requirements. These routines and programs are written in standard FORTRAN 77 [GRE82].

A requirement of real-time simulation was considered at the start of the research. However it was soon realised that the complexity of the cloth mesh and hence the extensive amount of computation time meant that real-time simulation was not possible due to the limitations of the existing hardware and software.

3.4 System Overview

The draping system is comprised of two FORTRAN programs, MSHGEN and DRAPE, and a C program DISPLAY.

MSHGEN defines the cloth mesh geometry and topology and provides the necessary cloth data. Some of the necessary physical property values have been taken from real cloth samples measured by Manchester University's Textile Department. The remaining properties are assumed to be constant to all cloth types.

DRAPE calculates the mesh displacement due to the application of a gravity force over a series of time steps using the NAG Library routines. Any interacting solids are assessed for potential collisions and if detected, collision response is performed.

DISPLAY uses the co-ordinates provided by DRAPE to show the simulation using the Silicon Graphics Library routines.

The body model data has been provided by the Loughborough Anthropometric Shadow Scanner (LASS). These data files need to be translated from the raw body model data into the draping system data and small programs have been written to accomplish this stage. The relationship between all these programs is show in Figure 3.1.

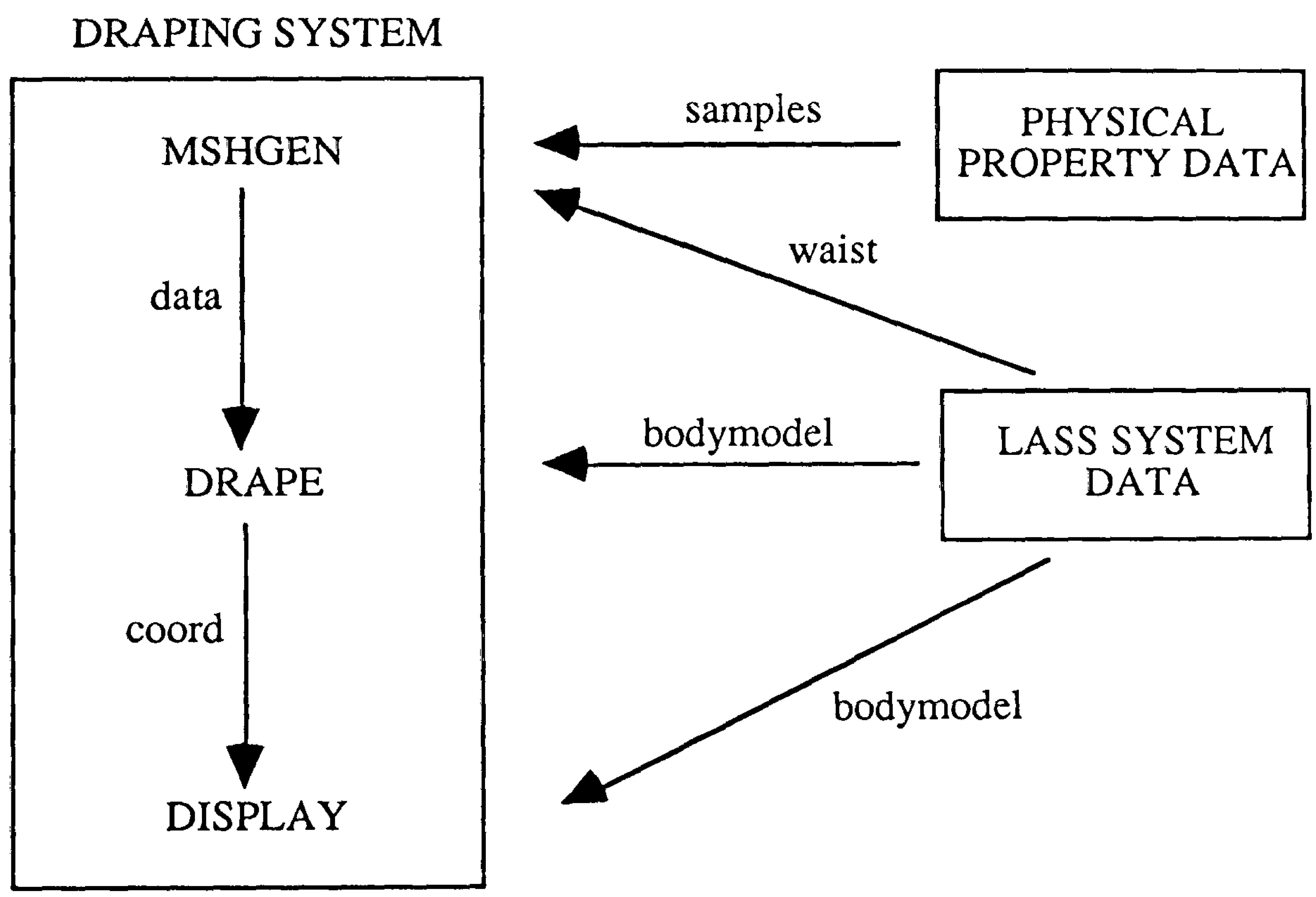


Figure 3.1 - system overview

3.4.1 Body Modelling

One of the main reasons for the research has been to display garments on an actual body model. The initial stages of the drape modeller showed cloth falling over a simple object such as a cylinder.

When the model had been made more reliable then a static body model was introduced to show the potential for the system. The body model data is used as a basis in the generation of the cloth mesh, within the draping program for collision checks and for the final display.

3.4.2 Cloth Mesh Generation

The information required by the draping calculations is stored in a sequential data file which lists the nodal co-ordinates, the element connections and some of the necessary physical properties.

At the beginning of the research when initial tests on string data were carried out, the input data file was concise and could be adopted straight from one of the NAG Library data files. As the meshes have increased in complexity it has been necessary to develop a mesh generator. This has been written in standard FORTRAN 77.

3.4.3 Draping Calculations

The NAG Library program **seg2p2e.f** has been developed to suit the project's needs. This program calculates the forced vibration of an elastic solid using direct integration by Newmark's method. This skeleton program has been radically modified for the drape system.

Modelling began with simple strings to ensure that the basic equations were stable and then progressed to small cloth meshes and eventually to larger structures. Interaction with underlying solid objects was introduced starting with regular shapes and eventually using a human body model.

Some cloth examples have been measured for various physical properties. Once the general cloth model had been developed to a reliable state these properties were incorporated to demonstrate the movement of different types of cloth.

The four cloth samples chosen were 100% wool, 100% cotton, 100% polyester and 50% polyester/50% cotton.

3.4.4 Display Routines

At the beginning of the research when simple strings were being used the results could be displayed using any graphical display software. The Unigraph software on the Hewlett Packard was used for the initial examples.

As the cloth meshes increased in complexity a better graphical display was required and a Silicon Graphics workstation was chosen to display the drape model. This enabled realistic 3D images to be developed for the dynamic simulation.

The Silicon Graphics Library routines are capable of handling 3D information and provide lighting and shading algorithms as well as real-time animation. The routines have been grouped together in a C program to display the results from the draping calculations.

3.5 Project Stages

The initial tests for the model started on single strings as shown in Figure 3.2. As the cloth model became more stable the test data increased in complexity. At first this covered small cloth patches but eventually the meshes extended to large areas of cloth.

Figure 3.2 shows a string restrained at one end with a load being applied at the other. The string is represented by a sequence of simple bar elements which are joined together to form the string. This one-dimensional problem was further tested by applying loads along the whole length of the string.

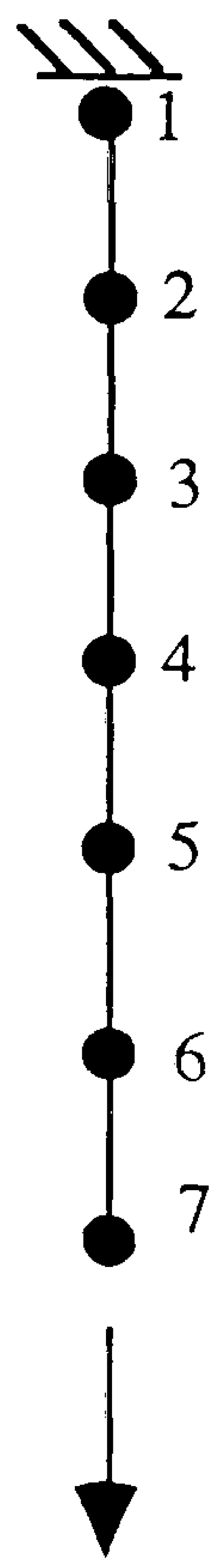


Figure 3.2 - one-dimensional string

To extend the example shown in Figure 3.2 into a 2D problem the string was restrained at each end and the load applied at the centre node as shown in Figure 3.3.

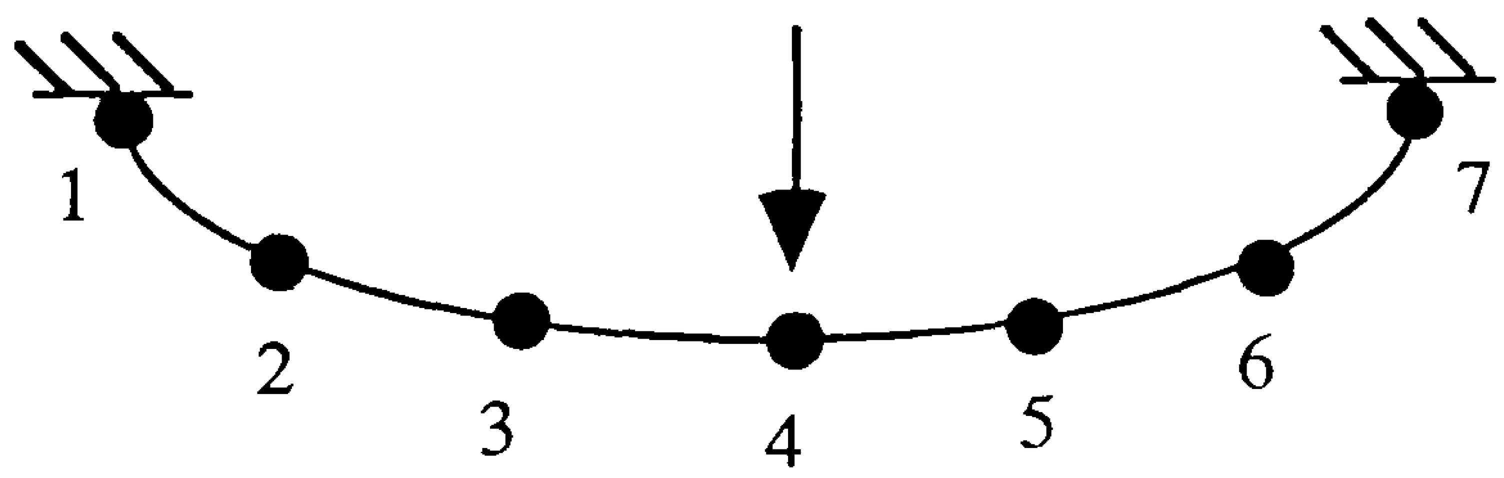


Figure 3.3 - two-dimensional problem

Again this was further developed by applying the load along the whole length of the string.

To extend these single string models into a 3D model, two strings were joined together and restrained at all four ends. Loads were then applied along their length. By adding to the number of strings, larger meshes were created.

When these models had been fully tested and reasonable results obtained then more complex meshes were developed where the model came into contact with simple solids. Eventually the cloth mesh was draped around a static body model.

3.6 Current Project Status

The mesh generator program develops a mesh to represent a circular skirt. This uses waist data taken from the body model. The display program uses the results of the draping calculations to display the simulation at each step. The mesh generator and display programs are necessary pre- and post-processors to the draping calculations.

The most important parts of the project are the draping calculations, combined with the measured physical property information, and the collision detection routines. The draping model has been approached in a new way to previous research, applying existing software but to a unique problem and this has proved challenging.

For this research, cloth samples were measured and some of these measurements have been included in the draping calculations. The measurements provided values for several properties of the cloth samples defining the strength, bending, surface, compression, thickness and mass. However not all of these values are necessary for the finite element analysis and hence only the values for bending, mass and thickness have been used.

The area of collision detection and response brought additional concerns to the research problem. When simple solids such as a cylinder were initially used, collision detection was a check on whether the cloth had penetrated the volume of the solid. In the case of the cylinder this could be achieved by comparing the node distance from the centre of the cylinder with the radius of the cylinder. If the distance of the node was less, then a collision had occurred.

However when the body model was introduced to the system, the whole process of collision detection had to be revised. This is because the body model is not a regular shape that can be described in a few equations. The cylinder, for example, needs only the height and radius to be defined for collision detection purposes.

A unique method for collision detection between complex objects has been developed during this research. The method currently solves collision detection between a solid object and a soft object but could be applied to any situation where collision detection needs to be considered.

4. BODY MODELLING

4.1 Introduction

Modelling the interaction of cloth with underlying solid objects is a very important part of cloth drape simulation. At the start of this research only simple solids such as boxes and spheres were considered because they were very easy to define geometrically.

Problems with modelling start to arise as the complexity of the solid objects increases. This is because these objects cannot be easily described in terms of circles or boxes as they have many protruding edges and uneven sides. This is the problem encountered with modelling human bodies.

Current body modelling research has become very realistic. Gray [GRA93] has developed a computerised catwalk along which strides a computerised model. The wireframe body has been created using existing software, the limbs are moved and the whole sequence is rendered to add realism. This model is then placed into a 3D catwalk scenery. Current plans involve increasing the realism and scope of the model and adding clothes.

Other interesting body model work has been carried out by Monheit and Badler [MON91] who have improved the realism of their models by defining spine attributes obtained from medical data. In their model the spine moves as a series of vertebrae connected by dependent joints. Previous models tended to have stiff backs which bent only from the waist.

Magnenat-Thalmann and Thalmann [MAG91a] developed a classification of human body modelling based on motion control methods and types of body interactions. Their body model was defined as a skeleton consisting of a connected set of segments and joints. Animation has been achieved through key framing [MAG87]. Initially garments were modelled as part of the body model but later they were modelled with their own properties. These models also involved interaction with body models [YAN92].

The research described in this thesis is concerned primarily with the drape of garments over bodies. It is therefore necessary to involve interaction with a body model during the simulation. The project uses a model based on real body data to enhance the realism of the display. Realistic visualisation is very important when showing the draping modeller to garment designers and other interested parties in the fashion industry.

The body modelling data used in this research has been provided by the Loughborough Anthropometric Shadow Scanner (LASS) and an example is given in Figure 4.1.

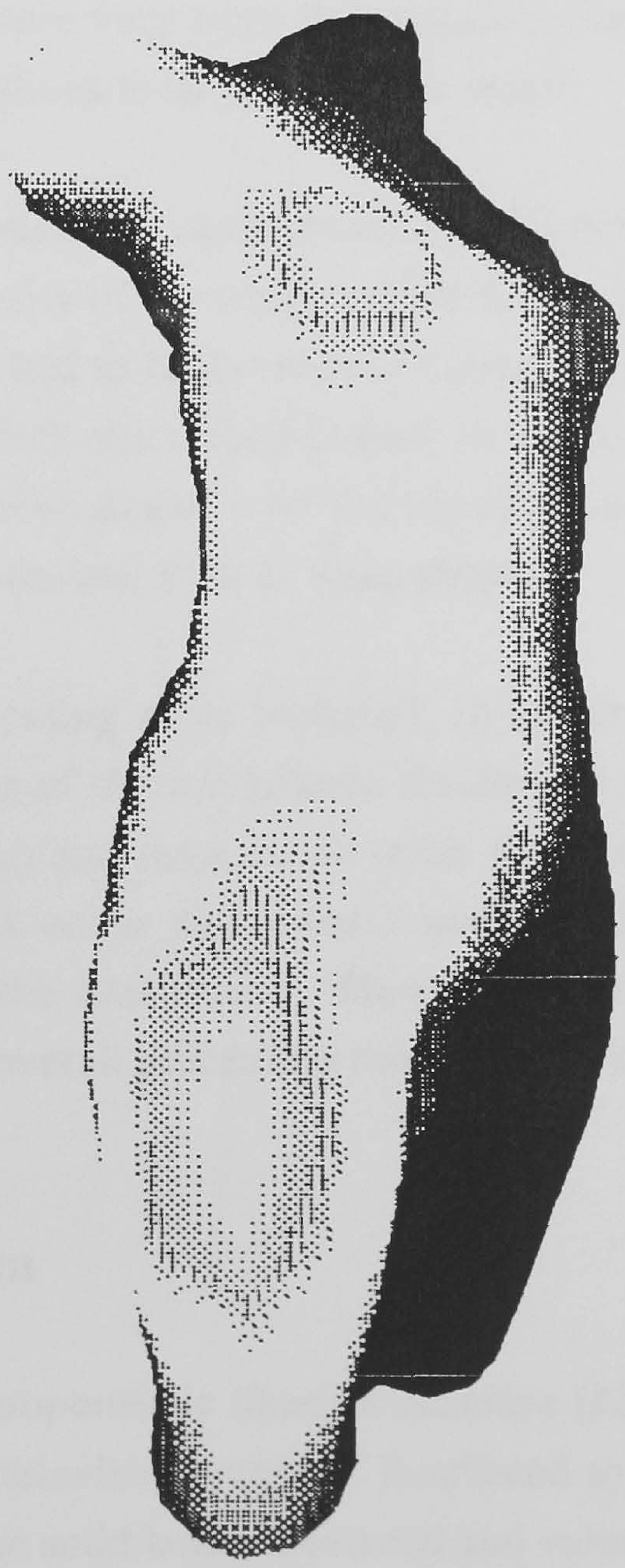


Figure 4.1 - LASS body model example

This data has enabled a static 3D body model to be used within the simulation. An extension of this research would be to incorporate a dynamic body model but currently animation occurs only within the cloth itself during interaction with and around the static body model. This chapter describes the process of converting the LASS body model data into the draping system co-ordinates.

4.2 Problems Associated with Body Modelling

In the early stages of this research, when simple strings were being modelled, interaction with other objects was not considered. When the problem became three dimensional, the cloth was draped over a flat topped cylinder. Simple solid shapes such as cylinders are very easy to program geometrically because they need only one or two equations to determine their shape.

Towards the end of the research a model based on real body data was introduced. This added to the complexity of the whole problem because it was not a uniform shape. More processing had to be carried out during the generation of the mesh so that the waist of the cloth skirt fitted closely to the waist of the body model. Previously the cloth had been draped over the top of the solid and it had not been necessary to consider dimensions such as waist shape.

There was no extra processing to be included in the draping calculations, but additional post processing of the co-ordinate results had to be carried out in the form of collision detection and response in order to maintain the realism of the simulation. If collisions occur and remain undetected, then this drastically reduces the realism of the animation. However because collisions must be detected, this adds to the overall processing time of the system.

4.3 The LASS System

The Loughborough Anthropometric Shadow Scanner [JON89] is a non-contact measuring system using television cameras interfaced to a computer. Strips of light are projected onto the solid being measured and values are obtained in terms of radii and angles in conjunction with the height. This data can then be turned into slices of x, y and z co-ordinates. An example of these data slices is shown in Figure 4.2.

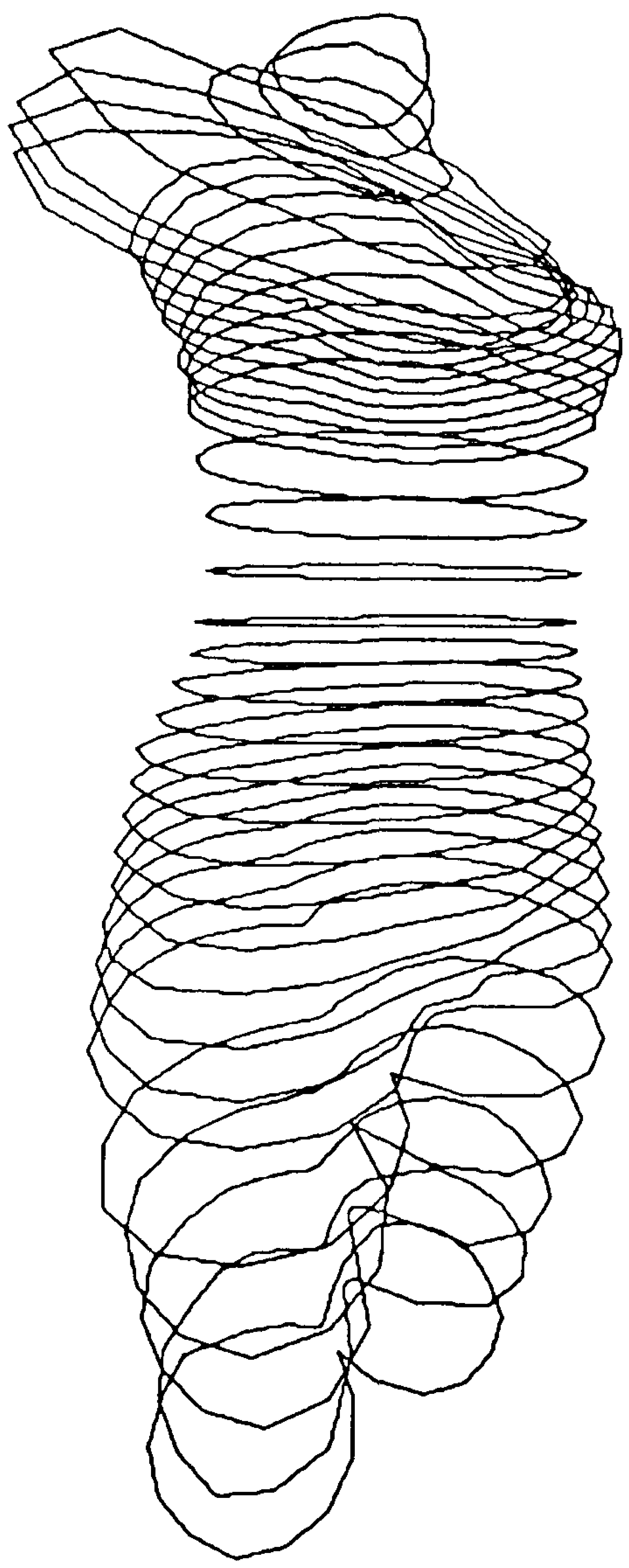


Figure 4.2 - LASS body data slices

The draping system uses a Cartesian co-ordinate system so it is fairly straightforward to adapt the data provided by the LASS team into the draping system co-ordinates. This translation method can be applied to any body model data supplied by the LASS system.

4.4 From LASS Data to Draping System

Several steps must be carried out to translate the LASS data into system co-ordinates for use in the draping system. Firstly co-ordinates must be obtained for a full body model. Then these values must be translated into draping system co-ordinates for direct use in the drape and display programs. Finally waist co-ordinates must be obtained for use in the mesh generation. An overview of these steps is shown in Figure 4.3.

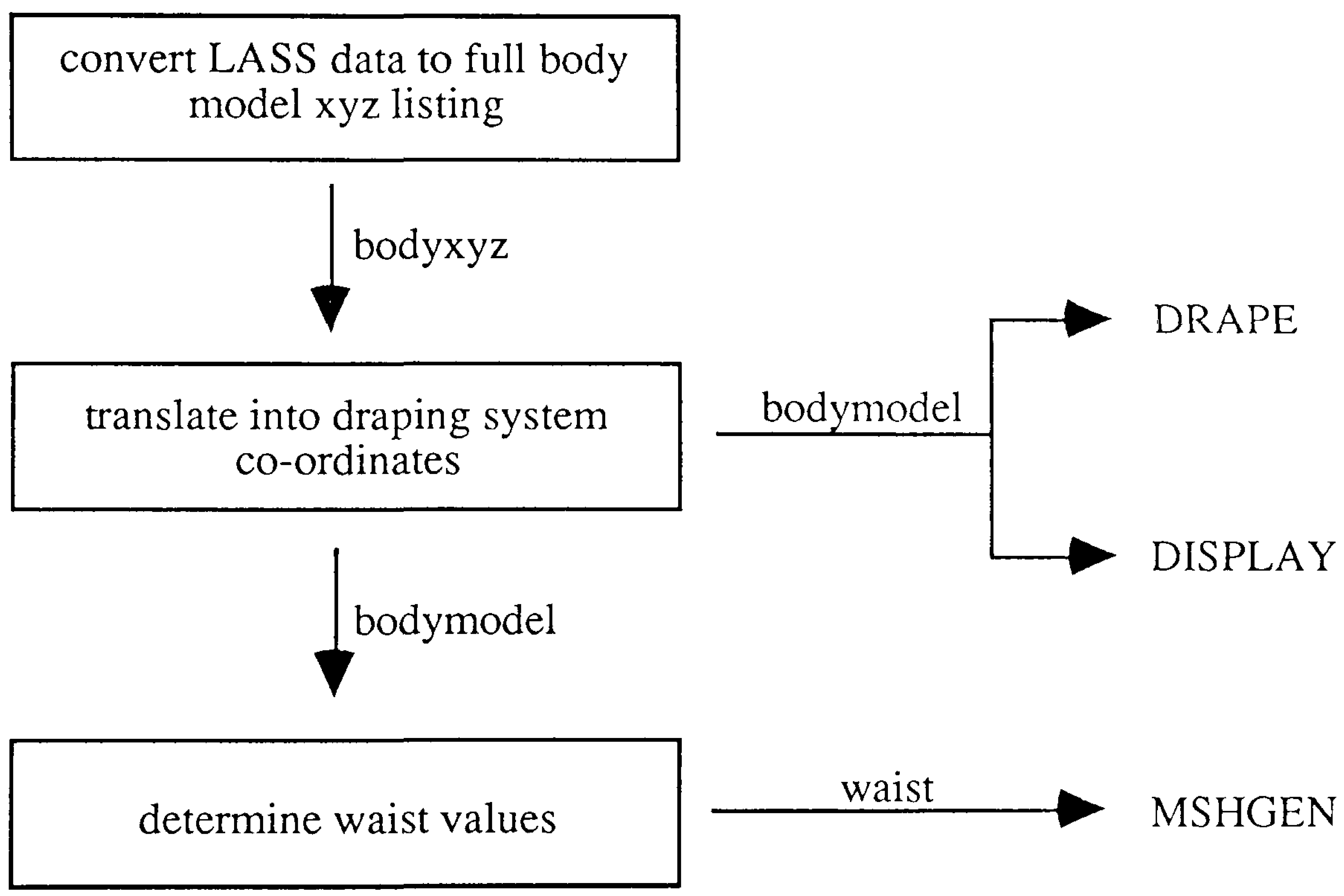


Figure 4.3 - overview of body model data

4.4.1 Obtaining Full Body Model Data

The first stage in the conversion process is to derive a complete set of body model data. The data received from LASS are the co-ordinates for a half body model and the symmetry of the body is exploited to determine the co-ordinates for the remaining half of the body.

The LASS body model is presented as a list consisting of sets of sixteen pairs of x-y data with one z value for each set. Points 2-15 for each set lie in the area where x is less than zero and points 1 and 16 lie on the x = 0 axis line as the example shown in Figure 4.4.

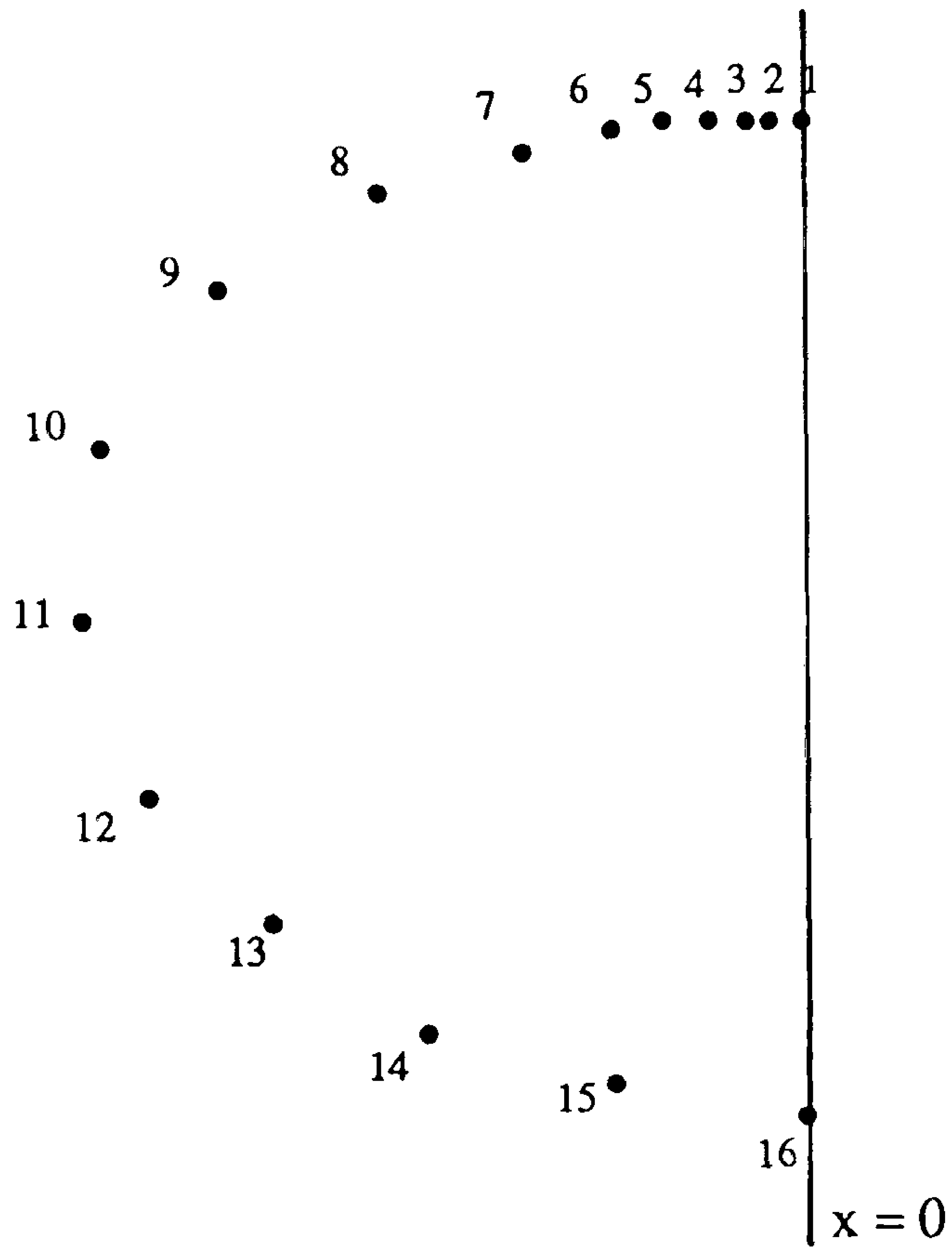


Figure 4.4 - format of the LASS data for one body slice

This means that only the points 2-15 need be considered when reproducing the remaining half of the body data. The y and z co-ordinates for this second half of the body are exactly the same as the first half while the positive value is taken for the x co-ordinate.

The result at this stage is a co-ordinate listing of thirty points for each body slice which is stored in the file "bodyxyz". An example of a full body model in wireframe is shown in Figure 4.5. The next stage is to convert these LASS co-ordinates into the range used by the draping system.

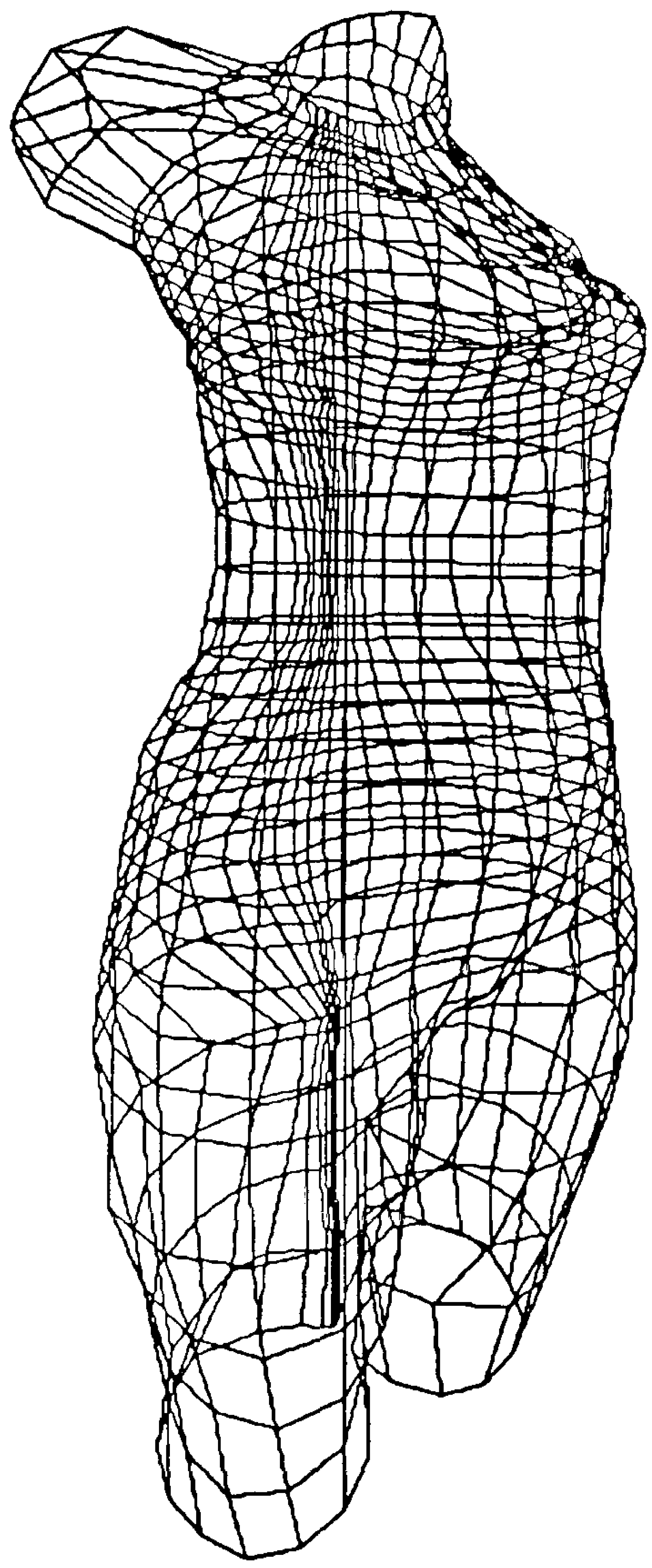


Figure 4.5 - wireframe body model

4.4.2 Translating into Draping System Co-ordinates

Translating the LASS system co-ordinates into the draping system co-ordinates is simply a matter of scaling the body model co-ordinate values, provided in millimetres, into metres which is the unit convention adopted for the draping system. The result from this process is a listing of body model data which is stored in the file "bodymodel".

4.4.3 Establishing the Waist Co-ordinates

The file "bodymodel" is used as input to both the draping program in the collision detection process and the display program where the body model is shown at the same time as the cloth mesh. However prior to these stages "bodymodel" is used as input to the third and final step in the body model process which is to find the waist co-ordinates for the mesh generator and store the results in the file "waist".

The slice relating to the waist data is known beforehand and it is a simple routine to just pick out and store this relevant slice. This file is used in the mesh generation program to determine the starting values for the cloth mesh.

4.5 Conclusion

The LASS body model is an excellent representation of real body data. The advantage of this type of information is that it is unique to individuals and as such garment representations can be customised for an individual's own needs.

The only drawback is that this body model image is static whereas moving models are required for a dynamic display such as a catwalk. Much work is being carried out on dynamic body models and to improve this system one of the extensions of this research would be to incorporate a moving body image.

A complete listing of the translation programs associated with the body model can be found in Appendix I. The programs have been written in standard FORTRAN 77 and run on a Hewlett Packard 9000 Series.

5. CLOTH MESH GENERATION

5.1 Introduction

The cloth mesh generation is an important pre-processor to the draping equations because it describes the geometry and topology of the nodes and elements of the mesh that represent the cloth. When a force is applied to a piece of cloth then the whole cloth is affected because of its structure and these important relationships must be included in the model. When a force is applied to a node then it affects not only that particular node but those nodes around it depending on the links made by the elements.

Mesh generation studies have been carried out extensively and reported in many journals. Often program listings have been provided but most of these generators involve complex elements and intricate shapes whereas the element adopted here is a simple beam element with one node at each end.

To have used an existing mesh generator would have meant extensive modification of a complex generator to obtain the more simple one required here. For this reason it was decided to develop one specifically for the project. This chapter outlines the types of methods which can be used to define a mesh and describes in detail the generator written for this particular project.

5.2 Mesh Generation Methods

One important research paper is by Ho-Le [HO88] who provides a scheme for classifying mesh generation methods. He groups them into the four main categories of mesh topology, node creation, adapted mesh template and geometry decomposition.

These four groups are outlined in more detail below and recent examples are given. Ho-Le's paper also gives many more references.

5.2.1 Mesh Topology

The first of Ho-Le's groupings is mesh topology where the actual topology of the mesh is created before nodes are placed. Regions are defined first and then mesh smoothing techniques are used to calculate the position of nodal points.

One example of this is given by Rank et al [RAN93] who describe an algorithm for generating pure triangular or quadrilateral meshes. In this example an initial triangular mesh is created by assuming a constant mesh density function over the whole domain. Nodes are then added by recursive region splitting until standard elements with three or four nodes have been established.

5.2.2 Node Creation

In the second of Ho-Le's groupings which is node creation the nodal positions are determined before being connected together to form triangular or quadrilateral elements.

An example of this is by Johnson and Sullivan [JOH92] who have developed a technique for the fully automatic generation of 2D meshes based on a normal offsetting procedure. Nodes are positioned by meshing the object boundary and then offsetting these locations along vectors normal to the boundary geometry. The next row of nodes is offset from this row and the process continues until the area is filled with nodes.

5.2.3 Adapted Mesh Template

The third group of mesh generators is where a mesh template is generated and then adapted to the object being meshed. Fernandez et al [FER93] describe a strategy for adaptive mesh generation which uses a density function to guide the generation of the new mesh. Another example of this type of mesh generation is shown by Lowther et al [LOW93] who describe a smoothing process used as the basis for an adaptive solver which is made to converge on a specified output result.

5.2.4 Geometry Decomposition

In this final grouping defined by Ho-Le, nodes and elements are created at the same time. The idea is to aim for good elements by considering the geometry of the object to be meshed. An example of this is described by Rebay [REB93] who provides details on a method for generating an efficient, unstructured triangular mesh.

5.3 The Mesh Generation Program MSHGEN

Using the above group definitions, the method adopted here falls into that of node creation (section 5.2.2) whereby node positions are determined first. The elements are created when these nodes are connected together. Beam elements have been adopted to describe the mesh. The beam element is represented by a straight line with one node at each end.

The earlier tests in the draping calculations had the loads being applied to strings. This meant that mesh generation did not need to be much more than a data file consisting of five or six nodes, the connecting elements and the physical property information. As the complexity of the examples increased so did the complexity of the meshes and eventually a more general mesh generator had to be written.

The first mesh generators for this research formed the nodes and elements in groups of triangles as shown in Figure 5.1.

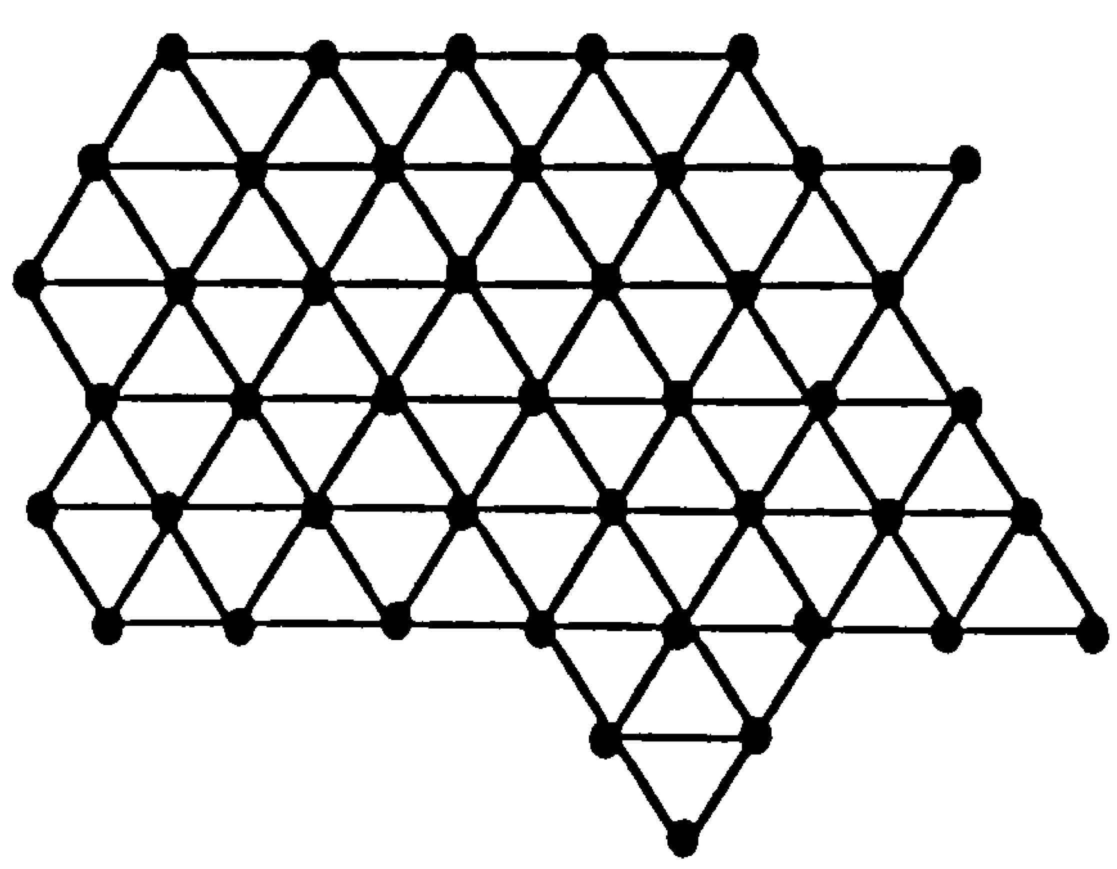


Figure 5.1 - triangular mesh arrangement

This formation worked quite well during interaction with simple solids. Later however when a body model was introduced to the problem the small faults seen with this type of mesh were exaggerated because of the non-uniformity of the body shape.

Previously, when the cylinder was being used, the solid could be easily represented by simple circles and straight lines. A coarse representation of the triangular cloth mesh fitted quite closely to this type of underlying solid. However with the uneven surfaces of a human body model a more refined cloth mesh was needed and this resulted in a significant increase in the processing time. This time increase was inconvenient but could be tolerated. The real problem was that the computer hardware limitations on matrix size were being reached during the calculation stages and the programs would not run. A new efficient mesh generator had to be created.

The current mesh generator takes into account the shape of the underlying solid. This means that the body model's waist data is read into the mesh generator and this information is used as a starting point to define the nodes and elements for a circular skirt.

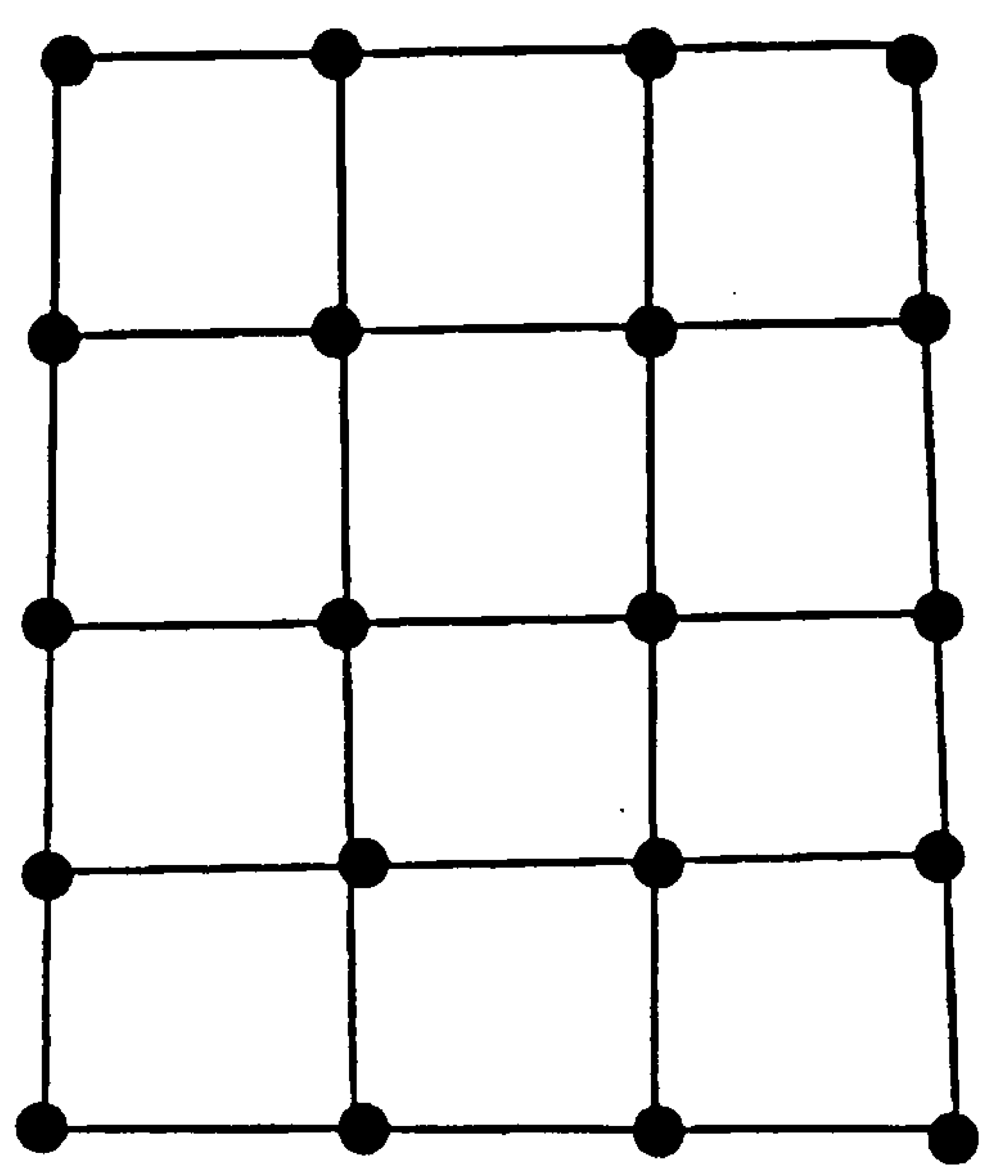


Figure 5.2 - current layout of cloth mesh nodes and elements

The element arrangement has also been altered so that each node has a maximum of four neighbours as shown in Figure 5.2 instead of the six as in the earlier mesh version. This results in fewer connections and thus fewer calculations being performed. The main benefits of this are that processing time is reduced and that hardware limitations on the size of the system matrices are not reached. This formation also mimics more closely the weave in cloth.

In essence the MSHGEN program creates a mesh of nodes and elements which represent a piece of cloth and defines various physical properties for that cloth. The output file is used as input to the draping program. MSHGEN is written in standard FORTRAN 77 and runs on a Hewlett Packard 9000 Series.

An overview of MSHGEN is given in Figure 5.3. The mesh generator for a circular skirt is described in further detail in the remainder of the chapter.

5.3.1 Input Data

The input data is in the form of two files. The first file is called "samples" and this is a database containing some measured physical property information for four cloth samples. The other physical properties which have been taken as common to all the cloth samples are established at the beginning of the program.

The second file is called "waist" and this contains a co-ordinate listing which describes the shape of the body model's waist. This body model data is obtained from the LASS body model data files and the exact method of how this is achieved has been described in the previous chapter.

5.3.2 Establishing the Cloth Waist Nodes

The next step is to define the waist nodes for the cloth mesh using the body waist data as a guide. The density of the nodes for the cloth mesh in both length and width are independent of each other and can be user-defined. These values are used to determine the number of nodes in each row of the cloth mesh and to calculate the distance between each row of nodes down the length of the skirt.

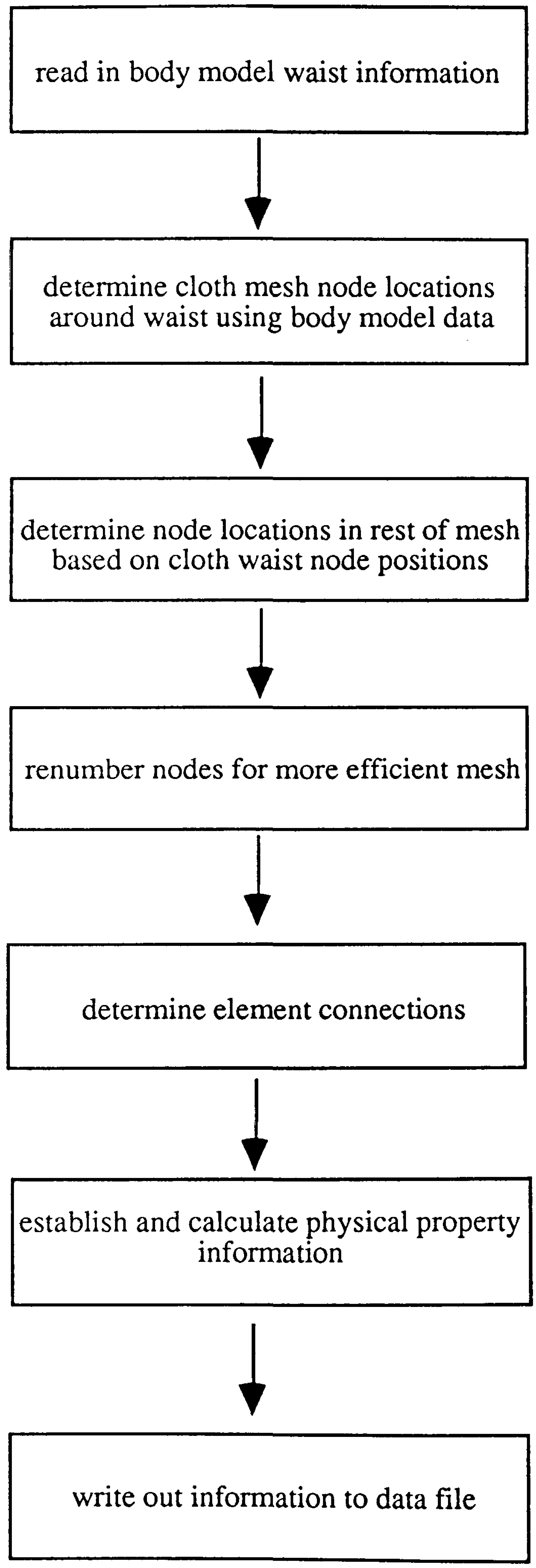


Figure 5.3 - overview of program MSHGEN

The body model waist co-ordinate information is not generally spaced equidistant apart so it is not possible to simply adopt this information as the starting node values. The body model uses thirty nodes to determine the waist. An example of waist data is shown in Figure 5.4.

Figure 5.4 - example of body model waist data

Calculation of the cloth mesh waist nodes is done in the following way. Node 1 is considered to coincide with the first node in the body model waist information and node 2 is a certain distance away. This distance is calculated as follows:

$$\text{distance between nodes} = \frac{\text{length around waist}}{\text{number of cloth nodes per row}} \tag{5.1}$$

where the length around the waist has been calculated from the information obtained in "waist" and the number of cloth nodes per row can be user-defined depending on how coarse a model is required. Node 3 is calculated to be this same distance away from node 2 and so on around the waist.

When the preliminary positions of these waist nodes have been established then an offset is added to their x and y co-ordinates. This means that the cloth model lies a very small distance away from the actual body waist.

Figure 5.5 shows the cloth mesh node positions as the larger dots to distinguish them from the underlying waist information. Fifteen cloth nodes have been created with the first cloth node in line with the first body node. The remaining newly created cloth nodes will not necessarily coincide with any existing body node position.

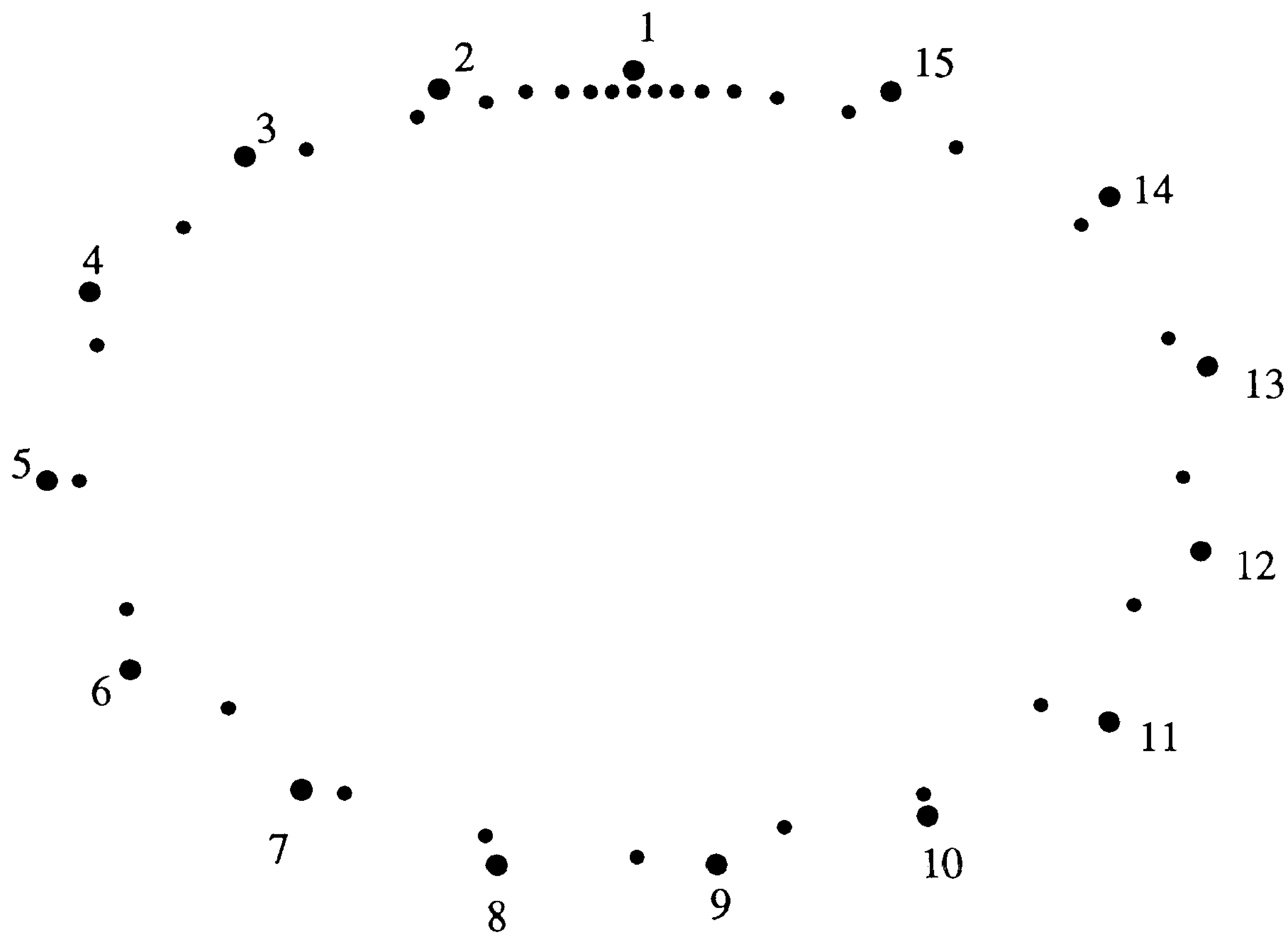


Figure 5.5 - cloth mesh waist nodes

5.3.3 Establishing the Remaining Cloth Mesh Nodes

The next row of nodes in the cloth mesh is calculated to be a certain distance away from the cloth waist nodes. This distance can be user-defined depending on the fineness of the mesh required. The length of the cloth can also be user-defined and the total number of rows of nodes is calculated using the following equation:

$$\text{total number of nodal rows} = \frac{\text{length of cloth}}{\text{node offset}} \tag{5.2}$$

The remaining rows of nodes are filled in with each row being the set distance away from the previous row. An example of this is shown in Figure 5.6.

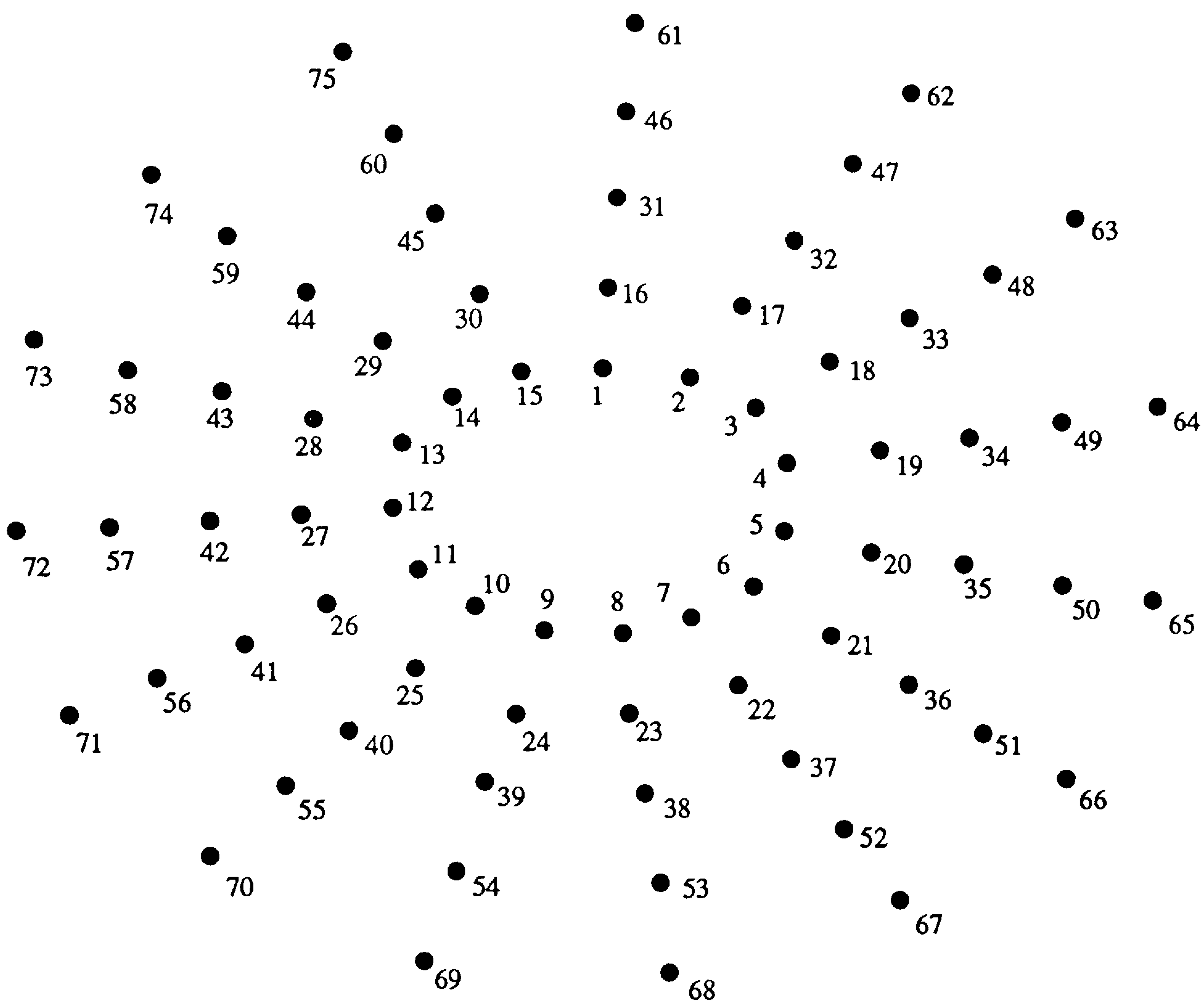


Figure 5.6 - cloth mesh nodes

There are always the same number of nodes in each row. The first node in each row is in relation to the first waist node and the row progresses in a clockwise direction around the waist. Each node lies along a line running from the centroid position through the respective cloth waist node.

5.3.4 Efficient Node Numbering

The way a mesh is numbered can seriously affect the efficiency of the calculations carried out in the draping program. One of the factors affecting the processing time is the half-bandwidth of the system matrices. The half-bandwidth is an important part of the draping system process and depends directly on the largest node number difference between neighbouring nodes in the mesh. The half-bandwidth is calculated to be:

$$\text{half-bandwidth} = \text{largest difference} + 1 \qquad (5.3)$$

A large half-bandwidth results in more calculations being performed. The half-bandwidth may be large because of the nature of the problem but to reduce the overall calculation time the mesh should be renumbered if possible to improve the efficiency.

In the example shown in Figure 5.6 the half-bandwidth is 16. By renumbering the mesh nodes, so that the differences between neighbouring nodes are less, a more efficient mesh can be achieved.

An alternative renumbering scheme is shown in Figure 5.7. Here the half-bandwidth is 11. Although this may not seem like a very large saving, in complex meshes the difference is more obvious and drastically reduces the calculation time.

The advantages of efficient mesh numbering are not only seen in faster processing times. Because the mesh has become more efficient, more refined meshes can be used and there is less likelihood that the hardware limits of the computer will be met due to sparse, inefficient matrices.

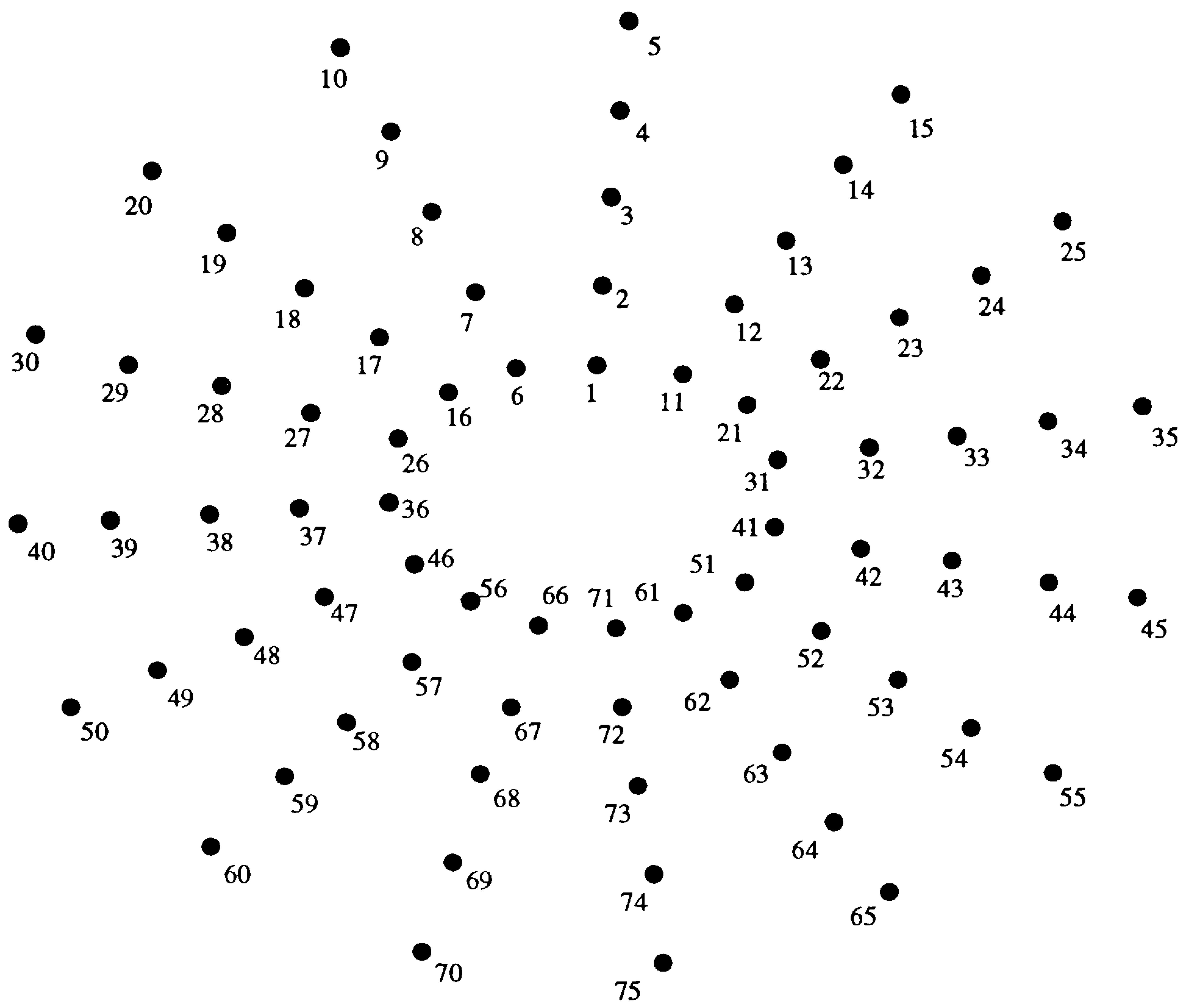


Figure 5.7 - efficient mesh numbering

5.3.5 Adding the Elements

As mentioned earlier the element type chosen is a simple beam element with one node at each end. For each node there are six degrees of freedom relating to the six forces: three directional p_x, p_y and p_z and three rotational m_x, m_y and m_z. This means that there are six corresponding displacements: d_x, d_y, d_z, θ_x, θ_y and θ_z. A representation of the element is shown in Figure 5.8.

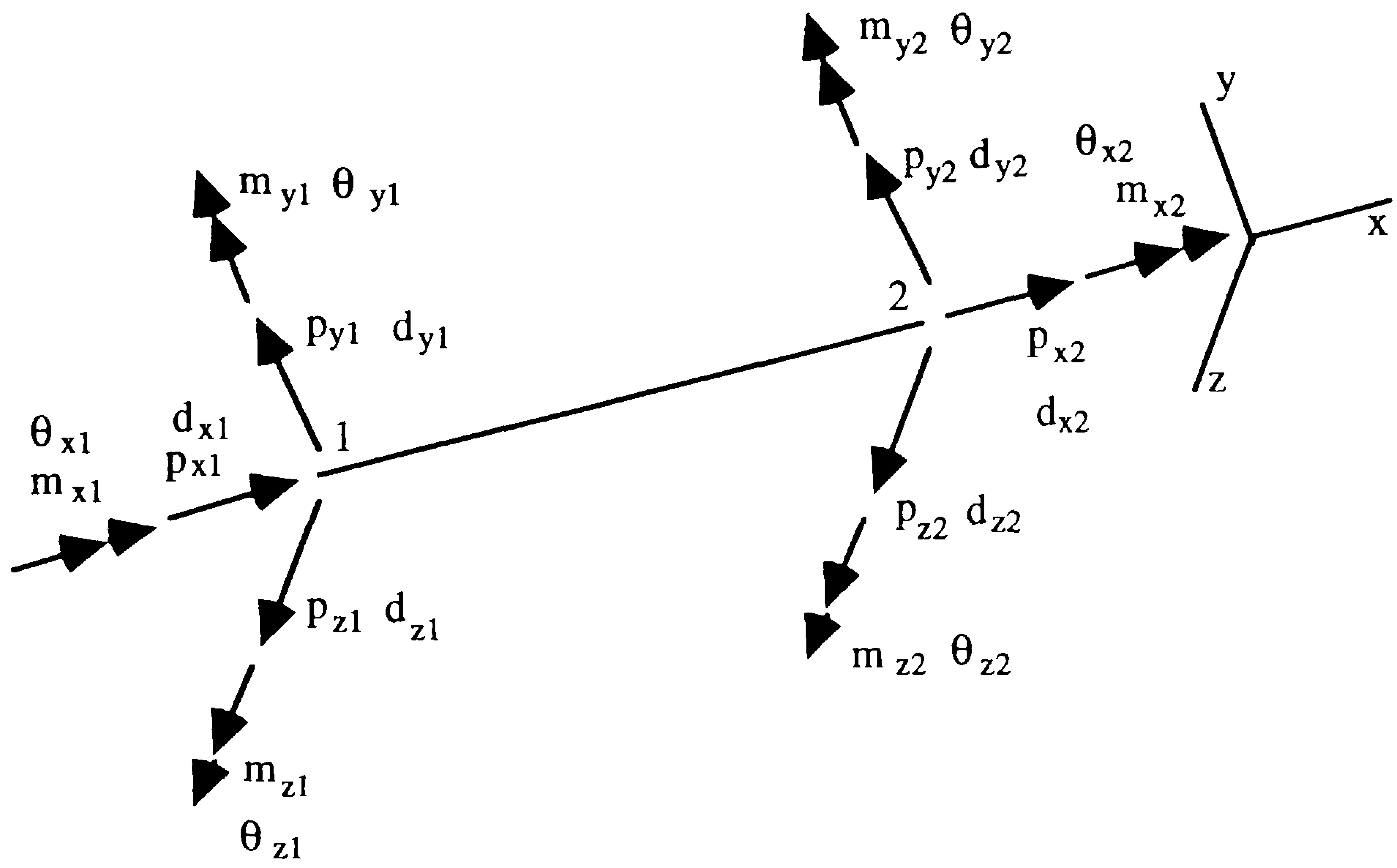


Figure 5.8 - beam element

Once the nodes have been positioned then the elements are defined. This is achieved by linking two neighbouring nodes together so that when the elements have all been entered, each node has four neighbours - except those nodes at the boundaries which will have only three neighbours. An example of this is shown in Figure 5.9.

The elements are not all the same length whereas for the earlier triangular mesh all the elements had been the same length. This caused problems because a very fine mesh was required for complex underlying solid shapes.

With this updated version of the mesh, elements are closer together where more folds are necessary such as around the waist area and further apart for larger folds around the hem of the skirt. The mesh is therefore more suited to the problem.

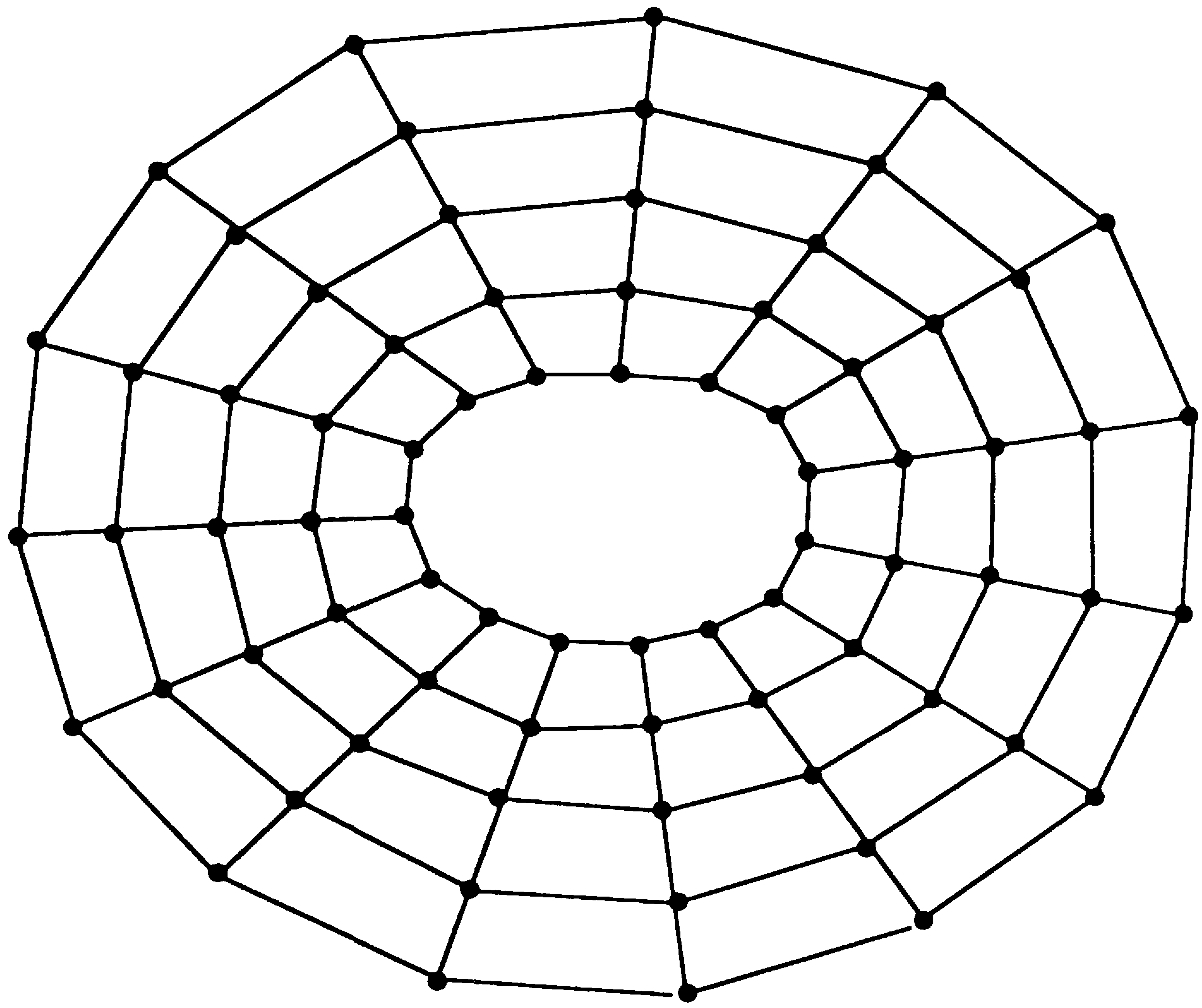


Figure 5.9 - element connections

5.3.6 Output Data

The nodal co-ordinate listing, element relationships and information about restrained nodes (those around the waist of the mesh) are written to the output file "data" along with the necessary physical properties. These are used as input to the draping program which is discussed in greater detail in the next chapter.

5.4 Conclusion

The addition of further nodes in each row adds to the realism of the drape as small folds become apparent due to the closer representation of the cloth. The examples given above show a very coarse mesh in order to keep the descriptions and diagrams straightforward - the actual programs have in general at least a hundred nodes to each row.

However the more refined the model, the longer the processing time for the calculations. There are also size limitations for the display because a complete body model must be displayed at each time step as well as the cloth model.

The mesh generator described above is really only suitable for a circular skirt. One idea for improving the cloth mesh generator to make it more applicable to any pattern piece is to align the nodes with the warp and weft of the cloth to be used. Pattern pieces have a straight grain line which is aligned with the warp of the cloth and this could be used as a starting point for defining the nodes. This also applies for garments cut on the bias because the elements will still lie in the warp and weft directions.

A program listing of MSHGEN and descriptions of the subroutines and variables are given in Appendix II.

6. DRAPE CALCULATIONS

6.1 Introduction

To make a piece of cloth look realistic during a simulation it is necessary to display folds and wrinkles occurring as the cloth drapes. Some draping methods impose fake folds onto the cloth to give the effect of realism while other methods have folds created as a natural part of the simulation.

It is the folding nature of cloth which makes drape modelling so complex. Cloth can fold and wrinkle and then regain its original shape or take up new shapes without its structure being damaged.

Another main problem to consider is collision detection when the cloth comes into contact with an underlying solid such as a body model. These problems add to the overall calculation time of the simulation as checks have to be made at each stage to ensure the model is behaving realistically.

This chapter concentrates on the draping part of the cloth animation problem. First some of the folding methods used in previous drape studies are outlined and then the actual method adopted for this particular project is given in detail. Collision detection and response is covered in more detail in the next chapter.

One of the aims of this project has been to incorporate real physical property data in the analysis. This has been achieved with the help of members of UMIST's Department of Textiles who measured four cloth samples for various property information. Some of these measured values have been incorporated into the system and are introduced in this chapter.

6.2 Drape Simulation Studies

Many studies have been carried out into the drape of cloth and these have been described in greater detail in Chapter 2. Through this work many different ways of displaying folds in cloth have also been discovered.

Imaoka *et al.* [IMA85] used finite element methods where the natural shape of the garment is developed by constrained non-linear programming methods. The shape of an underlying dummy and the physical properties of the cloth are taken into consideration to produce a static final form of a skirt.

Weil [WEI86] models a piece of cloth by using the catenary equations. Relaxation is carried out on all the points to arrive at an approximation to the cloth shape. Again the final result is a static representation of the cloth.

Terzopoulos *et al.* [TER87] simulate a flag waving in the wind. The material is modelled as a fixed metric membrane and the wind is modelled as a constant force. The effect of the wind on the flag is modelled by the viscous force using an ocean wave model based on that of Fournier and Reeves [FOU86].

The cloth model used by Aono [AON90] is based on the equilibrium equation in the field of elasticity theory and D'Alembert's principle. The equation is regarded as a wave equation which has a relationship with a vibration equation in that one always entails the other. The model is enhanced by the use of damping and anisotropic factors and modified equations between stress and strain. The model shows wrinkles which are very realistic.

Scanlon [SCA90] used Newton's Laws of Motion. In his animated model, folds develop as the cloth model drapes to its state of minimum energy.

Kunii and Gotoda [KUN90] use singularity theory to display global information and dynamics to represent local information in order to model and animate the formation processes of garment wrinkles.

Yang and Zhu [YAN91] create folds in relation to a curved surface by subdivision of that surface. They use the work of Lane and Riesenfeld [LAN80] who compare two subdivision algorithms and extend these to develop methods for surface display and finding the intersection of two curves or surfaces.

Hinds and McCartney [HIN90], Yang *et al.* [YAN92] and Gray [GRA93] use harmonic or sinusoidal functions superimposed onto panels to visualise folds and drape.

Fozzard and Rawling [FOZ91] consider only the immediate neighbours in their stress and bend force analysis of nodes. The distance between each node and its neighbours is calculated to obtain the extension or compression at that point in the cloth. The vector force due to this extension is then derived using the elasticity constant relevant to that connection depending on whether it is warp, weft or bias. The nodal displacements are calculated using a relaxation algorithm which moves the nodes so as to minimise the forces acting on them.

Research by Magnenat-Thalmann and Yang [MAG91b] determined that internal parameters drive local forces in the cloth such as stretching and curvature. They used a relaxation process to find the equilibrium at each stage in the animation. They found the most useful parameters for modifying the appearance of motion to be the density, damping, resistance coefficients, wind and time interval.

6.3 The Draping Calculation Program DRAPE

The DRAPE program drapes a piece of cloth around an underlying body model over a sequence of time steps. Cloth mesh information is passed through as the file "data" from the mesh generation program and body modelling information is provided in the shape of the file "bodymodel".

DRAPE uses as a skeleton the NAG Level 1 Library program "seg2p2e.f". This uses the finite element method to perform the forced vibration of an elastic solid through direct integration by Newmark's method. This skeleton program has been drastically modified to suit the current drape program.

The cloth drape calculations are performed taking into account the position of the underlying body model. The resulting x-y-z co-ordinate information for each node at each time step is output to the file "coord" which is used by the display program.

DRAPE uses standard FORTRAN 77 in addition to the NAG library and runs on a Hewlett Packard 9000 Series. An overview of DRAPE is given in Figure 6.1 and described in more detail below.

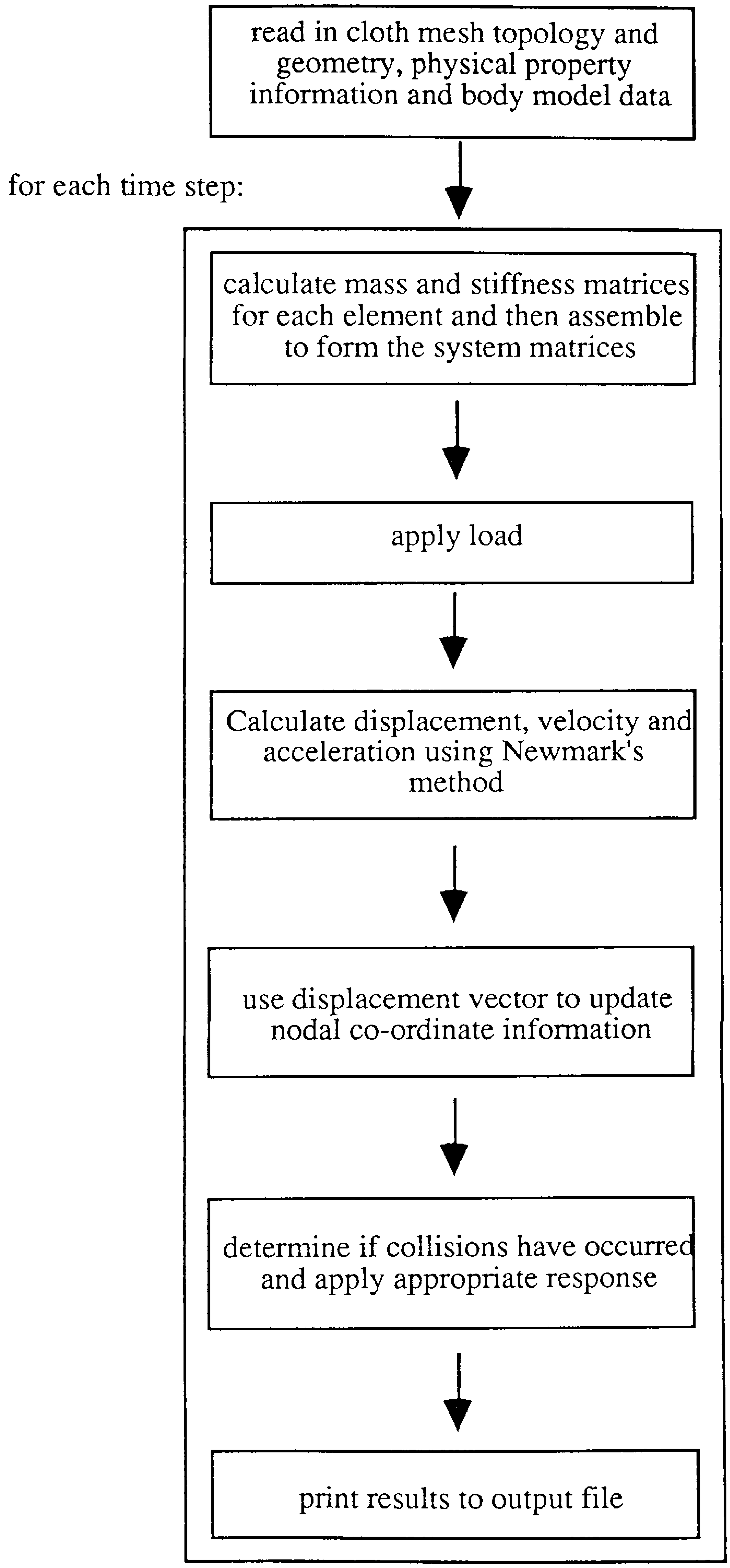


Figure 6.1 - overview of DRAPE

6.3.1 Input Data

The mesh topology and geometry information and the relevant physical properties are listed in file "data" while the body model nodal co-ordinates are found in "bodymodel". The draping of the cloth takes into account all the relationships between the nodes using the values provided and requires the body model data in order to carry out comparison checks for potential collisions.

6.3.2 The Time Step Loop

The time dimension is represented by a set of discrete points each a time increment apart. The system is solved at each of these points in time using as data the solution at the previous time.

For every time step the same series of calculations must be performed on all the elements. The results are assembled together and it is the cumulative effects of the forces that result in the displacements of each node. This section describes the basic set of equations that must be solved.

Hooke's Law states that when an elastic spring experiences a load p it deflects by an amount d and this can be represented by the following equation:

$$d = \frac{1}{k} p \tag{6.1}$$

where k is the spring stiffness constant.

Such a spring is shown in Figure 6.2 where:

$$p_1 = k\, d_1 - k\, d_2 \tag{6.2}$$

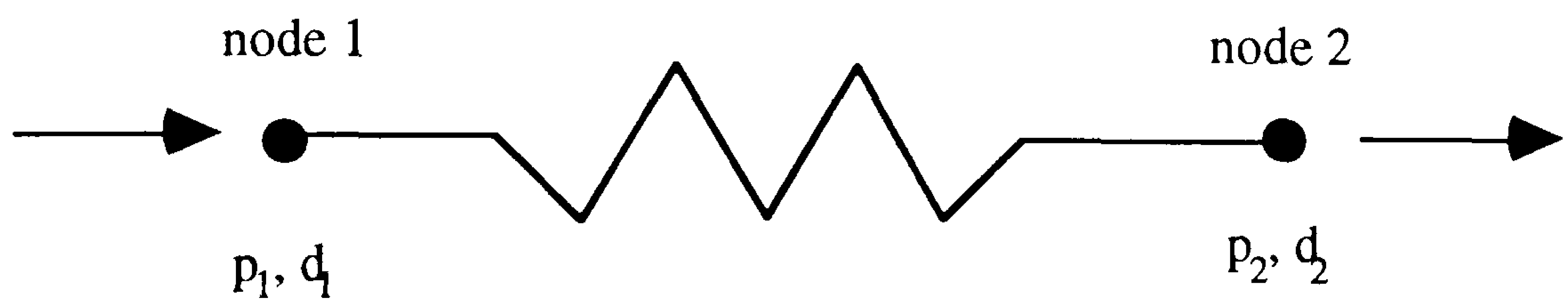


Figure 6.2 - elastic spring

For an equilibrium of forces:

$$p_2 = - p_1 \qquad (6.3)$$

so:

$$p_2 = - k\, d_1 + k\, d_2 \qquad (6.4)$$

Equations 6.2 and 6.4 can be represented as the matrix equation:

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \qquad (6.5)$$

or :

$$P = K D \qquad (6.6)$$

The external force P on a mass is made up of several components. D'Alembert's Principle states that impressed forces together with inertial forces form a system in equilibrium. This means that:

$$- P = G + Q + F \qquad (6.7)$$

where G is the damping force, Q the force in the spring element and F the inertia force.

According to Newton's Second Law of Motion:

$$F = - M\ddot{D} \tag{6.8}$$

where $\ddot{D}$ is acceleration and M is mass.

Rearranging equation 6.7 gives:

$$M\ddot{D} - G - Q = P \tag{6.9}$$

If the damping force G is proportional to the rate of deformation, i.e.:

$$G = - C\dot{D} \tag{6.10}$$

where C is a constant and $\dot{D}$ is velocity, and the spring force Q is proportional to the displacement, i.e.:

$$Q = - KD \tag{6.11}$$

where K is a constant, then equation 6.9 becomes:

$$M\ddot{D} + C\dot{D} + KD = P \tag{6.12}$$

and this is the basic equilibrium equation which is calculated at each time step for every element. The relevant matrices are developed for the system, the load is applied and Newmark's method is used to obtain values for the displacement of each node.

Various methods have been proposed to solve the equilibrium equations stated in equation 6.12 for dynamic analysis. There are two approaches which can be adopted, these being mode superposition and direct integration.

Mode superposition transforms the equilibrium equations into a set of independent equations, one for each unrestrained degree of freedom in the system. These are solved and then superposition of the results is carried out. The principle of superposition states that stresses and deformations produced in a structure by a set of loads acting in combination can be obtained by adding up the stresses and deformations produced by each load acting separately. The behaviour of a complex structure can therefore be simplified by analysing a series of unit loads applied at different points and then combining these basic solutions to form the whole solution.

Direct integration of equation 6.12 is the alternative approach that can be adopted, where the solutions are determined gradually over a series of time steps. Each time step calculates the displacements, velocities and accelerations for that time step and these are added together to form the total solutions.

Mode superposition presumes linearity which is why this technique has not been adopted here. Direct integration on the other hand can be applied to non-linear structures by updating a deformation-dependent stiffness matrix at the beginning of each time step. This approach however is subject to errors and numerical instability unless the time steps are small and small time steps in turn lead to longer processing times.

Stability in a system requires that the initial conditions of the system are not amplified artificially when a large value is used for the time step length. Also any errors due to round off by the computer in the displacements, velocities and accelerations should not increase as the integration is carried out. Conditional stability occurs when accuracy can only be achieved through the use of small time steps. Unconditional stability occurs when larger values for the time step length can be used without losing accuracy.

Direct methods can be approached either explicitly or implicitly. Implicit methods require the solution of the whole set of equations, simultaneously at each time step. Although this means a large computing effort is required, this does result in a stable system. Explicit methods require a much smaller computing effort because the solutions are calculated from 'uncoupled' equations. However this results in an unstable system.

Since unconditional stability was required for this research, and there do not seem to be any unconditionally stable explicit methods, an implicit integration scheme was sought. Of the implicit methods, perhaps that proposed by Newmark is most widely used in dynamic analysis problems. When used as a constant-average-acceleration method, Newmark's scheme is unconditionally stable. The method is described in more detail later in this chapter.

6.3.3 System Matrices

The matrices which must be developed to determine the solution at each stage are the mass matrix M, the damping matrix C and the stiffness matrix K. The mass and stiffness matrices are established first for each element and the information is then assembled to form the relevant system matrices.

The elements are defined in three dimensions so the problem is concerned with the equilibrium of forces in 3D space. This means that for each node there are six forces: three direct forces p_x, p_y and p_z and three bending moments around the axes m_x, m_y and m_z. As each element has two nodes, this means that there are twelve degrees of freedom for each element in the system. This is shown in Figure 6.3.

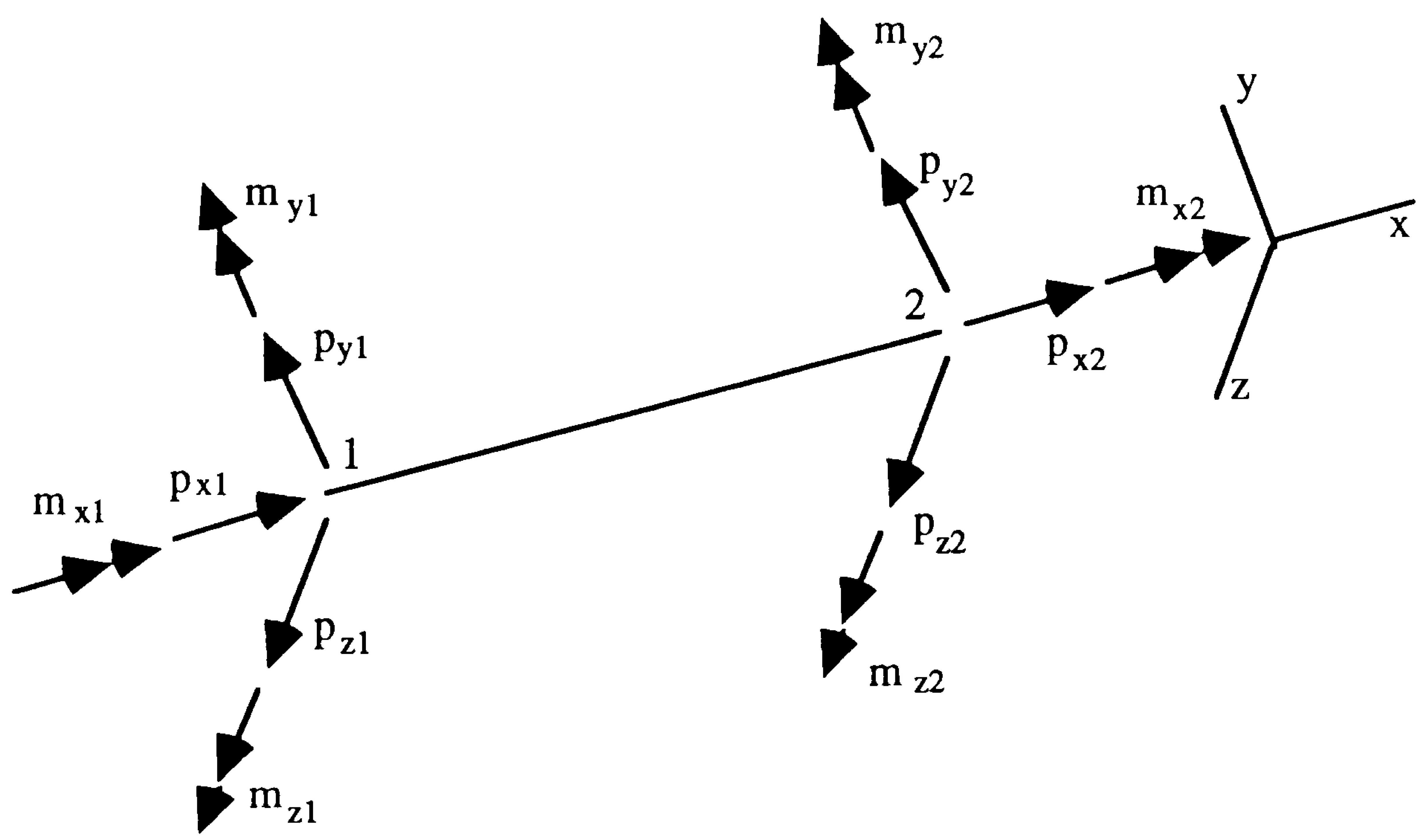


Figure 6.3 - forces and moments

Similarly there are potentially six displacements relating to each degree of freedom in each node: three linear displacements d_x, d_y and d_z and three rotations around the axes θ_x, θ_y and θ_z. These twelve displacements for one element are shown in Figure 6.4.

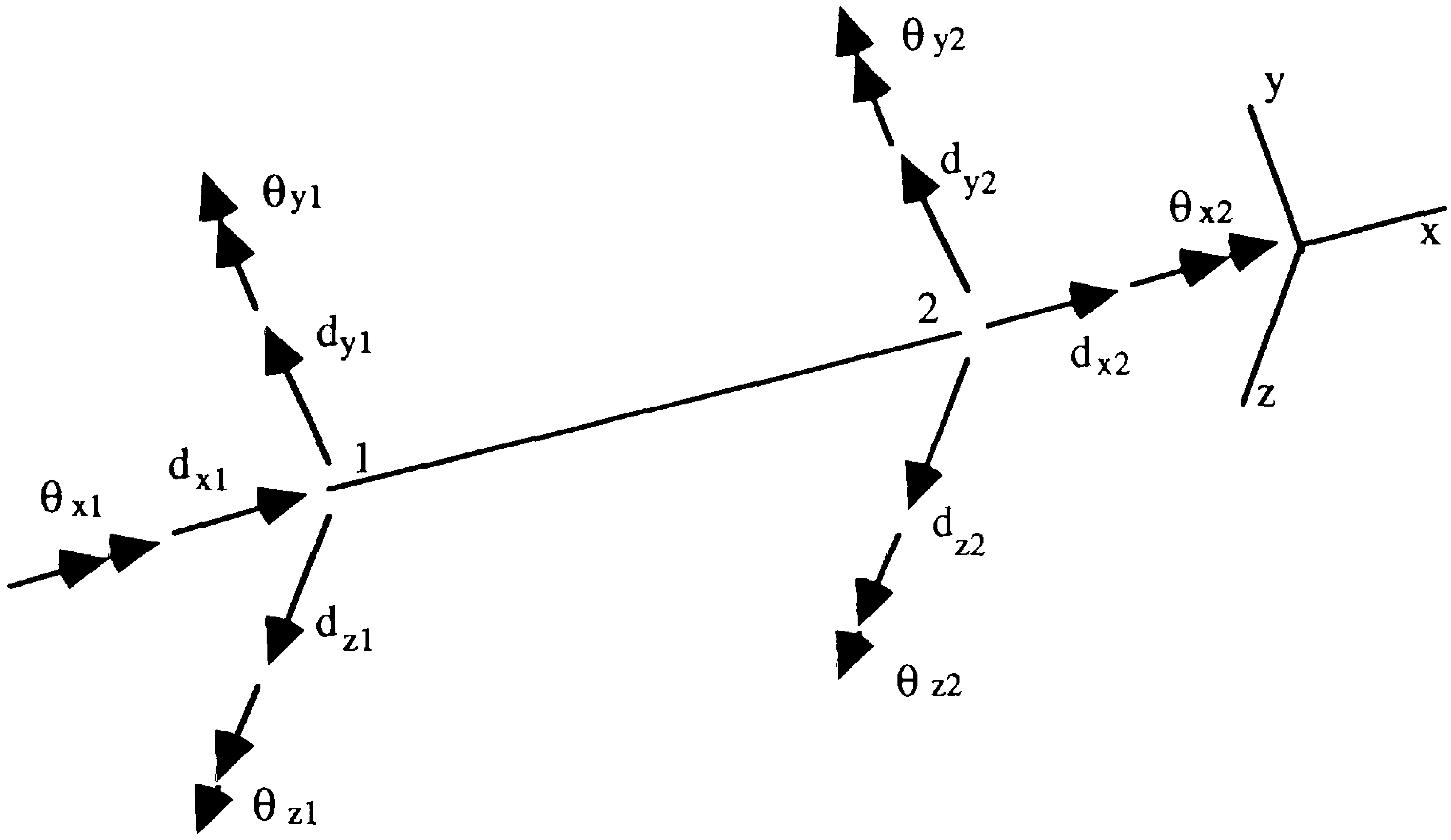


Figure 6.4 - displacements and rotations

This means that each element matrix must be developed relating to its twelve degrees of freedom.

6.3.3.1 Stiffness Matrix K

In a structural problem there are two types of non-linearity and these are the material and the geometric. The stiffness matrix K in this project is formed by adding together information from the elasticity stiffness matrix K_E and the geometric stiffness matrix K_G.

Material non-linearity is due to the non-linearly elastic and viscoelastic behaviour of the material and is represented by K_E. Geometric non-linearity occurs when the deflections are large enough to cause significant changes in the geometry of the structure. In this case equations of equilibrium must be formulated for the deformed configuration and these are represented by K_G.

The geometrical stiffness is required in order to simulate the effects of displacements on the equilibrium conditions. Problems requiring this analysis can be referred to as large-deflection problems. They differ from linear and small-deflection problems because axial stresses exist within the elements that exert significant influences on the stiffness of the structure.

For linear problems the stiffness matrix terms of the system are constant throughout the analysis, so the stiffness matrix is comprised only from the elasticity stiffness matrix, i.e.:

$$K = K_E \tag{6.13}$$

The stiffness matrix for the geometrically non-linear problem however contains terms that are functions of the deformation of the structure. This means that the stiffness matrix must be updated at each step with respect to the deformed geometry.

The geometrically non-linear problem can be tackled in two ways: incrementally by linear methods using a tangent stiffness matrix and directly using a secant stiffness matrix.

With the linear incremental method the load is applied as a series of small increments and for each of these increments the displacements are calculated using a linear analysis. A tangent stiffness matrix is used that is based on the geometry and internal forces existing at the beginning of the time step. The stiffness matrix therefore consists of two distinct matrices:

$$K = K_E + K_G \tag{6.14}$$

K_E is the linear elasticity stiffness matrix for uncoupled bending and axial behaviour. The terms in this matrix are constants and do not depend on the internal element forces and displacements. K_G contains linear functions of the internal axial force present at the beginning of each time step and provides a first order approximation of the interaction between the axial force and the transverse displacements.

With the direct method the entire load is applied in a single step. Whereas the previous method uses incremental loads and incremental displacements, this method deals with total loads and total displacements. A secant stiffness matrix is used that depends on the internal forces and displacements that exist when the total load is acting. Since these quantities are not known at the beginning of the calculation, the terms of this stiffness matrix must be determined by iteration. The stiffness matrix therefore consists of three distinct matrices:

$$K = K_E + \frac{AE}{2} K_1 + \frac{AE}{3} K_2 \qquad (6.15)$$

where the nodal displacements making up K are total displacements. K_E is as defined for the previous method. K_1 contains linear functions and K_2 quadratic functions of the incremental element displacements. A is the cross-sectional area of the element and E is the modulus of elasticity of the element.

Figure 6.5 shows the way in which these two methods approximate the actual solution, in addition to a third method. This latter method, the non-linear incremental, is comprised from stages taken from both the linear incremental and direct methods.

With the non-linear incremental method the load is applied as a series of small increments, as with the linear incremental method. An incremental stiffness matrix must be established that is a function of the internal forces and displacements existing at the beginning of the load step in addition to those that are developed during the load step. An iteration scheme is therefore used throughout the step to continuously update the stiffness matrix as better approximations are calculated, as with the direct method.

The stiffness matrix for this method therefore consists of the sum of the four distinct matrices:

$$K = K_E + K_G + \frac{AE}{2} K_1 + \frac{AE}{3} K_2 \qquad (6.16)$$

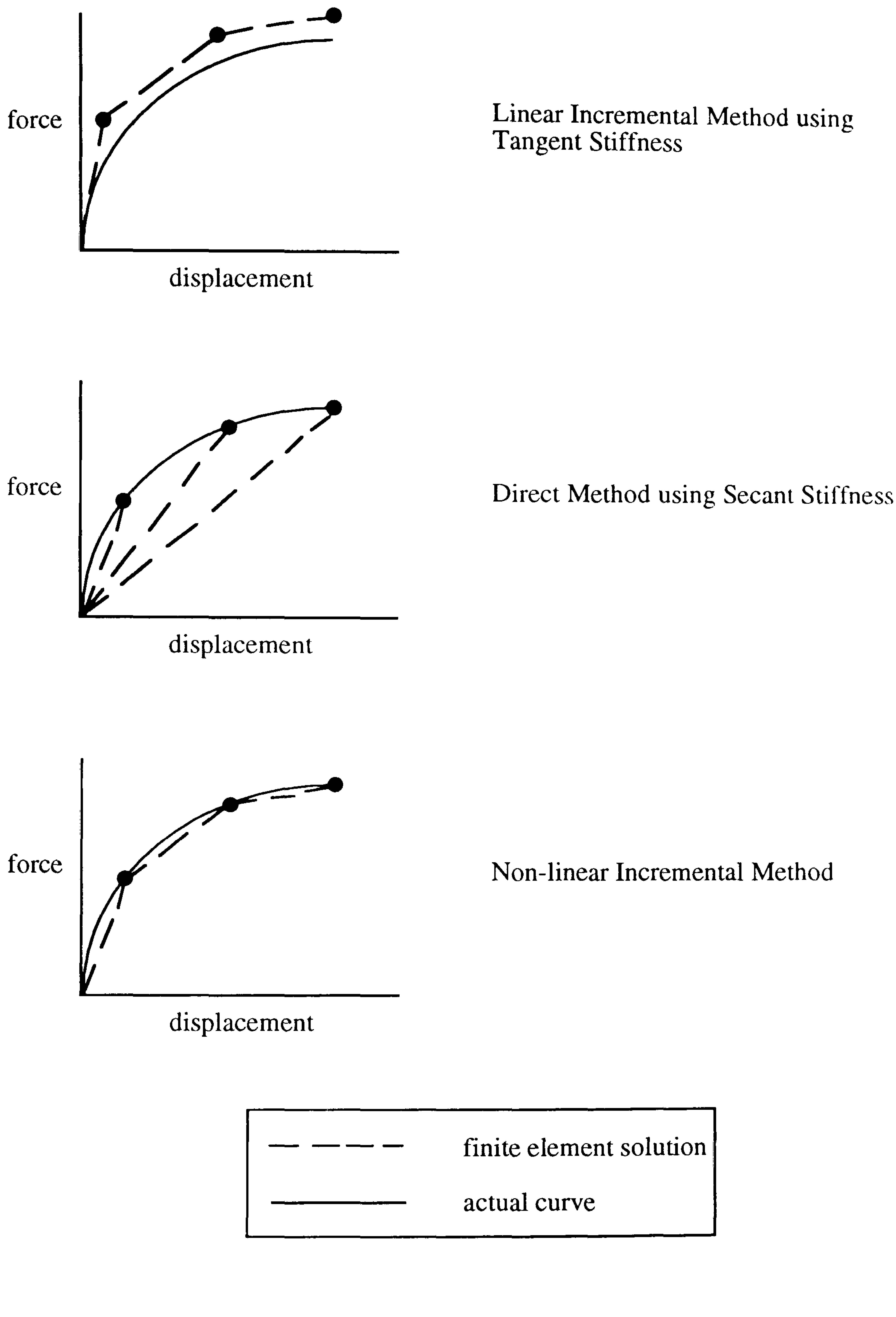


Figure 6.5 - non-linear geometric stiffness methods

Chajes and Churchill [CHA87] provide further detail on these methods and how the respective stiffness matrix components can be developed. As implied earlier the technique used here is the linear incremental method comprised of the elasticity matrix K_E and the geometric matrix K_G.

The elasticity stiffness matrix K_E for each element is declared in equation 6.17. The format of the matrix is used frequently in standard structural analysis problems and described in detail in many texts. It is not necessary to explain the proof here except to define the variables, however a full description of how the components of this matrix have been established is discussed by Coates *et al.* [COA88, Ch 5].

$$K_E = E\left[\begin{array}{cccccc|cccccc}
\frac{A}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{A}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12I_z}{L^3} & 0 & 0 & 0 & \frac{6I_z}{L^2} & 0 & -\frac{12I_z}{L^3} & 0 & 0 & 0 & \frac{6I_z}{L^2} \\
0 & 0 & \frac{12I_y}{L^3} & 0 & -\frac{6I_y}{L^2} & 0 & 0 & 0 & -\frac{12I_y}{L^3} & 0 & -\frac{6I_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{J}{2L(1+\nu)} & 0 & 0 & 0 & 0 & 0 & \frac{-J}{2L(1+\nu)} & 0 & 0 \\
0 & 0 & -\frac{6I_y}{L^2} & 0 & \frac{4I_y}{L} & 0 & 0 & 0 & \frac{6I_y}{L^2} & 0 & \frac{2I_y}{L} & 0 \\
0 & \frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{4I_z}{L} & 0 & -\frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{2I_z}{L} \\
\hline
-\frac{A}{L} & 0 & 0 & 0 & 0 & 0 & \frac{A}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12I_z}{L^3} & 0 & 0 & 0 & -\frac{6I_z}{L^2} & 0 & \frac{12I_z}{L^3} & 0 & 0 & 0 & -\frac{6I_z}{L^2} \\
0 & 0 & -\frac{12I_y}{L^3} & 0 & \frac{6I_y}{L^2} & 0 & 0 & 0 & \frac{12I_y}{L^3} & 0 & \frac{6I_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{-J}{2L(1+\nu)} & 0 & 0 & 0 & 0 & 0 & \frac{J}{2L(1+\nu)} & 0 & 0 \\
0 & 0 & -\frac{6I_y}{L^2} & 0 & \frac{2I_y}{L} & 0 & 0 & 0 & \frac{6I_y}{L^2} & 0 & \frac{4I_y}{L} & 0 \\
0 & \frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{2I_z}{L} & 0 & -\frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{4I_z}{L}
\end{array}\right] \tag{6.17}$$

The constants declared in equation 6.17 are described below.

Stress is defined to be the load per unit area and can be written as:

$$\sigma = \frac{P}{A} \tag{6.18}$$

where P is the load and A is the area of each element. The area is based on the width of the cloth per element pitch, b, and the cloth thickness, t, and related as follows:

$$A = b\,t \tag{6.19}$$

The values for t have been provided as part of the measured properties and are presented in Table 6.1.

Sample	Thickness m
100% Polyester	2.58E-04
100% Cotton	7.81E-04
100% Wool	6.41E-04
50% Polyester / 50% Cotton	4.46E-04

Table 6.1 - measured thicknesses of sample textiles

Strain is defined to be a measure of the deformation produced in the element by the load and can be written as:

$$\varepsilon = \frac{U}{L} \tag{6.20}$$

where U is the change in length of the element and L is the original length of the element.

Hooke's Law states that the strain is proportional to the stress, so:

$$E = \frac{\sigma}{\varepsilon} \tag{6.21}$$

where E is the modulus of elasticity known as Young's modulus. E forms part of the flexural rigidity B:

$$B = E\ I \tag{6.22}$$

where I is the moment of inertia or second moment of area. The values for B have been provided as part of the measured cloth data and are presented in Table 6.2.

Sample	Flexural Rigidity B Nm^2/m	
	warp	weft
100% Polyester	3.893240E-06	2.755669E-06
100% Cotton	1.998595E-05	1.587697E-05
100% Wool	1.319975E-05	9.286898E-06
50% Polyester / 50% Cotton	1.010085E-05	5.060231E-06

Table 6.2 - bending properties of sample textiles

The moments of inertia about the two axes in plane with the cross-section of an element are denoted as I_y and I_z and are known as the principal second moments of area. The moment of inertia about the axis perpendicular to these, logically denoted as I_x, is conventionally denoted as J in torsion studies. J is known as the polar second moment of area.

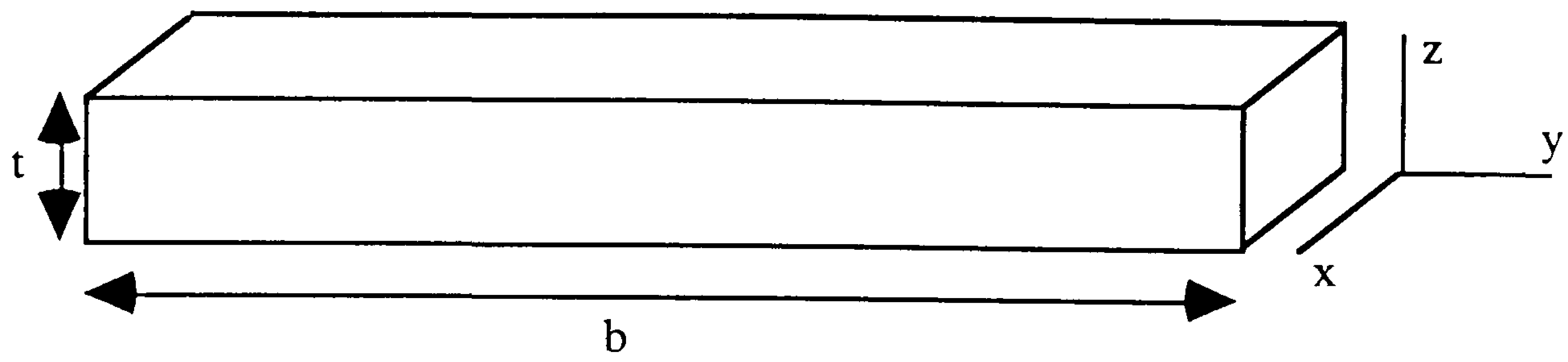


Figure 6.6 - cross-sectional area represented by element

Figure 6.6 shows a cross-sectional area of the cloth relating to one element. The x-axis coincides with the centroidal line of the element. The moment of inertia around the x-axis is given as:

$$J = I_y + I_z \quad (6.23)$$

As the cross-sectional area is rectangular, the moment of inertia around the y-axis is given as:

$$I_y = b\, t^3 \,/\, 12 \quad (6.24)$$

The assumption made to determine the moment of inertia around the z-axis is demonstrated in Figure 6.7.

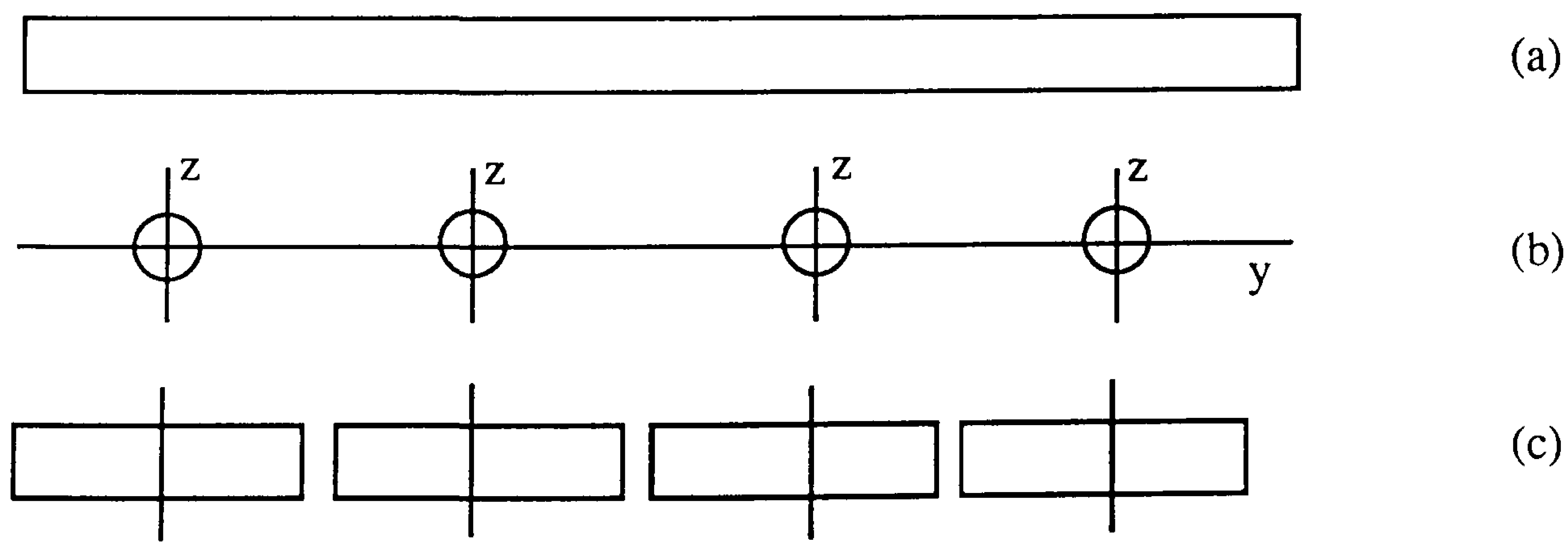


Figure 6.7 - assumed representation of continuous cloth

The cross-sectional area of cloth shown in Figure 6.7(a) is represented as a series of elements, Figure 6.7(b), in such a way that the cloth is assumed to be split into discrete sections, Figure 6.7(c). The value for the moment of inertia around the z-axis, using this assumption, is therefore:

$$I_z = t\, b^3 \;/\; 12 \tag{6.25}$$

Although this does not represent the cloth exactly, it has been used as a reasonable approximation to obtain a value for I_z. However future work would involve looking into establishing a more accurate value.

The direct strain ε_x is always accompanied by lateral strains ε_y and ε_z which are proportional to ε_x but of opposite sign:

$$\varepsilon_y = \varepsilon_z = -\nu\, \varepsilon_x \tag{6.26}$$

where ν is the Poisson's ratio of the material and is defined to be:

$$\nu = \frac{\text{lateral strain}}{\text{longitudinal strain}} \tag{6.27}$$

De Jong and Postle [DE77a] and Collier *et al.* [COL91] point out the problems associated in measuring Poisson's ratio in textiles. These problems arise due to the structure of the fabric and yarn crimp and also the difficulty in determining accurately any thickness changes.

For many cloth materials Poisson's ratio can be taken to be equal to 0.25 [TIM70] and so this was the value adopted for this research.

The geometric stiffness matrix K_G is declared in equation 6.28. A description of how this has been obtained is discussed in further detail by Przemieniecki [PRZ68, Ch 15].

$$K_G = P_{in} \left[\begin{array}{cccccc|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{6}{5L} & 0 & 0 & 0 & \frac{1}{10} & 0 & \frac{-6}{5L} & 0 & 0 & 0 & \frac{1}{10} \\
0 & 0 & \frac{6}{5L} & 0 & \frac{-1}{10} & 0 & 0 & 0 & \frac{-6}{5L} & 0 & \frac{-1}{10} & 0 \\
0 & 0 & 0 & \frac{2L}{15} & 0 & 0 & 0 & 0 & 0 & \frac{-L}{30} & 0 & 0 \\
0 & 0 & \frac{-1}{10} & 0 & \frac{2L}{15} & 0 & 0 & 0 & \frac{1}{10} & 0 & \frac{-L}{30} & 0 \\
0 & \frac{1}{10} & 0 & 0 & 0 & \frac{2L}{15} & 0 & \frac{-1}{10} & 0 & 0 & 0 & \frac{-L}{30} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-6}{5L} & 0 & 0 & 0 & \frac{-1}{10} & 0 & \frac{6}{5L} & 0 & 0 & 0 & \frac{-1}{10} \\
0 & 0 & \frac{-6}{5L} & 0 & \frac{1}{10} & 0 & 0 & 0 & \frac{6}{5L} & 0 & \frac{1}{10} & 0 \\
0 & 0 & 0 & \frac{-L}{30} & 0 & 0 & 0 & 0 & 0 & \frac{2L}{15} & 0 & 0 \\
0 & 0 & \frac{-1}{10} & 0 & \frac{-L}{30} & 0 & 0 & 0 & \frac{1}{10} & 0 & \frac{2L}{15} & 0 \\
0 & \frac{1}{10} & 0 & 0 & 0 & \frac{-L}{30} & 0 & \frac{-1}{10} & 0 & 0 & 0 & \frac{2L}{15}
\end{array} \right] \tag{6.28}$$

Essentially the matrix contains linear functions of the internal axial force P_{in} present at the beginning of each time step. The purpose of this matrix is to provide a first order approximation of the interaction between the axial force and the transverse displacements.

The internal force P_{in} is determined by:

$$P_{in} = (E\ A\ U) / L \tag{6.29}$$

where U is the change in length of the element.

The elasticity and geometric stiffness matrices are added together to form the stiffness matrix K for each element. These individual element stiffness matrices are then assembled together to form the system stiffness matrix, but first they must be translated from the local co-ordinate system to the global co-ordinate system.

Each element is defined in its local co-ordinate system with the x-axis coinciding with the centroidal line of the element. A structure consisting of N elements therefore has N local systems of axes. The forces and displacements must be stated in terms of the global co-ordinate system, with axes denoted as x', y' and z', before the system matrices are assembled. This is achieved by applying a transformation matrix T to the forces and displacements.

The axes x, y and z are seen as having been in line initially with x', y' and z' respectively and then rotated to lie in their final positions. The force vector in the global axes is denoted as P' and the displacement vector as D'. Any quantity defined in terms of the local axes can be redefined in terms of the global axes through multiplication by the transformation matrix. This is shown for the x-y plane in Figure 6.8.

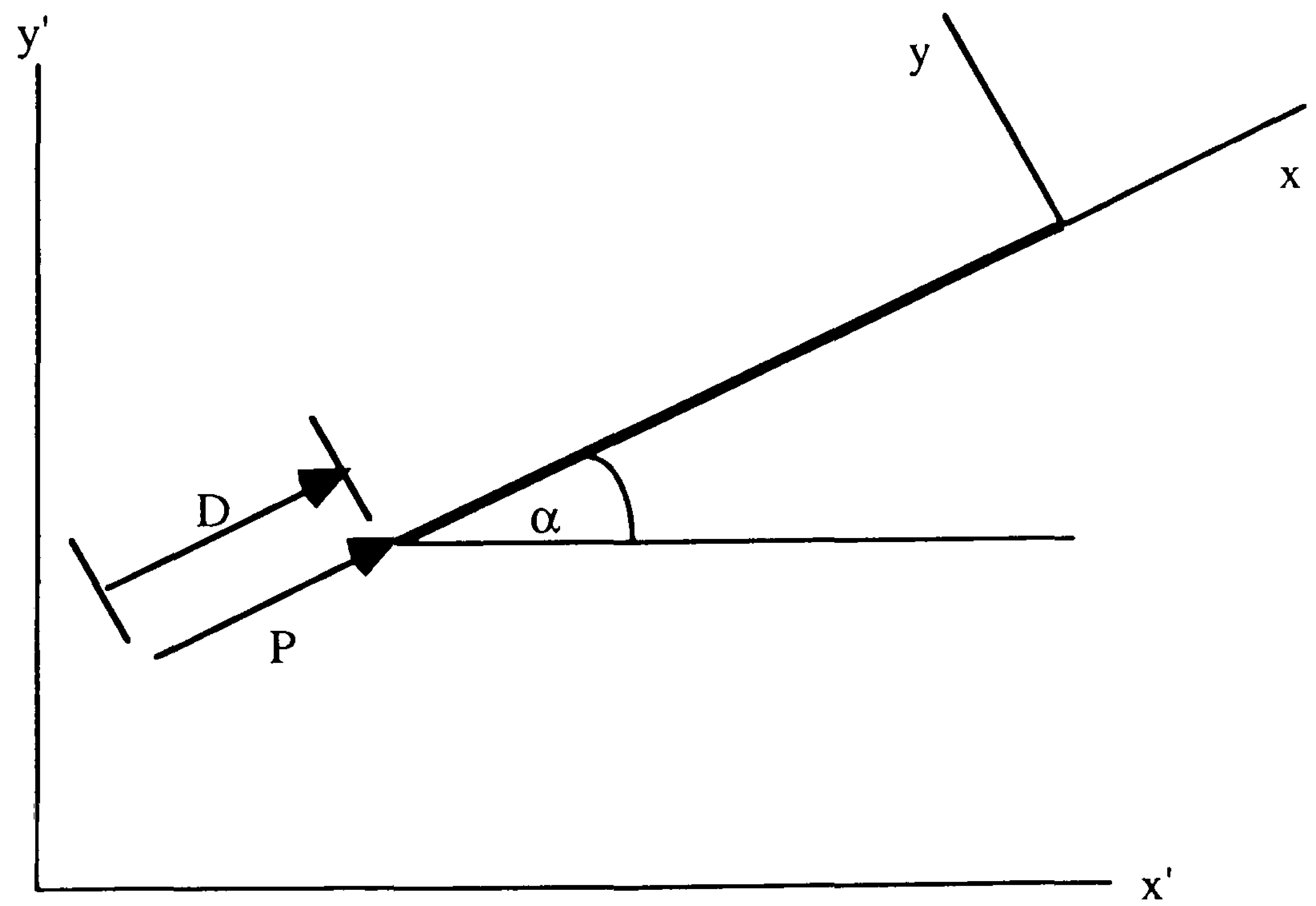


Figure 6.8 - co-ordinate transformation

Translating the example in Figure 6.8 from the local system to the global system means performing the calculations:

$$P'_x = P \cos \alpha \qquad (6.30a)$$
$$P'_y = P \sin \alpha \qquad (6.30b)$$

which can be written as:

$$P' = T P \qquad (6.31)$$

where:

$$T = \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix} \qquad (6.32)$$

The transpose of T is also the inverse of T:

$$T^T = [\cos \alpha \quad \sin \alpha] \qquad (6.33)$$

so:

$$P = T^T P' \qquad (6.34)$$

Displacement D can be treated in a similar fashion:

$$D = T^T D' \qquad (6.35)$$

Equation 6.6 gives P = K D, so multiplying this by T and using equation 6.31 gives:

$$P' = T P = T K D \qquad (6.36)$$

Substituting for D gives:

$$P' = (T K T^T) D' \qquad (6.37)$$

which can be written as:

$$P' = K' D' \quad (6.38)$$

where:

$$K' = T K T^T \quad (6.39)$$

This means that for the calculations to be performed, the stiffness matrix must be translated from the local to the global axis system.

The three-dimensional problem is more complex than the two-dimensional one. Figure 6.9 shows an element in the global axis frame x'-y'-z'. The local axis frame x-y-z has the x-axis coinciding along the length of the element.

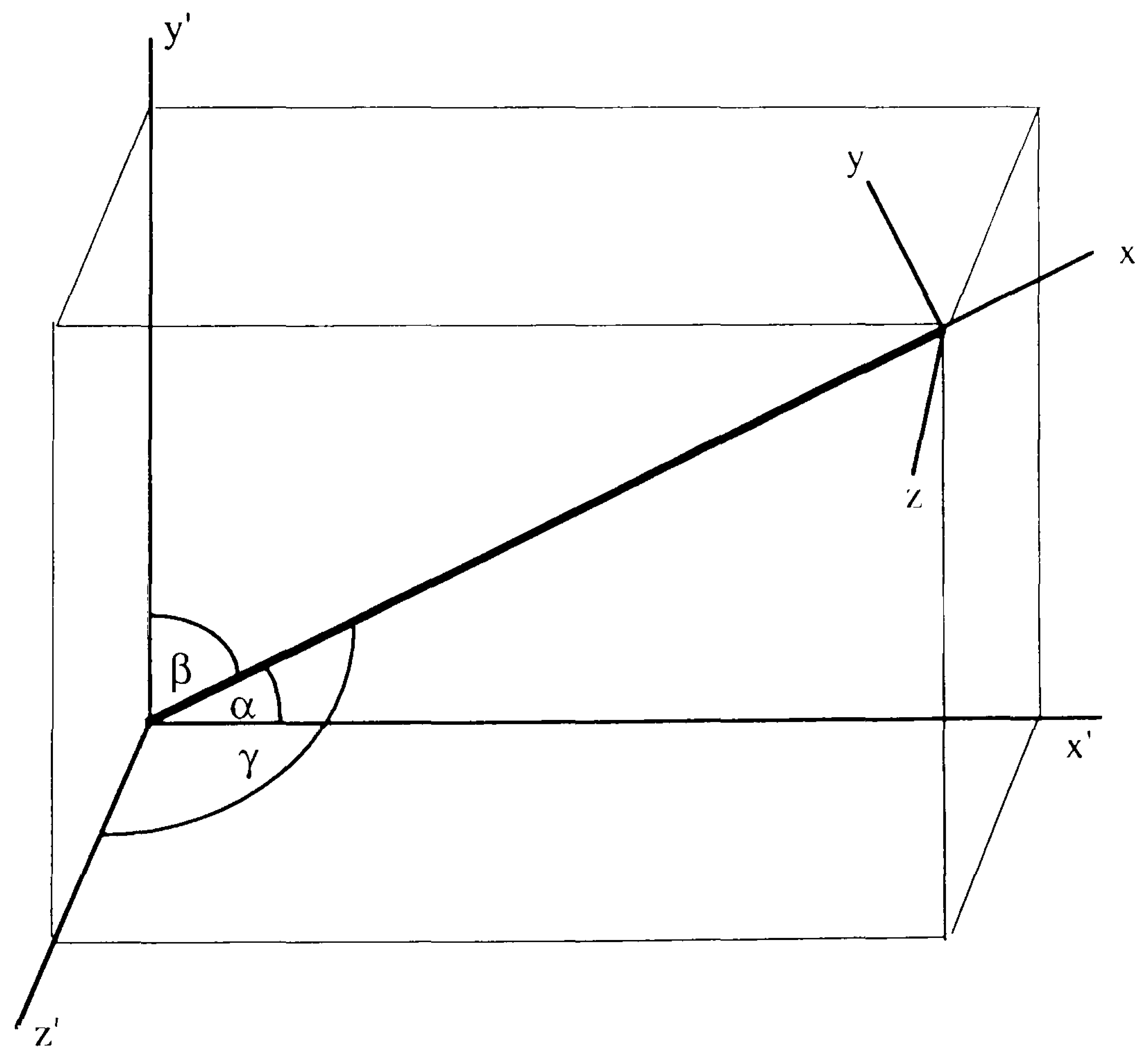


Figure 6.9 - rotation of axes in 3D

The transformation matrix is effectively achieved by assuming that the local axes initially coincide with the global axes and are rotated until they lie in the correct position. The matrix is actually achieved by calculating the angles between each local axis and each global axis.

Starting with x first, the angles between x in the local axis system and x', y' and z' in the global system are α, β and γ respectively, as shown in Figure 6.9, therefore:

$$\cos\alpha = \cos x'x = l_x \tag{6.40a}$$

$$\cos\beta = \cos y'x = m_x \tag{6.40b}$$

$$\cos\gamma = \cos z'x = n_x \tag{6.40c}$$

where l_x, m_x and n_x are known as the direction cosines of the local x-axis.

Applying the same reasoning for y and z, the equations for the transformation matrix can be written as:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6.41}$$

where:

$$T = \begin{bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \tag{6.42}$$

This matrix transforms the local axis system to the global. Transforming from global to local uses T^T:

$$T^T = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} \tag{6.43}$$

The moments of an element can be transformed in exactly the same way as the forces. This means that the T matrix defined in equation 6.42 applies to the moments as well. For an element with twelve degrees of freedom, the T matrix is therefore:

$$T = \begin{bmatrix} R & & & \\ & R & & \\ & & R & \\ & & & R \end{bmatrix} \qquad (6.44)$$

since linear displacements are independent of rotational displacements,and:

$$R = \begin{bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \qquad (6.45)$$

As stated earlier, l_x, m_x and n_x are the direction cosines of the local x axis with each of the global axes and:

$$l_x = (x_2 - x_1) / L \qquad (6.46a)$$
$$m_x = (y_2 - y_1) / L \qquad (6.46a)$$
$$n_x = (z_2 - z_1) / L \qquad (6.46a)$$

where (x_1,y_1,z_1) and (x_2,y_2,z_2) are the co-ordinates of the nodes at each end of the element and L is the element length.

The problem begins when finding the remaining elements of T. Balfour [BAL86] uses vector cross products to establish the remaining values and his method has been adopted here to give:

$$l_y = -m_x / \sqrt{(1 - n_x^2)} \qquad (6.47a)$$
$$m_y = l_x / \sqrt{(1 - n_x^2)} \qquad (6.47b)$$
$$n_y = 0 \qquad (6.47c)$$

for the direction cosines for the y axis and:

$$l_z = -l_x * n_x / \sqrt{(1 - n_x^2)} \quad (6.48a)$$
$$m_z = -m_x * n_x / \sqrt{(1 - n_x^2)} \quad (6.48b)$$
$$n_z = \sqrt{(1 - n_x^2)} \quad (6.48c)$$

for the direction cosines for the z axis.

Because the direction cosines have now been established in terms of l_x, m_x and n_x the subscript can be dropped to give:

$$R = \begin{bmatrix} l & -m/D & -ln/D \\ m & l/D & -mn/D \\ n & 0 & D \end{bmatrix} \quad (6.49)$$

where:

$$D = \sqrt{(1 - n^2)} \quad (6.50)$$

6.3.3.2 Mass Matrix M

An appropriate mass matrix can be established by lumping equal parts of the total mass to the nodal points. This is the simplest form of a mass matrix and is obtained by placing the point masses m_i at the displacement degrees of freedom. For an element with two degrees of freedom this means:

$$M = \begin{bmatrix} m_i & \\ & m_i \end{bmatrix} \quad (6.51)$$

Alternatively a continuous mass distribution could be used that has the kinetic energy of the element equal to the kinetic energy of the cloth, rather than lumped at the nodes. This would thus result in a full matrix and hence be more accurate. Although this represents the cloth more closely, replacing the continuous mass by a series of concentrated masses helps to simplify the problem.

The mass values have been provided as part of the measured physical property information and are presented in Table 6.3:

Sample	Mass kg/m^2
100% Polyester	1.182E-01
100% Cotton	2.578E-01
100% Wool	2.138E-01
50% Polyester / 50% Cotton	1.164E-01

Table 6.3 - mass properties of sample textiles

6.3.3.3 Damping Matrix C

Damping is a measure of energy dissipation in engineering materials and structures. The dissipation of energy ultimately stops the motion of a structure. Mass and stiffness are inherent characteristics, but for most structures it is extremely difficult to evaluate the damping characteristics. The damping forces may depend on the vibrating system as well as on the forces exterior to it. The selection of realistic damping values for a particular structure is usually based on experimental evidence from the structure or from similar structures [WIL72 and HUR64].

The purpose of the damping matrix C is to approximate the overall energy dissipation during the system response. The draping system considers damping using Rayleigh Damping parameters and the values for C are calculated by:

$$C = \alpha M + \beta K \qquad (6.52)$$

where α and β are usually determined experimentally. Experimental values for this system could not be obtained so an approximation to α and β have been determined. The method used in this system for calculating approximations to α and β is outlined below and is described in further detail by Bathe [BAT82, Ch 9].

Another way of representing equation 6.52 is:

$$\alpha + \beta \omega_i^2 = 2 \omega_i \xi_i \tag{6.53}$$

where ξ is the modal damping parameter and ω is the undamped natural circular frequency of vibration.

The cloth is assumed to be critically damped so taking two arbitrary modal damping parameters, ξ_1 and ξ_2, critical damping at 100% gives $\xi_1 = \xi_2 = 1.0$.

In a system with N degrees of freedom there are potentially N frequency modes. Two different modes, denoted as ω_1 and ω_2, are assigned different modal values of, for example, 0.3 and 0.9 respectively.

Substituting these values into equation 6.53 gives:

$$\alpha + \beta (0.3)^2 = 2 * 0.3 * 1.0 \tag{6.54a}$$
$$\alpha + \beta (0.9)^2 = 2 * 0.9 * 1.0 \tag{6.54b}$$

and this results in $\alpha = 0.45$ and $\beta = 1.67$.

Further work is needed on obtaining more accurate values for α and β but these have been adopted as an approximation.

6.3.4 Applying the Load

As stated earlier each element has 2 nodes. It is possible to apply a load in each of the x, y and z directions on each node and a moment around each of these three axes. This has been shown in Figure 6.3 and can be represented as:

$$P = \begin{bmatrix} p_{x1} \\ p_{y1} \\ p_{z1} \\ m_{x1} \\ m_{y1} \\ m_{z1} \\ \hline p_{x2} \\ p_{y2} \\ p_{z2} \\ m_{x2} \\ m_{y2} \\ m_{z2} \end{bmatrix} \tag{6.55}$$

The gravity load calculation is used to assign values to the loading vector P and is given as follows:

$$\text{load} = \text{mass per unit area} \; * \; \text{acceleration due to gravity} \; * \; \text{area} \tag{6.56}$$

where the mass per unit area has been provided in Table 6.3, the acceleration due to gravity is taken as 9.80665 m s^{-2} and the area is the element-contributing area around the node.

6.3.5 Newmark's Method of Integration

Newmark's method [NEW71] is an implicit time integration method that is unconditionally stable. It assumes the average acceleration over an integration time step is constant. By using initial conditions at the beginning of each time step, displacement and velocity at the end of the time step can be predicted through the constant acceleration formulas.

The relationships are given below and a more detailed discussion on how these are obtained is discussed by Greenough and Robinson [GRE82, Ch 3].

Displacement at the current time step $D_{t+\Delta t}$ is given by:

$$\{C1\,M + C2\,K\}\,D_{t+\Delta t} = C7\,P_{t+\Delta t} + C3\,P_t + C1\,M\,D_t + \frac{1}{\theta}M\,\dot{D}_t + C4\,K\,D_t \quad (6.57)$$

Velocity at the current time step $\dot{D}_{t+\Delta t}$ is given as:

$$\dot{D}_{t+\Delta t} = C5\,\{D_{t+\Delta t} - D_t\} - C6\,\dot{D}_t \quad (6.58)$$

Acceleration at the current time step $\ddot{D}_{t+\Delta t}$ is given as:

$$\ddot{D}_{t+\Delta t} = C5\,\{\dot{D}_{t+\Delta t} - \dot{D}_t\} - C6\,\ddot{D}_t \quad (6.59)$$

where D_t, $\dot{D}_t$ and $\ddot{D}_t$ are the displacement, velocity and acceleration respectively of the previous time step, Δt is the time step difference, and θ may be varied to produce different schemes. Unconditionally stable schemes are formed when θ is chosen in the range $0.5 \leq \theta \leq 1$. Various texts have researched the best value for θ and have suggested the value of $\theta = 0.5$ to provide the most stable scheme [see for example DUN72 and NIC71] so this value has been adopted here. The constants C1 - C7 are given below where α and β are the Rayleigh Damping constants:

$$C1 = \alpha + 1/(\theta\,\Delta t) \quad (6.60a)$$

$$C2 = \beta + \theta\,\Delta t \quad (6.60b)$$

$$C3 = (1 - \theta)\,\Delta t \quad (6.60c)$$

$$C4 = \beta - (1 - \theta)\,\Delta t \quad (6.60d)$$

$$C5 = 1/(\theta\,\Delta t) \quad (6.60e)$$

$$C6 = (1 - \theta)/\theta \quad (6.60f)$$

$$C7 = \theta\,\Delta t \quad (6.60g)$$

The displacements, velocities and accelerations are taken to be zero at the start of the first time step.

6.3.6 Updating Nodal Co-ordinate Values

The final results for the nodal displacements correspond to the forces applied. So there are potentially three linear displacements and three rotations at each end of the element. This has been shown in Figure 6.4 and can be written as:

$$d = \begin{bmatrix} d_{x1} \\ d_{y1} \\ d_{z1} \\ \theta_{x1} \\ \theta_{y1} \\ \theta_{z1} \\ \hline d_{x2} \\ d_{y2} \\ d_{z2} \\ \theta_{x2} \\ \theta_{y2} \\ \theta_{z2} \end{bmatrix} \qquad (6.61)$$

where d_x, d_y and d_z are displacements in the x, y and z directions and θ_x, θ_y and θ_z are the rotations around the x, y and z axes.

6.3.7 Output Data

At the end of the program, control values for the display program are output to the file "coord". Also output are the x, y and z co-ordinates for each node at each time step.

6.4 Conclusion

The draping equations have been refined over the duration of the project. The program started off as a skeleton program in the NAG Library and has been modified to suit the needs of the research.

One main problem with this method is the length of time required for solving the draping equations which not only hinders but actually prevents real-time animation. Ideally real-time animation is required, with the calculations being performed within the time available for each frame of the display. With the current system only real-time playback is possible. With improved hardware and software then the timing problem can be solved. Techniques such as parallel processing and dedicated machines could go a long way to achieving this.

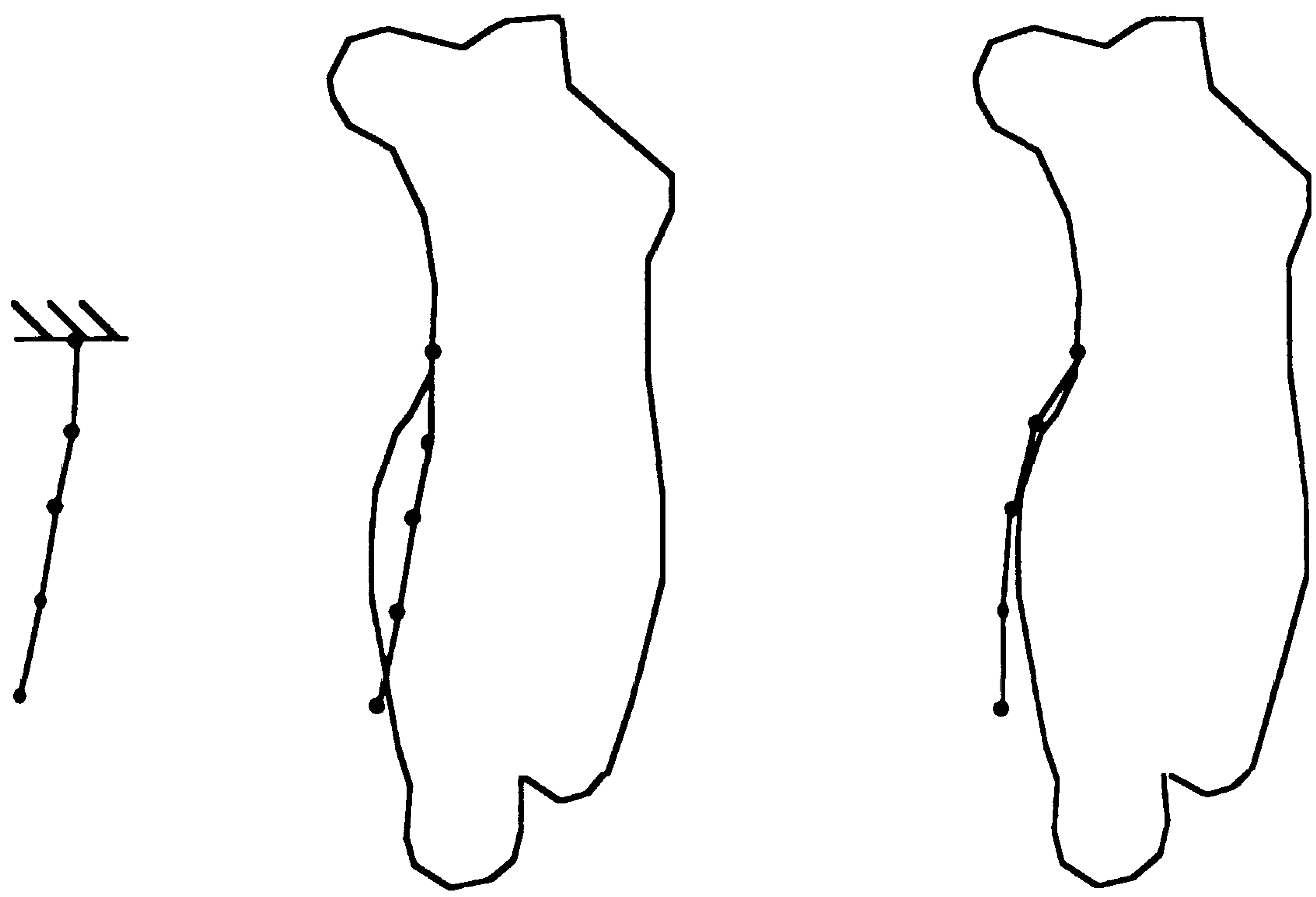
This chapter has covered the main draping calculations required for cloth to drape over a solid. The following chapter expands the DRAPE program to describe the method used for collision detection and response. A program listing of DRAPE and descriptions of the routines and variables used are given in Appendix III. The algorithm described here for drape calculation, including sections on collision detection, has been submitted as a paper (see section headed Publications for full title).

7. COLLISION DETECTION AND RESPONSE

7.1 Introduction

Introducing the body model into the draping system made a great improvement on the visual effect of the simulation. It was seen as very important by the fashion industry to have a body model when considering cloth drape in three dimensions. Incorporating the body model however did increase the overall calculation time for the draping equations because collision detection had become a more complex problem.

It is not enough to just determine that collisions have occurred. An appropriate response must then be carried out so that the realism of the simulation is maintained. Collision detection in this research considers the penetration by the dynamic cloth model with the static human body model. Looking at this in two dimensions, Figure 7.1(a) shows a string restrained at one end during one draping time step. Because there are no other objects involved in this situation, the problem of collisions does not arise.



(a) (b) (c)

Figure 7.1 - (a) free falling string, (b) collision, (c) corrected display

This same string is attached to a body model as shown in Figure 7.1(b). In this case a collision has taken place and this must not be shown in the display. The collision must be detected and a suitable response applied so that the image as shown in Figure 7.1(c) is observed. The response must result in the cloth model remaining outside the solid volume.

One paper outlining various techniques for collision detection and response is by Kamat [KAM93] who describes the technique of step-and-check which has been applied here. This chapter outlines existing techniques for collision detection and response and goes on to explain how this is accomplished for this research.

7.2 Collision Detection

The situation where two or more objects occupy the same volumetric space is known as object interference. In computer simulations of physical objects it is important that object interference is detected otherwise incorrect displays will occur.

The problem of object interference can be divided into the two general cases of static interference and dynamic collision detection. The former detects intersections among objects in fixed positions while the latter detects collisions among objects moving along specified trajectories.

Dynamic collision detection can be solved using several methods. The simplest method is to perform static interference testing at each time step and this is known as the step-and-check method. Another common method is to compute the volume swept out by objects over their trajectories. If these swept volumes intersect, then a collision between the objects has occurred.

Cameron [CAM90] extended this last method into a 4D intersection detection problem over space time. The objects each sweep out a prism in 3D space that defines the time dimension. If the prisms intersect then the objects have collided.

The calculation time involved in collision detection can seriously hinder the animation performance and can make real-time animation impossible. The step-and-check method can be very time consuming and this is a major drawback because collision detection is carried out at every time step in the animation. The swept solid method however needs only a single static interference check to be performed for the whole problem.

Collision detection of one object can be carried out by checking for simple vertex penetration of the faces of the other object. This can be enough for some systems but in more general cases collision detection must be performed between both objects' faces.

In this more extensive method of collision detection of objects, each face of one object needs to be tested against all the faces of the other object. This is a two step process where all the edges of each face for one object are tested for intersection with all the faces in the other. The second part of the check involves all the edges of all the faces in the second object being tested for intersection with all the faces in the first object.

One of the drawbacks of this method, apart from the potentially extensive amount of computation time involved, is that if one object is completely contained within the other object without any faces touching, then no collision will be detected. This means that an additional containment test must be performed. This is where a single point on the smaller object is tested for containment within the larger object.

One way of reducing the overall calculation time involves limiting the detection process to those time steps where objects are close together and there is a likelihood of objects colliding. Methods adopted here include bounding sphere and bounding box. The objects are each completely contained within the smallest sphere or box volume and it is a simple test to see if penetration of this boundary has occurred. These checks however test for the necessity of a full interference test and are not sufficient on their own.

Ganter and Isarankura [GAN93] use a space partitioning technique to reduce collision detection time by subdividing the bounding boxes into a set of partitions. All face testing can be confined to the local region of overlap between the two objects by using these partitions and so overall calculation time is reduced.

A similar method is described by Garcia Alonso *et al.* [GAR94]. Each object is placed in a container which is the smallest box able to hold the object. The faces of this container are parallel to the objects' local reference frame. A minimax test is applied which computes one box for each container. The sides of this second box are parallel to the global reference frame. A test is then carried out for interference of the containers. It this test proves positive, then a further test is carried out for interference among voxels which are small subdivisions of the container. If interference is again detected, then an interference check is run between pairs of facets within this area. As each test is performed and a positive result given, the subsequent test needs to be run only on the smaller region which has been found to be positive.

7.3 Development of a Collision Detection Method

Swept object collision detection methods are best for automation models such as cars along a track or robots, where the path can be defined prior to the motion. This is why the particular method adopted for this research is static interference testing at each time step. Checks for collision detection must be carried out at the end of each time step in the animation once the forces have been applied to the cloth mesh. If a collision is discovered, then an appropriate response is calculated and the time step is rerun with the updated values.

In theory, collision detection means that each node and element in the cloth mesh must be compared to the whole body model to see if a collision has occurred. However some limits can be applied to these checks as it can be unreasonable, due to the calculation time involved, to check every point on the cloth against every body face and often this is unnecessary. As well as the ability of expressing limits for the checks which will reduce the time involved, a mapping method has been developed in order to simplify the actual check for collision.

This method maps the irregular-shaped body model onto a regular-shaped geometrical shape, in this case a cylinder. By mapping in this way it is then a simple check to determine whether any point on the cloth mesh intersects with the mapped cylinder. If an object can be mapped topologically onto a regular shape, i.e. a cylinder, then it follows that a point is internal to the object if and only if the transformed point is internal to the cylinder.

Collision detection is activated at the end of each time step once the draping calculations have been performed and the displacements of the cloth nodes determined. Each node in the cloth model is mapped around the body cylinder depending on the distances moved by the closest body model nodes. The check for collision is simply whether each particular cloth node is within the volume of the cylinder. If the node is outside the volume of the cylinder then collision at that point has not occurred and a response does not need to be calculated.

Looking at the problem in two dimensions, Figure 7.2 shows a segment from an irregular-shaped area mapped to a regular segment from a circle. O and O' are the centre points for the irregular shape and the circle respectively.

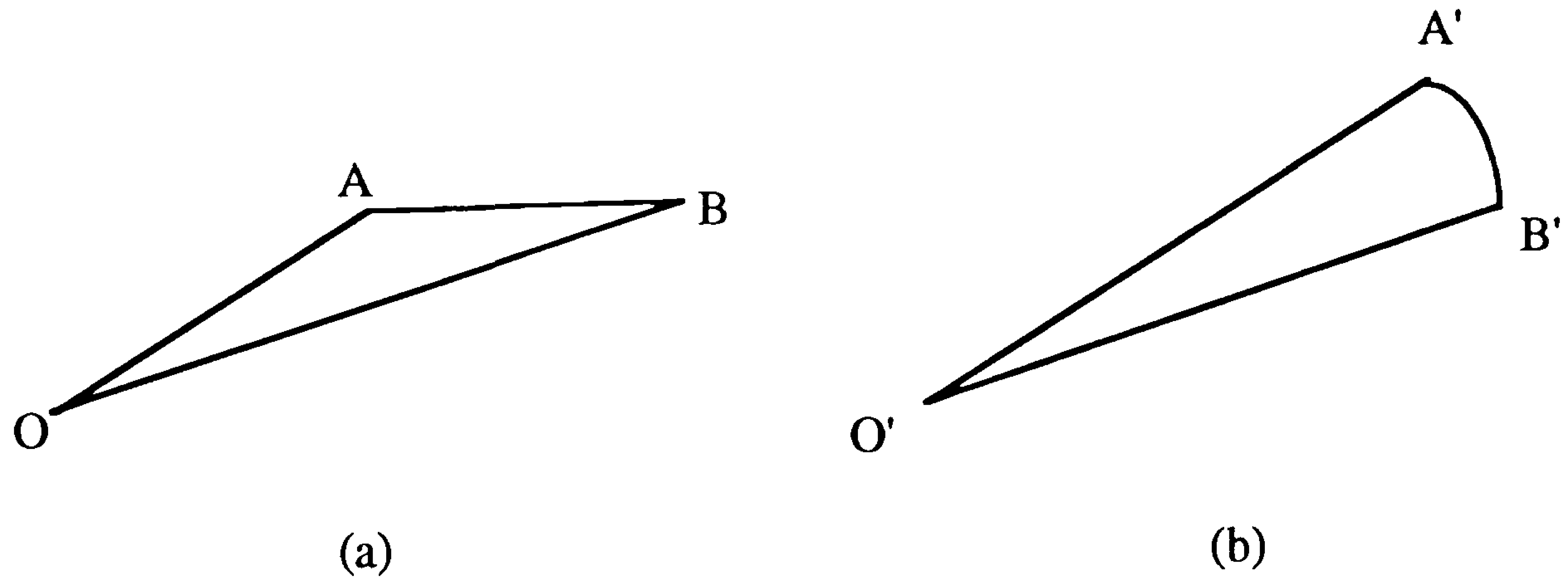


Figure 7.2 - (a) irregular to (b) regular 2D segment mapping

In the example shown in Figure 7.2(a) OA ≠ OB. In the mapped segment in Figure 7.2(b) O'A' = O'B', i.e. they are radii, and angle O' = angle O.

Figure 7.3 shows a node from a colliding object which has penetrated the irregular segment area.

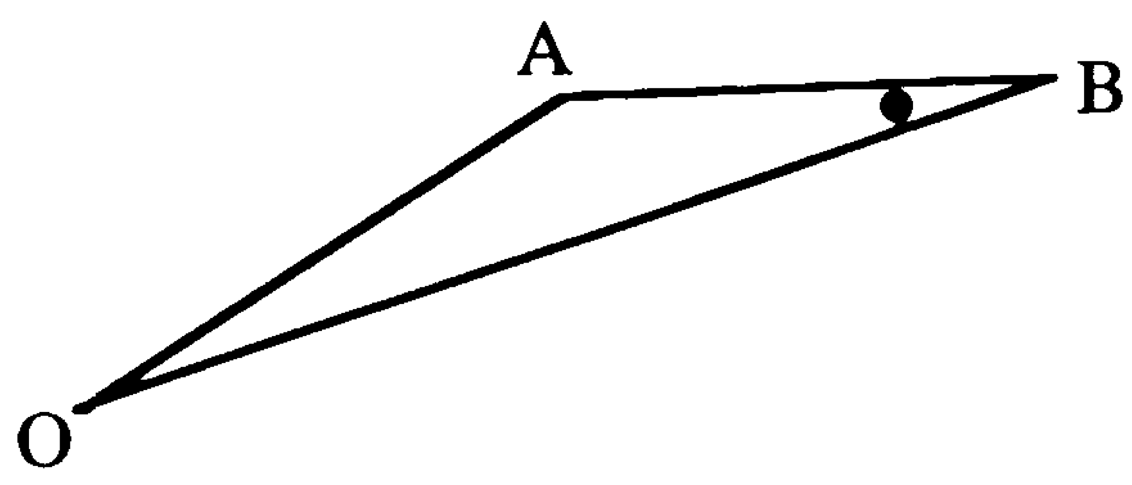


Figure 7.3 - node penetration of solid segment

This collision is easy to see visually but the detection must be made mathematically. The general method would be to calculate the equation of AB and establish on which side the node lies. Using the mapping method however, the check is simplified as shown in Figure 7.4.

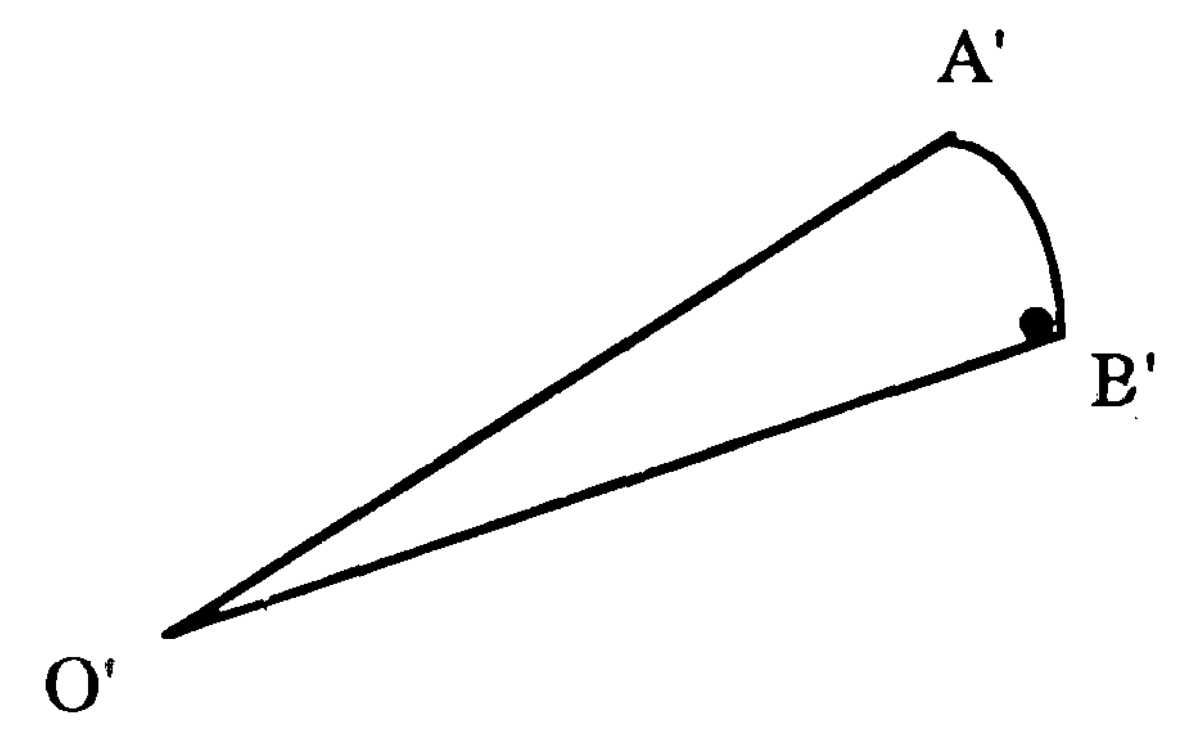


Figure 7.4 - node penetration of mapped segment

The check here is simply whether the node's distance from the centre of the circle is greater or less than the radius.

The problem of collision detection is made more complex when checking for collisions along elements as shown in Figure 7.5.

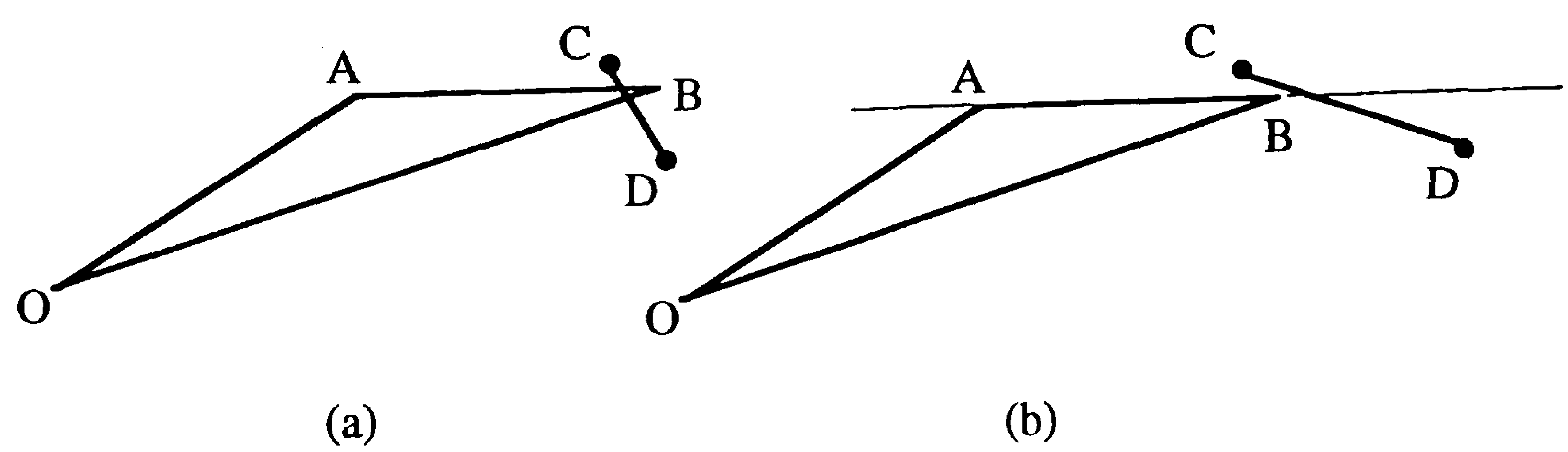


Figure 7.5 - (a) collision of element and (b) no collision for irregular segments

Figure 7.5(a) shows no collision at the points, but collision occurring along the connecting element. A check can be made for line equation intersections between

AB on the segment and CD on the colliding object. Unfortunately, as shown by Figure 7.5(b), a collision could also be flagged for non-colliding elements as line intersections have still occurred. Further checks would have to be made to establish whether a collision had actually taken place.

The mapping method solves this problem as the elements have a non-affine mapping onto the circle, i.e. a straight element is mapped to a curved element, as shown in Figure 7.6.

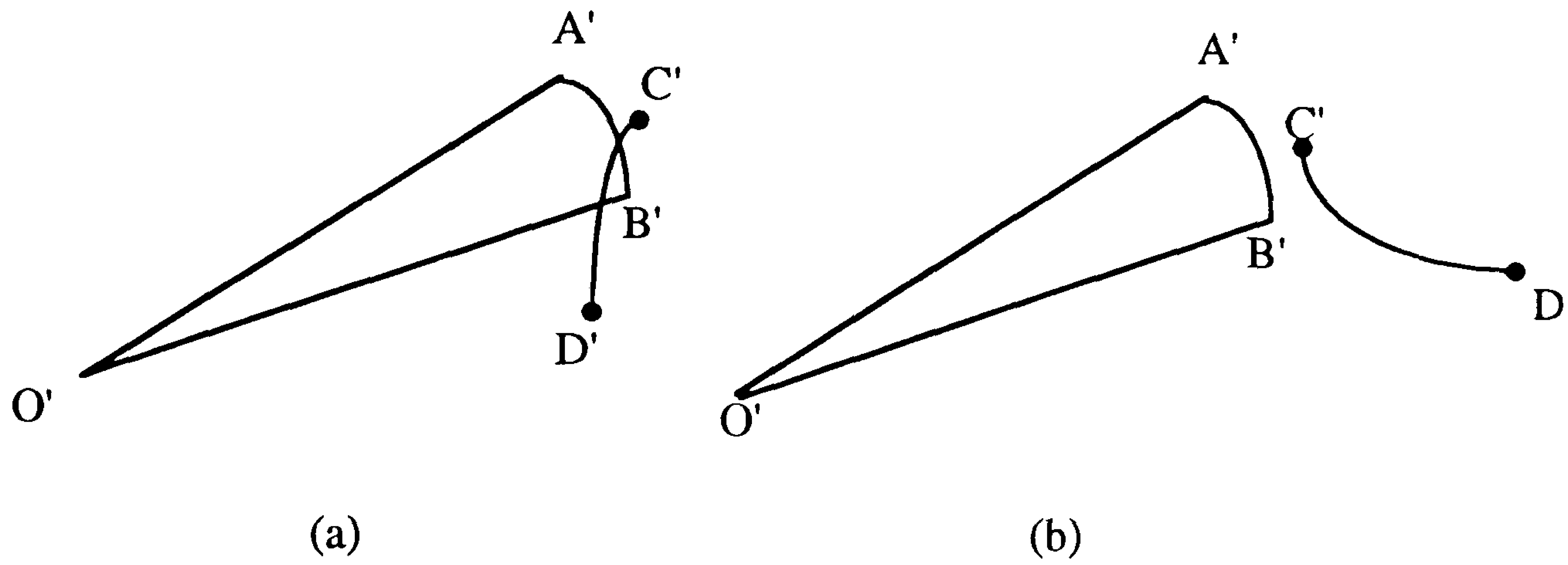


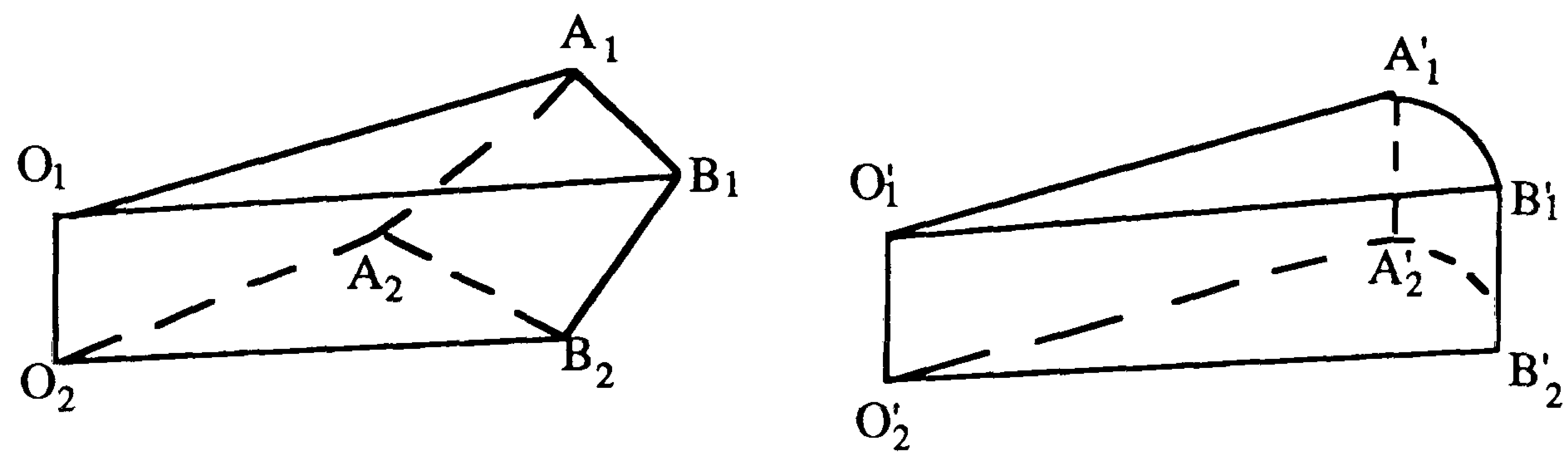
Figure 7.6 - (a) collision of element and (b) no collision for mapped segment

To establish if collisions have occurred in the mapping method, distances of points along the element are checked against the circle radius.

The mapping procedure shows even more benefits in three dimensions. Figure 7.7 shows an irregular to regular 3D mapped segment.

In Figure 7.7(a) the segment edges are not necessarily equal, i.e. $O_1A_1 \neq O_1B_1 \neq O_2A_2 \neq O_2B_2$. The angles O_1 and O_2 are also not necessarily the same.

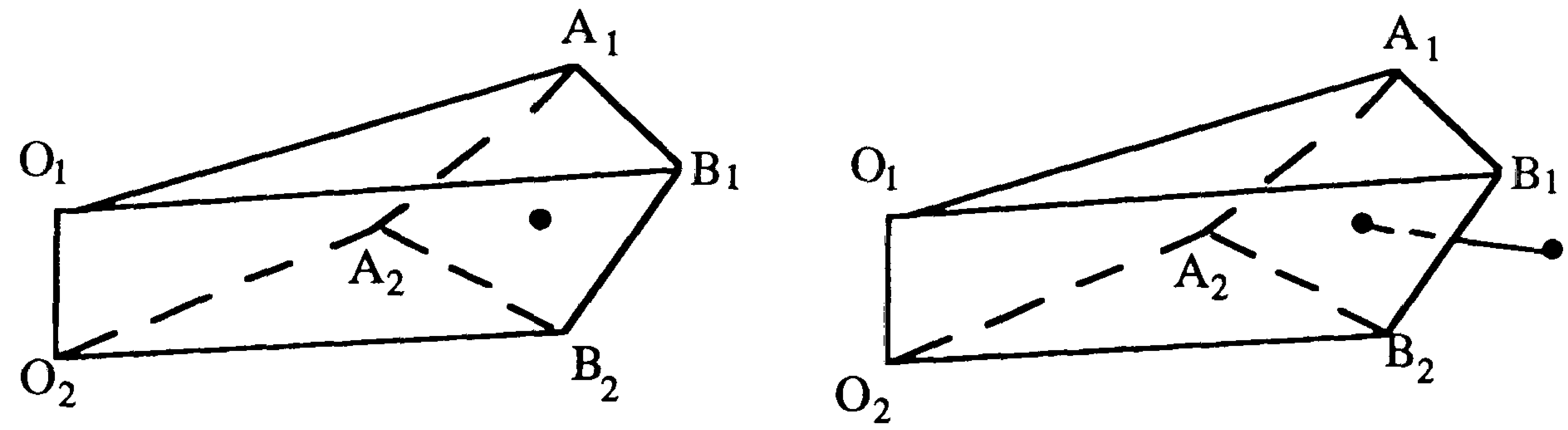
In Figure 7.7(b) however, the edges are all equal as they are all radii of the same circle, i.e. $O'_1A'_1 = O'_1B'_1 = O'_2A'_2 = O'_2B'_2$. Also angle O'_1 = angle O_1 and angle O'_2 = angle O_2.



(a) (b)

Figure 7.7 - (a) irregular to (b) regular 3D segment mapping

Figure 7.8 demonstrates the problems involved in 3D collision detection. The node in Figure 7.8(a) does not line up conveniently with the solid in a plane. To determine if a collision has occurred, equations must be established for the face of the segment.



(a) (b)

Figure 7.8 - (a) node and (b) element detection in 3D irregular segments

In the problem of colliding elements as shown in Figure 7.8(b), intersections can be determined between the element and the segment face, but again this can lead to falsely detected collisions as in the 2D problem. Further checks would have to be made to ensure collisions have been detected correctly.

Figure 7.9 shows the mapped solution to this. The check for collisions is the same as for the 2D problem, i.e. the distance of the node or element from the centre of the cylinder is compared with the radius.

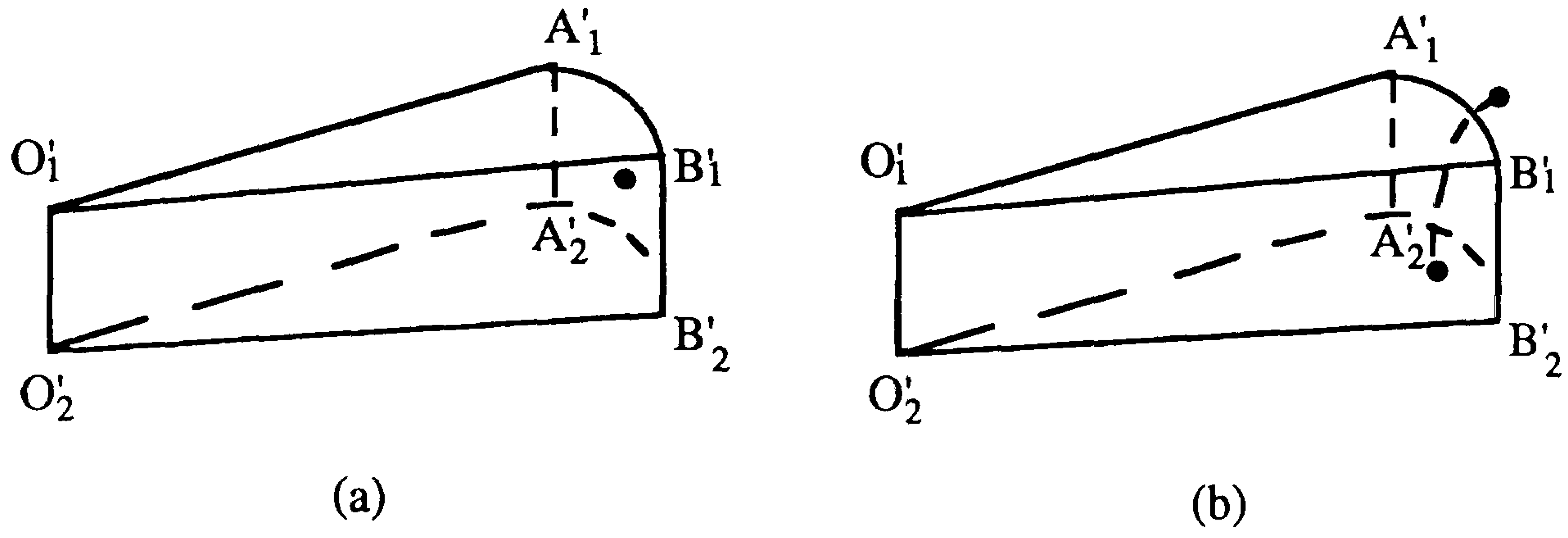


Figure 7.9 - (a) node and (b) element detection in 3D regular segments

The method introduced above has been developed for the research described in this thesis. It is outlined in Figure 7.10 and described in more detail below.

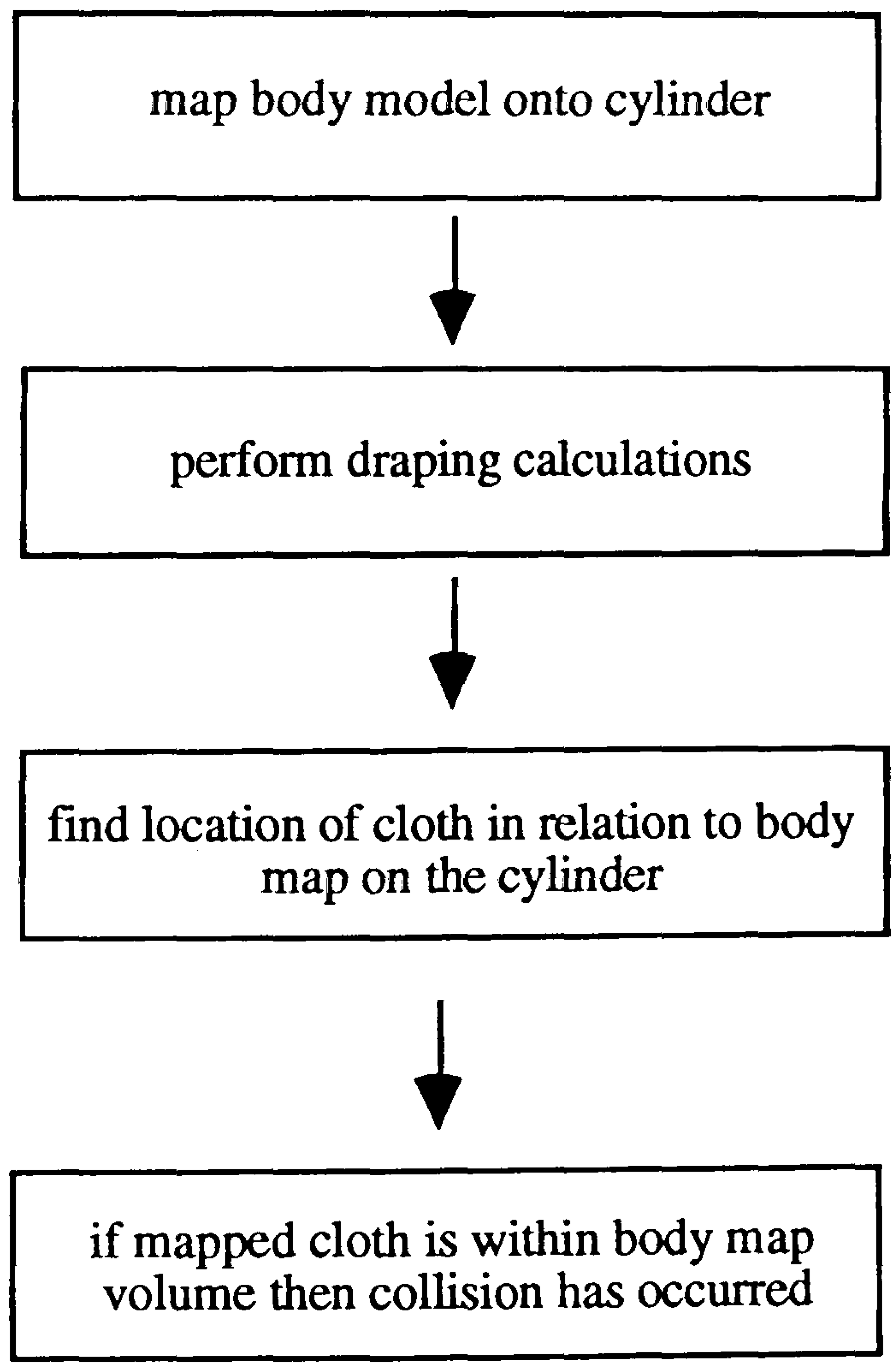


Figure 7.10 - collision detection

7.3.1 Body Model Mapping

The body model is mapped onto a cylinder at the beginning of the draping program. First a cylinder radius is determined and this is calculated to be just slightly bigger than the longest distance of all the body nodes to the centre of the model. Then the mapped location of each of the body nodes is determined as shown in Figure 7.11.

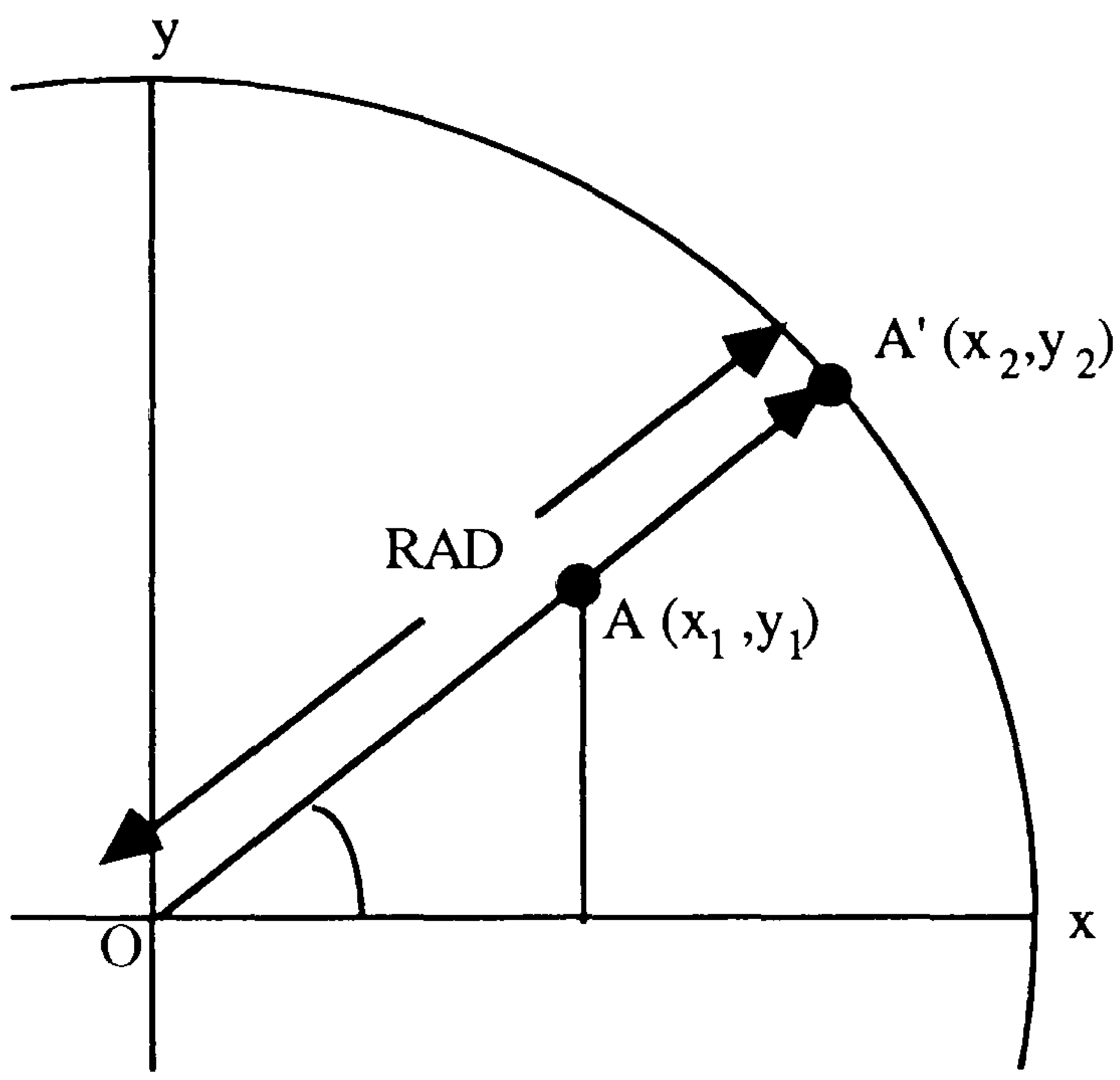


Figure 7.11 - calculating new location for body node mapping

In Figure 7.11 A (x_1,y_1) is the actual location of the body node on the body model and A' (x_2,y_2) is the location of the same node mapped onto the cylinder. The mapping takes place along the same x-y plane so the z co-ordinate remains the same. The angle between the body node and the x-axis is calculated as follows:

$$\text{angle } O = \tan^{-1} (y_1 / x_1) \qquad (7.1)$$

and the new values for the body node are:

$$x_2 = \text{RAD} \cos (O) \qquad (7.2a)$$
$$y_2 = \text{RAD} \sin (O) \qquad (7.2b)$$

where RAD is the radius of the cylinder.

These calculations are carried out for every body node and the resulting values are stored. Figure 7.12 shows an example of the mapping results for one slice of body data.

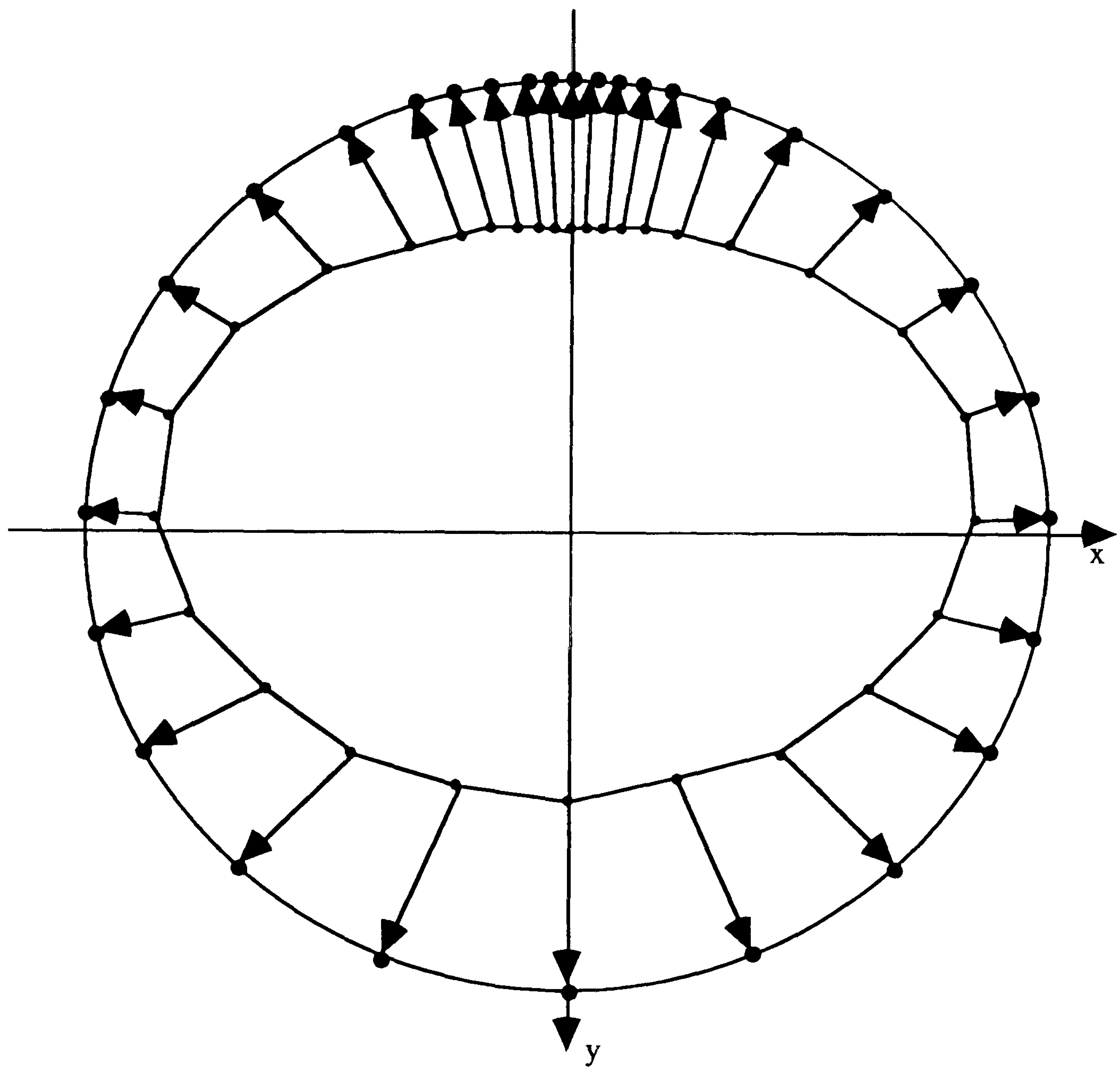


Figure 7.12 - body mapping onto the cylinder

When all the nodes have been mapped the result is a listing of body model nodes which if displayed would show a regular cylinder as in the example shown in Figure 7.13.

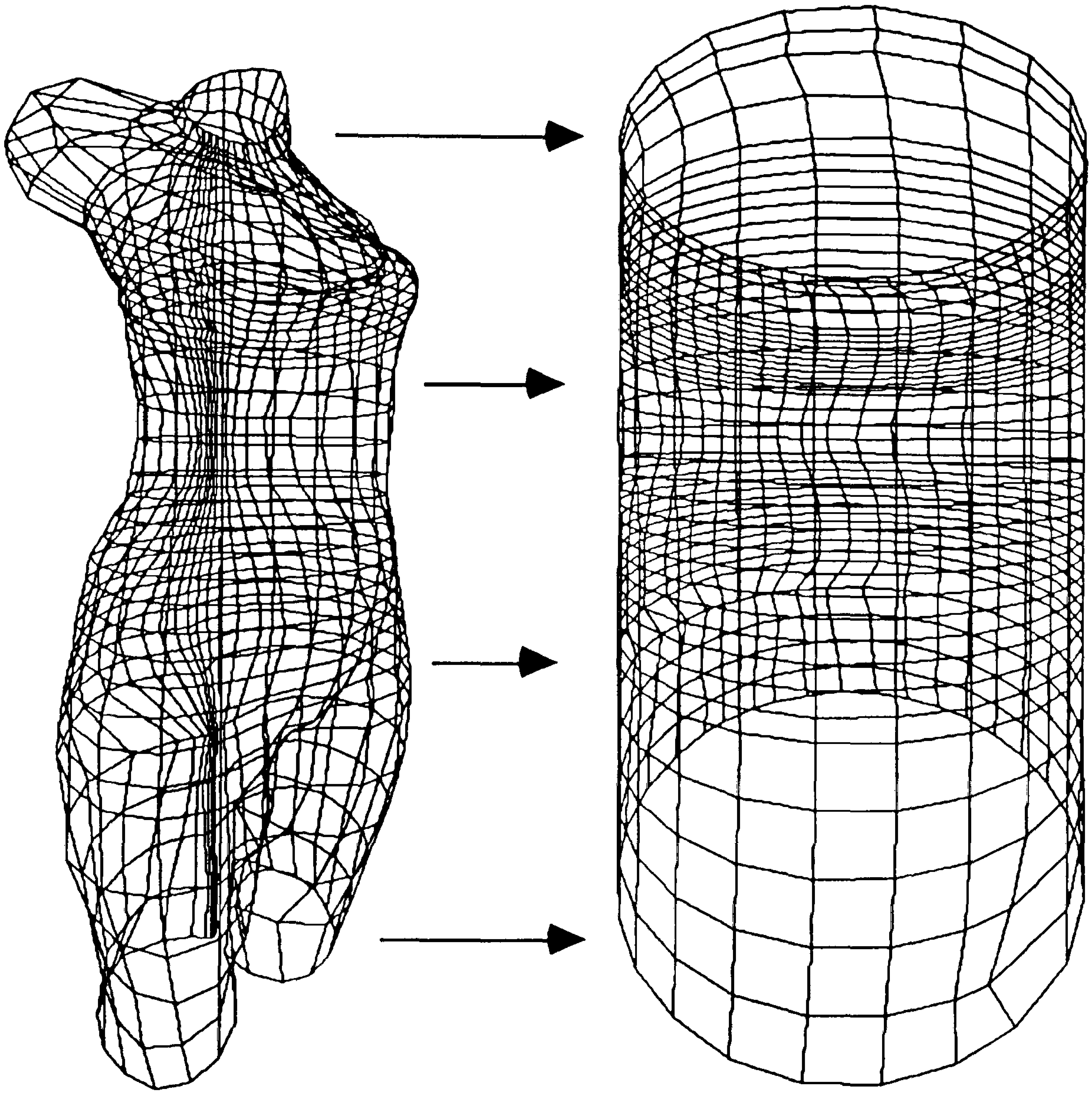


Figure 7.13 - body model mapping to cylinder

The cloth mesh mapping is then determined around this body model cylinder. The new locations for the mesh rely on the displacements between the original location of the body nodes and their cylinder mapped locations.

7.3.2 Cloth Model Mapping

The location of each unrestrained node on the cloth mesh must be found in relation to its four closest body node neighbours. This cloth mesh mapping is carried out in three stages as outlined in Figure 7.14.

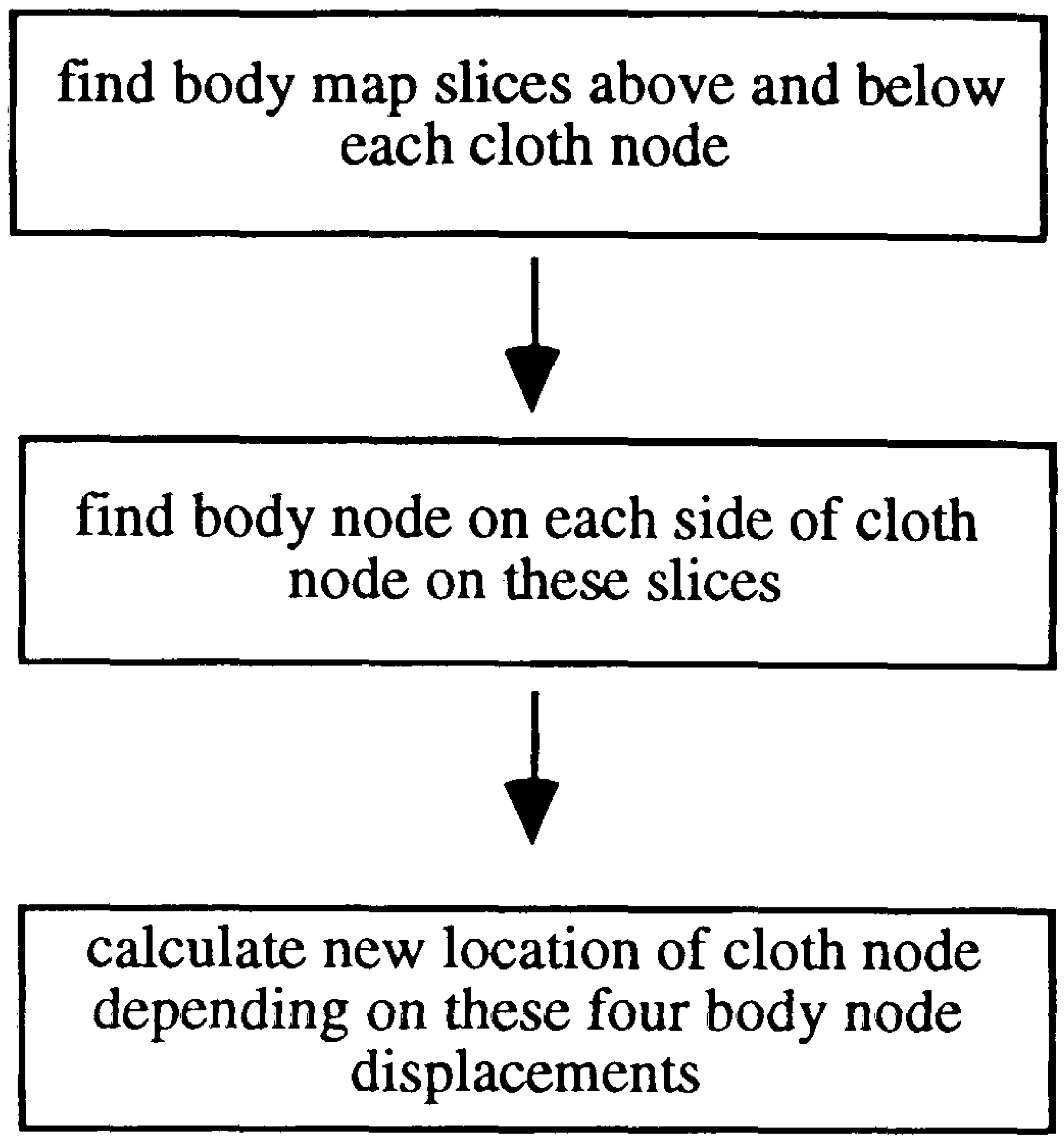


Figure 7.14 - mapping cloth mesh nodes around body cylinder

First the relevant body slices are determined which are the body slices just above and just below the cloth node. Then the body nodes on either side of the cloth node are determined for both of these slices. Finally the new location of the cloth node can be calculated depending on the distances moved by these four body model nodes onto the cylinder and their respective distances from the cloth node.

7.3.2.1 Finding the Closest Body Slices

The first step in the mapping for the cloth model is to determine the relevant pair of body data slices which lie above and below the cloth node. The LASS body data as mentioned earlier is provided as slices of data. It is a simple comparison check on the z co-ordinate to establish between which two slices of body data a particular cloth node lies and this is shown in Figure 7.15.

The example shows the cloth node is between slices 8 and 9. Once the relevant body data slices have been found it means that the range of nodes for the check can be limited to these two slices of data rather than the whole body model set of data.

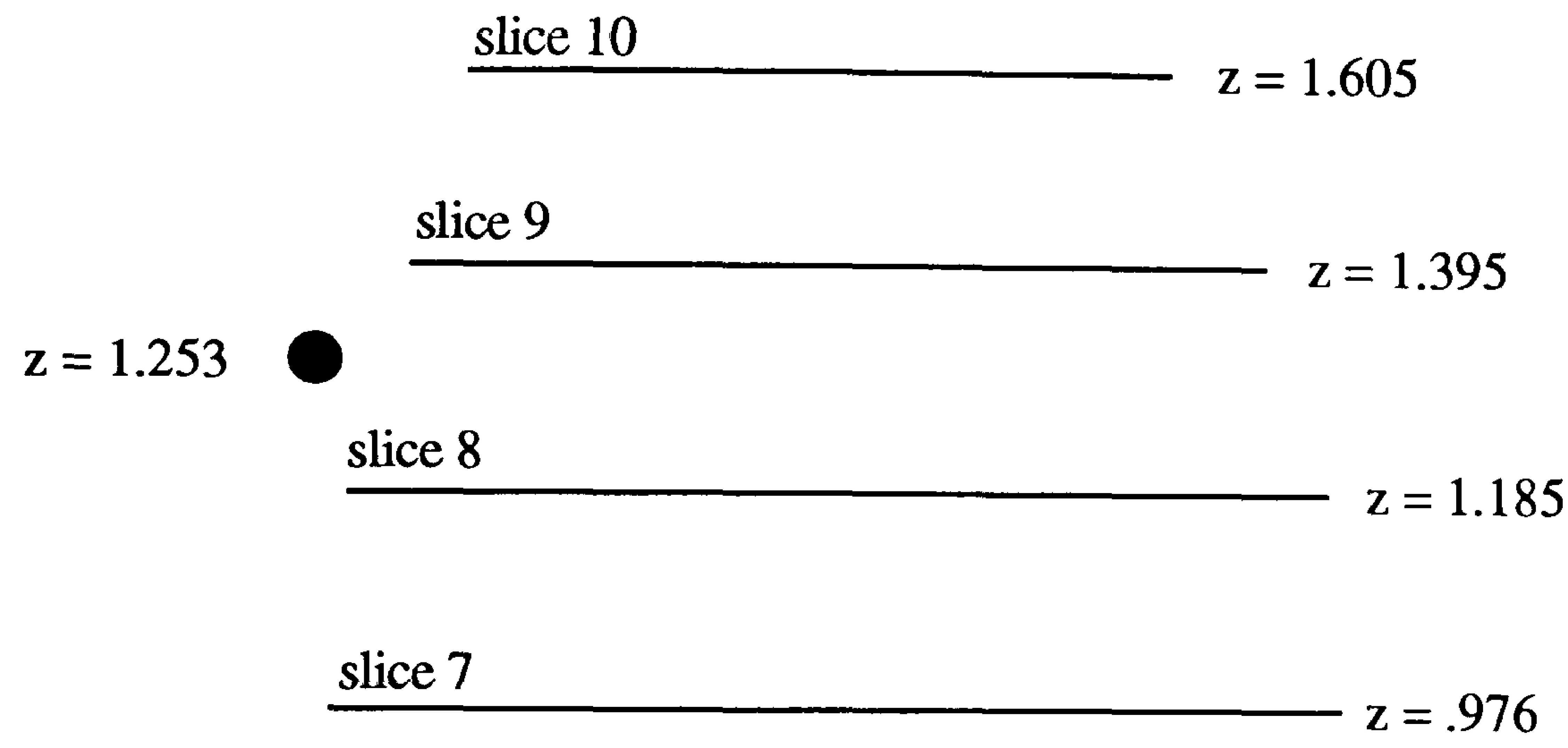


Figure 7.15 - position of cloth node between two body data slices

7.3.2.2 Finding the Four Closest Body Nodes

The next stage is to determine the location of the cloth node in relation to the body. This is performed first for the body slice above the cloth node and is achieved by finding the pair of consecutive body nodes lying either side of the cloth node as shown in Figure 7.16.

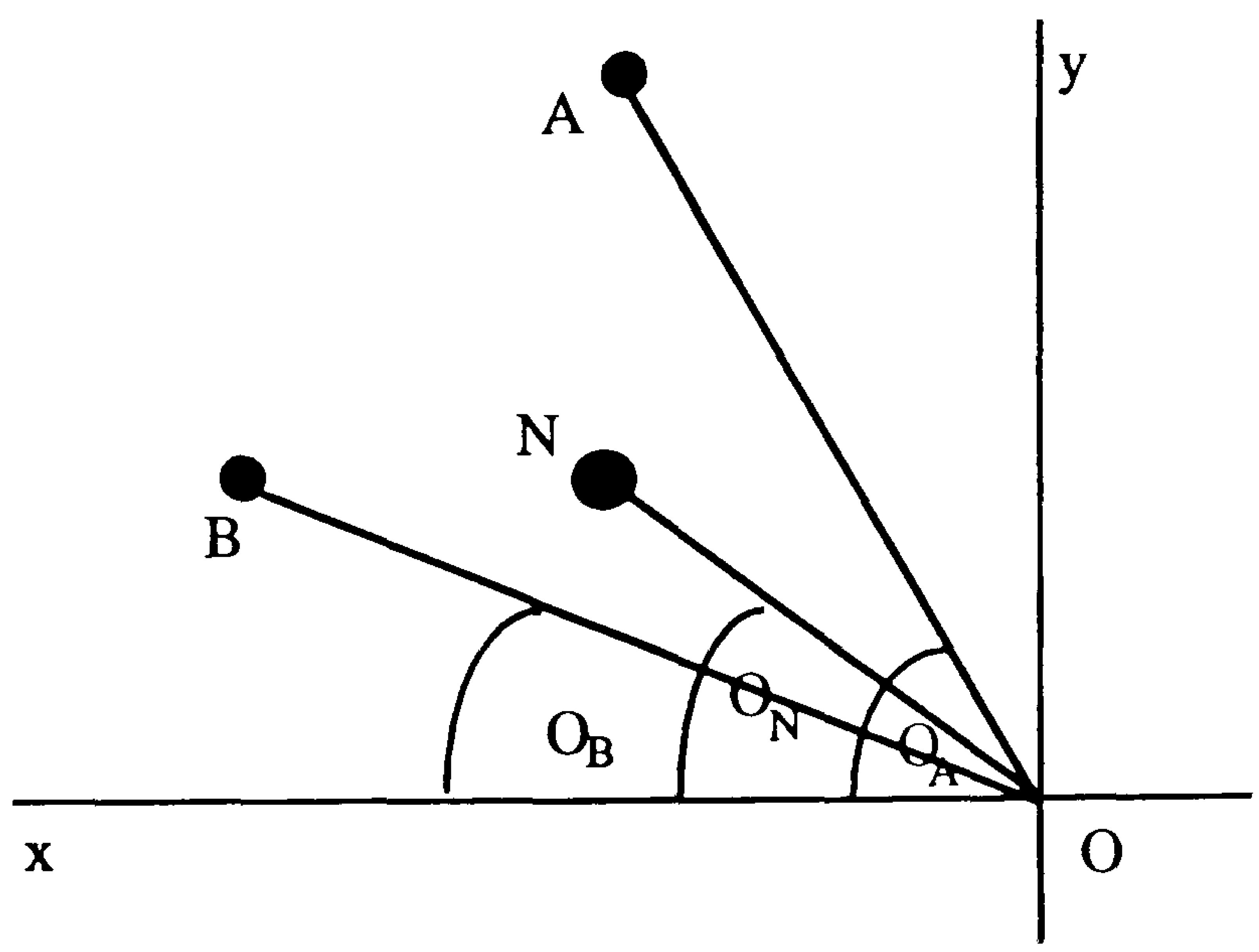


Figure 7.16 - position of cloth node between two body nodes

Again this is a fairly straightforward check on the angles formed with the x axis by each pair of consecutive body model nodes and the cloth node. The correct pairing is where one angle is greater than O_N and the other is less than O_N. In the example shown in Figure 7.16, A and B are body nodes on either side of N which is the current position of the cloth mesh node and O_A, O_B and O_N are the respective angles formed with the x axis. This calculation is performed only in the x-y plane as the values for the z direction are not necessary.

When these two nodes have been determined for the higher body slice then exactly the same operation is carried out for the lower body slice to find the remaining two body node neighbours. The result is the discovery of the four closest body node neighbours around the cloth node.

7.3.2.3 Calculating the Mapped Location

The next stage is to calculate the distances moved by these four nodes from their original body mesh onto the mapped body cylinder. The distance to be moved by the cloth node is a combination of all these four displacements weighted depending on the respective closeness of the four body nodes to the cloth node.

Figure 7.17 shows this for the upper body slice.

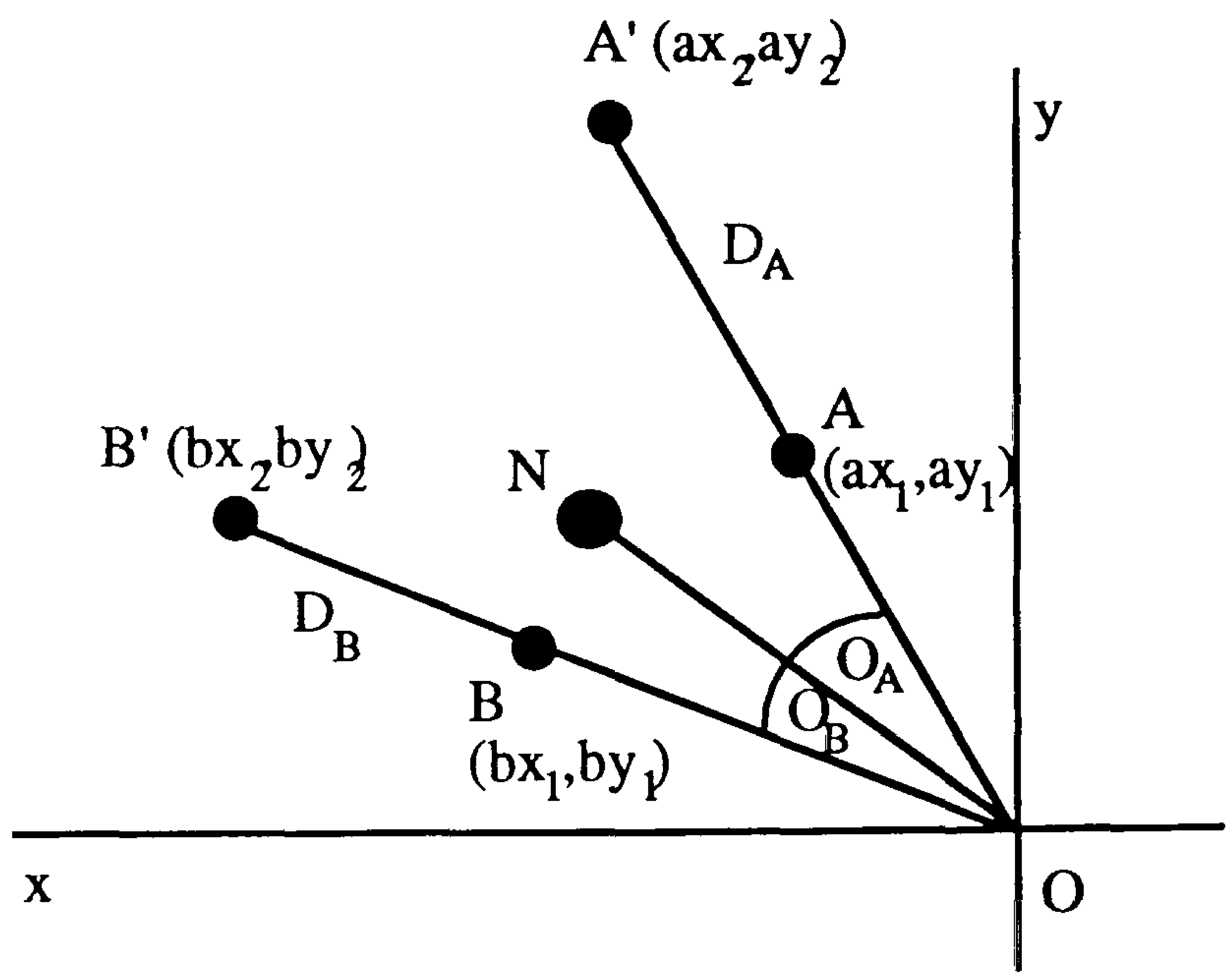


Figure 7.17 - distances moved by body model nodes onto cylinder mapping

In this example A (ax_1,ay_1) and B (bx_1,by_1) are the original locations of the upper body slice nodes in the body model and A' (ax_2,ay_2) and B' (bx_2,by_2) are the locations of these nodes mapped onto the cylinder. Their respective displacements are D_A and D_B, and the size of the angles formed with the cloth node N are O_A and O_B. The actual displacement D_U for this upper slice is calculated as follows:

$$D_U = \frac{O_B}{O_A + O_B} * D_A + \frac{O_A}{O_A + O_B} * D_B \qquad (7.3)$$

This operation can be repeated for the lower slice to discover D_L:

$$D_L = \frac{O_D}{O_C + O_D} * D_C + \frac{O_C}{O_C + O_D} * D_D \qquad (7.4)$$

where D_C and D_D are the displacements of the two body nodes on the lower slice and their respective angles with the cloth node are O_C and O_D. To find the actual displacement D_N of the cloth node the following calculation is performed:

$$D_N = \frac{D_E}{D_E + D_F} * D_U + \frac{D_F}{D_E + D_F} * D_L \qquad (7.5)$$

where D_E is the height difference between the cloth node and the lower slice and D_F is the height difference between the cloth node and the upper slice. The new location of the cloth node mapped around the body cylinder is calculated by adding the displacement D_N to the current location of the cloth node. The check for collision is now very straightforward. It is simply a check on the distance of the mapped cloth node from the centre of the cylinder against the radius .

7.3.3 Checking for Element Collisions

If the mapped cloth node distance is greater than the radius then no collision has been detected. For some applications a simple node by node check is adequate. However for this particular system a further check had to be carried out in the form of an element check as it is possible for the cloth elements to cross through the solid while their respective end nodes are outside the solid volume. If such a situation occurs then the collision would not be detected by the node check alone and this would be seen in the display where the cloth skirt penetrates the body model volume. For this reason the element check has been introduced.

The element check uses the same method as the node check. Groups of x-y-z co-ordinates are determined along the cloth element using the equation of a line:

$$x = (1 - t) N_x + t M_x \quad (7.6a)$$
$$y = (1 - t) N_y + t M_y \quad (7.6b)$$
$$z = (1 - t) N_z + t M_z \quad (7.6c)$$

where N and M are the cloth nodes at either end of the element and the value of t is determined by incrementing evenly from 0 to 1 depending on the level of accuracy required. For example if four checks were to be made along an element then t would be given the values 0.2, 0.4, 0.6 and 0.8.

Figure 7.18 shows an example of this for the body model (the element length has been exaggerated for a clearer description). These four additional x-y-z groupings are mapped onto the body cylinder using exactly the same method as described earlier for the node mapping and comparisons are made with the cylinder radius to see if collisions have occurred along the element length.

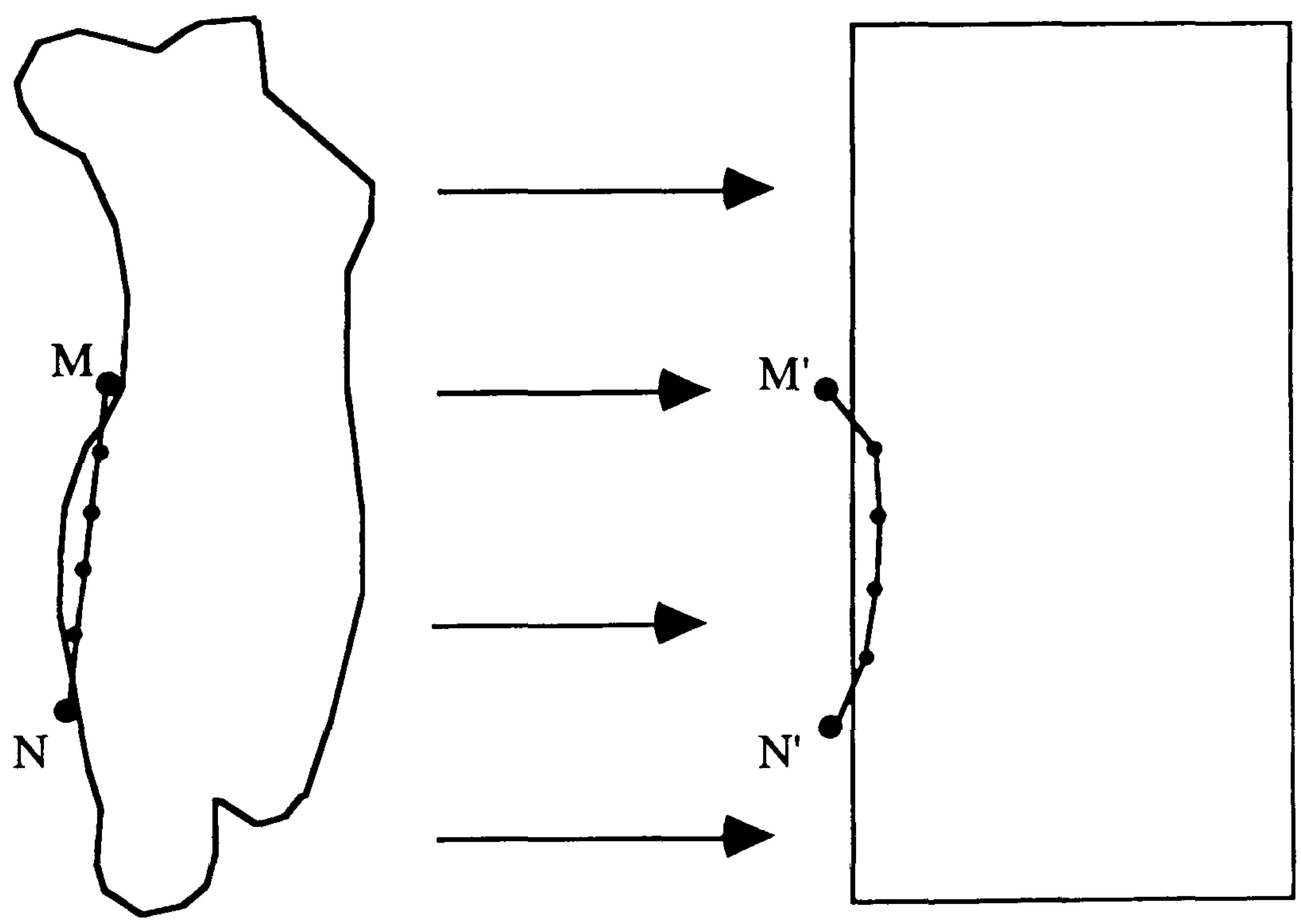


Figure 7.18 - element collision checking

7.4 Collision Response

Checks for collision detection need to be carried out at each stage in the animation and a suitable response applied to ensure that the cloth does not penetrate the solid volume during the animation.

Using the step-and-check method [KAM93] a time step is performed and checks for penetrating nodes are carried out at the end of that time step. If the cloth mesh does enter this volume then some method must be used to move it back out again while still retaining the realism and overall topology of the cloth structure. This means that the time step must be performed again, taking into account the values calculated during the collision detection.

Collision response has been approached in several ways. Scanlon [SCA90] simply moves any mass points which fall within the volume of the solid onto the nearest point outside the solid. Although this is a straightforward calculation and economical in processing time, this method can result in an alteration to the overall cloth geometry.

One method commonly adopted for collision response is the temporary insertion of a stiff spring at the closest points between two colliding objects. The spring force is applied equally in opposite directions to the paths of the objects and should be big enough to push these objects apart. This method has been described by Moore and Wilhelms [MOO88]. In a similar way Aono [AON90] activates a reaction force vector from the solid at the next time step in the direction opposite to that of the wrinkling cloth and another similar method is applied by Ohte [OHT73] where frictional forces are also considered.

An alternative way of looking at collision detection and response is to consider the problem as collision avoidance. If the collision can be anticipated then feedback may be provided and this can be used to prevent the collision taking place. This prevention can occur either immediately or by back-stepping. One way of doing this is to build force fields around the objects which repel contact between them. These fields can be graduated in relation to the distance from interference so that gentle deceleration or diversion can be implemented.

Lafleur *et al.* [LAF91] adopted this method and created a thin force field around their solid surfaces which acts as a shield rejecting points on the cloth mesh which come too close. The force field is divided into small cells and when a cloth mesh point enters one of these cells then a repulsive force is applied. The direction and magnitude of the force depends on the velocities, normals and distances between the point and the surface.

Because the model described in this thesis is a physical one, where forces are applied to produce movement, then the logical response to penetrating cloth nodes is to apply enough force to ensure that these nodes meet the body model but do not penetrate it. This also gives a more realistic result, rather than just moving the nodes to a point on the body model, and so this method has been adopted for the collision response.

7.5 Development of a Collision Response Method

When a collision has been detected during a time step then the first thing to establish is where the cloth node or element actually touches the body model boundary. The next stage is to recalculate the time interval so that when the time step is rerun, the force applied is enough to push the cloth onto the solid but not to penetrate it. The method described below was initially used for collision response within the draping system.

7.5.1 Finding the Contact Location

Figure 7.19 shows the status of a cloth element at two sequential time steps (again this diagram is exaggerated for clarity). The nodes M_1 (mx_1,my_1,mz_1) and N_1 (nx_1,ny_1,nz_1) are the locations of the cloth element nodes at the current time step, and M_0 (mx_0,my_0,mz_0) and N_0 (nx_0,ny_0,nz_0) are the locations of the element nodes at the previous time step.

This leaves M (mx,my,mz) and N (nx,ny,nz) as the locations of the nodes at the point where the element meets the body model during the time step. It is the co-ordinates of M and N which must be found in order to determine the correct collision response.

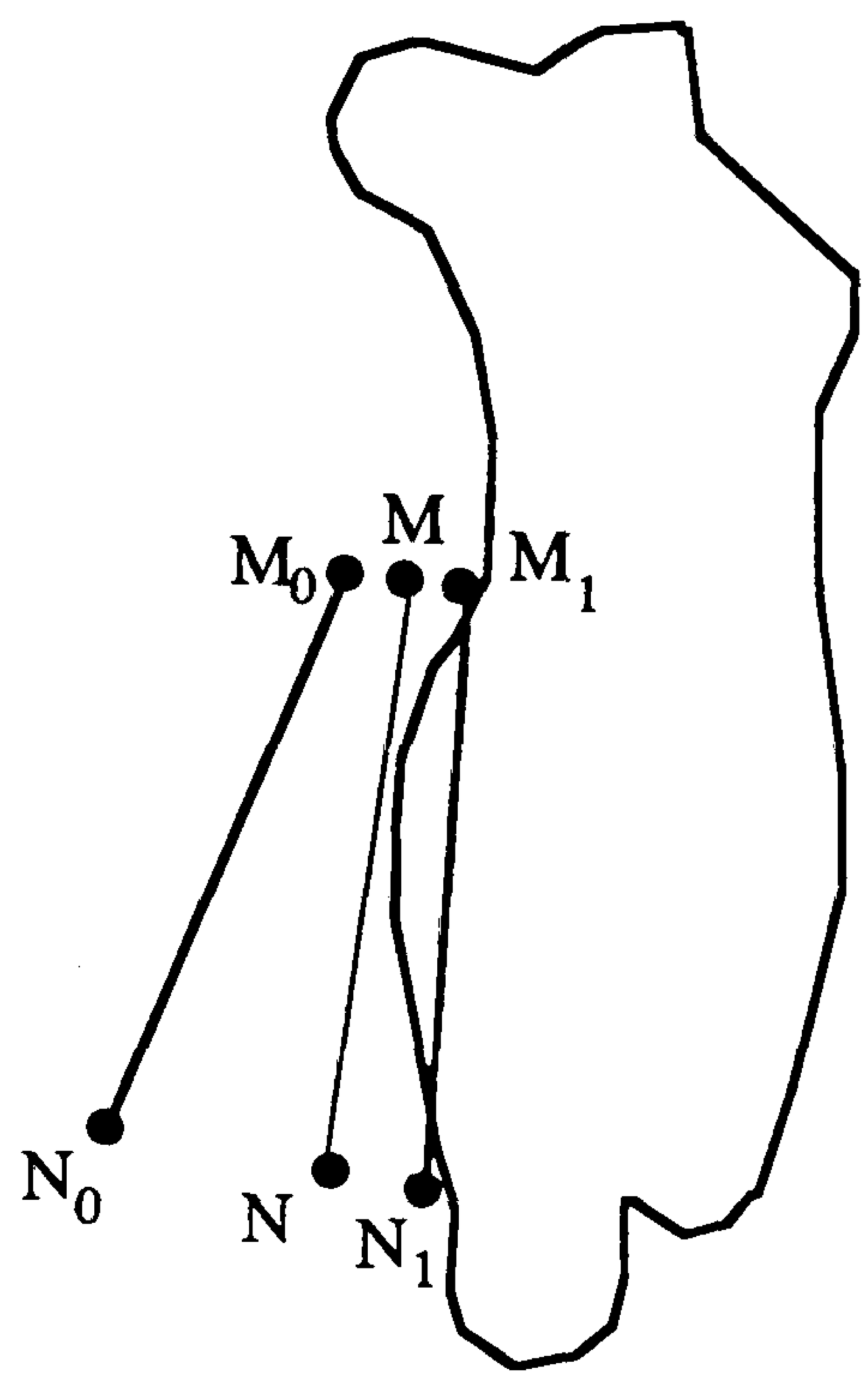


Figure 7.19 - collision point of element between two time steps

The co-ordinates for M and N are easy to determine. Groupings of x-y-z co-ordinates are calculated at intervals between M_1 and M_0 and between N_1 and N_0 using the same line equation method as established in equations 7.6. Here however a binary chop method is used. The method starts with the calculation of co-ordinates halfway between the two time step locations for the element using the following equations where t = 0.5:

$$mx = (1-t)\ mx_1 + t\ mx_0 \tag{7.7a}$$
$$my = (1-t)\ my_1 + t\ my_0 \tag{7.7b}$$
$$mz = (1-t)\ mz_1 + t\ mz_0 \tag{7.7c}$$

$$nx = (1-t)\ nx_1 + t\ nx_0 \tag{7.7d}$$
$$ny = (1-t)\ ny_1 + t\ ny_0 \tag{7.7e}$$
$$nz = (1-t)\ nz_1 + t\ nz_0 \tag{7.7f}$$

When these values have been established then a check is run along the whole length of the new element by mapping to the cylinder. The three possible outcomes are (i) a collision is still detected, (ii) a collision is not detected and (iii) the element location is at a tangent to the model.

If a collision has been detected, then another binary chop is taken halfway between this element location and the previous time step location. If no collision has been detected then the next binary chop is halfway between this element location and the current time step location. If the location results in a tangent then the checks can stop.

The line equations established in equations 7.7 are compared with the body cylinder using the mapping method until an element location is found that is at a tangent to the cylinder and it is this line which provides the correct location for the element and the co-ordinate values for M and N.

7.5.2 Calculating the Time Interval for Step Rerun

The new co-ordinate values for M and N are used to factorise the time step interval so that the step can be rerun until the point where a collision has occurred. The time is factorised by the following equation:

$$\text{factor} = D_M \,/\, D \tag{7.8}$$

where D is the displacement during the time step and D_M is the distance to the body between the previous time step and the point where the node is at a tangent to the body. Using the example above this means that for M the distance to the body is:

$$D_M = M_0 - M \tag{7.9}$$

It is usually the case that more than one collision occurs during each time step. In this situation the smallest factor calculated of all the collided nodes is used to factor the time. The time step length for the rerun Δt_R is then calculated:

$$\Delta t_R = \text{smallest factor} * \Delta t \tag{7.10}$$

Once a node has collided with the body model and the step has been rerun, then this node is restrained so that it is not included in any further calculations.

As stated earlier this method was initially used in this research. However it was found that in most collisions, the tangent position was actually the location at the previous time step, or very close to it. This meant that the additional calculations to determine the tangent could be eliminated without losing a great deal of accuracy.

The process for collision response used in the draping system, once a collision has been detected, is to reset the cloth to the previous time step's location, restrain any colliding nodes, and to rerun the time step, keeping the original time step length. This is possible because the distances moved at each step are very small compared to the total distance. However the response method described above is ideal for systems where larger distances are covered during one time step.

7.6 Conclusion

The approach followed by this research project to the collision detection problem is new and unique compared to other detection algorithms studied. The mapping method has reduced the overall calculation time involved by simplifying the checks for collision and this has resulted in a very accurate and realistic simulation.

This mapping method can potentially be applied to many collision detection problems. Although the simulation involves one static model and one dynamic model it can easily be applied to two (or more) moving models. Currently the body model mapping needs to be performed only once and this is at the beginning of the draping program. However in a dynamic situation the body model mapping will simply be carried out at each time step.

In addition, more complex objects could be partitioned locally. For example with a full body model, the torso would be mapped to a cylinder as described in this thesis and similar cylinders could be developed for the arms and legs so that more complex clothes involving sleeves and trouser legs could be modelled.

The body model mapping is simplified to some extent due to the information being supplied in slices. Where this is not the situation for other data sets then the slice test performed at the beginning of the check on each cloth node will have to be omitted. In this case the four closest body nodes around the cloth node will have to be determined from the whole body model rather than a subset of two slices.

Another advantage of this method is that it reduces the number of comparisons at each time step. The comparison has been reduced to one check between the distances moved by each cloth node or element section and the length of the cylinder radius, rather than several checks between all the faces of the body model. The method also removes the necessity of performing an additional containment test, in the situation where one object is completely contained by the other without any faces intersecting, as the mapping method will have already established whether the node is within the body model volume.

Criticism of the collision response has pointed out that restraining the collided nodes indefinitely works well for the current model as the cloth is draping under gravity and the restrained nodes have effectively reached their final resting place on the body.

However when extending the model so that other forces such as wind or a walking body model are included, the method would not prove realistic if these nodes were to be restrained indefinitely as the nodes would not necessarily have reached their final position. This problem can be solved however by releasing all these nodes at the beginning of each time step to ensure that any previously restrained nodes no longer in collision are once again included in the calculations. Those that are still in collision would be detected, restrained and the step rerun.

Appendix III provides the collision detection and response routines as part of the DRAPE program. The algorithm described here for collision detection has been submitted as a paper (see section headed Publications for full title).

8. DISPLAY SIMULATION

8.1 Introduction

Real-time animation is limited by the capabilities of the computer. The constraints imposed by the hardware and software such as cycle speed, storage capabilities and instruction set combine to increase the time taken to present each visual frame.

The illusion of continuous movement breaks down at speeds slower than one-fifteenth of a second. All the draping and display calculations must be completed in less than this time otherwise a flicker in the display will be noticeable to the viewer.

If the display for this system is carried out as the calculations are performed, then real-time continuous movement cannot be achieved. This is because the calculations at each time step for a reasonably sized mesh take in excess of one minute. Although special hardware such as array processors and techniques such as parallel processing will reduce the overall time, these facilities were not available for this project.

To display the cloth animation in this system the complete set of draping calculations are performed first. All the calculations for every time step are carried out and the results at each stage are stored. The system's display program needs only these results. It is possible to display these results as successive frames within the maximum time limit because the largest amount of processing has already been carried out. This is known as real-time playback.

Intermediate steps can be calculated to improve the display even further. In animation this is known as in-betweening. Two key frames are specified and additional drawings are calculated by computing linear distances between two corresponding points. In this system the key frames are the results at the end of each time step and the points are the co-ordinate information for each node.

It is important to use time steps which are not too large because of the flexible nature of cloth. Cloth does not necessarily move in a linear fashion so in-betweening should be performed between two sets of data which are fairly close to each other. If the distances are too big then realism may be lost.

A graphics workstation is a necessity to cope with the rapid display of a sequence of frames which have been positioned, rendered and subjected to lighting models. This chapter describes the features necessary for realistic results and how they have been used in the display simulation program.

8.2 Silicon Graphics

When the research was in its early stages and draping only 2D strings a Sun workstation was used to display the results. The available UNIRAS interactive software on the Hewlett Packard was more than adequate for displaying the early results as a series of graphs. Later however when the data became more complex, and animation a definite possibility, a better display system had to be found.

A Silicon Graphics Indy has been used to display the results of the DRAPE program. This is because its high performance graphical facilities give the best animation performance from the available hardware and this has proved adequate for the project's needs. The main time constraint is taken in performing the draping calculations and only technological improvements in this area will reduce the time for the system as a whole.

8.2.1 Mesh Description

The drawing subroutines used by the Silicon Graphics to display images are based on points, lines and polygons. The high performance library is made up of these routines which are tuned to the hardware architecture. The programmer declares the co-ordinates of points which define the shape to be displayed as well as their respective relationships in the form of polygons.

8.2.2 Colour

Colour is defined using RGB mode where each pixel is composed of three different phosphors which glow red, green or blue. The value of each phosphor is defined by a number between 0 and 255 where 0 is off, 255 is completely on and the numbers 2-254 are some intensity between the two. The combination of the red, green and blue values determine the colour of the resulting pixel.

8.2.3 Animation

Double buffering is used in dynamic display. This is where a frame is held on the display until the next frame has been completely generated. This is in comparison to single buffering where whatever is being drawn is immediately visible on the screen.

Animating in single buffer mode gives a visible flicker in anything remotely complex because the viewer sees each alteration to the image. This means that parts of two frames may be displayed at the same time. Single buffer mode is used for static images.

8.2.4 Hidden Surface Removal

The process of drawing surfaces which are not visible to the viewer can seriously degrade the performance of a complex display. Z-buffering is used to draw only the surface closest to the viewer.

Z-buffering operates by calculating the distance to the eye from the surfaces covering each pixel and drawing only the closest surface. When double-buffering is in operation, these calculations are carried out without the viewer knowing because only the completed frame is displayed.

8.2.5 Lighting

The colour of an image is affected by the colour, location and direction of light sources, the colour and surface properties of the image and the position and viewing direction of the viewer. The Graphics Library is capable of controlling these parameters to provide realistic images. There are routines which can provide both ambient and spot lighting.

8.3 The Display Simulation Program DISPLAY

The DISPLAY program reads in the nodal co-ordinate values generated at each time step by DRAPE. Body modelling data supplied by the LASS system is also read into the DISPLAY program and displayed at the same time. Details of how these input files are generated have been provided in earlier chapters. The visual effect of the system is to show the cloth skirt draping around a static human body. DISPLAY runs on the Silicon Graphics Indy machine. It is written in standard C and uses many of the Graphics Library functions. An overview of the program is provided in Figure 8.1.

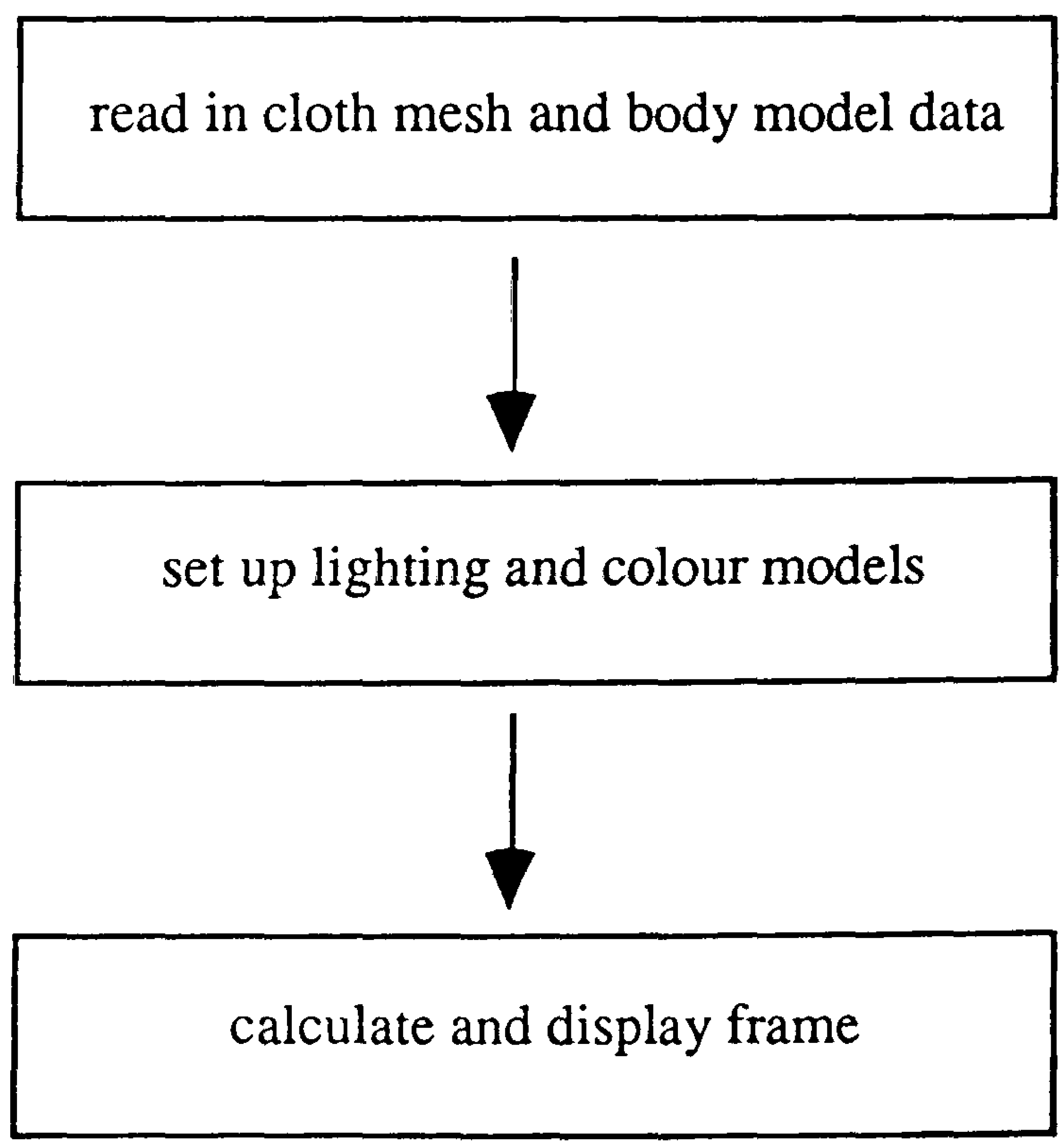


Figure 8.1 - overview of DISPLAY

8.3.1 Cloth Mesh and Body Model Data

The file "coord" holds the mesh control information and a listing of nodal co-ordinates. The control data required for accurate display are the total number of nodes in the cloth mesh, the number of time steps, the number of rows in the cloth mesh and the number of nodes in each row.

The nodal listing is the Cartesian co-ordinates of each node at each time step. This nodal information is stored in an array after being read in and if a smooth animation is required, intermediate locations can be calculated between these main time steps. This was initially carried out, but it was found that the time step length resulted in small displacements at each step, thus eliminating the need to "in-between" for this particular application.

The body modelling data is read in from the file "bodymodel". This includes the total number of body slices and the total number of nodes for the body model. The listing defines the x, y and z co-ordinates of each node for a static body model. DISPLAY reads this information into a matrix and displays the body data at the same time as the cloth mesh.

The nodes representing the body model and the cloth are linked together in their respective models into triangular meshes. These are then fully rendered to provide the full image.

8.3.2 Display Background

A graphics window is specified to allocate a region on the screen in which to display the information. Double buffering is requested for smooth animation, z-buffering is requested for a fast display of the information and colour and lighting models are declared.

8.3.3 Displaying Each Frame

For each frame the whole body model and a complete set of mesh data is displayed. The control data read in from "coord" and "bodymodel" are used to display the correct sequence of each node in both the cloth mesh and the body model.

8.4 Conclusion

The program is written for a static body model but could easily be adapted for a dynamic display. Currently the body model data is read in once and the same data is just displayed at each frame. The complete set of cloth mesh data is also read in once but each time step is displayed on successive frames. The same principle could be applied to a dynamic body model.

DISPLAY is a very straightforward program which uses many of the routines available in the Silicon Graphics Library. The program can display adequately the cloth mesh draping around the body model although one problem encountered is that as the cloth mesh is made more refined, the display tends to slow down. However this could be overcome with a faster processor than was available to the project.

One suggestion for improving the display is to have the lighting of the models adapted for each cloth type. For example a shinier type of cloth such as satin would have a different visual appearance to cotton, yet other material property information would be fairly similar.

A program listing and descriptions of the subroutines and variables are given in Appendix IV.

9. DISCUSSION AND CONCLUSIONS

9.1 Introduction

CAD/CAM offers many possibilities for improvements in textiles and apparel manufacture. Companies that neglect it do so at the risk of their long term survival particularly now that CAD has continued to develop rapidly and is a more sophisticated tool than when it first arrived in the work place. If CAD systems are intelligently employed by designers who have an empathy with the advanced technology then this will help to strengthen a company's design image and competitive edge.

Fabric and apparel buyers are starting to demand the time reduction of the design process made possible by CAD systems. If competitors can offer a service of instant design manipulation, then buyers will not be prepared to wait the three or four weeks while a garment sample is manufactured using traditional methods. Ideas can be displayed, discussed and altered instantly using CAD, designers can become more adventurous and buyers can study fabric ideas before any cloth is even printed.

The CAD system is not intended to replace the designer. It is there simply to provide another tool in addition to existing design tools. The designer may not wish to use the CAD system but after the initial learning time those that do so have found many benefits. The real challenge of new technology is to combine it with conventional methods, creating new work practices to make the best use of both. If the new technology complements conventional methods of design then the systems will be more readily accepted. Existing processes should be examined to look for ways in which the computer can create positive changes in working practice.

Computer modelling of cloth allows a large number of ideas to be examined in three dimensions. The most exciting benefits are those that offer new tools to designers rather than digitised versions of their existing ones. Although real time 3D animation may still be some way in the future there are many shorter term goals that need to be addressed by the fashion industry.

One reason why many of the earlier CAD tools failed may have been because they were being used at only single stages of the design-production-marketing loop. Ideally they should help link the whole process together. If the CAD system is networked to other computers then operations such as production control, financial control and costing can be integrated at an earlier stage in the design process. This means that mistakes or costly designs can also be recognised at an earlier stage and therefore corrected or modified.

The future for textile design lies in a wider use of CAD/CAM systems. In time the current limitations of the systems will be overcome. Designers should keep pace with developments in order to be ready to use them when they become available.

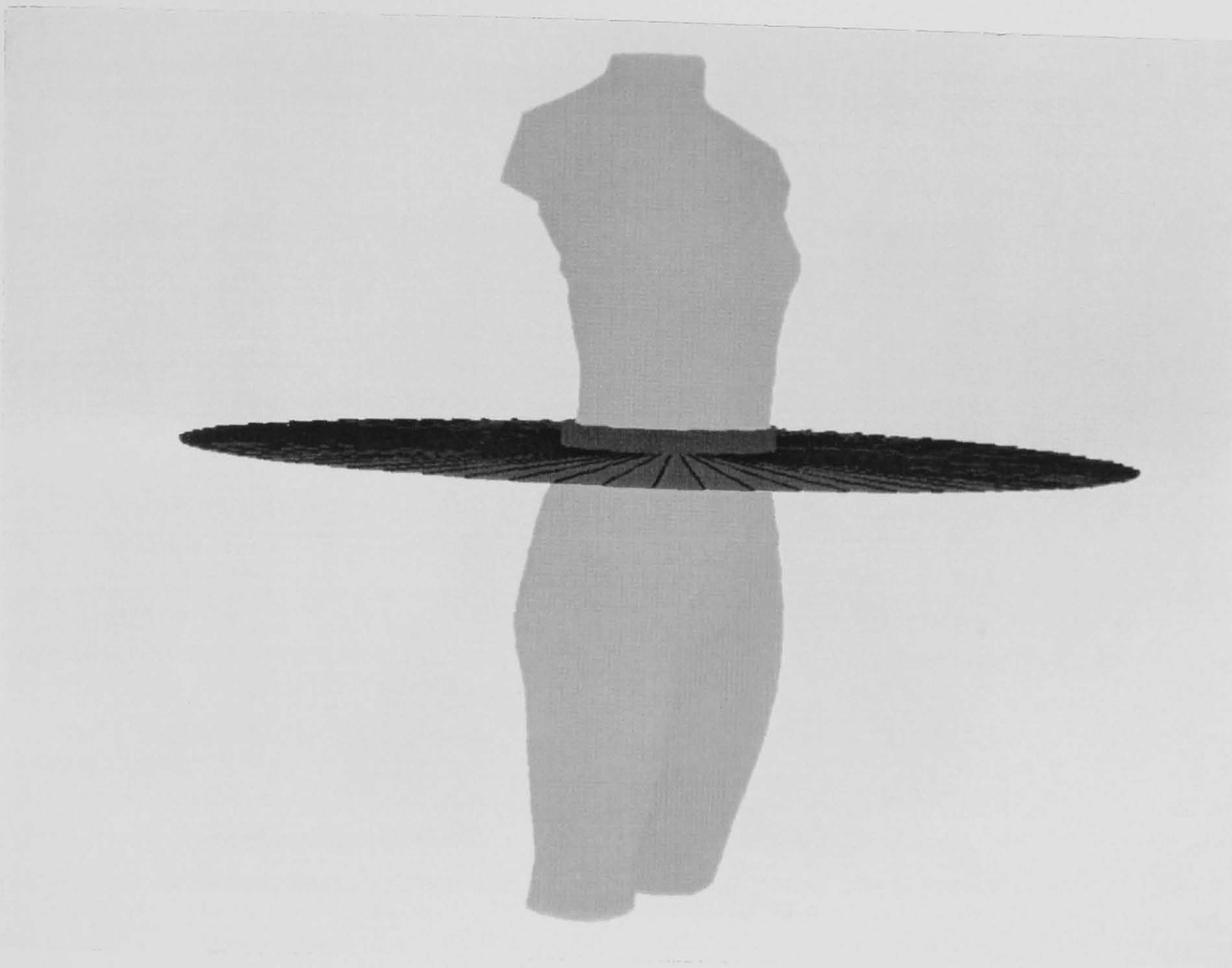
Computer aided design has been producing interesting work for some years and the initial distrust has been replaced by enthusiastic acceptance. It is up to the textile industry in collaboration with the computer graphics industry to ensure the right systems are reaching the designers not only in the work place but in educational establishments as well.

This chapter presents the achievements of this research project and discusses some potential extensions to the research for enhancement and improvement of the current model.

9.2 Results and Verification of the Modeller

In order to predict the behaviour of a particular structure under applied forces, a reliable model must be developed. This model relies on information taken from either the structure or its components. The drape modeller described in this thesis is such a model. This section provides images taken from several simulations in order to present visually how the original aims of the research have been achieved.

The thesis describes the programs developed to generate a cloth mesh representing a circular skirt. The skirt is held initially level with the body model's waist and then released to fall under the force of gravity to hang from the waist. The images shown in Figure 9.1 are a taken from a simulation to demonstrate this.



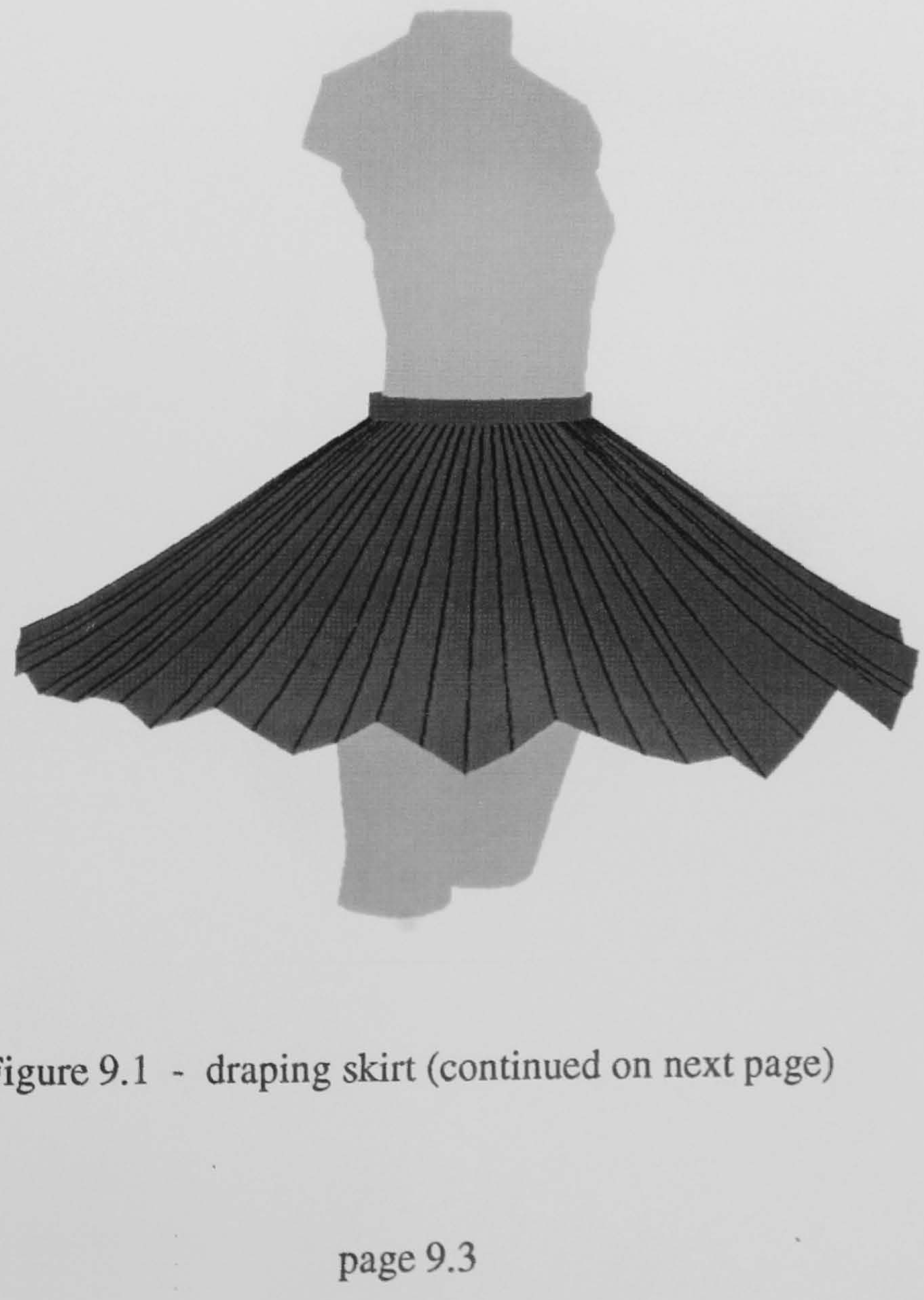


Figure 9.1 - draping skirt (continued on next page)

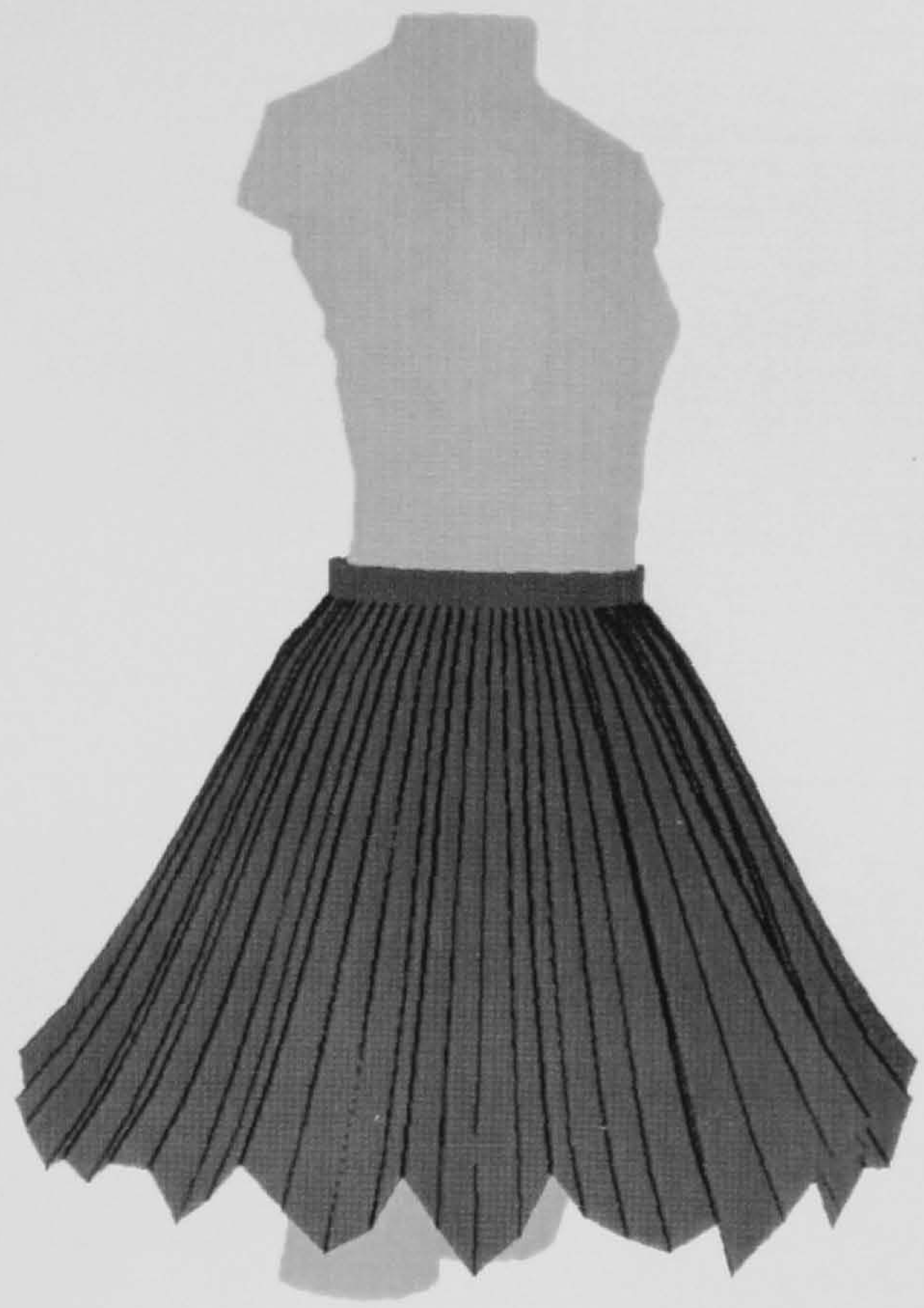


Figure 9.1 - draping skirt (continued from previous page)

To demonstrate this model further, a circular cloak was simulated, where the cloth is held around the neck of the body model and again allowed to drape due to gravitational forces, as shown in Figure 9.2. There is only one difference between the two sequences, as far as the system programs are concerned. Instead of the waist being defined as the basis for cloth mesh generation, it is the body model data slice relating to the base of the neck. The rest of the system is exactly the same.

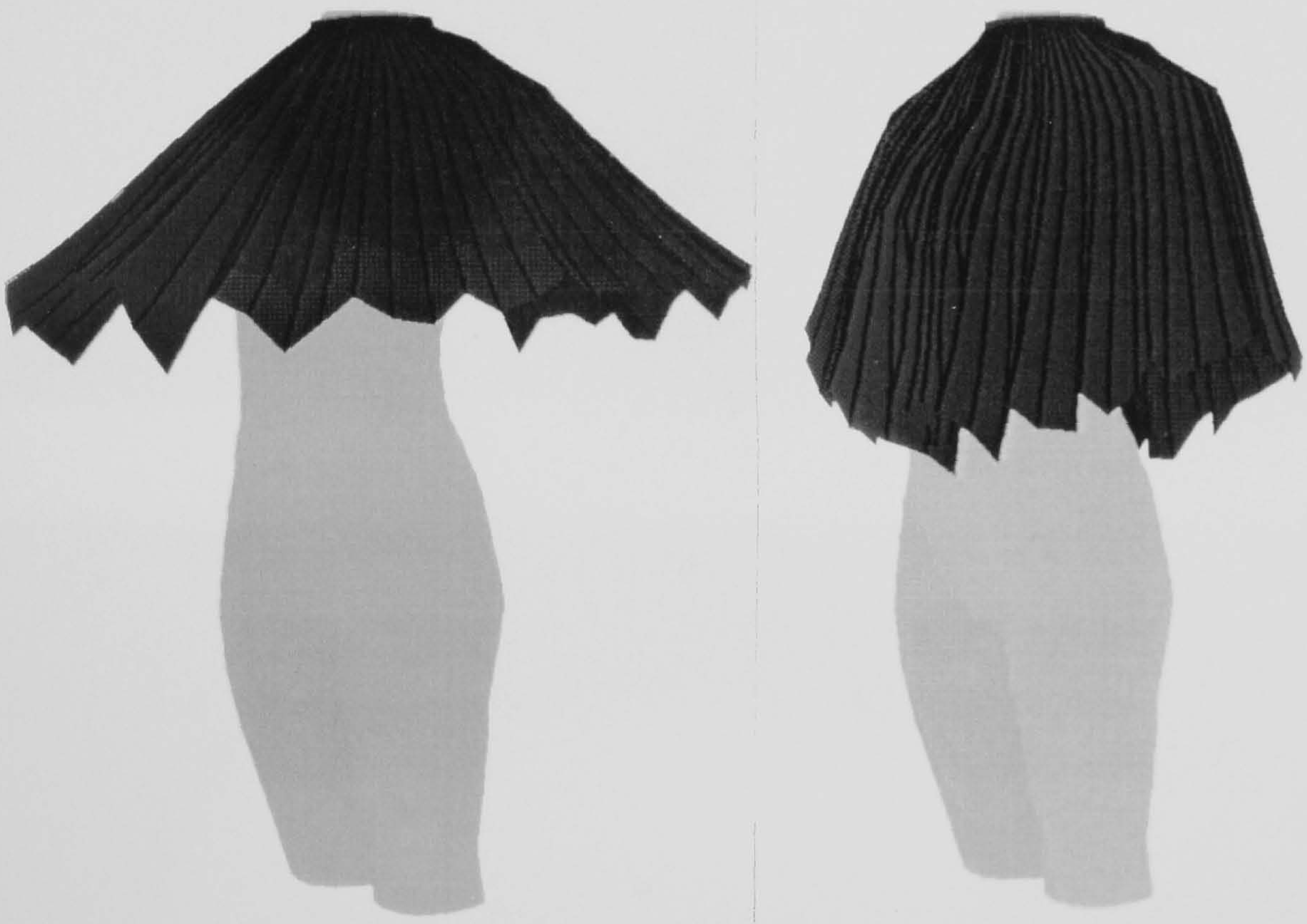
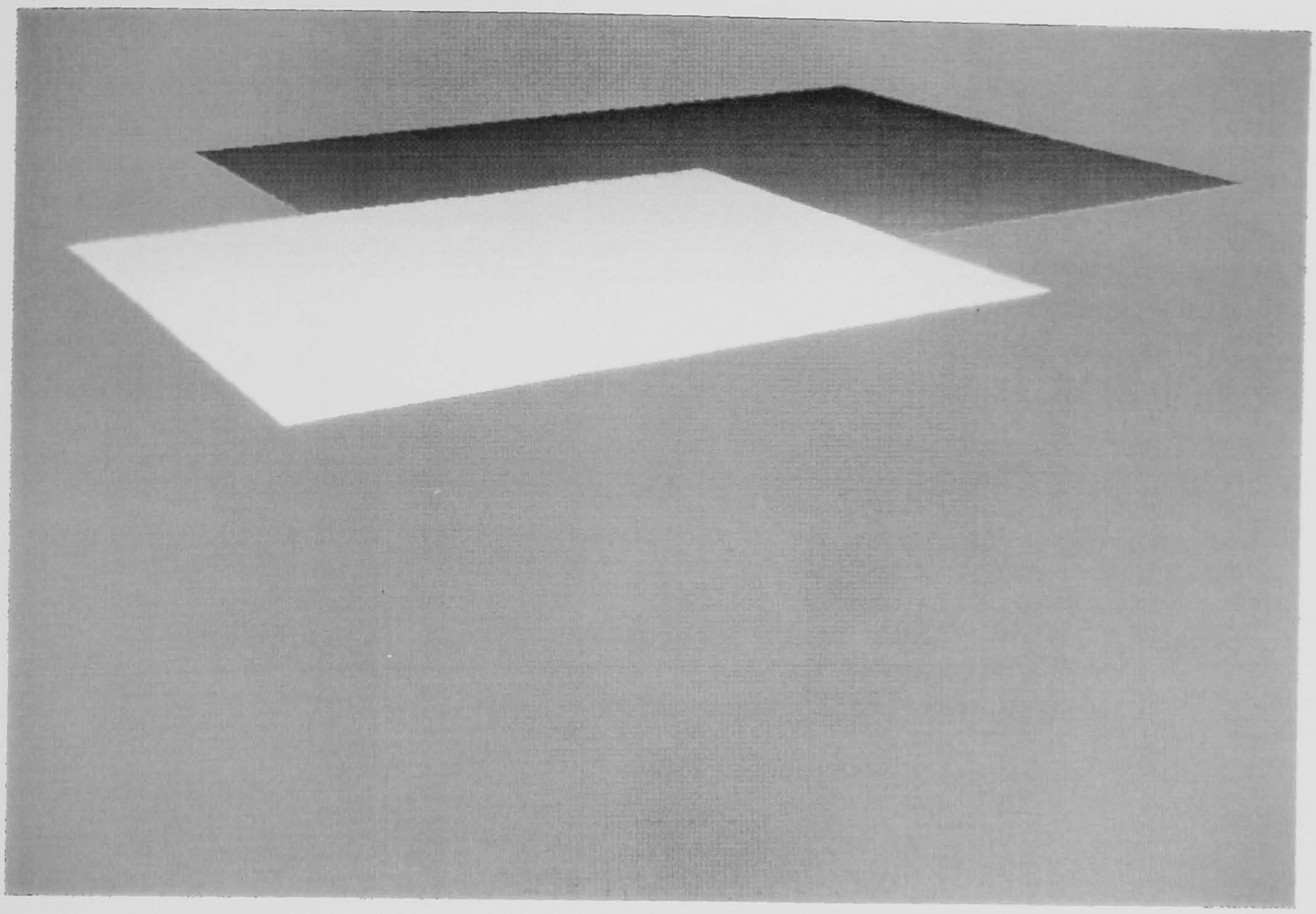


Figure 9.2 - draping cloak

The cloth model does have to be verified in some way and the most obvious is to compare the real cloth samples with the modelled cloth images. The images shown in Figure 9.3 show sequences of a rectangular piece of cloth draping over the corner of a table. The data for the model is based on the measured cotton cloth sample. The cloth model is initially held out horizontally and it falls to its fully draped position. The final image shown in Figure 9.4 can be compared with the real sample - a photograph of the real thing.



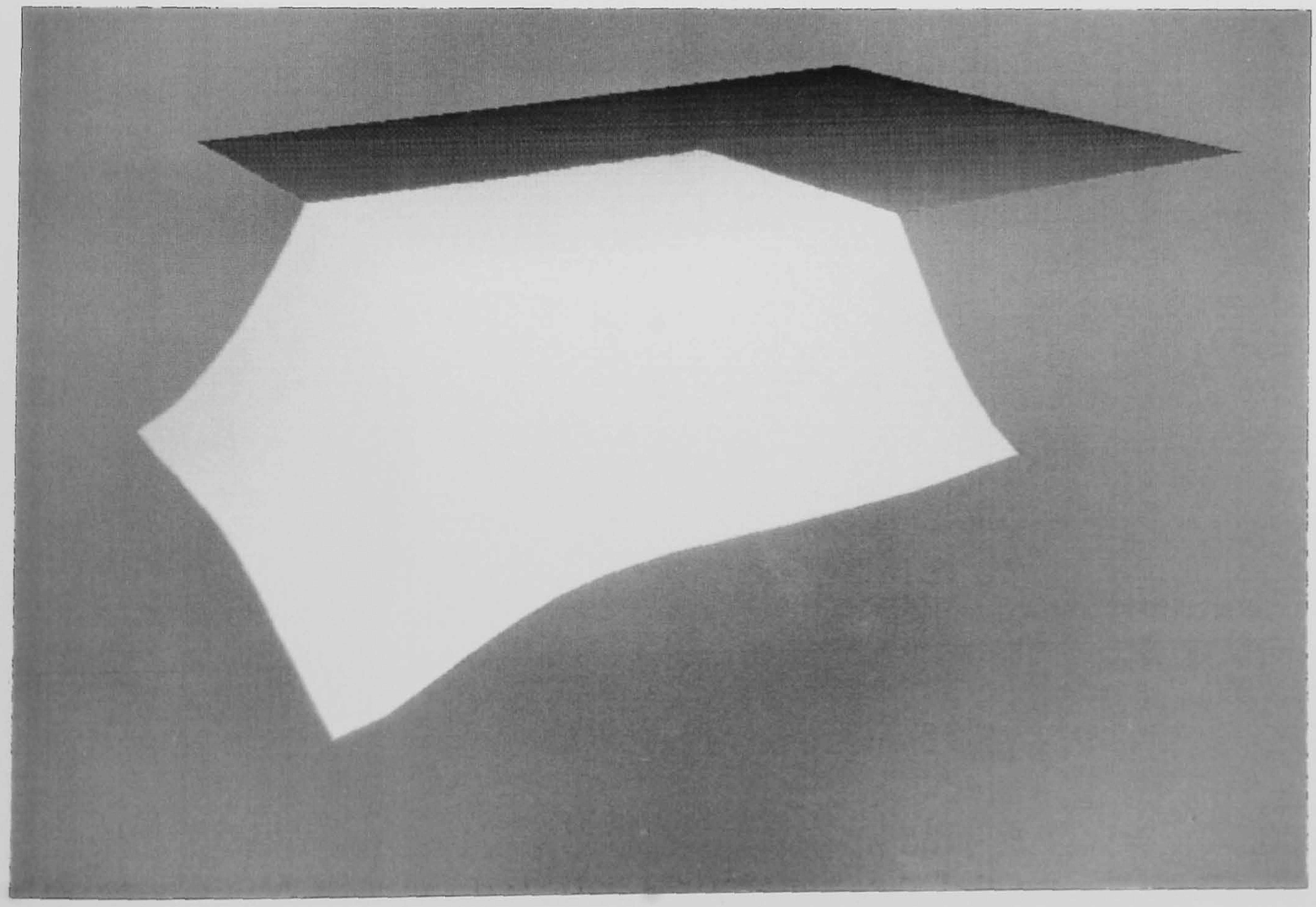


Figure 9.3 - cotton cloth draping sequence (continued on next page)

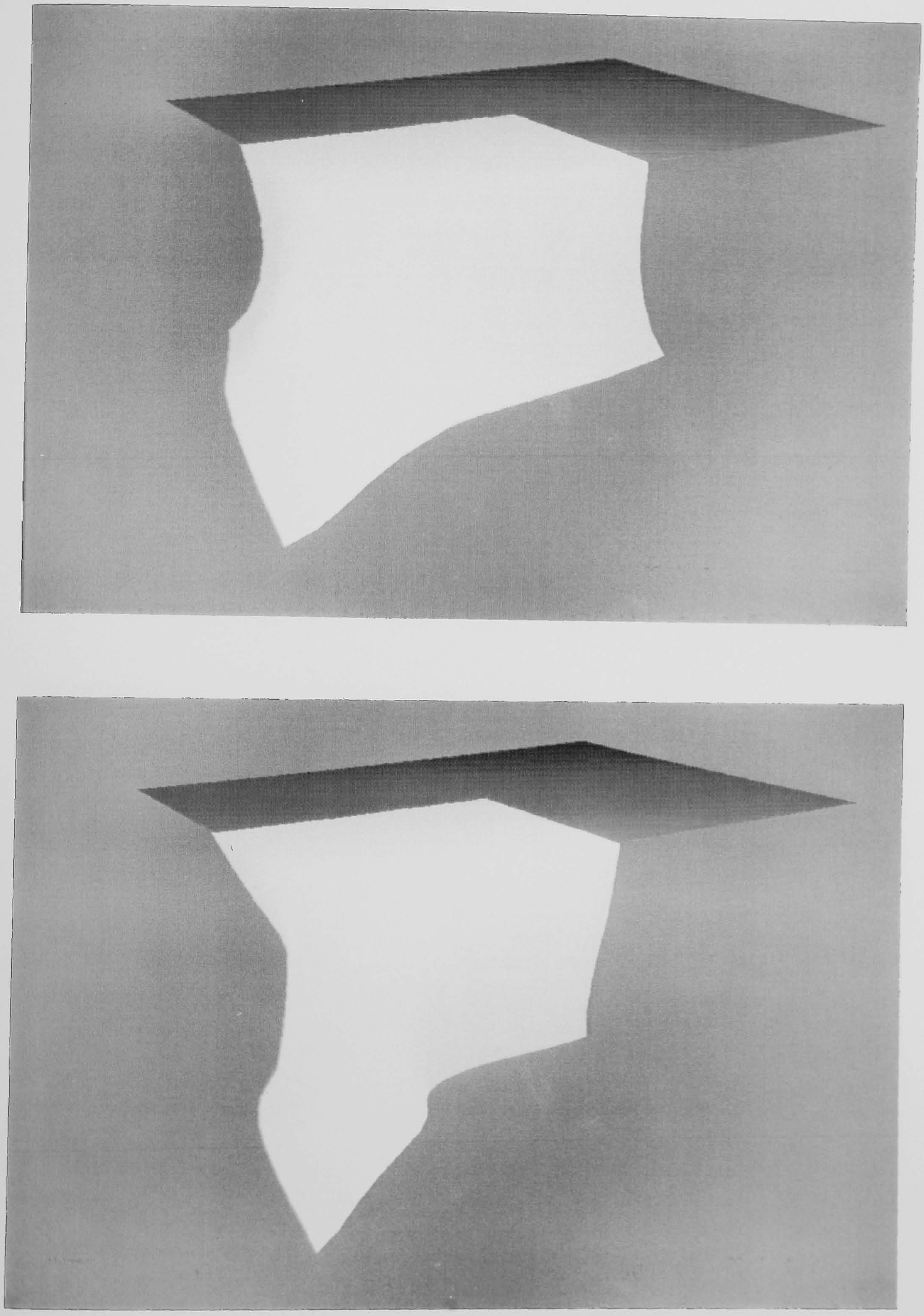


Figure 9.3 - cotton cloth draping sequence (continued from previous page)

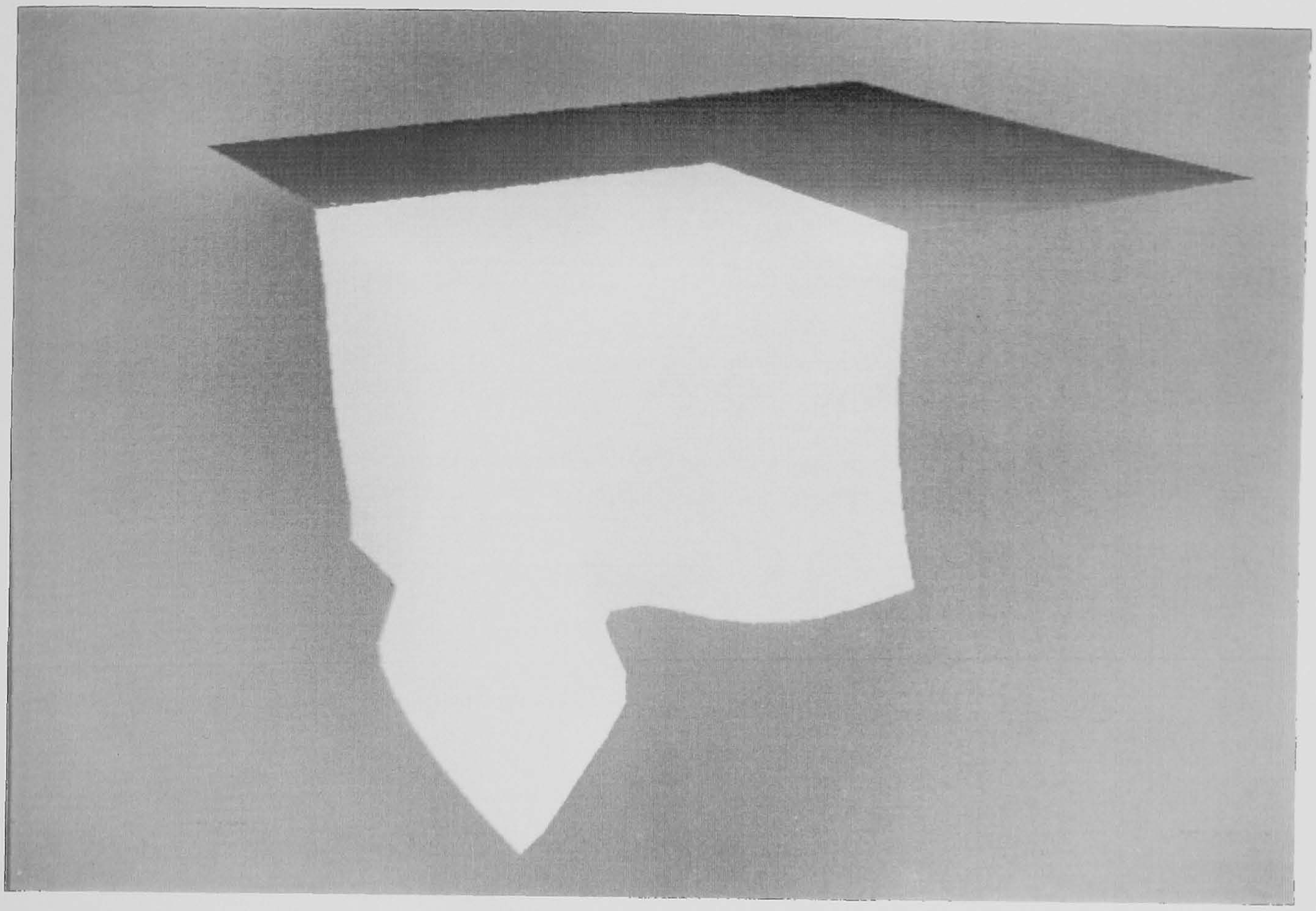


Figure 9.4 - comparison of cotton model with real cotton cloth

A similar comparison is made using the polyester measurements. The fully draped image can be compared with the real image, both shown in Figure 9.5.

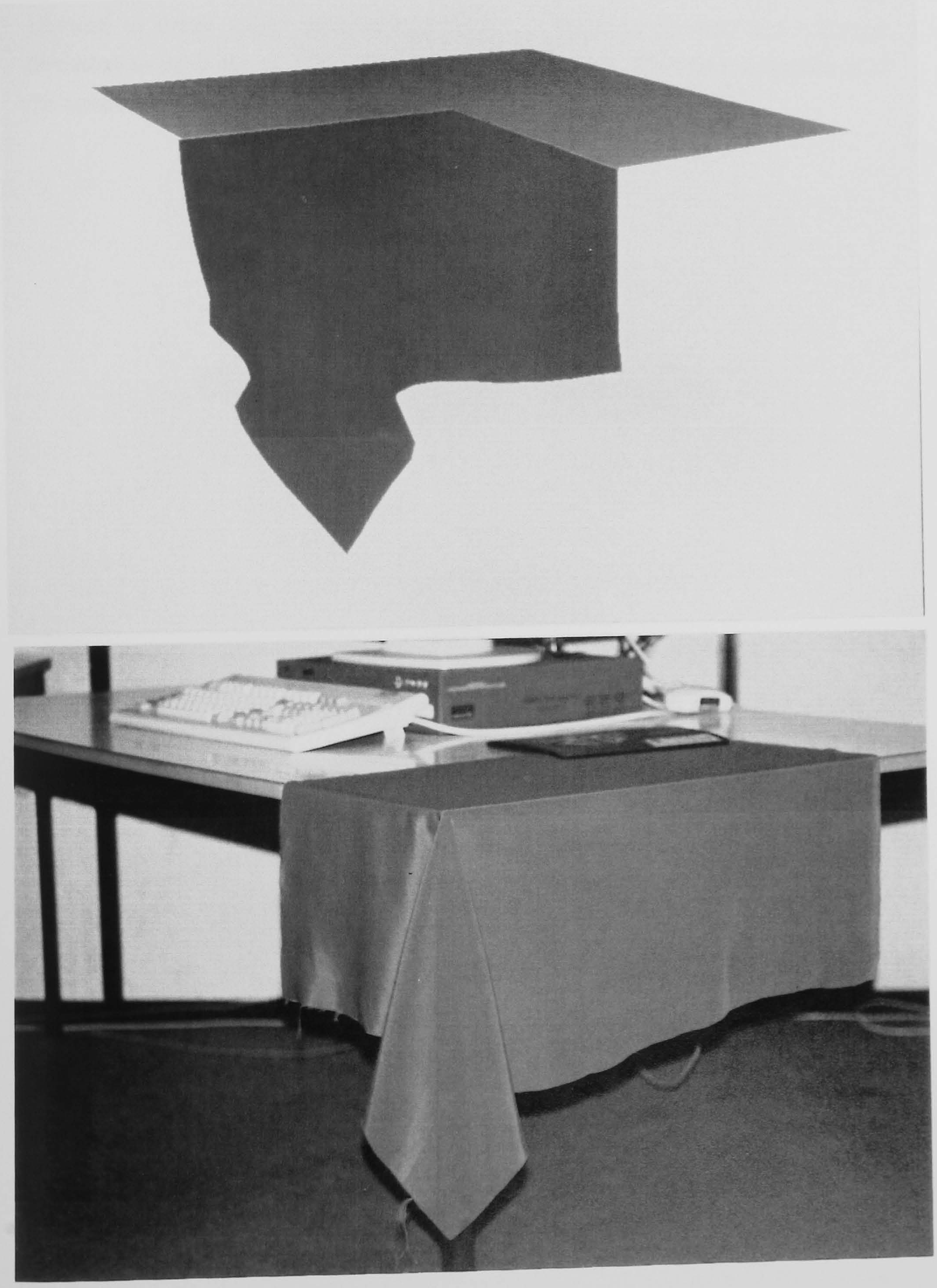


Figure 9.5 - comparison of polyester model with real polyester cloth

At the beginning of the research only simple shapes were modelled. The cloth models were circular and no collision detection was required. Figure 9.6 shows two images taken from a draping sequence. The cloth model is held flat and allowed to drape under gravitational forces. There is no need for collision detection as there are no other objects in the simulation. The cloth is restrained at the centre only.

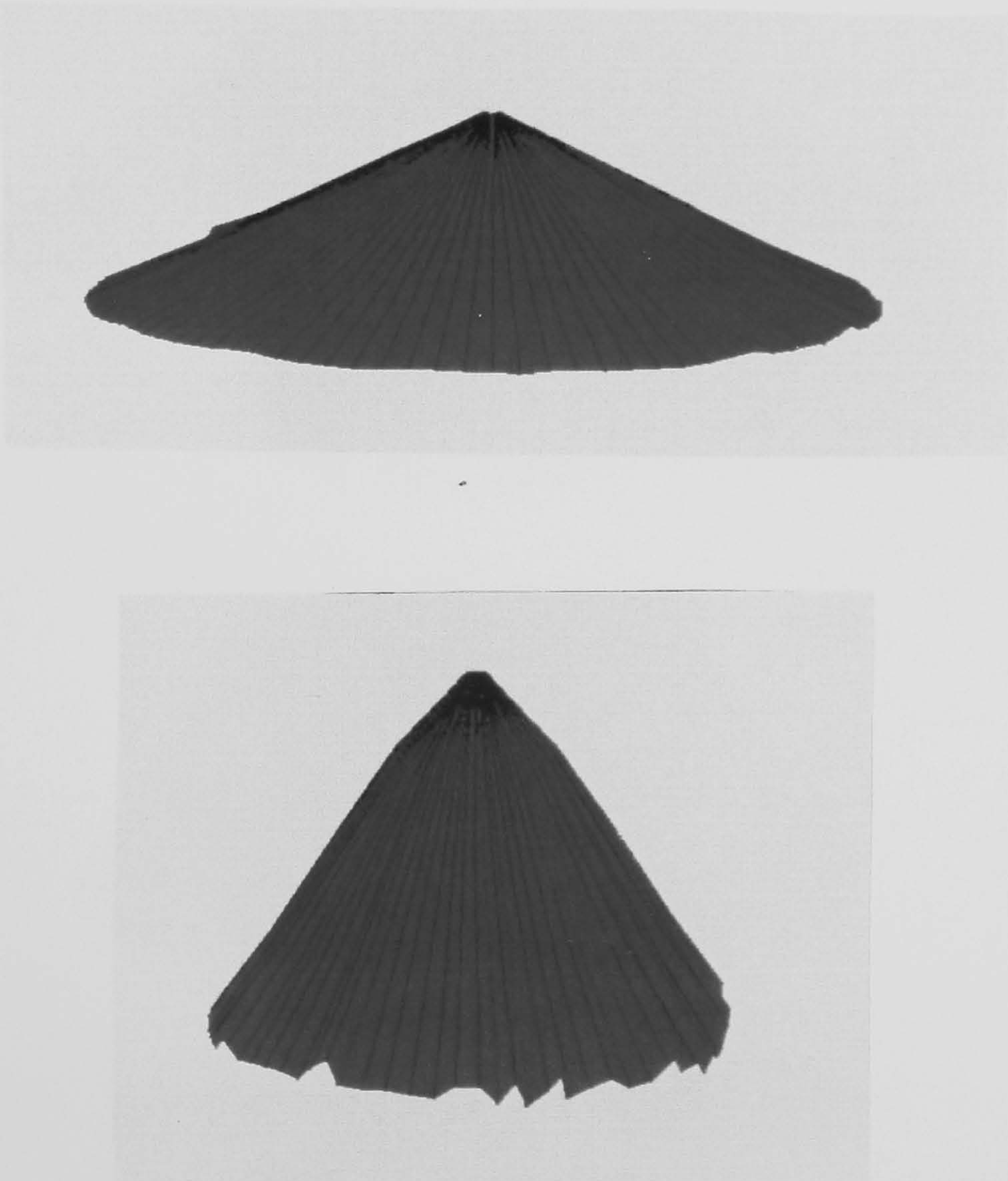


Figure 9.6 - cloth draping, restrained at centre only

Later as the drape modeller became more robust, solid objects were introduced into the simulations. Figure 9.7 shows images taken from a sequence of a circular piece of cloth draping over a solid cylinder. The collision detection routines for this sequence are more straightforward than those used for the body model, as the cylinder solid needs only the radius and height for the collision check.

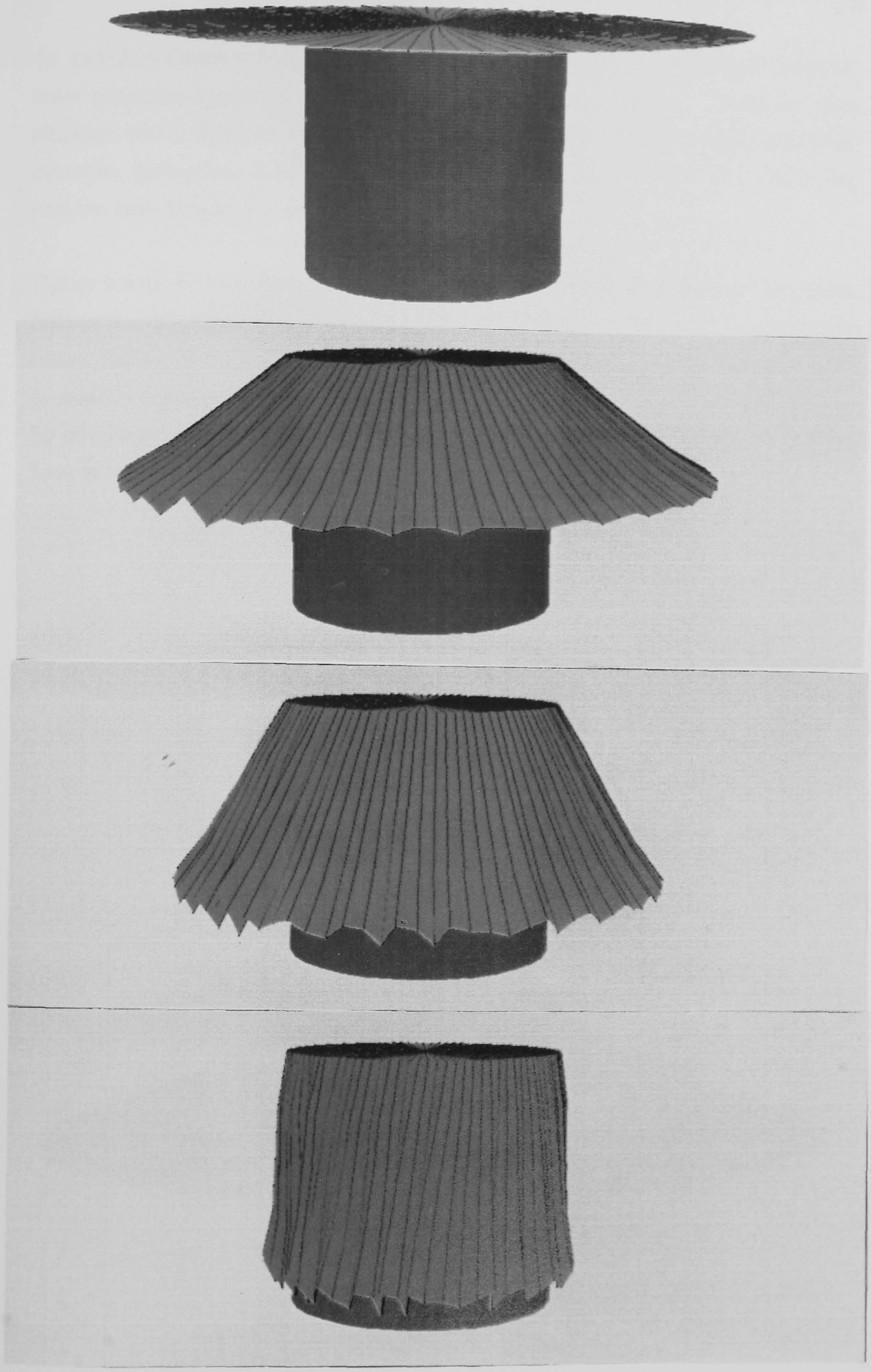


Figure 9.7 - cloth draping over a solid cylinder

As stated in Chapter 6 on the drape modeller the values for Rayleigh damping were obtained based on the assumption of critical damping. However in a situation where there are no collisions to be considered then the draping sequence overruns, giving less than critical damping. In this situation the cloth will swing past the fully draped position before settling.

Figure 9.8 show sequences taken from a simulation of this description. The cloth falls to the fully draped position and continues on towards the centre of the model before falling back to the equilibrium position. This demonstrates that the model accurately represents a real-life situation. Although accurate values still need to be obtained for Rayleigh damping, it appears that a reasonable simulation has been achieved with the approximations.

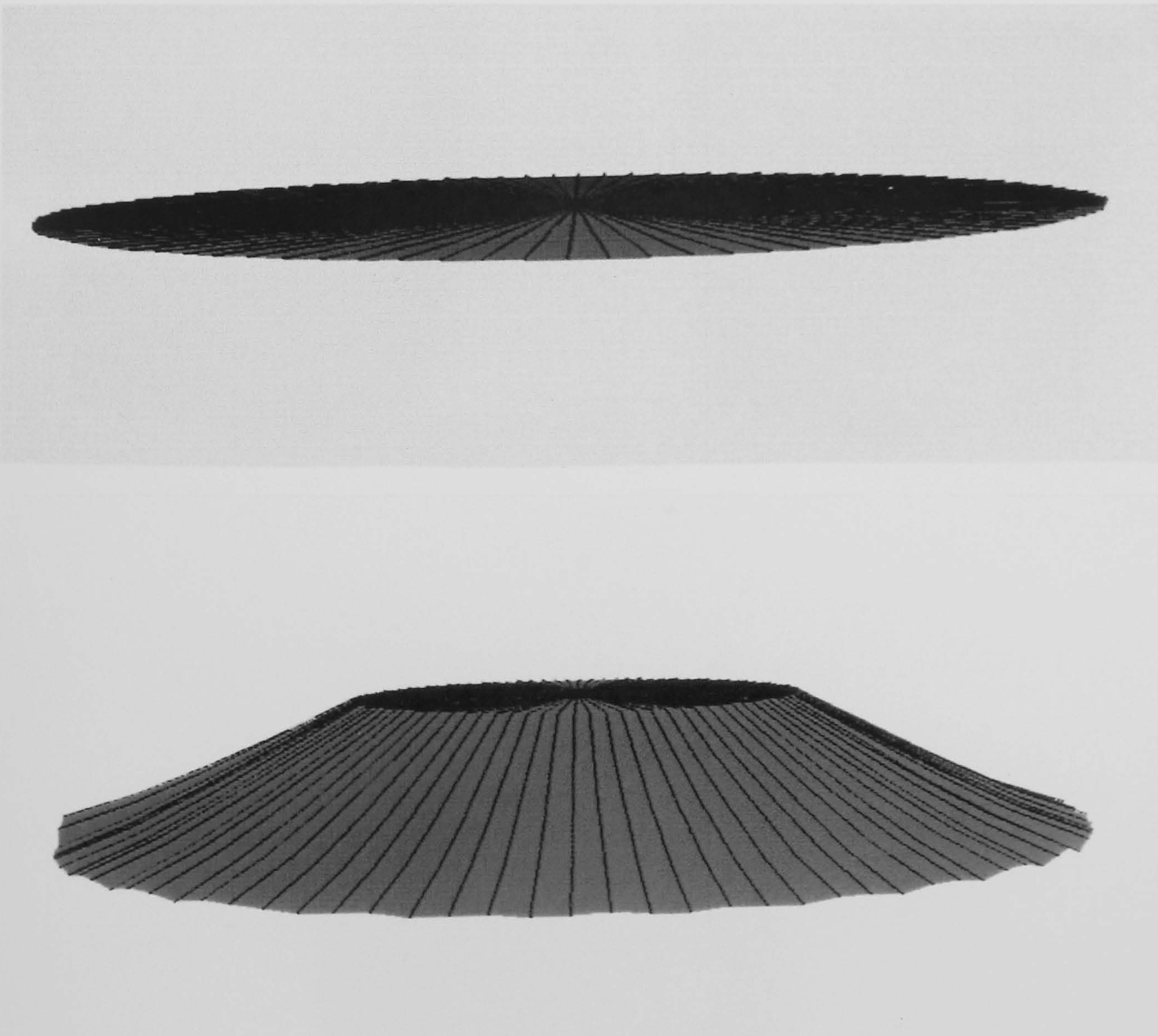


Figure 9.8 - settling into equilibrium (continued on next page)

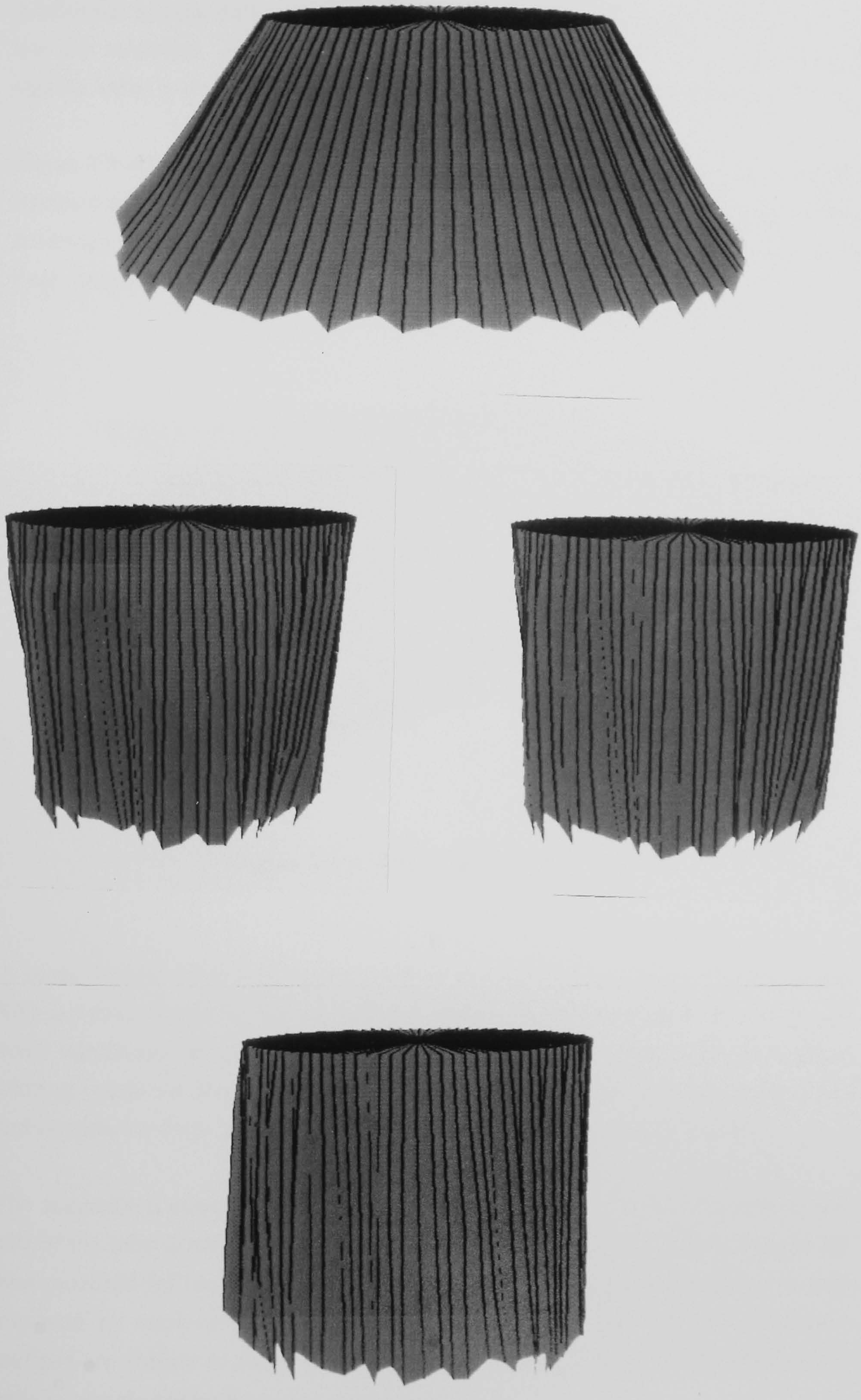


Figure 9.8 - settling into equilibrium (continued from previous page)

Additional sample data, not available to the research project, are required to fully test the modeller. However to demonstrate two alternative ideas, the flexural rigidity value was altered to determine the effect on the cloth simulations.

Figure 9.9 shows the simulation of increasing the flexural rigidity. This simulates a thick piece of cloth, like a rubber mat for example, draping over a table corner. Although this simulation was run for a series of two thousand steps, this is the final, fully draped position.

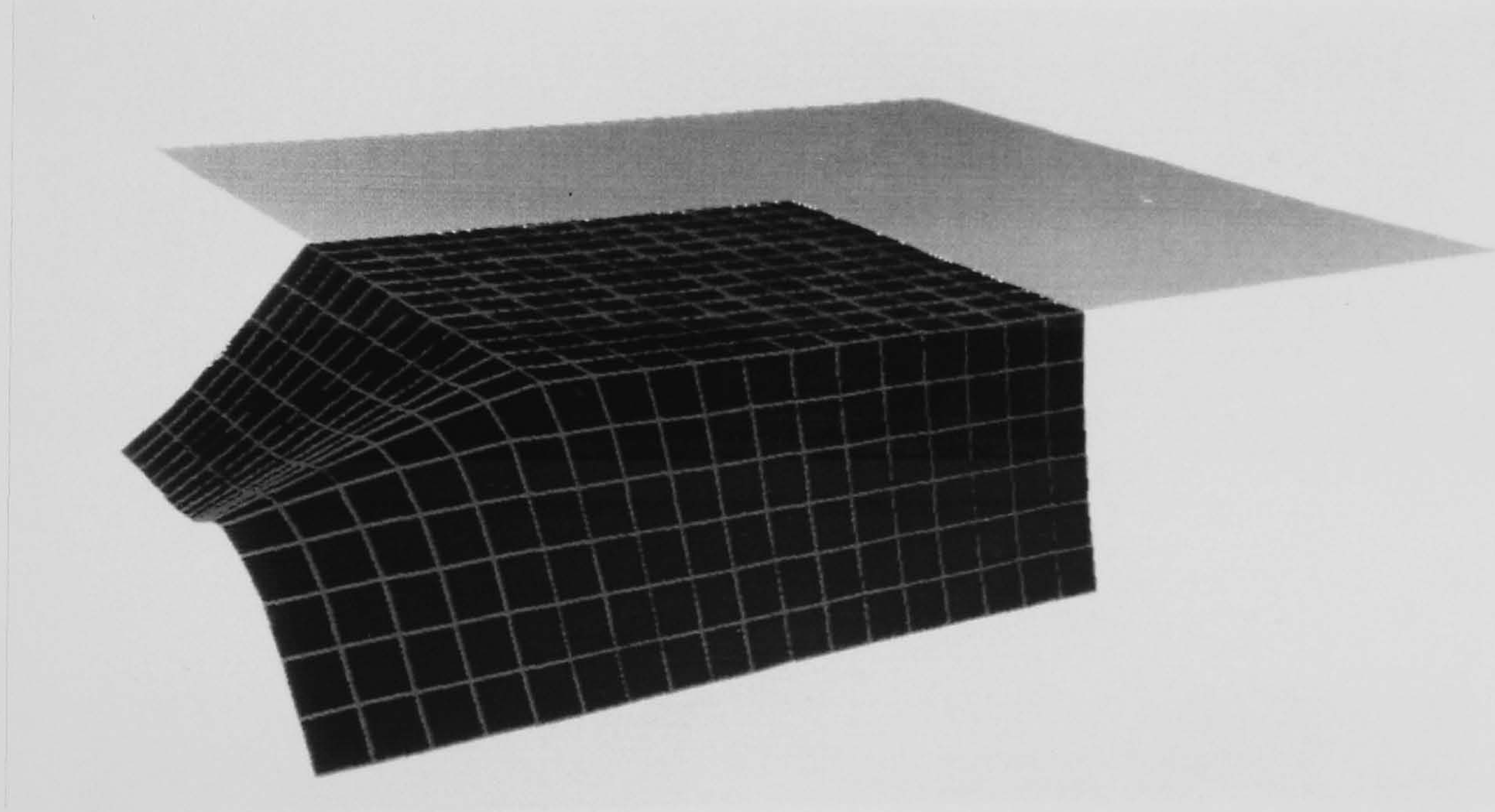


Figure 9.9 - simulation of rubber mat

A more flexible cloth can be obtained by decreasing the flexural rigidity value. This is demonstrated by the sequence of images shown in Figure 9.10, which uses some wireframe images for a clearer display. Here a cylindrical tube is modelled starting inside the body. The tube is pushed out during the sequence until it lies just outside the body and as each node reaches the outside, it is restrained.

The sequence is complete when all nodes are restrained. For this simulation, one half of the tube is restrained while the other half is pushed out and the results are then mirrored for the display. This is one way of reducing the computation time involved by exploiting the symmetry of the image. The collision detection routines are similar to those used for the skirt except that the test is whether the node is outside the body model, rather than inside.

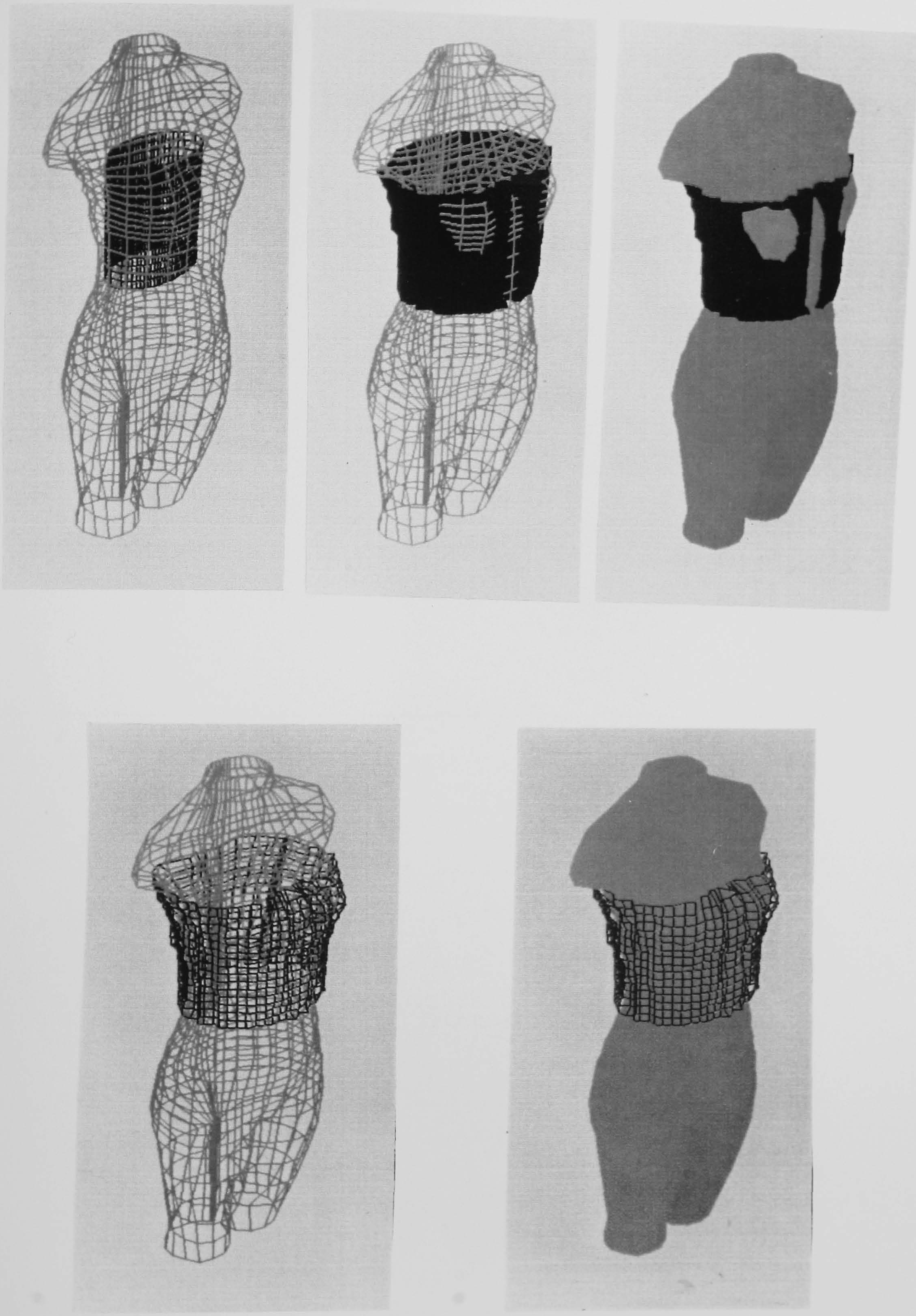


Figure 9.10 - simulation of body hugging garment

Figure 9.11 shows the fully rendered final image, as well as the fully draped skirt.

Figure 9.11 - fully rendered top and skirt

9.3 Summary of the Modeller

The drape research described in this thesis started with a skeleton finite element program taken from the NAG Library. The Newmark integration section has been retained but the remainder of the program has been drastically modified. In addition collision detection and response sections have been developed.

Finite element modelling has been used by other research groups in the study of cloth drape. However the approach here has been to improve the realism of these models by incorporating a body model, including physical properties to define the cloth model and by using enhanced graphics for the display.

Finite element modelling has provided realistic results, not only in this study but others as well. However because a complete solution must be performed at each step, the processing time is excessive for the intended application and would be unacceptable to a designer.

Techniques have been applied to obtain an efficient mesh and dedicated machines have been employed to run the programs, but real-time animation has not been achieved. Real-time realistic cloth drape is still a future research goal.

It was always the intention to include a body model in the research, although this was only fully incorporated towards the end of the project. This has proved to be one of the more exciting parts of the research in that a new method has been developed that simplifies the process of collision detection. Although efficiency was not necessarily the aim, due to the length of processing time involved for this project, applying the collision detection method to another system where real-time is possible should not seriously hinder the overall performance.

As mentioned earlier in this thesis, it should be straightforward to apply the algorithms developed here for collision detection to a completely dynamic situation. Further testing with a real-time simulation would be required.

9.4 Extensions to the Modeller

The research has achieved the initial aims of using the Finite Element Method to develop a cloth drape modeller, incorporating real cloth properties and interacting with a body model. To be of more practical use to the fashion industry additional garments based on real pattern pieces need to be draped around a dynamic body model.

The images presented earlier in this chapter demonstrate cloth draping under different conditions. In some situations collision detection is carried out, in others it is not necessary to do so. However they all use the same drape model program, it is only the mesh generators and display programs that need to be modified.

All the circular cloth mesh generators are similar to the circular skirt mesh generator. The rectangular cloth models, demonstrated by the cloth draping over the table corner, have a slightly different format - in fact a much simpler generator is required for these models. The display programs have been modified to show different colours and viewing locations. For a complete system, more general mesh and display generators need to be used. These can either be modified from existing generators or developed specifically for the application.

The thesis describes the drape model using a circular skirt draping around a human body. The skirt is a simple application and is very straightforward to represent when generating the mesh. The starting position of the whole skirt is level with the body model waist. In effect this is the whole pattern "seamed" together and held out at its full extent.

For an application such as a straight skirt, the starting position down the length of the skirt would be closer to the body. The same applies for more body fitting garments such as trousers and jackets. The problem here is generating the mesh topology and geometry. The circular skirt is straightforward because the starting position is developed in two dimensions only. The starting position for other garments will probably have to be meshed in three dimensions.

The problem of collision detection involving these more complex garments has already been mentioned in the relevant chapter. The body model mapping could be done in local sections. In the case of a pair of trousers, for example, three mapped cylinders would be required, one for the torso and one each for the legs.

This thesis has not tackled the effect on drape when applying pockets, seams, pleats, waistbands and other accessories. An enhancement on the research might be to apply knowledge based methods to incorporate these features. These methods aim to reduce the complexity of the mathematical solution. They could be applied as a modelling assistant that incorporates drape heuristics involving handling constants of materials, effects of seams and pleats and other garment features. The information would be obtained by interviewing garment designers to arrive at common maxims that can be applied. This novel approach of integrating the mathematical and knowledge based modelling offers the potential of both increased quality and speed of the 3D display.

Real-time processing is one important factor that could not be achieved during the research period. Until this has been achieved successfully, 3D display cannot be legitimately involved in a CAD package as part of an interactive 3D design facility. Speeding up the calculation process would involve faster processors and possibly parallel processing. Also the possibility of using explicit dynamic solutions for solving the draping equations could be researched.

Other interesting enhancements would be to include aerodynamic forces and moving body models. The current model has the ability to apply forces in any direction and this has been demonstrated by the simulations described earlier.

One major enhancement would be to include friction in the analysis. The model is currently not capable of dealing with cloth moving against a solid, for example in the situation where a cloth slips off the edge of a table.

There are also several cosmetic alterations that could be added to the display such as curve fitting to generate smoother folds as well as more adventurous use of lighting and material models.

9.5 Conclusion

This thesis describes the system and results obtained from a three year research period looking into the 3D simulation of cloth drape. The current system uses standard FORTRAN, C, NAG Library and Silicon Graphics Library routines.

Other methods of modelling cloth using finite element methods have been tried and this method works at least equally well. The use of Newmark's method has proved successful and this is demonstrated by the images presented earlier. The collision detection work resulted in a new method being developed that gives very satisfactory results. The research has achieved its aim for establishing the basis for a cloth drape modeller.

PUBLICATIONS RESULTING FROM THE RESEARCH

Bez HE, Bricis A, Ascough J

" A Collision Detection Method with Applications in CAD Systems for the Apparel Industry"

Submitted to CAD, October 1994

Ascough J, Bez HE, Bricis A

"Finite Element Simulation of Cloth Drape"

Submitted to Journal of the Textile Institute, December 1994

REFERENCES

ABB73 Abbott GM, Grosberg P, Leaf GAV. "The Elastic Resistance to Bending of Plain Woven Fabrics". J Text Inst, 1973, vol 64, pp 346-362

AMI86 Amirbayat J, Hearle JWS. "The Complex Buckling of Flexible Sheet Materials". Int J Mech Sci, 1986, vol 28, no 6, pp 339-358, pp 359-370

AMI89 Amirbayat J, Hearle JWS. "The Anatomy of Buckling of Textile Fabrics: Drape and Conformability". J Text Inst, 1989, vol 80, no 1, pp 51-70

AON90 Aono M. "A Wrinkle Propagation Model for Cloth". Proc Comp Graph Int'l 90, Springer-Verlag, Tokyo, pp 95-115

BAL86 Balfour JAD. "Computer Analysis of Structural Frameworks". Collins Professional and Technical Books, 1986

BAR84 Barr AH. "Global and Local Deformations of Solid Primitives". SIGGRAPH 84 ACM Comp Graph, July 1984, vol 18, no 3, pp 21-30

BAT82 Bathe KJ. "Finite Element Procedures in Engineering Analysis". Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1982

BEH61 Behre B. "Mechanical Properties of Textile Fabrics Part I: Shearing". Text Res J, 1961, vol 31, no 2, pp 87-93

BRE92 Breen DE, House DH, Getto PH. "A Physically-Based Particle Model of Woven Cloth". Visual Computer, 1992, vol 8, pp 264-277

BRO90 Brown III PR, Buchanan DR, Clapp TG. "Large Deflection Bending of Woven Fabric for Automated Material Handling". J Text Inst, 1990, vol 81, no 1, pp 1-14

BUL849 "Computerisation of Pattern Making from Apparel Design". Bulletin of Research Institute for Polymers and Textiles, Japan, 1984-89

CAM90 Cameron S. "Collision Detection by 4D Intersection Testing". IEEE Trans on Robotics and Automation, June 1990, vol 6, no 3, pp 291-302

CHA87 Chajes A, Churchill JE. "Nonlinear Frame Analysis by Finite Element Methods". J Structural Engineering, 1987, vol 113, no 6, pp 1221-1235

CLA91 Clapp TG, Peng H. "A Comparison of Linear and Nonlinear Bending Models for Predicting Fabric Deformation in Automated Handling". J Text Inst, 1991, vol 82, no 3, pp 341-352

COA88 Coats RC, Coutie MG, Kong FK. "Structural Analysis". 2nd Edition, Nelson, Surrey UK, 1988

COL91 Collier JR, Collier BJ, O'Toole G, Sargand SM. "Drape Prediction by Means of Finite Element Analysis". J Text Inst, 1991, vol 81, no 1, pp 96-107

CUS61 Cusick GE. "The Resistance of Fabrics to Shearing Forces". J Text Inst, 1961, vol 52, T395-T406

DAH61 Dahlberg B. "Mechanical Properties of Textile Fabrics Part II: Buckling". Text R J, 1961, vol 31, no 2, pp 94-99

DE77a De Jong S, Postle R. "An Energy Analysis of Woven Fabric Mechanics by Means of Optimal Control Theory Part I: Tensile Properties". J Text Inst, 1977, vol 68, pp 350-361

DE77b De Jong S, Postle R. "An Energy Analysis of Woven Fabric Mechanics by Means of Optimal Control Theory Part II: Pure Bending Properties". J Text Inst, 1977, vol 68, pp 362-369

DHA93 Dhande SG, Rao PVM, Tavikkoli S, Moore CL. "Geometric Modelling of Draped Fabric Surfaces". IFIP Transactions B - Applications in Technology, 1993, vol 9, pp 349-356

DUN72 Dunham RS, Nickell RE, Stickler DC. "Integration Operators for Transient Structural Response". Computers and Structures, 1972, vol 2, pp 1-15

FER93 Fernandez FA, Yong YC, Ettinger RD. "A Simple Adaptive Mesh Generator for 2D Finite Element Calculations". IEEE Trans on Magnetics, vol 29, no 2, 1993, pp 1882-1885

FOU86 Fournier A, Reeves W T. "A Simple Model for Ocean Waves". Comp Graph, vol 20, no 4, 1986, pp 75-84

FOZ91 Fozzard GJW, Rawling AJ. "Simulation of Dressing and Drape for Garment CAD". Proc 6th Int'l Forum on CAD, East Midlands, UK, 1991

FOZ92 Fozzard GJW, Rawling AJ. "CAD for Garment Design. Effective Use of the Third Dimension". Proc 8th Nat'l Conf for Manf Res, 8-10 Sept 1992, Birmingham, UK

GAN93 Ganter MA, Isarankura BP. "Dynamic Collision Detection Using Space Partitioning". Trans of ASME, Journal of Mech Design, 1993, vol 115, no 1, pp 150-155

GAR94 Garcia Alonso A, Serrano N, Flaquer J. "Solving the Collision Detection Problem". IEEE Comp Graph and Appl, 1994, vol 14, no 3, pp 36-43

GHO90 Ghosh TK, Batra SK, Baker RL. "The Bending Behaviour of Plain Woven Fabrics". J Text Inst, 1990, vol 81, no 3, pp 245-254, pp 255-271, pp 272-287

GRA92 Gray S. "Virtually Real". Investing in Design by Computer Seminar (Textile Institute), 2 Dec 1992, Royal College of Art, London, UK

GRA93 Gray S. "The Computerised Catwalk". Textile Horizons International, June 1993, pp 32-36

GRE82 Greenough C, Robinson K. "Finite Element Library, Level 1 Documentation". SERC Rutherford Appleton Lab, Didcot, Oxon, 1982

GRO66a Grosberg P, Kedia S. "The Mechanical Properties of Woven Fabrics Part I: The Initial Load-Extension Modulus of Woven Fabrics". Text Res J, 1966, vol 36, no 1, pp 71-79

GRO66b Grosberg P. "The Mechanical Properties of Woven Fabrics Part II: The Bending of Woven Fabrics". Text Res J, 1966, vol 36, no 3, pp 205-211

GRO66c Grosberg P, Swani NM. "The Mechanical Properties of Woven Fabrics Part III: The Buckling of Woven Fabrics". Text Res J, 1966, vol 36, no 4, pp 332-338

GRO66d Grosberg P, Swani NM. "The Mechanical Properties of Woven Fabrics Part IV: The Determination of the Bending Rigidity and Frictional Restraint in Woven Fabrics". Text Res J, 1966, vol 36, no 4, pp 338-345

GRO66e Grosberg P, Park BJ. "The Mechanical Properties of Woven Fabrics Part V: The Initial Modulus and the Frictional Restraint in Shearing of Woven Fabrics". Text Res J, 1966, vol 36, no 5, pp 420-431

GRO68 Grosberg P, Leaf GAV, Park BJ. "The Mechanical Properties of Woven Fabrics Part VI: The Elastic Shear Modulus of Plain Weave Fabrics". Text Res J, 1968, vol 38, no 11, pp 1085-1100

HEA72 Hearle JWS, Konopasek M, Newton A. "On some General Features of a Computer Based System for Calculation of the Mechanics of Textile Structures". Text Res J, 1972, vol 42, pp 613-626

HEA80 Hearle JWS, Thwaites JJ, Amirbayat J (editors). "Mechanics of Flexible Fibre Assemblies". (NATO ASI Series) Sijthoff and Noordhoff, Alpern aan der Rijn, 1980

HEI88a Heisey FL, Haller KD. "Fitting Woven Fabric to Surfaces in Three Dimensions". J Text Inst, 1988, no 2, pp 250-263

HEI88b Heisey F, Brown P, Johnson RF. "Three Dimensional Pattern Drafting: A Theoretical Framework". Clothing Textiles Res J, 1988, vol 6, no 3, pp 1-9

HEI90 Heisey F, Brown P, Johnson RF. "Three Dimensional Pattern Drafting. Part II: Garment Modelling". Text Res J, 1990, vol 60, no 12, pp 731-737

HIN90 Hinds BK, McCartney J. "Interactive Garment Design". The Visual Computer, 1990, vol 6, pp 53-61

HIN91 Hinds BK, McCartney J, Woods G. "Pattern Development for 3D Surfaces". Comp Aided Design, October 1991, vol 23, no 8, pp 583-592

HO88 Ho-Le K. "Finite Element Mesh Generation Methods: a Review and Classification". Comp Aided Design, 1988, vol 10, pp 27-38

HUR64 Hurty WC, Rubenstein MF. "Dynamics of Structures". Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1964

IMA849a Imaoka H, Okabe H, Akani H. "Structure Analysis of the Draped Material by a Computer Simulation". p57-61 in BUL849

IMA849b Imaoka H, Okabe H, Nishikawa S, Shibuya A, Akani H. "A Method of estimating the 3D shapes of Garments by use of Triangular Finite Elements". p73-80 in BUL849

IMA85 Imaoka H, Okabe H, Bessho Y, Akani H. "The Computer System for the Design and the Estimation of Three-Dimensional Shapes of Clothes". ISF-85, 20-24 Aug 1985, Hakone, Japan

JOH92 Johnston BP, Sullivan Jr JM. "Fully Automatic Two Dimensional Mesh Generation Using Normal Offsetting". Int J for Num Methods in Eng, 1992, vol 33, pp 425-442

JON89 Jones PRM, West GM, Harris DH, Read JB. "The Loughborough Anthropometric Shadow Scanner (LASS)". Endeavour, 1989, vol 13, no 4, pp 162-168.

KAM93 Kamat VV. "A Survey of Techniques for Simulation of Dynamic Collision Detection and Response". Comp and Graphics, 1993, vol 17, no 4, pp 379-385

KAW73a Kawabata S, Niwa M, Kawai H. "The Finite Deformation Theory of Plain Weave Fabrics Part I: The Biaxial Deformation Theory". J Text Inst, 1973, vol 64, no 1, pp 21-46

KAW73b Kawabata S, Niwa M, Kawai H. "The Finite Deformation Theory of Plain Weave Fabrics Part II: The Uniaxial Deformation Theory". J Text Inst, 1973, vol 64, no 2, pp 47-61

KAW73c Kawabata S, Niwa M, Kawai H. "The Finite Deformation Theory of Plain Weave Fabrics Part III: The Shear Deformation Theory". J Text Inst, 1973, vol 64, no 2, pp 62-85

KUN90 Kunii TL, Gotoda H. "Singularity Theoretical Modelling and Animation of Garment Wrinkle Formation Process". The Visual Computer, 1990, vol 6, pp 326-336

LAF91 Lafleur B, Magnenat-Thalmann N, Thalmann D. "Cloth Animation with Self-Collision Detection". Proc IFIP Conference on Modelling in Computer Graphics, Springer, Tokyo, 1991, pp 179-187

LAN80 Lane JM, Riesenfeld RF. "A Theoretical Development for Computer Generation and Display of Piecewise Polynomial Surfaces". IEEE Trans on Pattern Analysis and Machine Intelligence, 1980, vol 2, pp 35-45

LIN61 Lindberg J, Behre B, Dahlberg B. "Mechanical Properties of Textile Fabrics Part III: Shearing and Buckling of Various Commercial Fabrics". Text Res J, 1961, vol 31, no 2, pp 99-122

LIN93 Ling L, Damodaran M, Gay RKL. "A Quasi-Steady Force Motion for Animating Cloth Motion". IFIP Transactions B - Applications in Technology, 1993, vol 9, pp 357-363

LIU93 Liu Z. "An Investigation into Computerised Pattern Grading and 3 Dimensional Pattern Representation for Garments". PhD Thesis, University of Leeds, 1993

LLO80a Lloyd DW. "Complex Fabric Deformations, Buckling and Drape". Discussion IV in HEA80a

LLO80b Lloyd DW. "The Analysis of Complex Fabric Deformations" in HEA80a

LOW93 Lowther DA, Rong R, Forghani B. "Field Smoothing and Adaptive Mesh Generation". IEEE Trans on Magnetics, 1993, vol 29, no 2, pp 1890-1893

MAC56 Mack C, Taylor HM. "The Fitting of Woven Cloth to Surfaces". J Text Inst, 1956, vol 47, pp T477-T488

MAG87 Magnenat-Thalmann N, Thalmann D. "The Direction of Synthetic Actors in the Film 'Rendezvous a Montreal'". IEEE Comp Graph and Appl, Dec 1987, pp 9-19

MAG90 Magnenat-Thalmann N, Thalmann D. "Computer Animation: Theory and Practice". Comp Science Workbench; Ed: Tosiyasu L Kunii, Springer-Verlag 1990

MAG91a Magnenat-Thalmann N, Thalmann D. "Complex Models for Animating Synthetic Actors". IEEE Comp Graph and Appl, Sept 1991, pp 32-44

MAG91b Magnenat-Thalmann N, Yang Y. "Techniques for Cloth Animation" in 'New Trends in Animation and Visualisation'. Thalmann N, Thalmann D (eds), Wiley, NY, 1991, pp 243-256

MCC92 McCartney J, Hinds BK. "Computer Aided Design of Garments Using Digitised 3 Dimensional Surfaces". Proc Inst of Mech Eng, Part B - J of Eng Manf 1992, vol 206, no 3, pp 199-206

MEA92 Mealing S. "The Art and Science of Computer Animation". Intellect, Oxford, England, 1992

MON91 Monheit G, Badler NI. "A Kinematic Model of the Human Spine and Torso". IEEE Comp Graph and Appl, March 1991, pp 29-38

MOO88 Moore M, Wilhelms J. "Collision Detection and Response for Computer Animation", ACM SIGGRAPH, Comp Graph 1988, vol 22, no 4, pp 289-298

NEW71 Newmark NM, Rosenblueth E. "Fundamentals of Earthquake Engineering". Prentice-Hall, 1971

NIC71 Nickell RE. "On the Stability of Approximation Operators in Problems of Structural Dynamics". Int J Solids Structures, 1971, vol 7, pp 301-319

NIS87 Nisselson J. "Computer Graphics in Fashion". ACM SIGGRAPH Comp Graph, vol 21, no 4, July 1987, pp 333-334

OHT73 Ohte S. "Finite Element Analysis of Elastic Contact Problems". Bull J Soc Mech Eng, vol 16, no 95, May 1973, pp 798-804

OKA849 Okabe H, Akami H. "The Estimation of the 3D shapes of Garments". p63-71 in BUL849

OLO64 Olofsson B. "A General Model of a Fabric as a Geometric-Mechanical Structure". J Text Inst, 1964, vol 55, pp T541-T557

PEI37 Peirce FT. "The Geometry of Cloth Structure". J Text Inst, 1937, vol 28, T45-T96

PRZ68 Przemieniecki JS. "Theory of Matrix Structural Analysis". McGraw-Hill Inc, 1968

RAN93 Rank E, Schweingruber M, Sommer M. "Adaptive Mesh Generation and Transformation of Triangular and Quadrilateral Meshes". Comm in Num Methods in Eng, 1993, vol 9, no 2, pp 121-129

REB93 Rebay S. "Efficient Unstructured Mesh Generation by Means of Delauney Triangulation and Bowyer-Watson Algorithm". J Computational Physics, 1993, vol 106, no 1, pp 125-138

ROB81 Robertson RE, Hsiue ES, Sickafus EN, Yeh GSY. "Fiber Rearrangements During the Molding of Continuous Fiber Composites. I. Flat Cloth to Hemisphere". Polymer Composites, 1981, vol 2, pp 126-131

ROB84 Robertson RE, Hsiue ES, Yeh GSY. "Continuous Fiber Rearrangements During the Molding of Fiber Composites. II. Flat Cloth to a Rounded Cone". Polymer Composites, 1984, vol 5, pp 191-197

SCA90 Scanlon B, "Animating the Drape of Cloth". MSc Thesis, Teeside Polytechnic, September 1990

SHA78 Shanahan WJ, Lloyd DW, Hearle JWS. "Characterising the Elastic Behaviour of Textile Fabrics in Complex Deformations". Text Res J,1978, vol 48, pp 495-505

TER87 Terzopoulos D, Platt J, Barr A, Fleischer K. "Elastically Deformable Models". ACM SIGGRAPH'87, Anaheim, Comp Graph, July 27-31 1987, vol 21, no 4, pp 205-214

TER88 Terzopoulos D, Witkin A. "Physically Based Models with Rigid and Deformable Components". Proc Graphics Interface 88, Edmonton, Canada, June 1988, pp 146-154

TIM70 Timoshenko SP, Goodie JP. "Theory of Elasticity". 3rd Edition, McGraw-Hill, 1970

TRE65 Treloar LRG. "The Effect of Test Piece Dimensions on the Behaviour of Fabrics in Shear". J Text Inst, 1965, vol 56, T533-T550

VAN90 Van West BP, Pipes RB, Keefe M. "A Simulation of the Draping of Bidirectional Fabrics over Arbitrary Surfaces". J Text Inst, 1990, vol 81, no 4, pp 448-460

VAN91 Van der Weeën F. "Algorithms for Draping Fabrics on Doubly Curved Surfaces". Int J for Num Methods in Eng, 1991, vol 31, pp 1415-1426

WEI86 Weil J. "The Synthesis of Cloth Objects". ACM Comp Graph, 1986, vol 20, no 4, pp 49-54

WIL72 Wilson EL, Penzien J. "Evaluation of Orthogonal Damping Matrices". Int J for Numerical Methods in Engineering, 1972, vol 4, pp 5-10

WYV87 Wyvill G, Wyvill B, McPheeters C. "Solid Texturing of Soft Objects". IEEE Comp Graph and Appl, Dec 1987, pp 20-26

YAN91 Yang J, Zhu H. "3D Modelling of Computer Aided Garment Design". J of China Textile University (Eng Ed), 1991, vol 8, no 1, pp 7-10

YAN92 Yang Y, Magnenat-Thalmann N, Thalmann D. "3 Dimensional Garment Design and Animation - A New Design Tool for the Garment Industry". Computers in Industry, 1992, vol 19, no 2, pp 185-191 and pp 195-196

BIBLIOGRAPHY

Aldrich W. "Metric Pattern Cutting". Blackwell Scientific Publications, reprinted 1993.

Argyris JH. "Recent Advances in Matrix Methods of Structural Analysis". Progress in Aeronautical Sciences, vol IV, The Macmillan Company, New York, 1964

Bedworth D, Henderson MR, Wolfe PM. "Computer Integrated Design and Manufacturing". McGraw-Hill Inc, 1991

Besant CB, Lui CWK. "Computer Aided Design and Manufacture". 3rd Edition, Ellis Horwood Ltd, 1986

Clough RW, Penzien J. "Dynamics of Structures". McGraw-Hill, 1975

Cook RD. "Concepts and Applications of Finite Element Analysis". Wiley, New York, 1974

Dawson R. "Programming in Ansi C". Group D Publications, Loughborough, UK, 1993

Encaraçâo JL, Lindner R, Schlechtendahl EG. "Computer Aided Design Fundamentals and System Architectures". 2nd Edition, Springer Verlag, 1990

Foley JD, van Dam A. "Fundamentals of Interactive Computer Graphics". Addison-Wesley, Reading, Massachusetts, 1982

Greenough C, Robinson K. "Finite Element Library, Level 0 Documentation". SERC Rutherford Appleton Lab, Didcot, Oxon, 1981

Haigh MJ. "An Introduction to Computer Aided Design and Manufacture". Blackwell Scientific Publications, 1985

Heubner KH. "The Finite Element Method for Engineers". 2nd Edition, Wiley 1982

Ingham P. "CAD Systems in Mechanical and Production Engineering". Heinemann Newnes, Oxford, 1989

Kaplinsky R. "Computer Aided Design. Electronics, Comparative Advantage and Development". Frances Pinter Ltd, UK, 1982

Kernighan BW, Ritchie DM. "The C Programming Language". Prentice-Hall, NJ, USA, 1988

Lasseter J. "Principles of Traditional Animation Applied to 3D Computer Animation". ACM SIGGRAPH'87, Anaheim, July 27-31 1987, Comp Graphics, vol 21, no 4, pp35-44

Livesley RK. "Matrix Methods of Structural Analysis". Pergamon Press, 1964

McMahon C, Browne J. "CAD/CAM From Principles to Practice". Addison-Wesley, 1993

Mortenson ME. "Computer Graphics: An Introduction to the Mathematics and Geometry". Heinemann Newnes, 1989

Page C."The Professional Programmer's Guide to FORTRAN 77". Pitman, London,1988

Ryder GH. "Strength of Materials". 3rd Edition, Macmillan and Co Ltd, 1965

Silicon Graphics Computer Systems. "Graphics Library Reference Manual C Edition". Iris 4D Series, Doc no 007-1203-030, 1989

Silicon Graphics Computer Systems. "Graphics Library Programming Guide". Iris 4D Series, Doc no 007-1210-010, 1989

Zienkiewicz OC. "The Finite Element Method". Third Edition, McGraw-Hill, 1977

APPENDIX I - BODY MODEL TRANSLATION PROGRAMS

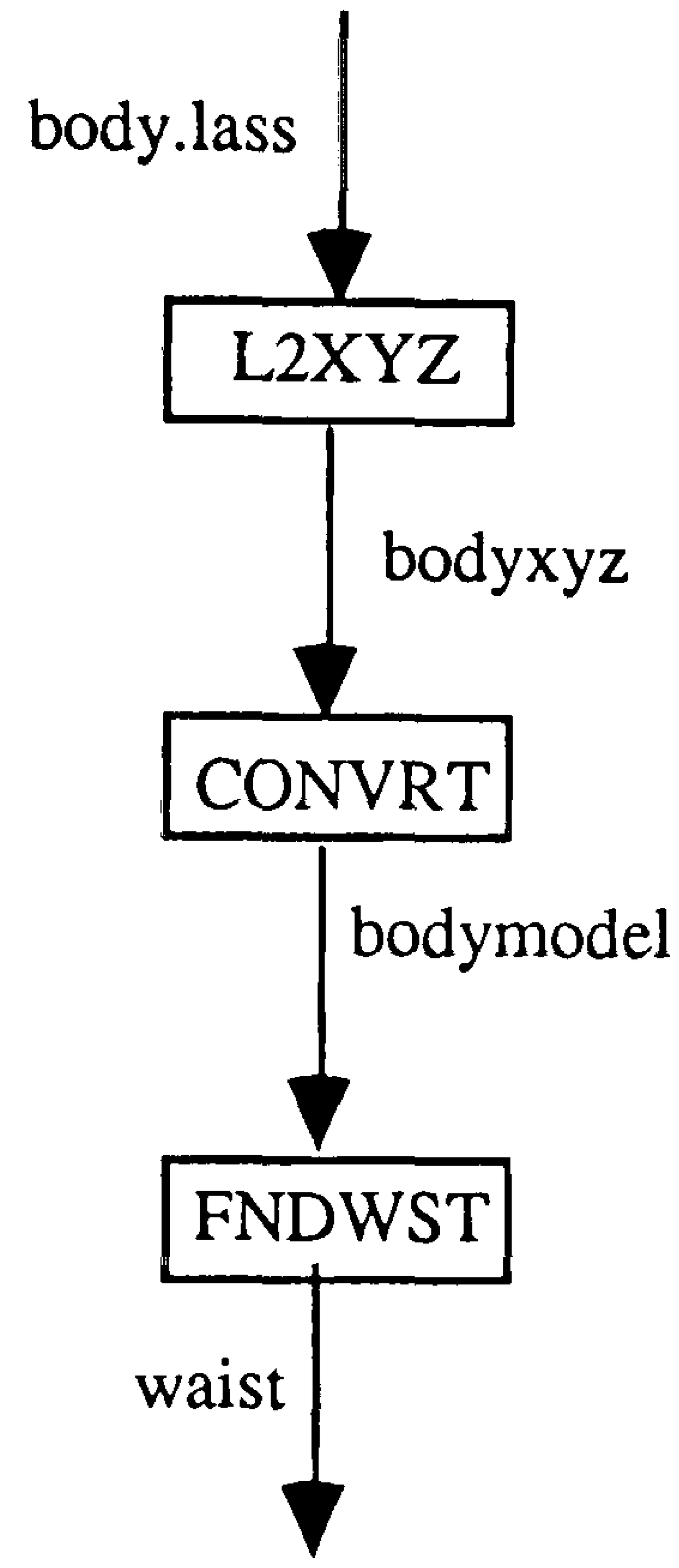


Figure I.1 - body model translation programs

I.1 Programs

There are three programs involved in translating the LASS body model into the draping system co-ordinates. The first reads in the LASS half body data ("body.lass") and generates a full body listing ("bodyxyz"). The second translates this listing into system co-ordinates to provide a complete list of body co-ordinates for use in the DRAPE and DISPLAY programs ("bodymodel"). The third generates a set of waist co-ordinates for use in the MSHGEN program ("waist"). All three programs are written in standard FORTRAN 77.

L2XYZ	Converts LASS formatted data for half body model into x-y-z listing of complete body model.
CONVRT	Converts body model data into drape system coordinates.
FNDWST	Creates listing of waist coordinates.

I.2 Variable Listing

(dp = double precision; int = integer)

COORD	array of x, y and z co-ordinates for cloth mesh (dp)
DIVIDE	scale value for LASS to DRAPE conversion (dp)
FINISH	final number in node list (int)
I, J, K, L	count (int)
NIN	input channel number for reading files (int)
NODNUM	node number (int)
NODSLC	number of nodes per slice of body data (int)
NOUT	output channel number for writing files (int)
NUMDOF	number of degrees of freedom (int)
START	starting number in node list (int)
TCOORD	vector of LASS data values (dp)
TOTNOD	total number of body nodes (int)
TOTSLC	total number of body data slices (int)
WSTLVL	slice number referring to waist (int)

I.3 Program Listings

```
c L2XYZ converts the body data from the LASS format into a full xyz listing for
c each body node.  The LASS format consists of slices of 16 pairs of x and y
c coords = 32 values per slice; the 33rd value is the height for that slice, the 34th
c value can be ignored.

c last updated: 1.11.94

      PROGRAM L2XYZ

      INTEGER I,J,K,L,NIN,NODNUM,NODSLC,NOUT,TOTNOD, TOTSLC
      DOUBLE PRECISION COORD,TCOORD
      DIMENSION COORD(1500,3),TCOORD(34)
      DATA NIN /3/,NODSLC /16/,NOUT /4/

c i/o data files:

      OPEN(UNIT=NIN,FILE='body.lass',STATUS='OLD')
      OPEN(UNIT=NOUT,FILE='bodyxyz',STATUS='OLD')

c read in total number of body data slices:

      READ(NIN,100) TOTSLC

c for each slice, pick out each xy pair and team up with the height for that slice
c (the z coord):

      DO 30 I=1,TOTSLC
       READ(NIN,200) (TCOORD(J),J=1,(NODSLC+1)*2)
       K=0
       L=TOTNOD+(2*NODSLC)-1
       DO 20 J=1,NODSLC
        TOTNOD=TOTNOD+1
        K=K+1
        IF (J.NE.1.AND.J.NE.NODSLC) L=L-1
        COORD(TOTNOD,1)=TCOORD(K)
        IF (J.NE.1.AND.J.NE.NODSLC) COORD(L,1)=-TCOORD(K)
        K=K+1
        COORD(TOTNOD,2)=TCOORD(K)
        IF (J.NE.1.AND.J.NE.NODSLC) COORD(L,2)=TCOORD(K)
        COORD(TOTNOD,3)=TCOORD(33)
        IF (J.NE.1.AND.J.NE.NODSLC) COORD(L,3)=TCOORD(33)
20     CONTINUE
       TOTNOD=TOTNOD+NODSLC-2
30    CONTINUE

c write out full body model node listing:

      WRITE(NOUT,100) TOTSLC,TOTNOD
      DO 40 NODNUM=1,TOTNOD
       WRITE(NOUT,300) NODNUM,(COORD(NODNUM,I),I=1,3)
40    CONTINUE
```

```
100  FORMAT(2I5)
200  FORMAT(34F8.0)
300  FORMAT(I5,3F10.3)
     END
```

```
c CONVRT translates the body model data from mm into m.  It also moves the
c body slightly over to centre it around the main z axis.

c last updated 1.11.94

     PROGRAM CONVRT
     INTEGER NIN,NOUT,NODNUM,NUMDOF,TOTNOD,TOTSLC
     DOUBLE PRECISION COORD,DIVIDE
     DIMENSION COORD(2000,3)
     DATA DIVIDE /1.0D+3/,NIN /4/,NOUT /5/,NUMDOF /3/

c i/o data files:

     OPEN(UNIT=NIN,FILE='bodyxyz',STATUS='OLD')
     OPEN(UNIT=NOUT,FILE='bodymodel',STATUS='OLD')

     READ(NIN,1000) TOTSLC,TOTNOD
     DO 100 I=1,TOTNOD
      READ(NIN,2000) NODNUM,(COORD(I,J),J=1,NUMDOF)
100  CONTINUE

c divide by 1000 to get from mm into m:

     DO 600 I=1,TOTNOD
      DO 600 J=1,3
       COORD(I,J)=COORD(I,J)/DIVIDE
600  CONTINUE

c write output file:

     WRITE(NOUT,1000) TOTSLC,TOTNOD
     DO 800 I=1,TOTNOD
      WRITE(NOUT,3000) (COORD(I,J),J=1,3)
800  CONTINUE

1000 FORMAT(2I5)
2000 FORMAT(I5,3F10.0)
3000 FORMAT(3F10.5)
     END
```

APPENDIX II - MESH GENERATION PROGRAM

CALCW
|
SETW
|
ADOFST
|
ADDN
|
MODMSH
|
ADDEL

Figure II.1 - Program MSHGEN

II.1 Subroutines

The MSHGEN program creates the cloth mesh using the body model information read in from file "waist". The measured physical properties are read in from file "samples" and the resulting node listing, element listing, restrained nodes and physical properties are written to the file "data". The program is written in standard FORTRAN 77.

ADDEL	Creates element listing of the relationships between each cloth node and its neighbours.
ADDN	Adds nodes to the mesh depending on the initial layout of the waist node data.
ADOFST	Adds offset to cloth waist nodes.
CALCW	Calculates the length around the waist based on the information from the body model data.
MODMSH	Renumbers nodes to improve efficiency of mesh.
SETW	Calculates the cloth waist node positions using the body model waist data as abase.

II.2 Variable Listing

(dp = double precision; int = integer)

ALPHA	Rayleigh damping coefficient (dp)
ANGLE	angle value (dp)
BASE	defines one of two rows of nodes for element creation (int)
BETA	Rayleigh damping coefficient (dp)
BWARP	bending value for cloth sample - warp (dp)
BWEFT	bending value for cloth sample - weft (dp)
COORD	global co-ordinates of nodes (dp)
COUNTW	count (int)
DIFFY	value to centre waist nodes (dp)
DIST	distance of waist node from centre of body (dp)
DIMEN	dimensionality of problem (int)
DOFNOD	number of degrees of freedom per node (int)
DTIM	time step length (dp)
ELELST	updated efficient element reference list (int)
ELEM	element listing (int)
ELNUM	element number (int)
FABRIC	cloth sample reference number (int)
FLAGW	count (int)
I	count (int)
J, K	count (int)
LEFT	node count (int)
LENGTH	length of cloth (dp)
LEVEL	defines one of two rows of nodes for element creation (int)

MASS	mass per unit are of cloth sample (dp)
NDIST	value for distance between nodes (dp)
NIN	input channel number for file "waist" (int)
NMAT	input channel number for file "matprop" (int)
NODE	node number (int)
NODEL	number of nodes per element (int)
NODLST	updated efficient node reference list (int)
NODNUM	node number (int)
NOUT	output channel number for file "data" (int)
NSTEPS	number of time steps to be performed (int)
OFFSET	distance of cloth waist nodes from body model waist (dp)
OPT1/2/3/4	material property selections for bending (warp and weft), density and thickness (int)
RESNOD	number of restrained nodes (int)
RIGHT	node count (int)
THETA	scheme parameter for direct integration techniques (dp)
THICKN	thickness of cloth sample (dp)
TOTELS	total number of elements (int)
TOTMAT	total number of materials in database (int)
TOTNOD	total number of nodes (int)
TOTROW	total number of rows of nodes (int)
TOTW	total number of cloth mesh nodes around the waist (int)
WARP	value for distance between nodes in length (dp)
WCOORD	body model waist node co-ordinates (dp)
WDIST	length around waist (dp)
WEFT	value for distance between nodes around waist (dp)
WNUM	number of nodes in each row of cloth mesh (int)
X, Y	co-ordinates for node (dp)
ZLIM	minimum point of Z (dp)

II.3 Program Listing

```
c MSHGEN reads in body model waist data and calculates a mesh to represent a
c circular skirt.

c units = m, kg, s

c last updated: 1.11.94

      PROGRAM MSHGEN

      INTEGER DIMEN,DOFNOD,ELELST,ELEM,ELNUM,FABRIC,I,J,NIN,
     *    NMAT,NODEL,NODLST,NODNUM,NOUT,NSTEPS,RESNOD,
     *    TOTELS,TOTMAT,TOTNOD,TOTROW,TOTW,WNUM

      DOUBLE PRECISION ALPHA,BETA,BWARP,BWEFT,COORD,DIFFY,
     *    DTIM,LENGTH,MASS,OPT1,OPT2,OPT3,OPT4,THETA,THICKN,
     *    WARP,WCOORD,WDIST,WEFT,ZLIM

      DIMENSION COORD(3000,3),ELELST(3000),ELEM(10000,2),
     *    NODLST(3000),WCOORD(100,3)

      DATA ALPHA /0.45/,BETA /1.67/,DIMEN /3/,DOFNOD /6/,ELTYP /1/,
     *    NODEL /2/,THETA /0.5/,TOTNOD /1/,WNUM /30/

c length of skirt (m), number of skirt nodes per row, node spacing in warp
c direction (m) - all user-defined:

      DATA LENGTH /4.0D-01/,TOTW /100/,WARP /5.D-02/

c time step length and number of time steps to be performed - user-defined:

      DATA DTIM /1.D-01/,NSTEPS /300/

c i/o files:

      DATA NIN /2/,NOUT /3/,NMAT /4/

      OPEN(UNIT=NIN,FILE='waist',STATUS='OLD')
      OPEN(UNIT=NMAT,FILE='samples',STATUS='OLD')
      OPEN(UNIT=NOUT,FILE='data',STATUS='OLD')

c read in waist information from body:

      DO 10 I=1,WNUM
       READ(NIN,9080) (WCOORD(I,J),J=1,3)
10    CONTINUE

c measure body waist, calculate weft direction node spacing and number of rows
c in warp direction:

      CALL CALCW(WCOORD,WDIST,WNUM)
      WEFT=WDIST/TOTW
      ZLIM=WCOORD(1,3)
```

```
      TOTROW=LENGTH/WARP

c calculate coordinates of skirt nodes:

      CALL SETW(COORD,DIFFY,RESNOD,TOTNOD,TOTW,WCOORD,
     *  WEFT,WNUM,ZLIM)
      CALL ADOFST(COORD,TOTNOD)
      CALL ADDN(COORD,TOTNOD,TOTROW,TOTW,WARP,ZLIM)

      DO 20 NODNUM=1,TOTNOD
       COORD(NODNUM,2)=COORD(NODNUM,2)+DIFFY
20    CONTINUE

c renumber node list for more efficient mesh and write out node data:

      CALL MODMSH(NODLST,TOTROW,TOTW)
      WRITE(NOUT,9020) TOTNOD,DIMEN
      DO 30 NODNUM=1,TOTNOD
       WRITE (NOUT,9030) NODNUM,(COORD(NODLST(NODNUM),I),I=1,3)
30    CONTINUE

c determine element data and write to output file:

      CALL ADDEL(ELEM,TOTELS,TOTNOD,TOTROW,TOTW)
      DO 40 NODNUM=1,TOTNOD
       ELELST(NODLST(NODNUM))=NODNUM
40    CONTINUE

      WRITE(NOUT,9020) ELTYP,TOTELS,NODEL
      DO 50 ELNUM=1,TOTELS
       WRITE(NOUT,9020) ELNUM,ELELST(ELEM(ELNUM,1)),
     *  ELELST(ELEM(ELNUM,2))
50    CONTINUE

c establish fabric selection:

      WRITE(*,*) "1.  Polyester"
      WRITE(*,*) "2.  Cotton"
      WRITE(*,*) "3.  Wool"
      WRITE(*,*) "4.  Polycotton"
      WRITE(*,*) "Enter material selection:"
      READ(*,*)  FABRIC

c assign correct values for chosen fabric and write to output file:

      READ(NMAT,9020) TOTMAT
      DO 60 I=1,TOTMAT
       READ(NMAT,*) SELECT,OPT1,OPT2,OPT3,OPT4
       IF (SELECT.EQ.FABRIC) THEN
        BWARP=OPT1
        BWEFT=OPT2
        MASS=OPT3
        THICKN=OPT4
       END IF
60    CONTINUE
```

```
      WRITE(NOUT,9020) FABRIC
      WRITE(NOUT,9060) BWARP,BWEFT,MASS,THICKN
      WRITE(NOUT,9030) NSTEPS,DTIM,THETA
      WRITE(NOUT,9060) ALPHA,BETA

c write out listing of restrained degrees of freedom for waist nodes:

      WRITE(NOUT,9020) DOFNOD
      WRITE(NOUT,9020) RESNOD
      DO 70 NODNUM=1,RESNOD
       WRITE(NOUT,9020) ELELST(NODNUM),1,2,3
70    CONTINUE

c write out node links for display program:

      WRITE(NOUT,9020) TOTROW,TOTW
      DO 80 NODNUM=1,TOTNOD
       WRITE(NOUT,*) ELELST(NODNUM)
80    CONTINUE

9020  FORMAT (7I5)
9030  FORMAT (I5, 3F10.5)
9060  FORMAT (6D15.7)
9080  FORMAT(3F10.0)
      END


c measure waist:

      SUBROUTINE CALCW(WCOORD,WDIST,WNUM)
      INTEGER I,J,WNUM
      DOUBLE PRECISION WCOORD,WDIST,X,Y
      DIMENSION WCOORD(100,3)
      DO 10 I=1,WNUM
       IF (I.EQ.WNUM) J=1
       IF (I.NE.WNUM) J=I+1
       X=WCOORD(I,1)-WCOORD(J,1)
       Y=WCOORD(I,2)-WCOORD(J,2)
       WDIST=WDIST+DSQRT(X**2+Y**2)
10    CONTINUE
      END


c calculate skirt waist nodes:

      SUBROUTINE SETW(COORD,DIFFY,RESNOD,TOTNOD,TOTW,
     *  WCOORD,WEFT,WNUM,ZLIM)
      INTEGER COUNTW,FLAGW,I,J,K,RESNOD,TOTNOD,TOTW,WNUM
      DOUBLE PRECISION ANGLE,COORD,DIFFY,NDIST,WCOORD,WEFT,
     *  X,Y,ZLIM
      DIMENSION COORD(3000,3),WCOORD(100,3)
      DIFFY=WCOORD(1,2)+(ABS(WCOORD(1,2))+WCOORD(16,2))/2
      DO 5 I=1,WNUM
       WCOORD(I,2)=WCOORD(I,2)-DIFFY
5     CONTINUE
```

```
      DO 10 I=1,3
       COORD(TOTNOD,I)=WCOORD(TOTNOD,I)
10    CONTINUE
      COORD(TOTNOD,1)=COORD(TOTNOD,1)-(WEFT/100)
      COUNTW=2
      DO 40 I=2,TOTW
       FLAGW=0
       DO 30 J=COUNTW,WNUM
        IF (COUNTW.EQ.WNUM) K=1
        IF (COUNTW.NE.WNUM) K=J
        IF (K.EQ.0) K=1
        IF (FLAGW.EQ.0) THEN
         X=COORD(I-1,1)-WCOORD(K,1)
         Y=COORD(I-1,2)-WCOORD(K,2)
         NDIST=DSQRT(X**2+Y**2)
         IF (NDIST.GT.WEFT) FLAGW=K
        END IF
30     CONTINUE
       COUNTW=FLAGW
       ANGLE=DATAN(ABS(X)/ABS(Y))
       TOTNOD=TOTNOD+1
       COORD(TOTNOD,1)=
     *  COORD(TOTNOD-1,1)-(X/ABS(X)*WEFT*DSIN(ANGLE))
       COORD(TOTNOD,2)=
     *  COORD(TOTNOD-1,2)-(Y/ABS(Y)*WEFT*DCOS(ANGLE))
       COORD(TOTNOD,3)=ZLIM
40    CONTINUE
      COORD(1,1)=(0.5*COORD(TOTNOD,1))+(0.5*COORD(2,1))
      COORD(1,2)=(0.5*COORD(TOTNOD,2))+(0.5*COORD(2,2))
      RESNOD=TOTNOD
      END


c add offset to waist coords so skirt waist nodes do not collide with body waist:

      SUBROUTINE ADOFST(COORD,TOTNOD)
      INTEGER NODE,TOTNOD
      DOUBLE PRECISION ANGLE,COORD,DIST,OFFSET,X,Y
      DIMENSION COORD(3000,3)
      OFFSET=1.D-02
      DO 10 NODE=1,TOTNOD
       X=COORD(NODE,1)
       Y=COORD(NODE,2)
       IF (X.EQ.0) THEN
        COORD(NODE,2)=Y+(Y/ABS(Y)*OFFSET)
       ELSE IF (Y.EQ.0) THEN
        COORD(NODE,1)=X+(X/ABS(X)*OFFSET)
       ELSE
        ANGLE=DATAN(ABS(Y/X))
        DIST=DSQRT(X**2+Y**2)+OFFSET
        COORD(NODE,1)=X/ABS(X)*DCOS(ANGLE)*DIST
        COORD(NODE,2)=Y/ABS(Y)*DSIN(ANGLE)*DIST
       END IF
10    CONTINUE
      END
```

c calculate remaining nodes for skirt mesh:

```
      SUBROUTINE ADDN(COORD,TOTNOD,TOTROW,TOTW,WARP,ZLIM)
      INTEGER I,J,TOTNOD,TOTROW,TOTW
      DOUBLE PRECISION ANGLE,COORD,WARP,X,Y,ZLIM
      DIMENSION COORD(3000,3)
      DO 30 I=1,TOTROW
       DO 10 J=1,TOTW
        X=COORD(J,1)
        Y=COORD(J,2)
        TOTNOD=TOTNOD+1
        DIST=DSQRT(X**2+Y**2)+(I*WARP)
        IF (Y.EQ.0) THEN
         COORD(TOTNOD,1)=DIST*X/ABS(X)
         COORD(TOTNOD,2)=0
        ELSE IF (X.EQ.0) THEN
         COORD(TOTNOD,1)=0
         COORD(TOTNOD,2)=DIST*Y/ABS(Y)
        ELSE
         ANGLE=DATAN(ABS(Y/X))
         COORD(TOTNOD,1)=DCOS(ANGLE)*DIST*X/ABS(X)
         COORD(TOTNOD,2)=DSIN(ANGLE)*DIST*Y/ABS(Y)
        END IF
        COORD(TOTNOD,3)=ZLIM
10     CONTINUE
30    CONTINUE
      END
```

c renumber mesh for improved efficiency in draping calcs:

```
      SUBROUTINE MODMSH(NODLST,TOTROW,TOTW)
      INTEGER COUNT,I,J,K,LEFT,NODE,NODLST,RIGHT,TOTROW,TOTW
      DIMENSION NODLST(3000)
      LEFT=0
      RIGHT=TOTW+1
      COUNT=0
      DO 20 I=1,(TOTW+1)/2
       LEFT=LEFT+1
       RIGHT=RIGHT-1
       DO 20 J=1,2
        IF (J.EQ.1) NODE=LEFT
        IF (J.EQ.2) NODE=RIGHT
        IF (J.EQ.1.OR.LEFT.NE.RIGHT) THEN
         DO 10 K=1,TOTROW+1
          COUNT=COUNT+1
          NODLST(COUNT)=NODE+((K-1)*TOTW)
10        CONTINUE
        END IF
20    CONTINUE
      END
```

```
c determine elements:

      SUBROUTINE ADDEL(ELEM,TOTELS,TOTNOD,TOTROW,TOTW)
      INTEGER BASE,ELEM,I,J,LEVEL,TOTELS,TOTNOD,TOTROW,TOTW
      DIMENSION ELEM(10000,2)
      DO 20 I=1,TOTROW+1
      BASE=(I-1)*TOTW
       DO 10 J=1,TOTW-1
        TOTELS=TOTELS+1
        ELEM(TOTELS,1)=J+BASE
        ELEM(TOTELS,2)=J+BASE+1
10      CONTINUE
       TOTELS=TOTELS+1
       ELEM(TOTELS,1)=TOTW+BASE
       ELEM(TOTELS,2)=1+BASE
20     CONTINUE
      DO 30 I=1,TOTROW
      BASE=(I-1)*TOTW
      LEVEL=I*TOTW
       DO 30 J=1,TOTW
        TOTELS=TOTELS+1
        ELEM(TOTELS,1)=J+BASE
        ELEM(TOTELS,2)=J+LEVEL
30     CONTINUE
      END
```

APPENDIX III - DRAPE CALCULATION PROGRAM

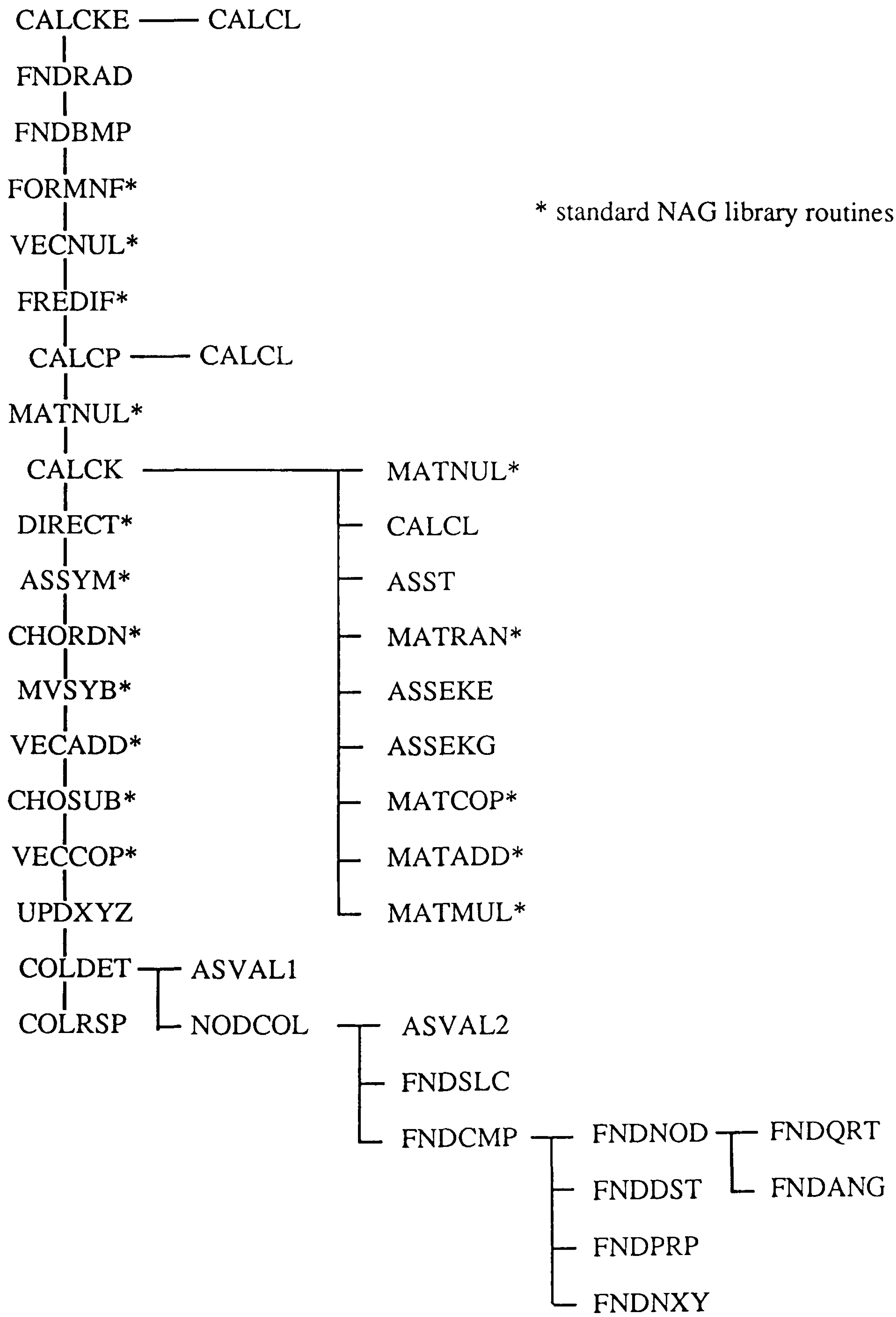


Figure III.1 - Program DRAPE

III.1 Subroutines

The DRAPE program reads in the mesh information from the file "data" and the body model co-ordinates from the file "bodymodel", calculates the nodal displacements by Newmark's Direct Integration Method and outputs the results to the file "coord". The program is written in standard FORTRAN 77 and uses routines from the NAG Library.

ASSEKE	Assigns the values to the element elasticity matrix ELKE.
ASSEKG	Assigns values to the element geometrical matrix ELKG.
ASST	Assigns values to the transformation matrix T.
ASVAL1 - 2	Assigns co-ordinate variables.
ASSYM	Assembles system matrices from element matrix contributions.
CALCK	Calculates element stiffness matrices.
CALCKE	Calculates ELK constants.
CALCL	Calculates the current length of an element.
CALCP	Calculates the load vector.
CHORDN	Performs Choleski reduction.
CHOSUB	Performs forward and backward substitution on a matrix reduced by CHORDN.
COLDET	Determines if cloth nodes have collided with body model.
COLRSP	Modify loads if collisions have occurred.
DIRECT	Constructs the steering vector STEER to direct the assembly of the system stiffness or mass matrix.
FNDANG	Finds angle value.
FNDBMP	Maps body model onto cylinder.
FNDCMP	Maps cloth nodes around cylinder.
FNDCOL	Finds nodes which have entered into the body model volume.
FNDDST	Finds distance of body node to cylinder.
FNDNOD	Finds body nodes on either side of cloth node.
FNDNXY	Finds new values for x and y co-ordinate.
FNDPRP	Finds angle formed by cloth and body nodes.
FNDQRT	Finds location of node in one of the four quarters.
FNDSLC	Finds location of node between two slices of body data.
FORMNF	Constructs the nodal freedom array NF from the restrained freedom matrix RESTR.

FREDIF	Calculates the maximum freedom number difference for a specified element.
MATADD	Adds one matrix to another.
MATCOP	Copies contents of one matrix to another.
MATMUL	Multiplies two matrices together.
MATNUL	Sets a specified matrix to zero.
MATRAN	Generates transform of a matrix.
MVSYB	Post multiplies a matrix by a vector.
NODCOL	Finds if cloth node has collided with body model cylinder map.
VECADD	Adds two vectors together.
VECCOP	Copies the contents of one vector into another.
VECNUL	Sets a specified vector to zero.
UPDXYZ	Updates the coordinate values of each node at each time step.

III.2 Variable Listing

(dp = double precision; int = integer; log = logical)

A	cross-sectional area of element (dp)
A0, A1	vector containing displacements (dp)
AB, AN	angle of nodes with origin (dp)
ALPHA	Rayleigh damping coefficient (dp)
ANGLE,AA,AB, AC,AD	angle reference value (dp)
BASE	element pitch - weft (dp)
BCOORD	listing of body model nodal co-ordinates (dp)
BETA	Rayleigh damping coefficient (dp)
BODYMP	matrix containing mapped body data (dp)
BRAD	comparison value for body radius (dp)
BWARP	bending value for cloth sample - warp (dp)
BWEFT	bending value for cloth sample - weft (dp)
C1-C7	constant values assigned for solution calculations (dp)
COLLID	flag for collision (log)
COORD	matrix containing global coordinates of each node (dp)

COUNT	count (int)
D2A0, D2A1	vectors containing accelerations (dp)
DA0, DA1	vectors containing velocities (dp)
DAX/AY/ BX/BY	displacement values for nodes (dp)
DIF	maximum freedom number difference for an element (int)
DIMEN	dimensionality of problem (int)
DOFEL	number of degrees of freedom per element (int)
DOFNOD	number of degrees of freedom per node (int)
DTIM	length of time step (dp)
E	Young's modulus (dp)
E1-E10	constants for elasticity stiffness matrix (dp)
ELELST	updated efficient element reference list (int)
ELEML	element length (dp)
ELK	element stiffness matrix (dp)
ELKE	element elasticity matrix (dp)
ELKG	element geometrical matrix (dp)
ELM	element mass matrix (dp)
ELNUM	element number (int)
ELTOP	element topology array (int)
ELTYP	type of element (int)
FABRIC	reference number of cloth sample (int)
FIRST	flags on first call to subroutine FREDIF (log)
FINISH	ending node number in collision check with body model (int)
FREEDM	vector containing number of degrees of freedom of each node (int)
FREDMA	freedom reference number (int)
G1-G4	value equations for geometrical stiffness matrix (dp)
GRAVTY	acceleration due to gravity (dp)
HBAND	semi-bandwidth of system matrices (int)
HEIGHT	element pitch - warp (dp)

I, II	count (int)
ICOORD	row limit of co-ordinate matrices (int)
IELTOP	row limit of matrix ELTOP (int)
ISTEER	row limit for vector STEER (int)
ISYS	row limit for system matrices and vectors (int)
ITEST	error indicator and control (int)
IY,IZ	2nd moments of inertia (dp)
J	count (int); polar moment of inertia (dp)
JCOORD	column limit for co-ordinate matices (int)
JELTOP	column limit for matrix ELTOP (int)
JNF	column limit for matrix NF (int
JRESTR	column limit for matrix RESTR (int)
JSYS	column limit for system matrices and vectors (int)
K	count (int)
KE	ELKE matrix constants (dp)
L	element length (dp); transformation matrix constant (dp)
LOADS	load vector (dp)
LX/LY/LZ	length of element projected onto respective axis (dp)
N	transformation matrix contant (dp)
NA/NB/NA1/NA2/NB1/ NB2/N1/N2	node reference numbers (int)
NBODY	input channel number for "bodymodel" (int)
NEWC	nodal co-ordinate values at previous step (dp)
NEWTOT	updated TOTNOD value (int)
NF	nodal freedom array (int)
NIN	input channel number for "data" (int)
NODEL	number of nodes per element (int)
NODNUM	node number (int)
NODSLC	number of nodes per slice of body model data (int)
NOUT	output channel number for file "coord" (int)
NSTEPS	number of time steps (int)
NX,NY	x and y co-ordinates (dp)

P	internal force of element (dp); vector of forces (dp)
PREVL	vector of previous step's element lengths (dp)
QB,QN	node quarter reference number (int)
RAD/C	value of node distance from origin (dp)
RESNOD	number of restrained nodes (int)
RESTR	restraine data array (int)
RSTRND	vector containing flags if node has collided (int)
SA/SB/SC	values for triangle sides (dp)
SLICEA/B	slice number above cloth node (int)
START	starting body model node number in collision check (int)
STEER	steering vector (int)
STEP	count for number of time steps (int)
SYSK	system stiffness matrix (dp)
SYSM	system mass matrix (dp)
SYSW	system work matrix (dp)
T	transformation matrix (dp)
TELK	element stiffness matrix - partly transformed (dp)
TELKTT	element stiffness matrix in global coordinates (dp)
THETA	scheme parameter for direct integration techniques (dp)
THICKN	thickness of cloth sample (dp)
TIME	count (int)
TOTBOD	total number of nodes in the body model (int)
TOTDOF	total number of degrees of freedom (int)
TOTELS	total number of elements (int)
TOTNOD	total number of nodes (int)
TOTROW	total number of nodal rows in the cloth mesh (int)
TOTSLC	total number of slices of body model data (int)
TOTW	number of nodes in each row in the cloth mesh (int)
TT	transformation of T (dp)
U	change in length of element (dp)
UX/UY	co-ordinate values for body nodes (dp)

V	Poisson's ratio (dp)
WORK	general work matrix (dp)
X,XX	x co-ordinate value (dp)
XAXIS	flags if node on x-axis (log)
XYZ1/2/3/E/F/G	nodal co-ordinate vector listing (dp)
Y,YY	y co-ordinate value (dp)
YAXIS	flags if node on y-axis (log)
Z	z coordinate value (dp)
ZSLICE	listing of z value for each slice of the body model data (int)

III.3 Program Listing

```
c DRAPE calculates the drape of cloth over a series of time steps.  Collision
c detection and response is carried out in relation to the underlying body model.

c units = m, kg, s

c last updated: 14.11.94

      PROGRAM DRAPE

      INTEGER DIF,DIMEN,DOFEL,DOFNOD,ELELST,ELNUM,ELTOP,
     *    ELTYP,FABRIC,FREEDM,HBAND,I,ICOORD,IELTOP,ISTEER,
     *    ISYS,ITEST,J,JCOORD,JELTOP,JNF,JRESTR,JSYS,K,NBODY,NA,
     *    NB,NF,NIN,NODEL,NODNUM,NODSLC,NOUT,NSTEPS,RESNOD,
     *    RESTR,RSTRND,STEER,STEP,TIME,TOTBOD,TOTDOF,TOTELS,
     *    TOTNOD,TOTROW,TOTSLC,TOTW

      DOUBLE PRECISION A0,A1,ALPHA,BWARP,BWEFT,BCOORD,BETA,
     *    BODYMP,C1,C2,C3,C4,C5,C6,C7,COORD,D2A0,D2A1,DA0,DA1,
     *    DTIM,ELM,GRAVTY,KE,LOADS,MASS,NEWC,NX,NY,P,PREVL,
     *    RAD,SYSK,SYSM,SYSW,THETA,THICKN,TELKTT,V,WORK,X,Y,
     *    ZSLICE

      LOGICAL COLLID,FIRST

      DIMENSION A0(6700),A1(6700),BCOORD(1200,3),BODYMP(1200,3),
     *    COORD(1300,3),D2A0(6700),D2A1(6700),DA0(6700),DA1(6700),
     *    ELELST(1300),ELM(12,12),ELTOP(2300,10),FREEDM(1300),
     *    KE(2300,6),LOADS(6700),NEWC(1300,3),NF(1300,6),P(6700),
     *    PREVL(2300),RESTR(1300,7),RSTRND(1300),STEER(12),
     *    SYSK(6700,250),SYSM(6700,250),SYSW(6700,250),TELKTT(12,12),
```

```
     *    WORK(6700),ZSLICE(100)

      DATA GRAVTY /9.80665/,ICOORD /1300/,IELTOP /2300/,ISTEER /12/,
     *    ISYS /6700/,ITEST /0/,JCOORD /3/,JELTOP /10/,JNF /6/,JRESTR /7/,
     *    JSYS /250/,NODSLC /30/,V /0.25/

c i/o files:

      DATA NBODY /5/,NIN /2/,NOUT /4/
      OPEN(UNIT=NIN,FILE='data',STATUS='OLD')
      OPEN(UNIT=NOUT,FILE='coord',STATUS='OLD')
      OPEN(UNIT=NBODY,FILE='bodymodel',STATUS='OLD')

c read in skirt mesh nodes and elements:

      READ(NIN,8010) TOTNOD,DIMEN
      DO 1010 I=1,TOTNOD
       READ(NIN,8020) NODNUM,(COORD(NODNUM,J),J=1,DIMEN)
 1010 CONTINUE

      DO 1013 NODNUM=1,TOTNOD
       DO 1013 J=1,JCOORD
        NEWC(NODNUM,J)=COORD(NODNUM,J)
1013  CONTINUE

      READ(NIN,8010) ELTYP,TOTELS,NODEL
      DO 1020 I=1,TOTELS
       READ (NIN,8010) ELNUM,(ELTOP(ELNUM,J+2),J=1,NODEL)
       ELTOP(ELNUM,1)=ELTYP
       ELTOP(ELNUM,2)=NODEL
1020  CONTINUE

c read in chosen fabric properties and establish element variables for stiffness
c matrices:

      READ(NIN,8010) FABRIC
      READ(NIN,8050) BWARP,BWEFT,MASS,THICKN
      CALL CALCKE(BWARP,BWEFT,COORD,ELTOP,KE,THICKN,TOTELS)

c read in time and damping data:

      READ(NIN,8020) NSTEPS,DTIM,THETA
      READ(NIN,8050) ALPHA,BETA

c read in restrained node information:

      READ(NIN,8010) DOFNOD
      READ(NIN,8010) RESNOD
      K=DOFNOD + 1
      DO 1030 I=1,RESNOD
       READ (NIN,8010) (RESTR(I,J),J=1,K)
       DO 1025 J=2,K
        IF (RESTR(I,J).NE.0) FREEDM(RESTR(I,1))=FREEDM(RESTR(I,1))+1
1025  CONTINUE
1030  CONTINUE
```

```
c read in cloth mesh link data for display:

      READ(NIN,8010) TOTROW,TOTW
      DO 1040 I=1,TOTNOD
       READ (NIN,8010) ELELST(I)
1040  CONTINUE

c read in body model node data:

      READ(NBODY,8010) TOTSLC,TOTBOD
      DO 1050 I=1,TOTBOD
       READ(NBODY,8030) (BCOORD(I,J),J=1,3)
1050  CONTINUE

c determine height of each body slice:

      K=0
      DO 1060 I=1,TOTBOD,NODSLC
       K=K+1
       ZSLICE(K)=BCOORD(I,3)
1060  CONTINUE

c develop body model mapping:

      CALL FNDRAD(BCOORD,RAD,TOTBOD)
      DO 1065 NODNUM=1,TOTBOD
       X=BCOORD(NODNUM,1)
       Y=BCOORD(NODNUM,2)
       CALL FNDBMP(NX,NY,RAD,X,Y)
       BODYMP(NODNUM,1)=NX
       BODYMP(NODNUM,2)=NY
       BODYMP(NODNUM,3)=BCOORD(NODNUM,3)
1065  CONTINUE

c set up starting conditions for time stepping:

      DOFEL = NODEL*DOFNOD
      CALL FORMNF(RESTR,ICOORD,JRESTR,RESNOD,TOTNOD,DOFNOD,
     *     NF,ICOORD,JNF,TOTDOF,ITEST)
      CALL VECNUL(A0,ISYS,TOTDOF,ITEST)
      CALL VECNUL(DA0,ISYS,TOTDOF,ITEST)
      CALL VECNUL(D2A0,ISYS,TOTDOF,ITEST)
      FIRST=.TRUE.
      DO 1070 ELNUM=1,TOTELS
       CALL FREDIF(ELNUM,ELTOP,IELTOP,JELTOP,NF,ICOORD,JNF,
     *     DOFNOD,FIRST,DIF,ITEST)
1070  CONTINUE
      HBAND=DIF+1

c assign mass matrix:

      DO 1080 I=1,DOFEL
       ELM(I,I)=MASS
1080  CONTINUE
```

```
c calculated loads:

      CALL CALCP(COORD,ELTOP,GRAVTY,MASS,NF,P,TOTELS,
     *     TOTNOD)

c print out control variables for display and starting position for cloth:

      WRITE(NOUT,8010) TOTNOD,NSTEPS,TOTROW,TOTW,FABRIC
      DO 1085 NODNUM=1,TOTNOD
       WRITE(NOUT,8040) (COORD(ELELST(NODNUM),J),J=1,JCOORD)
1085  CONTINUE

c begin time stepping process:

      DO 2000 STEP=1,NSTEPS
       DO 1180 TIME=1,2

c repeat step with new dtim if collision has been detected:

        IF (TIME.EQ.1.OR.COLLID.EQ..TRUE.) THEN

c calculate Newmark constants:

         C1=ALPHA+1.D0/(THETA*DTIM)
         C2=BETA+THETA*DTIM
         C3=(1.D0-THETA)*DTIM
         C4=BETA-(1.D0-THETA)*DTIM
         C5=1.D0/(THETA*DTIM)
         C6=(1.D0-THETA)/THETA
         C7=THETA*DTIM

c calculate load vector:

         CALL VECNUL(LOADS,ISYS,TOTDOF,ITEST)
         DO 1090 I=1,TOTDOF
          LOADS(I)=C7*P(I)+C3*P(I)
1090     CONTINUE

c set up system matrices:

         CALL MATNUL(SYSK,ISYS,JSYS,TOTDOF,HBAND,ITEST)
         CALL MATNUL(SYSM,ISYS,JSYS,TOTDOF,HBAND,ITEST)

         DO 1100 ELNUM=1,TOTELS
          NA=ELTOP(ELNUM,3)
          NB=ELTOP(ELNUM,4)
          CALL CALCK(COORD,DOFEL,DOFNOD,ELNUM,ITEST,KE,NA,NB,
     *      PREVL,STEP,TELKTT,V)
          CALL DIRECT(ELNUM,ELTOP,IELTOP,JELTOP,NF,ICOORD,JNF,
     *      DOFNOD,STEER,ISTEER,ITEST)
          CALL ASSYM(SYSK,ISYS,JSYS,TELKTT,DOFEL,DOFEL,STEER,
     *      ISTEER,HBAND,DOFEL,ITEST)
          CALL ASSYM(SYSM,ISYS,JSYS,ELM,DOFEL,DOFEL,STEER,
     *      ISTEER,HBAND,DOFEL,ITEST)
1100     CONTINUE
```

```
c Newmark:

      CALL MATNUL(SYSW,ISYS,JSYS,TOTDOF,HBAND,ITEST)
      DO 1120 I=1,TOTDOF
       DO 1120 J=1,HBAND
        SYSW(I,J)=C1*SYSM(I,J)+C4*SYSK(I,J)
        SYSK(I,J)=C1*SYSM(I,J)+C2*SYSK(I,J)
        SYSM(I,J)=SYSM(I,J)/THETA
1120    CONTINUE

      CALL CHORDN(SYSK,ISYS,JSYS,TOTDOF,HBAND,ITEST)
      CALL VECNUL(WORK,ISYS,TOTDOF,ITEST)
      CALL MVSYB(SYSM,ISYS,JSYS,DA0,ISYS,WORK,ISYS,TOTDOF,
     *   HBAND,ITEST)
      CALL VECADD(LOADS,ISYS,WORK,ISYS,TOTDOF,ITEST)
      CALL MVSYB(SYSW,ISYS,JSYS,A0,ISYS,WORK,ISYS,TOTDOF,
     *   HBAND,ITEST)
      CALL VECADD(LOADS,ISYS,WORK,ISYS,TOTDOF,ITEST)
      CALL CHOSUB(SYSK,ISYS,JSYS,LOADS,ISYS,TOTDOF,HBAND,
     *   ITEST)
      CALL VECCOP(LOADS,ISYS,A1,ISYS,TOTDOF,ITEST)
      DO 1140 K=1,TOTDOF
       DA1(K)=C5*(A1(K)-A0(K))-C6*DA0(K)
       D2A1(K)=C5*(DA1(K)-DA0(K))-C6*D2A0(K)
1140    CONTINUE

c calculate new nodal coordinate values:

      CALL UPDXYZ(A1,COORD,NF,TOTNOD)

c determine if collisions have occurred:

      IF (TIME.EQ.1) THEN
       DO 1150 I=1,TOTNOD
        RSTRND(I)=0
1150     CONTINUE
       COLLID=.FALSE.
       CALL COLDET(BCOORD,BODYMP,COLLID,COORD,ELTOP,
     *   FREEDM,ITEST,JCOORD,NEWC,NF,NODSLC,RAD,RSTRND,STEP,
     *   TOTELS,TOTSLC,ZSLICE)

c activate response to collisions if necessary:

      IF (COLLID.EQ..TRUE.) CALL COLRSP(A0,A1,COORD,DA0,D2A0,
     *   DOFNOD,ICOORD,ISYS,ITEST,JCOORD,NEWC,NF,P,RSTRND,
     *   STEP,TOTDOF,TOTNOD)
      END IF
     END IF
1180  CONTINUE

c update displacement, velocity and acceleration vectors for next time step:

     CALL VECCOP(A1,ISYS,A0,ISYS,TOTDOF,ITEST)
     CALL VECCOP(DA1,ISYS,DA0,ISYS,TOTDOF,ITEST)
     CALL VECCOP(D2A1,ISYS,D2A0,ISYS,TOTDOF,ITEST)
```

```
c record current nodal values:

      DO 1190 NODNUM=1,TOTNOD
       DO 1190 I=1,3
        NEWC(NODNUM,I)=COORD(NODNUM,I)
1190  CONTINUE

c print out results:

      DO 1195 NODNUM=1,TOTNOD
       WRITE(NOUT,8040) (COORD(ELELST(NODNUM),I),I=1,JCOORD)
1195  CONTINUE

2000  CONTINUE

      STOP

 8010 FORMAT (16I5)
 8020 FORMAT (I5,6F10.0)
 8030 FORMAT (3F10.0)
 8040 FORMAT (3F10.5)
 8050 FORMAT (6F15.7)
      END


c calculate local stiffness matrix for each element and convert from local to global
c data:

      SUBROUTINE CALCK(COORD,DOFEL,DOFNOD,ELNUM,ITEST,KE,
     *    NA,NB,PREVL,STEP,TELKTT,V)
      INTEGER DOFEL,DOFNOD,ELNUM,ITEST,NA,NB,STEP
      DOUBLE PRECISION A,COORD,E,ELEML,ELK,ELKE,ELKG,IY,IZ,J,
     *    KE,L,LX,LY,LZ,PREVL,T,TELK,TELKTT,TT,U,V
      DIMENSION COORD(1300,3),ELK(12,12),ELKE(12,12),ELKG(12,12),
     *    KE(2300,6),PREVL(2300),T(12,12),TELK(12,12),TELKTT(12,12),
     *    TT(12,12)

c clear matrices from last element:

      CALL MATNUL(T,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATNUL(TT,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATNUL(ELKE,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATNUL(ELKG,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATNUL(ELK,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATNUL(TELK,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATNUL(TELKTT,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)

c assign constants for ELKE and calculate variables for ELKG:

      A=KE(ELNUM,1)
      E=KE(ELNUM,2)
      J=KE(ELNUM,3)
      IY=KE(ELNUM,4)
      IZ=KE(ELNUM,5)
      L=KE(ELNUM,6)
```

```
      CALL CALCL(COORD,ELEML,LX,LY,LZ,NA,NB)
      IF (STEP.EQ.1) U=0.D0
      IF (STEP.NE.1) U=ELEML-PREVL(ELNUM)

c create T and T transposed:

      CALL ASST(ELEML,LX,LY,LZ,T)
      CALL MATRAN(T,DOFEL,DOFEL,TT,DOFEL,DOFEL,DOFEL,DOFEL,
     *     ITEST)

c assign ELKE and ELKG matrices:

      CALL ASSEKE(A,DOFEL,DOFNOD,E,ELKE,IY,IZ,J,L,V)
      CALL ASSEKG(A,DOFEL,DOFNOD,E,ELEML,ELKG,IY,IZ,U)
      CALL MATCOP(ELKE,DOFEL,DOFEL,ELK,DOFEL,DOFEL,DOFEL,
     *     DOFEL,ITEST)

c add together and transform from local to global values:

      CALL MATADD(ELK,DOFEL,DOFEL,ELKG,DOFEL,DOFEL,DOFEL,
     *     DOFEL,ITEST)
      CALL MATMUL(T,DOFEL,DOFEL,ELK,DOFEL,DOFEL,TELK,DOFEL,
     *     DOFEL,DOFEL,DOFEL,DOFEL,ITEST)
      CALL MATMUL(TELK,DOFEL,DOFEL,TT,DOFEL,DOFEL,TELKTT,
     *     DOFEL,DOFEL,DOFEL,DOFEL,DOFEL,ITEST)

c update element length for next time step:

      PREVL(ELNUM)=ELEML
      END


c calculate length of element:

      SUBROUTINE CALCL(COORD,ELEML,LX,LY,LZ,NA,NB)
      INTEGER NA,NB
      DOUBLE PRECISION COORD,ELEML,LX,LY,LZ
      DIMENSION COORD(1300,3)
      LX=COORD(NB,1)-COORD(NA,1)
      LY=COORD(NB,2)-COORD(NA,2)
      LZ=COORD(NB,3)-COORD(NA,3)
      ELEML=DSQRT(LX**2+LY**2+LZ**2)
      END


c establish element values necessary to calculate each ELKE matrix:

      SUBROUTINE CALCKE(BWARP,BWEFT,COORD,ELTOP,KE,THICKN,
     *     TOTELS)
      INTEGER ELNUM,ELTOP,NA,NB,NC,ND,TOTELS
      DOUBLE PRECISION A,BWARP,BWEFT,BASE,COORD,IY,IZ,J,KE,L,
     *     ,LX,LY,LZ,THICKN
      DIMENSION COORD(1300,3),ELTOP(2300,10),KE(2300,6)
      DO 20 ELNUM=1,TOTELS
       NA=ELTOP(ELNUM,3)
```

```
      NB=ELTOP(ELNUM,4)
      BASE=0.D0
      DO 10 I=1,TOTELS
       IF (I.NE.ELNUM) THEN
        NC=ELTOP(I,3)
        ND=ELTOP(I,4)
        IF (NC.EQ.NA.OR.ND.EQ.NA.OR.NC.EQ.NB.OR.ND.EQ.NB) THEN
         IF ((NB.EQ.NA+1.AND.ND.NE.NC+1).OR.
     *      (NB.NE.NA+1.AND. ND.EQ.NC+1)) THEN
          CALL CALCL(COORD,L,LX,LY,LZ,NC,ND)
          BASE=BASE+(L/2)
         END IF
        END IF
       END IF
10    CONTINUE
      BASE=BASE/2
      A=BASE*THICKN
      IY=(BASE*(THICKN**3))/12
      IZ=(THICKN*(BASE**3))/12
      J=IY+IZ
      IF (NB.EQ.NA+1) E=BWARP/J
      IF (NB.NE.NA+1) E=BWEFT/J
      CALL CALCL(COORD,L,LX,LY,LZ,NA,NB)
      KE(ELNUM,1)=A
      KE(ELNUM,2)=E
      KE(ELNUM,3)=J
      KE(ELNUM,4)=IY
      KE(ELNUM,5)=IZ
      KE(ELNUM,6)=L
20   CONTINUE
     END


c calculate load value for each node:

     SUBROUTINE CALCP(COORD,ELTOP,GRAVTY,MASS,NF,P,TOTELS,
    *    TOTNOD)
     INTEGER COUNT,ELNUM,ELTOP,NF,NA,NB,NODNUM,TOTELS,
    *    TOTNOD
     DOUBLE PRECISION BASE,COORD,GRAVTY,HEIGHT,L,LX,LY,LZ,
    *    MASS,P
     DIMENSION COORD(1300,3),ELTOP(2300,10),NF(1300,6),P(6700)
     DO 20 NODNUM=1,TOTNOD
      IF (NF(NODNUM,3).NE.0) THEN
       BASE=0.D0
       HEIGHT=0.D0
       COUNT=0
       DO 10 ELNUM=1,TOTELS
        NA=ELTOP(ELNUM,3)
        NB=ELTOP(ELNUM,4)
        IF (NA.EQ.NODNUM.OR.NB.EQ.NODNUM) THEN
         CALL CALCL(COORD,L,LX,LY,LZ,NA,NB)
         IF (NB.EQ.NA+1) THEN
          HEIGHT=HEIGHT+(L/2)
          COUNT=COUNT+1
```

```
        ELSE
         BASE=BASE+(L/2)
         COUNT=COUNT+1
        END IF
       END IF
10      CONTINUE
      IF (COUNT.EQ.3) HEIGHT=HEIGHT*2
      P(NF(NODNUM,3))=-BASE*HEIGHT*MASS*GRAVTY
     END IF
20    CONTINUE
     END
```

c assign transformation matrix T:

```
     SUBROUTINE ASST(ELEML,LX,LY,LZ,T)
     INTEGER I,II,JJ
     DOUBLE PRECISION D,ELEML,L,LX,LY,LZ,M,N,T
     DIMENSION T(12,12)
     L=LX/ELEML
     M=LY/ELEML
     N=LZ/ELEML
     D=DSQRT(1-(N**2))
     II=-2
     JJ=-2
     DO 10 I=1,4
      II=II+3
      JJ=JJ+3
      T(II,JJ)=L
      T(II,JJ+1)=-M/D
      T(II,JJ+2)=-(L*N)/D
      T(II+1,JJ)=M
      T(II+1,JJ+1)=L/D
      T(II+1,JJ+2)=-(M*N)/D
      T(II+2,JJ)=N
      T(II+2,JJ+1)=0
      T(II+2,JJ+2)=D
10    CONTINUE
     END
```

c assign element elasticity matrix:

```
     SUBROUTINE ASSEKE(A,DOFEL,DOFNOD,E,ELKE,IY,IZ,J,L,V)
     INTEGER DOFEL,DOFNOD,I,II
     DOUBLE PRECISION A,E,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,ELKE,
    *    IY,IZ,J,L,V
     DIMENSION ELKE(12,12)
     E1=E*A/L
     E2=12*E*IZ/(L**3)
     E3=12*E*IY/(L**3)
     E4=J*E/(2*L*(1+V))
     E5=4*E*IY/L
     E6=4*E*IZ/L
     E7=6*E*IZ/(L**2)
```

```
      E8=6*E*IY/(L**2)
      E9=2*E*IY/L
      E10=2*E*IZ/L
      ELKE(1,1)=E1
      ELKE(2,2)=E2
      ELKE(3,3)=E3
      ELKE(4,4)=E4
      ELKE(5,5)=E5
      ELKE(6,6)=E6
      DO 10 I=1,DOFNOD
       ELKE(I+DOFNOD,I+DOFNOD)=ELKE(I,I)
10     CONTINUE
      DO 20 I=1,4
       ELKE(I+DOFNOD,I)=-ELKE(I,I)
20     CONTINUE
      ELKE(11,5)=E9
      ELKE(12,6)=E10
      ELKE(5,3)=-E8
      ELKE(6,2)=E7
      ELKE(8,6)=-E7
      ELKE(9,5)=E8
      ELKE(11,3)=-E8
      ELKE(12,2)=E7
      ELKE(11,9)=E8
      ELKE(12,8)=-E7
      DO 30 I=1,DOFEL-1
       DO 30 II=I+1,DOFEL
        ELKE(I,II)=ELKE(II,I)
30     CONTINUE
      END
```

c assign element geomentric elasticity matrix:

```
      SUBROUTINE ASSEKG(A,DOFEL,DOFNOD,E,ELEML,ELKG, IY,IZ,U)
      INTEGER DOFEL,DOFNOD
      DOUBLE PRECISION A,E,ELEML,ELKG,G1,G2,G3,G4,IY,IZ,PIN,U
      DIMENSION ELKG(12,12)
      PIN=(E*A*U)/ELEML
      G1=(6*PIN)/(5*ELEML)
      G2=PIN/10
      G3=(2*PIN*ELEML)/15
      G4=(PIN*ELEML)/30
      DO 10 I=2,3
       ELKG(I,I)=G1
       ELKG(I+DOFNOD,I)=-G1
10     CONTINUE
      ELKG(4,4)=G3
      ELKG(10,4)=-G4
      DO 20 I=5,6
       ELKG(I,I)=G3
       ELKG(I+DOFNOD,I)=-G4
20     CONTINUE
      DO 30 I=1,DOFNOD
       ELKG(I+DOFNOD,I+DOFNOD)=ELKG(I,I)
```

```
30  CONTINUE
    ELKG(5,3)=-G2
    ELKG(6,2)=G2
    ELKG(8,6)=-G2
    ELKG(9,5)=G2
    ELKG(11,3)=-G2
    ELKG(12,2)=G2
    ELKG(11,9)=G2
    ELKG(12,8)=-G2
    DO 40 I=1,DOFEL-1
     DO 40 J=I+1,DOFEL
      ELKG(I,J)=ELKG(J,I)
40  CONTINUE
    END


c update x, y and z values of each node:

    SUBROUTINE UPDXYZ(A1,COORD,NF,TOTNOD)
    INTEGER I,NF,NODNUM,TOTNOD
    DOUBLE PRECISION A1,COORD
    DIMENSION A1(6700),COORD(1300,3),NF(1300,6)
    DO 10 NODNUM=1,TOTNOD
     DO 10 I=1,3
      IF (NF(NODNUM,I).NE.0) COORD(NODNUM,I)=
   *     COORD(NODNUM,I)+A1(NF(NODNUM,I))
10  CONTINUE
    END


c find location of cloth node with respect to
c body cylinder:

    SUBROUTINE FNDCMP(BCOORD,BODYMP,NODSLC,RAD,RADC,
   *    SLICEA,SLICEB,XYZ,ZSLICE)
    INTEGER NA1,NA2,NB1,NB2,NODSLC,SLICEA,SLICEB
    DOUBLE PRECISION AA,AB,AC,AD,BCOORD,BODYMP,DAX,DAY,
   *    DBX,DBY,DE,DF,DX,DY,LX,LY,RAD,RADC,UX,UY,XYZ,
   *    ZSLICE
    DIMENSION BCOORD(1200,3),BODYMP(1200,3),XYZ(3),
   *    ZSLICE(100)

    IF (SLICEA.NE.0) THEN

c find relvant body node on slice above node:

    CALL FNDNOD(BODYMP,NA1,NA2,NODSLC,SLICEA,XYZ)

c if cloth node coincides with body node, find distance using this body node only:

    IF (NA2.EQ.0) THEN
     CALL FNDDST(BCOORD,BODYMP,UX,UY,NA1)

c otherwise determine distance using both nodes:
```

```
      ELSE
       CALL FNDPRP(AA,BODYMP(NA1,1),BODYMP(NA1,2),XYZ)
       CALL FNDPRP(AB,BODYMP(NA2,1),BODYMP(NA2,2),XYZ)
       CALL FNDDST(BCOORD,BODYMP,DAX,DAY,NA1)
       CALL FNDDST(BCOORD,BODYMP,DBX,DBY,NA2)
       CALL FNDNXY(AA,AB,DAX,DAY,DBX,DBY,UX,UY)
      END IF
```

c if cloth node does not coincide exactly with a body slice, need to determine
c remaining distance values and the process commented above is repeated for the
c slice below the cloth node:

```
      IF (SLICEB.NE.0) THEN
       CALL FNDNOD(BODYMP,NB1,NB2,NODSLC,SLICEB,XYZ)
       IF (NB2.EQ.0) THEN
        CALL FNDDST(BCOORD,BODYMP,LX,LY,NB1)
       ELSE
        CALL FNDPRP(AC,BODYMP(NB1,1),BODYMP(NB1,2),XYZ)
        CALL FNDPRP(AD,BODYMP(NB2,1),BODYMP(NB2,2),XYZ)
        CALL FNDDST(BCOORD,BODYMP,DAX,DAY,NB1)
        CALL FNDDST(BCOORD,BODYMP,DBX,DBY,NB2)
        CALL FNDNXY(AC,AD,DAX,DAY,DBX,DBY,LX,LY)
       END IF
```

c distance based on location of upper and lower data:

```
       DE=ABS(XYZ(3)-ZSLICE(SLICEB))
       DF=ABS(ZSLICE(SLICEA)-XYZ(3))
       DX=(DE/(DE+DF)*UX)+(DF/(DE+DF)*LX)
       DY=(DE/(DE+DF)*UY)+(DF/(DE+DF)*LY)
```

c otherwise the distance is based on the coinciding slice only:

```
      ELSE
       DX=UX
       DY=UY
      END IF
      RADC=DSQRT((XYZ(1)+DX)**2+(XYZ(2)+DY)**2)
     END IF
     END
```

c check if node and/or element has collided:

```
     SUBROUTINE NODCOL(BCOORD,BODYMP,COLLEL,FREDMA,
    *    JCOORD,NODSLC,RAD,TOTSLC,XYZE,XYZF,ZSLICE)
     INTEGER FINISH,FREDMA,I,JCOORD,NODSLC,SLICEA,SLICEB,
    *    START,TOTSLC
     DOUBLE PRECISION BCOORD,BODYMP,RAD,RADC,T,XYZE,XYZF,
    *    XYZG,ZSLICE
     DIMENSION BCOORD(1200,3),BODYMP(1200,3),XYZE(3),XYZF(3),
    *    XYZG(3),ZSLICE(100)
     LOGICAL COLLEL
     COLLEL=.FALSE.
     FINISH=10
```

```
      IF (FREDMA.LT.3) THEN
       START=1
       T=0.D0
      ELSE
       START=2
       T=DBLE(1.D0/(FINISH-1))
      END IF
      DO 10 I=START,FINISH
       IF (COLLEL.EQ..FALSE.) THEN
        IF (I.NE.1) T=T+DBLE(1.D0/(FINISH-1))
        CALL ASVAL2(JCOORD,T,XYZE,XYZF,XYZG)
        CALL FNDSLC(SLICEA,SLICEB,TOTSLC,XYZG(3),ZSLICE)
        CALL FNDCMP(BCOORD,BODYMP,NODSLC,RAD,RADC,SLICEA,
     *     SLICEB,XYZG,ZSLICE)
        IF (RADC.LE.RAD+0.005) COLLEL=.TRUE.
       END IF
10    CONTINUE
      END


c assign coordinate variables - from cloth node data:

      SUBROUTINE ASVAL1(COORD,JCOORD,NODE,XYZ)
      INTEGER I,JCOORD,NODE
      DOUBLE PRECISION COORD,XYZ
      DIMENSION COORD(1300,3),XYZ(3)
      DO 10 I=1,JCOORD
       XYZ(I)=COORD(NODE,I)
10    CONTINUE
      END


c assign coordinate variables - stepping along element:

      SUBROUTINE ASVAL2(JCOORD,T,XYZ1,XYZ2,XYZ3)
      INTEGER I,JCOORD
      DOUBLE PRECISION T,XYZ1,XYZ2,XYZ3
      DIMENSION XYZ1(3),XYZ2(3),XYZ3(3)
      DO 10 I=1,JCOORD
       XYZ3(I)=((1-T)*XYZ1(I))+(T*XYZ2(I))
10    CONTINUE
      END


c see if collisions have occurred:

      SUBROUTINE COLDET (BCOORD,BODYMP,COLLID,COORD,ELTOP,
     *     FREEDM,ITEST,JCOORD,NEWC,NF,NODSLC,RAD,RSTRND,STEP,
     *     TOTELS,TOTSLC,ZSLICE)
      INTEGER ELNUM,ELTOP,FINISH,FREEDM,ITEST,JCOORD,NF,NA,
     *     NB,NODSLC,RSTRND,STEP,TOTELS,TOTSLC
      DOUBLE PRECISION BCOORD,BODYMP,COORD,NEWC,RAD,XYZA,
     *     XYZB,ZSLICE
      DIMENSION BCOORD(1200,3),BODYMP(1200,3),COORD(1300,3),
     *     ELTOP(2300,10),FREEDM(1300),NEWC(1300,3),NF(1300,6),
```

```
     *    RSTRND(1300),XYZA(3),XYZB(3),ZSLICE(100)
      LOGICAL COLLID,COLLEL
      FINISH=10
      DO 110 ELNUM=1,TOTELS
       NA=ELTOP(ELNUM,3)
       NB=ELTOP(ELNUM,4)

c if at least one node is not restrained in the element pair then run the checks:

       IF (FREEDM(NA).LT.3.OR.FREEDM(NB).LT.3) THEN
        CALL ASVAL1(COORD,JCOORD,NA,XYZA)
        CALL ASVAL1(COORD,JCOORD,NB,XYZB)

c determine if node/element collision has occurred:

        CALL NODCOL(BCOORD,BODYMP,COLLEL,FREEDM(NA),
     *    JCOORD,NODSLC,RAD,TOTSLC,XYZA,XYZB,ZSLICE)

c if collisions have occurred:

        IF (COLLEL.EQ..TRUE.) THEN
         COLLID=.TRUE.
         IF (FREEDM(NA).LT.3) THEN
          RSTRND(NA)=1
          FREEDM(NA)=6
         END IF
         IF (FREEDM(NB).LT.3) THEN
          RSTRND(NB)=1
          FREEDM(NB)=6
         END IF
        END IF
       END IF
110   CONTINUE
      END


c find closest body slices to cloth node:

      SUBROUTINE FNDSLC(SLICEA,SLICEB,TOTSLC,Z,ZSLICE)
      INTEGER I,SLICEA,SLICEB,TOTSLC
      DOUBLE PRECISION Z,ZSLICE
      DIMENSION ZSLICE(100)
      SLICEA=0
      SLICEB=0
      DO 10 I=1,TOTSLC
       IF (Z.EQ.ZSLICE(I)) THEN
        SLICEA=I
       ELSE IF (I.NE.TOTSLC.AND.Z.GT.ZSLICE(I).AND.
     *    Z.LT.ZSLICE(I+1)) THEN
        SLICEA=I+1
        SLICEB=I
       END IF
10    CONTINUE
      END
```

```
c find body nodes on either side of cloth node:

      SUBROUTINE FNDNOD(BODYMP,N1,N2,NODSLC,SLICE,XYZ)
      INTEGER FINISH,I,N1,N2,NODSLC,QB,QN,SLICE,START
      DOUBLE PRECISION AB,AN,BODYMP,XYZ
      DIMENSION BODYMP(1200,3),XYZ(3)
      LOGICAL XAXIS,YAXIS
      CALL FNDQRT(QN,XYZ(1),XYZ(2))
      CALL FNDANG(AN,XAXIS,XYZ(1),YAXIS,XYZ(2))
      N1=0
      N2=0
      START=(SLICE-1)*NODSLC+1
      FINISH=SLICE*NODSLC
      DO 10 I=START,FINISH
       IF (N1.EQ.0) THEN
        CALL FNDQRT(QB,BODYMP(I,1),BODYMP(I,2))
        CALL FNDANG(AB,XAXIS,BODYMP(I,1),YAXIS,BODYMP(I,2))
        IF (XAXIS.EQ..FALSE..AND.YAXIS.EQ..FALSE.) THEN
         IF (QN.EQ.QB) THEN
          IF (AB.EQ.AN) THEN
           N1=I
          ELSE IF (QN.EQ.4.AND.I.EQ.FINISH.AND.AB.LT.AN) THEN
           N1=START
           N2=FINISH
          ELSE IF (((QN.EQ.1.OR.QN.EQ.3).AND.AB.LT.AN).OR.
     *       ((QN.EQ.2.OR.QN.EQ.4).AND.AB.GT.AN)) THEN
           N1=I-1
           N2=I
          END IF
         ELSE IF (QB.EQ.QN+1) THEN
          N1=I-1
          N2=I
         END IF
        END IF
       END IF
10    CONTINUE
      END


c find which segment cloth node lies:

      SUBROUTINE FNDQRT(QUART,XX,YY)
      INTEGER QUART
      DOUBLE PRECISION XX,YY
      IF (XX.LE.0.AND.YY.LE.0) QUART=1
      IF (XX.LE.0.AND.YY.GT.0) QUART=2
      IF (XX.GT.0.AND.YY.GE.0) QUART=3
      IF (XX.GT.0.AND.YY.LT.0) QUART=4
      END
```

```
c find angle of node to x axis:

      SUBROUTINE FNDANG(ANGLE,XAXIS,XX,YAXIS,YY)
      DOUBLE PRECISION ANGLE,XX,YY
      LOGICAL XAXIS,YAXIS
      IF (XX.EQ.0) THEN
       XAXIS=.TRUE.
      ELSE
       XAXIS=.FALSE.
      END IF
      IF (YY.EQ.0) THEN
       YAXIS=.TRUE.
      ELSE
       YAXIS=.FALSE.
      END IF
      IF (XX.NE.0.AND.YY.NE.0) ANGLE=DATAN(ABS(YY/XX))
      END


c find new values for x and y:

      SUBROUTINE FNDNXY(AA,AB,DAX,DAY,DBX,DBY,XX,YY)
      DOUBLE PRECISION AA,AB,DAX,DAY,DBX,DBY,XX,YY
       XX=(AB/(AA+AB)*DAX)+(AA/(AA+AB)*DBX)
       YY=(AB/(AA+AB)*DAY)+(AA/(AA+AB)*DBY)
      END


c find distance of body node to cylinder:

      SUBROUTINE FNDDST(BCOORD,BODYMP,DX,DY,NODE)
      INTEGER NODE
      DOUBLE PRECISION BCOORD,BODYMP,DX,DY
      DIMENSION BCOORD(1200,3),BODYMP(1200,3)
      DX=BODYMP(NODE,1)-BCOORD(NODE,1)
      DY=BODYMP(NODE,2)-BCOORD(NODE,2)
      END


c find angle formed by cloth and body nodes:

      SUBROUTINE FNDPRP(ANGLE,NX,NY,XYZ)
      DOUBLE PRECISION ANGLE,NX,NY,SA,SB,SC,XYZ
      DIMENSION XYZ(3)
      SA=DSQRT((NX-XYZ(1))**2+(NY-XYZ(2))**2)
      SB=DSQRT(XYZ(1)**2+XYZ(2)**2)
      SC=DSQRT(NX**2+NY**2)
      ANGLE=(SB**2+SC**2-SA**2)/(2*SB*SC)
      END
```

```
c find radius of cylinder:

      SUBROUTINE FNDRAD(BCOORD,RAD,TOTBOD)
      INTEGER I,TOTBOD
      DOUBLE PRECISION BCOORD,BRAD,RAD
      DIMENSION BCOORD(1200,3)
      RAD=0.D0
      DO 10 I=1,TOTBOD
       BRAD=DSQRT(BCOORD(I,1)**2+BCOORD(I,2)**2)
       IF (BRAD.GT.RAD) RAD=BRAD
10    CONTINUE
      RAD=RAD+0.01
      END


c determine cylinder mapping of body model:

      SUBROUTINE FNDBMP(NX,NY,RAD,X,Y)
      DOUBLE PRECISION ANGLE,NX,NY,RAD,X,Y
      IF (X.EQ.0.AND.Y.EQ.0) THEN
       NX=X
       NY=-RAD
      ELSE IF (X.EQ.0) THEN
       NX=X
       NY=Y/ABS(Y)*RAD
      ELSE IF (Y.EQ.0) THEN
       NX=X/ABS(X)*RAD
       NY=Y
      ELSE
       ANGLE=DATAN(ABS(Y/X))
       NX=X/ABS(X)*RAD*DCOS(ANGLE)
       NY=Y/ABS(Y)*RAD*DSIN(ANGLE)
      END IF
      END


c prepare for rerun of step:

      SUBROUTINE COLRSP(A0,A1,COORD,DA0,D2A0,DOFNOD,ICOORD,
     *    ISYS,ITEST,JCOORD,NEWC,NF,P,RSTRND,STEP,TOTDOF,
     *    TOTNOD)
      INTEGER DOFNOD,I,ICOORD,ISYS,J,JCOORD,K,NEWTOT,NF,
     *    RSTRND,STEP,TOTDOF,TOTNOD
      DOUBLE PRECISION A0,A1,COORD,DA0,D2A0,NEWC,P
      DIMENSION A0(6700),A1(6700),COORD(1300,3),DA0(6700),
     *    D2A0(6700), NEWC(1300,3),NF(1300,6),P(6700),RSTRND(1300)

c reset cloth coordinates to start of current time step:

      DO 10 I=1,TOTNOD
       DO 10 J=1,JCOORD
        COORD(I,J)=NEWC(I,J)
10     CONTINUE
```

```
c realign system matrices and vectors:

      NEWTOT=TOTDOF
      DO 50 I=1,TOTNOD
       IF (RSTRND(I).EQ.1) THEN
        DO 30 J=I,TOTNOD
         DO 30 K=1,DOFNOD
          IF (J.EQ.I) NF(J,K)=0
          IF (NF(J,K).NE.0) THEN
           NF(J,K)=NF(J,K)-DOFNOD
           A1(NF(J,K))=A1(NF(J,K)+DOFNOD)
           A0(NF(J,K))=A0(NF(J,K)+DOFNOD)
           DA0(NF(J,K))=DA0(NF(J,K)+DOFNOD)
           D2A0(NF(J,K))=D2A0(NF(J,K)+DOFNOD)
           P(NF(J,K))=P(NF(J,K)+DOFNOD)
          END IF
30      CONTINUE
        NEWTOT=NEWTOT-DOFNOD
       END IF
50    CONTINUE
      TOTDOF=NEWTOT
      END
```

APPENDIX IV - DISPLAY PROGRAM

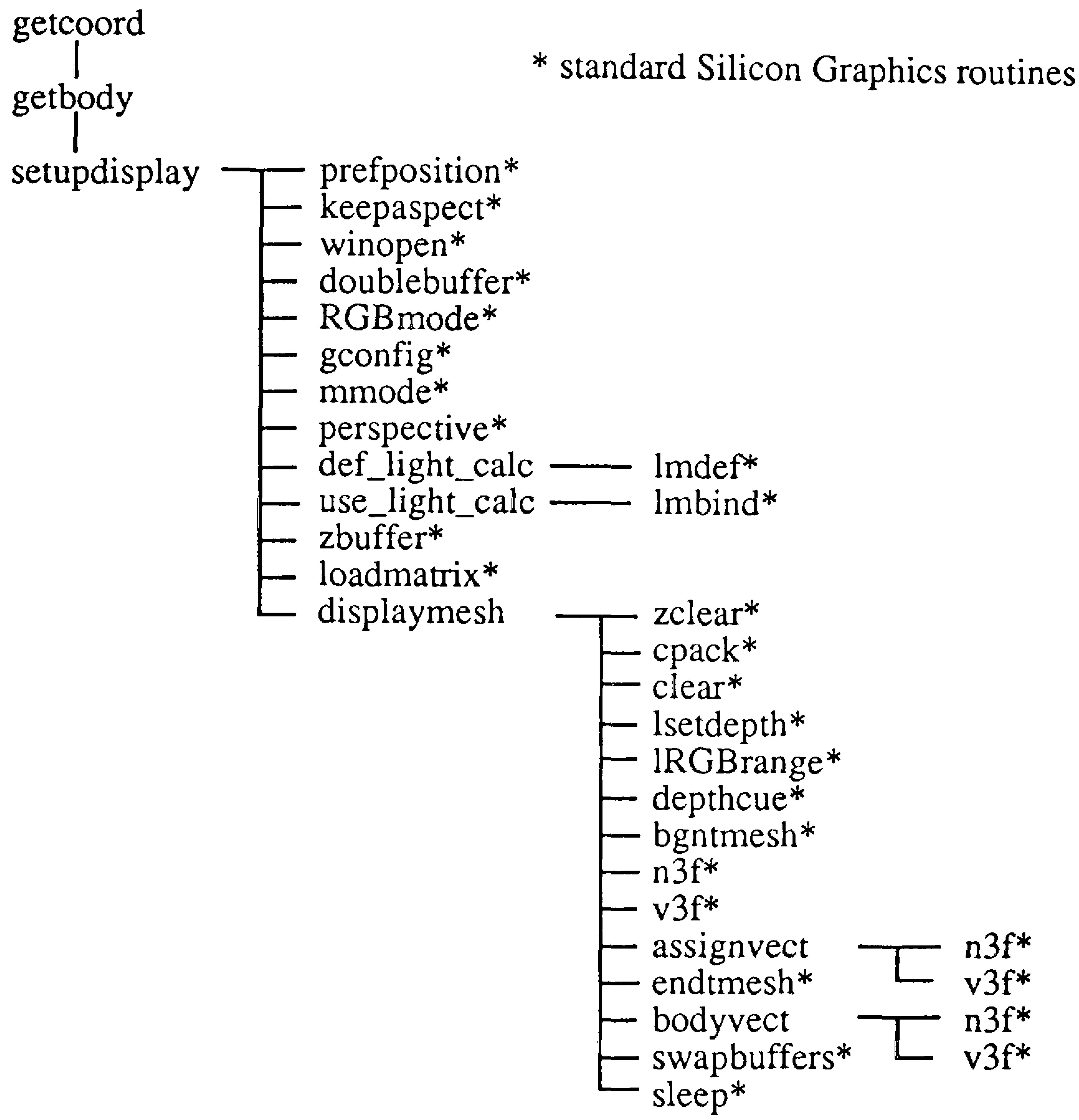


Figure IV.1 - program DISPLAY

IV.1 Subroutines

The DISPLAY program reads in the cloth mesh from file "coord" and body model information from file "bodymodel", sets up the display model parameters and displays the resulting frames. The program has been written in standard C and uses the Silicon Graphics Library.

assignvect	Assigns co-ordinates to a triangle vertex for the cloth mesh.
bgntmesh	Starts the triangle mesh.
bodyvect	Assigns co-ordinates to a triangle vertex for the body model.

clear	Clears the viewport.
cpack	Specifies RGBA colour with a single packed 32-bit integer.
def_light_calc	Defines the lighting calculations.
depthcue	Turns on depth cue mode.
displaymesh	Displays the mesh at each time step.
doublebuffer	Requests double buffer mode for realistic dynamic display.
endtmesh	Ends triangle mesh.
gconfig	Reconfigures the system if the default options have been altered.
getbody	Reads in the x, y and z co-ordinate values for each node in the body model from file "bodymodel".
getcoord	Reads in the x, y and z co-ordinate values for each node at each time step from file "coord".
keepaspect	Specifies the aspect ratio of a graphics window.
lmbind	Binds a lighting model definition to the current material.
lmdef	Defines a lighting model.
loadmatrix	Loads a transformation matrix.
lookat	Defines the viewpoint and a reference point on the line of sight.
lRGBrange	Sets the range of RGB colours for depth cueing.
lsetdepth	Sets the depth range.
mmode	Sets matrix mode.
n3f	Specifies a normal for lighting calculations.
perspective	Specifies a viewing pyramid into the display.
prefposition	Declares location of display window.
RGBmode	Makes the system interpret the contents of the pixel bitplanes as RGB values.
setupdisplay	Sets up the display parameters.
sleep	Pauses the frame for a specified period of time.
swapbuffers	Exchanges the front and back buffers in double buffer mode.
use_light_calc	Uses the lighting calculations.
v3f	Transfers 3D vertex to graphics pipe.
winopen	Opens a window in the position stated in prefposition and gives this window a title.
zbuffer	Starts zbuffer mode for faster image drawing so only the surfaces closest to the viewer are displayed.
zclear	Initialises the zbuffer.

IV.2 Variable Listing

actsteps	overall number of frames (int)
base	control value for node in mesh display (int)
BODYW	number of nodes in each body slice (int)
bcoord	array of x, y and z coordinates for body model (float)
bx, by, bz	x, y and z co-ordinates of body model nodes (float)
coord	array of x, y and z coordinates for cloth mesh (float)
i, j, k	count (int)
level	control value for node in mesh display (int)
material	material selection (int)
node	node number (int)
nodes	number of cloth nodes (int)
slice	number of body slices (int)
steps	number of steps (int)
STEPS	maximum number of time steps (int)
textile	materal selection (int)
tempvect	temporary array storing x, y and z co-ordinates of node (float)
totbod	total number of nodes in body model (int)
totnod	total number of nodes in cloth mesh (int)
totrow	total number of rows in cloth mesh (int)
totslc	total number of body data slices (int)
totw	number of nodes in each row for the cloth mesh (int)
warp	number of cloth rows (int)
wbvect	waistband vector (float)
weft	number of nodes per row (int)

x	x coordinate value (float)
xcoord	x coordinate array (float)
y	y coordinate value (float)
ycoord	y coordinate array (float)
z	z coordinate value (float)
zcoord	z coordinate array (float)

IV.3 Program Listing

```
/* program DISPLAY - rendered cloth and rendered body*/

#include <stdio.h>
#include "../include/gl/gl.h"

#define BODYW 30        /* number of nodes in each body slice */
#define STEPS 2000      /* maximum number of time steps */

struct mlist {float xcoord[STEPS],zcoord[STEPS],ycoord[STEPS];} coord[2000];
struct blist {float bx,bz,by;} bcoord[1500];
int actsteps,textile,totbod,totnod,totrow,totslc,totw;
Matrix idmat={1.0,0.0,0.0,0.0,
        0.0,1.0,0.0,0.0,
        0.0,0.0,1.0,0.0,
        0.0,0.0,0.0,1.0};

void getcoord();
void getbody();
void setupdisplay();
void def_light_calc();
void use_light_calc();
void displaymesh();
void assignvect(int node,int step);
void bodyvect(int node);


main()
{
 getcoord();
 getbody();
 setupdisplay();
}
```

```
void getcoord()
{
 FILE *fpcoord;
 float x,z,y;
 int i,j,material,nodes,steps,warp,weft;

 fpcoord=fopen("scoord","r");
 fscanf(fpcoord,"%d %d %d %d %d",&nodes,&steps,&warp,&weft,&material);
 totnod=nodes;
 actsteps=steps+1;
 totrow=warp;
 totw=weft;
 textile=material;
 for (j=0;j<actsteps;j++)
 {
  for (i=0;i<totnod;i++)
  {
   fscanf(fpcoord,"%f %f %f",&x,&z,&y);
   coord[i].xcoord[j]=x;
   coord[i].zcoord[j]=z;
   coord[i].ycoord[j]=y;
  }
 }
 fclose(fpcoord);
 return;
}


void getbody()
{
 FILE *fpbody;
 float x,y,z;
 int i,node,slice;

 fpbody=fopen("bodymodel","r");
 fscanf(fpbody,"%d %d",&slice,&node);
 totslc=slice;
 totbod=node;
 for (i=0;i<totbod;i++)
 {
  fscanf(fpbody,"%f %f %f",&x,&y,&z);
  bcoord[i].bx=x;
  bcoord[i].by=z;
  bcoord[i].bz=y;
 }
 fclose(fpbody);
 return;
}
```

```
void setupdisplay()
{
 int i,j;

 prefposition(50,1000,50,1000);
 keepaspect(1,1);
 winopen("3d mesh");
 doublebuffer();
 RGBmode();
 gconfig();
 mmode(MVIEWING);
 perspective(400,1.0,1.0,30.0);
 def_light_calc();
 use_light_calc();
 zbuffer(TRUE);
 linesmooth(TRUE);
 for (j=0;j<10;j++)
 {
  for (i=0;i<8;i=i+2)
  {
   loadmatrix(idmat);
   if (i==7) lookat( 2.0, 1.2,  0.0, 0.0, 1.0, 0.0, 0);
   if (i==6) lookat( 1.4, 1.2,  1.4, 0.0, 1.0, 0.0, 0);
   if (i==5) lookat( 0.0, 1.2,  2.0, 0.0, 1.0, 0.0, 0);
   if (i==4) lookat(-1.4, 1.2,  1.4, 0.0, 1.0, 0.0, 0);
   if (i==3) lookat(-2.0, 1.2,  0.0, 0.0, 1.0, 0.0, 0);
   if (i==2) lookat(-1.4, 1.2, -1.4, 0.0, 1.0, 0.0, 0);
   if (i==1) lookat( 0.0, 1.2, -2.0, 0.0, 1.0, 0.0, 0);
   if (i==0) lookat( 1.4, 1.2, -1.4, 0.0, 1.0, 0.0, 0);
   displaymesh();
  }
 }
 return;
}


void def_light_calc()
{
 lmdef(DEFLMODEL,1,0,NULL);
 lmdef(DEFMATERIAL,1,0,NULL);
 lmdef(DEFLIGHT,1,0,NULL);
}


void use_light_calc()
{
 lmbind(LMODEL,1);
 lmbind(LIGHT0,1);
 lmbind(MATERIAL,1);
}
```

```
void displaymesh()
{
 int base,i,j,k,level;
 float wbvect[3];

 for (i=0;i<actsteps;i=i+10)
 {
  zclear();
  cpack(0xFF707070);
  clear();
  lsetdepth(0,0x7f0000);
  lRGBrange(0,0,0,50,50,100,0,0x7f0000);
  depthcue(TRUE);

  bgntmesh();                    /* draw in waistband */
  for (j=0;j<totw;j++)
  {
   wbvect[0]=coord[j].xcoord[0];
   wbvect[1]=coord[j].ycoord[0]+0.03;
   wbvect[2]=coord[j].zcoord[0];
   n3f(wbvect);
   v3f(wbvect);
   assignvect(j,i);
  }
  wbvect[0]=coord[0].xcoord[0];
  wbvect[1]=coord[0].ycoord[0]+0.03;
  wbvect[2]=coord[0].zcoord[0];
  n3f(wbvect);
  v3f(wbvect);
  assignvect(0,i);
  endtmesh();

  for (j=0;j<totrow;j++)         /* draw skirt */
  {
   base=j*totw;
   level=(j+1)*totw;
   bgntmesh();
   for (k=0;k<totw;k++)
   {
    assignvect(k+base,i);
    assignvect(k+level,i);
   }
   assignvect(base,i);
   assignvect(level,i);
   endtmesh();
  }

  lsetdepth(0,0x7f0000);
  lRGBrange(0,0,0,250,150,50,0,0x7f0000);
  depthcue(TRUE);
  for (j=0;j<totslc-1;j++)                        /* draw body model */
  {
   base=j*BODYW;
   level=(j+1)*BODYW;
   {
```

```
      bgntmesh();
      for (k=0;k<BODYW;k++)
      {
       bodyvect(k+level);
       bodyvect(k+base);
      }
      bodyvect(level);
      bodyvect(base);
      endtmesh();
     }
   }
   swapbuffers();
  }
  sleep(10);
  return;
}


void assignvect(int node,int step)
{
  float tempvect[3];

  tempvect[0]=coord[node].xcoord[step];
  tempvect[1]=coord[node].ycoord[step];
  tempvect[2]=coord[node].zcoord[step];
  n3f(tempvect);
  v3f(tempvect);
  return;
}


void bodyvect(int node)
{
  float tempvect[3];

  tempvect[0]=bcoord[node].bx;
  tempvect[1]=bcoord[node].by;
  tempvect[2]=bcoord[node].bz;
  n3f(tempvect);
  v3f(tempvect);
  return;
}
```

APPENDIX V - MEETINGS AND CONFERENCES

The following is a list of meetings, conferences and exhibitions. All were extremely useful to the project and I would like to thank everybody for their valuable time spent with me.

24.10.91	Kevin Miller, Computer Design Inc. Tad Paluchowski, Coats Viyella
21.11.91	Roe van Fossen (Chairman) and Kevin Miller, Computer Design Inc
26.11.91	Sally Smith and Debbie Lundberg, Coats Viyella
14.1.92	Tad Paluchowski and Debbie Lundbertg, Coats Viyella. Tom Schlagater, AVA
23.1.92	CIMTEX, Leicester Polytechnic. Brian Hinds and John McCartney, Belfast University
30.1.92	Debbie Lundberg and Barbara Tomzcak, Coats Viyella
6.2.92	Eleanor Curtis, Royal College of Art, London
18.2.92	Linda Grady (Head of Fashion and Textiles) and Gary Powles (lecturer), Loughborough College of Art and Design
21.2.92	Linda Grady, Maria Parkes (designer and lecturer) and Andrew Hitchkiss (designer and lecturer), Loughborough College of Art and Design
28.2.92	Maria Parkes, Loughborough College of Art and Design
27.3.92	Clotech Exhibition, GMex Manchester
6.4.92	Eleanor Curtis, Royal College of Art, London
5.5.92	Althea McNish, International Textile Designer
6.5.92	Tom Schlagater, AVA
7.7.92	Bob Moulsdale, Coats Viyella
5.11.92	Designers Mean Business, Textile Institute Seminar, London College of Fashion
2.12.92	Investing in Design by Computer, Textile Institute Seminar, Royal College of Art, London
14.1.93	G Fozzard and A Rawling, CIMTEX, Leicester De Montfort University
9.3.93	CADCAM'93, NEC Birmingham
22.3.93	Mechatronics in Textile Industries Conference, Loughborough University of Technology

8.7.93	Smitex, Leicester
25.11.93	Bob Moulsdale, Coats Viyella
5-7.1.94	SERC Prime Skills Course, University of Surrey, Guildford
3.2.94	Bob Moulsdale, Coats Viyella
15.2.94	Prof Peter Jones and Peng Li, Loughborough Anthropometric Shadow Scanner (LASS), Dept of Human Sciences, Loughborough University
30.3.94	Bob Moulsdale, Coats Viyella
23.6.94	Bob Moulsdale, Coats Viyella
13.7.94	Stephen Gray, Anthony Rosella, Richard Aitken and Neil, Fashion Information Systems, Nottingham Trent University
9.11.94	Computers in Clothing Exhibition, Harrogate Exhibition Centre, 7-9 November 1994
8.12.94	Bob Moulsdale and Debbie Lundberg, Coats Viyella