FORMEX
CONFIGURATION PROCESSING FOR
SPACE STRUCTURES

MISS DEEPA LI HADKER

THEESIS SUBMITTED FOR THE
DOCTOR OF PHILOSOPHY

SPACE STRUCTURES RESEARCH CENTRE
DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF SURREY

JUNE 1993
ABSTRACT

The present research deals with extending the repertoire of shapes and forms for space structures using computer aided techniques. A major part of the research concerns the introduction of certain composite transformations. These are termed "paragenic transformations" which combine the effects of cylindrical and spherical transformations to create families of new shapes. It is shown that surfaces obtained from paragenic transformations may be used for a variety of structural forms such as grids, vaults, domes, cable nets, membranes and shell surfaces.

Another important area covered by the present research is concerned with pattern generation. For this purpose, the concept of a "protomorph" is introduced. A protomorph acts as an underlying pattern which can be used as a starting point to create a continuum of patterns. The patterns studied represent cable, bar or beam elements or finite elements for modelling of plate, shell or membrane structures. The research aims at developing a methodology for generating and manipulating space structure forms.

The material in the Thesis is presented as follows: Chapter One contains a brief examination of some notable space structures built world-wide. Chapter Two describes the basic concepts of "formex algebra", a mathematical tool which is ideally suited for the purpose of representing and manipulating forms. Formex algebra is used in conjunction with the programming language Formian which is described in the second part of Chapter Two. A strategy for pattern generation is presented in Chapter Three. Examples in the study include patterns for single layer, double layer and multilayer space structures. Paragenic transformations are introduced in Chapter Four with the help of a number of examples. This part of the study is a major contribution towards expanding the repertoire of available shapes and forms for different classes of space structures. Chapter Five presents the conclusions of the work together with some ideas for future research.
to my parents for all their support and encouragement ...
ACKNOWLEDGEMENTS

My sincere gratitude is due to my supervisor Dr H Nooshin for his dedicated guidance and fatherly affection throughout the development of my research work. What I admire is his clarity of thought, creative thinking and the ability to stimulate positive actions ...

My special thanks to my parents for their patience and understanding. Their contribution cannot be expressed in words. Thanks also to the assistance rendered by my family and friends whose contribution to this thesis was not only essential but much appreciated.

Guildford, June 1993

Deepali Hadker
## CONTENTS

**ABSTRACT**  
2

**ACKNOWLEDGEMENTS**  
4

**CHAPTER ONE**  
**A BRIEF EXAMINATION OF DIFFERENT TYPES OF SPACE STRUCTURES**

1.1 INTRODUCTION  
8

1.2 SPACEFRAMES  
10

1.3 TENSION STRUCTURES  
21

1.4 HYBRID SPACE STRUCTURES  
29

**CHAPTER TWO**  
**CONCEPTS OF FORMEX ALGEBRA AND FORMIAN**

2.1 INTRODUCTION  
32

2.2 FORMEX CONFIGURATION PROCESSING  
33

2.3 FORMEX CONSTITUENTS  
35

2.4 EQUALITY OF FORMICES  
38

2.5 VARIANTS OF A FORMEX  
40

2.6 SEQUATIONS OF A FORMEX  
40

2.7 COMPOSITION OF FORMICES  
41

2.8 FORMEX GRAPHICS  

2.8.1 INTRODUCTION  
43

2.8.2 FORMEX PLOTS  
43

2.8.3 STANDARD RETRONORMS  
52

2.8.4 PLOTTING STYLES  
57

2.8.5 RETROBASES AND PROBASES  
61

2.9 FORMEX FUNCTIONS  

2.9.1 INTRODUCTION  
64
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.9.2</td>
<td>CARDINAL FUNCTIONS</td>
<td>65</td>
</tr>
<tr>
<td>2.9.3</td>
<td>TENDIAL FUNCTIONS</td>
<td>68</td>
</tr>
<tr>
<td>2.9.4</td>
<td>PROVIAL FUNCTIONS</td>
<td>73</td>
</tr>
<tr>
<td>2.9.5</td>
<td>INTRFLECTION FUNCTIONS</td>
<td>78</td>
</tr>
<tr>
<td>2.10</td>
<td>LIBRA COMPOSITION</td>
<td>89</td>
</tr>
<tr>
<td>2.11</td>
<td>FORMIAN</td>
<td>94</td>
</tr>
<tr>
<td>2.11.1</td>
<td>PRIMITIVE CONSTITUENTS</td>
<td>96</td>
</tr>
<tr>
<td>2.11.2</td>
<td>CONSTANTS</td>
<td>97</td>
</tr>
<tr>
<td>2.11.3</td>
<td>ASSIGNMENT STATEMENTS AND VARIABLES</td>
<td>99</td>
</tr>
<tr>
<td>2.11.4</td>
<td>INFORMATION TRANSFER STATEMENTS</td>
<td>100</td>
</tr>
<tr>
<td>2.11.5</td>
<td>ORGANISATIONAL STATEMENTS</td>
<td>105</td>
</tr>
<tr>
<td>2.12</td>
<td>FORMIAN GRAPHICS</td>
<td>108</td>
</tr>
<tr>
<td>2.12.1</td>
<td>FORMIAN RETROCORDS</td>
<td>129</td>
</tr>
<tr>
<td>2.13</td>
<td>SCHEMES AND INDUCTION STATEMENTS</td>
<td>134</td>
</tr>
<tr>
<td>2.14</td>
<td>GENERALISATIONS</td>
<td>139</td>
</tr>
</tbody>
</table>

CHAPTER THREE

PATTERN GENERATION

3.1 INTRODUCTION | 144 |
3.2 CONCEPT OF PROTOMORPHS | 145 |
3.3 PROTOMORPHS FOR SINGLE LAYER CONFIGURATIONS | 147 |
3.4 PROTOMORPHS FOR DOUBLE LAYER CONFIGURATIONS | 170 |
| 3.4.1 | BINATE ON BINATE PATTERNS | 171 |
| 3.4.2 | OBNATE ON OBNATE PATTERNS | 174 |
| 3.4.3 | BINATE ON OBNATE PATTERNS | 176 |
| 3.4.4 | OBNATE ON BINATE PATTERNS | 178 |
| 3.4.5 | TERNATE ON TERNATE PATTERNS | 178 |
| 3.4.6 | HEXAGONAL ON HEXAGONAL PATTERNS | 182 |
| 3.4.7 | TERNATE ON HEXAGONAL PATTERNS | 186 |
CHAPTER FOUR
PARAGENIC TRANSFORMATIONS
4.1 INTRODUCTION 223
4.2 CYLINDRICAL AND SPHERICAL RETRONORMS 223
4.3 DOUBLE RETRONORMIC TRANSFORMATIONS 229
  4.3.1 CYLINDRICAL PARAGENIC TRANSFORMATIONS 232
  4.3.2 SPHERICAL TRANSFORMATIONS OF CYLINDRICAL
      RETRONORMS 279
  4.3.3 SPHERICAL PARAGENIC TRANSFORMATIONS 305
  4.3.4 CYLINDRICAL TRANSFORMATIONS OF SPHERICAL
      RETRONORMS 308

CHAPTER FIVE
CONCLUSIONS AND PROPOSALS FOR FUTURE WORK
5.1 INTRODUCTION 321
5.2 CONCLUSIONS 321
5.3 FUTURE WORK 325

REFERENCES 328
CHAPTER ONE

A BRIEF EXAMINATION OF DIFFERENT TYPES OF SPACE STRUCTURES

1.1 INTRODUCTION

Architects and engineers have shown special interest in structural systems which enable them to cover large spans. Sports stadia, exhibition halls, swimming pools, shopping malls and industrial buildings are typical examples of structures where large unobstructed areas are essential with minimum interference from internal supports. Since the early seventies, space structures have been used to provide economical solutions for large span structures. The architectural scene all over the world has been influenced by space structures largely due to the availability of advanced constructional techniques, new structural materials and outstanding technological developments in the last two decades. The progress made in the field of high speed electronic computers has been a major contributing factor to aid the manner in which sophisticated structures may be designed and analysed accurately and more conveniently than with previously available techniques.

Some examples of recently constructed space structures for a variety of building types are shown in Figs 1.1 to 1.4. Space structures may be broadly grouped under three main categories. The first group consists of spaceframes, which include single layer, double layer and multilayer configurations used in the form of grids, vaults and domes. The next group consists of tension structures, comprising of prestressed cable nets and membranes. The third group consists
**Fig 1.1** Internal View of the Orio Gymnasium, Japan completed in 1909. Height of the spaceframe is 26 metres covering an area of 3070 sq metres.

**Fig 1.2** Performance stage and Amphitheatre at the Theme Park, Nemunosato, Japan.

**Fig 1.3**
INTEX Osaka International Exhibition Centre, 1995 with a spaceframe covering an area of 74,918 sq metres.

**Fig 1.4** PVC coated polyester fabric used for air supported structures at the Silk Road Exhibition.
of "hybrid space structures" which are combinations of spaceframes and tension structures. A few significant examples in each category have been discussed in the present Chapter.

1.2 SPACEFRAMES

A spaceframe may be described as a skeletal framework consisting of bar elements interconnected to form a three dimensional assembly. Spaceframes have been constructed in the form of grids, vaults, domes and various curved surfaces using single layer, double layer and multilayer configurations. The components of a spaceframe are often standardised and prefabricated so that they can be mass produced to make them cost efficient. Different types of commercially available systems with standardised components are being used all over the world.

A double layer grid consists of members arranged in two layers placed parallel to each other and interconnected by web members. The two layers may or may not have identical patterns. The arrangement of members in grids can be varied to give rise to different patterns. These patterns will be discussed in detail in Chapter Three on "Pattern Generation". Some significant double layer grids have been described below.

The world's largest aluminium structure is a double layer grid built for the Exhibition Centre at Sao Paolo, Brazil, shown in Fig 1.5. Designed by Cedric Marsh, the structure covers an area of 260 metres × 260 metres. The double layer square grid with nodes spaced at 3.3 metres has a structural depth of 2.4 metres, Ref 1.

Another significant application of double layer grids is for two large span hangars at Heathrow airport, London, UK, constructed for the Boeing 747 jets.
Fig 1.5 260 metres x 260 metres exhibition hall roof in Sao Paulo, Brazil.

Plan of the BA hangar 01. 1, fascia girder. 2, spine girder. 3, back girder. 4, main columns. 5, stabilising ties to bottom chords. 6, crane runways. 7, roof lights. 8, ancillary buildings. 9, fork brace to fascia girder overhang. 10, roof purlins. 11, door hood. 12, gutters. 13, expansion joints.

Fig 1.6 Overall dimensions of the British Airways Hangar at London Airport, UK.
The structural system consists of prefabricated lattice type diagonal grids. A plan view of the hangar is shown in Fig 1.6. Each hangar provides an uninterrupted span of 138 metres. A detailed description of the roof structure for the hangars is available in Ref 1. This type of double layer grid pattern proved to be so successful that a number of designers adopted a similar pattern for aircraft hangars all over the world.

The first recorded and documented attempts to develop multilayer prefabricated grids were those by Alexander Graham Bell, the famous inventor of the telephone, who in 1906, along with several colleagues, built powered kites and biplanes constructed as three dimensional prefabricated structures. In 1907, Graham Bell built several experimental multilayer grid structures consisting of members of the same length, joined together by simple connectors, identical for all nodes. His structures were based on prefabricated tetrahedral units, Fig 1.7. His reports refer to the "extraordinary" strength of such prefabricated space grids. He seems to be the first engineer to show over eighty years ago, how to make lighter, stronger and economical structures consisting of mass produced elements, Ref 1.

In the last two decades, there have been thousands of single layer, double layer and multilayer grids built throughout the world. A comprehensive account of developments in the design, analysis and construction of double layer grids is available in Ref 2.

The rapid acceptance of single layer and double grids led to the development of spaceframes for curved surfaces such as vaults and domes. An early example of a prefabricated single layer barrel vault was the famous iron barrel vault designed by Joseph Paxton, Fig 1.8. The vault was erected over the central nave of the Crystal Palace for the Great Exhibition of 1851, in Hyde Park, London, UK. This structure consisted of a rectangular grid of
Fig 1.7 Multi layer spaceframe used in 1907 by Graham Bell in his flying machines.

Fig 1.8 Internal view of the iron barrel vault built in 1851 for the Crystal Palace.
prefabricated modular semi-circular ribs and longitudinal members, stiffened by slender diagonal iron rods, Ref 3.

Another well known example of the successful use of barrel vaults is the hangar built in 1935, in Cecchignola, near Rome. The hangar covers an area of 102.5 metres × 39.6 metres and is supported at only six points, as shown in Fig 1.9, Ref 3. The barrel vault was designed by the famous Italian designer Pier Luigi Nervi. Several large span barrel vaults were built by Nervi, using precast, prefabricated modular reinforced concrete units. These units were interconnected on site through welding of the protruding steel reinforcing bars at the nodal points. Reinforced concrete slabs were cast as the top layer, which became an integral part of the cylindrical barrel vault structure.

In 1942 the MERO system became the first commercially available space structure system. Developed by Dr Mengeringhausen, this system allows up to 18 bars to be connected to the same node without any eccentricity. The MERO system has been used for numerous structures throughout the world. One of the most remarkable structures built using the MERO system is the roof of the stadium at Split in Croatia for the 1979 Mediterranean Games, Fig 1.10. Two wide span double layer barrel vaults are designed to cantilever 45 metres over the spectator areas and are supported only along the perimeter. Each vault has a span of almost 215 metres. The spaceframe is based on a 3 metres × 3 metres module with a structural depth of 2.3 metres. Transparent acrylic vaulted units are used as cladding, Ref 4.

Barrel vaults and domes have often been used in combination to create more complex structural forms. For example, the building for the Liverpool International Garden Exhibition is a composition of two half domes and a vault. Fig 1.11 shows the arrangement of structural members. The two single layer half domes are separated by a double layer barrel vault, Ref 5. A detailed
Fig 1.9 A view of Nervi's hangar

Fig 1.10 Bird's eye view of the Split stadium
Fig 1.11 The arrangement of the structural framework for the Liverpool Garden festival building

Fig 1.12 The world's largest clear span aluminium dome
study on the methods of structural analysis and construction of a variety of braced barrel vaults built world-wide is available in Ref 6.

Various types of single layer and double layer configurations and combinations of trussed patterns have been successfully used for domes. Also, stressed skin domes where the covering is connected to the bracing members and forms an integral part of the load carrying system have been used. Another class of domes which are called surface domes are constructed using thin steel, aluminium or glass fibre reinforced plastic sheets which are bent and interconnected to form the main bracing system.

The world’s largest single layer aluminium dome was built in 1981. The dome was designed by Don Richter with a span of 126 metres and a height of 40 metres, Fig 1.12. The dome is designed to house a huge flying boat as seen in Fig 1.12. A triangular bracing pattern is used for the dome and 0.127 mm thick aluminium panels are used for cladding the structure, Ref 7.

Double layer triangulated configurations have been used to construct two of the world’s largest domes. The New Orleans Superdome is the world’s largest dome with a clear span of 213 metres and the second largest dome is the Houston Astrodome, with a clear span of 200 metres shown in Fig 1.13. Both domes were designed by Dr G R Kiewitt.

An example of a trussed dome is that of the Indraprastha Stadium which was built for the Asian Games in New Delhi, India in 1982, Fig 1.14. The dome has a diameter of 150 metres, making it the third largest dome in the world and the largest indoor stadium in Asia. The stadium was completed in a record time of 22 months from the time of its conceptual design to the day of its official inauguration. The stadium has a seating capacity of 25,000 spectators and an arena size of 78 metres $\times$ 60 metres, furnished with facilities required
Fig 1.13 The Houston Astrodome during erection

Fig 1.14 An external view of the Indraprastha stadium in New Delhi, India
for Olympic standards. The dome covering the stadium has a collapsible 100 ton soundproof curtain, 150 metres long and 40 metres high, enabling the subdivision of the huge stadium into two independent areas. The steel dome resembles a folded plate structure, consisting primarily of trusses radiating from a central octagonal compression ring 40 metres in diameter, Fig 1.15. The compression ring itself is formed by a lattice work of trusses. In plan, the roof is like an eight pointed star within a circle. Roof ridges and troughs that extend from the centre to the perimeter are formed by steel work of lattice trusses. The cladding used is corrugated aluminium sheeting. The roof is supported by eight 40 metres tall reinforced concrete towers placed along the perimeter. These towers also act as main service cores, housing staircases, air conditioning and lighting services. About 1,800 tonnes of steel was fabricated on site. All the in situ joints were designed and fabricated as bolted joints using high tensile bolts. However, welding was adopted for all shop connections and splicing, Ref 8.

The famous United States Pavilion at the 1967 Montreal Exhibition is an impressive example of a double layer geodesic dome built in tubular steel, Fig 1.16. Designed by Buckminster Fuller, the three quarter sphere is 76 metres in diameter. The external layer of the grid is fully triangulated and the inner layer is a hexagonal grid.

A comprehensive source of information for architects and engineers who wish to follow the developments in recently constructed braced domes is available in Ref 9.
Fig 1.15 A part plan of the Indraprastha stadium

Fig 1.16 The United States Pavilion at Expo '67 in Montreal, Canada
1.3 TENSION STRUCTURES

The demand for lightweight large span roofs, unobstructed by columns led to the development of another family of space structures referred to as tension structures consisting of nets of steel cables and membranes. Tension structures may be described as structural systems where the principal load carrying elements are in tension supported between anchorages. In general, cables, membranes and bar elements have been used in different combinations to achieve a wide variety of tension structures some examples of which have been described below.

The development of cable net roofs stabilised by double curvature and pretensioning were initiated by the success of the Raleigh Arena in North Carolina, USA. Designed by Nowicki and Severud in 1953, the roof over the arena was made up of a network of steel cables spanning between two reinforced concrete arches. Fig 1.17 shows a perspective view and an elevation of the structure. The roof of this arena was a tremendous source of inspiration for many architects and the following decade saw many similar projects. Two architects in particular developed the basic principles behind Raleigh to produce unique and even more striking large span spaces - Saarinen’s Yale Hockey Rink completed in 1956 and Kenzo Tange’s two Olympic stadia in Tokyo built in 1961.

The work of Frei Otto in Germany has been regarded as an important contribution in stimulating an understanding of the possibilities of shapes using cable nets and membranes. Otto studied different families of tensile structures with the help of small scale models. An account of Otto’s experiments is found in the book "Das hangende Dach", (Hanging Roofs) which was submitted as his doctoral dissertation at the Technical Institute of Berlin in 1953. The most interesting and important research work stemmed from Otto’s study of soap
Fig 1.17 The Raleigh Arena, North Carolina, USA

Fig 1.18 A view of the Montreal Pavilions, Canada
films since they represented surfaces of uniform tension with almost any set of boundaries, Ref 11.

Important cable net structures built by Otto include the German pavilion in Montreal, for Expo '67 shown in Fig 1.18 and the roof of the stadia for the Olympic games in Munich shown in Fig 1.19, Ref 12. The pavilions consists of a stressed cable net system covering an area of 7,700 m². Eight masts of varying heights, bounded by 30 edge cables supported the doubly curved roof. A PVC coated polyester fabric skin hangs 35 - 40 cms below the cable net formed of 12 mm thick steel cables. For the 1972 Olympic games in Munich, Frei Otto designed a cable net roof over the stadium, sports arena and the swimming pool. The prestressed cable net roof covered an area of 74,800 m². The cladding material was translucent acrylic glass, used in sheets flexibly linked together.

In many situations, cable net systems and membranes have become almost standard solutions for a wide variety of temporary and permanent coverings. The availability of Teflon coated fibre glass membranes over the last decade has enabled architects to design tension structures where the membrane itself is highly prestressed. Translucent weather proof and durable membranes have replaced the need for a cable net and a separate weather proof cladding. Two notable examples of the applications of Teflon coated fibre glass membranes are the roof for the stadium in Riyadh, Fig 1.20 and the roof for the Hajj Terminal in Saudi Arabia, Fig 1.21.

The roof of the Riyadh International Stadium in Saudi Arabia, completed in May 1985 was the largest span roof structure at that time. Teflon coated fibreglass membrane was used for the 24 tent shaped units. A ring cable of 134 metres diameter and 24 main masts support the entire roof, Ref 13.
Fig 1.19 The Munich Olympic games sports complex

Fig 1.20 The Riyadh Stadium, Saudi Arabia
Fig 1.21 The roofs of the Hajj Terminal, Saudi Arabia

Fig 1.22 The United States Pavilion at Expo '70 in Osaka, Japan
The largest fabric roof in the world, covering about 105 acres is at the Hajj Terminal in Saudi Arabia. The roof structure of the Haj Terminal consists of 210 tent units, square in plan spanning 45 metres. Each unit is supported by 32 radial cables and valley cables along the edges. The cables are suspended from a system of steel pylons, 45 metres in height.

In the tension structures discussed so far, the prestress in membranes and cable-nets was induced by tensioning the surface via boundary cables and supporting elements. However, another family of tension structures are created by using air pressure for tensioning the membrane. These are referred to as "pneumatic structures" and these fall under two broad categories. Pneumatic structures may either be air supported or air inflated. In the case of air supported structures, the entire enclosure is maintained at a small pressure differential to support and prestress the membrane. Air is supplied continuously by blowers to maintain a constant air pressure inside a totally sealed environment. The increased air pressure is only 1/4% greater than atmospheric pressure and is unnoticed by occupants.

In an air inflated system, the main support is obtained by air pressure inside two membrane layers which are held apart by internal air pressure as well as drop chords. The inflated membranes form an air beam or an air mat type construction. As such, the occupied space is not pressurised. Air inflated structures operate at much higher pressures than do air supported structures. A brief history of the early developments in pneumatic structures is presented below.

With the advent of electrically driven blowers, F W Lanchester in a 1917 patent proposed an air supported building for use as a portable field hospital whereby one entered the pressurised space by means of an air-lock. It was not until 1946 that the concepts suggested by Lanchester were put into practice by
Walter Bird in the construction of a pneumatic structure 15 metres in diameter built to protect military radomes from the weather while allowing for the transmission of radar waves through the fabric, Ref 14.

The first major pneumatic structure was built for the United States Pavilion at Expo '70 in Osaka, Japan, Fig 1.22. It was designed by David Geiger. The membrane covering an area of 9,300 m² was a white, transparent, vinyl-coated fibreglass fabric 2.33 mm thick. There are 32 reinforcing cables in two directions, arranged in a diamond shaped pattern and spaced 6.1 m apart, Ref 17.

An example of an air inflated structure is the Fuji Pavilion at Expo '70 designed by Murata and Kawaguchi. It consisted of 16 air-inflated semi-circular arches each 4 metres in diameter and 72 metres long, connected to a concrete ring foundation 50 metres in diameter.

Another impressive air supported structure designed by David Geiger is the roof of the Pontiac Silverdome Stadium in Michigan, USA. The stadium covers an area of 35,000 m² without any intermediate columns, Fig 1.23. The membrane comprises a glass fibre fabric coated with Teflon spanning between an oblique cable net anchored to a polygonal compression ring.

An example of the application of negative pressure to a membrane, is the tensile structure of the Franklin Park Zoo in Boston. Fig 1.24 shows a plan and elevation of the structure. Designed by Huygens and Tappe, the roof consists of a doubly curved surface with the membrane reinforced and prestressed using primary cables, Ref 15.

Various combinations of cables and struts have been experimented with to give rise to three dimensionally rigid systems referred to as "tensegrity systems".
Fig 1.23 The Pontiac Silverdome stadium, Michigan, USA

Fig 1.24 The Franklin Park Zoo, Boston, USA
These are characterised by a set of discontinuous compression elements which interact with a set of continuous tensile elements to create a stable structure in space. The system can be prestressed by lengthening some of the bars or shortening some of the cables. The bars are arranged in such a way that they cross each other but none of the bars actually touch each other nor are they connected by nodes. The roof of the American Pavilion at the 1958 Brussels exhibition was a tensegrity system and was very similar to a great bicycle wheel. A circle of columns support a 104 metre diameter steel ring. In the centre is a drum shaped "hub" supported by two sets of radially stressed cables, like "spokes", Ref 17.

1.4 HYBRID SPACE STRUCTURES

The term "hybrid space structures" has been used to refer to different classes of structures created by using combinations of spaceframes and tensile elements. The rigidity and ability to cover long spans, which is characteristic of spaceframes, can be combined with the lightweight, translucency and sculptural forms of membranes to create hybrid configurations. However, the rigidity of the spaceframe or space truss is the major contributing factor in the integration of the two systems. It helps to reduce and control flutter inherent in membranes and cable nets, without penalising the whole assembly with a substantial weight increase.

A project completed in Venafro, Italy in 1990 is an example of a hybrid space structure where horse shoe shaped arched trusses support intermediate membranes. Fig 1.25 shows a view of the model of the structure. Designed by P Samyn the structure houses a research laboratory within the large oval plan measuring 80 metres × 35 metres. The membrane integrates steel cables along its surface and at the edges. Six horse shoe shaped metal arches are held at the base by six longitudinal cables as shown in Fig 1.26. The steel arches
Fig 1.25 The model of the Research Centre at Venafro, Italy

Fig 1.26 Arrangement of members for the hybrid structure

Fig 1.27 An Indoor Hockey Rink in Japan
are made of tubular three dimensional trusses, maintained transversely by six prestressed steel cables.

Another example of a hybrid structure is the Indoor Horse Riding Rink in Japan. Completed in October 1986, it encloses an area of 44 metres × 22 metres. The building creates a sense of being outdoors due to the soft diffused light penetrating through the membrane into the interior, Ref 19. The membrane is hung from the eight top layer nodes of the double layer grid. The edges of the membrane are fixed continuously to the boundary beam. To ensure a watertight surface and an even distribution of stresses, one large continuous sheet of membrane was used. The frame is built up of steel pipes. The building is finished using decorative concrete blocks, Fig 1.27.

The examples discusses in this Chapter show structures where good engineering practice and architecture form a symbiotic relationship. This also shows the remarkable progress made in the design and analysis of space structures. Advancements in materials available and modern construction techniques have made it possible to construct different classes of space structures. However, there is an urgent need for a general strategy to help architects and engineers in their search for different shapes and forms. Extensive research is still required in order to have a methodology for generating and transforming structural forms in a convenient manner. The present research uses computer aided techniques in an attempt to find solutions to these problems.

In the next Chapter, the emphasis is on how the concepts of a mathematical tool called "formex algebra" can be used conveniently for the creation and representation of structural forms. Some of the basic concepts of formex algebra and its programming language Formian have been demonstrated with a number of examples.
CHAPTER TWO

CONCEPTS OF FORMEX ALGEBRA
AND
FORMIAN

2.1 INTRODUCTION

Remarkable progress in computer aided techniques has made it possible for architects and engineers to design and realise more and more complicated and innovative structural forms with ease and elegance. The visualisation and analysis of any structure, including a space structure, on a computer, requires information about various aspects of the structural system. This information could be initially used for graphical visualization of the structure, or may be submitted as input data to an analysis package.

To begin with, the architect may produce a preliminary sketch of the structure. An essential requirement in the design process is to be able to represent the architect’s preliminary sketch in a precise manner. Also, it is important to have a convenient system for manipulating and changing various aspects of the form to examine different solutions. For large and complex structural forms, the shear volume of information to be handled can make data generation a time consuming and error prone task. To overcome this problem, suitable systems have been developed by which computer graphics and data generation for any type of structure can be done conveniently. Formex algebra is one such mathematical system. The ideas of formex algebra can be applied to many branches of science and technology, in the present work the concepts of formex algebra have been described in relation to a variety of space structure
configurations.

The basic philosophy behind formex algebra originated during the period 1972-73. A record of these ideas may be found in Ref 20. A major revision of these concepts was introduced in 1979 and later published in 1981, Ref 21. The first text book on formex algebra appeared in 1984, Ref 22.

Formex algebra has proved to be a medium for communication between people as well as a medium for communication between people and computers. The concepts of formex algebra have been taught, learnt, discussed and recorded using the well defined terminology and notation, evolved gradually during the last two decades. The present Chapter describes some of the basic concepts of formex algebra with the help of a number of examples.

2.2 FORMEX CONFIGURATION PROCESSING

The term "configuration" is used to mean an "arrangement of parts". The elements of a structure for instance, constitute a configuration and so do the parts that form an electrical network. The most common usage of the term configuration is in reference to a geometric composition consisting of points and/or lines and/or surfaces. Such a geometric composition may itself be the subject of study or it may be representing another arrangement of objects.

Configurations may be described using numerical models. In particular, when digital computers are involved, the internal representation of a configuration is bound to be in terms of a numerical model. The term "configuration processing" is used to mean "creation and manipulation of numerical models representing configurations", Ref 23. Formex algebra is a mathematical system which is ideally suited for configuration processing.
In using the formex approach, a "formex" (plural formices) is used to represent a configuration. The main role of a formex is to provide information regarding the constitution of a configuration. In addition, a formex may be used to provide geometric information about a configuration in terms of coordinates of the nodal points. For a better understanding of this concept, it is essential to know the primary components of a formex and what they represent. These have been elaborated in the next subsection.
2.3 FORMEX CONSTITUENTS

Consider the configuration shown in Fig 2.1. Let the part of the configuration shown in bold lines, be represented by a formex. Then the formex may be written as

\{[2,3; 2,2],[2,2; 3,2],[3,2; 4,4],[4,4; 2,2],[4,4; 2,3]\}

The structure of the formex primarily consists of a series of "cantles", enclosed in square brackets as elaborated in Fig 2.2. The number of cantles in a formex is referred to as the "order" of the formex. Therefore, the above formex is of the fifth order. Graphically, each cantle represents a primary component of a configuration. For instance, the first, second and third cantles in the formex represent the elements marked 1, 2 and 3 in the configuration shown in Fig 2.1. Each cantle consists of "signets", separated by semicolons. In the above formex, each cantle consists of two signets. Graphically a signet represents a point relative to a reference system, shown in dotted lines in Fig 2.1. This reference system which consists of two families of intersecting lines, is called a "normat", and the lines are called "normat" lines. The directions in which the normat lines run, are indicated in Fig 2.1. The points of intersection of these lines are called "normat points". So a signet represents a normat point.

The numeric constants that constitute a signet are separated by commas and are referred to as "uniples". For example, 2 and 3 are the first and second uniples of the first signet of the formex. The number of uniples in a signet, is referred to as the "grade" of the signet. The above formex is said to be of the fifth order and second grade.

The number of signets that constitute a cantle is referred to as the "plextitude" of the cantle. For example,
Fig 2.1

cantle cantle

\{
[2,3; 2,2],[2,2; 3,2],[3,2; 4,4],[4,4; 2,2],[4,4; 2,3]
\}

cantle cantle

signet signet signet

Fig 2.2 Constituents of a formex

Fig 2.3 Series of points represented by an ingot
[2,-4,1; 3,7,0; 5,5,2; 6,-1,-5]

is referred to as a formex of the third grade and fourth plextitude.

A formex is referred to as a "homogeneous" formex if all its cantles are of the same plextitude. For example,

{[3,-1; 9,0],[6,6; -3,5],[3,-8; -10,4]}

is a homogeneous formex of the second plextitude. An example of a "non homogeneous" formex may be given by the construct,

{[1,2],[6,-3; 5,5],[3,-5; 9,10; 12,-4]}

A homogeneous formex of the first plextitude is referred to as an "ingot". For instance,

{[2,4],[0,1],[2,2],[4,1]}

is an example of an ingot. This ingot may represent a series of points as shown in the configuration of Fig 2.3. The two directions of the normat lines are indicated by U1 and U2 as shown in Fig 2.3.

A formex of the first order is referred to a "maniple", and is written without the enclosing curly brackets. A maniple of the first plextitude is referred to as a "reglet". For instance,

[2,1; 1,3; 2,7; 3,3]

and

[5,7,1; 12,6,-2; 3,-4,-1]
are examples of two maniples. The first one consists of four signets of the second grade and may represent a four noded finite element as shown in Fig 2.4. The second one represents a three ended element relating to a three directional normat which will be explained later.

Amongst all the set of formices, there exists a formex that has no maniples, this special formex is referred to as an "empty formex", and is indicated by,

\[
\{ \} 
\]

The empty formex may not appear to be very useful at this stage, but the empty formex is similar to the number zero appearing in natural numbers.

Consider a formex

\[
\{[2,1,1; 3,-3,1],[9,5,0; 5,-4,0],[2,2,8; 4,7,1]\}
\]

The serial position of a cantle in a formex is referred to as the "ordrate" of that cantle. Therefore the orderates of \([2,2,8; 4,7,1]\) and \([2,1,1; 3,-3,1]\) with respect to the above formex are 3 and 1 respectively.

2.4 **EQUALITY OF FORMICES**

Two formices are said to be equal provided that they are of the same grade and the same order. Also they must be identical which implies that every uniple in one is equal to the corresponding uniple in the other. For example,

\[
\{[a,s,e,b]\} = \{[2,6,-8,1]\}
\]

implies that \(a=2\), \(s=6\), \(e=-8\) and \(b=1\).
Fig 2.4 A four noded finite element

Fig 2.5 A formex plot
2.5 VARIANTS OF A FORMEX

Two formices are said to be variants of each other provided that every cantle in one is obtained from the corresponding cantle of the other by simply rearranging the positions of its signets. For example,

\[ F = \{[2,2; 6,-1],[4,4; 0,6]\} \]

and

\[ G = \{[6,-1; 2,2],[4,4; 0,6]\} \]

are variants of each other. Sometimes, a formex could have cantles which are variants of each other. For example, consider the formex,

\[ H = \{[1,0; -9,6],[4,7; 5,-2],[-9,6; 1,0],[0,0; 4,8]\} \]

The first and third cantles in the above formex are variants of each other, such a formex is referred to as a "prolate" formex. The formex G above, then, is called a "nonprolate" formex.

2.6 SEQUATIONS OF A FORMEX

Two formices are said to be sequations of each other if one may be obtained from the other by a rearrangement of its cantles. For instance,

\[ H_1 = \{[7,7; -8,3],[2,2; 1,-2],[5,-3; 1.5,1.5]\} \]

and

\[ H_2 = \{[5,-3; 1.5,1.5],[7,7; -8,3],[2,2; 1,-2]\} \]

are sequations of each other.
2.7 COMPOSITION OF FORMICES

If $G_1$ and $G_2$ are two formices of the same grade, then they can be combined to form a formex, say $G$. The new formex contains all the cantles of $G_1$ appearing in the same order as in $G_1$, followed by all the cantles of $G_2$ appearing in the same order as in $G_2$. The relationship between $G$, $G_1$ and $G_2$ can be written as,

$$G = G_1 \# G_2$$

the symbol $\#$ is referred to as the "duplus" symbol, and is read as "duplus". For example, if

$$G_1 = \{[4, -2; 5,10],[12,-9; 6,13]\}$$

and

$$G_2 = \{[2,-10; 4,-7],[2,2; 1,0]\}$$

then the formex $G = G_1 \# G_2$ is given by,

$$G = \{[4, -2; 5,10],[12,-9; 6,13],[2,-10; 4,-7],[2,2; 1,0]\}$$

Formex compositions have the following basic properties:

(1) If $A$ and $B$ are two formices of the same grade, then

$$A \# B = B \# A$$

This means that formex composition is not commutative.

(2) If $A$, $B$ and $C$ are formices of the same grade, then
\[ A \# (B \# C) = (A \# B) \# C \]

This means that formex composition is associative.

(3) For any formex \( A \),

\[ A \# \{\} = \{\} \# A \]

(4) If \( A \) and \( B \) are two formices of the same grade, then

\[ A \# B \]

and

\[ B \# A \]

will give two formices which are sequations of each other.
2.8 FORMEX GRAPHICS

2.8.1 INTRODUCTION

A formex may be used to represent any configuration. A graphical representation of a formex is referred to as a "plot" of the formex. For instance, Figs 2.1, 2.3 and 2.4 are all examples of formex plots. The interrelationship between formices and geometric configurations plays a vital role in the practical applications of formex algebra. Some of the aspects of this interrelation have been discussed below.

2.8.2 FORMEX PLOTS

A formex plot may be obtained by using certain sets of rules. Firstly, there are rules through which the particulars for the representation of the signets and cantles are determined. An aspect of a rule of this type is referred to as a "retrocord". Secondly, there are rules through which the signets in the formex are mapped into points in a coordinate system. Such a set of rules is referred to as a "retronorm".

As an example, consider the formex

\[ D = \{[2,1; 1,3; 3,3],[8,1; 7,3; 9,3],[3,3; 7,3],[3,3; 5,9],[5,9; 7,3]\} \]

A plot of \( D \) is given in Fig 2.5. The part of the plot that represents a signet is referred to as a "tenon" and the part of the plot that represents a cantle is referred to as a "frond". Here, a tenon is represented as a little circle, and a frond is represented as a straight line as shown in Fig 2.6. An arrowhead is placed on the straight line to indicate the order of appearance of the signets in the cantle. The plot in Fig 2.6 consists of seven tenons and five fronds. Here,
Fig 2.6 A formex plot

Fig 2.7 A formex plot
the fronds representing the cantles of the second plextitude consist of two tenons and a straight line, while the fronds representing the cantles of the third plextitude consist of three tenons and three straight lines. A tenon may belong to a frond exclusively, or may belong to a number of fronds simultaneously. For instance, the peripheral tenons in Fig 2.6 are exclusive tenons, while each of the two inner tenons belong to two fronds simultaneously.

By defining a particular retrocord, the tenons in Fig 2.6 are chosen to be drawn as little circles. But a tenon or a frond may be represented in an infinite number of ways by choosing the appropriate retrocord. For instance, a tenon may be represented as a shape resembling the node of a structure, a star or a flower, while a frond could be represented as a structural member, an arch or a coil to suit a specific application. If the particulars regarding the shape of the tenons or fronds are to be turned into instructions for a plotting machine, then supplementary to the retrocord that specifies the shape, one can have a retrocord to specify the actual size, colour and other details regarding the appearance of the tenons and fronds.

Every tenon is drawn relative to a point which is referred to as its "pivot". In the plot of D, the centres of the little circles are the pivots which are located by specifying its coordinates with respect to a coordinate system using the equations,

\[ X = U_1 \quad \text{and} \quad Y = U_2 \]

Coordinate equations of this type specify the coordinates of a pivot in terms of the uniples of a typical signet. So, if the signet \([U_1, U_2]\) was represented by the coordinate equations

\[ X = U_2 - U_1 \quad \text{and} \quad Y = U_1 + 2 \]
the resulting configuration, shown in Fig 2.7, will be another plot of D.

Now, consider the same formex D, and let the retronorm be given by the coordinate equations

\[ X = U_2 \text{ and } Y = U_1 \]

The resulting configuration, shown in Fig 2.8, is yet another plot of D.

Once again, consider the formex D, but this time let the pivots be represented in terms of polar coordinates, using the equations

\[ r = U_2 \]
\[ \theta = (U_1 - 2)\pi/8 \]

The configuration thus obtained, shown in Fig 2.9, is also another plot of D.

It is possible to plot a formex of any grade with respect to a one, two or three dimensional coordinate system. For instance, consider a formex of the second grade,

\[ E = \{[2, -1; 4, 1], [2, 0; 4, 5], [3, -6; 4, 1], [3, -6; 4, 5], [2, -1; 3, -6], [2, 0; 4, 1]\} \]

A plot of E with respect to a one dimensional coordinate system may be obtained as shown in Fig 2.10. Here the retronorm for a typical signet is given by the equation,

\[ X = 2U_1 - U_2 \]
Fig 2.8 A formex plot

Fig 2.9 A basipolar retronorm

Fig 2.10
The fronds in the plot are arched so as to avoid overlapping.

In the examples given so far, plots of formices of the first and second grade were obtained. Now consider a formex $E$ of the third grade where

$$E = \{[1,2,0; 2,2,0; 2,1,0; 1,1,0; 1,1,0; 1,2,0],
        [1,1,0; 1,1,1; 1,1,0; 1,1,0; 1,1,0; 1,1,0; 1,1,0; 1,1,0];
        [1,1,1; 1,2,1; 1,2,1; 2,2,0; 1,2,1; 1,2,0] \}$$

The plot of $E$ with respect to a three dimensional coordinate system is as shown in Fig 2.11. Here the retronorm is given by the equations

$$X = U_1, \ Y = U_2 \text{ and } Z = U_3$$

Plots of formices of different grades may be obtained using different retronorms to specify the location of pivots.

There are six categories of retronorms that relate to commonly used coordinate systems. They are described as follows:

(1) A "unifect retronorm", relates to a one dimensional Cartesian coordinate system and is defined by coordinate equations of the form

$$x = f_1(U_1, U_2, \ldots, U_n)$$

where $(U_1, U_2, \ldots, U_n)$ is a typical signet. An example is the plot in Fig 2.10.

(2) A "bifect retronorm" relates to a two dimensional Cartesian coordinate system and is defined by coordinate equations of the form
Fig 2.11 A cylindrical coordinate system

Fig 2.12 A spherical coordinate system
\begin{align*}
x &= f_1(U_1, U_2, \ldots, U_n) \\
y &= f_2(U_1, U_2, \ldots, U_n)
\end{align*}

The plots in Figs 2.7 and 2.8, show the use of a bifect retronorm.

(3) A "trifect retronorm" relates to a three dimensional Cartesian coordinate system and is defined by coordinate equations of the form

\begin{align*}
x &= f_1(U_1, U_2, \ldots, U_n) \\
y &= f_2(U_1, U_2, \ldots, U_n) \\
z &= f_3(U_1, U_2, \ldots, U_n)
\end{align*}

An example is the plot in Fig 2.11.

(4) A "polar retronorm" relates to a polar coordinate system and is defined by coordinate equations of the form

\begin{align*}
r &= f_1(U_1, U_2, \ldots, U_n) \\
\theta &= f_2(U_1, U_2, \ldots, U_n)
\end{align*}

An example is the plot in Fig 2.9.

(5) A "cylindrical retronorm" relates to a cylindrical coordinate system and is defined by coordinate equations of the form

\begin{align*}
r &= f_1(U_1, U_2, \ldots, U_n), \\
\theta &= f_2(U_1, U_2, \ldots, U_n) \\
z &= f_3(U_1, U_2, \ldots, U_n)
\end{align*}

where \( r, \theta \) and \( z \) are as shown in Fig 2.12.
(6) A "spherical retronorm" relates to a spherical coordinate system and is defined by coordinate equations of the form

\[ r = f_1(U_1, U_2, \ldots, U_n), \]
\[ \Theta = f_2(U_1, U_2, \ldots, U_n) \]
\[ \Gamma = f_3(U_1, U_2, \ldots, U_n) \]

where \( r, \Theta \) and \( \Gamma \) are as shown in Fig 2.13.
2.8.3 STANDARD RETRONORMS

There are three families of "standard retronorms". These are special cases of the above retronorms and are called "basiant retronorms", "pariant retronorms" and "metriant retronorms". The "basiant retronorms" are the first family of standard retronorms which are further divided into six types. Table 2.1 includes particulars of basiant as well as metriant retronorms.

Table 2.1

<table>
<thead>
<tr>
<th>NAME</th>
<th>COORDINATE EQUATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basiunifect</td>
<td>$x = b_1 U_1$</td>
</tr>
<tr>
<td>Metriunifect</td>
<td>$x = b_1 \text{met}(U_1, m_1)$</td>
</tr>
<tr>
<td>Basibifect</td>
<td>$x = b_1 U_1, y = b_2 U_2$</td>
</tr>
<tr>
<td>Metribifect</td>
<td>$x = b_1 \text{met}(U_1, m_1), y = b_2 \text{met}(U_2, m_2)$</td>
</tr>
<tr>
<td>Basitrifect</td>
<td>$x = b_1 U_1, y = b_2 U_2, z = b_3 U_3$</td>
</tr>
<tr>
<td>Metritrifect</td>
<td>$x = b_1 \text{met}(U_1, m_1), y = b_2 \text{met}(U_2, m_2), z = b_3 \text{met}(U_3, m_3)$</td>
</tr>
<tr>
<td>Basipolar</td>
<td>$r = b_1 U_1, \theta = b_2 U_2$</td>
</tr>
<tr>
<td>Metripolar</td>
<td>$r = b_1 \text{met}(U_1, m_1), \theta = b_2 \text{met}(U_2, m_2)$</td>
</tr>
<tr>
<td>Basicylindrical</td>
<td>$r = b_1 U_1, \theta = b_1 U_2, z = b_3 U_3$</td>
</tr>
<tr>
<td>Metricylindrical</td>
<td>$r = b_1 \text{met}(U_1, m_1), \theta = b_2 \text{met}(U_2, m_2), z = b_3 \text{met}(U_3, m_3)$</td>
</tr>
<tr>
<td>Basispherical</td>
<td>$r = b_1 U_1, \theta = b_2 U_2, \Gamma = b_3 U_3$</td>
</tr>
<tr>
<td>Metrispherical</td>
<td>$r = b_1 \text{met}(U_1, m_1), \theta = b_2 \text{met}(U_2, m_2), \Gamma = b_3 \text{met}(U_3, m_3)$</td>
</tr>
</tbody>
</table>
Each of the entities $b_1$, $b_2$ and $b_3$ is a coefficient which is referred to as a "basifactor". There are two types of basifactors. Firstly those that are associated with linear coordinates $x$, $y$, $z$ and $r$ and are referred to as "linear basifactors". A linear basifactor should be given in terms of a unit of length. The second type of basifactors are referred to as "angular basifactors", these are associated with angular coordinates $\Theta$ and $\Gamma$. An angular basifactor is given in terms of a unit of angle. For instance, consider a formex

$$A = [2,1; 1,2; 1,3; 2,4; 2,3; 3,3; 3,4; 4,3; 4,2; 3,1; 3,2; 2,2]$$

the plot of $A$ as shown in Fig 2.14(a), illustrates the use of a basibifect retronorm where

$$b_1 = 15 \text{ unit length and } b_2 = 20 \text{ unit length}$$

Also if a basipolar retronorm with

$$b_1 = 8 \text{ unit length and } b_2 = \pi/8$$

is used it would give rise to the normat, a part of which is shown in Fig 2.14(b). Thus one may specify different values for each of the basifactors for various retronorms.

The second family of standard retronorms are referred to as "pariant retronorms". These are special cases of basiant retronorms, because they are obtained by allowing every linear basifactor to be equal to one unit length. There are six different types of pariant retronorms namely, pariunifect, paribifect, paritrifect, paripolar, paricylindrical and parispherical.
Fig 2.14(a) A part of a basibifect normat

Fig 2.14(b) A part of a basipolar normat

Fig 2.15 A metriant retronorm
The third family of standard retronorms are referred to as "metriant retronorms" as shown in Table 2.1. The terms $b_1$, $b_2$ and $b_3$ appearing in the table are the basifactors similar to those described for the basiant retronorms. The terms $m_1$, $m_2$ and $m_3$ which are non zero positive numbers are referred to as "metrifactors". The definitions of metriant retronorms involve a particular scalar function called the "metril function". The function $\text{met}(U_1,m_1)$ implies acceleration (deceleration) of scaling. To illustrate the use of metriant retronorms, consider the formex

$$A = \{[4;2],[2;1],[1;3],[3;4]\}$$

a plot of which is shown in Fig 2.15 for which the metriunifect retronorm is specified by

$$b_1 = 4 \text{ unit length and } m_1 = 0.5$$

here the intervals between successive normat lines decreases progressively controlled by the value of $m_1$.

Consider the configuration in Fig 2.16(a) which represents a flat grid. A metribifect plot for which

$$b_1 = 1 \text{ and } b_2 = 1$$

and

$$m_1 = 0.9 \text{ and } m_2 = 1.2$$

will give rise to the configuration in Fig 2.16(b). The intervals between successive normat lines increases progressively in the second direction controlled by the value of $m_2$. If $m_1 = 1$ the lengths of the intervals remain the
Fig 2.16(a)

Fig 2.16(b)
The designation for a standard retronorm may also be used in relation to a graphical representation of that retronorm. For instance, the normats of Figs 2.15 and 2.16(b) may be referred to as basiunifect and basibifect normats.

2.8.4 PLOTTING STYLES

In plotting a formex, one may specify a retronorm and then use a set of retrocords by which the tenons and fronds may be drawn. But often it is found to be more convenient to have several groups of retrocords, where each group caters for a plotting style, suitable for a particular application. Three plotting styles have been discussed here, they are referred to as the "radix", "natural" and "Zygmunt" plotting styles, and the plots obtained are referred to as "R-plot", "N-plot" and "Z-plot" respectively. The radix plotting style gives rise to plots that closely reflect the particulars of their respective formices. Thus if

\[ F = \{[2,1;2,2]; 1,2],[2,2; 4,2],[4,2,4,4],[4,4,2,4], [2,4; 2,2],[5,3],[2,2,2,4] \} \]

then the R-plot of E with respect to a two dimensional Cartesian coordinate system using the equations

\[ X = U1 \text{ and } Y = U2 \]

will be as shown in Fig 2.17(a). The orderates of the cantles are also indicated in the plot. A retrocord that specifies an aspect of the radix plotting style is referred to as a "radix retrocord".
Fig 2.17(a)

Fig 2.17(b)
The next type of plotting style is the natural plotting style which has been illustrated in the N-plot of E in Fig 2.17(b). The retrocords used in drawing this plot may be described as follows:

(1) The frond of a cantle of the second plextitude is obtained by drawing a straight line to connect the pivots relating to its signets. For a cantle of the third plextitude, the pivots relating to the first and the third signet are connected by a straight line.

(2) No symbol is used for a tenon, except when the tenon represents a cantle, in which case it is drawn as a little circle.

(3) The order of appearance of the signets in the cantles is not indicated.

(4) A part of the plot that involves overlapping fronds is represented only once.

The above retrocords are the "principal natural retrocords". In producing a plot one needs these basic retrocords and additional retrocords may be specified for other details such as the colour or the thickness of tenons and fronds. Also one may from time to time add to these set of retrocords for special requirements.

The Zygmunt plotting style is used mainly to represent multi-layer configurations. For instance, consider the configuration in Fig 2.18(a), which represents a double layer grid. It consists of two parallel layers of elements that are interconnected by web elements. The plan view of the same structure is shown again in Fig 2.18(b) where the elements in the top layer are drawn in full lines, the lower layer are drawn in dotted lines, and the web elements are shown in broken lines. In this form of representation, it is much easier to
visualize the configuration as compared to the one in Fig 2.18(a). As an alternative technique, the nodes in different layers may be represented in different styles as in Fig 2.18(c). One may even use a combination of both the techniques to draw the configuration shown in Fig 2.18(d). In general, one may define a Z-plot as a formex plot that represents a multi-layer configuration in which different methods of representation are used to identify tenons and/or fronds lying in different layers.

2.8.5 RETROBASES AND PROBASES

We have seen how a formex plot may be obtained by a combination of a retronorm and a collection of retrocords, that is by using a "retrobasis". A retrobasis is a set of rules through which a given formex may be plotted.

If a geometric configuration has to be represented by a formex, a different set of rules, called "probasis", is used. The rules that constitute a probasis are of two types, the first type supply information regarding the correlation between the component parts of the configuration and the signets and cantles of the formex. A rule of this type is referred to as a "procord". The second set of rules provide information about the values of the uniples in the formex and a combination of all the rules of this type is referred to as a "pronorm".

As an example, consider the configuration in Fig 2.19. Suppose that a formex has to be written to represent the configuration and that the procords are specified as follows:

(1) Every one of the numbered triangles in the configuration should be represented as a three-plex cantle.
A graphical representation of the translation function.

Fig 2.19

Fig 2.20(a)

A graphical representation of the translation function.

Fig 2.20(b) The general form of the translation function

function designator

function

$H = \text{tran}(h, q) \mid G$

imprint

direction of translation

argument

amount of translation

rallus symbol
(2) The cantles must appear in the order indicated by the numbers written in their corresponding triangles.

(3) Each corner of a triangle should be represented by a signet.

(4) The order of appearance of the signets in the cantles should be as indicated for triangles 1 and 3.

Also, the pronorm is specified graphically in Fig 2.19. The two families of dotted lines provide information about the correspondence between the corners of the triangles and the uniples of the required formex. So one may write the required formex as

\[
G = \{[3,3; 4,4; 5,3], [2,2; 3,3; 4,2], ...
\]

\[
[4,2; 3,3; 5,3], [4,2; 5,3; 6,2], ...
\]

\[
[3,1; 2,2; 4,2], [3,1; 4,2; 5,1], [5,1; 4,2; 6,2]\}
\]

The concept of probasis is the converse of the concept of retrobasis. Similarly, the concept of procord is the converse of the concept of retrocord and the concept of pronorm is the converse of the concept of retronorm.
2.9 FORMEX FUNCTIONS

2.9.1 INTRODUCTION

We are familiar with a relation such as

\[ y = \sin x \]

in scalar algebra, which may be used to evaluate \( y \) for any given \( x \). It is customary to refer to \( x \) and \( y \) as independent and dependent variables, respectively. In the above example, the term \( \sin \) is referred to as a function and symbolises the rule through which \( y \) is obtained from \( x \).

In a similar manner, formices can play the roles of dependent and independent variables. If a rule is established by which from a given formex, say \( E \), another formex \( G \) is obtained, then this rule may be represented by a symbol say \( \phi \), and the relation between \( E \) and \( G \) may be represented as

\[ G = \phi | E \]

The symbol \( | \) is referred to as the "rallus symbol" and is read as "rallus" or "of".

It often happens that the need for a particular way of processing a formex arises repeatedly. Under such circumstances it is convenient to standardise the process by turning it into a function. A function that gives rise to a formex is referred to as a formex function.
2.9.2 CARDINAL FUNCTIONS

The first group of standard functions are referred to as cardinal functions. There are eight cardinal functions. One can fully appreciate the nature of formex functions by considering a few examples.

The first cardinal function is the "translation function". To illustrate the use of a translation function, consider the formices

\[ G = \{[3,3; 2,3],[2,3; 2,1],[2,1; 1,1]\} \]

and

\[ H = \{[6,3; 5,3],[5,3; 5,1],[5,1; 4,1]\} \]

Graphical representations of these formices are shown in Fig 2.20(a). The convention of using a bar over a formex to denote its plot has been introduced here and will be used henceforth. It may be noticed that the plot of H is obtained by translating the plot of G in the first direction by three units. The relationship between the formices G and H may be expressed by the statement

\[ H = \text{tran}(1,3)\mid G \]

The construct tran(1,3) is a formex function representing a rule for transformation of formex G into formex H, where tran is an imprint for translation. The value given by 1 denotes the direction of translation while the value given by 3 denotes the pace or amount of translation. The constituents of the above relation are shown in a more general form in Fig 2.20(b). Parameters such as h and q are referred to as "canonic parameters" and are parts of the rule defining the particulars of the transformation. As another example of the translation function, consider the formex
\[ A = [1,1; 1,2; 3,2; 3,1; 2,2] \]

a plot of which is shown in thick lines in Fig 2.21. Then

\[ A_1 = \text{tran}(1,2) | A \]

and

\[ A_2 = \text{tran}(2,1) | (A \# A_1) \]

will give rise to the formex variables \( A_1 \) and \( A_2 \) whose plots are represented in Fig 2.21.

The next cardinal function is the "rindle function". The term rindle is an old English word meaning a watercourse. In formex algebra, the rindle function performs translational replication as illustrated below. Let the formex

\[ B = [3,1; 1,3; 3,5; 5,3; 3,3] \]

represent the element shown in bold lines in Fig 2.22. Then

\[ B_1 = \text{rin}(1,5,4) | B \]

and

\[ B_2 = \text{rin}(2,4,4) | B_1 \]

would represent the entire pattern shown in Fig 2.22. In obtaining \( B_1 \), the direction of replication is denoted by the first canonic parameter 1, the spread or number of replications is denoted by the second canonic parameter 5 and the third canonic parameter 4, denotes the pace or amount of translation at each step.
Fig 2.21

Fig 2.22 An illustration of the rindle function
The various cardinal functions are translation, rindle, reflection, lambda, vertition, rosette, projection and dilatation. These have been discussed in Table 2.2 with the help of some simple examples. In Table 2.2, the values denoted by $h$, $h_1$, $h_2$, and $s$ can only be integer entities. The values of $q$, $p$, $q_1$ and $q_2$ may either be integer or non-integer real entities.

2.9.3 TENDIAL FUNCTIONS

Very often Cardinal functions are used in combinations and can therefore be considered as functions in their own rights. These are called "tendial functions". Depending on the directions in which they operate, tendial functions are further categorised into four sub groups. Each tendial function is equivalent to a cardinal function acting in one direction proceeded by a cardinal function of the same type acting in the second direction. For example, a construct such as

$$\text{tran}(2,q_2)\mid \text{tran}(1,q_1)$$

is referred to as a nested or composite function. The same can be expressed as a tendial function

$$\text{tranid}(q_1,q_2)$$

where the suffix "id" stands for actions in the first and second directions, $q_1$ and $q_2$ denote the spread or number of translations in the first and second directions, respectively.

Similarly, the construct
<table>
<thead>
<tr>
<th>NAME</th>
<th>FUNCTION</th>
<th>EXAMPLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation</td>
<td>$\text{tran}(h,q)$</td>
<td>$A = {[3,4;2,4],[2,4;2,2],[2,2;1,2]}$</td>
</tr>
<tr>
<td></td>
<td>$h =$ direction of translation</td>
<td>$A_1 = \text{tran}(1,2)</td>
</tr>
<tr>
<td></td>
<td>$q =$ amount of translation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rindle</td>
<td>$\text{rin}(h,s,p)$</td>
<td>$A = {[1,1;3,3],[1,1;3,3],[1,1;3,3]}$</td>
</tr>
<tr>
<td></td>
<td>$h =$ direction of replication</td>
<td>$A_1 = \text{rin}(1,4,3)</td>
</tr>
<tr>
<td></td>
<td>$s =$ number of replications</td>
<td>$A_2 = \text{rin}(2,3,3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reflection</td>
<td>$\text{ref}(h,q)$</td>
<td>$B = {[1,2;2,2],[2,1;1,1],[1,2;3,3]}$</td>
</tr>
<tr>
<td></td>
<td>$h =$ direction of reflection</td>
<td>$B_1 = \text{ref}(1,3)</td>
</tr>
<tr>
<td></td>
<td>$q =$ position of the plane of reflection</td>
<td>$B_2 = \text{ref}(2,4)</td>
</tr>
<tr>
<td>Lambda</td>
<td>$\text{lam}(h,q)$</td>
<td>$B = {[1,1;1,4],[2,3;1,2],[1,2;2,1]}$</td>
</tr>
<tr>
<td></td>
<td>$h =$ direction of reflection</td>
<td>$B_1 = \text{lam}(1,3)</td>
</tr>
<tr>
<td></td>
<td>$q =$ position of the plane of reflection</td>
<td>$B_2 = \text{lam}(1,3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 2.2 (continued ...)

<table>
<thead>
<tr>
<th>Type</th>
<th>Formula/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertition</td>
<td>$\text{ver}(h_1,h_2,q_1,q_2)$&lt;br&gt;h$_1$ and h$_2$ define the plane of rotation, while q$_1$ and q$_2$ are the coordinates of the point of rotation</td>
</tr>
<tr>
<td>Rosette</td>
<td>$\text{ros}(h_1,h_2,q_1,q_2)$&lt;br&gt;h$_1$ and h$_2$ define the plane of rotation, while q$_1$ and q$_2$ are the coordinates of the point of rotation</td>
</tr>
<tr>
<td>Projection</td>
<td>$\text{proj}(h,q)$&lt;br&gt;h=direction of projection&lt;br&gt;q=position of the plane of projection</td>
</tr>
<tr>
<td>Dilatation</td>
<td>$\text{dil}(h,q)$&lt;br&gt;h=direction of dilatation&lt;br&gt;q=factor of dilatation</td>
</tr>
</tbody>
</table>
G2 = rinid(6,4,2,2)|G1

creates a formex whose plot is the whole pattern shown in Fig 2.23. The plot of G1 consists of four line segments shown in thick lines and can be expressed as

G1 = lam(1,2)|{[2,1; 1,2],[1,2; 2,4]}

Tendial functions are not defined for vertition and rosette functions. There are twenty four tendial functions divided into four groups as shown below.

(1) Tendid functions: The suffix "id" implies actions in the 1st and 2nd directions. There are six tendid functions namely, tranid, rinid, refid, lamid, projid and dilid.

(2) Tendis functions: The suffix "is" implies actions in the 1st and 3rd directions. There are six tendis functions namely tranis, rinis, refis, lamis, projis and dilis.

(3) Tendit functions: The suffix "it" implies actions in the 2nd and 3rd directions. There are six tendit functions namely tranit, rinit, refit, lamit, projit and dilit.

(4) Tendix functions: The suffix "ix" implies actions in the 1st, 2nd and 3rd directions. There are six tendix functions namely tranix, rinix, refix, lamix, projix and dilix.
Fig 2.23 An illustration of the tendial rindle function

Fig 2.24 An illustration of the use of a provial function
2.9.4 PROVIAL FUNCTIONS

The formex functions discussed so far belong to a family of functions referred to as transflection functions. This family has another group of functions called provial functions. Each provial function relates to two or more directions. To illustrate the use of a provial function, consider the plot in Fig 2.24. The plot of the formex

\[ M = \text{rosad}(2,2)|[1,1; 3,1] \]

gives rise to the square plot shown in thick lines. The relation

\[ M_1 = \text{rinad}(1,1,3,3,5)|M \]

will give rise to the five squares shown in the figure. The construct \( \text{rinad}(1,1,3,3,5) \) is a provial function, where the suffix "ad" indicates actions in directions one and two. The line joining the coordinates (1,1) and (3,3) indicates the direction of replication and is referred to as the direction vector. The last canonic parameter 5 denotes the spread or the number of replications.

There are eight provial functions which relate to directions 1 and 2. They are referred to as "proviad functions". Also there are eight provial functions relating to directions 1, 2 and 3. These are referred to as proviax functions where the suffix "ax" is added to the abbreviation of the cardinal function.

Proviad and proviax functions have been discussed in Table 2.3. The canonic variables \( A_1, A_2, A_3, B_1, B_2 \) and \( B_3 \) are the coordinates of the direction vector \( AB \) as shown in Figs 2.25 and 2.26. The canonic parameter \( s \) that indicates the spread is always an integer while \( p, q \) and \( \alpha \) can be integer or
<table>
<thead>
<tr>
<th>FUNCTIONS</th>
<th>DESCRIPTION OF CANONIC PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provad $\text{Tranad}(A_1,A_2,B_1,B_2[,q])$</td>
<td>Direction of translation is given by vector $AB$ and the amount of translation is given by the value of $q$ (or the length of $AB$ in the absence of $q$)</td>
</tr>
<tr>
<td>Provix $\text{Tranax}(A_1,A_2,A_3,B_1,B_2,B_3[,q])$</td>
<td></td>
</tr>
<tr>
<td>$\text{Rinad}(A_1,A_2,B_1,B_2,s[,p])$</td>
<td>Direction of replication is given by vector $AB$, number of replications is given by the value of $s$ and amount of translation at each step is given by the value of $p$ (or the length of $AB$ in the absence of $p$)</td>
</tr>
<tr>
<td>Provix $\text{Rinax}(A_1,A_2,A_3,B_1,B_2,B_3,s[,p])$</td>
<td></td>
</tr>
<tr>
<td>$\text{Refad}(A_1,A_2,B_1,B_2)$</td>
<td>Direction of reflection is given by vector $AB$, with the plane of reflection being normal to $AB$ at $B$</td>
</tr>
<tr>
<td>Provix $\text{Refax}(A_1,A_2,A_3,B_1,B_2,B_3)$</td>
<td></td>
</tr>
<tr>
<td>$\text{Lamad}(A_1,A_2,B_1,B_2)$</td>
<td>Direction of reflection is given by vector $AB$, with the plane of reflection being normal to $AB$ at $B$</td>
</tr>
<tr>
<td>Provix $\text{Lamax}(A_1,A_2,A_3,B_1,B_2,B_3)$</td>
<td></td>
</tr>
<tr>
<td>$\text{Verad}(A_1,A_2[,\alpha])$</td>
<td>$A_1$ and $A_2$ are the coordinates of the centre of rotation is the angle of rotation in degrees. If $\alpha$ is not given it implies $\alpha = 90^\circ$</td>
</tr>
<tr>
<td>Provix $\text{Verax}(A_1,A_2,A_3,B_1,B_2,B_3[,\alpha])$</td>
<td>Axis of rotation is given by vector $AB$ and amount of rotation is given by $\alpha$ in degrees</td>
</tr>
<tr>
<td>$\text{Rosad}(A_1,A_2[,s,p])$</td>
<td>The number of replications is denoted by the value of $s$ and amount of rotation at each step is given by the value of $p$ in degrees. If $s$ and $p$ are not specified it implies $s=4$ and $p=90^\circ$</td>
</tr>
<tr>
<td>Provix $\text{Rosox}(A_1,A_2,A_3,B_1,B_2,B_3[,s,p])$</td>
<td></td>
</tr>
<tr>
<td>$\text{Projad}(A_1,A_2,B_1,B_2)$</td>
<td>Direction of projection is given by vector $AB$, with the plane of projection being normal to $AB$ at $B$</td>
</tr>
<tr>
<td>Provix $\text{Projax}(A_1,A_2,A_3,B_1,B_2,B_3)$</td>
<td></td>
</tr>
<tr>
<td>$\text{Dilad}(A_1,A_2,B_1,B_2,q)$</td>
<td>Direction of dilatation is given by vector $AB$ and the factor of dilatation is given by the value of $q$</td>
</tr>
<tr>
<td>Provix $\text{Dilax}(A_1,A_2,A_3,B_1,B_2,B_3,q)$</td>
<td></td>
</tr>
</tbody>
</table>
Fig 2.25

Fig 2.26
non-integer real entities. In Table 2.3, some of the canonic parameters are enclosed in special brackets called optional brackets. For example, the canonic parameter $p$ for the rindle functions represents the pace. There is an option of either specifying the pace directly or choosing the direction vector such that its length determines the pace. For example, in Fig 2.24, the pace was determined by the length of the direction vector $AB$.

Another example which illustrates the use of provial functions is the formex representing the pattern shown in Fig 2.27. The lowermost part of the figure may be represented as

$$F = \text{rosad}(15,3,4,90) | \text{lamad}(14,0,13,1) | [13,1; 13,2; 14,3]$$

The direction of reflection is given by vector $AB$ and the plane of reflection is normal to $AB$ at $B$. In the above equation, the first two parameters specify the point of rotation which is $(15,3)$. The next two parameters specify the number of replications and the angle of rotation at each step of replication. The whole configuration can be created by

$$F_1 = \text{rinad}(13,1,17,5,4) | F$$

and

$$F_2 = \text{rinad}(17,1,13,5,4) | F_1$$

For the plots of $F_1$ and $F_2$, the direction vector specifies the pace of the translation and the last canonic variable represents the spread.

Provial functions can be divided into four major groups. The first two groups namely the proviad and proviax functions have been discussed. The remaining two groups are referred to as "provias" and "proviat" functions. Provias
Fig 2.27 An illustration of the use of a provial function
functions relate to directions 1 and 3 and are used by adding the suffix "as" to the abbreviation of the cardinal function. Proviaat functions relate to directions 2 and 3 and are used by adding the suffix "at" to the abbreviation of the cardinal function.

2.9.5 INTROFLECTION FUNCTIONS

The next major group of formex functions belongs to the family of introflection functions. Just as formices for different configurations can be written using the concepts of formex composition and transflection functions, they can be clipped or curtailed using introflection functions. Introflection functions perform "editing" operations in various ways. They are particularly useful when irregular patterns are to be formulated. Introflection functions fall under three groups namely the "pexum function", "rendition functions" and "rejection functions".

(a) PEXUM FUNCTION

The first introflection function to be considered is the pexum function. The following example illustrates the use of the pexum function. Let the configuration in Fig 2.28 be formulated as

\[ A = \text{rinid}(4,2,4,4) | \text{lamil}(3,3) | \{[3,1; 1,2], [1,2; 1,3], \ldots \} \]

This manner of describing the configuration poses a problem because the elements parallel to the U1 direction are doubly represented as shown in Fig 2.29. This could have been avoided by using a different formex formulation. However, by the relation
Fig 2.28

Fig 2.29
one can easily get rid of the superfluous elements. The imprint pex is an abbreviation for the pexum function. Pexum is a Latin word meaning combed and is used in formex algebra in the sense of combing out extra cantles. The effect of the pexum function on a formex is to eliminate any cantle which is a variant of a preceding cantle. Sometimes it is easier to formulate a configuration such that there are a few extra cantles in the initial stages of the formulation and later dispose of the extra cantles by using the pexum function.

A formex formulation in which extra cantles are generated for convenience, is referred to as a "surplete formulation". In contrast a formex formulation which does not allow the generation of superfluous cantles is referred to as a "perplete formulation".

As another example of the use of the pexum function, consider the barrel vault shown in Fig 2.30. All the joints of the vault lie on the surface of a right circular cylinder with radius equal to 10 units of length. A formex formulation for the interconnection pattern relative to the indicated pronorm may be given by formex A2, where

\[
A = \begin{bmatrix} 10,1,1; & 10,2,2 \end{bmatrix} \# \text{ros}(2,3,1.5,1.5) \begin{bmatrix} 10,1,1; & 10,2,1 \end{bmatrix}
\]

\[
A1 = \text{lamit}(5,2) \begin{bmatrix} \text{rin}(2,4,1) \end{bmatrix} A
\]

and

\[
A2 = \text{pex} \begin{bmatrix} \text{rin}(3,4,2) \end{bmatrix} A1
\]

Here the superfluous cantles generated in the formulation were eliminated by using the pexum function.
Fig 2.30

Plot of E

Plot of F

Fig 2.31
(b) RENDITION FUNCTIONS

The next group of Introflection functions to be discussed belong to the family of "rendition functions". There are six rendition functions each of which represents a rule for curtailing a formex in a particular manner. Rendition functions have been discussed with the help of examples in Table 2.4. Formices E and F referred to in the table are given by the statements

\[
E_1 = \text{lamad}(2,0,1,1) | \{[1,1; 1,2],[1,2; 2,3],[2,3; 3,3]\}
\]

\[
E_2 = \text{pex} | \text{rinid}(2,2,4,4) | \text{rosad}(3,3) | E_1
\]

\[
E_3 = \text{rosad}(5,5) | [4,3; 6,3]
\]

\[E = E_2 \# E_3\]

and

\[
F = \text{lamad}(6,0,3,3) | \text{rinad}(5,2,7,4,2) | \{[4,3; 6,3],[5,4; 5,2]\}
\]

The plots of E and F are represented in Fig 2.31.

The term "luxum" implies disconnected parts, the term "nexum" implies connected parts while the term "pactum" implies coincident parts. The prefix co is used to denote "the complement of".

(c) RELECTION FUNCTIONS

The third group of introflection functions is referred to as relection functions. Theoretically, relection functions are of a more general nature than the other introflection functions discussed so far.

Formices are inspected in terms of a condition set down by a rule and curtailed accordingly. This rule can be specified as required for a particular application.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUXUM LUX(F)</td>
<td>Plot of lux(F)</td>
<td>E is obtained by removing every element of E that has one or more nodes connected to the nodal points of F.</td>
</tr>
<tr>
<td>COLUXUM COL(F)</td>
<td>Plot of col(F)</td>
<td>E is obtained by removing every element of E that is a part of the plot of lux(F)</td>
</tr>
<tr>
<td>NEXUM NEX(F)</td>
<td>E</td>
<td>Plot of nex(F)</td>
</tr>
<tr>
<td>CONEXUM CON(F)</td>
<td>E</td>
<td>Plot of con(F)</td>
</tr>
<tr>
<td>PACTUM PAC(F)</td>
<td>Plot of pac(F)</td>
<td>E is obtained by removing every element of E all whose nodes are not coincident with all the nodes of an element of F.</td>
</tr>
<tr>
<td>COPACTUM COP(F)</td>
<td>Plot of cop(F)</td>
<td>E is obtained by removing every element of E that is a part of the plot of pac(F)</td>
</tr>
</tbody>
</table>
For example, consider a formex $E$ and let there be a condition, denoted by $P$, such that every cantle of $E$ either satisfies or does not satisfy $P$, which means that $P$ with respect to a cantle of $E$ is either true or false. Let a formex $G$ be obtained from $E$ by inspecting the cantles of $E$, proceeding in the natural order, and deleting every cantle for which the condition specified by $P$ is not satisfied. Then the rule by which $E$ is transformed into $G$ is symbolised in terms of a function. This function is denoted by

$$rel(P)$$

and is referred to as a "relection function". The formex $G$ is referred to as a relection of $E$ with respect to $P$ and the relation between $E$, $P$ and $G$ is written as

$$G = rel(P)|E$$

Any relection of an empty formex is considered to be the empty formex itself.

As an example, consider the formex

$$F = \{[5;3],[4;2],[1;7],[5;9],[8;3],[9;4]\}$$

and let $P$ be defined as

$P$ is true provided that the first signet is greater than the second signet and $P$ is false otherwise.

The relection of $F$ with respect to $P$ is found to be
\[ \text{rel}(P) | F = \{[5;3],[4;2],[8;3],[9;4]\} \]

Any formex can be curtailed by employing a set of rules and that is why relection functions are said to be the most general of all introflection functions. It is possible to write every one of the previously described introflections in terms of a relection function. For example, if

\[ G = \text{pex} | E \]

then this may equivalently be written as

\[ G = \text{rel}(P) | E \]

where \( P \) with respect to a cantle \( C \) of \( E \) may be defined as

\[ P \text{ is true provided that there is no cantle of } E \text{ that has an orderate lower than } C \text{ and is a variant of } C \text{ and } P \text{ is false otherwise.} \]

A condition of this type used in conjunction with relection functions is referred to as a "perdicant". A perdicant can be described using a natural language or a combination of mathematical formulae and statements in a natural language. In some cases perdicants may be written in a convenient notation which is discussed below.

BREVIC NOTATION: In writing simple types of commonly used perdicants, there is a shorthand notation that may be conveniently used. The notation is referred to as the "brevic notation". The symbols that constitute the notation together with their meanings are described in Table 2.5. In the table the symbols \( M_a \) and \( M_b \) denote two maniples which may or may not be of the same
plextitude and of the same grade.

Table 2.5

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU</td>
<td>every uniple of $M_a$</td>
</tr>
<tr>
<td>AU</td>
<td>any uniple of $M_a$</td>
</tr>
<tr>
<td>EU(j)</td>
<td>$j$ th uniple of every signet of $M_a$</td>
</tr>
<tr>
<td>AU(j)</td>
<td>$j$ th uniple of any signet of $M_a$</td>
</tr>
<tr>
<td>U(i,j)</td>
<td>$j$ th uniple of the $i$ th signet of $M_a$</td>
</tr>
<tr>
<td>EW</td>
<td>every uniple of $M_b$</td>
</tr>
<tr>
<td>AW</td>
<td>any uniple of $M_b$</td>
</tr>
<tr>
<td>EW(j)</td>
<td>$j$ th uniple of every signet of $M_b$</td>
</tr>
<tr>
<td>AW(j)</td>
<td>$j$ th uniple of any signet of $M_b$</td>
</tr>
<tr>
<td>W(i,j)</td>
<td>$j$ th uniple of the $i$ th signet of $M_b$</td>
</tr>
</tbody>
</table>

To illustrate the use of the brevic notation, consider the maniple

$$M = [4,5,9; 2,2,6; 4,7,8]$$

The perdicant

$$U(2,1) + 2 < U(1,3)$$

is true with respect to $M$ since
\[ U(2,1) + 2 = 4 \] and \[ U(1,3) = 9. \]

The perdicant

\[ \text{EU}(2) > 4 \]

is false with respect to \( M \), since the second uniple of the second signet does not satisfy the condition.

Finally, the perdicant

\[ (\text{AU} > 10) \text{ OR } (\text{EU} < 1) \]

is false with respect to \( M \), since it is neither true that any uniple of \( M \) is greater than 10 nor is it true that every uniple of \( M \) is less than 1.

As an example of the practical application of the brevic notation consider the configuration in Fig 2.32 which represents a part plan of a flat grid. To write a formex formulation for the interconnection pattern in terms of the pronorm shown one can begin with a surplete formulation such as

\[ F_1 = \text{rosad}(2.5,2.5) | \{[[1,1; 1,2],..} \]

\[ [1,1; 2,1],[2,1; 1,2],[2,1; 3,1]\]

and

\[ F_2 = \text{pex} | \text{rinid}(5,3,3,3) | F_1 \]

The interconnection pattern represented by \( F_2 \) as shown in Fig 2.33, contains a lot of superfluous fronds. These can now be deleted by writing
Fig 2.32

Fig 2.33

Fig 2.34
\[ F_3 = \text{rel}(P) \mid F_2 \]

where \( P \) is given by

\[ 4 \leq EU(2) \leq 7 \quad \text{AND} \quad 13 \leq EU(1) \leq 4 \]

and

\[ 7 \leq EU_2 \leq 10 \quad \text{AND} \quad 7 \geq EU_1 \geq 10 \]

### 2.10 LIBRA COMPOSITION

In formex algebra, serial summations are used quite often. For example, if

\[ F = F_1 \# F_2 \# F_3 \ldots \# F_{n-1} \# F_n \]

one may represent this summation as

\[ F = \text{lib}(i=0,n) \mid Fi \]

The imprint "lib" is used for the libra symbol.

As an example of the use of the libra composition, consider the configuration in Fig 2.34. Let it be required to write a formex \( H \) representing the elements. Thus one may write a formulation for the first element say \( F \) where

\[ F = \text{pex} \mid \text{lam}(1,2) \mid \{[1,1; 2,2],\ldots \]

\[ [2,2; 2,4],[2,4; 1,5] \}

and

\[ H = F \# E \]
Fig 2.35

Fig 3.36
where $E$ is the second element. This can also be expressed as

$$H = F \# \text{tran}(1,2)|F$$

or

$$H = \text{lib}(i=0,1)|\text{tran}(1,2i)|F$$

In the above equation $\text{lib}(i=0,1)|\text{tran}(1,2i)|F$ is equal to

$$(\text{tran}(1,0)|F) \# (\text{tran}(1,2)|F)$$

because translation of $F$ by two units in the first direction is $F$ itself.

A libra composition may also be employed to represent a combination of elements shown in Fig 2.35. These may be represented by

$$H_1 = \text{lib}(i=0,6)|\text{tran}(1,2i)|F$$

Similarly, the configuration in Fig 2.36 may be represented by

$$H_2 = \text{lib}(j=0,2)|\text{tran}(2,4j)|H_1$$

If the same configuration represents a flat grid with supports as shown in Fig 2.37, a formex listing the support positions in terms of the indicated normat may be written as

$$E = \text{lam}(2,7)|\text{lib}(i=0,5)|\text{tran}(1,2i)|\{[3,5],[2,6]\}$$

As another example, consider the configuration in Fig 2.38 which may be represented by
Fig 2.37

Fig 2.38
\[ E_1 = \{(6,0; 7,1), (7,1; 5,1), (5,1; 6,0)\} \]
\[ E_2 = \text{lib}(i=7,12) | \text{tranid}(7-i, i-7) | \text{rin}(1,i,2) | E_1 \]
\[ E_3 = E_2 \# \text{rin}(1,6,2) | (6,0; 8,0) \]
\[ E_4 = \text{pex} | \text{lam}(2,6) | E_3 \]
\[ N = \{(8,2), (16,2), (12,10)\} \# \text{lamid}(12,6) | \{(11,5), (10,6), (11,7)\} \]

and
\[ G = \text{lux}(N) | E_4 \]

the above example brings out the power of a libra composition where a complex formulation can be conveniently expressed in terms of a few equations.
2.11 FORMIAN

The concepts of formex algebra discussed so far have a close relationship with Formian which is in fact the programming language of formex algebra. Formian is a suitable computing software that can solve problems of computer graphics and data generation in an effective and elegant way.

The ideas of Formian originated in the late seventies and various versions of the language have been used since. Formex algebra is the mathematical basis of Formian. A complete description of Formian is given in Ref 24.

Information regarding various aspects of a structural system can not only be generated using Formian, but the data can be stored in the form of a "rule". Thus a few lines of formulation can represent complete information about a structural system like the interconnection pattern of elements, support positions and loading particulars. This information can be conveniently stored and readily modified whenever required.

Fig 2.39 shows a sketch of the manner in which a computing set up operates using Formian. For instance, as shown in the figure, the formex formulation for a structure, is typed in through the keyboard and a graphical output is obtained on the VDU (visual display unit). This information may either be modified by changing the formulation, or used directly for obtaining a plot of the structure. The same information may then be submitted to a structural analysis package for postprocessing. The component parts of Formian that are most commonly used have been described below.
FORMIAN SET UP

keyboard → cpu → SUBMIT → analysis package

Fig 2.39

Fig 2.40(a)
"Characters" are the main building blocks of the constructs of Formian. A character is a digit or a letter or a symbol or a layout character. A digit is any one of the ten decimal digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. A letter is any one of the fifty two uppercase and lowercase letters of the English alphabet. A symbol is any printable mark that can be represented by the computing system and is not a digit or a letter. For example, +, -, *, %, ?, and > are symbols. A layout character is either a space character or a newline character.

"Keywords" are any one of the following twelve sequences of letters

KEEP TAKE PRINT GIVE DRAW SHOW

SUBMIT ERASE USE RECALL EXIT STOP

A keyword can be printed in uppercase or lowercase letters. Thus KEEP, Keep and keep will have the same effect when used as a keyword. Keywords prompt various activities which will be discussed shortly.

An "Identifier" is a sequence of letters and digits but the first character of an identifier should be a letter. An identifier may not be the same as a keyword and may not have more than twelve characters. Examples of identifiers are

K MESH h21 vz467

Variables and functions can be assigned names using identifiers. Here again both uppercase and lowercase letters can be used in specifying identifiers and this means that
2.11.2 CONSTANTS

A constant is a sequence of characters which defines the value of an abstract entity. Formian has five types of constants. These represent integer numbers, noninteger real numbers, integer formices, noninteger formices and character strings.

An "integer constant" can be defined as a sequence of one or more digits which may be preceded by a plus or minus sign. An integer constant may not contain an intermediate layout character. For example, +84 and 1219 are acceptable layout characters but 22.08 and +4.2 are not.

A "floatal constant" can be described in three different ways, that is as

1. An integer constant followed by a decimal point followed by a sequence of one or more digits. For example, 92.0, +64,555 and 7.126 are examples of floatal constants.

2. A floatal constant of the first form followed by the letter E followed by an integer constant. For example, +985.4E3 and 643.02E-1 are acceptable floatal constants.
(3) An integer constant followed by the letter E followed by an integer constant, as in -56E72 and +32E214.

A floatal constant may not contain any intermediate layout character. A floatal constant of the form mEn is considered to represent m*10^n. Thus the floatal constants 2.1, 0.21E1, 2100E3 and 2.1E+00 have the same value.

A "string constant" is a sequence of characters enclosed between quote symbols. A string constant is of the form 'D' where ' is the quote symbol and D is a sequence of characters. If D contains any intermediate quote symbols, then these appear in one or more batches and every such batch consists of an even number of consecutive quote symbols. For example,

'""Untouchability is a sin against God and Man", said Gandhi.'

is a valid string constant.

The value of a string constant is the sequence of characters which are enclosed between the initial and terminal quote symbols. For example the value of the above string constant is

'Untouchability is a sin against God and Man', said Gandhi.

Layout characters in a string constant retain their normal typographical significance.

Among the set of all string constants there is one whose value is a null character sequence. This special string constant is denoted by "" and is referred to as the empty string constant.
2.11.3 ASSIGNMENT STATEMENTS AND VARIABLES

A "variable" is an identifier which has been assigned a value. Values are assigned to identifiers by "assignment statements". There are five types of variables, namely, integer variables, floatal variables, integer formex variables, floatal formex variables and string variables.

The general form of an assignment statement is as follows:

\[ \text{identifier} = \text{expression} \]

In this context the symbol = is referred to as the assignment symbol. To the right of the assignment symbol is an expression which is a structured sequence of constants, variables and operators. As an example, consider the assignment statement

\[ K = 270 \]

This effectively creates a variable K representing the integer number 270. If there already exists a variable K, then the old value of K will be discarded and the new value will be recorded. If the above statement is followed by the assignment statement

\[ K = -546.22 \]

then the value of K which was the integer variable 270 will be replaced by the floatal variable -546.22.
2.11.4 INFORMATION TRANSFER STATEMENTS

A computing system on which Formian is implemented is assumed to include a terminal, a working memory, a repository and some input-output channels.

Formian instructions are inputted from a device called the "terminal" through which system messages and other items of information are outputted. The terminal has an associated graphics output medium which may be a part of the screen of the terminal or a separate graphics screen.

During a Formian session, values of variables are stored on a temporary basis in the "working memory". The working memory is empty at the beginning of each Formian session. During the session variables are created, processed and their values are stored in the working memory. However at the end of the session the working memory is wiped out completely and the variables are irrecoverably lost.

If information has to be stored on a permanent basis, it has to be transferred to the "repository". The storage capacity of the repository is normally much greater than the working memory. Before one can perform any operations on the values of variables stored in the repository, they have to be transferred to the working memory.

A printing channel indicates a particular way of textual output on a printing medium. A printing channel can be specified by a number. There may be more than one printing channels but at a given time only one of them is current.

Information can be easily transferred from the repository to the working
memory as the need may be. There are certain Formian statements, namely, KEEP, TAKE, PRINT, GIVE, DRAW, SHOW, and SUBMIT which are referred to as information transfer statements. These have been discussed below.

KEEP STATEMENT: Variables once created can be conveniently stored and used repeatedly. This can be achieved by means of the KEEP statement which is a construct of the form

\[ \text{KEEP V1, V2, ..., Vn} \]

where keep is a keyword and V1, V2, ..., Vn are variables. The execution of the above statement will cause a copy of the values of V1, V2, ..., Vn to be stored in the repository. These will become covariables which means that they will have associated identifiers representing a value in the repository. This does not affect their values in the working memory. However, if at the moment of execution of the KEEP statement there was an existing covariable, say, V2 representing a value in the repository then the old value of V2 will be discarded and replaced by the new value.

TAKE STATEMENT: Variables stored in the repository can be retrieved through the TAKE statement. A TAKE statement is a construct of the form

\[ \text{TAKE C1, C2, ..., Cn} \]

where TAKE is a keyword and C1, C2, ..., Cn are covariables. This effectively places a copy of the values of C1, C2, ..., Cn in the working memory and associates these values with the identifiers C1, C2, ..., Cn. This will not affect the covariables in the repository, but if at that time, say, C2 was
a current variable in the working memory, its previous value will be irrecoverably lost.

PRINT STATEMENT: The PRINT statement is used to print the values of the specified variables on the medium indicated by the current printing channel. A PRINT statement is a construct of the form,

\[
\text{PRINT } V_1, V_2, \ldots, V_n
\]

where PRINT is a keyword and \(V_1, V_2, \ldots, V_n\) are variables whose values will be printed on the medium relating to the current printing channel. Also each of the entities \(V_1, V_2, \ldots, V_n\) may be the symbol \(>\) which in the present context is referred to as the trude symbol. When the statement is being executed, each trude symbol will have the effect of a pagethrow (newpage). The comma between a trude symbol and a variable may be omitted.

For example, consider the following statements

\[
T = '\text{Membrane cutting pattern}'
\]
\[
V = \text{rosad}(3,3) \begin{bmatrix} 1,1 ; 1,5 \end{bmatrix}
\]
\[
\text{PRINT } >T,V
\]

The execution of these statements results in the following text to be printed at the top of a newpage on the medium indicated by the current printing channel:

\[
\text{Membrane cutting pattern}
\]
\[
\{[1,1;1,5], [1,5;5,5], [5,5;5,1], [5,1;1,1]\}
\]

GIVE STATEMENT: The effect of a GIVE statement is the same as that of
a PRINT statement except that the output will appear on the screen of the terminal. The GIVE statement is a construct of the form

\[
\text{GIVE } V_1, V_2, \ldots, V_n
\]

where GIVE is a keyword and \( V_1, V_2, \ldots, V_n \) are as described for the PRINT statement.

**DRAW STATEMENT**: The effect of a DRAW statement is to cause graphical representation of formices together with textual material to appear on the medium indicated by the current graphical output channel. A DRAW statement is a construct of the form

\[
\text{DRAW } H_1, H_2, \ldots, H_n
\]

where DRAW is a keyword each of the entities \( H_1, H_2, \ldots, H_n \) is either a formex variable or a string variable. Also, similar to the case of the PRINT statement, each of the entities \( H_1, H_2, \ldots, H_n \) may be a trude symbol with the comma between the variable and the trude symbol being optional. In the case of a DRAW statement the effect of a trude symbol depends on the type of output medium selected. For example,

(a) on a graphics screen, the effect of a trude symbol is to clear the screen,

(b) on a device that issues sheets of paper, the effect of a trude symbol is to issue a sheet and

(c) on a device whose output medium is a roll of paper, the effect of a trude symbol is to shift the plotting area by a certain length along the roll.
SHOW STATEMENT: The effect of a show statement is the same as the DRAW statement except that the output will appear on the graphics medium associated with the terminal. A SHOW statement is a construct of the form

\[
\text{SHOW } V_1, V_2, \ldots, V_n
\]

where SHOW is a keyword and the entities \( V_1, V_2, \ldots, V_n \) are variables as described for the DRAW statement. The construct

\[
\text{SHOW } >
\]

will have the effect of clearing the graphics medium associated with the terminal. When the only item following the keyword SHOW is a trude symbol, then the keyword may be omitted and the symbol > appearing as a statement will have the same effect as the statement SHOW >.

SUBMIT STATEMENT: The role of a SUBMIT statement is to transform formices and other entities into files that may be used as input data for various application programs and packages. A SUBMIT statement is a construct of the form

\[
\text{SUBMIT } P_1, P_2, \ldots, P_n
\]

where SUBMIT is a keyword and each of the entities \( P_1, P_2, \ldots, P_n \) is a data structure called a plenix.
2.11.5 ORGANISATIONAL STATEMENTS

There are five organisational statements in Formian which are used for various purposes as discussed below. These are ERASE, USE, RECALL, EXIT and STOP statements.

ERASE STATEMENT: The variables and covariables together with their values can be erased from the working memory and repository by using an ERASE statement. An ERASE statement is a construct of the form

\[
\text{ERASE } F_1, F_2, (F_3), \ldots, F_n
\]

where ERASE is a keyword and where each of the entities \( F_1, F_2, (F_3), \ldots, F_n \) are either variables or covariables enclosed in parantheses.

USE STATEMENT: A USE statement is a construct of the form

\[
\text{USE } A_1, A_2, \ldots, A_n
\]

where USE is a keyword and each of the entities \( A_1, A_2, \ldots, A_n \) is referred to as a USE-item. There are a variety of USE-items employed for a variety of specifications like specifying current input-output channels. For example the statement

\[
\text{USE CH(9)}
\]

may have the effect of specifying a pen plotter as the current graphical output channel where CH stands for channel.
RECALL STATEMENT: A RECALL statement is either of the form

RECALL V

or of the form

RECALL

where RECALL is a keyword and V is a string variable. The first statement would display an assignment statement on the screen of the terminal with the left-hand side of the statement being the identifier V and the right-hand side being the string constant whose value is specified by V. This assignment statement becomes the current statement which can be edited and saved. The effect of the RECALL statement of the second type is to display the statement which has just been executed. This statement can then be edited and re-entered.

EXIT STATEMENT: An EXIT statement is entered as

EXIT

where EXIT is a keyword and this effectively causes the Formian session to pause temporarily and one can go to the operating system of the computer. One can always return to the Formian environment and continue to work without finding any changes.

STOP STATEMENT: A STOP statement is of the form

STOP
where STOP is a keyword. This effectively causes the termination of the Formian session, the working memory is cleared and the user is returned to the operating system of the computer.
2.12 FORMIAN GRAPHICS

Once formex formulations are written, the images of their plots can be obtained on a graphical output media using Formian. Consider a formex plot that consists of nine line segments as shown in Fig 2.40(a). It is difficult to visualise the correct shape of the three dimensional plot unless it is explained by a perspective drawing or by making an actual physical model. Using the concepts of Formian it is possible to create two-dimensional plots and these have been described below.

A plot which is to be viewed is referred to as the "object" and the coordinate system relative to which the plot was produced is referred to as the "object coordinate system". The space in which the plot is situated is called the "object space".

The point from where the object is to be viewed is called the "view point" and is specified by a USE-item of the form

\[ \text{USE VP}(x,y,z) \]

where VP stands for view point and x, y and z are numeric expressions whose values are the coordinates of the view point relative to the object coordinate system.

The point which is directly viewed is referred to as the "view centre" and is specified by a USE-item of the form

\[ \text{USE VC}(x,y,z) \]
where VC stands for view centre and where x, y and z are numeric expressions whose values are the coordinates of the view centre relative to the object coordinate system. The line that joins the view point and view centre is referred to as the "view line" as shown in Fig 2.40(b).

It is assumed that there exists a plane which is normal to the view line at the view centre. This plane is referred to as the "trace plane". Also it is assumed that there exists a family of lines referred to as "view rays" where each view ray passes through the view point and a point of the object. When the view rays intersect the trace plane they create an "image" of the object as shown in Fig 2.40(b). This image is referred to as the "trace of the object". The type of projection used to obtain the trace is referred to as perspective projection which results in a perspective view of the object. In the case of parallel projection, it is assumed that the object is viewed through an infinitely large eye whose mid-point is at the view point and that all the view rays are parallel to the view line as shown in Fig 2.41. The required type of view may be specified by a USE-item of the form

\[
\text{USE VT(n)}
\]

where VT stands for "view type" and n is an integer expression whose value is either 1, which specifies a parallel view or 2 which specifies a perspective view.

The next step is to produce an image of the trace on the output medium of a device such as a VDU (visual display unit) or a plotter. This image is referred to as the picture of the object. Also the plane in which the picture lies is called the "picture plane" and the coordinate system of the output device is called the "device coordinate system", Fig 2.42. One may specify a rectangular frame in the picture plane restricting the region for graphic production to the area
Fig 2.40(b)

Fig 2.41
Fig 2.42
enclosed within the frame. This frame is referred to as the "view frame" and may be specified by a USE-item of the form

\[
\text{USE VF}(p_1, q_1, p_2, q_2)
\]

where VF stands for view frame and \( p_1, q_1, p_2 \) and \( q_2 \) are numeric expressions whose values are the coordinates of the corners \( A_1 \) and \( A_2 \) of the view frame relative to the device coordinate system. The view frame and the device coordinate system are not graphically produced and are therefore shown in dotted lines in Fig 2.42. This convention of using dotted lines to represent entities that are not graphically produced has been used henceforth.

When the picture of the object is to be produced from its trace, it is necessary to have information about

1. the required orientation of the picture,
2. the required position of the picture and
3. the required size of the picture

The required orientation of the picture can be determined using a vector which is referred to as the "view rise". This is defined relative to the object coordinate system. The orientation of the picture is chosen such that the image of the view rise in the picture plane is parallel to the q-axis, as shown in Fig 2.43. The view rise is specified by a USE-item of the form

\[
\text{USE VR}(x_1, y_1, z_1, x_2, y_2, z_2)
\]

where VR stands for view rise and \( x_1, y_1, z_1, x_2, y_2 \) and \( z_2 \) are numeric expressions whose values are the coordinates of the starting point of the view
rise and the end point of the view rise, respectively relative to the object coordinate system.

If the starting point of the view rise is coincident with the view centre, it is possible to specify the view point, the view centre and the view rise at the same time. This is achieved by using the concept of "view helm" which is defined as a broken vector consisting of the view rise and the view line, Fig 2.44. The view helm is specified by a USE-item of the form

\[
\text{USE VH}(x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3)
\]

where VH stands for view helm and \(x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\) are numeric expressions whose values are the coordinates relative to the object coordinate system, of the view point, the view centre which is also the starting point of the view rise and the end point of the view rise.

The position and size of the picture may be controlled using three different approaches. Firstly, one may specify a point in the object space referred to as the "view base", and a point in the picture plane, referred to as the "view nave". The picture is positioned such that the image of the view base in the picture plane will coincide with the view nave as shown in Fig 2.45.

The view base may be specified by a use item of the form

\[
\text{USE VB}(x, y, z)
\]

where VB stands for view base and \(x, y, z\) are numeric expressions whose values are the coordinates of the view base relative to the object coordinate system. The view nave may be specified by a USE-item of the form
Fig 2.45
USE VN(p,q)

where VN stands for view nave and where p and q are numeric expressions whose values are the coordinates of the view nave relative to the device coordinate system. This method of positioning the picture is used in conjunction with the concept of "view scale" or "view gauge" to control the size of the picture. The view scale may be specified by a USE-item of the form

USE VS(r)

where VS stands for view scale and r is a numeric expression whose value controls the size of the picture. It is assumed that there is a circle, referred to as the "unit circle", in the trace plane whose centre is at the view centre and whose radius is a unit length. The scale for the picture is chosen such that the radius of the image of the unit circle in the picture plane is r millimetres, as shown in Fig 2.46.

The view gauge is specified by a USE-item of the form

USE VG(x1,y1,z1,x2,y2,z2,r)

where VG stands for view gauge and where x1, y1, z1, x2, y2 and z2 are numeric expressions whose values are the coordinates relative to the object coordinate system of a line segment in the object space which is referred to as the view gauge, Fig 2.47. The value of r, which is a numeric expression controls the size of the picture. The scaling of the trace for producing the picture is chosen such that the length of the image of the view gauge in the picture plane is r millimetres, Fig 2.47.
Fig 2.47
When the concepts of view nave, view base and view scale are used to control the position and size of the picture, this mode is referred to as the "nave mode." An alternative way of controlling the position and size of the picture automatically is to use the "range mode". This is achieved by considering a rectangle in the trace plane whose sides are either parallel or perpendicular to the image of the view rise in the trace plane. This rectangle is referred to as the "trace range", as illustrated in Fig 2.48 and Fig 2.49.

The third method of controlling the size and position of the picture is referred to as the "zone mode". This involves the creation of a "view zone", which is demarcated by a rectangular solid, in the object space with its facets being either parallel or perpendicular to the object coordinate axes, Fig 2.50. This method allows zooming effects as well. The zone mode operates by choosing the position and size of the picture such that the image in the view zone will be produced as the picture, Fig 2.50. The view zone is specified by a USE-item of the form

\[
\text{USE VZ}(x_1, y_1, z_1, x_2, y_2, z_2)
\]

where VZ stands for view zone and \(x_1, y_1, z_1, x_2, y_2\) and \(z_2\) are numeric expressions whose values are the coordinates relative to the object coordinate system, of two diagonally opposite vertices of the view zone.

The mode of picture control may be specified by a USE-item of the form

\[
\text{USE VM}(n)
\]

where VM stands for view mode and \(n\) is an integer expression whose value is 1, 2 or 3, specifying nave mode, range mode or zone mode, respectively.
Fig 2.49
What the human eye can see is only what falls within the field of vision. A similar setting is simulated by assuming that the view point is the vertex of a conical volume which is referred to as the "view field". The view line is the axis of the view field and the angle between the lines of intersection of the view field and a plane that contains the view line is referred to as the "view angle", Fig 2.51. The view angle is chosen to be $\pi$ radians for parallel projection and 3 radians for perspective projection. Only those parts of the object which are within the view field are considered for picture creation.

Each one of the USE-items described above with the exception of view gauge has a default setting. At the beginning of every Formian session the default settings become effective unless changed through a USE statement. The default settings are so selected that they represent the most commonly used settings and these are listed in Table 2.6.

In the nave mode, the size of the picture is determined through the concept of view scale. Thus, unless a view gauge is specified by a USE statement, the picture will be created using the view scale and this is why view gauge does not have a default setting.

If default settings are changed during a Formian session, they can be restored by specifying the actual settings. So one may enter

$$\text{USE VT}(1), \text{VP}(0,0,10000), \text{VR}(0,0,0,0,1,0)$$

Alternatively, one may enter the statement

$$\text{USE &VT, &VP, &VR}$$
The ampersand symbol implies default settings. Also, when the ampersand symbol is used by itself it implies the collection of all default settings. For instance, the statement

```
USE &, VM(3), VZ(10,10,20,0,1,0)
```

restores all the default settings and then changes the view mode and sets the view zone as specified.

**Table 2.6**

<table>
<thead>
<tr>
<th>VIEW SPECIFIER</th>
<th>DEFAULT SETTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>view base</td>
<td>VB(0,0,0)</td>
</tr>
<tr>
<td>view centre</td>
<td>VC(0,0,0)</td>
</tr>
<tr>
<td>view frame</td>
<td>VF(10,10,250,250)</td>
</tr>
<tr>
<td>view mode</td>
<td>VM(1)</td>
</tr>
<tr>
<td>view nave</td>
<td>VN(10,10)</td>
</tr>
<tr>
<td>view point</td>
<td>VP(0,0,10000)</td>
</tr>
<tr>
<td>view rise</td>
<td>VR(0,0,0,0,1,0)</td>
</tr>
<tr>
<td>view scale</td>
<td>VS(10)</td>
</tr>
<tr>
<td>view type</td>
<td>VT(1)</td>
</tr>
<tr>
<td>view zone</td>
<td>VZ(0,0,0,10,10,10)</td>
</tr>
</tbody>
</table>

The term "Formian session" has been used frequently, it actually refers to the sequence of activities that the user performs while working with Formian. The session begins when the user sits in front of a computing system and types
FORMIAN to start with. Then the session progresses as the user creates variables, stores them and finally ends the session by typing STOP and returns to the operating system of the computer.

Now suppose that at the beginning of a Formian session the following statements are entered

\[ A = \text{rinid}(5,5,2,2) | \text{rosad}(2,2) | \{[2,2,0; 1,1,1], \ldots \\
[1,1,1; 3,1,1]\} \]
\[ B = \text{rinid}(4,4,2,2) | \text{rosad}(3,3) | [2,2,0; 4,2,0] \]
\[ C = \text{pex} | (A \# B) \]
\[ \text{DRAW} \ C \]

The first three statements will create the formex variable C which represents a double layer grid. The DRAW statement causes a plan view of the plot of C to be drawn on the current graphics output medium as shown in Fig 2.52. The coordinate axes shown in the figure have been added for convenience only and are not drawn through the DRAW statement in the formulation.

To obtain a perspective view of the plot of C, one may enter the following statements

\[ \text{USE} \ VT(2), \ VM(2) \]
\[ \text{USE} \ VC(6,-6,6), \ VP(6,6,0), \ VR(6,6,0,6,6,1) \]
\[ \text{USE} \ VF(20,20,250,250) \]
\[ \text{DRAW} \ C \]

The view thus obtained will be a front view of the spaceframe from the top, with one side parallel to the viewer, as shown in Fig 2.53. A different aspect
Fig 2.52

Fig 2.53

Fig 2.54
of the plot of C may be obtained by entering the statements

USE VH(-10,-5,-25,9,9,5,9,9,6)
DRAW C

This would result in a view as if one were looking at the spaceframe from below, along one of its diagonals, as shown in Fig 2.54.

2.12.1 FORMIAN RETROCORDS

Another useful feature of Formian is the possibility of specifying properties such as colour of lines and typeface of textual material on graphical output. A choice for a feature of this kind is referred to as a retrocord. There are a number of USE-items that specify the type of tenons and fronds to be drawn. The standard types are specified by default settings but one can obtain a variety of tenon and frond styles as described below.

Line width for tenons and fronds is controlled by a USE-item of the form

USE LW(r)

where LW stands for line width and r is a numeric expression which specifies the line width in millimetres. The default setting for the line width is 0.3 millimetres.

Tenon style is determined by a USE-item of the form

USE TS(n,d)
where TS stands for tenon style, n is an integer expression and d is a numeric expression. The value of n may be from 1 to 10, representing the symbols shown in Table 2.7.

For different Formian installations this list can be freely extended as required. The value of d is a positive number which specifies the dimension of the symbol in millimetres. For values of n from 1 to 5, d specifies the diameter of the circle and for values of n from 6 to 10, d specifies the length of the side of the square. The default setting for the tenon style USE-item is TS(5,0).

Line style for fronds is specified by a USE-item of the form

\[
\text{USE LS}(p)
\]

where LS stands for line style and p is an integer expression. The value of p may range from 1 to 4 representing the patterns shown in Table 2.8.

Again this list can be freely extended to suit various needs. The default settings for the line style USE-item is LS(1).

Frond style is determined by a USE-item of the form

\[
\text{USE FS}(n)
\]

where FS stands for frond style and n is an integer expression whose value is 1, 2 or 3. If the frond represents a cantle with t signets and if the points representing these signets in the picture plane are P1, P2, ..., Px, Py, then different values of n have the following implications:
### TABLE 2.7

<table>
<thead>
<tr>
<th>n</th>
<th>Symbol</th>
<th>n</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>□</td>
<td>6</td>
<td>□</td>
</tr>
<tr>
<td>2</td>
<td>⊙</td>
<td>7</td>
<td>□</td>
</tr>
<tr>
<td>3</td>
<td>□</td>
<td>8</td>
<td>□</td>
</tr>
<tr>
<td>4</td>
<td>□</td>
<td>9</td>
<td>□</td>
</tr>
<tr>
<td>5</td>
<td>Empty circle</td>
<td>10</td>
<td>Empty square</td>
</tr>
</tbody>
</table>

### TABLE 2.8

- **p=1**: Full line
- **p=2**: Dashed line
- **p=3**: Dotted line
- **p=4**: Axis line

---

**Fig 2.55**
If \( n = 1 \), the frond consists of \( t \) tenons and \( t-1 \) line segments \( P_1 \) to \( P_2 \), \( P_2 \) to \( P_3 \), ..., \( P_x \) to \( P_y \).

If \( n = 2 \), frond consists of \( t \) tenons and \( t \) line segments \( P_1 \) to \( P_2 \), \( P_2 \) to \( P_3 \), ..., \( P_x \) to \( P_y \), \( P_y \) to \( P_1 \).

If \( n = 3 \), the structure of the frond is the same as for \( n = 2 \), with any area(s) enclosed by the line segments being infilled. The colour of the infill may be the same or different from those of the line segments and/or tenons, as determined by the colour USE-item.

The value of \( n \) can be specified in various ways to suit different Formian installations. The default setting for the frond style USE-item is FS(2). The line segments do not override any tenons, in particular when a tenon may be represented as an empty circle or an empty square.

The font of textual material on graphical output is determined by a USE-item of the form

\[
\text{USE TF}(t, h, w)
\]

where TF stands for "text font", \( t \) is a nonzero positive integer that specifies the typeface of the characters. The values of \( h \) and \( w \) are nonzero positive numbers that specify the height and width of a character in millimetres, respectively. The default setting for the text font USE-item for any Formian installation is chosen to suit the installation's environment.

The position and orientation of text in the picture plane is determined by a USE-item of the form
USE TG(p, q, α)

where TG stands for "text guide". The values of p and q specify the position of the first character of the text in the picture plane in relation to the device coordinate system, Fig 2.52. The value of α in degrees, determines the angle of inclination of the text with the horizontal, Fig 2.52. The default setting for the text guide USE-item is TG(10, 10, 0).

Colour effects may be specified by a USE-item of the form

USE C(n, h)

where C stands for colour and n and h are integer expressions. The values of n may be from 1 to 5, specifying

\[
\begin{align*}
n &= 1 \quad \text{line} \\
n &= 2 \quad \text{tenon} \\
n &= 3 \quad \text{infill} \\
n &= 4 \quad \text{text} \\
n &= 5 \quad \text{background}
\end{align*}
\]

The value of h is a nonzero positive integer specifying a colour. In addition to hue when \( n = 3 \) or \( n = 5 \), the value of h may specify various styles of half tone and hatching. The colour USE-item has 5 default settings relating to the five values of n. These are chosen to suit individual Formian installations.

Colour and line width cannot be changed when pen plotters are used because the choice of the colour and width of the line will be determined by the pen used in the plotter. Therefore, colour and line width USE-items are not applicable
134

to pen plotters.

Pen selection for a pen plotter is made by a USE-item of the form

\[
\text{USE PEN(n)}
\]

where the value of \( n \) is a nonzero positive integer that accepts one of the pens in the pen holder. The default setting for the pen USE-item is PEN(1).

2.13 SCHEMES AND INDUCTION STATEMENTS

Consider the configuration in Fig 2.53, which represents a double layer grid. A formex formulation for the grid was written in the previous section. Now, suppose that it is required to carry out a series of analysis with different grid sizes and grid densities without changing the general interconnection pattern. One possibility is to write a formulation for each individual case, however, a better idea is to write a general formulation that relates to all cases. Such a formulation is referred to as a "generic formulation". If it is assumed that there are \( N_1 \) divisions in the first direction and \( N_2 \) divisions in the second direction, a generic formulation may be written as

\[
A = \text{rinid}(N_1, N_2, 2, 2) | \text{rosad}(2, 2) | [2, 2, 0; 1, 1, 1], \ldots
\]

\[
[1, 1, 1; 3, 1, 1]}
\]

\[
B = \text{rinid}(N_1 - 1, N_2 - 1, 2, 2) | \text{rosad}(3, 3) | [2, 2, 0; 4, 2, 0] \]
\]

\[
C = \text{pex} | (A \# B)
\]

A convenient way of using this formulation in Formian is to include it in an assignment statement of the form
GRID = ':N1,N2:
A=rinid(N1,N2,2,2)|rosad(2,2)|\{[2,2,0; 1,1,1],...
\[1,1,1; 3,1,1]\}
B=rinid(N1-1,N2-1,2,2)|rosad(3,3)|\[2,2,0; 4,2,0\]
C=pex|(A#B)
DRAW C'

The effect of this statement is to create a string variable GRID whose value is the generic formulation, preceded by a list of parameters enclosed in colons. The statements which lie between the quote symbols constitute a construct which is referred to as a "scheme".

A scheme consists of a heading followed by a body. The heading consists of a list of identifiers that are separated by commas and are enclosed in colons. These are referred to as "nominal parameters". A sequence of Formian statements make up the body.

A string variable whose value is a scheme is referred to as a "scheme variable". Thus the above variable GRID is a scheme variable. Also, an assignment statement through which a scheme variable is made is referred to as a "scheme statement".

The body of the scheme in the above example may be executed through a statement of the form

USE GRID(10,6)

which is referred to as an "induction statement". Here, N1 assumes the value 10 and N2 assumes the value 6. The result of the execution of the scheme is
the grid shown in Fig 2.56. Different values can be assigned to N1 and N2 to obtain different configurations.

The description of the grid can also be given in terms of the actual nodal coordinates instead of normat coordinates. Thus if \([U1,U2]\) is a typical signet of \(C\), then the coordinates of the corresponding node are given by

\[
\begin{align*}
x &= (D1/N1)U1 \\
y &= (D2/N2)U2
\end{align*}
\]

An extended scheme can now be written as

\[
\text{GRID} = ':N1,N2,D1,D2:
A=\text{rinid}(N1,N2,2,2)|\text{rosad}(2,2)|\{[2,2,0;1,1,1],...
\quad[1,1,1;3,1,1]\}
B=\text{rinid}(N1-1,N2-1,2,2)|\text{rosad}(3,3)|[2,2,0;4,2,0]
C=\text{pex}|(A\#B)
\text{USE} \ BB(D1/N1,D2/N2)'
\]

Then, the induction statements

\[
\text{GRID}(4,3,5,7)
\]

and

\[
\text{GRID}(8,9,3,6)
\]

will produce the plots shown in Figs 2.57 and 2.58.

The concept of a scheme allows complex configurations to be described in a brief and accurate manner. It is effective in creating grids having the same
interconnection pattern but varying densities. Also, a scheme may be saved, retrieved and altered as and when required. Some of the advantages of using a scheme for a formex formulation have been listed below.

(a) A scheme may be used to describe a trial shape by using parameters which could be changed easily to examine various possible shapes. The scheme may be kept in the computer memory and used whenever required.

(b) The manner in which the scheme is modified is similar to the way in which a text is edited using a word processor.

(c) The scheme is stored as a character string in the computer and therefore its storage requirement is quite small.
2.14 GENERALISATIONS

A configuration has been described as an arrangement of parts or objects. A configuration may be represented with respect to various properties. For instance, there are those properties of a configuration that relate to the manner in which the components of the configuration relate to each other, there are properties of the configuration that relate to the exact coordinates of the configuration in the global x-y-z coordinate system and there are properties of the configuration that relate to graphical effects and methods of representing various parts of the configuration. These properties of a configuration may be conveniently described with the help of a formex or different formices.

It may be essential in a particular context or application to represent a configuration with respect to any of the above mentioned properties. To elaborate, those particulars of the configuration that determine the manner in which the component parts of the configuration are interrelated are referred to as the "compretic" properties of a configuration. As illustrated earlier, the fundamental role of a formex is to determine the compretic properties of a configuration. The compret of a configuration is specified by listing all the component parts of the configuration and describing the manner in which the components are related to each other. The collection of all compretic properties of a configuration is referred to as the "compret" of the configuration. A thorough and technical definition of a compret may be given as follows:

The compret of a configuration is defined as the arrangement of the component parts of the configuration. More specifically, the compret of a configuration is the arrangement of the tenons and fronds of the configuration. The compret of a configuration is a "property" of the configuration. Moreover, it is a subjective property, in that it depends
on the manner in which the configuration is interpreted, with the manner of interpretation being implied through a procorda. A group of or several procords are referred to as procorda. If the compret of a configuration is represented by a formex relative to a pronorm, then any variant of any sequation of this formex also represents the compret of the configuration relative to the same pronorm.

The above definition is purely mathematical. However, it may not always be necessary to define the compret in such a precise manner. It is often only essential to know and appreciate the gist of the concept without memorising the technical definition.

Secondly, those particulars of a configuration that relate to information about the actual set of reference points in the configuration are referred to as the "normic" properties of the configuration. For instance, the nodal coordinates are the normic properties of a configuration. Deforming and/or displacing a configuration results in a change to nodal points thereby changing the normic properties of the configuration.

Thirdly, there are those properties that relate to the method of representation adopted for different parts of a configuration. These relate to graphical effects such as the thickness of lines, colour of components and line styles used. These particulars are referred to as the "cordic" properties of a configuration.

Different applications may require specific properties of a configuration to be used. In certain situations where the compret of the configuration is the focus of attention, importance may only be given to the compretic properties while the normic and cordic properties may not receive any consideration. In other cases where graphical effects of a configuration are important, normic properties of the configuration may receive little or no attention.
It is interesting to note that by keeping the compretic properties of a configuration intact, the normic properties may be changed. This is because the normic properties of a configuration could vary according to many factors. In a structural context for instance, take the example of grid configurations. In prefabricated spacegrids, even if a certain pattern is chosen for the grid, the module dimensions depend on factors such as the type of space structure system used, the span, loading conditions, type of cladding, architectural appearance and erection procedure. Therefore, the compretic properties of the configuration may be specified, and then the normic properties may change as often as required to suit a set of design requirements.

The concepts of formex algebra provide a suitable means for configuration processing activities. There are a number of different processes involved in generating configurations. To elaborate, every activity in configuration processing involves a sequence of one or more of the following basic processes:

(1) the process through which configurations with regular patterns are generated,
(2) the process through which a configuration is modified by removing parts from it or adding parts to it,
(3) the process through which a configuration is deformed or displaced and
(4) the process through which the components of a configuration are ordered in a manner to comply with specified requirements.

These basic processes are recurring themes in all configuration processing activities and have been briefly discussed in the sequel.

The first basic process consists of all the operations that are performed to formulate one or more "regular" configurations which provide a suitable basis for creation of a "target configuration". This process is referred to as the
process of "propagation". During the propagation stage, the overall configuration is considered as having a regular pattern and a compret is defined. Any "irregularities" in the configuration are often ignored to begin with, these are then dealt with subsequently. In this stage, the focus of attention is on the compretic aspects of the configuration without worrying about the normic properties.

The term "target configuration" as used to imply the configuration required to be created. The terms "regular" and "irregular" are subjective and relative. It is not really possible to define regularity or irregularity in a general sense. However, inspite of the difficulty in providing precise definitions for regularity and irregularity, there is a large measure of consensus on what is regular and what is irregular in practical situations. It should also be mentioned at this point, that in the context of the propagation process, regularity or irregularity is only considered from a compretic point of view rather than a general geometric point of view, Ref 25.

There are no strict rules about which formex functions have to be used during different stages while processing configurations. However, in the propagation stage, where configurations are generated, transflection functions such as cardinal, tendial and provial functions which facilitate propagation of formices are predominantly used.

The second basic process consists of all the operations that are performed to modify the propagated regular configuration(s). This may be achieved by removing and/or adding parts to the configuration to take care of any irregularities. This process is referred to as the "modification" process. This stage involves all the processes that are brought about by transflection functions such as the pexum function (where overlapped or superfluous elements are removed) and the rendition functions in order to curtail formices. Like the
propagation stage, the modification stage is also concerned with compretic properties.

The third basic process consists of all the operations that are necessary for generation of information about the normic properties of the target configuration. This process is referred to as the "adaptation" process. The adaptation process involves displacements and/or retronormic transformations of configurations. The process of adaptation is particularly used when configurations which are combinations of different types of surfaces are to be generated. The formulations for different surfaces are often done separately and functions that facilitate the displacement of surfaces are used to move them around in the global x-y-z coordinate system and then fit them together.

The fourth basic process is referred to as the process of "organisation". This process involves all the operations that are performed to "order" specified parts of a configuration to satisfy required arrangement patterns.

Every configuration processing activity is bound to involve one or more of the above mentioned basic processes. However, the order in which these processes follow each other is not rigid. Also, a configuration processing activity may consist of cycles involving repeated applications of the basic processes in various combinations.
CHAPTER THREE

PATTERN GENERATION

3.1 INTRODUCTION

In dealing with a configuration processing problem for a space structure, it is often required to start by investigating various possibilities for a suitable configuration. This is because a major aspect in the initial stages of the design relates to the arrangement of members that constitute the structure. A variety of configurations have been used for the construction of large span space structures such as stadiums, gymnasiums, industrial buildings, shopping centres, multipurpose halls and aircraft hangars. The richness of the architectural effects, has been expressed in several single, double and multilayer configurations, displaying simple but elegant patterns. However, there still remains a need for a general strategy to deal with the problem of pattern generation for space structures. Since there is a wide range of possible patterns for space structure configurations, it has become increasingly important to have a methodology for perceiving, generating and manipulating patterns to assist architects and engineers in their exploration for new configurations.

In the present Chapter a number of space structure configurations have been studied from the point of view of generating patterns and at the same time trying to trace their origin and family background.
3.2 CONCEPT OF PROTOMORPHS

To begin with, consider the set of patterns shown in thick lines in Fig 3.1. All these patterns have been derived from a basic triangular pattern, shown in the background in thin lines. The method of presentation adopted here is such that it makes it easy to visualise how a variety of patterns may be obtained from a single, simple pattern. The triangular pattern shown in thin lines is a basis for all the other patterns shown in thick lines. The objective of the above example is to bring out the importance of a common underlying pattern which can serve as a starting point for pattern generation. Such a basic pattern is referred to as a "protomorph". The term protomorph implies "basic pattern" and is derived from two Greek words "protos" meaning "first" and "morph" meaning "form".

The significance of protomorphs is comparable to that of colour in our everyday experience of the environment. Just as the concept of colour occurs in almost all aspects of human activities, the concept of protomorphs is relevant to any discipline where pattern generation is required. Although protomorphs may be used in various applications where patterns play a significant role, in the present context, the field is narrowed down, where the use of protomorphs is examined in relation to space structure configurations.
Fig 3.1 Transformations of a triangular pattern
3.3 PROTOMORPHS FOR SINGLE LAYER CONFIGURATIONS

Let it be required to generate single layer space structure configurations. Experience has shown that the patterns of Figs 3.2 to 3.8 are the bases of configurations that are commonly used in single layer grids, domes and barrel vaults. Thus in the context of these types of structures, the patterns shown in Figs 3.2 to 3.8 may serve as protomorphs. The terms "binate", "ternate" and "quadrinate" as shown in Figs 3.2 to 3.8, are used to imply two-way, three-way and four-way patterns. The term "obnate" is used to imply a diagonal pattern.

Conceptually, protomorphs are patterns of infinite extent. To draw attention to this fact, patterns in Figs 3.2 to 3.8 are shown as continuing patterns. Also, the basic triangular pattern in Fig 3.1, which may be regarded as a ternate protomorph, is shown without a boundary. Protomorphs as abstract entities do not have any size or boundary associated with them. However, for all practical applications it is only natural that protomorphs are seen in relation to surfaces or actual physical objects where they acquire definite shapes and boundaries.

Consider the configurations in Figs 3.9 to 3.14. The single layer grid in Fig 3.9, the vault in Fig 3.10, the network dome in Fig 3.11, the ribbed dome in Fig 3.12, the saddle shaped membrane in Fig 3.13, and the folded plate structure in Fig 3.14 are all based on the binate pattern. The barrel vault configuration of Fig 3.10, is shown along with a normat which will be used to formulate the configuration in the sequel. In the context of structural configurations, such as the ones shown in Figs 3.9 to 3.14, protomorphs may represent patterns involving cable, bar or beam elements or finite elements for modelling of plate, shell or membrane structures. For example, in the configuration of Fig 3.9, the planar single layer grid pattern is based on the binate protomorph where the pattern may represent interconnected line elements. In Figs 3.10 to 3.13, the binate pattern is used in relation to a
PROTOMORPHS FOR SINGLE LAYER CONFIGURATIONS

Binate Protomorph
Fig 3.2

Obnate Protomorph
Fig 3.3

Ternate Protomorph
Fig 3.4

Ternate Protomorph
Fig 3.5

Ternate Protomorph
Fig 3.6

Quarter Protomorph
Fig 3.7

Quarter Protomorph
Fig 3.8
Configurations based on the binate protomorph

Flat Single layer Grid
Fig 3.9

Barrel Vault
Fig 3.10

Network Dome
Fig 3.11

Ribbed Dome
Fig 3.12

Saddle shaped Membrane
Fig 3.13

Folded Plate Structure
Fig 3.14
number of curved surfaces. In the case of the vault, the domes and the saddle shaped membrane, the binate pattern may represent commonly used skeletal patterns. If the same configurations were required to be designed as shell structures or plate structures, the patterns may be interpreted as finite element meshes for modelling of the structures.

In the case of the saddle shaped membrane in Fig 3.13, the binate pattern may represent the cutting pattern for the membrane or it may even represent a cable net system which supports the membrane panels. However, the manner of graphical presentation adopted here is not enough to give an idea as to what the patterns represent. Let us assume for the time being that the barrel vault configuration consists of interconnected line elements. Then the formex formulation of the configuration, in terms of the indicated normat in Fig 3.10, may be written as

\[
E_1 = \text{PEX} | \text{RINIT}(12,18,2,2) | \text{ROSAT}(1,1) | [1,0,0; 1,2,0] \\
E = \text{BC}(10,5,0.75) | E_1
\]

In the above formulation, the radius of the vault is taken as 10 units. Formex E1 represents the compret of the configuration in that it gives a complete description of the constitution of the configuration. The formex [1,0,0;1,2,0] represents a single line element which is initially formulated and then repeated to constitute the binate configuration. The pexum function is used to get rid of overlapping elements which arise due to replications. Formex E represents the precise x-y-z coordinates of the barrel vault. The term "binate configuration" is used to mean a configuration based on a binate protomorph. The manner of writing the formex formulation for the vault is not described in any further detail here because this has been explained earlier in section 2.8.3. If the same barrel vault was required to be formulated as a plate structure, then the formex formulation may be written as
In the above formulation, the formex \([1,0,0;1,2,0;1,2,2;1,0,2]\) represents a finite element which is repeated throughout the configuration to constitute the binate pattern on the barrel vault. The graphical effects obtained as a result of both the above formulations would be identical but the formex formulation itself would depend on whether the binate configuration represents line elements or tile elements.

The binate configuration of the ribbed dome in Fig 3.12, is formulated using the basispherical retronorm. Figs 3.15a, 3.15b and 3.15c show three intermediate stages illustrating the manner in which the binate pattern curves to create the dome configuration of Fig 3.12. The two ends of the pattern "close in" as indicated by the arrows, to create the complete dome. Along the line where the pattern closes in, there are overlapping elements as shown in Fig 3.15c. The curved edges along which the pattern starts and at which the pattern ends are further enlarged in Fig 3.16. Also, part of the normat which is used to formulate the compret of the dome is shown in dashed lines in Fig 3.16. The formex formulation of the dome with respect to the normat in Fig 3.16 is given as

\[
\text{F} = \text{PEX} | \text{RINIT}(22,6,2,2) | \text{ROSAT}(1,5) | [1,0,4;1,0,6]
\]

\[
\text{F} = \text{BS}(10,360/44,50/12) | \text{F} \text{I}
\]

In the above formulation, the radius of the dome is 10 units. The formex \([1,0,4;1,0,6]\) represents a single line element which is repeated throughout to create the binate pattern. The problem of overlapping elements, which would occur due to the nature of the formulation is solved by using the pexum function.
However, another problem is encountered at this stage. Along the edges, where the pattern closes in, there are two sets of nodes instead of one. These nodal points are shown in Fig 3.16. Although this effect may not be seen graphically, it implies that as far as formex F is concerned, there are no connections between the members on the right and the members on the left of the nodal points. In order to signify this fact, these nodal points are shown with a "gap" in Fig 3.16. A solution to this problem is offered by a formex function which has the effect of merging two nodes and eliminating the "gap". This is referred to as the "novation" function which would bring together the specified nodes. A complete description of the novation function is given in Ref 22.

The network dome in Fig 3.11 is obtained by mapping the binate pattern on a spherical surface. Also, the configuration in Fig 3.13, is created by mapping the binate pattern on a saddle shaped surface.

The folded plate structure in Fig 3.14, may be regarded as a combination of six planes with a binate pattern. If the configuration consists of line elements, then along the ridges and valleys where the six planes are connected to form the folded plate structure, lines would overlap as shown in Fig 3.17. However, if a formex formulation is written to represent the folded plate structure, the problem of overlapping elements may be solved using the pexum function. The effect of this function is that extra elements are "combed out" and only a single row of elements would remain along the ridges and valleys. From a purely graphical point of view, the overlapping elements do not pose a problem. However, if the formulation is to serve as data describing the interconnection pattern of the configuration for the purposes of structural analysis, then account must be taken for overlapping elements. If the formulation for the folded plate structure is written such that the pattern represented a finite element mesh, then there would be no question of overlapping elements.
Fig 3.16

Fig 3.17
In the examples of Figs 3.9 to 3.14, single layer configurations were generated using the binate protomorph. Similarly, ternate and quadnate protomorphs can be used for single layer space structures such as grids, barrel vaults and domes as elaborated in the sequel.

Configurations in Figs 3.18 to 3.24 are all based on the obnate protomorph. To begin with, consider the configurations in Figs 3.18 to 3.20 where the obnate protomorph is used for single layer grids. Although the three grids are based on the obnate pattern, there is a difference in the configurations with respect to the arrangement of elements at the corners of each of the grids. To draw attention to this, the corners have been encircled in Figs 3.18 to 3.20. The grids in Figs 3.18 and 3.19, differ with respect to the arrangement of elements at their corners. The grid in Fig 3.20 has a combination of two different types of arrangements for corners of the same grid. Also, the grid in Fig 3.20 has additional boundary elements along all four edges.

The obnate protomorph is used for the barrel vault configuration in Fig 3.21. The arrangement of elements at the corners of the vault is similar to the planar grid shown in Fig 3.19. If additional elements are added along the curved and straight edges of the vault in Fig 3.21, the resulting configuration would appear as shown in Fig 3.22.

Configurations in Figs 3.23 and 3.24 are examples of single layer braced domes with an obnate pattern. In order to generate the dome in Fig 3.23, the obnate pattern is simply curved using the basispherical retronorm. However, for the configuration of Fig 3.24, the dome is based on the obnate pattern and additional elements are added along the meridians, at the base and near the crown of the dome.

A family of single layer configurations based on ternate protomorphs are shown
Single Layer Configurations based on the Obnate Protomorph

Fig 3.18

Fig 3.19

Fig 3.20

Fig 3.21

Fig 3.22

Fig 3.23

Fig 3.24
in Figs 3.25 to 3.30. The ternate protomorphs shown in Figs 3.4 to 3.6 have been used as a bases for these configurations. For example, the single layer flat grid in Fig 3.25, and the barrel vault in Fig 3.26 are based on the ternate protomorph shown in Fig 3.6. The patterns used for the dome and torus configurations of Figs 3.27 and 3.28 are based on the ternate protomorph shown in Fig 3.5. The network dome and the braced dome in Figs 3.29 and 3.30 are based on the ternate pattern shown in Fig 3.4.

Different types of quadnate protomorphs are used as bases for the configurations in Figs 3.31 to 3.36. To begin with, the single layer flat grids shown in Figs 3.31 and 3.32 are based on the quadnate protomorphs of Figs 3.7 and 3.8, respectively.

Both, the barrel vault of Fig 3.33 as well as the braced dome of Fig 3.34, are based on the quadnate protomorph shown in Fig 3.8. The configuration of Fig 3.35 is another example of how the quadnate pattern of Fig 3.8 is mapped on a saddle shaped surface. The quadnate protomorph in Fig 3.7 is used as a basis for the network dome configuration in Fig 3.36.

In the examples discussed so far, configurations were based on fairly simple patterns in order to explain the use of protomorphs. However, in any configuration processing activity, there are many situations where more complex patterns need to be generated. In such situations, some basic patterns need to be established and these may be used as springboards to create new configurations. The role of protomorphs is to help in generating initial configurations which may be modified in various ways to arrive at the required configuration. Sometimes, there may be several stages of transformations of a pattern until a satisfactory solution is evolved. These modifications may involve changes to the initial patterns by adding and/or removing elements from configurations or making combinations of different patterns to create new
Single layer configurations based on ternate protomorphs

Fig 3.25

Fig 3.26

Fig 3.27

Fig 3.28

Fig 3.29

Fig 3.30
Single layer configurations based on quadrate protomorphs

Fig 3.31

Fig 3.32

Fig 3.33

Fig 3.34

Fig 3.35

Fig 3.36
configurations. In the material that follows, these methods of modifying patterns to evolve new ones, have been discussed with the help of a number of examples.

Consider the configurations shown in Figs 3.37 and 3.39. Both these configurations are planar grids based on a binate pattern. Let us begin with the "L" shaped single layer grid configuration in Fig 3.37. For the purpose of discussions, the grid has been reproduced in Fig 3.38 and the corners of the grid have been marked as A, B, C, D, E and F. One way of creating the grid would be to imagine that the rectangle marked by A, B, E and G is the initial binate configuration from which the smaller rectangle marked C, D, F, and G is removed. The members which are imagined to have been removed are shown in dashed lines in Fig 3.38. Therefore, the "L" shape is created by cutting out a part of a binate configuration. Another way of creating the grid would be to imagine it as a combination of two rectangular binate grids marked A, H, F and E and H, B, C and D. This method of creating the grid would involve the combination of two binate configurations. The method of visualisation of course is entirely based on what is convenient to the user.

As another example, consider the "cross shaped" flat grid configuration of Fig 3.39. The same grid has been reproduced in Fig 3.40. The grid may be generated in different ways. One way of creating the configuration is to imagine it as a combination of five square shaped binate grids. For example, the grid marked B, C, G, F may be repeated five times. Another way of creating the grid is to visualise a larger square grid which is marked as A, D, P, M, in Fig 3.40, with four corner parts removed to create the cross shaped configuration. The elements which are imagined to have been removed are shown in dashed lines in Fig 3.40. In addition to these modifications, the grid also has four square openings in the centre of each of the four arms of the cross. These may be obtained by removing the unwanted elements shown in
Fig 3.37

Fig 3.38

Single layer binate configuration

Fig 3.39

Fig 3.40
Referring once again to the grid in Fig 3.9, the binate protomorph was used for the flat grid without any modifications to the basic pattern, while in the grids of Figs 3.37 and 3.39, the pattern was modified to arrive at the target configurations. The changes made were compretic changes involving the addition or removal of elements from the patterns or making combinations of patterns. The above examples illustrate how the binate protomorph may be used as a bases for generating a variety of configurations.

Another method of creating new configurations is by scaling patterns. This has been illustrated with the help of an obnate pattern, part of which is shown in Fig 3.41. A formex formulation may be written to describe the compret of the obnate pattern with respect to the normat in Fig 3.41. A scheme describing the obnate pattern may be written as

\[
\text{NET} = ':M,N,D1,D2:
\]

\[
G1 = \text{RINID}(M,N,2,2) | \text{ROSAD}(1,1) | [0,1; 1,0]
\]

\[
G = \text{BB}(D1/M,D2/N) | G1
\]

\[
\text{DRAW } G'
\]

In the above formulation, formex G1 represents the obnate pattern. The parameters M and N are variables controlling the number of subdivisions of the pattern. This implies that different values may be assigned to the variables to obtain different sizes of the pattern. D1 and D2 refer to the dimensions of the whole pattern. The basibifect retronorm is used for scaling the pattern in two directions. The scheme may be executed using different induction statements. Configurations in Figs 3.42, 3.43 and 3.44 are obtained respectively, after executing the three induction statements given below.
In another instance, if a metribifect retronorm is used in relation to the same formex G1 and a scheme written as

\[
\text{NET} = ': M, N: \\
G1 = \text{RINID}(M, N, 2, 2) | \text{ROSAD}(1, 1) | [0, 1; 1, 0] \\
G = \text{MB}(1, 1, 1.12, 1) | G1 \\
\text{DRAW } G'
\]

then the induction statement

\[
\text{NET}(8, 8)
\]

will result in the configuration shown in Fig 3.45. The effect of the metribifect retronorm is to bring about accelerated or decelerated scaling. All three patterns in Figs 3.42 to 3.45 have the same compretic properties, but the scaling changes their normic properties. These examples illustrate how patterns may be created by distorting or scaling such that their compretic properties remain intact but their normic properties vary.

In the examples described so far, configurations have been modified in several ways. In general, one may say that these modifications have involved adding or removing elements from configurations or cutting and combining parts to configurations. Modifications of these type have involved compretic changes. Secondly, there are those modifications that involve scaling or distorting configurations to generate new ones. These modifications involve normic changes but the compretic properties of the configurations remain intact. The
term "adaptation" has been used to imply normic changes to a configuration. Thirdly, there are those modifications which are a combination of both compretic and normic changes to configurations. So far, the term "modification" has been used, in a very general sense, to imply any of the above mentioned ways of transforming configurations. However, it is found that the term modification is not precise enough to indicate the exact nature of changes made in a configuration. Therefore, at this point, the term "retination" is introduced to imply the process involving modifications to the compret of a configuration. Thus, one may say that a particular configuration is "retinated" to imply that the configuration has been subjected to modifications involving changes to the compret of the configuration. Both the terms, modification and retination have been used throughout the following study and whenever a precise or exact meaning is required to be conveyed, the term retination is used. As a matter of interest, the term retination is derived from the Latin word "rete" meaning "net" and the combining form "natus" meaning "to be born". In addition to the examples discussed earlier, a few more instances of how configurations may be retinated are discussed in the sequel.

Consider the barrel vault configuration in Fig 3.46. This configuration which is based on a binate protomorph is modified to create the curved configuration and further retinated by removing certain elements to create window like openings shown in Fig 3.46.

Another example is the barrel vault in Fig 3.47. A quadnate protomorph is used as a basis for the configuration. Certain elements have been removed from the configuration to create the diamond shaped openings in the vault as shown in Fig 3.47.

Consider the barrel vaults shown in Figs 3.48 and 3.49. Both vaults are based on a ternate pattern. The barrel vault in Fig 3.48 may be described as
consisting of two halves with the dividing line along the longitudinal axis of the vault. A ternate configuration is used for each half of the vault such that together the configurations are mirror images of each other with the mirror line coinciding with the longitudinal axis of the vault. The vault in Fig 3.49 is a slightly more complicated configuration using retinations of the ternate protomorph. Here the vault may be visualised as consisting of four quarters, each with a ternate pattern. The four parts are arranged such that the ternate configurations are mirror images of each other along the longitudinal axis and the shorter axis of the vault. The two vaults in Figs 3.48 and 3.49 which are retinations of the ternate protomorph, are also examples of how combinations of configurations with simple patterns may be used to generate more complex patterns.

If the configurations in Figs 3.26, 3.48 and 3.49 are compared, it is seen how the ternate pattern may be used for a simple configuration as in Fig 3.26 and also used in different combinations to create more complex patterns of Figs 3.48 and 3.49.

Consider the dome configuration in Fig 3.50. This is an example of a configuration based on a quadnate protomorph. The quadnate pattern is modified by removing some of the diagonal elements to create the pattern in Fig 3.50.

The pattern used for the dome configuration of Fig 3.51, may appear rather complicated and one may wonder how to create it in terms of a basic pattern. A closer examination would reveal that the dome configuration may be regarded as a combination of two types of quadnate patterns. Looking back at Figs 3.7 and 3.8, it may be noticed that these two quadnate protomorphs have been used in relation to the two parts of the dome configuration. Therefore, the dome may be regarded as a combination of two types of quadnate patterns.
The examples of Figs 3.46 to 3.51 illustrate how protomorphs may be used as an "alphabet" for pattern generation in any configuration processing activity.

The protomorphs shown in Figs 3.2 to 3.8 were the bases of all the patterns discussed so far. Now, let it be required to generate the configurations shown in Figs 3.52 to 3.54. In this case, the protomorphs in Figs 3.2 to 3.8 may not be convenient starting points. It may be noticed that the family of patterns shown in Figs 3.52 to 3.54 have a common underlying pattern which is the "six-way" pattern shown in Fig 3.55. One may begin by using the pattern in Fig 3.55 as a protomorph and retinate it to obtain all the three patterns in Figs 3.52 to 3.54. The above example illustrates that there are no fixed rules about which patterns are to be used as protomorphs. Any pattern which may act as a convenient starting point in a particular application may be regarded as a protomorph.

To summarise the discussions so far, it may be said that configurations in general are based on a variety of patterns. In the course of this study it has emerged that the vast variety of seemingly different patterns are based on a few primary patterns referred to as protomorphs. In a pattern generation exercise, protomorphs serve as a starting point for creation of initial configurations. In this study the problem of pattern generation is approached by making oneself aware of protomorphs as a basis for creating patterns rather than having a catalogue of existing patterns.

In continuation of our present discussion, which involved single layer patterns, the next section deals with double layer patterns and how protomorphs may be used in relation to double layer configurations.
A "six-way" protomorph

Fig 3.55
3.4 PROTOMORPHS FOR DOUBLE LAYER CONFIGURATIONS

A double layer grid may be described as consisting of members arranged in two layers which are placed parallel to each other and separated by web members. The two layers may or may not have identical patterns. Also, the two layers may be placed exactly one above the other or at an offset with respect to each other.

Double layer grids where two identical parallel layers are placed exactly above each other are commonly referred to as "lattice grids". Other type of double layer grids where the two layers are placed at an offset with respect to each other are referred to as "offset grids". A wide variety of patterns have been used for lattice as well as offset double layer grids. These have been grouped as shown below:

(1) Binate on binate patterns
(2) Obnate on obnate patterns
(3) Binate on obnate patterns
(4) Obnate on binate patterns
(5) Ternate on ternate patterns
(6) Hexagonal on hexagonal patterns
(7) Ternate on hexagonal patterns
(8) Hexagonal on ternate patterns

The term used for describing each grid is based on the combination of patterns used for the top and bottom layers of the grid. Each of these patterns have been discussed in the sequel.
3.4.1 BINATE ON BINATE PATTERNS

The simplest type of grid arrangement is the "binate on binate lattice" shown in Fig 3.56. This is the plan view of the grid. Both top and bottom layers have identical binate patterns and are placed exactly above each other. The web members may be vertical and/or inclined but are not seen in plan. Fig 3.57 is an oblique view of the grid showing an arrangement of vertical web members. However, there are a variety of ways in which the web members may be placed. Some of the commonly used arrangements for web members are shown in Figs 3.58 to 3.61. All the binate on binate lattices in Figs 3.57 to 3.61 have the same plan view but differ with respect to the arrangement of web members. The grid pattern in Fig 3.57 where only vertical web members are used gives rise to what is known as the "Cubic System".

In Figs 3.57 to 3.61 the top layer members are drawn in thick lines and the bottom layer members as well as the web members are drawn in thin lines. This manner of representing a double layer grid has been used throughout this section. The term "binate on binate" used in Fig 3.56 implies the combination of two binate patterns that make up the double layer grid.

Fig 3.62 shows a plan view as well as an oblique view of a "binate on binate offset" pattern. The grid consists of two identical binate layers placed parallel to each other with one layer set at an offset to the other in plan. The web members are inclined as seen in the oblique view of the grid in Fig 3.62.

The binate on binate offset pattern has been used very often for double layer grids and may serve as a protomorph to create a variety of patterns. The configurations in Figs 3.63 to 3.68 are all based on the binate on binate offset pattern. For example, the grid in Fig 3.63, is obtained by retinating the binate on binate offset pattern in such a manner that there are four openings in the
Fig 3.56

A binate on binate lattice grid

Fig 3.57
Fig 3.58
Fig 3.59
Fig 3.60
Fig 3.61
grid. Each opening is obtained by removing a set of four web members and the corresponding four bottom layer members. The same manner of retination is used for the patterns in Figs 3.64 and 3.65 to create five and nine openings, respectively. The pattern in Fig 3.66 has four square openings in the central region of the grid. These are obtained by removing the top layer members and the corresponding web elements.

The method of retination used for the grids in Figs 3.63 to 3.66 involved only the removal of members, however, for the grids in Figs 3.67 and 3.68, the corners are chamfered. Additional diagonal elements are added to the top layer pattern after removing certain elements to create the octagonal boundaries. The patterns in Figs 3.67 and 3.68 have been retinated such that in the central region of the grid there are octagonal and square openings, respectively.

3.4.2 OBNATE ON OBNATE PATTERNS

Consider the grid in Fig 3.69. This pattern is referred to as an "obnate on obnate lattice". Both the top and bottom layers have identical obnate patterns and are placed exactly one above the other. There are vertical as well as inclined web members connecting the top and bottom layers as shown in the oblique view of the grid in Fig 3.70. As mentioned earlier, the arrangement of web members can be varied as shown for the binate on binate lattice grids in Figs 3.56 to 3.60.

Another commonly used pattern for double layer grids is the "obnate on obnate offset" pattern shown in Fig 3.71. Both the top and bottom layers have obnate patterns placed at an offset with respect to each other with interconnecting inclined web members. It may be noticed that the arrangement of elements at the corners of the top layer differs from those of the bottom layer in order to complete the pattern. Fig 3.72 is an oblique view of the same grid.
The obnate on obnate offset pattern may be regarded as a protomorph and used as a basis for the configurations shown in Figs 3.73 to 3.76. The patterns in Figs 3.73, 3.74 and 3.75 are created by removing web elements as well as bottom layer elements to form openings in different regions of the grid. The top layer members are left intact. In Fig 3.76 the pattern is created by eliminating some of the web members and top layer members to obtain four openings in the grid while keeping the bottom layer members intact.

Configurations in Figs 3.61 and 3.71 show some of the frequently used patterns for double layer grids. These patterns are created using either binate or obnate patterns. However, an interesting family of double layer grids can be generated using combinations of binate and obnate patterns for top and bottom layers of the same grid. These configurations will be explored in the next two sections.

3.4.3 BINATE ON OBNATE PATTERNS

Consider the family of configurations shown in Figs 3.77, 3.79 and 3.81. These are all based on the "binate on obnate" pattern. In all the three grids, the top layer consists of a binate pattern and the bottom layer consists of an obnate pattern interconnected by inclined web members. In the grid shown Fig 3.77 the inclined web members are seen along the periphery of the grid and in the view shown in Fig 3.78. The web members in the grids of Figs 3.79 and 3.81 are inclined and placed exactly below the top binate layer and can only be seen in the oblique views shown in Figs 3.80 and 3.82, respectively.
3.4.4 OBNA TE ON BIN ATE PATTERNS

Another family of double layer grids arises out of combinations of binate and obnate patterns when the obnate pattern is used for the top layer and the binate pattern is used for the bottom layer. These grids are shown in Figs 3.83, 3.85 and 3.87. The corresponding oblique views of the three grids are shown in Figs 3.84, 3.86 and 3.88, respectively. This family of double layer grids may be obtained by simply reversing the binate on obnate grids discussed in the previous section.

3.4.5 TERN ATE ON TERN ATE PATTERNS

Consider the configuration in Fig 3.89. This is based on a "ternate on ternate lattice" pattern. Both top and bottom layers have identical ternate patterns with vertical web members which are seen in the view of the grid in Fig 3.90. Although this is the simplest arrangement of web members for the lattice grid, the web members can be arranged in a variety of different ways as discussed earlier.

The ternate on ternate lattice pattern may be used as a protomorph to generate a variety of patterns. For example, the grids in Figs 3.91 and 3.92 are based on the ternate on ternate lattice pattern. Certain elements from the top and bottom layers and the corresponding web elements have been removed to create triangular and hexagonal openings. The pattern in Fig 3.93 is a combination of six sections based on the ternate on ternate lattice pattern and retinated so that each section has a hexagonal opening in it. The six sections fit together to form a larger hexagonal opening in the centre of the grid.

The double layer grid shown in Fig 3.94 is based on a "ternate on ternate offset" pattern. This is also another very commonly used pattern which offers
Ternate on ternate lattice grid
Fig 3.89

Oblique view
Fig 3.90

Retinations of the ternate on ternate lattice grid
Fig 3.91 Fig 3.92

Fig 3.93
a wide variety of possibilities after retination to giving rise to a repertoire of patterns.

The ternate on ternate offset pattern in Fig 3.94 was used as a protomorph for generating configurations for a space-based reflector for one of NASA's projects. It was referred to as the "octahedral-tetrahedral truss" or an "octet" truss, Ref 26. In addition to performance related design requirements such as stiffness and accuracy, mass and on-orbit assembly time, the number of component parts was a major consideration in the design process. Thus there was a strong motivation for reducing the number of components of the reflector structure. Patterns in Figs 3.95 to 3.99 were some of the retinations of the truss configuration of Fig 3.94 considered for the reflector configuration.

The patterns in Figs 3.100 to 3.102 are three more examples of how the ternate on ternate offset pattern may be retinated to create other configurations. In Fig 3.100 the top, bottom and web members have been retinated to create hexagonal openings. In the pattern shown in Fig 3.101 the bottom layer is kept intact while the top and web members are removed in certain regions of the grid. In Fig 3.102, the pattern is obtained by keeping the top and web members intact and only removing certain bottom layer elements.

3.4.6 HEXAGONAL ON HEXAGONAL PATTERNS

A family of double layer grids based on hexagonal patterns are discussed in this section. Consider the configuration shown in Fig 3.103. This is a plan view of a hexagonal lattice grid. Both the top and bottom layers consist of identical hexagonal patterns. Fig 3.104 shows an oblique view of the same grid where the web members and the bottom layer members are also seen. Although Fig 3.104 shows a simple arrangement of vertical web members for the hexagonal lattice grid, a variety of arrangements of web members are possible as
Fig 3.103

Fig 3.104

Fig 3.105

Fig 3.106

oblique view of a part of the grid

Fig 3.107

Fig 3.108

Fig 3.109
discussed in the previous sections for other lattice grids.

In the double layer grid in Fig 3.105 there are two hexagonal layers placed exactly above each other like a lattice grid. In addition to the vertical web members, there are diagonal web members meeting at the centre of each hexagonal subdivision of the grid. Fig 3.105 is an oblique view of the grid.

The grid pattern shown in Fig 3.106 is another example of a hexagonal double layer grid. Fig 3.107 shows an enlarged oblique view of part of the same grid where there are vertical struts placed at the centre of each hexagonal subdivision. Both ends of each strut are connected to the corners of the hexagonal subdivision by web members. Also, the struts are connected to each other by horizontal top and bottom layer members which form a triangular pattern at the base and at the top of the grid. If the horizontal members connecting the top and bottom of each strut are removed, the pattern of the grid becomes similar to the Jeffrey Lindsay type bracing. This pattern has been used for several shell structures designed by Jeffrey Lindsay. Each hexagonal unit is stiffened by a central spreader fixed by means of stainless steel cables to the corner points. The spreader allows the pre-tensioning of the hexagonal module resulting in a light and rigid module.

The double layer grid in Fig 3.108 is obtained by using a smaller hexagonal grid for the bottom layer and inclined web members connecting it to the larger top layer hexagonal grid. The web members lie exactly above the bottom layer members and are not seen in the plan view of the grid.

The double layer grid pattern in Fig 3.109 is another example of a hexagonal on hexagonal grid. The pattern consists of a regular hexagonal pattern for the top layer members and two types of hexagonal subdivisions for the bottom layer pattern.
3.4.7 TERNATE ON HEXAGONAL PATTERNS

Consider the pattern shown in Fig 3.110. This may be referred to as a "ternate on hexagonal" grid. The top layer members form a ternate pattern and the bottom layer members form a hexagonal pattern. An oblique view of a part of the ternate on hexagonal grid is shown in Fig 3.111. The bottom layer may be regarded as consisting of hexagonal pyramid type units connected at their apices as shown in Fig 3.111.

The ternate on hexagonal pattern in Fig 3.110 may be used as a protomorph to create the patterns shown in Figs 3.112 and 3.113. The pattern in Fig 3.112 may be regarded as a retination of the pattern of Fig 3.111 where some of the top layer elements are removed while keeping the bottom layer as well as the web members intact. In the pattern of Fig 3.113 some of the top layer elements along with the corresponding web elements have been removed to create three hexagonal openings in the grid.

The configuration in Fig 3.114 may also be regarded as a "ternate on hexagonal" double layer grid. The pattern may be visualised as a number of pyramid type hexagonal units fitted together such that the bottom layer consists of hexagonal and triangular subdivisions. An oblique view of part of the grid is shown in Fig 3.115. If this pattern is compared to the one in Fig 3.110, it is noticed that the arrangement of the hexagons in the bottom layer varies in both the patterns while the top layer ternate pattern is the same.

The ternate on hexagonal pattern in Fig 3.114 may be used as a protomorph to created the grid pattern shown in Fig 3.116. Three hexagonal openings in the grid have been created by eliminating some of the top layer elements along with the corresponding web elements.
Ternate on hexagonal patterns

Fig 3.110

ternate pattern for top layer
web members
hexagonal pyramid type units

oblique view of part of the grid
Fig 3.111

Fig 3.112

Fig 3.113

ternate pattern for top layer
web members
hexagonal units

oblique view of part of the grid
Fig 3.114

Fig 3.116
3.4.8 HEXAGONAL ON TERNATE PATTERNS

Hexagonal on ternate grids may be obtained by inverting the grids discussed in the previous section. The hexagonal pattern forms the top layer while the bottom layer consists of a ternate pattern.

For example, the configuration in Fig 3.117 is an example of a "hexagonal on ternate" grid. The three configurations shown in Figs 3.118 to 3.120 are based on the hexagonal on ternate pattern shown in Fig 3.117. Both the patterns in Figs 3.118 and 3.119 are obtained by eliminating some of the top layer elements and web elements to create four openings and seven hexagonal openings in the grids, respectively. For the pattern in Fig 3.120 some bottom layer elements have been removed but the web members and top layer members have been left intact.

The double layer grid in Fig 3.121 is also based on another type of "hexagonal on ternate" pattern. It can also be visualised as pyramid type hexagonal units interconnected to each other at their bases to form the top hexagonal layer. The bottom layer elements which connect the apices of the hexagonal units form a ternate pattern. Fig 3.122 shows an oblique view of part of the hexagonal on ternate pattern.

The configurations in Figs 3.123 and 3.124 are examples of grids obtained after retinating the pattern in Fig 3.121. In Fig 3.123 some of the bottom layer elements and web elements have been removed but the top layer elements are kept intact. In Fig 3.124 three hexagonal openings have been created in the grid by removing some bottom layer elements and the corresponding web elements.
Hexagonal on ternate grids

Fig 3.117

Fig 3.118

Fig 3.119

Fig 3.120
Hexagonal on ternate grids

Fig 3.121

Fig 3.122

Fig 3.123

Fig 3.124
3.4.9 SOME ADDITIONAL PATTERNS

The double layer grids discussed so far have mainly been based on well known and conventional patterns. In this section some new double layer grid configurations have been presented.

To begin with, consider the binate on binate pattern shown in Fig 3.125. An oblique view of the grid is shown in Fig 3.126. The grid consists of two binate layers placed at an offset with respect to each other. However, unlike the binate on binate offset grid discussed earlier (shown in Fig 3.61) the top binate layer is offset in one direction only. This gives rise to trusses along one direction as seen in the oblique view of the grid in Fig 3.126.

In the grid shown in Fig 3.127 a binate pattern is used for the top layer and a binate pattern with smaller subdivisions is used for the bottom layer. Web members are located in every other subdivision of the smaller binate grid. This is seen more clearly in the oblique view of the grid in Fig 3.128.

The pattern in Fig 3.129 consists of a bottom layer which has rectangular subdivisions and a top layer with square subdivisions. An oblique view of the grid in shown in Fig 3.130.

All the three grid patterns shown in Figs 3.125, 3.127 and 3.129 may be regarded as binate on binate patterns. The binate on smaller binate grid pattern shown in Fig 3.127 may be used as a protomorph to create the patterns shown in Figs 3.131 and 3.132. In both these patterns certain bottom layer elements and the corresponding web elements have been removed from different regions of the grids to create openings in the patterns.

Also, the binate on binate offset grid pattern of Fig 3.129 may be used as a
Some Additional binate on binate grids

binate on binate offset grid

Fig 3.125

oblique view

Fig 3.126

binate on smaller binate grid

Fig 3.127

oblique view

Fig 3.128

binate on binate offset grid

Fig 3.129

oblique view

Fig 3.130
protomorph to generate the two patterns shown in Figs 3.133 and 3.134. The manner of retination adopted here is such that certain bottom layer elements as well as the corresponding web elements have been removed to create some openings in the grids.

If the double layer binate on binate grid in Fig 3.127 is inverted such that the top layer pattern becomes the bottom layer and the bottom layer pattern becomes the top layer a different configuration is obtained which is shown in Fig 3.135. Similarly, if the grid pattern in Fig 3.129 is inverted, the resulting pattern will be as shown in Fig 3.136. In double layer grids where the two layers have different patterns, very often the grids can be inverted as illustrated in Figs 3.135 and 3.136 to create other patterns.

The above approach discussed for creating different binate on binate grids can also be adopted in order to create a family of obnate on obnate grids. These grid patterns are shown in Figs 3.137, 3.139 and 3.140. Consider the obnate on obnate offset grid in Fig 3.137. A part of this grid has been enlarged in Fig 3.138 where the arrangement of members is clearly seen. The grid consists of two obnate layers placed at an offset with respect to each other in one direction.

Configurations in Figs 3.139 and 3.140 show two types of obnate on obnate patterns. The grid pattern in Fig 3.139 may be regarded as a obnate on smaller obnate pattern where the bottom layer consists of smaller obnate subdivisions. The pattern in Fig 3.140 consists of an obnate pattern for the top layer and an obnate pattern with rectangular subdivisions for the bottom layer.

The obnate on smaller obnate grid pattern shown in Fig 3.139 may be used as a basis for generating the patterns shown in Figs 3.141 and 3.142. The top layer elements in both patterns are kept intact. However, some of the bottom
Fig 3.137

Fig 3.138

Fig 3.139

Fig 3.140

Fig 3.141

Fig 3.142
layer elements and web elements have been removed to create some openings in the grid. The obnate on obnate pattern shown in Fig 3.140 may also be used as a protomorph to generated the patterns shown in Figs 3.143 and 3.144.

Two more obnate on obnate grid patterns may be created by inverting the grid patterns in Fig 3.139 and 3.140. These have been shown in Figs 3.145 and 3.146, respectively.

The double layer grid pattern in Fig 3.147 shows a "ternate on smaller ternate" pattern. The web members are inclined and exactly below the top layer members due to which they are not seen in the plan view of the grid. Fig 3.148 shows a pattern arising out of retinating the ternate on smaller ternate grid of Fig 3.147. After removing some of the bottom layer elements and web elements six additional elements have been introduced in the grid to create the central hexagonal opening.

Fig 3.149 shows another example of a ternate on smaller ternate grid. The bottom layer ternate pattern consists of smaller triangular subdivisions as well as hexagonal subdivisions. This grid has been retinated in order to generate the grid pattern shown in Fig 3.150. Here, some of the top layer elements have been removed along with some bottom layer elements and web elements. Also, three bottom layer elements have been added to the pattern in order to create the central hexagonal opening in the grid.

The grid pattern in Fig 3.151 is yet another example of a ternate on smaller ternate grid. The top layer consists of a regular pattern of triangular subdivisions while the bottom layer consists of triangular as well as hexagonal subdivisions.

Three new configurations arise when the ternate on ternate grids discussed
Additional ternate on ternate patterns

Fig 3.147

Fig 3.148

Fig 3.149

Fig 3.150

Fig 3.151

Fig 3.152

Fig 3.153

Fig 3.154
above are inverted. Figs 3.152, 3.153 and 3.154 are grid patterns arising out of inverting the patterns shown in Figs 3.147, 3.149 and 3.151, respectively.

Consider the three double layer grids in Figs 3.155, 3.157 and 3.159. These may be regarded as patterns consisting of combinations of hexagonal and ternate subdivisions. All the three grids have been represented in terms of their oblique views in Figs 3.156, 3.158 and 3.160 where the arrangement of members can be clearly seen.

The configurations in Figs 3.161 to 3.165 are examples of different patterns possible using prefabricated components. Modular components are available in the form of pyramidal units with triangular, square or hexagonal bases which may be assembled together. The double layer grid patterns in Figs 3.161 to 3.165 are created using a number of square, triangular, hexagonal and octagonal prefabricated units connected to each other at their apices by means of struts which form the bottom layer of the assembly.

Very often it may be necessary to design double layer grids to cover irregular areas. For example, the configurations in Figs 3.166 and 3.167 are some examples of grids over irregular shaped areas. The grid in Fig 3.166 shows a triangular double layer grid where pyramid type units are used.

It is also possible to use combinations of different types of double layer grids. In the grid pattern shown in Fig 3.167, the plan consists of a hexagonal area in the central region and six rectangular areas around it. The pattern chosen to cover this area shows a ternate on ternate double layer grid over the hexagonal plan and an obnate on binate pattern over the six rectangular areas.

An interesting family of structures has been designed and researched upon recently. These are referred to as "harmonic structures" since the subdivisions
Fig 3.166

Fig 3.167
in the patterns of the structures are based on harmonic progression. Presently, research is being conducted with respect to simple trusses, Ref 27. As an example, consider the truss configuration in Fig 3.168. This truss has eight subdivisions which are equal in length. A formex formulation may be written describing the truss in Fig 3.168 with respect to the indicated normat in terms of a scheme as shown below

\[
\text{TRUSS=':::}
\]
\[
\text{T1=RIN(1,8,2)|\{(0,0;2,0),(2,0;1,2),(1,2;0,0)\}}
\]
\[
\text{T2=RIN(1,7,2)|[1,2;3,2]}
\]
\[
\text{T=T1#T2}
\]
\[
\text{DRAW T'}
\]

Now, let the truss pattern be changed such that the subdivisions are not equal but based on a harmonic progression. This effect can be obtained by using the relations

\[
\text{E1=TRAN(1,-8)|T}
\]
\[
\text{E=MB(0.7708,0.6,0.85,0.6)|E1}
\]

To begin with, formex E1 represents the truss shown in Fig 3.169. The function TRAN(1,-8) effects translation in the first direction along the negative x-axis for a distance of 8 units so that the centre of the truss coincides with the origin in the global coordinate system as shown in Fig 3.169. Next, a metribifect retronorm is used for scaling the truss. A detailed explanation of the metribifect retronorm has been given earlier in Section 2.8.3. The basifactors 0.7708 and 0.6 as well as the metrifactors 0.85 and 0.6 are chosen such that the overall length of the truss and its depth remains the same. The positions of the inclined bars change such that they are nearer to each other at the two ends and further apart at the centre of the truss. Formex E represents
An Example of a Harmonic Truss

E = MB(0.777087033, 0.6, 0.85, 0.6) | E1

Fig 3.170
the "harmonic truss" shown in Fig 3.170. Researchers working on developing such harmonic structures believe that this concept is now possible to pursue and build due to the widespread use of robots and numerically controlled machine tools in the steel construction industry which allow economic manufacture of plane and three dimensional structures in which the bars and nodes are of different sizes.

A new family of grid patterns are made available based on harmonic subdivisions. The grid in Fig 3.171 is an example of a "single layer harmonic grid" with a binate pattern. To begin with, a binate grid with equal subdivisions is generated. A part of the grid is shown in Fig 3.172 along with a normat. The formex formulation used to generate the grid with respect to the indicated normat may be written as

\[ F_1 = \text{RINID}(6,6,2,2) | \text{ROSAD}(1,1) | [0,0;2,0] \]

To begin with, only a quarter of the grid is generated. The above formulation gives rise to a binate grid pattern with six subdivisions in both directions. Then the relation

\[ F_2 = \text{MB}(1,1,1.08,1.08) | F_1 \]

is used, where the metribifect retronorm is used for accelerated scaling of the grid in both directions. Finally, the grid is replicated around the coordinates of the right hand top corner of the grid. This is achieved by using the rosad function as shown below

\[ F = \text{ROSAD}(18.97713,18.97713) | F_2 \]

Formex F gives rise to the harmonic grid shown in Fig 3.171.
Fig 3.171

Fig 3.172

part of the binate
on binate pattern

normat
lines

Fig 3.173
A similar method is used to create the harmonic grid with a ternate pattern in Fig 3.173. In Figs 3.174 and 3.175, two types of quadnate patterns have been used for the single layer harmonic grids. The metrifactors 1.08 and 1.08 used for the binate grid are used in relation to all the three single layer grids shown in Figs 3.174 and 3.175.

The metritrifect retronorm has been used for creating some "double layer harmonic grids". Figs 3.176 and 3.177 show two examples of double layer harmonic grids. The configuration in Fig 3.176 is an oblique view of a binate on binate lattice grid with harmonic subdivisions and the configuration in Fig 3.179 is a binate on binate offset grid with harmonic subdivisions.

Another family of patterns are shown in Figs 3.178 to 3.181. In these patterns all the lengths and two dimensional shapes are identical, all the joints are in rows and only two types of shapes make up the entire pattern. Such patterns may be used for single layer grids. If two identical patterns are placed directly above each other interconnected by web members, they may be used as lattice type grids.
Fig 3.174

Fig 3.175

Fig 3.176

Fig 3.177
Fig 3.178  Fig 3.179

Fig 3.180  Fig 3.181
3.5 MULTILAYER PATTERNS

Double layer grid patterns discussed in the previous section can be extended to create "triple layer" grid patterns. A triple layer grid consists of members arranged in top, bottom and middle layers interconnected by two sets of web members. One set of web members connects the top layer to the middle layer and another set of web members connects the middle layer to the bottom layer. In many cases, there can be a number of layers arranged on top of one another giving rise to "multilayer grids".

The greater structural depth available with multilayer grids makes them advantageous for spanning larger spans as compared to those possible with double layer grids. In recent years, the spaces within multilayer spaceframes have been explored by several researchers to examine the possibility of using them as habitable spaces. Multilayer configurations have been used for a number of mass housing projects and town planning schemes. Research on these ideas has been conducted throughout the world by various investigators. The principal contribution in this field is the work of Prof J F Gabriel at the School of Architecture in Syracuse University, New York, USA. In the studies conducted by Gabriel multilayer configurations and the possibility of dwellings inside them have been examined, Ref 28. Some of the basic concepts put forth by Gabriel have been discussed in the sequel.

In the approach adopted by Gabriel, a multilayer spaceframe is regarded as an aggregation of prismatic or polyhedral volumes. Such volumes have been used as architectural spaces. The multilayer spaceframe functions both as a "structural framework" and as an "architectural matrix". The search for structural patterns on one hand and the search for habitable spaces on the other, has led to the development of a "form language" which has been described in the sequel.
3.5.1 MEGAPOLYHEDRA

The basic unit developed by Gabriel is the "space truss" configuration, an example of which is shown in Fig 3.182, Ref 28. This particular space truss consists of a combination of eight octahedra and fourteen tetrahedra. The truss is made up of bars which are 4 metres in length. This length is chosen to correspond with the average floor to floor height of a conventional multistorey building. The total length of the truss is 32 metres. Fig 3.182 shows a front elevation, a side elevation and an oblique view of the space truss. Since the truss is made up of eight octahedra, it is referred to as an "8-frequency truss". The frequency of the truss module denotes the number of octahedra used to construct the space truss.

The space truss configuration is used to replace the edges of a polyhedron. For example, consider a tetrahedron and let the edges of the tetrahedron be replaced by 8-frequency space trusses. Different views of the resulting configuration are shown in Fig 3.183. Three plan views and three elevations of the tetrahedron from different view points are shown in the figure. Since the tetrahedron consists of six space trusses placed along its edges the configuration is referred to as a "megatetrahedron". The above example illustrates one of the main concepts used by Gabriel where the edges of a polyhedra are replaced by a more complex pattern. In general, if this concept is used in relation to any polyhedron using a space truss pattern to replace its edges, the resulting configuration may be referred to as a "megapolyhedron".

The concept of megapolyhedra opens up a variety of possibilities for creating multilayer configurations. For example, consider the megatetrahedron. If the megatetrahedron is replicated in two directions the resulting configuration may be represented by the sketch in Fig 3.184. To elaborate, the mega-tetrahedron can be used to create a sort of a double layer pattern as shown in Fig 3.184.
Fig 3.182 Different Views of a Megatetrahedron
Fig 3.184 A double layer megatetrahedral pattern

Fig 3.185

Fig 3.186

Fig 3.187
Although the sketch shows only one megatetrahedron in the centre of the pattern, each line element representing the pattern may be replaced by the space truss module. This three dimensional pattern can be described as an aggregation of several megatetrahedra with their apices connected to each other by space truss modules. If additional layers are added to this type of a double layer pattern, triple layer and multilayer patterns can be obtained.

The basic idea introduced in the above example may be extended to include different types of polyhedra. In fact, one may even use a combination of two types of polyhedra and replace their edges by a space truss module. This is illustrated in one of Gabriel’s configurations shown in Fig 3.185. The configuration consists of a combination of an octahedron and two tetrahedra where each of the edges of the polyhedra are replaced by an 8-frequency space truss. Figs 3.186 and 3.187 show different views of the same configuration.

As explained for the megatetrahedral matrix, the configuration in Fig 3.185 may be replicated three dimensionally to create a multilayer matrix consisting of an aggregation of megaoctahedra and megatetrahedra.

A similar concept is used to generate the configuration in Fig 3.188. The polyhedron used in this example is a cuboctahedron. In Fig 3.188 each of the edges of the cuboctahedron have been substituted by a 7-frequency space truss. Twenty four space trusses make up the megacuboctahedron. This megacuboctahedron can be replicated to fill space to create a megacuboctahedral matrix.

Another type of megacuboctahedron is shown in Fig 3.189. This time, the square faces of the polyhedron are replaced by binate on binate double layer grids. A number of these megacuboctahedra can be packed together to create a space filling megacuboctahedral matrix.
Fig 3.190 shows a megacuboctahedral configuration where the triangular faces of the cuboctahedron are replaced by ternate on ternate double layer grids. On comparing the configurations in Figs 3.188 and 3.190, it is seen that certain members have been removed from the central region of the ternate on ternate grids in Fig 3.190 such that only the space truss configurations remain, Fig 3.188. Therefore, the configuration in Fig 3.188 is a "retination" of the configuration of Fig 3.190.

Two other multilayer patterns are shown in Figs 3.191 and 3.192. In the configuration in Fig 3.191 ternate on ternate grids have been used along planes passing through the centre of the megacuboctahedron. Four sets of planes pass through the centre of the cuboctahedron as shown in the plan view and elevation in Fig 3.191.

In the megacuboctahedron shown in Fig 3.192 twelve space trusses come together at the centre and twenty four space trusses replace the edges of the megapolyhedron. This is another method of using a polyhedron and replacing its edges as well as the diagonals joining the opposite vertices to create a megapolyhedron. An elevation and an oblique view of the cuboctahedron are shown in Fig 3.192.

The megacuboctahedra in Figs 3.188 to 3.192 may be replicated in two directions to create a variety of space filling megacuboctahedral matrices. For example, if the megacuboctahedron shown in Fig 3.189 is replicated in two directions the resulting pattern would be as shown in Fig 3.193. Here again, the positions of adjacent megacuboctahedra are shown in thicker lines in the sketch. If additional layers of megacuboctahedra are added to this assembly, multilayer space filling configurations can be created.

The above examples illustrate some of the main concepts used by Gabriel.
Fig 3.193 A megacuboctahedral pattern

Fig 3.194 The Hexmod

Fig 3.195
Some basic patterns have been used to replace the edges or faces of a polyhedron to create a megapolyhedron. This concept may be further extended to include several megapolyhedra which can be assembled to create a megapolyhedral matrix. One may use different configurations for the space truss module and different types of polyhedra to generate families of megapolyhedra. Different types of polyhedra can also be used in combination to create a variety of megapolyhedra.

The next important concept which is discussed in the sequel is the filling up of voids in a megapolyhedron. In Gabriel's study, megapolyhedra are used as habitable spaces. The architectural framework must be able to support dwellings as well as services like staircases, elevators and corridors. This also means that the architectural framework has to be compatible with the structural framework. To ensure this compatibility the octahedron which is the basic element of the space truss is used to create the rooms inside a megapolyhedron. The habitable areas are based on a hexagonal module. Each module is a combination of six octahedra and is referred to as a "hexmod", Fig 3.194, Ref 29. Several hexmods can be assembled together as shown in Fig 3.195 to create a double layer pattern. The voids in the pattern in Fig 3.195 show areas where certain diagonal elements are removed in order to incorporate stairs and elevators for multilayer dwelling units.

For the actual dwelling unit, two hexmods are placed exactly above each other and separated by 4 metres long diagonals. This gives rise to a habitable cell as shown in Fig 3.196. These hexagonal cells are used to fill up the voids inside a megapolyhedron. For example, consider a megatetrahedron and let the void inside the polyhedron be filled in with hexmods. This is elaborated with the help of Figs 3.197 to 3.200. Four different stages in which the megatetrahedron is filled in with hexmods are illustrated in Figs 3.197 to 3.200.
Fig 3.196 A habitable cell

Fig 3.197

Fig 3.198

Fig 3.199

Fig 3.200
The same concept is used for the megapolyhedron shown in Fig 3.201. The megapolyhedron consists of an octahedron and a tetrahedron where the edges of both polyhedra are replaced by 8-frequency space trusses. The voids created within such a structural assembly are then filled in with hexmods. Four stages of the assembly are shown in Figs 3.201 to 3.204, Ref 30.

The concepts put forth by Gabriel illustrate the variety of possibilities with non-cubic configurations. Moreover, multilayer configurations such as the ones discussed above when used as habitable spaces, can be altered at a later stage providing flexibility for future expansion or change of use. This is why Gabriel often refers to them as "infinite structures", Ref 31.

The examples discussed above reveal that there are a number of basic underlying patterns which have been used for generating multilayer configurations. In the previous sections the concept of protomorphs was used to generate a variety of single and double layer patterns. The examples in this section which relate to multilayer configurations also bring out the relevance of protomorphs. For example, the megatetrahedral matrix in Fig 3.184 emerges as a protomorph for generating other megatetrahedral patterns. The megatetrahedral matrix in Fig 3.184 has been modified by adding hexmods to the pattern as seen in Figs 3.197 to 3.200. Also, the megatetrahedral pattern can be modified using the methods discussed in the previous section to create different patterns.

Similarly, a variety of megacuboctahedral patterns described earlier can serve as protomorphs to generate other multilayer configurations. The pattern consisting of megatetrahedra and megaoctahedra in Fig 3.185, also emerges as part of a basic underlying pattern of megatetrahedra and megaoctahedra which fill space.
Another basic pattern which emerges from this study is the pattern created by hexmods arranged to form a double layer or a multilayer pattern. This hexmod matrix can serve as a basis for creating other hexagonal configurations.

During the above discussions, protomorphs for multilayer configurations have not appeared in a very obvious manner. However, in every pattern generation activity certain underlying patterns are bound to emerge. Similarly, protomorphs which have emerged as a result of this study may be used as a basis for creating other multilayer space filling patterns. These protomorphs may be listed as:

1) megatetrahedral matrix
2) megatetrahedra-megaoctahedral matrix
3) megacuboctahedral matrix
4) hexmod matrix

It must be mentioned here that this list may be extended to suit particular applications. Any of these patterns may be used for generating multilayer space structure patterns or three dimensional space filling patterns.

A variety of space structure patterns have been created and discussed in this Chapter. It is shown that the concept of a protomorph provides a valuable aid in the search for patterns providing a systematic approach for pattern finding. The next Chapter deals with an approach for evolving different families of curved configurations.
CHAPTER FOUR

PARAGENIC TRANSFORMATIONS

4.1 INTRODUCTION

In order to generate space structure forms using the formex approach, the idea of a retronorm is widely employed. A formex formulation for a configuration may be transformed into geometric coordinates using a retronorm. Although standard retronorms like the cylindrical and spherical retronorms are presently available in Formian, a designer needs a considerably large vocabulary of shapes and forms. In an attempt to fulfil these requirements, the present Chapter deals with the introduction and applications of certain composite transformations. These transformations are results of using combinations of cylindrical and spherical retronorms. Although the concept of retronorms has been explained in Section 2.8.3, the introductory part of this Chapter briefly describes the characteristics of cylindrical and spherical retronorms. The major part of the study deals with establishing an approach whereby these standard retronorms may be combined to create a number of new retronorms. These ideas have been elaborated with the help of examples in the sequel.

4.2 CYLINDRICAL AND SPHERICAL RETRONORMS

A "graphical retronorm" may also be referred to as a "normat" and this term has been used throughout the study. A cylindrical retronorm may be graphically represented as shown in Fig 4.1. This cylindrical normat may be described as consisting of layers of concentric cylinders which are infinitely
Fig 4.1 Cylindrical Retronorm

Fig 4.2 Barrel Vault
long. Only a finite part of the retronorm is shown in the figure. Each division in the U1 direction relates to the radius $R$ of the cylinder and two layers of the retronorm obtained by $R=8$ and $R=10$ are shown in Fig 4.1. Divisions in the U2 direction relate to the angular coordinate $S$ usually given in degrees. The linear coordinate along the axis of the cylinder is specified by the divisions in the U3 direction.

The above retronorm provides a convenient reference system for describing a cylindrical configuration. For instance, consider the configuration in Fig 4.2. The formex formulation of the configuration may be given by the following Formian statements

\begin{align*}
A1 &= \text{RINIT}(10,10,2,2) | \text{ROSAT}(1,1) | [10,2,2; 10,2,0] \\
A2 &= \text{RIN}(2,5,2) | [10,0,0; 10,2,2] \\
A3 &= \text{RIN}(3,5,4) | \text{LAMIT}(10,2) | A2 \\
A &= \text{PEX} | (A1#A3) \\
B &= \text{BC}(1,120/20,1) | A
\end{align*}

In the above formulation, the first four statements describe the compret of the configuration with respect to the indicated normat. At this point it may be noted that the task of describing the configuration in terms of the x-y-z coordinate system would have been a difficult one. However, describing the configuration in terms of the indicated normat becomes much more convenient. In the last statement, BC stands for the basicylindrical retronorm which relates cylindrical and Cartesian coordinates. When the basicylindrical retronorm is activated, it transforms the signets of formex A into global coordinates. Therefore, both formices A and B represent the same configuration but while formex A represents the configuration with respect to the indicated normat, formex B represents the configuration with respect to the x-y-z coordinate system.
The linear basifactors for the radius of the cylinder and its length are specified as 1. The angular basifactor is specified as $120/20=6$ therefore, each division along $S$ measures 6 degrees.

The general form of a cylindrical retronormic transformation may be given as

$$BC(b_1,b_2,b_3)$$

where the parameters $b_1$, $b_2$ and $b_3$ are basifactors in the $R$, $S$ and $Z$ directions, respectively.

The above example illustrates how the basicylindrical retronorm may be used to generate a cylindrical configuration. Similarly, the basispherical retronorm may be conveniently used to generate a spherical configuration. A graphical representation of the spherical retronorm is shown in Fig 4.3. This retronorm may be described as consisting of layers of spheres of different radii. Each division in the $U_1$ direction relates to the radius of the sphere. Divisions in directions $U_2$ and $U_3$ relate to angular coordinates given in terms of degrees. $S$ relates to the latitudinal coordinate of the retronorm and $T$ relates to the longitudinal coordinate as shown in the figure.

The normat in Fig 4.3 may be used as a convenient reference system to formulate a spherical configuration. For instance, consider the configuration shown in Fig 4.4. A formex formulation for the configuration may be written as

$$E_1 = \text{PEX} | \text{RINIT}(20,6,2,2) | \{[10,1,1;10,3,1] \# \text{ROSAT}(1,2) | \ldots \}$$

$$E_2 = \text{BS}(1,360/40,48/12) | E_1$$
Fig 4.3 Spherical Retronorm

Fig 4.4 Spherical Configuration
Formex variable $E_1$ represents the configuration with respect to the indicated normat and $E_2$ represents the configuration relative to the $x$-$y$-$z$ global coordinate system. BS stands for the basispherical retronormic function which relates spherical and Cartesian coordinates. The basifactor along the radius of the sphere is specified as 1. The angular basifactor in the latitudinal direction is chosen as $360/40=9$. Therefore each division in the $U_2$ direction measures 9 degrees. The angular basifactor in the longitudinal direction is specified as $48/12=4$ so that each division in the $U_3$ direction measures 4 degrees.

In general a spherical retronormic transformation may be represented as

$$BS(b_1,b_2,b_3)$$

where parameters $b_1$, $b_2$ and $b_3$ are basifactors in the R, S and T directions, respectively.

The configurations in Figs 4.2 and 4.4 are results of using cylindrical or spherical retronormic transformations. In both the above examples, the cylindrical or spherical retronorm was used just once in order to create the curved configuration. In the next section, configurations have been generated using double retronormic transformations and these have been elaborated in the sequel.
4.3 DOUBLE RETRONORMIC TRANSFORMATIONS

Once again, consider the configuration in Fig 4.2. The formex formulation of the configuration which was discussed in the previous section, has been reproduced below

\[
A_1 = \text{RINIT}(10,10,2,2) | \text{ROSAIT}(1,1) | [10,2,2; 10,2,0]
\]
\[
A_2 = \text{RIN}(2,5,2) | [10,0,0; 10,2,2]
\]
\[
A_3 = \text{RIN}(3,5,4) | \text{LAMIT}(10,2) | A_2
\]
\[
A = \text{PEX} | (A_1 \# A_3)
\]
\[
B = \text{BC}(1,120/20,1) | A
\]

Formex variable \( B \) represents the configuration with respect to the global coordinate system as shown in Fig 4.2. Now, let this configuration be subjected to a second cylindrical transformation. This may be achieved by writing

\[
C = \text{BC}(1,120/20,1) | B
\]

The result of the second cylindrical transformation is the configuration shown in Fig 4.5, represented by formex variable \( C \). The two transformations may also be represented as a composite function by writing

\[
C = \text{BC}(1,120/20,1) | \text{BC}(1,120/20,1) | A
\]
\[
b_4\ b_5\ b_6\ b_1\ b_2\ b_3
\]

Here, formex variable \( A \) represents the compret of the configuration with respect to the normat in Fig 4.2 and formex variable \( C \) represents the configuration with respect to the global coordinate system as shown in Fig 4.5. In the above relation, retronormic functions have been used in combination.
The basifactors $b_1$, $b_2$ and $b_3$, relate to the first cylindrical transformation and $b_4$, $b_5$ and $b_6$ relate to the second cylindrical transformation. In the above example, since both transformations were identical, one may say that

$$b_1 = b_4 = 1$$

$$b_2 = b_5 = 6$$

and

$$b_3 = b_6 = 1.$$ 

The general form of a double cylindrical transformation may be represented as

$$BC(b_4, b_5, b_6) \mid BC(b_1, b_2, b_3)$$

Basifactors for both transformations control different aspects of the resulting configuration. If the basifactors are varied the resulting configuration would change. For instance, if the basifactor $b_2=6$ in the first cylindrical transformation is changed to $b_2=8$ and the Formian statement given as

$$D = BC(1, 6, 1) \mid BC(1, 8, 1) \mid A$$

then formex variable $A$ represents the compret of the configuration and formex variable $D$ represents the configuration shown in Fig 4.6. It may be noted that by increasing the basifactor $b_2$, the divisions in the $S_1$ direction increase in dimension for the configuration in Fig 4.6.

In the above composite transformation, a repeated use of retronormic functions was made. A composite retronormic transformation of this type is referred to as a "paragenic transformation". A retronorm created as a result of a paragenic transformation is referred to as a "paragenic retronorm". The term paragenic is constructed from the Greek prefix "para" implying "beyond" and the combining form "genic" implying "producing".
4.3.1 CYLINDRICAL PARAGENIC TRANSFORMATIONS

Configurations in Figs 4.5 and 4.6 were obtained as a result of paragenic transformations. However, since both transformations were cylindrical, they may be regarded as "cylindrical paragenic transformations". By changing any of the basifactors in the transformation, a family of retronorms may be generated.

Both configurations in Figs 4.5 and 4.6 relate to a family of paragenic retronorms. A graphical representation of a typical member of this family is shown in Fig 4.7. The perspective view and cross-section shows only a single layer of the retronorm. Similar to the description of the cylindrical retronorm, the retronorm in Fig 4.7 may also be described as being infinitely long in the direction of the z-axis and consisting of layers as will be elaborated in the sequel.

Each of the basifactors relate to different aspects of the retronorms. Basifactors b1 and b4 relate to the radii of the two cylindrical retronorms. The angular basifactor b2 relates to the curved cross-section of the retronorm, along S1 in the x-y plane as shown in Fig 4.7. The angular basifactor b5 along S2 relates to the angle of the curve with respect to the x-axis. The linear basifactors b3 and b6 are in the same direction as the z-axis.

Since b1 relates to the radius of the first cylindrical retronorm, changes in the value of b1 alters the resulting retronorm. This has been illustrated through the following example. Consider the formex E which is subjected to two cylindrical transformations given below

\[ E = \text{RIN}(2,150,2)|[10,0,0;10,2,0] \]
\[ F = B C(1,2,1)|B C(b1,2,1)|E \]
Fig 4.7
The value of the basifactor $b_1$ in the first transformation is increased, while keeping the value of $b_4$ constant. With different values of $b_1$, the plot of the resulting formex $F$ changes as shown in Fig 4.8. The different values of $b_1$ are given alongside in the figure. Only a single retronorm has been shown in the series of cross-sections in Fig 4.8 to avoid confusion. With increasing values of $b_1$, the retronorm repeatedly intersects itself. It is interesting to observe that the manner in which the retronorm intersects itself follows a particular pattern which can be seen in Fig 4.8.

The linear basifactor $b_4$ relates to the radius of the second cylindrical transformation. If $b_4$ is increased keeping the value of $b_1$ constant, the resulting retronorms get larger and larger as shown in Fig 4.9. Formex $E$ used for this illustration is given by

$$E = \text{RIN}(2,70,2) | [10,0,0;10,2,0]$$

Formex $E$ is then subjected to two cylindrical transformations as shown in Fig 4.9. Three layers of the retronorm corresponding to different values of $b_4$ are shown in the figure.

Since $b_2$ is an angular basifactor it is associated with the curved cross-section of the retronorm as shown in Fig 4.10. Numbers 1 to 5 indicate the path traced by the curve. The curve lies in the x-y plane and crosses at the origin of the global coordinate system. For negative values of $b_2$, the curve only changes its direction, that is, it travels from point 5 to 1, but does not deviate from its path along the cross-section.

The angular basifactor $b_5$ is associated with the angle of the curve with respect to the x-axis as shown in Fig 4.11. The effect of increasing $b_5$ is that the internal angle goes on increasing. At one stage the retronorm begins to
$E=\text{RIN}(2,150,2) \mid [10,0,0;10,2,0]$

$F=\text{BC}(1,2,1) \mid \text{BC}(b1,2,1) \mid E$

$\begin{align*}
\text{b1} &= 2.5 \\
\text{b4} &= 1 \\
\end{align*}$

$\begin{align*}
\text{b1} &= 7 \\
\text{b4} &= 1 \\
\end{align*}$

$\begin{align*}
\text{b1} &= 10 \\
\text{b4} &= 1 \\
\end{align*}$

$\begin{align*}
\text{b1} &= 12.5 \\
\text{b4} &= 1 \\
\end{align*}$

$\begin{align*}
\text{b1} &= 15 \\
\text{b4} &= 1 \\
\end{align*}$

Fig 4.8 Paragenic Cross-sections
A = BC(1,360/140,1) | BC(10,450/140,1) | E

Fig 4.9 Effect of changing values of $b_4$

A = BC(1,360/140,1) | E
A1 = BC(1,3,1) | A
A2 = BC(1,12.5,1) | A

Fig 4.10

A = BC(1,360/140,1) | E
S2

Fig 4.11 Effect of changing values of $S_2$
intersect itself, as shown in the figure. This is similar to the effect of keeping the value of b4 constant and increasing the value of b1.

Consider a case where the first cylindrical normat is placed at a distance from the z-axis but still kept parallel to it as shown in the first section of Table 4.1. The normat is then subjected to a second cylindrical transformation as represented by the second figure shown alongside. In the course of this study, the cylindrical normat is placed in three different positions in the global coordinate system and subjected to a second cylindrical transformation. The result of these transformations have been elaborated in the sequel.

These double cylindrical transformations give rise to three families of paragenic retronorms which have been represented in Table 4.1. Each one of them will be discussed in detail in the sequel.

The first set of paragenic transformations shown in Table 4.1 may be represented in a general manner by writing

\[ BC(b_4,b_5,b_6) \mid \text{TRAN}(1,m) \mid BC(b_1,b_2,b_3) \]

The above paragenic function consists of two retronormic transformations along with an intermediate cardinal function. The translation function moves the cylindrical normat in the first direction and the amount of translation is specified by the value of m. The first composite transformation discussed earlier with respect to configurations in Figs 4.5 and 4.6 may also be represented by the above paragenic function simply by assigning the value of m as 0.

In Fig 4.7, the paragenic retronorm obtained as a result of using m=0 was illustrated. Fig 4.12 shows a view and cross-section of the retronorm obtained as a result of using m=2.5. Comparing the two retronorms, shows how the
<table>
<thead>
<tr>
<th>Table 4.1</th>
<th>CYLINDRICAL PARAGENIC TRANSFORMATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First Transformation</strong></td>
<td><strong>Second Transformation</strong></td>
</tr>
<tr>
<td>$z, Z_1(b_3)$</td>
<td>$z, Z_2(b_6)$</td>
</tr>
<tr>
<td>$S_1(b_2)$</td>
<td>$S_2(b_5)$</td>
</tr>
<tr>
<td>$x, R_1(b_1)$</td>
<td>$x, R_2(b_4)$</td>
</tr>
</tbody>
</table>

$$BC(b_4, b_5, b_6); \text{TRAN}(1, m); BC(b_1, b_2, b_3)$$

<table>
<thead>
<tr>
<th>$z, Z_2(b_6)$</th>
<th>$z, Z_2(b_6), S_2(b_5)$</th>
<th>$z, Z_2(b_6)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1(b_2)$</td>
<td>$S_2(b_5)$</td>
<td>$S_1(b_2)$</td>
</tr>
<tr>
<td>$x, Z_1(b_3)$</td>
<td>$x, Z_1(b_3), R_2(b_4)$</td>
<td>$x, Z_1(b_3), R_2(b_4)$</td>
</tr>
<tr>
<td>$R_1(b_1)$</td>
<td></td>
<td>$R_1(b_1)$</td>
</tr>
</tbody>
</table>

$$BC(b_4, b_5, b_6); \text{TRAN}(1, m); \text{VERAS}(0, 0, -90); BC(b_1, b_2, b_3)$$

$$BC(b_4, b_5, b_6); \text{TRAN}(1, m); \text{VERAT}(0, 0, -90); BC(b_1, b_2, b_3)$$
Perspective View

Cross-section

Fig 4.12
introduction of a translation motion changes the resulting retronorm. A family of paragenic retronorms are created as the value of m is increased. For example, consider the scheme given below

```
SECTION=':m:
L1=RIN(2,100,2)|[10,0,0;10,2,0]
L2=BC(1,4,1)|TRAN(l,m)|BC(1,4,1)|L1
DRAW L2'
```

The above scheme can be executed by giving different values for m. Some of the plots obtained by assigning different values to m have been shown in Fig 4.13. The figure shows how the cross-sections of the resulting retronorms change with different values of m. The induction statements used to execute the schemes have been given alongside the cross-sections. The retronorms on the left hand side are a result of positive values of m. For the same values of m, if a negative sign is assigned, a mirror image of the retronorms is obtained as shown in the right half of the figure.

On examining the retronorms shown in Fig 4.13, it is interesting to observe various effects brought about by different values of m. One such effect seen with respect to configurations in Figs 4.5 and 4.6 is referred to as a "lemnic" effect. Configurations which display an effect where the concave surface changes to a convex one are regarded as lemnic surfaces. For instance, configurations relating to paragenic retronorms with certain values of m display the lemnic effect. The term lemnic is derived from the Latin word "lemniscus" meaning ribbon. It may be noted that the vertical and horizontal disposition of the cross-sections in Fig 4.13 is only a matter of accommodating the cross-sections and does not represent any variable in that direction.

Configurations relating to paragenic retronorms may be regarded as paragenic
configurations. Figs 4.5 and 4.6 were examples of single layer paragenic configurations. A variety of surfaces can be generated using cylindrical paragenic retronorms some examples of which have been described in the sequel.

Consider the double layer arched configuration shown in Fig 4.14. A cross-section of the arch is shown in Fig 4.15. A formex formulation describing the configuration may be given in terms of a scheme as shown below

```
ARCH='::
A1=RINIT(25,4,2,2)|ROSAT(1,1)\{[10,0,0;10,2,0],..
  [10,0,0;11,1,1]\}
A2=RINIT(24,3,2,2)|ROSAT(2,2)\{11,1,1;11,3,1\}
A3=BC(1,120/50,1)|PEX|(A1#A2)
A4=BC(1,120/50,1)|TRAN(1,10)|A3
A5=VERAD(0,0,45)|A4
A6=VERAD(0,0,-90)|LAM(2,4.5)|A5
A=RIN(3,4,2)|A6'
```

Initially, formex A4 is generated which represents part of the arched configuration shown in cross-section in Fig 4.16. The configuration relates to the surface of the paragenic retronorm shown in Fig 4.17. The bold lines in the cross-section represent the part of the retronorm to which the arched configuration of Fig 4.16 relates. The configuration is then rotated by 45 degrees as represented by formex A5. Next, a mirror image is obtained using the lambda function and then the arched configuration is rotated once again in the x-y plane as represented by formex A6. Finally, the configuration is replicated in the third direction to create the arched configuration in Fig 4.14 represented by formex A.
The configuration has been generated as a narrow strip, with only four divisions in the third direction, since it is the cross-section which is of interest here rather than the length of the configuration. If required, the configuration may be easily formulated such that there are more number of divisions along its length.

A similar technique is used to obtain the double layer vaulted configuration in Fig 4.18. The configuration is based on a paragenic retrornorm which is shown in Fig 4.19. The part of the retrornorm that relates to the curved configuration is shown in bold lines in Fig 4.19. A formex formulation describing the configuration may be given in terms of a scheme as shown below

\[
\text{VAULT} = \text{''::}
\]

\[
\text{V1} = \text{RINIT(35,4,2,2)} \mid \text{ROSAT(1,1)} \mid \{[10,0,0;10,2,0],\ldots [10,0,0;11,1,1]\}
\]

\[
\text{V2} = \text{RINIT(34,3,2,2)} \mid \text{ROSAT(2,2)} \mid [11,1,1;11,3,1]
\]

\[
\text{V3} = \text{BC(1,2,1)} \mid \text{PEX} \mid (\text{V1}\#\text{V2})
\]

\[
\text{V4} = \text{BC(1,5,1)} \mid \text{TRAN(1,30)} \mid \text{V3}
\]

\[
\text{V5} = \text{VERAD(0,0,90)} \mid \text{LAM(2,0)} \mid \text{V4}
\]

\[
\text{V} = \text{RIN(3,4,2)} \mid \text{V5}'
\]

In the above formulation, formex V4 represents only half of the vaulted configuration which is then reflected as shown in Fig 4.20 and rotated to obtain the complete vault represented by formex V5. Formex V represents the entire vaulted configuration which is obtained by replicating the vault in the third direction four times using the rindle function.

The next example describes the formulation for the corrugated configuration shown in Fig 4.21. A cross-section of the configuration is shown in Fig 4.22. A single unit of the configuration is shown in thicker lines, enclosed in a box defined by dotted lines. Initially, the single unit is generated based on a
Fig 4.18 Perspective View

Fig 4.19

Fig 4.20 Cross-section
Fig 4.21 Perspective View

Fig 4.22 Cross-section
cylindrical paragenic retronorm. This single unit is then replicated thrice in order to create the entire configuration. A formex formulation describing the configuration may be given as

\[
\text{CORRUGATE} = ':: \\
C1 = \text{PEX}|\text{RINIT}(15,10,2,2)|\text{ROSAT}(1,1)[10,0,0;10,2,0] \\
C2 = \text{BC}(1,6,1)|C1 \\
C3 = \text{BC}(1,6,1)|\text{TRAN}(1,5)|C2 \\
C = \text{RIN}(1,3,20)|C3'
\]

Formex variable C3 represents the single unit of the configuration shown in the cross-section in Fig 4.22 in thicker lines. Formex C represents the entire configuration of Fig 4.21 obtained by replicating the single unit thrice using the rindle function.

Structures which are combinations of spaceframes and membranes may be regarded as hybrid space structures. For example, consider the configuration in Fig 4.23 which consists of three arches with membranes spanning between them. This may be regarded as a hybrid space structure. A plan view of the configuration is shown in Fig 4.24 and an elevation is shown in Fig 4.25.

The final shape of the membrane can only be determined after an analysis process. In Figs 4.23 to 4.25 the membrane has been represented by a series of dots using a hand rendering technique. However, a scheme for the arches may be written as shown below

\[
\text{HYBRID} = ':: \\
H1 = \text{RIN}(2,15,2)|\text{ROSAT}(1,1)|[[10,0,0;10,2,0],[10,0,0;11,1,1]} \\
H2 = \text{RIN}(2,14,2)|[11,1,1;11,3,1] \\
H3 = \text{BC}(1,4,1)|\text{PEX}|(H1#H2)
\]
Fig 4.23 Perspective View

Fig 4.24 Plan View

Fig 4.25 Elevation
Formex $H_4$ represents part of the arch shaped configuration shown in Fig 4.26. The configuration is then rotated and reflected as shown in Fig 4.27 and represented by formex $H_5$ to obtain the horse shoe shaped arch. This complete arch is rotated once again as shown in Fig 4.28. Formex $H$ represents the whole configuration where the single arch is repeated thrice using the rindle function.

The above examples illustrate how cylindrical paragenic retronorms may be used to generated surfaces for spaceframes, shells and hybrid space structures. Another family of retronorms is generated when the cylindrical normat is placed parallel to the x-axis and then subjected to a second cylindrical transformation. This family of retronorms is discussed in the sequel.

**CYLINDRICAL NORMAT PARALLEL TO THE X-AXIS:**

Consider the cylindrical configuration shown together with the global coordinate system in Fig 4.29. A formex formulation for the configuration may be given as

\[
E_1 = \text{RINIT}(15,15,2,2) | \text{ROSAT}(1,1) | 10,0,0; 10,2,0
\]
\[
E_2 = \text{BC}(1,5,1) | E_1
\]

Formex variable $E_1$ represents the configuration with respect to the normat in Fig 4.29 and formex variable $E_2$ represents the same configuration but with respect to the x-y-z global coordinate system. The four corners of the vault
Fig 4.26

Fig 4.27

Fig 4.28
have been labelled as A, B, C and D. Now, the configuration is placed parallel to the x-axis by rotating the vault using the relation

\[ E_3 = \text{VERAS}(0,0,-90) \mid E_2 \]

Formex variable \( E_3 \) represents the configuration in its new position parallel to the x-axis. Next, the configuration is subjected to a second cylindrical transformation by writing

\[ E = \text{BC}(1,5,1) \mid E_3 \]

The resulting configuration represented by formex \( E \) is shown in Fig 4.30. Comparing the configurations in Figs 4.29 and 4.30 it is observed that the arched opening AB is transformed into a flattened end and the opening CD is curved at an angle with respect to the x-axis.

The paragenic configuration in Fig 4.30 relates to a new retronorm created as a result of the above composite transformation. A general description of the retronorm may be given in terms of a paragenic function of the form

\[ \text{BC}(b_4,b_5,b_6) \mid \text{TRAN}(1,m) \mid \text{VERAS}(0,0,-90) \mid \text{BC}(b_1,b_2,b_3) \]

This paragenic function relates to the second section in Table 4.1. Basifactors \( b_1, b_2 \) and \( b_3 \) relate to the first cylindrical normat. Vertition and translation functions reposition the cylindrical normat such that it is parallel to the x-axis at a distance \( m \) from the origin as shown in the table. Basifactors \( b_4, b_5 \) and \( b_6 \) relate to the second cylindrical transformation.

By changing the basifactors or the canonic parameter \( m \) in the above transformation, a family of retronorms may be generated. For example, the
configuration in Fig 4.30 may also be represented in terms of the above paragenic function simply by assigning the value of m as 0.

A typical retronorm which may be regarded as a representative of this family of paragenic transformations has been shown in Figs 4.31 to 4.33. Fig 4.31 is a general view of the retronorm. A view of the retronorm from the x-axis is shown in Fig 4.32 and Fig 4.33 shows a side view of the retronorm from the z-axis.

Basifactors of both cylindrical transformations and the canonic parameter m in the intermediate cardinal function affect the resulting retronorm. The retronorm has been represented as a single layer throughout the study so that the effect of every basifactor and canonic parameter on the retronorm may be clearly studied. The linear basifactor b1 relates to the larger opening of the retronorm along R1, parallel to the z direction. The linear basifactor b4 relates to the length of the retronorm in the direction of the x-axis.

The angular basifactor b2 relates to the coordinates of the curve along the larger opening of the retronorm as shown in Fig 4.31. The angular basifactor b5, relates to the included angle of the retronorm. Increasing the value of b5 increases the included angle of the retronorm. For example, Fig 4.34 shows three layers of the retronorm representing different values of b5.

The parameter m specifies the amount of movement of the cylindrical normat in the first direction. Changes in the values of m changes the cross-section of the resulting retronorm. If m=0, one end of the retronorm is flattened to a straight line as shown in the configuration in Fig 4.30. As the value of m increases, the flat end opens up as shown in Fig 4.31. However, the sizes of both openings are different.
Fig 4.31

view from x-axis

Fig 4.32

view from z-axis

Fig 4.33

Fig 4.34
Fig 4.35 shows a double layer paragenic configuration with both the front and rear openings of different shapes and sizes. A formex formulation for the configuration may be given as

\[
SAC = \prime:: \\
A1 = \text{RINIT}(15,7,2,2) | \text{ROSAT}(1,1) | \{[10,0,0; 10,2,0], \ldots [10,0,0; 11,1,1]\} \\
A2 = \text{RINIT}(14,6,2,2) | \text{ROSAT}(2,2) | [11,1,1; 11,3,1] \\
A3 = \text{VERAS}(0,0,-90) | \text{BC}(1,6,1) | \text{PEX} | (A1\#A2) \\
A = \text{BC}(1,5,2) | \text{TRAN}(1,5) | A3'
\]

Paragenic configurations like the one in Fig 4.30 may be used as shell structures where one end is required to have a curved opening and the other has to be a flattened end. There may also be a need for a configuration which is like a barrel vault but with different sizes of openings at either ends as shown in Fig 4.35. As illustrated in the above examples, such configurations may be conveniently generated using the above retronorm.

Now, consider the single layer configuration shown in Fig 4.36. A formex formulation for the configuration may be given as

\[
D1 = \text{PEX} | \text{RINIT}(15,10,2,2) | \text{ROSAT}(1,1) | [10,0,0; 10,2,0] \\
D2 = \text{BC}(1,6,3) | D1 \\
D3 = \text{TRAN}(1,-10) | \text{VERAS}(0,0,-90) | D2 \\
D = \text{BC}(1,5,2) | D3
\]

It may be noted that the amount of translation in the first direction is specified as \( m = -10 \). Therefore the cylindrical normat is shifted in the direction of the negative x-axis before being subjected to the second cylindrical transformation. Formex variable \( D \) represents the configuration shown in Fig 4.36. This
demonstrates the effect of using negative values for m.

A systematic study of the changes in the retronorms for different values of m is represented in Fig 4.37. The configuration which is plotted in Fig 4.37 may be written in terms of a scheme as shown below

\[ \text{HOOD} = ': \text{m}: \n\text{H1} = \text{RIN}(2,100,2) | [10,0,0; 10,2,0] \\
\text{H2} = \text{BC}(1,4,1) | \text{RIN}(3,2,20) | \text{H1} \\
\text{H3} = \text{TRAN}(1, \text{m}) | \text{VERAS}(0,0,-90) | \text{H2} \\
\text{H} = \text{BC}(1,3.5,1) | \text{H3}' \]

The above scheme is executed by giving different values to the variable m and the resulting retronorms have been shown in Fig 4.37. As the value of m is increased, the flattened end of the retronorm begins to open up. Further increase in the value of m causes the two ends to draw closer and closer as shown in the figure.

If for the same values of m negative values are assigned, mirror images of the paragenic retronorms are obtained. However, an interesting phenomenon occurs for certain negative values of m as shown in Fig 4.37, where the retronorm consists of two oppositely oriented parts. The configuration in Fig 4.36 relates to a retronorm in this range. It may be noted that in Fig 4.37, the disposition of retronorms in the vertical direction is only a matter of accommodating them and does not represent any variable in that direction.

On examining the array of retronorms in Fig 4.37, it is observed that the retronorms show various effects as the values of m change. Therefore a variety of surfaces relating to these retronorms may be generated as required by designers in a particular architectural context. One such effect is referred to as
a "peric" effect. This is characterised by the surface of the retronorm being
curved like a vault at one end and flattened at the other. Configurations having
a peric effect may be seen in Figs 4.30 and 4.35. The word "peric" is derived
from a Latin word "pera" meaning a "bag".

Configurations in Figs 4.38 to 4.40 are based on this family of retronorms.
The configuration in Fig 4.38 is also shown in a plan view in Fig 4.38. The
formex formulation for the configuration may be given as

\[
\begin{align*}
A1 &= \text{RINIT}(15,7,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \\
A2 &= \text{VERAS}(0,0,-90) | \text{BC}(1,3,3) | A1 \\
A3 &= \text{BC}(1,4,2) | \text{TRAN}(1,0) | A2 \\
A &= \text{LAM}(1,0) | \text{VERAD}(0,0,-90) | \text{LAM}(3,0) | A3'
\end{align*}
\]

In the above formulation formex A represents the entire configuration. If the
amount of translation is changed from 0 to 5, in formex A3, the resulting
configuration changes as shown in Fig 4.39. There is an opening in the
configuration which is a result of increasing the amount of translation. The plan
view of the configuration is shown alongside in Fig 4.39.

The configuration in Fig 4.40 is another example of a paragenic configuration
belonging to the family of retronorms discussed above. The single barrel vault
type unit is repeated thrice to create a series of barrel vaults. The vaults have
one curved opening and one flat end as shown in Fig 4.40.

**CYLINDRICAL NORMAT PARALLEL TO THE Y-AXIS:**

Consider the third set of paragenic transformations in Table 4.1. The
cylindrical normat is placed parallel to the y-axis in the global coordinate system
as shown and subjected to a second cylindrical transformation. These composite
Fig 4.38  
Plan view

Fig 4.39  
Plan view
transformations may be represented in a general manner by a paragenic function of the form

\[ BC(b_4,b_5,b_6)|\text{TRAN}(1,m)|\text{VERAT}(0,0,-90)|BC(b_1,b_2,b_3) \]

In the above paragenic transformation basifactors \( b_1, b_2 \) and \( b_3 \) relate to the first cylindrical transformation. The effect of the vertition function is to rotate the normat from its position parallel to the z-axis such that it lies parallel to the y-axis. Translation motion in the first direction, specified by \( m \), causes the normat to move along the x-axis as shown in the Table 4.1. The normat is then subjected to the second cylindrical transformation represented by basifactors \( b_4, b_5 \) and \( b_6 \).

A family of paragenic retronorms may be obtained by assigning different values to the basifactors and the parameter \( m \). A typical member of this family of retronorms is graphically represented in Fig 4.41. The shape of the retronorm resembles a doughnut but only a part of the retronorm is shown in the figure for clarity. The linear basifactor \( b_4 \) relates to inner radius of the retronorm. The linear basifactor \( b_1 \) relates to the internal radius of the retronorm. The angular basifactors \( b_2 \) and \( b_5 \) relate to coordinates along the inner and outer radii of the retronorm respectively, Fig 4.41.

If the value of \( m=0 \), the retronorm simply changes to a spherical one. This may be regarded as a special case in these paragenic transformations. Different parts of the doughnut shaped retronorm may be used to create a variety of configurations. Some examples have been illustrated in the sequel.

Consider the configuration shown in Fig 4.42 which consists of a series of interconnected saddle shapes. A single saddle shape may be created using the cylindrical paragenic configuration discussed above. This saddle shape relates
Fig 4.41

Fig 4.42
to a part of the doughnut shaped retronorm. Once a single saddle shape is obtained, it can then be replicated to create the entire configuration. A formex formulation for a single saddle shaped configuration may be given in terms of a scheme as shown below

$$\text{SADDLE} = 
\begin{align*}
E_1 &= \text{PEX} | \text{RINIT}(10,10,2,2) | \text{ROSAT}(1,1) | [5,0,0;5,2,0] \\
E_2 &= \text{VERAD}(0,0,30) | \text{BC}(1,6,1) | E_1 \\
E_3 &= \text{VERAT}(0,0,-90) | E_2 \\
E_4 &= \text{TRAN}(1,20) | \text{VERAS}(0,0,-90) | E_3 \\
E_5 &= \text{VERAD}(0,0,30) | \text{BC}(1.5,4,1.5) | E_4'
\end{align*}$$

Figs 4.43 to 4.46 illustrate the step by step procedure of obtaining the saddle shaped configuration. In the above formulation formex $E_2$ represents the cylindrical configuration relative to the global coordinate system as shown in Fig 4.43. Formex $E_3$ represents the configuration after it is rotated such that the axis of the cylinder lies parallel to the y-axis as shown in Fig 4.44. Formex $E_4$ represents the position of the configuration after a rotation and a translation movement as shown in Fig 4.45. The configuration is then subjected to a second cylindrical transformation and the result is the saddle shaped configuration represented by formex $E_5$ as shown in Fig 4.46. Three such saddle shapes can be combined to create the configuration in Fig 4.42. This may be obtained by using the rindle function as shown below

$$E = \text{RIN}(3,3,13) | E_5$$

The above statement creates formex variable $E$ which represents the three saddle shapes shown in Fig 4.42.

A similar technique can be used to create the configuration in Fig 4.47 which
$E_1 = \text{RINIT}(10, 10, 2, 2) \| \text{ROSAT}(1, 1) \| [5, 0, 0; 5, 2, 0]$  
$E_2 = \text{VERAD}(0, 0, 30) \| \text{BC}(1, 6, 1) \| E_1$

Fig 4.43

$E_3 = \text{VERAT}(0, 0, -90) \| E_2$

Fig 4.44

$E_4 = \text{TRAN}(1, 20) \| \text{VERAS}(0, 0, -90) \| E_3$

Fig 4.45

$E_5 = \text{VERAD}(0, 0, 30) \| \text{BC}(1.5, 4, 1.5) \| E_4$

Fig 4.46
consists of three different sizes of saddle shaped surfaces. A formex formulation for the configuration may be written in terms of a scheme as shown below

\[
\text{MESH}='::\text{A1 }=\text{RINIT}(10,18,2,2)\text{IROSAT}(1,1)\text{I}[5,0;0;5,2,0]\text{A2 }=\text{TRAN}(1,20)\text{I}BC(1,9,1)\text{I}A1\text{A3 }=\text{TRAN}(1,17.5)\text{I}BC(1,5,9,1)\text{I}A1\text{A4 }=\text{BC}(2,9,1)\text{I}A1\text{A5 }=\text{TRAN}(1,-20)\text{I}(A2\#A3\#A4)\text{A6 }=\text{VERAS}(0,0,-90)\text{I}VERAT(0,0,-90)\text{I}A5\text{A }=\text{VERAD}(0,0,30)\text{I}BC(1,3,1)\text{I}TRAN(1,40)\text{I}A6'\]

In the above scheme, formices A2, A3 and A4 represent three cylindrical configurations with different diameters. These cylindrical configurations are then subjected to a second cylindrical transformation represented by formex A to obtain the final configuration of Fig 4.47.

As another example, consider the configuration in Fig 4.48. The configuration in Fig 4.47 may be used as a basis to create the configuration in Fig 4.48. After obtaining formices A2, A3 and A4, one may write,

\[
\text{C1 }=\text{A4}\#\text{LAM}(1,0)\text{I}(A2\#A3)\text{C2 }=\text{VERAS}(0,0,-90)\text{I}VERAT(0,0,-90)\text{I}C1\text{C }=\text{VERAD}(0,0,45)\text{I}BC(1,2,1)\text{I}TRAN(1,40)\text{I}C2\]

Formex C1 represents a combination of the three cylindrical configurations as well as a reflection of the two smaller cylinders. These five interconnected cylindrical configurations are then rotated and translated and subjected to a
second cylindrical transformation as represented by formices C2 and C. Formex C represents the entire configuration in Fig 4.48.

The doubly curved configurations in Figs 4.42, 4.47 and 4.48 may be regarded as trial shapes for structures such as shells, membrane surfaces or cable net systems. Consider the hybrid space structure shown in Fig 4.49. There are three semicircular arches placed at unequal distances from each other. This configuration is generated using the paragenic retronorm discussed above.

Tent shaped configurations can also be created using paragenic transformations. For example, consider the configuration shown in Fig 4.50. A formex formulation describing the configuration may be given as

\[
T_1 = \text{RINIT}(10,40,2,2)|\text{ROSAT}(1,1)[10,0,0;10,2,0] \\
T_2 = \text{BC}(1,4,1)|T_1 \\
T_3 = \text{TRAN}(1,-12.5)|\text{VERAT}(0,0,-90)|T_2 \\
T = \text{BC}(1,360/80,1)|T_3
\]

Formex T2 represents a vault shaped configuration which is rotated and translated as represented by formex T3. Formex T represents the tent shaped configuration obtained after a second cylindrical transformation as shown in Fig 4.50.

A hybrid tent shaped configuration is shown in Fig 4.51. This can also be obtained by using the paragenic transformations discussed above. The configuration represents a tent shaped membrane held in place by a spaceframe along its edges. The entire assembly can be anchored to another support system like a wall.

A number of configurations for pneumatic structures may be generated using the
above paragenic transformations as illustrated in the following examples. For instance, consider the configuration shown in terms of a plan view as well as an oblique view in Fig 4.52. This consists of a tubular surface with a binate mesh projected on it. The formex formulation for the configuration may be written in terms of a scheme given by

\[
\text{PIE}=':
\text{P1}=\text{PEX}|\text{RINIT}(10,20,2,2)|\text{ROSAT}(1,1)|[10,0,0;10,2,0]
\text{P2}=\text{BC}(1,9,1)|\text{P1}
\text{P3}=\text{TRAN}(1,15)|\text{VERAT}(0,0,-90)|\text{P2}
\text{P}=\text{VERAT}(0,0,90)|\text{ROSAD}(0,0)|\text{BC}(1,4.5,1)|\text{P3'}
\]

In the above formulation, formex P2 represents a cylindrical configuration and formex P3 represents the configuration in its rotated position in the global coordinate system. The second cylindrical transformation creates the tubular configuration in Fig 4.52.

Another configuration which could be used as a trial shape for a pneumatic structure is shown in Fig 4.53. The configuration may be formulated as

\[
\text{PNEU1}=':
\text{P1}=\text{RINIT}(10,20,2,2)|\text{ROSAT}(1,1)|[5,0,0;5,2,0]
\text{P2}=\text{VERAD}(0,0,30)|\text{BC}(1,6,1)|\text{P1}
\text{P3}=\text{VERAD}(20,0,-90)|\text{VERAS}(0,0,-90)|\text{P2}
\text{P}=\text{VERAD}(0,0,60)|\text{BC}(1,2,1)|\text{P3'}
\]

Formex P2 represents a cylindrical configuration which is parallel to the z-axis. The configuration is then rotated twice so that it lies parallel to the y-axis as represented by formex P3. The second cylindrical transformation creates the required shape as shown in Fig 4.53. If this configuration is replicated by
As another example, consider the configuration in Fig 4.55. A cross-section of the configuration is shown in Fig 4.56. There are two different sizes of tubular shapes which may be formulated as shown below

\[
A_1 = \text{RINIT}(10,18,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \\
A_2 = \text{TRAN}(1,20) | \text{BC}(1,9,1) | A_1 \\
A_3 = \text{TRAN}(1,10) | \text{BC}(1,4.5,1) | A_1 \\
A_4 = \text{LAM}(1,0) | A_3 \\
A = A_2#A_3#A_4
\]

In the above formulation, formex A represents the three interconnected vaults. Fig 4.56 shows the plots of formices A2, A3 and A4. Next, let this configuration be subjected to two rotations and a translation given by the following Formian statements

\[
B = \text{VERAS}(0,0,90) | \text{VERAT}(0,0,-90) | A \\
C = \text{VERAS}(0,0,30) | \text{BC}(1,2,1) | \text{TRAN}(1,40) | B
\]

The resulting configuration, represented by formex C is shown in Fig 4.57.

Another combination of three sizes of curved surfaces is seen in the configuration of Fig 4.58. A cross-section of the configuration is shown in Fig 4.59. The formex formulation for the configuration of Fig 4.58 may be given
Fig 4.55

Fig 4.56

Fig 4.57
C1 = PEX | RINIT(10,18,2,2) | ROSAT(1,1) | [5,0,0;5,2,0]  
C2 = TRAN(1,30) | BC(1,180/20,1) | C1  
C3 = TRAN(1,17.5) | BC(1.5,180/20,1) | C2  
C4 = BC(2,9,1) | C1  
C = C2 # C3 # C4  

In the above formulation C2, C3 and C4 represent three cylindrical configurations, as shown in the cross-section in Fig 4.59. Formex C represents three interconnected barrel vaults of three different diameters. This configuration is then subjected to a second cylindrical transformation after two rotations and two translations as given by

C5 = TRAN(1,-20) | C  
C6 = VERAS(0,0,90) | VERAT(0,0,-90) | C5  
C = VERAD(0,0,45) | BC(1,2,1) | TRAN(1,40) | C6  

The resulting configuration is shown in Fig 4.60. This may be used as a trial shape for a shell structure or a pneumatic structure.

Two more examples of trial shapes for pneumatic structures are shown in Figs 4.61 and 4.62. The configuration in Fig 4.61 may be generated in a manner similar to the one used for the configuration in Fig 4.60. The two smaller cylindrical shapes can be reflected to create the five interconnected cylindrical shapes and then subjected to a second cylindrical transformation.

The configuration in Fig 4.62 may be formulated as a combination of four tubular shapes and four cylindrical shapes. The formex formulation may be written as
\[ \text{PIE2} = \quad : \]
\[ A_1 = \text{RINIT}(10,10,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \]
\[ A_2 = \text{BC}(1,9,1) | A_1 \]
\[ B = \text{VERAT}(0,0,-90) | A_2 \]
\[ C = \text{VERAT}(0,0,90) | \text{BC}(1,4.5,1) | \text{TRAN}(1,20) | B \]
\[ E = \text{TRAN}(1,20) | \text{TRAN}(3,20) | C \]
\[ D = E \# \text{TRAN}(1,40) | A_2 \]
\[ G = \text{ROSAS}(10,10) | D' \]

In the above formulation, formex \( A_2 \) represents the cylindrical configuration. Formex \( E \) represents the tubular configuration. Formex \( D \) represents a combination of the two shapes which is replicated using the rosas function to create the configuration in Fig 4.62.

In the examples illustrated throughout this study, a binate pattern has been used for all the configurations. This is because the emphasis in the study is on developing curved surfaces and not on patterns. However, any configuration can be generated with a different pattern whenever required as illustrated in the examples in the previous chapter. One may use the concept of paragenic surfaces in conjunction with a variety of patterns to create different configurations.
4.3.2 SPHERICAL TRANSFORMATIONS OF CYLINDRICAL RETRONORMS

This section introduces a family of paragenic retronorms which have been created using combinations of cylindrical and spherical retronorms. As in the previous sections, a systematic approach is adopted where the cylindrical normat is placed in different positions parallel to the three axes in the global coordinate system and then transformed spherically. Table 4.2 shows these three sets of paragenic transformations. The resulting retronorms are described using paragenic functions, consisting of double retronormic transformations along with intermediate cardinal and provial functions which have also been included in Table 4.2.

CYLINDRICAL NORMAT PARALLEL TO Z-AXIS :

The first set of transformations shown in Table 4.2 involve a cylindrical normat placed parallel to the z-axis and subjected to a spherical transformation. An example of such a transformation has been described below. Consider the cylindrical configuration shown in Fig 4.63. The same configuration was used earlier in Fig 4.29. The formex formulation of the configuration with respect to the indicated normat is given by formex variable E1 and formex variable E2 represents the same configuration with respect to the global coordinate system as shown below

\[ E1 = \text{PEX} | \text{RINIT}(15, 15, 2, 2) | \text{ROSAT}(1, 1) | [10, 0, 0; 10, 2, 0] \]
\[ E2 = \text{BC}(1, 5, 1) | E1 \]

The four corners of the vault have been marked as A, B, C and D to study the effect of the second transformation. Let the cylindrical configuration be subjected to a spherical transformation using the relation
### TABLE 4.2

**SPHERICAL TRANSFORMATIONS OF CYLINDRICAL NORMATS**

<table>
<thead>
<tr>
<th>First Transformation</th>
<th>Second Transformation</th>
<th>Resulting Retronorm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z, Z_l(b3)$</td>
<td>$T(b6)$</td>
<td>$S_1(b2)$</td>
</tr>
<tr>
<td>$x, R_1(b1)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$z, Z_l(b3)$</td>
<td>$T(b6)$</td>
<td>$S_1(b2)$</td>
</tr>
<tr>
<td>$x, R_2(b4)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$z, Z_l(b3)$</td>
<td>$T(b6)$</td>
<td>$S_1(b2)$</td>
</tr>
<tr>
<td>$x, R_2(b4)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$z, Z_l(b3)$</td>
<td>$T(b6)$</td>
<td>$S_1(b2)$</td>
</tr>
<tr>
<td>$x, R_1(b1), R_2(b4)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Diagrams

1. BS(b4, b5, b6) | TRAN(1, m) | BC(b1, b2, b3)
2. BS(b4, b5, b6) | TRAN(1, m) | VERAS(0, 0, -90) | BC(b1, b2, b3)
3. BS(b4, b5, b6) | TRAN(1, m) | VERAT(0, 0, -90) | BC(b1, b2, b3)
The resulting paragenic configuration is shown in Fig 4.64. This is an oblique view of the configuration and a plan view of the same configuration is shown in Fig 4.65. The vaulted configuration of Fig 4.63 with two arched openings AB and CD, Fig 4.63, changes due to the spherical transformation. The circular opening CD is flattened to a straight line as shown in Fig 4.64. Also the axis of the vault which was parallel to the z-axis is curved as seen in Figs 4.64 and 4.65. The configuration relates to a new retronorm.

The above transformation may be represented in a general manner by a paragenic function of the form

\[ BS(b4, b5, b6) \mid TRAN(1, m) \mid BC(b1, b2, b3) \]

Basifactors b1, b2 and b3 relate to the cylindrical retronorm and b4, b5 and b6 relate to the spherical retronorm as shown in Table 4.2. The value of m determines the amount of movement of the cylindrical normat along the x-axis. The above paragenic function represents a family of paragenic retronorms which may be generated using different values for the basifactors and for the canonic parameter m. Fig 4.66 shows a typical member of this family of retronorms. Also, a cross-section of the retronorm is shown in Fig 4.67.

It is interesting to observe that the curved cross-section of the retronorm is similar to the first set of cylindrical paragenic transformations studied in the previous section, Table 4.1. As the values of m change, the changes in the cross-sections follow a similar pattern to the previous paragenic retronorms.

As indicated in the Fig 4.66, b2 is an angular basifactor which relates to the curve along the cross-section of the retronorm. Let A, B, C, D and E be points
Oblique View
Fig 4.66

Cross-section
Fig 4.67
on a single layer of the retronorm as shown in Fig 4.66. As b2 travels from A to B, the angular coordinate T travels from point D to A. Also, as b2 travels from B to C, T travels from E to C. The points D, B and E always remain colinear. As the value of b6 changes, the cross-section of the retronorm changes from a straight line to a curved one as shown in Fig 4.66.

Another aspect of the retronorm is shown in Figs 4.68 to 4.70. For the value of m = 0 the retronorm comprises of two equal parts positioned such that they are staggard and facing each other as shown in Fig 4.68. As the value of m increases, which implies that the cylindrical normat moves towards the left, the right half of the retronorm becomes smaller and the left half becomes larger and changes its shape as shown in Figs 4.69 and 4.70. Figs 4.68 to 4.70 show three examples where the retronorms change as the value of m increases from 0 to 5 and 10.

Some paragenic configurations relating to this family of retronorms may be formulated as described below. For example, consider the configuration represented by formex variable E3 in Fig 4.64. This paragenic configuration may be reflected by writing

\[ E = \text{LAM}(1,0) \mid E3 \]

The resulting configuration is shown in Fig 4.71. Now, if the configuration in Fig 4.64 is reflected in the third direction by writing

\[ F = \text{LAM}(3,0) \mid E3 \]

The plot of F is the configuration shown in Fig 4.72. The configuration in Fig 4.72 can be reflected once again by using the relation
Fig 4.68

$x, R_1(b_1), R_2(b_4)$

$m = 0$

Fig 4.69

$m = 5$

Fig 4.70

$m = 10$
F2 = LAM(1,0) | F

The resulting configuration is shown in Fig 4.73. All the three configurations in Figs 4.71 to 4.73 relate to the same family of paragenic transformations shown in the first section of Table 4.2.

Another family of paragenic configurations is shown in Figs 4.74 to 4.76. Formex variable E1 is used once again to create the configuration in Fig 4.74 as given below

\[ \begin{align*}
E1 &= \text{RINIT}(15,15,2,2) | \text{ROSAT}(1,1)[10,0,0;10,2,0] \\
F1 &= \text{BC}(1,3,1) | E1 \\
F2 &= \text{BS}(1,6,3) | \text{TRAN}(1,0) | F1 \\
F3 &= \text{LAM}(1,0) | F2
\end{align*} \]

In the above set of Formian statements, formex variable F1 represents a cylindrical configuration which is subjected to a spherical transformation. However, the cylindrical configuration is not translated in the first direction and \( m=0 \). The resulting configuration is then reflected using the lambda function to obtain the configuration in Fig 4.74 represented by formex variable F3.

If the plot of formex F2 is reflected in the third direction using the relation

\[ F4 = \text{LAM}(3,0) | F2 \]

The resulting configuration is as shown in Fig 4.75. Let this configuration be reflected once again in the first direction using the relation

\[ F5 = \text{LAM}(1,0) | F4 \]
The plot of formex F5 is shown in Fig 4.76.

The above examples bring out some of the variety of possibilities for creating curved configurations using paragenic transformations with a cylindrical transformation followed by a spherical one. These configurations may be used for different types of shell structures or pneumatic structures.

**CYLINDRICAL NORMAT PARALLEL TO X-AXIS:**

In the second set of paragenic transformations in Table 4.2, the cylindrical normat is positioned such that it lies parallel to the x-axis. The normat is then transformed spherically and the resulting retronorm is shown in the table. If a cylindrical configuration is transformed spherically, it may be represented in terms of a scheme given by

\[
\begin{align*}
\text{WAVE} &= ':: \\
\text{W1} &= \text{PEX} | \text{RINIT}(20,10,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \\
\text{W2} &= \text{BC}(1,180/40,1) | \text{W1} \\
\text{W3} &= \text{TRAN}(1,5) | \text{VERAS}(0,0,-90) | \text{W2} \\
\text{W} &= \text{BS}(1,180/40,90/20) | \text{W3}'
\end{align*}
\]

In the above scheme, formex variable \(W2\) represents a vaulted configuration with respect to the x-y-z global coordinate system as shown in Fig 4.77. The corners of the vault are marked as A, B, C, and D to show the effect of the second transformation. Formex \(W3\) represents the configuration in its rotated position such that it now lies parallel to the x-axis as seen in Fig 4.78. This configuration is then transformed spherically and the resulting configuration is as shown in Fig 4.79. The vaulted configuration changes to an almost wave-like form. Another view of the configuration from the z-axis, Fig 4.80, helps to visualise how the four corners of the vault are transformed. This
Fig 4.77

Fig 4.78
Oblique View
Fig 4.79

View from -ve z-axis
Fig 4.80

View from y-axis
Fig 4.81
configuration relates to a new paragenic retronorm which is described in detail in the sequel.

In general, the above transformations may be given by

$$BS(b_4, b_5, b_6) \ | \ TRAN(l, m) \ | \ VERAS(0, 0, -90) \ | \ BC(b_1, b_2, b_3)$$

Different values for the basifactors $b_1$ to $b_6$ and the canonic parameter $m$ would represent different retonorms. An oblique view, plan and cross-section of a typical member of this family of retronorms may be graphically represented as shown in Figs 4.82 to 4.84. The basifactor $b_1$ relates to the length of the retronorm along the $z$-axis, $b_2$ relates to the larger opening of the retronorm and $b_4$ relates to the length of the retronorm in the direction of the $x$-axis as shown in Fig 4.82. Both $b_2$ and $b_5$ are angular basifactors relating to the curved cross-section of the retronorm. The basifactor $b_6$ relates to the included angle of the retronorm. As the value of $b_6$ increases, the included angle of the retronorm increases.

When $m=0$, one end of the resulting retronorm assumes a pointed end as shown in Fig 4.85. However, as the values of $m$ increase, the pointed end changes to a cross-section similar to the other end. With increasing values of $m$ the two ends of the retronorms are drawn closer and closer. The scheme which represents all the paragenic retronorms in Fig 4.85 may be given as

$$WAVE(m)=':$$

$$W_1 = RIN(3, 2, 6) | RIN(3, 20, 2) | [10, 0, 0; 10, 2, 0]$$

$$W_2 = BC(1, 6, 1) | W_1$$

$$W_3 = TRAN(1, m) | VERAS(0, 0, -90) | BS(1, 4, 3) | W_2'$$

Different induction statements with different values of $m$ can be used to execute
Oblique View
Fig 4.82

Plan View
Fig 4.83

View from z-axis
Fig 4.84
Fig 4.85
the above scheme. Fig 4.85 shows the retronsorms along with the induction statements used to execute the schemes. For values of $m$ between -2.5 and -15, the retronsorms consist of two conical parts oriented in opposite directions. As the value of $m$ goes on increasing, the two ends of the retronsorm are drawn closer together. Configurations can be generated relating to any of these curved surfaces or specific parts of the retronsorms.

This family of retronsorms is particularly suitable for creating curved fan shaped configurations like the one in Fig 4.79. As another example, consider the configuration in Fig 4.86. This is a plan view of the configuration which consists of four curved sections. A perspective view of the same configuration is shown in Fig 4.87. Once again formex $E_1$ is used for the compret of the configuration of each section. The formex formulation for the configuration may be written as

$$E_2 = BC(1,6,1) | E_1$$
$$E_3 = \text{TRAN}(1,5) | \text{VERAS}(0,0,-90) | \text{BS}(1,4.5,4.5) | E_2$$

In the above formulation, formex $E_2$ represents the vaulted configuration and formex $E_3$ represents the paragenic fan shaped configuration which is first reflected in the second direction and then replicated in the x-z plane using the relation

$$E = \text{ROSAS}(0,0,4,90) | \text{REF}(2,0) | E_3$$

The resulting configuration is shown in Figs 4.86 and 4.87. A similar transformation is used to create the configuration in Fig 4.88. This is a plan view of the configuration which consists of eight paragenic shell type surfaces. Using the same formex $E_1$ one may write
Plan View
Fig 4.86

Perspective View
Fig 4.87
Plan View
Fig 4.88

Perspective View
Fig 4.89
\[ E_2 = BC(1,3,1) | E_1 \]
\[ E_3 = TRAN(1,5) | VERAS(0,0,-90) | BS(1,4.5,4.5) | E_2 \]
\[ E = ROSAS(0,0,8,45) | REF(2,0) | E_3 \]

Formex E represents the configuration in Figs 4.88 and 4.89. Comparing the above formulation with the previous one, it is seen that the value of \( b_2 \) is changed from 6 to 3. This gives rise to a curved configuration which is reflected in the second direction and then replicated eight times to create the required configuration. This may be used to represent a shell structure comprising of eight paragenic curved shells.

**CYLINDRICAL NORMAT PARALLEL TO THE Y-AXIS :**

In the next set of paragenic transformations a cylindrical normat is placed parallel to the y-axis in the x-y-z global coordinate system. The normat is then transformed spherically as shown in Table 4.2. If these set of transformations are represented in a general manner, they may be written in terms of a paragenic function given by

\[ BS(b_4,b_5,b_6) | TRAN(1,m) | VERAT(0,0,-90) | BC(b_1,b_2,b_3) \]

In the above set of transformations, shown in Table 4.2, the vertition function rotates the cylindrical normat such that it lies parallel to the y-axis and translation in the first direction, moves the normat along the x-axis. The amount of movement is specified by the value of \( m \). The result of the above set of transformations is that a family of new retronorm is generated. A general view and cross-section of a typical member of this family of retronorms is presented in Figs 4.90 and 4.91, respectively.

Basifactors of both transformations affect the family of paragenic retronorms.
Oblique View
Fig 4.90

View from x-axis
Fig 4.91
Basifactors of both transformations control different aspects of the resulting retronorm. The linear basifactors b1 and b4 relate to the radius of the first cylindrical transformation and the first spherical transformation, respectively and act along the x-axis as shown in Fig 4.90. This family of retronorms display a cross-section similar to the family of retronorms generated in the first set of paragenic transformations in Table 4.1. The cross-sections change with different values of m. The angular basifactor b6 is associated with the internal angle of the retronorm as shown in Fig 4.90. The angular basifactor b2 relates to the extent of the curve in the x-z plane. The basifactor b5 relates to the angle of the cross-section with the x-z plane as seen in Fig 4.90.

Configurations based on this family of retronorms have been described in the sequel. Consider the family of configurations shown in Figs 4.92 to 4.95. All these configurations have been generated using the same paragenic function but varying the value of the canonic parameter m as shown along side the figures. To begin with, formex E1 has been used once again for the configuration in Fig 4.92 and a basicylindrical transformation has been used as shown below

\[ E1 = \text{RINIT}(15,15,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \]
\[ P1 = \text{BC}(1,3,1)|E1 \]

Next, the cylindrical configuration is rotated such that it lies parallel to the y-axis and then translated in the first direction. The configuration is then subjected to a second spherical transformation. These operations may be represented by the following Formian statements

\[ P2 = \text{TRAN}(1,2.5) | \text{VERAT}(0,0,-90)|P1 \]
\[ P3 = \text{BS}(1,180/30,90/30)|P2 \]

The result of these set of equations is the configuration in Fig 4.92. The
Fig 4.92

$m = 2.5$

Fig 4.93

$m = 5$

Fig 4.94

$m = 15$

Fig 4.95

$m = 25$
parameter \( m \) which specifies the amount of movement of the cylindrical configuration in the first direction is given as 2.5. With different values of \( m \), the resulting configuration changes as demonstrated in Figs 4.93 to 4.95.

The basifactor \( b_1 \) relates to the radius of the first cylindrical transformation. If the value of \( b_1 \) increases the resulting configuration changes as shown in the next example. The configurations in Figs 4.96 and 4.98 are plots of the same formex \( P \) where \( b_1 = 1 \) in the case of Fig 4.96 and \( b_1 = 3 \) in the case of Fig 4.98. The configurations are also represented in terms of their plan views in Figs 4.97 and 4.99.

The next set of examples illustrate the effect of changing the angular basifactor \( b_6 \) which relates to the internal angle of the retronorm. Consider the configuration in Fig 4.100. This is obtained by using the same formex \( P \) but changing the value of \( b_6 \) to 45/30. A plan view of the configuration is shown in Fig 4.101. If the value of \( b_6 \) is changed to 135/30, the resulting configuration changes to the one shown in Fig 4.102. A plan view of the same configuration is shown in Fig 4.103. This family of configurations can be extended as required by varying the basifactors \( b_1 \) to \( b_6 \) as well as the canonic parameter \( m \) to obtain a wide variety of paragenic configurations.
4.3.3 **SPHERICAL PARAGENIC TRANSFORMATIONS**

The concept of paragenic transformations is extended in this section to study the effects of transforming spherical normats. This has been illustrated in Table 4.3 which shows a spherical normat subjected to a second spherical transformation and a typical member of the resulting family of paragenic retronorms is shown in the table.

The spherical normat in relation to the x-y-z axis remains parallel to the three axes even if it is rotated in any direction. Therefore, unlike the cylindrical normat, the spherical normat is not placed in different positions in the global coordinate system before the second retronormic transformation.

A general form of the double spherical transformation describing the resulting family of retronorms is included in the table and also reproduced below

\[
\text{BS}(b4,b5,b6) \mid \text{TRAN}(1,m) \mid \text{BS}(b1,b2,b3)
\]

By assigning different values to each of the basifactors or canonic parameter \(m\), a family of retronorms may be obtained. A typical member of this family of retronorms has been enlarged as shown in Fig 4.104. Also, a plan view of the retronorm is shown in Fig 4.105.

The linear basifactors \(b1\) and \(b4\) act along the direction of the x-axis controlling the length of the retronorm in that direction. The angular basifactor \(b2\) controls the length of the curve of the retronorm as shown in Fig 4.104. The angular basifactor \(b5\) controls the angle of the curve with respect to the x-axis as shown in Fig 4.104. The angular basifactor \(b3\) acts along the curve of the retronorm in the x-z plane as shown in Figs 4.104 and 4.105.
<table>
<thead>
<tr>
<th>First Transformation</th>
<th>Second Transformation</th>
<th>Resulting Retronorm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1(b_3)$</td>
<td>$T_2(b_6)$</td>
<td></td>
</tr>
<tr>
<td>$x, R_1(b_1)$</td>
<td>$x, R_2(b_4)$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$T_1(b_3)$</th>
<th>$T_2(b_6)$</th>
<th>$T_1(b_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1(b_2)$</td>
<td>$S_2(b_5)$</td>
<td>$T_2(b_6)$</td>
</tr>
<tr>
<td>$x, R_1(b_1)$, $R_2(b_4)$</td>
<td>$x, R_1(b_1)$, $R_2(b_4)$</td>
<td>$x, R_1(b_1)$, $R_2(b_4)$</td>
</tr>
</tbody>
</table>

BS(b4,b5,b6) | TRAN(1,m) | BS(b1,b2,b3)
Fig 4.104 Oblique View

Fig 4.105 Plan View
4.3.4 CYLINDRICAL TRANSFORMATIONS OF SPHERICAL RETRONORMS

The next family of paragenic retronorms is generated by subjecting a spherical normat to a cylindrical transformation as illustrated in Table 4.4. Consider a spherical configuration which is subjected to a cylindrical transformation. In general, the retronorm may be described in terms of a paragenic function which is given below

$$BC(b_4,b_5,b_6) | TRAN(l,m) | BS(b_1,b_2,b_3)$$

A typical member of this family of retronorms is shown in Figs 4.106 and 4.107. This retronorm also shows the lemnic effect in cross-section. The angular basifactors $b_2$ and $b_5$ relate to the extent of the curve and its internal angle respectively. This has been shown in Figs 4.106 and 4.107. Also basifactors $b_1$ and $b_4$ act along the x-axis. The basifactors $b_1$ to $b_6$ and the canonic parameter $m$, can be changed to create families of paragenic surfaces. This has been illustrated with the help of some examples in the sequel.

Once again, consider the formex $E_1$ which has been used in most of the previous examples. Let formex $E_1$ be transformed spherically using the relation

$$E_1 = RINIT(15,15,2,2) | ROSAT(1,1) | [10,0,0;10,2,0]$$
$$E_2 = BS(1,180/30,90/30) | E_1$$

Formex $E_2$ represents a spherical configuration. Next, let this spherical configuration be subjected to a cylindrical transformation using the relation

$$E_3 = BC(1,5,1) | E_2$$
<table>
<thead>
<tr>
<th>First Transformation</th>
<th>Second Transformation</th>
<th>Resulting Retronorm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1(b_3)$</td>
<td>$T(b_3)$</td>
<td>$BC(b_4, b_5, b_6)$</td>
</tr>
<tr>
<td>$x, R_1(b_1)$</td>
<td>$x, R_2(b_4)$</td>
<td></td>
</tr>
<tr>
<td>$z, Z(b_6)$</td>
<td>$z, Z$</td>
<td></td>
</tr>
</tbody>
</table>

$BC(b_4, b_5, b_6) \perp \text{TRAN}(1, m) \perp BS(b_1, b_2, b_3)$
Oblique View
Fig 4.106

View from z-axis
Fig 4.107
The resulting configuration is shown in Fig 4.108. The configuration relates to a paragenic retronorm. If the basifactor \( b_2 \) in the above transformation is changed from \( 180/30 \) to \( 90/30 \), the resulting configuration changes to the one shown in Fig 4.109. The paragenic transformation used for the configuration of Fig 4.109 may be written as

\[
E_3 = BC(1,5,1) | BS(1,90/30,90/30) | E_1
\]

Let the plot of formex \( E_3 \) be reflected in the first direction using the relation

\[
E_4 = LAM(1,0) | E_3
\]

The resulting configuration is shown in Fig 4.110.

Consider the configuration in Fig 4.111. This may be obtained by reflecting the plot of formex \( E_4 \) in the third direction using the relation

\[
E_5 = LAM(3,0) | E_4
\]

These examples illustrate some of the configurations which are possible to create using cylindrical transformations of spherical configurations.

The next set of examples illustrate the effect of changing the value of the angular basifactor \( b_3 \). Once more, consider the relation

\[
E_3 = BC(1,5,1) | BS(1,180/30,90/30) | E_1
\]

In the above transformation, the value of \( b_3 \) is taken as \( 90/30 = 3 \). The resulting configuration is shown in Fig 4.108. If the value of \( b_3 \) is changed to 4 the configuration changes to the one shown in Fig 4.112, that is, the angle of the
Fig 4.108

Fig 4.109

Fig 4.110

Fig 4.111
surface in the x-z plane increases. Figs 4.113 and 4.114 show two more examples of configurations obtained by using b3=5 and b6=6, respectively.

The linear basifactor b6 relates to the length of the retronorm along the z-axis. To illustrate this, let formex E3 be written as

\[ E3 = BC(1,5,1.5) | BS(1,6,3) | E1 \]

The resulting configuration gets elongated in the direction of the z-axis as shown in Fig 4.115. This configuration can then be reflected in the third direction using the equation

\[ F = LAM(3,0) | E3 \]

The plot of formex F is shown in Fig 4.116.

The configuration in Fig 4.117 illustrates the effect of changing the value of the canonic parameter m. In all the above examples, the spherical configuration was not translated in the first direction and therefore, the value of m was 0. If the spherical configuration represented by formex E2 is translated in the first direction such that m=10 and then subjected to a second cylindrical transformation, the resulting configuration changes. This configuration can then be reflected in the third direction to obtain the configuration in Fig 4.117. The formex formulation for the configuration may be written as

\[ G = LAM(3,0) | BC(1,5,1) | TRAN(1,10) | E2 \]

Configurations generated using the above family of paragenic retronorms have a cross-section similar to the family of retronorms introduced in the first set of transformations in Table 4.1. Therefore, configurations from both these
Fig 4.115

Fig 4.116

Fig 4.117
families may be combined as shown in the following examples.

Consider the configuration in Fig 4.118. This is created using a combination of two paragenic surfaces. The formex formulation of the configuration is written in terms of a scheme as shown below:

\[
\begin{align*}
\text{DUNE} &= '::' \\
\text{D1} &= \text{RINIT}(20,20,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \\
\text{D2} &= \text{BS}(1,180/40,90/40) | \text{D1} \\
\text{D} &= \text{BC}(1,180/40,1) | \text{D2} \\
\text{F1} &= \text{RINIT}(20,10,2,2) | \text{ROSAT}(1,1) | [10,0,0;10,2,0] \\
\text{F} &= \text{TRAN}(3,-10) | \text{BC}(1,180/40,1) | \text{BC}(1,180/40,1) | \text{F1} \\
\text{G} &= \text{D} # \text{F}'
\end{align*}
\]

In the above scheme, formex D represents the circular curved configuration and formex F represents the lemnic configuration. The basifactors b2 and b5 for both the paragenic configurations are chosen to be identical so that the resulting configurations are compatible. The same concept can be extended to create the configuration in Fig 4.119. One may write the last Formian statement in the above scheme as

\[
\text{G} = \text{F} # \text{LAM}(3,-5) | \text{D}
\]

The circular paragenic configuration is reflected to fit on either ends of the lemnic configuration. The plot of formex G is the configuration in Fig 4.119.

Two more examples illustrating the use of paragenic transformations are shown in Figs 4.120 and 4.121. Both these configurations relate to the first set of paragenic transformations in Table 4.2. To begin with, the configuration in Fig 4.120 may be used as a trial shape for a membrane structure or a shell structure.
Fig 4.118

Fig 4.119
consisting of six doubly curved surfaces. These surfaces are obtained by first creating six interconnecting cylindrical configurations and placing them parallel to the z-axis at a certain distance from the origin and then transforming them spherically. The formex formulation for the configuration may be written in terms of a scheme as shown below

\[
\text{MEMBRANE} = ':::
M1 = RINIT(10,10,2,2) | ROSAT(1,1) | [10,0,0;10,2,0]
M2 = VERAD(0,0,-90) | BC(1,9,1) | M1
M3 = RIN(2,6,20) | REF(1,10) | M2
M = BS(1,360/120,90/20) | TRAN(1,10) | M3'
\]

The plot of formex M is shown in Fig 4.120. Similar concepts have been used to create the configuration in Fig 4.121. This may be regarded as a hybrid space structure. The five doubly curved surfaces with a ternate pattern can represent membrane surfaces which are supported between five arches with paragenic cross-sections. The formex formulation for the configuration has been written in terms of a scheme as shown below

\[
\text{HYBRID2} = ':
F1 = RINIT(10,10,2,2) | [[10,0,0;10,2,2] # ROSAT(1,1) | [10,0,0;10,2,0]}
F2 = BC(1,180/20,1) | F1
S1 = RIN(3,10,2) | ROSAS(11,1) | [[10,0,0;12,0,0],[10,0,0;11,-1,1]]
S2 = S1 # RIN(3,9,2) | [11,-1,1;11,-1,3]
S3 = TRAN(1,10) | RIN(2,5,22) | REF(1,10) | VERAD(0,0,-90) | S2
S = BS(1,360/110,90/20) | S3
A3 = VERAD(0,0,-90) | F2
A4 = REF(1,10) | A3
A5 = BS(1,360/110,90/20) | TRAN(1,10) | RIN(2,5,22) | A4
A = S # A5'
\]
The trussed arches are represented by formex S and the membrane surfaces are represented by formex A5. The plot of formex A represents the configuration in Fig 4.121.

The examples in this Chapter bring out the wide variety of curved shapes and forms which can be conveniently generated using paragenic configurations. Eight new families of paragenic retronorms have been introduced which may be used for creating configurations for shells, membranes and pneumatic structures. The main effects were the lemnic and peric effects which could be used for different surfaces as well as in combination with other paragenic surfaces. Using the concept of paragenic transformations, configurations may be created and manipulated simply by changing the parameters in a paragenic function. This approach may be used to extend the variety of shapes for space structure configurations.
CHAPTER FIVE

CONCLUSIONS AND PROPOSALS FOR FUTURE WORK

5.1 INTRODUCTION

The concepts of formex algebra have provided a suitable medium for configuration processing throughout the present research. It is shown that although formex configuration processing is a young discipline, there has been enough accumulation of experience in the field to allow the development of various strategies, techniques and principles. The objective of this Chapter is to present the conclusions which have emerged as a result of the research conducted so far. Some ideas and thoughts along the main avenues for possible future research have also been discussed.

5.2 CONCLUSIONS

When designing a space structure, an architect begins by mentally visualising a configuration and preparing conceptual sketches. At this stage, if the design involves a complex structural pattern, there is also a need to communicate with an engineer to discuss various structural schemes and come up with an ideal solution. Formex algebra provides a platform for communication between designers as well as a medium for communication between designers and computers.

In situations where a repetitive type of structural pattern is required, an architect need not create a detailed drawing but a sketch of a small part of the
configuration is enough to act as a guideline for writing a formex formulation for the structure. Designers would normally examine a number of possibilities before settling for a solution in which case formex algebra and Formian serve as convenient means for generating and manipulating structural forms.

Although formex algebra has a well defined terminology and notation, a user who is not familiar with these ideas is bound to find a little difficulty in following the new terms. Fortunately, this problem is short lived and the initial effort is rewarded by the ability to deal with configuration processing with confidence.

Throughout the present work the formex approach is used in relation to interconnection patterns of structural systems. However, the generation of information relating to any other aspect of a structural system may be handled in a similar manner. For example, one may write formices describing the positions of applied loads and support arrangements for a structural system. During a design process, the formex formulation that represents the configuration of a structure, the associated support arrangements and loading cases may be subjected to the same changes that affect the changing ideas of the designer. At every given point the updated formulation provides complete information about the latest particulars of the structural system being designed. This is because the formex formulation can be modified as easily as a text can be modified using a word processor. With the configuration processing tool in hand, an architect's preliminary sketch can be transformed into a precise numerical model.

In the present Thesis the emphasis is on convenient techniques for generating patterns and curved configurations. The problem of pattern generation is approached separately from the problem of generating different shapes and forms. This is because of the manner in which configurations are conceived.
For example, a designer trying to evolve a barrel vault or a dome structure will most probably begin by imagining a sort of pattern on a curved surface. It is very unlikely that the designer may have an idea of the complete pattern together with an exact definition of the surface.

While investigating various possibilities for a suitable configuration, a designer may begin by examining different patterns. The concept of "protomorphs", as put forth by the author, provides a valuable aid in the search for patterns. A protomorph is a basic pattern that recurs frequently in a particular field of application. Protomorphs for single layer, double layer and multilayer configurations are used throughout this study.

In the context of structural configurations, protomorphs represent patterns involving cable, bar or beam elements or finite elements for modelling of plate, shell or membrane structures. A set of protomorphs is used as an "alphabet" for pattern generation. Conceptually, protomorphs are patterns of infinite extent. In any particular situation, it is only natural that protomorphs appear in relation to surfaces or actual physical objects where they have definite shapes and boundaries.

Awareness of the idea of protomorphs encourages a systematic approach in pattern finding. Thus one may begin by concentrating on a simple pattern or a combination of simple patterns that can provide a suitable basis for a target configuration. Patterns may then be subjected to modifications or deformations as appropriate. Catalogues of protomorphs for particular applications are of value since they act as helpful aide-memoirs. Separating the process of decision making of the pattern from that of the details of the surface, simplifies the design procedure and provides more scope for manoeuvre.

When dealing with different classes of space structures, the designer needs a
considerably large variety of shapes and forms. In an attempt to find a solution to this problem, the author’s contribution lies in introducing the concept of "paragenic transformations". There are a number of well established formex transformations, including cylindrical and spherical retronormic transformations for creation of curved surfaces. In addition, paragenic transformations provide a rich repertoire of useful and interesting shapes. These paragenic surfaces are defined through composite functions involving multiple usage of cylindrical and spherical transformations.

Cylindrical and spherical normats are placed systematically in different positions parallel to each of the axes in the global coordinate system and then subjected to a second cylindrical or spherical transformation. This results in a number of families of retronorms which can be used to create a variety of curved configurations. Parameters involved in the transformations control various aspects of the resulting retronorms. Therefore, one may easily change these parameters to obtain different retronorms. Examples in this study illustrate how paragenic forms can be applied to a wide variety of space structures such as, vaults, grids, domes, cable nets, membranes and shell surfaces.

Another advantage of using paragenic transformations is that the designer need not specify the exact mathematical equations for the curves which would be notoriously difficult. In fact, paragenic surfaces may be described conveniently with the help of paragenic functions making data generation of complex surfaces a convenient task.
5.3 FUTURE WORK

Although a variety of space structure patterns for single layer, double layer and multilayer configurations are generated throughout the present research, the study of patterns is still in its initial stages. Especially in the area of multilayer patterns there is a need for protomorphs which may act as a basis for generating other multilayer configurations. The study conducted for dwellings inside spaceframes opens up an entire field of research to explore patterns that are compatible not only as structural frameworks but also as habitable spaces.

Paragenic transformations studied so far involve the use of cylindrical and spherical retronormic transformations. However, this may be regarded as the first generation of retronormic transformations involving cylindrical and spherical transformations. A variety of possibilities still remain to be explored if additional cylindrical or spherical transformations are added to the present paragenic transformations. This may give rise to second and third generation paragenic retronorms.

In the paragenic transformations investigated in this study, cylindrical and spherical normats were placed parallel to each of the axes in the global coordinate system before they were subjected to a second retronormic transformation. Another area of research could involve the effects of placing cylindrical and spherical normats in intermediate positions where they lie in between the axes of the global coordinate system before they are subjected to a second cylindrical or spherical transformation. This may result in families of new and interesting retronorms. Future research would involve a thorough investigation of the potentials of different classes of paragenic transformations in these areas.

A number of doubly curved surfaces have been created using paragenic
transformations. A proposal for future work in this area would be to connect different surfaces to the supporting boundaries. For instance, consider the tent shaped configuration in Fig 5.1. The circular boundary of the membrane is required to be connected to a rectangular boundary as shown in the figure. In another instance, two saddle shaped surfaces may be required to fit together as shown in Fig 5.2. These problems may be overcome by exploring suitable techniques in future research work.
Fig 5.1

Fig 5.2
REFERENCES

CHAPTER ONE


CHAPTER TWO


CHAPTER THREE


