COMPUTER AIDED DESIGN
OF BRACED DOMES

BY

DIMITRA CHRISTOS TZOURMAKLIOTOU

A THESIS SUBMITTED FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF SURREY
DEPARTMENT OF CIVIL ENGINEERING
SPACE STRUCTURES RESEARCH CENTRE
JUNE 1993
ABSTRACT

Domes are one of the oldest structural forms and have been used since the earliest times. The present work is concerned with a systematic investigation of the problem of configuration processing for grid domes and geodesic forms. To deal with the problem of data generation for grid domes a special transformation is evolved. This enables configurations to be projected on different surfaces, such as spheres, ellipsoids, paraboloids, hyperbolic paraboloids and cylinders. The transformation is referred to as the "tractation retronorm". To explore the range of possibilities of shapes and forms four types of projections have been used. These are central, parallel, axial and radial projections. The tractation retronorm allows a grid dome to be generated from a concise and yet readily understood formulation.

Generation of geodesic forms is achieved in two stages. Firstly, a transformation called the polyhedron function is used to generate a configuration modelled on a polyhedron. The resulting configuration is referred to as a "polyhedric configuration". The polyhedron function constitutes the kernel of the problem handling strategy for the configuration processing of geodesic forms. In the next stage the tractation retronorm is employed to obtain the projection of the polyhedric configuration on one or more surfaces. The concepts of formex algebra and its associated programming language Formian have been used together with the above ideas to deal with the configuration processing of braced domes.

The material of the Thesis is organised as follows: Chapter 1 contains a history of the development of domes and a review of recent achievements in this area worldwide. Chapter 2 contains a description of the configuration processing techniques used in the present work. The basic concepts of formex algebra and Formian are described in this chapter. In Chapter 3, fundamental concepts of the configuration processing of grid domes are established. A number of examples are provided to illustrate the various possibilities for grid domes. Chapter 4 deals with the configuration processing of polyhedric and geodesic forms. Finally, Chapter 5 presents the conclusions of the work.
Στους γόνεις μου
και στον αδερφό μου
ACKNOWLEDGEMENTS

My sincere gratitude is due to my supervisor, Dr H Nooshin, for his dedicated and encouraging guidance throughout the development of my research work. My admiration for his clarity of mind, creative thinking and ability to stimulate positive actions.

I would like to thank the Commission of the European Community which made it possible for me to study at the University of Surrey. I also wish to extend my thanks to Professor E G Galousis and lecturer C N Kalfas for their help in coming abroad for further studies; to C Yamamoto, P Disney and D Hadker for their helpful advice on various occasions. Thanks are due to my friends, Nicolas, Stavros, George, Inaki, Vania, Chris, June and Ane for their love and support.

Special thanks to Babis for his support and in particular to my parents for being strong and helping me through during difficult moments.
CONTENTS

ABSTRACT .................. 2
ACKNOWLEDGEMENTS ........ 4

CHAPTER ONE  DOMES FROM ANTIQUITY TO PRESENT AND A REVIEW OF RECENT ACHIEVEMENTS

Introduction ................ 10
1.1 Development of Various Types of Braced Domes ........ 17
1.1.1 Ribbed Domes .......... 19
1.1.2 Schwedler Domes ....... 19
1.1.3 Lamella Domes ........ 19
1.1.4 Grid Domes .......... 20
1.1.5 Geodesic Domes ....... 20
1.1.5.1 Geometry of Geodesic Domes ........ 23
1.1.5.2 From Polyhedra to Geodesic Domes ........ 23
1.2 Further Examples of Domes World-Wide ........ 24
1.3 Design of Braced Domes .... 26
1.4 Analysis of Braced Domes .... 27
1.5 Dead Weight of Braced Domes .... 28

CHAPTER TWO  CONFIGURATION PROCESSING

Formex Algebra and Formian: An Introduction ........ 30
2.1 Elements of Formex Algebra .................. 31
2.1.1 Equality of Formices ........ 34
2.1.2 Variants of a Formex ........ 35
2.1.3 Sequations of a Formex ........ 35
2.1.4 A Prolate Formex ........ 36
2.1.5 Formex Composition ........ 36

5
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 Formex Functions</td>
<td>38</td>
</tr>
<tr>
<td>2.2.1 Tendial Functions</td>
<td>43</td>
</tr>
<tr>
<td>2.2.2 Provial Functions</td>
<td>52</td>
</tr>
<tr>
<td>2.2.3 Introflection Functions</td>
<td>59</td>
</tr>
<tr>
<td>2.2.4 Relection Functions</td>
<td>62</td>
</tr>
<tr>
<td>2.2.5 Libra Function</td>
<td>66</td>
</tr>
<tr>
<td>2.2.6 Brevic Notation</td>
<td>70</td>
</tr>
<tr>
<td>2.3 Formex Graphics</td>
<td>73</td>
</tr>
<tr>
<td>2.3.1 Standard Retronorms</td>
<td>82</td>
</tr>
<tr>
<td>2.3.2 Plotting Styles</td>
<td>89</td>
</tr>
<tr>
<td>2.3.3 Retrobases and Probases</td>
<td>93</td>
</tr>
<tr>
<td>2.4 Node Numbering</td>
<td>96</td>
</tr>
<tr>
<td>2.4.1 Concept of a Catena</td>
<td>97</td>
</tr>
<tr>
<td>2.4.2 Dictum and Redictum Functions</td>
<td>98</td>
</tr>
<tr>
<td>2.4.3 Sevation Functions</td>
<td>106</td>
</tr>
<tr>
<td>2.4.4 Novation Functions</td>
<td>108</td>
</tr>
<tr>
<td>2.4.5 Collation Functions</td>
<td>111</td>
</tr>
<tr>
<td>2.4.5.1 Introduction</td>
<td>111</td>
</tr>
<tr>
<td>2.4.5.2 Concept of Rapportance</td>
<td>112</td>
</tr>
<tr>
<td>2.4.5.3 Process of Rapportation</td>
<td>113</td>
</tr>
<tr>
<td>2.4.5.4 Raported Sequation Functions</td>
<td>116</td>
</tr>
<tr>
<td>2.4.5.5 Raported Variant Functions</td>
<td>117</td>
</tr>
<tr>
<td>2.4.6 Sedilate Functions</td>
<td>118</td>
</tr>
<tr>
<td>2.4.7 Medulla Functions</td>
<td>125</td>
</tr>
<tr>
<td>2.5 FORMIAN: The Programming Language of Formex Algebra</td>
<td>127</td>
</tr>
<tr>
<td>2.5.1 Primitive Constituents</td>
<td>127</td>
</tr>
<tr>
<td>2.5.2 Constants</td>
<td>129</td>
</tr>
<tr>
<td>2.5.3 Assignment Statements and Variables</td>
<td>131</td>
</tr>
<tr>
<td>2.5.4 Formex Functions</td>
<td>132</td>
</tr>
<tr>
<td>2.5.5 Operations and Expressions</td>
<td>133</td>
</tr>
</tbody>
</table>
2.5.6 Schemes .......................... 140
2.5.7 Information Transfer Statements .......................... 144
2.5.8 Organisation Statements .......................... 147
2.6 Formian Graphics .......................... 150
2.7 Data Submission in Formian .......................... 178
2.7.1 Submission Plenix .......................... 181
2.7.2 Element Panel .......................... 182
2.7.3 Support Panel .......................... 183
2.7.4 Load Panel .......................... 184
2.7.5 Concept of an Album .......................... 184
2.7.6 Codet Album .......................... 185
2.7.7 Plenix Codet .......................... 186
2.7.8 Title Codet .......................... 186
2.7.9 Operation Codet .......................... 186
2.7.10 Destination Codet .......................... 187
2.7.11 Structure Codet .......................... 188
2.7.12 Analysis Codet .......................... 188
2.7.13 Induction of the Scheme .......................... 188
2.7.14 Output of Information .......................... 189
2.7.15 Concept of Folio .......................... 191
2.7.16 Output of Analytical Results .......................... 193
2.7.17 Current Album .......................... 198
2.7.18 Current Load Case .......................... 199

CHAPTER THREE TRACTATION RETRONORM

Introduction .......................... 200
3.1 An Example .......................... 201
3.2 Projection Specifier .......................... 203
3.2.1 Central Projection .......................... 204
3.2.2 Parallel Projection .......................... 204
3.2.3 Axial Projection........ 207
3.2.4 Radial Projection........ 212
3.3 Projection Specifier........ 213
3.4 Coefficients of a Surface........ 213
3.5 Selector........ 216
3.6 Sphere........ 216
3.6.1 Intersection of a Sphere and a Straight Line........ 218
3.6.2 Examples for Projection on a Sphere........ 220
3.7 Ellipsoid........ 238
3.7.1 Intersection of an Ellipsoid and a Straight Line........ 238
3.7.2 Examples for Projection on an Ellipsoid........ 242
3.8 Paraboloid........ 254
3.8.1 Intersection of an Elliptic Paraboloid and a Straight Line........ 257
3.8.2 Intersection of a Hyperbolic Paraboloid and a Straight Line........ 260
3.8.3 Examples for Projection on an Elliptic Paraboloid........ 263
3.9 Cylinder........ 269
3.9.1 Intersection of a Circular Cylinder and a Straight Line........ 271
3.9.2 Intersection of an Elliptic Cylinder and a Straight Line........ 274
3.9.3 Intersection of a Parabolic Cylinder and a Straight Line........ 277
3.9.4 Examples for Projection on a Cylinder........ 279
3.10 Plane........ 295
3.10.1 Intersection of a Plane That is Specified by
Three Points and a Straight Line........ 300
3.10.2 Intersection of a Plane That is Specified by Two Points
Defining a Vector Normal to the Plane and a Straight Line........ 302
3.11 Multi Layer Grid Domes........ 303

CHAPTER FOUR Polyhedralic and Geodesic Forms

Introduction........ 306
4.1 The Platonic and Archimedean Polyhedra........ 307
4.2 Polyhedron Function .......... 338
4.2.1 Polyhedron Code .......... 342
4.2.2 Symmetry Operations .......... 342
4.2.3 Basic Procedures for the Creation of the Platonic Polyhedra .......... 348
4.2.4 Basic Procedures for the Creation of the Archimedean Polyhedra .......... 377
4.3 Geodesic Forms .......... 386
4.3.1 An Example .......... 387
4.4 Further Interconnection Functions .......... 392

CHAPTER FIVE CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

Introduction .......... 407

5.1 Conclusions .......... 407

5.2 Suggestions for Future Work .......... 410

REFERENCES .......... 411
Chapter 1

Domes from Antiquity to Present and
a Review of Recent Achievements

INTRODUCTION

Natural forms and structural shapes - mineral, vegetable, animal and even human - have always been adopted and adapted, by man and society, in a rational and aesthetic manner, thus gradually contributing to the development of the scientific and technological achievements of the human mind. In order to adapt himself to his environment, and to mould it to suit his needs, man created, more or less consciously but in rather a haphazard manner, his tools, his shelter, his clothes and his house, using nature as his model. Among the most astonishing creations of nature, an egg is the masterpiece of the most shrewd use of structural material. The exceptional economy of the material of doubly curved shell structures of revolution emphasizes the advantages of this particular form. Ovoidal shapes, however, are only rarely used in civil engineering structures or architecture, and in practice the sphere becomes the
optimal form of pure geometry, (Ref 8).

Nature uses a profusion of various shell forms of single or double curvature, these natural structures provide the inspiration for architects and engineers in their search for novel solutions.

Domes are one of the oldest structural forms and have been used since the earliest times. They enclose a maximum amount of space with a minimum surface and can be very economical in terms of material. The earliest indication of the use of the dome seems to have been given by an Assyrian bas-relief found in the ruins of the palace of Senna-Chezibbo in Nineveh (705-681 BC). This bas-relief shows a group of buildings believed to have been covered with hemispherical and pointed domes. However, though there is little doubt that the dome started only as a utilitarian form of roofing, the domical shape became a religious symbol associated in pagan, early Christian and Islamic periods with tombs, tabernacles, baptisteries, churches, mosques and royal audience halls, (Refs 14 and 15).

A study by Balwin Smith shows that the dome had a special relationship with religious ideas. He proved that the dome is not merely a practical form of masonry, but possesses a special symbolic value which made it the exalted feature of three great styles of architecture Byzantine, Islamic and Indian, (Ref 32)

A key to the origin of the domical shape as a house concept is furnished by the derivation of our modern word "dome" from the Greek and Latin domus. In Middle and Late Latin doma meant "house", "roof" and only at times was used in its modern sense. During the Middle Ages and the Renaissance the word "dome" was used all over Europe to designate a revered house, a Domus Dei. This persistent association with the idea of an important house, which will be seen going back to the beginnings of domical architecture survived in the Italian duomo, the German, Icelandic and Danish Dom, meaning "cathedral". Also, since 1656 in English dome meant town-house, Guild-Hall, State-house or an important Meeting-house in a city, (Ref 32)
The Romans used domes very frequently as the covering for basilicas, garden pavilions, large public baths and mausoleums. The greatest of all Roman domed structures is the Pantheon in Rome, erected between 120-124 AD. Pantheon's dome remained the largest dome ever built for almost 1800 years. Originally it was believed that it had been constructed of solid concrete. However, the latest investigations show that it was actually built of bricks, embedded in thick mortar. The interior of this remarkable roof is a hemisphere and the theoretically high hoop forces which develop around the supports are resisted by a massive concrete base which has a thickness of almost 7m. The Pantheon dome is circular in plan, has a diameter of 44m and is built on a circular substructure, Fig 1.1, [Ref 14].

Building a circular dome over a square or rectangular chamber is a problem which the Romans failed to solve. Byzantine architects successfully surmounted this difficulty by developing the pendentive, Fig 1.2. This is a triangular segment of a sphere, built in masonry and filling in the upper corners of a square or polygonal room to form a circular support for the dome above. One of the earliest and largest examples of this form of construction is in the dome of St. Sophia, Constantinople, Fig 1.3.

The outstanding Byzantine achievement -the church of St. Sophia in Constantinople- was built between 532 and 537 AD. The two known creators of St. Sophia, Anthemios and Isidoros, were both from the West coast of Minor. The building we see today has undergone extensive modifications. In 558 AD, the central dome fell after being subject to two earthquakes, first in August 553 and again in December 557. A nephew of Isidorus then erected the second dome having a higher profile than its predecessor. The form of the second dome remains basically unchanged despite its partial collapse first after an earthquake in the 10th century, and again after another in the 14th century. The main dome is spherical in form and relatively flat rising only 14m for a span of 32.6m. The dome involves four huge pendentives rising more than 18.3m and overhanging by 7.6m. They form a circular springing for the dome, which was constructed of special light-weight tiles varying from
68.6*68.6cm to 61*61cm, each 5.1cm thick and embedded in 5.1cm of thick mortar. Figure 1.4 provides a comparison of the cross section of the Pantheon with that of the St. Sophia (taken across the diagonal illustrated in Fig 1.2). It shows that the lower portion of both domes are similarly massive and provide support to similar, lighter "shells" above and both "shells" are of almost the same span. Hence, it may be inferred that the limiting scale of the first dome of St. Sophia was determined by the Pantheon model. (Ref 16).

A shallow dome of large span exerts tremendous horizontal thrust at the springings and at St. Sophia this was resisted by a system of massive buttresses and semi-domes. In this building, the central shallow spherical dome built from tiles is almost entirely in compression. The horizontal and vertical reactions from the dome are transferred to four pendentives, which in turn carry them to four great arches. These horizontal reactions are absorbed along one axis by two semi-domes and along the other axis by four massive buttresses. Though the span of the dome covering St. Sophia is only 32.6m (less than the 44m clear span of the Pantheon in Rome) the actual impression of the span is much greater, because the church has a greatly increased span due to the semi-domes and the inclined buttresses. The final impression of spaciousness is quite breathtaking. According to the historian Procopius, who described the structure in 560 AD, the four great piers supporting the central dome were built with molten lead as the binding material, (Ref 16).

Several beautiful ribbed domes were built in masonry during the Middle Ages in Italy. Probably the Finest Italian Renaissance dome was designed by the Michelangelo for St. Peter's in Rome. The history of the construction of this dome is quite unusual. The original design was done by Donato Bramante; Raphael replaced him and in turn was replaced by Peruzzi. Part of the structure was built by the next architect Sangallo and Michelangelo became the fifth architect of St. Peter's when he was already 72 years of age. He did not see his work completed. He redesigned the structure and on his death he left a model and drawings which were used by Giacomo della Porta, who finally was able to supervise the completion of the
dome in 1590, following the original drawings and instruction left by Michelangelo, Fig 1.5, (Ref 15)

Della Porta used Michelangelo's wooden model as the basis for his design, keeping the same drum but altering the external profile of the dome. Both designs employed 16 radial ribs connected by an inner and outer shell of brick. On the exterior, Della Porta's dome is very similar to Michelangelo's, except that Della Porta slightly pointed the dome profile and raised the springing of the dome 4.8 meters higher than Michelangelo's project, Fig 1.6.

It has to be remembered that the design of this structure was based on the experience of the masons and the intuition of the designers since no mathematical theory existed at the time. This was developed later, in an effort to explain the cracks which developed in the dome after its construction. Several additional tie rings had to be added in 1744 to the dome to strengthen it and to prevent its collapse, (Ref 15).

A dome is a typical example of a synclastic surface in which the curvature at any point is of the same sign in all directions. Synclastic surfaces are also called surfaces of positive Gaussian curvature and are not developable. That is, domical surfaces cannot be flattened into a plane without stretching or shrinking. This property is one of the reasons why in practice domes cannot be built from members of the same length.

Most domes built in practice have a surface which can be generated by the rotation of a plane curve around a vertical line. The rotating curve is called its meridian and the horizontal sections are known as the parallels. Any curve can be used as a meridian; a circle gives rise to a sphere, an ellipse to an ellipsoid of revolution and a parabola to a paraboloid. The three above mentioned surfaces are all synclastic.

The development of domes is closely associated with the development of available materials. In antiquity domes have been built in stone but brickwork gradually
replaced the stone masonry. Timber was the principal roofing material used in the Middle Ages and some timber domes from this period still exist, mostly in Germany, France, Italy, Russia and Scandinavia. They were often used as an external protective cover over the masonry dome proper. The frameworks of these structures have been picturesquely described as "forests of wood" owing to the complexity of the bracing members.

Masonry domes are no longer used nowadays owing to the very high cost of labour and scaffolding. Reinforced concrete shell domes also have their limitations in spite of the fact that the flexibility and ease of moulding of concrete make it possible to construct any shape. The necessity for complicated formwork to support the reinforced concrete shells during the casting and curing periods makes the reinforced concrete domes rather costly. The use of precast prefabricated units tends to compensate for high labour cost, but various other disadvantages, such as the heavy dead weight and the difficulty of transport, still remain even for this form of construction, (Ref 14).

Iron was first used in dome construction in 1811, when Belanger and Brunet covered the central part of the Corn Market in Paris with an iron dome. Much interest was aroused among the engineers and architects by this construction though it was not much more than an adaptation of the methods of timber construction to the use of wrought iron.

The introduction of steel, with its greatly improved properties of high strength, proved to be a fundamental influence in the development of various types of braced domes and their use for large spans. The largest nineteenth-century wrought-iron dome still in existence is the dome covering the Royal Albert Hall in London. It has an oval plan with axes of 76m and 56m and it was built from 1862 to 1873.
1.1 DEVELOPMENT OF VARIOUS TYPES OF BRACED DOMES

The large spans which could be covered by iron domes stirred the imagination of engineers and the general public alike. The great interest in this form of construction in the 19th century resulted in the development of various types of bracing for such structures. In Germany, France and Switzerland this movement was greatly influenced by the writings of Schwedler (popularly known as the father of domed structures) Henneberg, Mohr, Ritter, Muller-Breslau, Scharowsky and Zimmermann, each of whom contributed greatly to the development of braced domes and the understanding of their structural behaviour.

Most of the early iron domes consisted of a number of lattice truss ribs, usually having a straight lower boom and a curved upper one. They were simply supported on circular walls and connected to a lattice ring girder at the centre. For smaller domes the central ring and lantern were omitted and the trusses were directly connected together. In this form of construction, though the external appearance gives the shape of a dome, the horizontal thrust is virtually non-existent. But in spite of this, an outer horizontal ring of a polygonal shape was usually provided stiffening the whole structure. Many of the areas covered by such systems were square or octagonal. For larger spans the lower booms of the trusses were curved, producing lattice ribbed domes, (Ref 14).

All the earlier domes were hemispherical in shape, producing only vertical reactions on the supports. Later the ratio of the rise to span was decreased, introducing appreciable horizontal trusts at the supports, which often exceeded the capacity of the tie rings. Several cases are known where this led to the collapse of the structure during the erection. This induced many engineers to make a more detailed analysis of braced domes.

The revival of the interest in braced domes and their further development took place after the Second World War, no doubt influenced by the activities of the famous
American "comprehensive designer" Buckminster Fuller. With his geodesic domes he again turned the attention of architects to this very efficient form of construction. The main credit for recent structural development should however go to Lederer, Kiewitt, Soare, Wright, du Chateau, Kadar, Tsuboi, Matsushita and several other distinguished designers who are responsible for the further development of various types of braced domes and the construction of a large number of such structures in many parts of the world.

Makowski classified the braced domes into ten principal types, (Refs 13 and 14):

1. Ribbed domes
2. Schwedler domes
3. Plate-type domes
4. Network domes
5. Zimmermann domes
6. Stiffly jointed framed domes
7. Lamella domes
8. Kiewitt domes - parallel lamella dome
9. Grid domes
10. Geodesic domes

Out of such a large variety of possible types of braced domes, only the following types have proved to be frequently used:

(i) ribbed domes which are frequently constructed in prefabricated tubular arched rib units;

(ii) Schwedler domes introduced in 1863 and remain still popular;

(iii) a lot of braced domes are now being built as three way single-layer grids;
(iv) parallel-lamella domes continue to be used for large spans;

(v) geodesic domes.

1.1.1 RIBBED DOMES

A ribbed dome consists of meridional ribs connected to a number of horizontal polygonal rings, Fig 1.7. If the ribs are directly pin-connected to the foundation, the dome is of the unstiffened type. If however, the ribs are connected at the bottom to the horizontal base ring, the dome is of the stiffened type and the reaction components from the ribs on the ground are only vertical, the horizontal thrust being taken by the base ring.

1.1.2 SCHWEDLER DOMES

A Schwedler dome is a derivation of the ribbed dome. The trapeziums which are created by the intersecting meridional ribs and the horizontal rings are further subdivided by the introduction of one or two diagonal members, Figs 1.8 and 1.9.

The arrangement of bracing members described above, was introduced for the first time by J. W. Schwedler, a German engineer in 1863. During his life time he built a large number of domes of this type, reaching the maximum span of 64m in one of his domes built in 1874 in Vienna.

1.1.3 LAMELLA DOMES

A lamella dome is composed of members forming a diamond or diagonal pattern as shown in Fig 1.10. Also, in some case rings may be added to the configuration as
shown in Fig 1.11. A parallel lamella dome is a cyclically configuration divided into a number of sectors braced by two sets of diagonal members each parallel to one of the main radial ribs, Fig 1.12. Very often latitudinal members are also added to the configuration to create a triangulated pattern.

The lamella system was invented in Europe in 1906 by a Mr. Zollinger, City Architect in Dessau, Germany. The system proved to be exceptionally popular in Germany before the First World War and its use spread rapidly to Sweden, Norway, Holland and Switzerland. It was introduced by Dr. G. P. Kiewitt (parallel-lamella) to the U.S.A. in 1925. Since that time numerous timber lamella structures have been built.

The two largest domes in the world, the New Orleans Superdome (diameter 213m) and the Houston Astrodome (diameter 200m) are examples of the parallel lamella system.

1.1.4 GRID DOMES

A grid dome is obtained by projection of a grid pattern on a surface such as a sphere, ellipsoid, ..., Figs 1.13 and 1.14.

1.1.5 GEODESIC DOMES

A geodesic dome is obtained from projecting the triangulated faces of a regular or semiregular polyhedron on a sphere, or ellipsoid, ..., Figs 1.15 and 1.16.

Geodesic domes have been developed by the American designer Buckminster Fuller (Bucky), who 45 years ago turned the architect's attention to the advantages of skeleton domes. In this type of dome, the bracing members lie on the great circles
<table>
<thead>
<tr>
<th>Structure and date built</th>
<th>Dead weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diameter (m)</td>
</tr>
<tr>
<td>Pantheon dome, AD 120-124</td>
<td>44</td>
</tr>
<tr>
<td>Dome over St. Peter’, Rome 1585</td>
<td>43</td>
</tr>
<tr>
<td>Reinforced concrete dome, Poland, 1912</td>
<td>65</td>
</tr>
<tr>
<td>Houston Astrodome, steel dome, USA, 1965 (The ring girder not included)</td>
<td>200</td>
</tr>
</tbody>
</table>

Fig 1.13
Fig 1.14
Fig 1.15
Fig 1.16
of a sphere, thus following the geodesic lines of the surface.

Numerous geodesic domes have been designed and constructed. At this point, some fundamental geometrical aspects have to be explained with reference to their importance in practical design.

1.1.5.1 GEOMETRY OF GEODESIC DOMES

The geometry of the geodesic domes is based either on regular polyhedra (known as Platonic solids) or on semiregular polyhedra. The semiregular polyhedra were originally described by Archimedes and therefore referred to as the "Archimedean solids". The faces of the polyhedra of this group are equal regular polygons of several classes. They occur everywhere in nature, in one form or another. For example a crystal of common salt is a cube, and the crystalline form of carbon called a diamond, is octahedral (Ref 18).

1.1.5.2 FROM POLYHEDRA TO GEODESIC DOMES

In 1951, Buckminster Fuller filed his initial patent application for a building construction consisting of spherical surfaces divided into triangles. A geodesic line between two points on a sphere is the shortest distance between the points. For the sphere a geodesic line is a part of a great circle.

In Buckminster Fuller's original geodesic domes, the grids were formed on the faces of a spherical icosahedron. As known, an icosahedron exploded onto a surface of a sphere, can be divided into 20 equilateral spherical triangles. It must be noted that this is the maximum number of equilateral triangles into which a sphere can be divided. Each of these triangles can be subdivided into six triangles by drawing medians and bisecting the sides of each triangle. It will be found that the medians follow great circles which are the extensions of the sides of the basic equilateral triangles into which a spherical icosahedron can be divided. Using this method of
construction it is possible to form 15 complete great circles regularly arranged on the surface of a sphere. In this geodesic domes the members forming the framework are usually straight, being the chords of geodesic arcs, \((\text{Ref } 18)\).

For large span domes the primary type of bracing, which is truly geodesic, is not sufficient as it leads to an excessive slenderness ratio of the members which are too long and a secondary bracing has to be introduced. To obtain a more or less regular network of the bracing bars, the edges of the basic equilateral triangles are divided up regularly. The number of modules into which each edge of the spherical icosahedron is divided, depends mainly upon the span of the dome and its cladding. This subdivision is usually referred to as a "frequency". It must be pointed out that during such a subdivision the resulting triangles are no longer equilateral.

Various types of subdivision are used in practice. The basic geodesic subdivision of a sphere results in a triangulated framework. Six triangles interconnected at the same point produce a hexagon. It must be noted, however, that no matter how distorted these hexagons are, the sphere cannot be covered with hexagons only. A minimum of 12 pentagons will have to be introduced, \((\text{Ref } 14)\).

The first experimental geodesic domes were built without recourse to any theoretical analysis. The "geodesic" mathematics refers only to the geometry, but not to the actual method of stress analysis. The introduction of electronic computers and the use of exact mathematical methods of analysis enabled designers to find the exact stress distribution in geodesic domes. Their behaviour can be determined by the same techniques applicable to the other types of skeletal systems. Geodesic domes are also subjected to the same problems of instability as any other form of braced dome.

1.2 FURTHER EXAMPLES OF DOMES WORLD-WIDE

Several attempts have been made in the past to cover large span stadia with retractable roofs. One of the most unusual domes is the famous Pittsburgh public
an auditorium retractable dome erected in 1961. This structure which is an all-season sports arena, exhibition hall and convention centre, is quite remarkable. Its stainless-steel roof, the first structure of its kind ever built, is made of pie-shaped leaves, two fixed and six retractable, the latter capable of being opened or closed in 2.5 minutes. This dome, is nearly circular in plan, has a maximum diameter of 127m and a rise of about 33m. The retractable-dome roof is held by a cantilevered, curved, tripod "space frame" outside the structure. The movable roof is divided radially into eight approximately equal sections which will slide one over another to rest on the top of the two fixed sections, thus opening the huge arena to the sky.

Another interesting example of a retractable stadium is the Toronto Sky Dome erected in 1989. The structure has a clear span of 205m with a rise of 44m. The steel roof consists of four segments, the central two segments are braced barrel sections which will slide to the north to stack over a fixed spherical segment. The southern segment is also spherical and retracts to the north, rotating around a circular track. The roof when fully retracted will be 91% open and takes 20 minutes to open or close. The structure covers 8 acres. It is constructed of steel parabolic arches, with a skin of single PVC membrane on an insulated acoustic steel deck. The stadium has the following seating capacity:

- baseball  53426 seats
- football  55190 seats
- concerts and special events  10000-70000 seats

An interesting feature is the dome of the Sports Palace, built in Mexico City in 1968 for the Olympic Games. This dome has a clear span of 134m. It has the shape of a spherical shell created by a giant two-way grid of steel trusses forming intersecting arches. The square areas between the arches are filled with a triangular grid of aluminium tubes in the shape of single-layer hyperbolic paraboloid. There are altogether 484 hyperbolic paraboloidal units, each with four axes of symmetry.
One must also mention the Indraprastha dome, the third largest dome in the world erected in New Delhi, 1982, for the Asian Games, Asia 1982. The dome, having a diameter of 150m and being the largest clear span structure in the whole India, is the principal building of a large sports complex, built recently.

The dome covering the stadium is provided with a giant suspended soundproof collapsible 100 ton curtain, 150m long and 40m high, enabling the subdivision of the huge stadium into two independent areas. The steel dome of a very unusual design resembles a folded plate structure and consists of primary trusses radiating outwards from a central octagonal compression ring of 40m diameter.

The huge space frame erected recently for the Olympic Games in Barcelona in 1992 will be of great interest, not only because of its size, but especially because of its unusual construction technique.

The steel tubular structure, of nearly trapezoid shape with its sides curved towards the exterior has maximum dimensions of 105.6mx127.8m. The central dome part has approximate dimensions of 80mx57.6m. The perimetral area consists of four toroidal sections connected at four corners by joining cylinders. It is essentially a double-layer space frame of a distorted square grid. The ORTZ connection system has been used by the Spanish civil engineering contractor of the AOSMA. There are 2353 nodes in the whole structure. The erection of this structure followed the Pantadome method, developed by Prof. Mamoru Kawaguchi. The assembly is arranged with the aim of carrying out the maximum on-site work at ground level, reducing to a minimum the use of scaffolding and cranes.

### 1.3 DESIGN OF BRACED DOMES

The practical design of any large dome requires that at least three different loading systems should be considered viz:
a) dead load and snow over the whole dome;

b) dead load and asymmetrical snow load. It is usual to allow for the possibility of a built-up of snow on one side of the dome;

c) dead load and wind load.

Designers of major domes must remember that the use of the usually available wind distribution formulae can be applied only to a preliminary analysis, the final design should be based on wind tunnel tests.

The determination of snow-loading action is extremely difficult and several known examples of collapse of domes show that the failure has been produced by the unsymmetric accumulation of snow, (Ref 15).

1.4 ANALYSIS OF BRACED DOMES

A structural analyst uses numerical, graphical and physical models to represent structural systems. In the numerical approach, when a computer is used, it is necessary to create a digital description of the structural system within the computer. This process is commonly referred to as the process of "data generation". This has been one of the most challenging problems in the use of computers and to make efficient use of the power of modern computers, it is essential that the process of data generation is dealt with conveniently.

To this end, the first step is to find a tool which is suitable for automatic data generation. In this work, the data generating techniques are based on a mathematical system known as formex algebra which is conceived with representation and processing of configurations. Here, the term "configuration" is used in a general sense to refer to a collection of entities of any kind. The data generation for an
engineering analysis can be conveniently accomplished by using formex algebra.

Analysis methods for space structures have been well developed and computer programs are generally available for this purpose. However, before an engineer can design or analyse any structure, he must have a sound qualitative understanding of how the structure works.

The way a braced dome works depends on the configuration of the members. Braced domes which are fully triangulated, will have a high stiffness in all directions at the surface of the dome. These configurations are kinematically stable (no mechanisms when idealized as a pin joined structure.

A dome which is not fully triangulated is kinematically unstable when idealised as a pin joined space structure and may also have widely different stiffness in different directions at the surface of the dome, (Ref 15).

1.5 DEAD WEIGHT OF BRACED DOMES

The dead weight of braced domes depends primarily on the span of the structure, the type of bracing used, the external load acting upon it, the number of supports, the type of covering and, of course, the material in which the structure will be built. Some domes have been designed to support large, concentrated loads, some are only under the action of uniformly distributed loading (Ref 15).

The initial cost of the structure is frequently an important criterion in selecting a particular type of system. A responsible consulting engineer will, no doubt, remind his client that many other factors have to be taken into account - the availability of the material, the ease and speed of erection, the transportation cost and the maintenance cost, which unfortunately is only rarely taken into account at the time of the final decision.
The comparison of the unit weight of domes built in various periods of development is of real interest. Table 1.1 shows clearly that the developments of new structural materials and improvements in constructional technology help to reduce the dead weight of domes, and at the same time allow designers to increase in a substantial way the clear spans of their structures.
Chapter 2

Formex Algebra and Formian

FORMEX ALGEBRA AND FORMIAN: AN INTRODUCTION

The introduction of computers between 1950 and 1960, created a new working environment in all professional activities, specifically in the field of science where the use of more powerful and complicated mathematical methods is now possible. For the civil engineer, the computer has turned into a modern and every day tool. This helps engineers to analyse and design larger and more complicated structures.

Due to these developments and applications, the civil engineers of today have found new working conditions and now one of the main problems for modelling and analysing a large structure is the input of the data, such as coordinates, members, incidences and loading.
Formex algebra is a mathematical system that provides a convenient basis for the solution of problems in data generation and computer graphics. The concepts of formex algebra are particularly suitable for use in computer aided analysis and the design of space structures and are extensively used in this field.

Formian is an interactive programming language which acts as a vehicle for implementing the concepts of formex algebra. The language is designed to provide a structured approach to the problem of data generation and, in particular, to the generation of data related to structural configurations. Being modelled on formex algebra, the language allows statements to be written in a concise and yet readily understood manner. It also has simple to use graphics facilities and a built-in editor, enabling problems of data generation to be accomplished in one programming environment. Also, in Formian the generated data may be stored in the form of a rule thus enabling information about a structural system to be represented in just a few lines of formulation. This allows the data to be modified easily and provides a convenient means of keeping the information for future reference.

2.1 ELEMENTS OF FORMEX ALGEBRA

Consider the configuration given in Fig 2.1. The configuration represent a truss. Now, consider the numerical construct:

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

Such a construct is known as a "formex" (plural formices). A formex is an arrangement of numeric value, commas, semicolons, square brackets and curly brackets. The above formex may represent the encircled part of the truss of Fig 2.1.

The primary components of a formex are enclosed in square brackets and are referred to as "cantles". The number of cantles in a formex is referred to as the "order" of
Fig 2.1

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

Fig 2.2

Fig 2.3
the formex. For instance, the formex of Fig 2.2 contains three cantles and consequently is a formex of the third order.

Each cantle, in turn, consists of a number of "signets" that are separated by semicolons. For instance, each of the cantles in the formex of Fig 2.2 consists of two signets. Graphically, a signet represents a point relative to a reference system. A reference system of the kind shown by the dotted lines in Fig 2.1 is referred to as a "normat". This normat consists of two families of "normat lines" that intersect at "normat points", where each normat line is associated with an identification number as shown in Fig 2.1. The signets in the above formex represent the normat points indicated by little circles in Fig 2.1.

The number of signets in a cantle is referred to as the "plexitude" of the cantle. Thus, the first cantle in the above formex contains two signets and is said to be of plexitude two. The numbers that constitute a signet are separated by commas and are referred to as "uniples". The number of uniples in a signet is referred to as the "grade" of the signet. All the signets in a formex must be of the same grade. For instance the numbers 1 and 1 are the uniples of the last signet of the formex of Fig 2.2. Thus the formex is said to be of the second grade, since it consists of signets of the second grade.

Graphically, each cantle represents a primary component of the truss. For instance, the 1st, 2nd and 3rd cantles of the above formex represent the segments indicated by 1,2 and 3 in Fig 2.3, respectively, and each little circle in Fig 2.3, is described by a signet, that represents a joint of the truss.

A formex of the first order is written without the enclosing curly brackets. For instance,

\[[2,-4; 2,0; 0,-1; 0,-2.5]\]
This is a formex of the first order and the second grade and may represent a four noded finite element. A formex of the first order is also referred to as "maniple". A maniple of the first plexitude is also referred to as "reglet".

A formex may have no cantles, in which case the formex is referred to as an "empty formex" and is denoted by \( \{ \} \). The order of the empty formex is zero but its grade is considered to be arbitrary.

A formex is said to be "homogeneous" provided all its cantles are of the same plexitude and is said to be nonhomogeneous otherwise. A homogeneous formex of the first plexitude is referred to as an "ingot".

For example

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

is an ingot of the fifth order and third grade and

\[
\{[11,10], [12,20], [13,30], [14,40]\}
\]

is an ingot of the fourth order and second grade.

### 2.1.1 EQUALITY OF FORMICES

Two formices are said to be of the same "constitution" provided that they are of the same order and grade and that every cantle in one is of the same plexitude as the corresponding cantle in the other. For instance, formices

\[
\{[23,27.5; 25,29.5], [30,-40], [13,23; 31,34; 42,22]\}
\]

and
Two formices are said to be "equal" if they are of the same constitution and that every uniple in one is equal to the corresponding uniple in the other. Thus,

\{[i,j; m,n], [p,q]\}\{[22,24; -15,-17], [11,13]\}

implies that i=22, j=24, m=-15, n=-17, p=11 and q=13.

2.1.2 VARIANTS OF A FORMEX

Two formices are said to be variants of each other provided that they are of the same constitution and that every cantle in one may be obtained from the corresponding cantle of the other by a rearrangement of the positions of its signets. Two equal formices are considered to be variants of each other. That is, the relationship of equality is regarded as a special case of the relationship of being variants.

For example, if

\[ F_1 = \{[4,6,8; 14,7,5], [9,8; -3,-2]\} \]

and

\[ F_2 = \{[14,7,5; 4,6,8], [9,8; -3,-2]\} \]

then formices \( F_1 \) and \( F_2 \) are variants of each other because the first cantle of \( F_2 \) can be obtained by interchanging the positions of the signets of the first cantle of \( F_1 \).

2.1.3 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 the positions of its cantles. Two equal formices are considered to be sequations of each other. That is, the relationship of equality is regarded as a special case of the relationship of being sequations.

For example, if

\[ E_1 = \{[10,1,2; 4,6,7; -1,-2,4], [22,1; 27,3]\} \]

and

\[ E_2 = \{[22,1; 27,3], [10,1,2; 4,6,7; -1,-2,4]\} \]

then \( E_1 \) and \( E_2 \) are sequations of each other. Also, there could be formices that are sequations of variants of each other.

### 2.1.4 A PROLATE FORMEX

A formex is said to be prolate provided that it contains cantles that are variants of each other.

For example

\[ P = \{[1,2; 4,5; 7,8], [4,5; 1,2; 7,8], [1,1; 3,4]\} \]

is a prolate formex.

### 2.1.5 FORMEX COMPOSITION

If \( F_1 \) and \( F_2 \) are two formices of the same grade, then the composition of \( F_1 \) and \( F_2 \) is defined as a formex \( F \) that consists of all the cantles of \( F_1 \) appearing in the same order as in \( F_1 \), followed by all the cantles of \( F_2 \) appearing in the same order as in
The relationship between $F_1$, $F_2$ and $F$ is written as

$$F = F_1 \# F_2$$

The symbol used to denote this combination is "#". This is known as the "duplus symbol" and the expression $F_1 \# F_2$ is read as "$F_1$ duplus $F_2$". Formices of different grades cannot be composed.

For example if,

$$F_1 = [3, 3; 5, 3]$$

and

$$F_2 = [5, 3; 4, 4]$$

then

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

The formex composition has the following properties:

(a) If $F_1$ and $F_2$ are two formices of the same grade then, in general,

$$F_1 \# F_2 = F_2 \# F_1,$$

but

$$F_1 \# F_2 \text{ and } F_2 \# F_1$$

are sequations of each other. That is, in general, a formex composition is not commutative.

(b) If $F_1$ and $F_2$ are two formices of the same grade then,

$$\text{(}F_1 \# F_2) \# F_3 = F_1 \# (F_2 \# F_3)$$
In other words, a formex composition is associative.

(c) The composition of a formex with the empty formex is the formex itself. That is,

\[ F \# \{\} = \{\} \# F = F \]

2.2 FORMEX FUNCTIONS

In scalar algebra, the equation

\[ y = f(x) \]

indicates that variables \( x \) and \( y \) relate to each other through the function \( f \). The variable \( x \) is referred to as the "independent" variable and \( y \) is referred to as the "dependent" variable.

In formex algebra, if a formex \( F \) may be obtained from another formex \( E \) through a set of rules denoted by, say \( \Phi \), then the relation between \( F \) and \( E \) is written as

\[ F = \Phi \mid E \]

Formices \( E \) and \( F \) are referred to as the "independent" and "dependent" formex variables, respectively, and \( \Phi \) is referred to as a "formex function". The symbol "\( \mid \)" is referred to as the "rallus symbol" and is read as "rallus" or "of".

If the above formex \( E \) is also expressible in terms of \( F \), then the function \( \Phi \) is said to have an "inverse". The inverse of the function \( \Phi \) is denoted by \( \Phi \) and has the property that
It often happens that a particular way of processing a formex arises repeatedly. It would then be convenient to standardise the process by turning it into a function and a number of frequently used formex functions are described in the sequel.

In discussing the formex functions, the following terminology and notation are used:

(1) A composite function obtained from repeated application, say \( r \) times, of a function \( \Phi \) is written as \( \Phi^r \). Thus,

\[
\Phi | \Phi | E \quad \text{is written as } \Phi^2 | E
\]

\[
\Phi | \Phi | \Phi | E \quad \text{is written as } \Phi^3 | E
\]

(2) The zeroth power of any function \( \Phi \) (that is \( \Phi^0 \)) is referred to as the "identity function" and has the property that

\[
E = \Phi^0 | E
\]

Just as in scalar algebra there are functions in formex algebra. A function designator is a construct that represents a value obtained using the rule of a function. For instance, if \( X \) is a number representing an angle, then in normal mathematical notation, one may write

\[
\text{TAN} \ X
\]

to represent the tangent of \( X \). Here TAN is a function and TAN \( X \) is a function designator with \( X \) being referred to as its argument. The designator of a formex function is referred to as a formex function designator.
There exist a variety of formex functions and each one of them has its particular use. At the beginning it seems rather difficult to learn all of them, but as with normal algebra it is easier to start by learning and applying the simple ones and by practice learn to use the more difficult ones. As with all the other fields of algebra and mathematics, only practice can help in understanding and correctly applying all the functions that formex algebra offers.

The idea of a formex function is introduced through an example. Consider the formices

\[
E_1 = \{[1,1; 2,2], [2,2; 1,3], [1,3; 2,4], [2,4; 1,5]\}
\]

and

\[
E_2 = \{[4,1; 5,2], [5,2; 4,3], [4,3; 5,4], [5,4; 4,5]\}
\]

Graphical representations of \(E_1\) and \(E_2\) are shown in Fig 2.4 and are denoted by \(E_1\) and \(E_2\), respectively. A graphical representation of a formex is referred to as its plot and the convention of using a bar over a formex variable to denote its plot will be used henceforth.

It may be noticed that the plot of \(E_2\) may be obtained by translating the plot of \(E_1\) in the first direction by 3 units. An implication of this fact is that the formex \(E_2\) may be obtained from \(E_1\) by adding 3 to the first uniple of every signet of \(E_1\), as may be verified by examining the above formices. The relationship between formices \(E_1\) and \(E_2\) may be expressed by the construct

\[
E_2 = \text{TRANS}(1,3) | E_1
\]
Fig 2.4

Fig 2.5
The construct TRAN(1,3) is a formex function representing a rule for transformation of a given formex E1 into a formex E2. The construct

\[
\text{TRAN}(1,3) \mid E1
\]

is a function designator with E1 being the argument. The parameters 1 and 3 are parts of the rule defining the particulars of the transformation and are referred to as canonic parameters.

The above function is referred to as a translation function and its general form may be written as

\[
\text{TRAN}(h, q)
\]

Thus the formices

\[
E1 = \{[2,1; 3,2], [3,2; 2,3], [2,3; 1,2], [1,2; 2,1]\}
\]

\[
E2 = \text{TRAN}(1,3) \mid E1
\]

\[
E3 = \text{TRAN}(2,3) \mid E2
\]

will give rise to plots shown in Fig 2.5. In this figure, the axis representing the first direction uniples is denoted by U1 and that representing the second direction uniples is denoted by U2. This convention for denoting normal directions will be followed henceforth.
The first and most basic group of formex functions are referred to as "cardinal functions" and are described in Table 2.1. Cardinal functions include the translation function described above. In addition there are rindle, reflection, lambda, vertition, rosette, projection and dilatation functions. The terms translation, reflection, projection and dilatation are used to imply actions that are suggestive of their literal meanings. Rindle is an old English word meaning watercourse. This term is used in formex algebra to refer to translational replication. The term lambda is used to refer to reflectional replication (prompted by the shape of the uppercase Greek letter lambda, that is Λ). The term vertition is used to refer to rotation by a quarter turn. The rotation can be clockwise or anticlockwise depending on the values of h1 and h2. The canonic parameters h1 and h2 define the plane of rotation and the direction of the turn is from the axis given by h1 to the axis given by h2.

The canonic parameters of functions in Table 2.1 which are denoted by h, h1, h2 and s can be only integers. The other canonic parameters, namely p, q, q1 and q2, may be either integer or non-integer.

2.2.1 TENDIAL FUNCTIONS

Tendial functions constitute the commonly used combinations of cardinal functions. There are twenty four tendial functions which are divided into four groups. The first group is referred to as tendid functions and are described in Table 2.2 Each tendid function is seen to consist of a cardinal function acting in the first direction preceded by a cardinal function of the same type acting in the second direction. The name for a tendial function is obtained by adding the suffix ID to the abbreviated name of the corresponding cardinal function. Tendid functions are not defined for vertition and rosette functions.

A construct such as

\[ \text{LAM}(2,q2) \mid \text{LAM}(1,q1) \]
Table 2.1  
Cardinal Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| TRAN(h, q) | Direction of translation is given by \( h \) and amount of translation is given by \( q \). The construct \( \text{TRAN}(h, q) \) is referred to as a "translation function". Example: \( E_1 = \{(1,1;3,1), (3,1;3,2)\} \)  
\( E_2 = \text{TRAN}(1,3)E_1 \) |
| RIN(h, s, p) | Direction of replication is given by \( h \), number of replications (spread) is given by \( s \) and the amount of translation at each step (pace) is given by \( p \). The construct \( \text{RIN}(h,s,p) \) is referred to as a "rindle function". Example: \( E_1 = \{(1,1;3,1), (3,1;3,2)\} \)  
\( E_2 = \text{RIN}(1,2,3)E_1 \)  
\( E_3 = \text{RIN}(2,3,2)E_1 \) |

The formex \( E_2 \) is said to be a translation of \( E_1 \).

The formices \( E_2 \) and \( E_3 \) are said to be a rindle of \( E_1 \).
| Reflection | Direction of reflection is given by $h$ and position of the plane of reflection is given by $q$. The construct $\text{REF}(h,q)$ is referred to as a "reflection function".

Example:
$E_1=[1,1;3,1],[3,1;3,2]]$
$E_2=\text{REF}(1,4)E_1$
$E_3=\text{REF}(2,3)E_1$

![Reflection Diagram]

The formices $E_2$ and $E_3$ are said to be reflections of $E_1$. |
|---|
| Lambda | Direction of reflection is given by $h$ and position of the plane of reflection is given by $q$. The construct $\text{LAM}(h,q)$ is referred to as a "lambda function".

Example:
$E_1=[1,1;3,1],[3,1;3,2]]$
$E_2=\text{LAM}(1,4)E_1$

![Lambda Diagram]

The formex is said to be a lambda of $E_1$. |
Vertition

<table>
<thead>
<tr>
<th>VER(h₁, h₂, q₁, q₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directions given by the values of h₁ and h₂ define the plane of rotation and the coordinates of the point of rotation are given by the values of q₁ and q₂. The construct ( \text{VER}(h₁, h₂, q₁, q₂) ) is referred to as a &quot;vertition function&quot;.</td>
</tr>
</tbody>
</table>

Example:
\[
E₁ = \begin{bmatrix} 1,1; 3,1 \\ 3,1; 3,2 \end{bmatrix} \\
E₂ = \text{VER}(1,2,4,3) \text{E₁}
\]

The formex E₂ is said to be a vertition of E₁.

![Vertition Diagram](attachment:vertition_diagram.png)

---

Rosette

<table>
<thead>
<tr>
<th>ROS(h₁, h₂, q₁, q₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directions given by the values of h₁ and h₂ define the plane of rotation and the coordinates of the point of rotation are given by the values of q₁ and q₂. The construct ( \text{ROS}(h₁, h₂, q₁, q₂) ) is referred to as a &quot;rosette function&quot;.</td>
</tr>
</tbody>
</table>

Example:
\[
E₁ = \begin{bmatrix} 2,1; 4,1 \\ 4,1; 4,2 \end{bmatrix} \\
E₂ = \text{ROS}(1,2,3,3) \text{E₁}
\]

The formex E₂ is said to be a rotational replication of E₁.

![Rosette Diagram](attachment:rosette_diagram.png)
TABLE 2.1 CONTINUED

<table>
<thead>
<tr>
<th>PROJECTION</th>
<th>DILATATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJ(h, q)</td>
<td>DIL(h, q)</td>
</tr>
</tbody>
</table>

Direction of projection is given by the value of \( h \) and position of the plane of projection is given by the value of \( q \). The construct \( \text{PROJ}(h, q) \) is referred to as a "projection function".

Example:
\[
E_1 = \begin{bmatrix} 1,1; 3,3 \end{bmatrix},
E_2 = \text{PROJ}(1,5)E_1,
E_3 = \text{PROJ}(2,4)E_1
\]

The formices \( E_2 \) and \( E_3 \) are said to be the projections of \( E_1 \).

Direction of dilatation is given by the value of \( h \) and factor of dilatation is given by the value of \( q \). The construct \( \text{DIL}(h, q) \) is referred to as a "dilatation function".

Example:
\[
E_1 = \begin{bmatrix} 1,1; 3,1 \end{bmatrix}, \begin{bmatrix} 1,2; 3,3 \end{bmatrix}, \begin{bmatrix} 3,1; 2,3 \end{bmatrix},
E_2 = \text{DIL}(1,3)E_1,
E_3 = \text{DIL}(2,3)E_1
\]

The formices \( E_2 \) and \( E_3 \) are said to be the dilatations of \( E_1 \).
### Table 2.2  Tendid Functions

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANID(q₁,q₂)</td>
<td>TRAN(2,q₂) TRAN(1,q₁)</td>
</tr>
<tr>
<td>(translation tendid)</td>
<td></td>
</tr>
<tr>
<td>RINID(s₁,s₂,p₁,p₂)</td>
<td>RIN(2,s₂,p₂) RIN(1,s₁,p₁)</td>
</tr>
<tr>
<td>(rindle tendid)</td>
<td></td>
</tr>
<tr>
<td>REFID(q₁,q₂)</td>
<td>REF(2,q₂) REF(1,q₁)</td>
</tr>
<tr>
<td>(reflection tendid)</td>
<td></td>
</tr>
<tr>
<td>LAMID(q₁,q₂)</td>
<td>LAM(2,q₂) LAM(1,q₁)</td>
</tr>
<tr>
<td>(lambda tendid)</td>
<td></td>
</tr>
<tr>
<td>PROJID(q₁,q₂)</td>
<td>PROJ(2,q₂) PROJ(1,q₁)</td>
</tr>
<tr>
<td>(projection tendid)</td>
<td></td>
</tr>
<tr>
<td>DILID(q₁,q₂)</td>
<td>DIL(2,q₂) DIL(1,q₁)</td>
</tr>
<tr>
<td>(dilatation tendid)</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2.3  Tendix Functions

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANIX(q₁,q₂,q₃)</td>
<td>TRAN(3,q₃) TRAN(2,q₂) TRAN(1,q₁)</td>
</tr>
<tr>
<td>(translation tendix)</td>
<td></td>
</tr>
<tr>
<td>RINIX(s₁,s₂,s₃,p₁,p₂,p₃)</td>
<td>RIN(3,s₃,p₃) RIN(2,s₂,p₂) RIN(1,s₁,p₁)</td>
</tr>
<tr>
<td>(rindle tendix)</td>
<td></td>
</tr>
<tr>
<td>REFIX(q₁,q₂,q₃)</td>
<td>REF(3,q₃) REF(2,q₂) REF(1,q₁)</td>
</tr>
<tr>
<td>(reflection tendix)</td>
<td></td>
</tr>
<tr>
<td>LAMIX(q₁,q₂,q₃)</td>
<td>LAM(3,q₃) LAM(2,q₂) LAM(1,q₁)</td>
</tr>
<tr>
<td>(lambda tendix)</td>
<td></td>
</tr>
<tr>
<td>PROJIX(q₁,q₂,q₃)</td>
<td>PROJ(3,q₃) PROJ(2,q₂) PROJ(1,q₁)</td>
</tr>
<tr>
<td>(projection tendix)</td>
<td></td>
</tr>
<tr>
<td>DILIX(q₁,q₂,q₃)</td>
<td>DIL(3,q₃) DIL(2,q₂) DIL(1,q₁)</td>
</tr>
<tr>
<td>(dilatation tendix)</td>
<td></td>
</tr>
</tbody>
</table>
shown in Table 2.2 is referred to as a composite (nested) function. Also, a construct such as

\[ \text{LAM}(2,3) | \text{LAM}(1,4) | [1,1; 3,1] \]

is referred to as a composite (nested) function designator.

The equation

\[ E_1 = \text{LAMID}(2,2) | [1,1; 2,2] \]

provides an example of the use of tendid functions. The plot of \( E_1 \) consists of four line segments that are shown in thick lines in Fig 2.6.

Also,

\[ F_2 = \text{RINID}(5,4,2,2) | E_1 \]

creates a formex whose plot is the whole of the pattern shown in Fig 2.6.

The second group of tendial functions are referred to as tendis functions. There are six tendis functions each of which is similar to one of the tendid functions in Table 2.2, except for their names and the directions of action. A tendis function relates to actions in the first and third directions. A tendis function name is obtained by attaching the suffix \( IS \) to the name of the corresponding cardinal function. Thus

\[ \text{RINIS}(a,b,c,d) \]

is equivalent to

\[ \text{RIN}(3,b,d) | \text{RIN}(1,a,c) \]
Fig 2.6

Fig 2.7
The third group of tendial functions are referred to as tendit functions. There are six tendit functions each of which is similar to one of the tendid functions in Table 2.2. A tendit function relates to actions in the second and third directions. Their abbreviated name is obtained by attaching the suffix IT to the name of the corresponding cardinal functions. Thus

\[ \text{PROJIT}(a,b) \]

is equivalent to

\[ \text{PROJ}(3,b) \parallel \text{PROJ}(2,a) \]

The equation

\[ G1 = \text{ROS}(2,3,1,10) \parallel \{[9,0,10; 9,1,11], [9,0,10; 9,1,10]\} \]

The plot of G1 consists of four line segments that are shown in thick lines in Fig 2.7.

Also,

\[ G2 = \text{RINIT}(4,20,-2,2) \parallel G1 \]

creates a formex whose plot is the whole of the pattern shown in Fig 2.7.

The last group of tendial functions are referred to as tendix functions. There are six tendix functions and they relate to actions in directions one, two and three as defined in Table 2.3 The name of a tendix function is obtained by adding the suffix IX to the name of the corresponding cardinal function.

All the standard functions described so far belong to a family of formex functions referred to as transflection functions. This family has another major group of
functions that are referred to as "provial functions". Provial functions are generalisations of cardinal functions and are described in next chapter.

2.2.2 PROVIAL FUNCTIONS

Sometimes it is useful for the user to be able to generate a desired shape along a direction which is not a cardinal direction. Provial functions are helpful for this task. These are generalisations of cardinal functions. The generalisation exists in the sense that each provial function relates to two or more cardinal directions.

Provial functions generate any shape along a straight line defined by the coordinates of two points on the normat. Provial functions as in the case of tendial functions, are the cardinal functions with an additional suffix specifying the direction of the generation.

For example, take the formex

\[ P_1 = \text{LAM}(2,2)[[1.5,1; 2.5,1], [2.5,1; 3,2], [1.5,1; 1,2]] \]

which represents the configuration shown in Fig 2.8. The formex representing the hexagons shown in Fig 2.9 can be created by entering

\[ P = \text{RINAD}(1.5,1,3,2,5)P_1 \]

The construct

\[ \text{RINAD}(1.5,1,3,2,5) \]

is a provial function. This is a rindle provial function whose replicational effect may be along any direction in plane U1-U2. The suffix AD indicates that the action
involves the cardinal directions 1 and 2. The direction along which replication takes place is specified by the coordinates of the points A and B as shown in Fig 2.9. The canonic parameters 1.5, 1 in the above function are the coordinates of point A and the canonic parameters 3, 2 are the coordinates of point B. The last canonic parameter that is 5 specifies the spread, that is, the number of replications.

There are eight provial functions that are related to cardinal directions 1 and 2. These are referred to as proviad functions and are described in Table 2.4. This table also includes the description of another eight provial functions that relate to the cardinal directions 1, 2 and 3 and are referred to as proviax functions. The suffix AX is used to imply the involvement of cardinal directions 1, 2 and 3.

The canonic parameters A1, A2, A3, B1, B2 and B3 appearing in Table 2.4 are the coordinates of the end points of a vector AB. These coordinates may be integer or non-integer. Vector AB is referred to as the "direction vector", Figs 2.10 and 2.11. Also, in Table 2.4, the canonic parameter s that represents the number of replications (spread) is an integer and canonic parameters p, q and a can be integer or non-integer.

Some of the canonic parameters in Table 2.4 are enclosed in square brackets. These are option brackets. For instance, the canonic parameter p for the rindle functions in Table 2.4 is enclosed in option brackets. This parameter represents the pace, that is, the amount of translation in each step of replication. One has the option of either specifying the pace directly or choosing the direction vector so that its length determines the pace. In formulating the hexagons of Fig 2.9 in the above example, the pace was determined by the length of the direction vector AB. However, one may also formulate the configuration by entering

\[ HS = \text{RINAD}(4, 2, 7, 4, 5, \sqrt{1.5^2 + 1^2})\] \[ P1 \]

The direction of replication in this case is given by the coordinates of points C and
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAND(A1, A2, B1, B2[, q])</td>
<td>Direction of translation is given by vector AB and amount of translation is given by q (or the length of AB, in the absence of q).</td>
</tr>
<tr>
<td>TRANX(A1, A2, A3, B1, B2, B3[, q])</td>
<td>Direction of translation is given by vector AB and amount of translation is given by q (or the length of AB, in the absence of q).</td>
</tr>
<tr>
<td>RINAD(A1, A2, B1, B2, s[, p])</td>
<td>Direction of replication is given by vector AB, number of replications is given by s and amount of translation at each step (pace) is given by p (or the length of AB, in the absence of p).</td>
</tr>
<tr>
<td>RINAX(A1, A2, A3, B1, B2, B3, s[, p])</td>
<td>Direction of replication is given by vector AB, number of replications is given by s and amount of translation at each step (pace) is given by p (or the length of AB, in the absence of p).</td>
</tr>
<tr>
<td>REFAD(A1, A2, B1, B2)</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>REFAX(A1, A2, A3, B1, B2, B3)</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>LAMAD(A1, A2, B1, B2)</td>
<td>Coordinates of centre of rotation in plane U1–U2 are given by A1 and A2. Amount of rotation is given by α in degrees with the sense of rotation being such that when α=90 then U1 is mapped into U2. Absence of α implies α=90.</td>
</tr>
<tr>
<td>LAMAX(A1, A2, A3, B1, B2, B3)</td>
<td>Coordinates of centre of rotation in plane U1–U2 are given by A1 and A2. Amount of rotation is given by α in degrees with the sense of rotation being such that when α=90 then U1 is mapped into U2. Absence of α implies α=90.</td>
</tr>
<tr>
<td>VERAD(A1, A2[, α])</td>
<td>Axis of rotation is given by vector AB and amount of rotation is given by α in degrees. The sense of rotation is such that when α is positive then the rotation causes a right-handed screw to move from A towards B. Absence of α implies α=90.</td>
</tr>
<tr>
<td>VERAX(A1, A2, A3, B1, B2, B3[, α])</td>
<td>Axis of rotation is given by vector AB and amount of rotation is given by α in degrees. The sense of rotation is such that when α is positive then the rotation causes a right-handed screw to move from A towards B. Absence of α implies α=90.</td>
</tr>
<tr>
<td>ROSAD(A1, A2[, s, p])</td>
<td>Coordinates of centre of rotation in plane U1–U2 are given by A1 and A2. Number of replications (spread) is given by s and amount of rotation at each step (pace) is given by p in degrees. Sense of rotation is as defined for vertion proviad function (with p instead of α). Absence of s and p implies that s=4 and p=90.</td>
</tr>
<tr>
<td>ROSAX(A1, A2, A3, B1, B2, B3[, s, p])</td>
<td>Coordinates of centre of rotation in plane U1–U2 are given by A1 and A2. Number of replications (spread) is given by s and amount of rotation at each step (pace) is given by p in degrees. Sense of rotation is as defined for vertion proviad function (with p instead of α). Absence of s and p implies that s=4 and p=90.</td>
</tr>
<tr>
<td>PROJAD(A1, A2, B1, B2)</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>PROJAX(A1, A2, A3, B1, B2, B3)</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>DILAD(A1, A2, B1, B2, α)</td>
<td>Direction of dilatation is given by vector AB and factor of dilatation is given by α.</td>
</tr>
<tr>
<td>DILAX(A1, A2, A3, B1, B2, B3, α)</td>
<td>Direction of dilatation is given by vector AB and factor of dilatation is given by α.</td>
</tr>
</tbody>
</table>
D Fig 2.9. Here the distance CD is not equal to the required pace and therefore the pace is directly specified by the last canonic parameter.

As another illustration of the use of provial functions consider the configuration shown in Fig 2.12. A formex representing the part shown in thick lines may be created by entering

\[ E = [1,1;3,1;1,3] \]

A formex representing the whole of the star configuration may be created by entering

\[ F = \text{ROSAD}(3,3,8,45)|E \]

The first two canonic parameters of the above ROSAD function specify the point of rotation and the next two canonic parameters are the optional parameters that specify the number of replications and the angle of rotation at each step of replication.

The cardinal vertition and rosette functions also have generalised versions where it is possible to specify the angle of rotation and the number of replications. The general form of these functions may be written as

\[ \text{VER}(h_1,h_2,q_1,q_2[,a]) \]

\[ \text{ROS}(h_1,h_2,q_1,q_2[,s,p]) \]

Here \( h_1, h_2, q_1 \) and \( q_2 \) are canonic parameters as described in Table 2.1 and \( a, s \) and \( p \) are optional parameters as described in Table 2.4.

There are altogether four groups of provial functions and Table 2.4 covers two of these groups. The remaining groups are referred to as provias and proviat functions. There are eight provias functions each of which is similar to one of the proviad
Fig 2.12

Fig 2.13

Fig 2.14
functions. There are, however, two differences between a provias function and each corresponding proviad function. Firstly a provias functions relates to cardinal directions 1 and 3 rather than 1 and 2. Secondly, the suffix for a provias function is AS rather than AD. Proviat functions are similar to proviad and provias functions but they relate to cardinal directions 2 and 3 and are identified by the suffix AT.

2.2.3 INTROFLECTION FUNCTIONS

The next family of formex functions to be discussed are referred to as introflection functions. These allow formices to be curtailed in various ways. The first member of this family is called the pexum function and is described through an example.

Consider the configuration shown in Fig 2.13. This present part of a structure consisting of a number of line elements connected together at joints. A formex formulation for this configuration may be written as

\[ F = \text{RINID}(4,2,2,2)|A \]

where

\[ A = \text{ROS}(1,2,2,2)|[1,1;3,1] \]

This is a convenient formulation for the configuration of Fig 2.13 but it involves a problem. Namely, all the elements, except the boundary elements, that are parallel to the 2nd direction and to the 1st direction are doubly represented. One could have avoided this problem by using a different formex formulation. However, one can also eliminate the problem easily through the use of a function as explained below. In fact, there are many situations when one can simplify a formex formulation, by allowing superfluous parts to be generated at an intermediate stage of formulation with unwanted parts "pruned" at a later stage.

The pruning of the superfluous elements in the above example may be achieved by
The next group of introflection functions to be discussed are referred to as rendition functions. There are six rendition functions each of which represent a rule for curtailing a formex in a particular manner as explained in Table 2.5. It is convenient to describe the effect of the rendition functions graphically and this approach is adopted here.

The formices E and F referred to in Table 2.5 are given by the constructs

\[ E = RINID(4,4,2,2) \upharpoonright ROS(1,2,2,2) \upharpoonright \{[1,1; 3,1] \cup [1,1; 2,2] \} \]

\[ F = RINID(3,2,2,2) \upharpoonright LAMID(3,4) \upharpoonright [3,3; 2,4] \]

with their plots shown in Figs 2.15 and 2.16.

To interpret the descriptions of Table 2.5, one has to imagine that the plots of Figs 2.15 and 2.16 are superimposed. Also, a nodal point of a plot is to be regarded as a point that is a graphical representation of a signet.

The terms luxum, nexum, and pactum are Latin words used in formex algebra to imply disconnected parts, connected parts and coincident parts respectively. Also the
### Table 2.5  Rendition Functions

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUX(F)</td>
<td>Plot of LUX(F)</td>
</tr>
<tr>
<td>COL(F)</td>
<td>Plot of COL(F)</td>
</tr>
<tr>
<td>NEX(F)</td>
<td>Plot of NEX(F)</td>
</tr>
<tr>
<td>CON(F)</td>
<td>Plot of CON(F)</td>
</tr>
<tr>
<td>PAC(F)</td>
<td>Plot of PAC(F)</td>
</tr>
<tr>
<td>COP(F)</td>
<td>Plot of COP(F)</td>
</tr>
</tbody>
</table>

Fig 2.15

Fig 2.16
prefix "co" is used to imply the "complement of".

The argument of a pexum or a rendition function is a formex and so is the canonic parameter of a rendition function.

Introflection functions involve comparisons of uniples of formices for equality. Such comparisons may be carried out without any problem provided that the formices involved are of type integer, as in the above examples.

Formices created though provial functions are usually of the non-integer type. In some cases, one need not be conscious of the type of formices and may deal with them regardless of their type.

However, there are situations where the type of a formex has a bearing on the manner in which it is used. For instance, if a non-integer formex is to appear as the argument of the pexum function then the result is unlikely to be correct unless a suitable tolerance is used. The reason is that the pexum functions relies on comparison of the uniples of a formex for equality. Thus, if the uniples are non-integer numbers then their checking for equality is not meaningful unless an appropriate tolerance is considered. This problem is discussed in a later chapter.

The transflection and introflection functions described so far cover a substantial proportion of standard formex functions. However there a number of other standard formex functions in formex algebra and some of these will be described in the next chapter.

2.2.4 RELECTION FUNCTIONS

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 in an unambiguous manner. That is,
P with respect to a cantle of E is either true or false. Let a formex G be obtained from E by examining the cantles of E, proceeding in the natural order, and deleting every cantle for which the condition P is false.

The rule by which E is transformed into G is symbolized in terms of a function. This function is denoted by

\[ \text{REL}(P) \]

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

\[ G = \text{REL}(P) \mid E \]

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

For example, consider the formex

\[ F = \{(1,1,2; 3,1,2), (1,0,2; 2,4,1), (8,6,1; 7,5,1)\} \]

and let P be specified as follows:

\[ P \text{ is true provided that the third uniple of the first signet is equal to the third uniple of the second signet and } P \text{ is false otherwise.} \]

The relection of F with respect to P is found to be

\[ \text{REL}(P) \mid F = \{(1,1,2; 3,1,2), (8,6,1; 7,5,1)\} \]

A condition of the type used in conjunction with relection functions is referred to as a "perdicant". In, general, a perdicant is defined as a Boolean function which has
one or more formices as arguments. Thus, the canonic parameter of a relection function is a Boolean entity.

A relection function has the following basic properties:

(1) A relection function has no inverse.

(2) If k is a nonzero positive integer, then

$$REL(P)^k|E = REL(P)|E$$

Conceptually, relection functions are the most general of all introflection functions. In fact, every one of the previously described introflection functions may be written in terms of a relection function. For instance, if

$$G = P|E$$

then this may equivalently be written as

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

where the perdicant P, with respect to a cantle C of E, may be defined as follows:

P is true provided that there is no cantle of E that has an orderate lower than C and is a variant of C and P is false otherwise.

Perdicants for relection functions may always be described in a mixture of mathematical formulae and statements in a natural language. However, in some cases, perdicants may be written in a convenient notation which is discussed in the sequel.
2.2.5 LIBRA FUNCTION

Let $E_i$ denote a formex that is given in terms of the integer variable $i$. For instance, $E_i$ may be given as follows:

$$E_i=\{[i-1,i+1; i+4], [i,2i; -3,-1]\}$$

Let $m$ and $n$ be two integers and let $G$ denote a formex that is the result of the composition of a sequence of $|m-n+1|$ formices that are obtained by substituting for $i$, the value of each integer between $m$ and $n$. More precisely,

- If $n > m$ then $G = E_m \# E_{m+1} \# \ldots \# E_{n-1} \# E_n$
- If $n = m$ then $G = E_m$
- If $n < m$ then $G = E_m \# E_{m-1} \# \ldots \# E_{n+1} \# E_n$

where $E_m, E_{m+1}, \ldots, E_n$ denote the formices that are obtained from $E_i$ by substituting $m, m+1, m+2, \ldots, n-1, n$ for every occurrence of $i$.

The rule through which formex $G$ is obtained from $E_i$ is symbolised in terms of a function. This function is referred to as a "libra composition" and is denoted by

$$\text{LIB}(i=m,n)$$

The integer variable $i$ is referred to as a "libra variable" and the construct $\text{LIB}(i=m,n)$ is read as "libra $i=m$ to $n$". The relation between $G$ and $E_i$ is written as

$$G = \text{LIB}(i=m,n)\{E_i\}$$
For instance if

\[ E_i = \{ [2i-1,1; 2i+1,1], [2i-1,1; 2i,3], [2i+1,1; 2i,3] \} \]

and

\[ E_1 = LIB(i=1,4)|E_i \]

then \( E_1 \) is found to be

\[ E_1 = \{ [1,1; 3,1], [1,1; 2,3], [3,1; 2,3], [3,1; 5,1], [3,1; 4,3], [5,1; 4,3], \ldots \}
\[ [5,1; 7,1], [5,1; 6,3], [7,1; 6,3], [7,1; 9,1], [7,1; 8,3], [9,1; 8,3] \}

Figure 2.23 shows a plot of \( E_1 \).

It is also meaningful to have a formex as the argument of a libra function that does not contain any libra variable. For instance

\[ LIB(i=1,3)|\{ [-1; 2], [-3; 4] \} \]

is a valid construction with the result being

\[ \{ [-1; 2], [-3; 4], [-1; 2], [-3; 4], [-1; 2], [-3; 4] \} \]

The libra functions in the above examples are of a simple type characterised by the fact that the resulting formex is obtained using a single libra function. In fact a formex may be the argument of a sequence of libra functions. The general form of a nested libra function is

\[ G = LIB(i1=m1,n1)|LIB(i2=m2,n2)|\ldots|LIB(ir=mr, nr)|F, \]

where \( r \geq 1 \) and where \( F \) is a formex is precisely in terms of \( i_1, i_2, \ldots, i_r \). The formex \( G \) is found by proceeding from the left and substituting for the libra variable
Fig 2.23

Fig 2.24
in the order $i_1, i_2, \ldots, i_r$.

For instance, if

$$F = \text{LIB}(i=1,2) | \text{LIB}(j=1,2) | [i,j]$$

then $F$ is obtained by first substituting for $i$, which results in

$$\text{LIB}(j=1,2) | [1,j] \neq \text{LIB}(j=1,2) | [2,j]$$

and then substituting for $j$, which gives rise to

$$\{[1,1], [1,2], [2,1], [2,2]\}$$

To illustrate the practical applications of libra functions, consider the configuration shown in Fig 2.24 which represents a finite element mesh consisting of an array of 8 by 6 square elements, where each element has four nodal points situated at its corners.

Let it be required to write a formex describing the interconnection pattern of the mesh in terms of the libra function. Such a formex may be written as

$$F = \text{LIB}(i=1,8) | \text{LIB}(j=1,6) | E_i$$

where

$$E_i = \{[i,j; i+1,j], [i+1,j; i+1,j+1], [i+1,j+1; i,j+1], [i,j+1; i,j]\}$$

represents a typical square element.
2.2.6 BREVIC NOTATION

There is a shorthand notation that may be conveniently employed in writing certain simple types of commonly used perdicants. Namely, those perdicants that are expressible in terms of the uniples of a single formex of the first order (maniple), or pairs of formices of the first order (maniples). The notation is referred to as the "brevic" notation and is described below.

Let Ma and Mb be two formices of the first order which need not be of the same plexitude and of the same grade. With reference to these formices of the first order, the symbols that constitute the brevic notation together with their meanings are given in Table 2.6.

For further discussion, the brevic notation will be used in connection with the relection function. In this context, a perdicant is expressed in terms of a single formex of the first order and this formex taken to be Ma. That is, the letter U is used to signify a uniple.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU</td>
<td>Every uniple of Ma</td>
</tr>
<tr>
<td>AU</td>
<td>Any uniple of Ma</td>
</tr>
<tr>
<td>EU(j)</td>
<td>jth uniple of every signet of Ma</td>
</tr>
<tr>
<td>AU(j)</td>
<td>jth uniple of any signet of Ma</td>
</tr>
<tr>
<td>U(i,j)</td>
<td>jth uniple of the ith signet of Ma</td>
</tr>
<tr>
<td>EW</td>
<td>Every uniple of Mb</td>
</tr>
<tr>
<td>AW</td>
<td>Any uniple of Mb</td>
</tr>
<tr>
<td>EW(j)</td>
<td>jth uniple of every signet of Mb</td>
</tr>
<tr>
<td>AW(j)</td>
<td>jth uniple of any signet of Mb</td>
</tr>
<tr>
<td>W(i,j)</td>
<td>jth uniple of the ith signet of Mb</td>
</tr>
</tbody>
</table>
To exemplify the use of the brevic notation consider the formex

\[ M = [3,2,1; 4,3,2; 3,2,1; 5,4,3] \]

The perdicant

\[ U(1,2) + 1 < U(4,2) \]

is true with respect to \( M \), since \( U(1,2) + 1 \) is equal to 3 and \( U(4,2) \) is equal to 4.

The perdicant

\[ EU(3) \geq 3 \]

is false with respect to \( M \), since the third uniple of fourth signet satisfies the condition but the third uniples of the first, second and third signets do not. The perdicant

\[ AU(1) = 3 \]

is true with respect to \( M \), since the first uniples of the first and third signets satisfy the condition.

The perdicant

\[ EU < 6 \; \text{AND} \; AU = 5 \]

is true with respect to \( M \), since every uniple of \( M \) is less than 6 and, in addition, there is a uniple in \( M \) that is equal to 5.

Further illustration of the use of brevic notation is provided in conjunction with relection functions in the following examples. Consider the configuration of Fig 2.25
Fig 2.25

Fig 2.26

Fig 2.27
and assume that it represents the plan view of a flat grid structure. Let it be required to write a formex formulation for the interconnection pattern of the structure. A convenient way of formulating the pattern is to adopt a superpansive approach by choosing

\[ D_1 = \{[{1,1}; 3,1], [{1,1}; 1,3]\} \# \text{ROS}(1,2,2,2)|[1,1; 2,2] \]

as the basic unit and writing

\[ D_2 = \text{RINID}(11,8,2,2)|D_1 \]

A plot of \(D_1\) is shown in Fig 2.26 and a plot of \(D_2\) is shown in Fig 2.27. Comparison of Figs 2.25 and 2.27 shows that the interconnection pattern represented by \(D_2\) contains some superfluous elements on the right beyond the normat line \(U_1 = 21\) and on the top beyond the normat line \(U_2 = 15\). The cantles of \(F_2\) representing the superfluous elements may be deleted by a reflection function. Thus, one may write

\[ D_3 = \text{REL}(EU(1) < 21 \text{ AND } EU(2) < 17)|D_2 \]

A plot of \(D_3\) would give rise to a configuration identical to the one shown in Fig 2.25.

2.3 FORMEX GRAPHICS

Once a structural shape has been configured in terms of a formex, one may obtain a graphical representation of the formex. This graphical representation is referred to as a "formex plot". A formex plot is obtained by using two types of rules. Firstly, there are rules through which the geometric particulars for representing signets and cantles are determined. An aspect of the rules of this type is referred to as a
"retrocord". Secondly, there are rules through which the signets in a formex are mapped into specific points in a given coordinate system. A set of rules of this type is referred to as a "retronorm".

As an example, consider the formex

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

A plot of A is given in Fig 2.28. The retrocords are given as follows:

1. A signet should be represented by a little circle. This part of a plot which is represented by a signet is referred to as a "tenon".

2. A cantle of the second plexitude should be represented by a straight line connecting the little circles of its signets and should be indicated by an arrowhead placed on the line and where the orderate of cantle should be written within a little square. This part of the plot which represents a cantle is referred to as a "frond". The plot in Fig 2.29 consists of 7 tenons and 8 fronds. A tenon may belong to a frond exclusively, or may belong to a number of fronds simultaneously.

3. A cantle of the third plexitude should be represented by a shaded triangle and a cantle of the fourth plexitude should be represented by a shaded quadrilateral. The order of appearance of the signets in a cantle of the third and fourth plexitude should be indicated by arrowheads placed on its edges and the edge that corresponds to the first and the last signets should not have any arrowhead.

The choice of retrocords specifying the shapes representing signets and cantles is rather arbitrary. A tenon or a frond may be represented in an infinite variety of ways. For instance, a tenon may be represented as a shape resembling the node of a structure or a leaf, while a frond could be represented as a structural member, a trunk of a tree and so on. If the particulars regarding the shape of the tenons or
Fig 2.28

Fig 2.29
fronds are to be turned into instructions for a plotting machine, then in addition to the retrocords that specify the shape, one should have retrocords to specify the actual size, colour and other details regarding the appearance of the tenons and fronds. In reality, the choices for retrocords are governed by the suitability in relation to a particular application.

Every tenon is drawn relative to a point which is referred to as its "pivot". In the plot of A in Fig 2.29, the centres of the little circles are the pivots which are located by specifying their coordinates with respect to a coordinate system using the equations,

\[ X = U_1 \]
\[ Y = U_2 \]

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

\[ X = U_1 + 1 \]
\[ Y = U_1 + \frac{U_2}{2} \]

the resulting configuration, shown in Fig 2.30, will be another plot of A.

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

\[ X = U_2 \]
\[ Y = U_1 \]

The resulting configuration obtained, shown in Fig 2.31, is yet another plot of A.
Fig 2.30

Fig 2.31
Once again consider the formex A, but this time let the pivots be represented in terms of polar coordinates, using the equations

\[ r = U_2 + U_1 \]
\[ \Theta = (U_1 - 2) \pi / 4 \]

The configuration obtained, shown in Fig 2.32, is also another plot of A.

In the above examples, the grade of a formex happens to be the same as the dimensions of the coordinate systems with respect to which the plot of the formex is drawn. In fact, a formex of any grade may be plotted with respect to a one, two or three dimensional coordinate system. For instance, consider the following formex of the second grade:

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

A plot of B with respect to a one dimensional coordinate system may be obtained as shown in Fig 2.33, where the retronorm is given by the equation

\[ X = 2U_1 - U_2 \]

Here a frond is drawn as an arc to avoid overlapping.

The formex B is also plotted with respect to two and three dimensional Cartesian coordinate systems and the resulting plots are shown in Figs 2.34 and 2.35, respectively. The plot of Fig 2.34 is obtained using the retronorm

\[ X = 2U_1 \]
\[ Y = 2U_2 \]
Fig 2.32
Fig 2.34

Fig 2.35
and the plot of Fig 2.35 is obtained using the retronorm

\[
X = 2U1 \\
Y = 2U2 \\
Z = 4 - 2(3 - U1)^2 - 2(2 - U2)^2
\]

Plots of formices of different grades may be obtained using different retronorms. There are six commonly used retronorms which 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 \(x\) is a function of a typical uniple.

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

\[x = f_1(U_1, U_2, \ldots, U_n) \]
\[y = f_2(U_1, U_2, \ldots, U_n)\]

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

\[x = f_1(U_1, U_2, \ldots, U_n) \]
\[y = f_2(U_1, U_2, \ldots, U_n) \]
\[z = f_3(U_1, U_2, \ldots, U_n)\]

(4) A **polar retronorm** relates to a polar coordinate system and is defined by coordinate equations of the form
\[ r = f_1(U_1, U_2, \ldots, U_n) \]
\[ \Theta = f_2(U_1, U_2, \ldots, U_n) \]
\[ \gamma = f_3(U_1, U_2, \ldots, U_n) \]

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

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

(6) A spherical retronorm relates to a spherical coordinate system and is defined by coordinate equations of the form

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

2.3.1 STANDARD RETRONORMS

There are three families of standard retronorms. These are special cases of the above retronorms and are called "pariant retronorms", "basiant retronorms" and "metrian retronorms". The standard retronorms are described in Table 2.7. The first column of this table lists the names of the standard retronorms. The second and third column of Table 2.7 contain the definitions of the retronorms and the corresponding coordinate systems.

The first 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 which are described in Table 2.7.
### Table 2.7: Standard Retronorms

<table>
<thead>
<tr>
<th>NAME OF RETRONORM</th>
<th>RETRONORM</th>
<th>COORDINATE SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>pariunifect</td>
<td>$x = U_1$</td>
<td></td>
</tr>
<tr>
<td>basiunifect</td>
<td>$x = b_1 U_1$</td>
<td></td>
</tr>
<tr>
<td>metriunifect</td>
<td>$x = b_1 \text{met}(U_1,m_1)$</td>
<td></td>
</tr>
</tbody>
</table>
| paribifect        | $x = U_1$
|                   | $y = U_2$ |
| basibifect        | $x = b_1 U_1$
|                   | $y = b_2 U_2$ |
| metribifect       | $x = b_1 \text{met}(U_1,m_1)$
|                   | $y = b_2 \text{met}(U_2,m_2)$ |
| paritrifect       | $x = U_1$
|                   | $y = U_2$
|                   | $z = U_3$ |
| basitrifect       | $x = b_1 U_1$
|                   | $y = b_2 U_2$
|                   | $z = b_3 u_3$ |
| metritrifect      | $x = b_1 \text{met}(U_1,m_1)$
|                   | $y = b_2 \text{met}(U_2,m_2)$
|                   | $z = b_3 \text{met}(U_3,m_3)$ |

- **One dimensional Cartesian Coordinate System**
- **Two dimensional Cartesian Coordinate System**
- **Three-dimensional Cartesian Coordinate System**
<table>
<thead>
<tr>
<th>Coordinate System</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Polar Coordinate System</strong></td>
<td>( r = U_1 ), ( \Theta = b_2 \times U_2 )</td>
</tr>
<tr>
<td>paripolar</td>
<td>( r = b_1 \times U_1 ), ( \Theta = b_2 \times U_2 )</td>
</tr>
<tr>
<td>basipolar</td>
<td>( r = b_1 \times U_1 ), ( \Theta = b_2 \times U_2 )</td>
</tr>
<tr>
<td>metripolar</td>
<td>( r = b_1 \times \text{met}(U_1, m_1) ), ( \Theta = b_2 \times \text{met}(U_2, m_2) )</td>
</tr>
<tr>
<td><strong>Cylindrical Coordinate System</strong></td>
<td>( r = U_1 ), ( \Theta = b_2 \times U_2 ), ( z = U_3 )</td>
</tr>
<tr>
<td>paricylindrical</td>
<td>( r = b_1 \times U_1 ), ( \Theta = b_2 \times U_2 ), ( z = U_3 )</td>
</tr>
<tr>
<td>basicylindrical</td>
<td>( r = b_1 \times U_1 ), ( \Theta = b_2 \times U_2 ), ( z = b_3 \times U_3 )</td>
</tr>
<tr>
<td>metricylindrical</td>
<td>( r = b_1 \times \text{met}(U_1, m_1) ), ( \Theta = b_2 \times \text{met}(U_2, m_2) ), ( z = b_3 \times \text{met}(U_3, m_3) )</td>
</tr>
<tr>
<td><strong>Spherical Coordinate System</strong></td>
<td>( r = U_1 ), ( \Theta = b_2 \times U_2 ), ( \gamma = b_3 \times U_3 )</td>
</tr>
<tr>
<td>parispherical</td>
<td>( r = b_1 \times U_1 ), ( \Theta = b_2 \times U_2 ), ( \gamma = b_3 \times U_3 )</td>
</tr>
<tr>
<td>basispherical</td>
<td>( r = b_1 \times U_1 ), ( \Theta = b_2 \times U_2 ), ( \gamma = b_3 \times U_3 )</td>
</tr>
<tr>
<td>metrispherical</td>
<td>( r = b_1 \times \text{met}(U_1, m_1) ), ( \Theta = b_2 \times \text{met}(U_2, m_2) ), ( \gamma = b_3 \times \text{met}(U_3, m_3) )</td>
</tr>
</tbody>
</table>
The "basiant retronorms" are the second family of standard retronorms. The particulars of these are as given in Table 2.7. 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 the formex

$$D = [4,1; 4,2; 2,2; 5,5; 5,4; 7,4]$$

The plot of $D$ is shown in Fig 2.36 illustrating the use of a basibifect retronorm where

$$b_1 = 20 \text{ unit length}$$

and

$$b_2 = 15 \text{ unit length}$$

Also if a basipolar retronorm with

$$b_1 = 16 \text{ unit length}$$

and

$$b_2 = \pi/8$$

is used it would give rise to the normat, a part of which is shown in Fig 2.37. Thus one may specify different values for each of the basifactors for various retronorms.

The third family of standard retronorms are referred to as "metriant retronorms" as shown in Table 2.7. The terms $b_1$, $b_2$ and $b_3$ appearing in the table are the
Fig 2.36

Fig 2.37
basifactors similar to those described for the basiant retronorms. The terms m₁, m₂ and m₃ which are nonzero positive numbers are refereed to as "metrifactors". The definitions of metriant retronorms involve a particular scalar function called the "metril function". The function met(U₁,m₁) implies acceleration (deceleration) of scaling. To illustrate the use of metriant retronorms, consider the formex

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

a plot of which is shown in Fig 2.38. This is a plot based on a metriunifect retronorm specified by

$$b₁=4 \text{ unit length}$$

and

$$m₁=0.5$$

Also the configuration in Fig 2.39 is a metribifect plot of E for which

$$b₁=1 \text{ and } b₂=1.5$$

and

$$m₁=1.5 \text{ and } m₂=2$$

In Fig 2.38 the intervals between successive normat lines decreases progressively, controlled by the value of m₁. In Fig 2.39, the intervals between successive normat lines increase progressively in the second direction controlled by the value of m₂. If m₁=1 then the length of the intervals remains the same.

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 2.38 and 2.39 may be referred to as basiunifect and basibifect normats.
2.3.2 PLOTTING STYLES

In plotting a formex, one may start with specifying a retronorm and then adopt a set of retrocords in accordance with which the tenons and fronds can be drawn. In practice 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, they are referred to as the "radix", "natural" and "Zygmun" plotting styles. The radix plotting style gives rise to plots that closely reflect the particulars of their respective formices. Thus if

\[
G = \{[1,1; 3,1], [3,1; 2,3], [2,3; 1,1], [2,3; 3,1], \\
[3,1; 5,5], [5,5; 4,7], [4,7; 2,3], [1,1; 2,3], [2,5] \}
\]

then the paribifect R-plot of \(G\)

\[
X = U_1 \\
Y = U_2
\]

will be as shown in Fig 2.40. The orderates of the cantles are also indicated in the plot. A retrocord which specifies an aspect of the radix plotting is referred to as a "radix retrocord". The retrocords used in drawing the plot in Fig 2.40 may be described as:

1. Every signet \([U_1, U_2]\) is represented by a little circle.

2. Every cantle is represented by a straight line joining these circle together with an arrowhead.

3. The order of appearance of signets in the cantle is indicated by an arrowhead.

4. A segment of the plot that involves overlapping parts is represented in displaced
positions.

(5) The graphic representation of a cantle of the first plexitude is a tenon.

The next type of plotting style is the natural plotting style which has been illustrated in the N-plot in Fig 2.41.

The retrocords used in drawing this plot may be described as follows:

(1) The frond of a cantle of the second plexitude is obtained by drawing a straight line to connect the pivots relating to its signets.

(2) No symbol is used for a tenon, except when a tenon represents a cantle, in which case it is drawn as a little solid 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 and in which variations in the forms of the fronds and/or tenons are used to identify the layers from each other. For instance, consider the configuration in Fig 2.42, 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 2.43 where the elements in the top layer are drawn in full lines, the elements in the bottom layer are drawn in broken lines and the web members are drawn in...
dotted lines. In this form of representation, it is much easier to visualise the configuration as compared to the one in Fig 2.42. As an alternative technique, the nodes in different layers may be represented in different styles as in Fig 2.44. One may even use a combination of both the techniques to draw the configuration as shown in Fig 2.45. In general, one may define a Z-plot as a formex plot that presents a multilayer configuration in which different methods of representation are used to identify tenons and/or fronds lying in different layers.

2.3.3 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 2.46. 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.

(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 by the dashed line for triangle 1.

Also, the pronorm is specified graphically in Fig 2.47. 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

\[ F = \operatorname{LIB}\{[I=1,4]|\operatorname{RIN}(1,5-I,2)|\operatorname{TRANID}(I,I*2)|Ei \] 

\[ Ei = \{(1,1; 3,1), (3,1; 2,3), (2,3; 1,1)\} \]

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.

2.4 NODE NUMBERING

When a structural system is to be analysed by a digital computer, it would be necessary to prepare the description of the system providing information about the interconnection pattern, geometric particulars, material properties, external loads and support conditions. These items of information constitute the "data" which is to be used in conjunction with a computer program.

Formex algebra may be used in various ways to overcome the difficulties of data preparation. In particular, the information regarding the interconnection pattern of
a structural system may be conveniently formulated through the concepts of formex algebra. Furthermore, the disposition of external loads and supports may be described using formex formulations.

A commonly used technique for the description of the interconnection pattern of a structural system consists of identifying the nodal points of the system by a sequence of natural numbers and specifying the interconnection pattern in terms of these numbers. Node numbers may also be used to describe the positions of the external loads and supports.

### 2.4.1 CONCEPT OF A CATENA

Consider a formex $E$ and let $T$ be an ingot of the same grade as $E$. The ingot $T$ is said to be a "catena" of $E$ provided that for every chosen signet $S$ of $E$ there is at least one signet in $T$ that is equal to $S$.

For example, if

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

$$T_1 = \{[1, 1], [3, 1], [4, 3], [6, 5]\}$$

and

$$T_2 = \{[7, 5], [3, 1], [4, 3], [-1, -2], [1, 1], [6, 5], [-1, -2]\}$$

then both $T_1$ and $T_2$ are catenas of $E$.

If $T$ is a catena of a formex $E$ then $T$ is said to be an "exclusive catena" of $E$ provided that $T$ is nonprolate and that every signet of $T$ is contained in $E$. For instance, $T_1$ in the above example is an exclusive catena of $E$. 

97
If T is a catena of a formex E and if T does not satisfy the conditions for being an exclusive catena of E, then T is referred to as an "inclusive catena" of E. For instance, T2 in the above example is an inclusive catena of E.

Every nonempty ingot is considered to be an inclusive catena of the empty formex but the only exclusive catena of the empty formex is the empty formex itself.

2.4.2 DICTUM AND REDICTUM FUNCTIONS

Consider a formex E and let N be a catena of E. Let a formex F be obtained from E by replacing every signet S of E by the orderate, with respect to N, of the first occurrence of S in N. The rule through which F is obtained from E is symbolised in terms of a formex function. This function is referred to as a "dictum function" and is denoted by

$$\text{DIC}(N)$$

The formex F is referred to as the "dictum" of E with respect to N and the relation between E and F is written as

$$F = \text{DIC}(N)\mid E$$

The ingot N is referred to as a "numerant". In general, the terms numerant is used to refer to an ingot that acts as a mapping associating its signet with their respective orderates, acting as the node numbers of a structural system.

For instance, if

$$E = \{[3,2; 4,3; 5,4], [7,5; 5,3], [9,7], 5,4; 9,7]\}$$

and
\[ N = \{9,7, [0,0], [5,3], [1,1], [3,2], [7,5], [2,2], [5,4], [4,3]\} \]

then

\[ \text{DIC}(N) | E = \{5; 9; 81, [6; 31, 1, [8; 1]\} \]

The dictum of the empty formex with respect to any ingot is the empty formex and so is the dictum of any formex with respect to the empty formex. That is, for any formex \( F \) and any ingot \( N \)

\[ \text{DIC}(N) | {} = {} \]

To illustrate the application of dictum functions consider the flat grid whose plan view shown in Fig 2.48. Let a formex representing the interconnection pattern of the grid relative to the indicated pronorm be written as

\[ \text{FG} = \text{LUX}([7,9; 9,9; 11,7; 11,9]) | \text{RINID}(8,6,2,2) | .. \]

\[ ([1,1; 3,3] \# \text{ROSAD}(2,2) | [1,1; 3,1]) \]

Also, let an ingot representing the nodal points of the grid be obtained as

\[ \text{NI} = \text{LUX}([11,7; 7,9; 9,9; 11,9]) | \text{RINID}(8,6,2,2) | \text{ROS}(1,2,2,2) | [1,1] \]

This ingot consists of 59 signets and is an exclusive catena of \( \text{FG} \). The grid of Fig 2.48 whose interconnection pattern is given by \( \text{FG} \) is shown again in Fig 2.49, but this time the orderates of the signets of \( \text{NI} \) are written near the nodal points.

Now, let the dictum of \( \text{FG} \) with respect to \( \text{NI} \) be denoted by \( \text{FG1} \). That is,

\[ \text{FG1} = \text{DIC}(\text{NI}) | \text{FG} \]
Fig 2.48

Fig 2.49
As a second example, let an ingot $N_2$ be obtained as

$$N_2 = \text{LUX}([11,7; 7,9; 9,9; 11,9]) \text{REL}(U1 < 18) \text{RIN}(1,9,2) \text{RIN}(2,7,2) | [1,1]$$

The resulting numbering system will be as shown in Fig 2.50 and the description of the interconnection pattern of the grid with respect to this numbering system is given by

$$\text{JOINTS}_1 = \text{DIC}(N_2) | \text{FG}$$

The formex $\text{JOINTS}_1$ is found to be of the form

$$\{[1; 9], [1; 8], ..., [51; 59], [58; 59]\}$$

Also, let an ingot $N_3$ representing the nodal points of the grid be obtained in terms of

$$N_3 = \text{LUX}([11,7; 7,9; 9,9; 11,9]) \text{REL}(U2 < 14) \text{RIN}(2,7,2) \text{RIN}(1,9,2) | [1,1]$$

The resulting numbering system will be as shown in Fig 2.51. The description of the interconnection pattern of the grid with respect to this numbering system is given by

$$\text{JOINTS}_2 = \text{DIC}(N_3) | \text{FG}$$

In this case, the formex $\text{JOINTS}_2$ is found to be of the form

$$\{[1; 11], [1; 2], ..., [59; 58], [58; 49]\}$$

Now, suppose that the grid is subjected to point loads at the positions indicated by the little solid circles in Fig 2.52. A formex describing the positions of those loaded joints may be written as
LOADS = LUX([11, 5; 11, 7; 7, 9; 9, 9; 11, 9]) | RINID(5, 4, 2, 2) | [5, 5]

A formex describing the positions of the loaded joints in terms of the node numbering scheme of Fig 2.51 may be obtained as

\[ \text{LDS} = \text{DIC(JOINTS2)} | \text{LOADS} = \{21, 22, \ldots, 48\} \]

Using the approach exemplified above one may also be able to obtain formex formulation describing support positions in terms of a node numbering scheme. For instance, suppose that the grid discussed above has two types of supports indicated by the little squares in Fig 2.52, where a solid square is used to indicate a joint which is fully fixed and where a hollow square is used to indicate a joint that has a single vertical constraint. A formex representing the positions of the fully-fixed supports may be written as

\[ \text{SUPPORTS1} = \text{RINID(2, 2, 16, 12)} | [1, 1] \]

A formex representing the positions of the vertically-constrained supports may be written as

\[ \text{SUPPORTS2} = \text{RINID(4, 2, 4, 12)} | [3, 1] \# \text{RINID(2, 2, 16, 4)} | [1, 5] \]

Formices SUPPORTS1 and SUPPORTS2 may be transformed in following the manner:

\[ \text{SPTS1} = \text{DIC(JOINTS1)} | \text{SUPPORTS1} = \{1, 7, 59, 53\} \]

and

\[ \text{SPTS2} = \text{DIC(JOINTS1)} | \text{SUPPORTS2} = \{8, 22, 34, 46, 14, 27, 38, 52, 3, 5, 55, 57\} \]
Formices SPTS1 and SPTS2 describe the support positions in terms of the node numbering scheme shown in Fig 2.50.

Every dictum function has an inverse, where if

\[ G = \text{DIC}(N)|E \]

then

\[ E = \text{DIC}(N)^{-1}|G \]

This relationship may alternatively be written as

\[ E = \text{RED}(N)|G \]

where \( \text{RED}(N) \) is referred to as a "redictum function". That is a redictum function is the inverse of a dictum function. A formex \( G \) appearing as the independent variable of a redictum function

\[ \text{RED}(N) \]

should satisfy the following conditions:

(1) The grade of \( G \) must be equal to unity.

(2) Every uniple of \( G \) must be nonzero and positive.

(3) Every uniple of \( G \) must be less than or equal to the order of the numerant \( N \).

The practical application of redictum function is described in terms of the example of Fig 2.50, where \( FG \) represents the interconnection pattern of the grid relative to
the indicated pronorm and JOINTS1 represents the interconnection pattern of the grid with respect to the ingot N2, that is, with respect to the joint numbering scheme of the Fig 2.50. In using the information about the interconnection pattern of the grid in a computer program, one may proceed by interpreting the cantles of FG, one at a time, as need may arise, in terms of the joint numbering system without actually forming the formex JOINTS1. Thus, one keeps FG and N2 and generates JOINTS1 in a piecewise fashion, not needing to store more than one cantle of JOINTS1 at any given time, where if L is a typical cantle of FG the corresponding cantle of JOINTS1 is given by

\[ \text{DIC}(N2)|L \]

Alternatively, one may actually create JOINTS1 and dispose of FG. This approach may be advantageous from the point of view of economy in storage, since JOINTS1 is of the first grade and requires less storage area than FG. One may then require to recreate FG in a subsequent stage of the analysis and a redictum function may be used for this purpose. Thus, one may write

\[ FG = \text{RED}(N2)|\text{JOINTS1} \]

Again, one need not necessarily recreate the whole of FG and may proceed to generate parts, of FG, as the need arises, where if L is a typical cantle of JOINTS1 then the corresponding cantle of FG is obtained as

\[ \text{RED}(N2)|L \]

### 2.4.3 SEVIATION FUNCTIONS

Consider a formex of the first grade and let it be required to compare the cantles of E to establish the greatest difference 'δ' between every possible pairs of uniples. Finally, let δ's for all the cantles of E be compared and the greatest of these is
denoted by $\Delta$. The relation by which the integer $\Delta$ is obtained from $E$ is written as

$$\Delta = SEV|E$$

The integer $\Delta$ is referred to as the seviation of $E$. The function

$$SEV$$

is referred to as a "seviation function". The seviation function has no inverse. For example, if

$$E = \{[10; 3], [22; 58], [31; 63; 12], 9\}$$

then $\delta$'s for the first, second, third and fourth cantles of $E$ are 7, 36, 51 and 0, respectively, and

$$SEV | E = 51$$

The seviation of the empty formex is considered to be zero. That is,

$$SEV | \{\} = 0$$

The practical significance of the seviation function will be illustrated in terms of the examples of the preceding section. Thus if, $FG$ is the formex of the first grade representing the interconnection pattern of the grid of Fig 2.51 in terms of the indicated node numbering scheme then

$$\Delta = SEV | FG = 10$$

This number is, in fact, the greatest difference between the terminal node numbers for an element in the configuration of Fig 2.51. The number is a measure of the
band-width of the stiffness matrix of a structure with that configuration. Thus, the
seviation function can be used to provide band-width information in automated
structural analysis processes.

2.4.4 NOVATION FUNCTIONS

Consider a formex $E$ and let $F$ be a formex of the second plexitude and with the same
grade as $E$. Suppose a typical cantle of $F$ is denoted by

$$[S_1; S_2]$$

Where $S_1$ and $S_2$ are the first and the second signets in the cantle. Let $E$ be modified
by replacing every signet of it that is equal to $S_1$ by $S_2$ and let this process be
repeated for all the cantles of $F$ proceeding in the natural order. Let the resulting
formex be denoted by $G$.

The rule through which $G$ is obtained from $E$ is symbolised in terms of a formex
function. This function is denoted by

$$\text{NOV}(F)$$

The formex $G$ is referred to as the "novation" of $E$ with respect to $F$ and the relation
between $G$ and $E$ is written as

$$G = \text{NOV}(F) | E$$

For instance, if

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

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

then a modification of \( E \) with respect to the first cantle of \( F \) results in

\[ E_1 = \{[2,2; 2,2], [3,4; 2,2]\} \]

A modification with respect to the second cantle of \( F \) gives rise to

\[ E_2 = \{[3,3; 3,3], [3,4; 3,3]\} \]

Therefore,

\[ \text{NOV}(F)E = \{[3,3; 3,3], [3,4; 3,3]\} \]

As an example to illustrate the practical application of novation functions, consider the configuration shown in Fig 2.53 and let this represent a view of a braced dome. Suppose that the nodal points lie on a surface of a sphere whose radius is equal to 12 units of length. A formex representing the interconnection pattern of the dome may be written, in terms of the pronorm shown, as

\[ \text{DOME} = \text{RINIT}(12,3,2,2) | \text{ROSAT}(1,3) | [10,0,3; 10,1,4]. \]

In the above formex, the nodes encircled are represented by two different sets of signets associated with lines \( U2 = 0 \) and \( U2 = 20 \). These two sets of signets are shown in Fig 2.54.

In a purely geometric consideration, the above formex \( \text{DOME} \) may be regarded as representing the interconnection pattern of the dome. However, if it is to be used as data describing the interconnection pattern of the configuration for a structural analysis and if it is to be subjected to a dictum function to give rise to a formex that describes the interconnection pattern in terms of a node numbering system, then the
**Table 2.8**

<table>
<thead>
<tr>
<th>SYMBOLIC REPRESENTATION</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[&gt;P]F$</td>
<td>$E$ is more rapportant than $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[&lt;P]F$</td>
<td>$E$ is less rapportant than $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[=P]F$</td>
<td>$E$ and $F$ are of the same rapportance with respect to $P$.</td>
</tr>
<tr>
<td>$E[&gt;=P]F$</td>
<td>$E$ is of a rapportance higher than or equal to $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[=&lt;P]F$</td>
<td>$E$ is of rapportance lower than or equal to $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[&gt;P]F$</td>
<td>$E$ is not more rapportant than $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[&lt;P]F$</td>
<td>$E$ is not less rapportant than $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[=P]F$</td>
<td>$E$ and $F$ are not of the same rapportance with respect to $P$.</td>
</tr>
<tr>
<td>$E[&gt;P]F$</td>
<td>$E$ is not of a rapportance higher than or equal to $F$ with respect to $P$.</td>
</tr>
<tr>
<td>$E[&lt;P]F$</td>
<td>$E$ is not of a rapportance lower than or equal to $F$ with respect to $P$.</td>
</tr>
</tbody>
</table>
nodes encircled will not be assigned a unique node number. This implies that there are no connections between the members on the right and the members on the left to the nodes in question, as shown by the gaps in Fig 2.54.

To overcome the problem of discontinuity of this kind, one may use a novation function and obtain

\[ DOME_1 = NOV(F) \mid DOME \]

where \( F \) is a formex of the form

\[ F = RIN(3,3,-2) \mid [10,20,7; 10,0,7] \]

It can be verified that the signets representing the nodes consideration are modified and the dome represented by the formex \( DOME_1 \) is free of any problems regarding the discontinuities at the nodes.

2.4.5 COLLATION FUNCTIONS

2.4.5.1 INTRODUCTION

There are situations when it is required to rearrange the order of appearance of a sequence of formices in a manner that satisfies a given set of conditions. For instance, if \( F \) is a formex representing the interconnection pattern of a structure then it may be required to rearrange the signets of a catena of \( F \) such that, in connection with a dictum function it gives rise to an optimum band-width for the stiffness matrix of the structure.

Operations of this type may be performed using the concept of rapportance and this is described in the sequel.
2.4.5.2 CONCEPT OF RAPPORTANCE

Let E and F be two formices and let P be a perdicant that may be evaluated with respect to E and F. For example, consider the following formices

\[ F_1 = \{(3,4; 6,5), (1,3; -1,7)\} \]
\[ F_2 = [4,3,5; -1,2,2] \]
\[ F_3 = [-1,2; 4,1; 1,1] \]
\[ F_4 = \{(4,2,3,3; 5,3,1,2), (1,2,4,3)\} \]

and let a perdicant P be specified in terms of a typical pair of these formices as follows:

\[ P \text{ is true provided that the sum of the uniples of the first signet of the first cantle of the left element of the pair is greater than that of the right element of the pair.} \]

Examining \( F_1 \) and \( F_2 \) with respect to P one finds that the sum of the uniples of the first signet of the first cantle of \( F_1 \) is 7 and the sum of the uniples of the first signet of the first cantle of \( F_2 \) is 12.

Thus, P is false in the order \( F_1, F_2 \) and is true in the order \( F_2, F_1 \). Therefore, \( F_1 \) is less rapportant than \( F_2 \) with respect to P. That is,

\[ F_1 < P F_2 \]

or stating it differently, \( F_2 \) is more rapportant that \( F_1 \) with respect to P. That is,

\[ F_2 > P F_1 \]
Similarly, it is found that

\[ F_1[>P]F_3, \]

\[ F_1[<P]F_4, \]

\[ F_2[>P]F_3, \]

\[ F_2[=P]F_4 \]

and

\[ F_3[<P]F_4 \]

Various relationships between E and F may be symbolically represented as shown in Table 2.8.

Note that two formices whose relative rapportance is examined may be of different order, plexitude and grade.

2.4.5.3 PROCESS OF RAPPORTATION

Consider a sequence \( S \) of \( n \) formices where \( n \geq 2 \) and let there be a perdicant \( P \) which may be evaluated with respect to any two elements of the sequence.

Let the phase "\((i,j)\) forward seriation" be used to refer to the following procedure:

For \( k = i, i+1, i+2, \ldots, j-1 \) the \( k^{th} \) and the \( (k+1)^{th} \) elements of \( S \) are compared and if the latter is more rapportant than the former with respect to \( P \) then these elements are interchanged.

Also, let the phase "\((i,j)\) backward seriation" be used to refer to the following
procedure:

For \( k = j, j-1, j-2, \ldots, i+1 \) the \( k^{\text{th}} \) and \( (k-1)^{\text{th}} \) elements of \( S \) are compared and if the latter is less rapportant than the former with respect to \( P \) then these elements are interchanged.

The term "seriation" is used to refer to either a forward seriation or a backward seriation.

Let the sequence \( S \) be subject to the following process:

For \( k = 1, 2, \ldots a (k,n-k+1) \) forward seriation is carried out and this followed by an \((n-k,k)\) backward seriation. The process is brought to an end either when the total number of completed seriations is equal to \( n-1 \) or when a seriation does not involve any interchange of elements.

The above process is referred to as the "process of rapportation".

For example, consider the sequence of maniples

\[ E = \begin{bmatrix} 4,2; 3,1 \end{bmatrix}, \]

\[ F = \begin{bmatrix} 7,4,3; 6,4,2 \end{bmatrix}, \]

\[ G = \begin{bmatrix} 6,5; 5,5; 4,6 \end{bmatrix} \]

and

\[ H = [5,2] \]

and let a perdicant \( P \) with respect to two typical elements \( M_a \) and \( M_b \) be stated as
follows:

\[ P \text{ is true provided that the second uniple of the first signet of } Ma \text{ is greater than that of } Mb \text{ and } P \text{ is false otherwise.} \]

This predicant in brevic notation may be written as

\[ U(1,2) > W(1,2) \]

It will be found that

\[ E[<Q]F, \]

\[ E[<Q]G, \]

\[ E[=Q]H, \]

\[ F[<Q]G, \]

\[ F[>Q]H \]

and

\[ G[>Q]H \]

In reporting the sequence

\[ E,F,G,H \]

with respect to \( P \), one begins by applying a \((1,4)\) forward seriation and this will turn the sequence into

\[ F,G,E,H \]

A \((3,1)\) backward seriation is applied next and this will turn the sequence into

\[ G,F,E,H \]
Finally, a (2,3) forward seriation is applied and this will leave the sequence unaltered.

Note that the above sequence contains formices that are of different grades and plexitudes. In general, if S is a sequence of formices which is to be rapported with respect to a perdicant P, then the only condition that the elements of S should satisfy must be applicable to every pair of elements of S.

2.4.5.4 RAPPORTED SEQUATION FUNCTIONS

Consider the formex E and let there be a perdicant P which may be evaluated with respect to every pair of the cantles of E. Let E be modified by subjecting its cantles to the process of rapportation with respect to P and the resulting formex be denoted by G. The rule through which E is transformed into G is represented by a function. This function is denoted by

\[ \text{RAS}(P) \]

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

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

For example, suppose that

\[ E = \{[1,2], [3,4], [3,2], [2,4], [3,1], [2,2], [1,1]\} \]

and let a perdicant P, in brevic notation, be given by

\[ (U(2) > (W2)) \text{ OR } (U(2) = W(2) \text{ AND } U(1) < (W1)) \]
The rapported sequation of $F$ with respect to $P$ is found to be

$$\text{RAS}(P)|E=\{[2,4], [3,4], [1,2], [2,2], [3,2], [1,1], [3,1]\}$$

Any rapported sequation of the empty formex is the empty formex itself. Also, if $E$ is a formex of the first order, that is $E$ is a maniple, then any rapported sequation of $E$ is $E$ itself. A rapported sequation function has no inverse.

### 2.4.5.5 RAPPORTED VARIANT FUNCTIONS

Consider a formex $E$ and let there be a perdicant $P$ which may be evaluated with respect to every pair of signets that are contained in a cantle of $E$. Let a formex $G$ be obtained by replacing every cantle $C$ of $E$ by a maniple which is obtained by rapporting the signets of $C$ with respect to $P$. The rule through which $E$ is transformed into $G$ is represented by a function. This function is denoted by

$$\text{RAV}(P)$$

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

$$G=\text{RAV}(P)|E$$

For example, let

$$F=\{[5,3; 5,1; 5,6], [3,6], [2,1; 3,5], [1,3; 2,3]\}$$

and let $P$ in brevic notation be given by

$$(U(1)+U(2))<(W(1)+(W2))$$
The rapported variant of $F$ with respect to $P$ is found to be

$$RAV(P)|F=\{(5,1; 5,3; 5,6], [3,6], [3,5; 2,1], [2,3; 1,3]\}$$

Any rapported variant of the empty formex is the empty formex itself. Also, if $E$ is a 1-plex formex then anyrapported variant of $E$ is $E$ itself. A rapported variant function has no inverse.

### 2.4.6 SEDILATE PERDICANTS

There is a family of perdicants that finds frequent application in providing canonic parameters for collation functions. A typical member of this family may be written as

$$(U_i < > W_i)$$

OR $$(U_i=W_i \text{ AND } U_j < > W_j)$$

OR $$(U_i=W_i \text{ AND } U_j=W_j \text{ AND } U_k < > W_k)$$

... 

OR $$(U_i=W_i \text{ AND } U_j=W_j \text{ AND } U_k < > W_k \text{ AND} ... \text{ AND } U_q=W_q \text{ AND } U_r < > W_r)$$

where the symbol $< >$ is used to indicate the symbol $<$ or the symbol $>$ and where it is understood that the entities being compared with respect to the perdicant are either two maniples of the first plexitude or two maniples of the first grade. A perdicant of this kind is referred to as a "sedilate perdicant" and is represented by

$$SED(+i, +j, +k, ..., +q, +r)$$

where each item $i,j,k, ..., q$ or $r$ is positive if the corresponding symbol in the
perdicant is $<$ and is negative otherwise.

For example, the perdicant

\[ (U(1) < W(1)) \text{ OR } (U(1) = W(1) \text{ AND } U(2) < W(2)) \]

is represented

\[ \text{SED}(1,2) \]

the perdicant

\[ (U(5) < W(5)) \text{ OR } (U(5) = W(5) \text{ AND } U(1) > W(1)) \]

\[ \text{OR } ((U5) = W(5) \text{ AND } U(2) < W(2)) \]

is represented

\[ \text{SED}(5,-1,2) \]

and the perdicant

\[ U(4) > W(4) \]

is represented by

\[ \text{SED}(-4) \]

To illustrate the practical applications sedilate perdicants, consider the configuration shown in Fig 2.55 and let this represent a solid rectangular finite element with corner nodes. A formex representing the compret of the element relative to the indicate
pronorm may be written as

\[ F1 = [1, 2, 1; 2, 1, 1; 2, 1, 2; 1, 1, 2; 2, 2, 2; 2, 2, 1; 1, 1, 1; 1, 2, 2] \]

The order in which the nodes of the element are represented in \( F1 \) is indicated in Fig 2.55. Now let it be required to arrange the nodes of the element in a different order and let it be assumed that this required manner of ordering may be defined in terms of "direction precedence". A formex representing the compret of the element may then be obtained through a rapported variant function using a sedilate perdicant. For instance, the ordering of the nodes indicated in Fig 2.56 may obtained by writing

\[ F2 = \text{RAV}(\text{SED}(1, 2, 3))\mid F1 \]

Similarly the ordering of the nodes indicated in Figs 2.57 and 2.58 may be obtained, respectively by writing

\[ F3 = \text{RAV}(\text{SED}(2, -1, 3))\mid F1 \]
and

\[ F4 = \text{RAV}(\text{SED}(-3, -2, 1))\mid F1 \]

A sedilate perdicant need not necessarily refer to all the directions involved in a problem. For instance, the ordering of the nodes indicated in Figs 2.59 to 2.62 may be obtained, respectively, by formices

\[ F5 = \text{RAV}(\text{SED}(2, 3))\mid F1, \]
\[ F6 = \text{RAV}(\text{SED}(1, -3))\mid F1, \]
\[ F7 = \text{RAV}(\text{SED}(-1))\mid F1 \]
and

\[ F8 = \text{RAV}(\text{SED}(2))\mid F1 \]

As the second example consider the double layer Z-plan of which is shown in
Fig 2.63 and let it be required to write a formex formulation for the compret of the structure in terms of the given node numbering scheme. One may begin the formulation by writing the compret relative to indicated trifect pronorm. This may written as

\[
A_1 = RINID(7,5,2,2) | ROSAD(2,2) | \{[1,1,1; 3,1,1], [1,1,1; 2,2,0]\}
\]
\[
A_2 = RINID(6,4,2,2) | ROSAD(3,3) | [2,2,0; 4,2,0]
\]
\[
A = PEX(A_1 \# A_2)
\]
\[
A_3 = RINID(3,2,4,4) | [4,4,0]
\]
\[
A_4 = LUX(A_3) | A
\]

The formex \(A_4\) represents the compret of the double layer grid relative to the trifect normat of Fig 2.63. A formex representing the compret of the double layer grid relative to the required node numbering scheme may now be written as

\[
G = DIC(N) | A_4
\]

where

\[
N = RAM(SED(2,1)) | A_4
\]

As the final example, consider the flat grid a plan view of which is shown in Fig 2.64. A formex representing the compret of the grid relative to the indicated paribifect pronorm may be written as

\[
D = LUX([9,5; 9,7]) | PEX | RINID(6,5,2,2) | \ldots
\]
\[
LAMAD(1,1,2,2) | \{[1,1; 3,1],[1,1; 1,3]\}
\]

The compret of the structure relative to the given node numbering scheme may be written as

\[
D_1 = DIC(N_1) | D
\]

where the numerant \(N_1\) is given by

\[
N_1 = RAM(SED(1,2)) | D
\]
The formex D1 consists of 64 cantles and the order of appearance of these cantles in D1 is as indicated in Fig 2.65. This order of appearance of cantles is dictated by the manner in which D is generated.

2.4.7 MEDULLA FUNCTION

Consider a formex E and let an ingot T be constructed from all the signets of E, with the signets appearing in exactly the same order as in E. For instance, if

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

then

\[ T=\{[1,2], [3,3], [1,1], [2,2], [3,1], 1,3]\} \]

Let an ingot G be obtained as

\[ G=PEX|T \]

The rule through which G is obtained from E is symbolised in terms of a formex function. This function is referred to as the "medulla function" and the relation between E and G is written as

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

The formex G is referred to as the medulla of E.

For instance, if

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

then

\[ \text{MED} | E=\{[1,2], [3,3], [2,2], [3,1], [1,3]\} \]
As an example consider the flat grid in Fig 2.48. A formex formulation of the interconnection pattern with respect to the pronorm shown may be written as

\[ \text{GRID} = \text{LUX}(7,9; 9,9; 11,7; 11,9) | \text{RINID}(8,6,2,2) | \ldots \]

\[(1,1; 3,3) \# \text{ROSAD}(2,2) | [1,1; 3,1) \]

The formex

\[ \text{JOINTS} = \text{MED} | \text{GRID} \]

is an ingot representing all the nodal points of the grid in the manner shown in Fig 2.49, where the integer associated with each node is the orderate of the corresponding cantle in JOINTS.

The above example illustrates the fact that once a formex F, representing the interconnection pattern of a structural system is formulated, an ingot T, representing the nodal points may then be conveniently obtained from F by means of the medulla function. Furthermore, if the ingot T is to be used as data providing information about nodal coordinates in a computer analysis, then the order of appearance of cantles in T represents a particular node numbering scheme.

On the other hand, for an analysis program incorporating band width solution techniques, the node numbering scheme is often required to be such that the resulting band width is minimised. Therefore, the order of appearance of the signets in ingot T obtained using medulla function has to be rearranged. This may be done through the following two constructs:

\[ \text{RAM(SED}(1,2)) \]

and

\[ \text{RAM(SED}(2,1)) \]
Both the constructs \( \text{RAM}(\text{SED}(1,2)) \) and \( \text{RAM}(\text{SED}(2,1)) \) are special forms of a formex function known as "rapported medulla function".

A rapported medulla function has the effect of forming the medulla of a formex and then transforming the medulla into a sequation of it, with the resulting sequation satisfying a given condition. The use of functions \( \text{RAM}(\text{SED}(1,2)) \) and \( \text{RAM}(\text{SED}(2,1)) \) may be illustrated in terms of the formices

\[
\text{JOINTS}_1 = \text{RAM}(\text{SED}(1,2)) | \text{FG}
\]

\[
\text{JOINTS}_2 = \text{RAM}(\text{SED}(2,1)) | \text{FG}
\]

where \( \text{GRID} \) is the formex representing the configuration of Fig 2.48. \( \text{JOINTS}_1 \) and \( \text{JOINTS}_2 \) are ingots which are sequations for Fig 2.49. Formices \( \text{JOINTS}_1 \) and \( \text{JOINTS}_2 \) give rise to the representation of structural nodes in the manner shown in Figs 2.50 and 2.51.

2.5 FORMIAN: THE PROGRAMMING LANGUAGE OF FORMEX ALGEBRA

Formian is the name of an interactive programming language which may be employed for data generation using the concepts of formex algebra. The data generated are stored in the computer for analysis and design purposes. The advantage of using a programming language for data generation is flexibility, the type of configuration that can be generated being limited only by the user's imagination.

2.5.1 PRIMITIVE CONSTITUENTS

Characters are the main building blocks of the syntactic constructs of Formian. Characters sequences of various forms are used to represent mathematical entities,
define computational procedures and effect data management.

CHARACTERS: 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 distinctive printable mark that can be represented by a computing system and is not a digit or a letter. Examples of symbols are +, -, {, %, *, > and &. A layout character is either a space character or a newline character.

KEYWORDS: Any one of the following twelve sequences of letters is referred to as a keyword.

KEEP TAKE PRINT GIVE DRAW SHOW

SUBMIT ERASE USE RECALL EXIT STOP

Keywords are used for initiation of various activities. A keyword may be given in uppercase or lowercase letters. Thus, TAKE, Take and take will have the same effect when used as a keyword.

IDENTIFIERS: An identifier is a sequence of letters and digits whose first character is a letter. An identifier may not be the same as a keyword and may not have more than twelve characters. Examples of identifiers are

VAULT G i98 dome5 S6u1d.

Identifiers are used for names of entities such as variables and functions. Both uppercase and lowercase letter may be use in construction of identifiers. However, in comparing the identifiers, no distinction is made between uppercase and lowercase. Thus, the identifiers
are considered to be identical.

2.5.2 CONSTANTS

A "constant" is a sequence of characters which is an explicit representation of an abstract entity. The abstract entity that is represented by a constant is referred to as the value of the constant. Formian has five types of constants. These represent integer numbers, noninteger real numbers, integer formices, noninteger formices and character strings.

**INTEGER CONSTANTS:** An integer constant is a sequence of one or more digits which may be preceded by a plus or minus sign. For example, 1967 and -14 are acceptable integer constants but -14.00 and 1,967 are not. The value of an integer constant is the numerical value that results from the interpretation of the constant as a decimal integer number.

**FLOATAL CONSTANTS:** A floatal constant is one of the following forms:

(1) An integer constant followed by a decimal point followed by a sequence of one or more digits, for example, 1.24, -8.456 and +543.0,

(2) A floatal constant of the first form followed by the letter E followed by an integer constant, for example, 0.34E3 and +298.56E5,

(3) An integer constant followed by the letter E followed by an integer constant, for example, 76E8 and -123E3.

The value of a floatal constant is the numeric value that results from the interpretation
of the constant as a decimal real number. The representation of the value of a floatal constant in a computer is in floating point form which provides an approximation to the actual value. The term numeric constant is used to refer to an integer constant or a floatal constant.

**FORMEX CONSTANTS:** A formex constant is a structured sequence of numeric constants, commas, semicolons, square brackets and curly brackets. For example,

```
{[1, 2; 2, 2], [4.5, 6], [-3, -2; -1, 0; 4.5, 3]}
```

is a formex constant. The value of a formex constant is a "formex". A formex constant is said to be an integer formex constant provided that all the numeric constants in it are integer constants. A formex constant is said to be a floatal formex constant provided that at least one of the numeric constants in it is floatal constants. For instance the above formex constant is a floatal formex constant.

**STRING CONSTANTS:** A string constant is of the form 'C' where ' is the quote symbol and C is a sequence of characters with the provision that if C contains any 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

"""In rivers the water that you touch is the last of what has passed and the first of that which comes", said Leonardo da Vinci'.

is a valid string constant.

The value of a string constant is the sequence of characters that is enclosed between the initial and terminal quote symbols which every batch of 2n quote symbols
replaced by n quote symbols. For example, the value of the above string constant is

\[ 'In \text{ rivers the water that you touch is the last of what has passed and the first of that which comes}, \text{ said Leonardo da Vinci.} \]

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.

### 2.5.3 ASSIGNMENT STATEMENTS AND VARIABLES

A variable is an identifier which has been assigned a value. Assignment statements are the vehicles through which values are assigned to identifiers. There are five types of variables namely, integer variables, floatal variables, integer formex variables, floatal formex variables, and string variables.

An assignment statement is of the general form

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

where the symbol = is referred to as the assignment symbol. The right-hand side of an assignment statement is an expression, which is a structured sequence of constants, variables, operators, ..., etc, as will be discussed later. However, the effects of an assignment statement may be conveniently explained in terms of the simplest forms of expressions, namely, constants and this approach is adopted to describe assignment statements in the sequel.

The assignment statement

\[ P = 678 \]
has the effect of creating a variable P which will represent the integer number 678. This variable is said to be an integer variable since its value is an integer number.

If at the moment of appearance of the above assignment statement, N is already a variable then the effect of the assignment statement would be to discard the old value of N and assign a new value to it. Thus, if the above assignment statement is followed by

\[ P = 5.34E+3 \]

then the current value of P which is the integer number 678 is irrecoverably lost and the real number 5340.0 in floating point form will be recorded as the value of P. The identifier P at this point will cease to be an integer variable and will become a floatal variable. The term numeric variable is used to refer to an integer variable or a floatal variable.

Further examples of assignment statements are

\[
\begin{align*}
\text{POPI} & = [2,3,0; 6,7,1; -8,5,0] \\
\text{SUM} & = \\
\text{GUM} & = 'Floor'
\end{align*}
\]

The first assignment statement creates a floatal formex variable POPI, the second assignment statement creates an integer formex variable SUM and the third one creates a string variable GUM. The term formex variable is used to refer to an integer formex variable or a floatal formex variable.

2.5.4 FORMEX FUNCTIONS

A function is a sequence of characters that represents a rule for the production of a
value. Functions are defined in two different ways. Firstly, there are a number of functions whose definitions are incorporated into the formian Interpreter and are available to all users. These are referred to as standard functions and are described in the sequel. Secondly, there are functions that may be defined by adding program segments to the Formian Interpreter and creating special versions of Formian. These functions are referred to as supplementary functions.

A function that gives rise to a numeric value is referred to as a numeric function. There are fourteen numeric functions and these are described in Table 2.9. A function that gives rise to a formex is referred to as a formex function. The appearance and the use of formex function in Formian conforms to the definition in formex algebra. For example, the construct

\[ F_1 = \text{RINID}(4,5,2,2) | \text{ROSAD}(2,2) | E \]

where

\[ E = \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix} \]

has the same appearance in both formex algebra and Formian.

The evaluation of a sequence of nested functions proceeds from right to left. That is, in the above example, the ROSAD function will be executed before the RINID function.

2.5.5 OPERATIONS AND EXPRESSIONS

An operation is a rule for production of a value from one or more operands. An operation is represented by an operator. For example, the operation of addition is a rule for obtaining the sum of two numeric entities and this sum may be denoted by
<table>
<thead>
<tr>
<th>Function</th>
<th>Value of Function Designator with Argument X Being a Numeric Expression</th>
<th>Type of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIC (rounded integer conversion)</td>
<td>The value of RIC(X) is the integer number nearest to the value of (X). For example, the values of RIC(4.5), RIC(-4.5), RIC(3.456), RIC(0.674) and RIC(9) are 4, -5, 3, 1, and 9 respectively.</td>
<td>Integer</td>
</tr>
<tr>
<td>TIC (truncated integer conversion)</td>
<td>The value of TIC(X) is the integer number obtained by truncating the fractional part of the value of (X). For example, the values of TIC(4.5), TIC(-4.5), TIC(3.456), TIC(0.674) and TIC(9) are 4, -4, 3, 0 and 9 respectively.</td>
<td>Integer</td>
</tr>
<tr>
<td>SIGN</td>
<td>The value of SIGN(X) is the integer number 1, 0 or -1 depending on the value of (X) being positive, zero or negative, respectively. For example, the value of SIGN(-3.8) is -1.</td>
<td>Integer</td>
</tr>
<tr>
<td>ABS (absolute value)</td>
<td>The value of ABS(X) is the absolute value of (X). For example, the values of ABS(-2.45) and ABS(6) are +2.45 and 6, respectively.</td>
<td>Type of ABS(X) is the same as that of (X).</td>
</tr>
<tr>
<td>FLOC (floatal conversion)</td>
<td>The value of FLOC(X) is the value of (X) in the floating point form.</td>
<td>Floatal</td>
</tr>
<tr>
<td>SQRT (square root)</td>
<td>The value of SQRT(X) is the square root of the value of (X). For example the value of SQRT(16.25) is 2.5.</td>
<td>Floatal</td>
</tr>
<tr>
<td>SIN</td>
<td>The value of SIN(X) in the sine of the value of (X), regarded as an angle in degrees. For example, the value of SIN(60) is 0.707.</td>
<td>Floatal</td>
</tr>
<tr>
<td>COS</td>
<td>The value of COS(X) is the cosine of the value of (X), regarded as an angle in degrees. For example, the value of COS(60) is 0.5.</td>
<td>Floatal</td>
</tr>
<tr>
<td>TAN</td>
<td>The value of TAN(X) is the tangent of the value of (X), regarded as an angle in degrees. For example, the value of TAN(45) is 1.0</td>
<td>Floatal</td>
</tr>
<tr>
<td>ASIN (arcsine)</td>
<td>The value of ASIN(X) is the angle in degrees (in the range (-90) to (90)) whose sine is equal to the value of (X) (The value of (X) must be in the range (-1) to (1)). For example, the value of ASIN(1) is 90.</td>
<td>Floatal</td>
</tr>
<tr>
<td>ACOS (arccosine)</td>
<td>The value of ACOS(X) is the angle in degrees (in the range (0) to (180)) whose cosine is equal to the value of (X). (The value of (X) must be in the range (-1) to (1)). For example, ACOS(0.5) is 60.0.</td>
<td>Floatal</td>
</tr>
<tr>
<td>ATAN (arctangent)</td>
<td>The value of ATAN(X) is the angle in degrees (in the range (-90) to (90)) whose tangent is equal to the value of (X). For example, ATAN(1) is 45.0.</td>
<td>Floatal</td>
</tr>
<tr>
<td>(LN) (natural logarithm)</td>
<td>The value of LN(X) is the natural logarithm of the value of (X) (The value of (X) must be positive). For example, the value of LN(1) is 0.0.</td>
<td>Floatal</td>
</tr>
<tr>
<td>EXP (exponential)</td>
<td>The value of EXP(X) is (e^v) where (e) is the base natural logarithm and (v) is the value of (X). For example, EXP(0) is 1.0.</td>
<td>Floatal</td>
</tr>
</tbody>
</table>
a construct such as

\[ 6 + 3 \]

where 6 and 3 are the operands and the symbol + is the operator.

There are altogether seven operations in Formian. Six of these operations are for numeric operands and are referred to as numeric operations. The seventh operation is referred to as a composition and applies to formex operands. Numeric operations are described in Table 2.10. The first five of these operations are binary, that is, they have two operands. The last operation is unary and has a operand. This unary operation involves the placing of a plus or minus sign before an unsigned numeric entity. The operands of all the operations in Table 2.10 are numeric expressions.

A numeric expression is a mathematically meaningful combination of numeric constants, numeric variables, numeric function designators, numeric operators and parentheses. Examples of valid numeric expressions are

\[
\begin{align*}
5 + N / (45.6 + M) \\
-6.9 - (Y + B / 3) + TAN|A \\
3.78 ^ 9.5
\end{align*}
\]

where N, M, Y, B and A are assumed to be numeric variables. The last expression represents 3.78 to the power of 9.5. A single numeric constant, variable or function designator is a special case of a numeric expression.

A numeric expression is said to be an integer expression if its value is integer and is said to be a floatal expression if its value is floatal.

The operator for the operation of formex composition is the symbol # which is referred to as the duplus symbol. Two formices of the same grade may be composed
### Table 2.10 Numeric Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operator</th>
<th>Number of Operands</th>
<th>Type of Resulting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>addition</td>
<td>+ (plus)</td>
<td>2</td>
<td>If both operands have integer values then the operation yields an integer value, otherwise the operation yields a floatal value.</td>
</tr>
<tr>
<td>subtraction</td>
<td>− (minus)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>multiplication</td>
<td>× (sidus)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>division</td>
<td>/ (solidus)</td>
<td>2</td>
<td>Operation yields a floatal value.</td>
</tr>
<tr>
<td>exponentiation</td>
<td>^ (tantis)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>sign prefixion</td>
<td>+ (plus)</td>
<td>1</td>
<td>Operation yields a value of the same type as that of the operand.</td>
</tr>
<tr>
<td></td>
<td>or − (minus)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](image)

**Fig 2.66**
and the result is obtained by their concatenation. Thus

\[((5,7; 1,2), [-3,1; 8,4]) \# ([1,3; 2,0], [6,7; -2,-5])\]

will give rise to

\[((5,7; 1,2), [-3,1; 8,4], [-1,3; 2,0], [6,7; -2,-5])\]

The operands in the operation of composition are formex expressions.

A formex expression is a meaningful combination of formex variables, formex function designators, duplus symbols, parentheses and formex functions. A formex formation is a construct whose constitution is similar to a formex constant and in which the uniples are numeric expressions. For instance, if i, j and m are numeric variables, then

\[-4, i+2*j; i*5, j\]

and

\[((3,2,4; -1,1,6*m), [1,0,-3; 4, M-1,3*m])\]

are formex formations. A formex constant is a special case of a formex formation.

Examples of valid formex expressions are

\[\text{RIN}(3,p)\|[4,n; n-1,3] \# [3,2; 4,5], [8,1; 3,6]\]

and

\[\text{ROSAD}(p,1)\|[2,4,0; 5,6,1] \# D\]

where p and n are assumed to be numeric variables and D is assumed to be a formex
A formex expression is said to be an integer formex expression if it results in a value which is a formex of type integer. Also, a formex expression is said to be a floatal formex expression if its value is of type floatal.

A string expression is either a string constant or a string variable. The term expression is used to refer to a numeric expression or a formex expression or a string expression.

Operators are not allowed to follow one another at any point of an expression. Thus

\[ 3.5 \ ^ {-6.7} \]

is not a valid expression, but

\[ 3.5 \ ^ {(-6.7)} \]

is an acceptable one.

If the order of performance of various processes for evaluation of an expression is not completely determined by the nature of processes and the parentheses in the expression, then amongst all possible ways of execution at any stage of evaluation, the one which is compatible with the precedence order of Table 2.11 will be chosen.

Furthermore, if the order of execution for some operations remains undetermined after the above considerations then these operations are performed from left to right. Thus

\[ \text{RIN}(2,3,1) | \text{LAM}(-2,4) | \text{D} \ # \ F \]

and

\[ N + \text{COS} | -Y^2 - 4.2/3 \]

are executed as though they were

\[ (\text{RIN}(2,3,1)) | (\text{LAM}(-2,4) | \text{D})) \ # \ F \]
### Table 2.11 Precedence Order

<table>
<thead>
<tr>
<th>Process</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>sign prefixion</td>
<td>Highest Precedence (executed first)</td>
</tr>
<tr>
<td>evaluation of function designator</td>
<td>Decreasing Degree of Precedence</td>
</tr>
<tr>
<td>exponentiation</td>
<td></td>
</tr>
<tr>
<td>multiplication and division</td>
<td>Lowest Precedence (executed last)</td>
</tr>
<tr>
<td>addition and subtraction</td>
<td></td>
</tr>
<tr>
<td>formex composition</td>
<td></td>
</tr>
</tbody>
</table>

**Fig 2.67**
and

\[(N + ((\cos(\cdot Y))^2)) - (4.2/3)\]

Also, the expression

\[2 - m^3 \# 7*(m-4) + 6\]

is considered to mean

\[(2 - (m^3)) \# ((7*(m-4)) + 4)\]

This is equivalent to the formex formation

\[\{2 - (m^3), (7*(M-4)) + 4\}\]

where \(m\) is assumed to be a numeric variable. The above example illustrates the fact that a numeric entity is a special case of a formex.

### 2.5.6 SCHEMES

In Formian, a character string can provide a means for collecting a sequence of statements together in a construct referred to as a "scheme". A scheme has the merit of allowing a collection of statements to be used more than once. This provides a useful means of tackling repetitive problems such as the generation of data for double layer grids which may occur regularly but vary in size colour and/or the number members used. For example, a scheme for a configuration similar to that used in Fig 2.66 may be constructed by entering

\[
\text{GRID=': X, Y:
A1=\text{RINID(X-1,Y-1,2,2)}|\text{ROSAD(3,3)}|\text{[2,2,0; 4,2,0]}
\]

A2 = RINID(X, Y, 2, 2)|ROSAD(2, 2)|{[3, 1, 1; 1, 1, 1; 2, 1, 2, 0]}
A = A1#A2
D1 = RINID(X-4, Y-2, 2, 2)|{[11, 7, 1]}
D2 = RINID(X-5, Y-3, 2, 2)|{[12, 8, 0]}
D3 = [6, 6, 0]
D4 = RIN(2, Y-1, 2)|{[1, 1, 1]}
D = D1#D2#D3#D4
G = LUX(D)|A'

assuming that there are X divisions in the first direction and Y divisions in the second
direction. The effect of entering the above text is to create the string variable GRID
where the sequence of characters represents a number of valid Formian statements.

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. The body
consists of a sequence of Formian statements that may incorporate the identifiers
listed in the heading. In the above example the heading is

:X, Y:

and the body consists of the assignment statements. The identifiers listed in the
heading are referred to as nominal parameters. The body of the scheme in the above
example may be executed through a statement of the form

GRID(7, 5)

which is referred to as an induction statement. The statement causes the body of the
scheme to be executed with X assuming the value 7 and Y assuming the value 5, as
though the body was preceded by the assignment statements

X = 7
Y = 5
The result of the execution of the scheme is a formex variable that represents the flat grid of Fig 2.66. Now, consider a statement of the form

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

The statement causes the body of the scheme to be executed with X and Y assuming the values 9 and 8, respectively. The resulting plot shown in Fig 2.67.

The concept of a scheme allows a sequence of statements to be turned into a program unit that can be put forward for execution in a convenient manner. At the same time, a scheme provides a suitable vehicle for generic formulation of problems and allows complex configurations to be described in a concise and elegant manner. Also, a scheme may be saved, retrieved and modified repeatedly and may be used as a means of recording information for future reference.

A scheme need not necessarily include any nominal parameters. For instance, the scheme in the following statement does not have any nominal parameters

\[
\text{DOME}='::
E=\begin{bmatrix}
1,0,3;1,0,2;1,1,1
\end{bmatrix},\begin{bmatrix}
1,1,1;1,2,2;1,2,3
\end{bmatrix}
D1=\text{RINIT}(30,3,2,6)|\text{LAM}(3,3)|E
D2=\text{RINIT}(30,2,2,6)|[1,1,5;1,1,7]
D3=\text{RIN}(2,30,2)|[1,1,1;1,3,1]
D=D1#D2#D3'
\]

The above scheme may be executed through the induction statement

\[
\text{DOME}
\]

which will give rise to the view of the dome shown in Fig 2.68. Note that when there are no nominal parameters then the heading of the scheme reduces to a double
Table 2.12 A Channel Chart

<table>
<thead>
<tr>
<th>Channel Number</th>
<th>Description of Channel</th>
<th>Type of Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dot Matrix Printer</td>
<td>Printing Channel</td>
</tr>
<tr>
<td>2</td>
<td>Laser Printer</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Daisy Wheel Printer</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Mouse</td>
<td>Graphical Input Channel</td>
</tr>
<tr>
<td>5</td>
<td>Graphics Tablet</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Light Pen</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Colour Graphics Screen</td>
<td>Graphical Output Channel</td>
</tr>
<tr>
<td>8</td>
<td>Laser Plotter</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Pen Plotter</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Thermal Plotter</td>
<td></td>
</tr>
</tbody>
</table>

Fig 2.68
colon. Also, the induction statement corresponding to a scheme that has no nominal parameters simply consists of the scheme variable.

2.5.7 INFORMATION TRANSFER STATEMENTS

Formian has facilities for the transfer of information between different storage and input-output media.

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

The terminal is advice through which Formian instructions are inputted and from which system messages and other items of information are outputted. In addition, the terminal has an associated graphics output medium. This may be a region of the screen.

The working memory is a medium for storage of values of variables during a Formian session. Storage of information in the working memory is on a temporary basis. At the commencement of a Formian session the working memory is empty. As variables are created and processed, their values are stored in the working memory but at the end of the session all these values are irrecoverably lost.

The repository is a medium for storage of information on a permanent basis. The storage capacity of the repository is normally much greater than that of the working memory but before one can perform any operation on the values in the repository they must be transferred to the working memory.

A printing channel indicates a particular way of textual output on a printing medium. A printing channel is specified by a number. There may be more than one printing
channel but, at any given moment, one of them will be the current graphical output channel.

Information transfer statements which initiate the transfer of information between the Formian Interpreter by which Formian statements are processed. The information transfer statements which will be described are KEEP, TAKE, PRINT, GIVE, DRAW, SHOW and SUBMIT statements.

**KEEP STATEMENT:** It is often convenient to be able to store items such as formices permanently, so that one can use them repeatedly over a period of time, or keep them for future reference. This may be achieved by means KEEP statements.

A KEEP statement is a construct of the form

\[
\text{KEEP E}_1, \text{E}_2, \ldots, \text{E}_n
\]

where KEEP is a keyword and \(\text{E}_1, \text{E}_2, \ldots, \text{E}_n\) are variables. For example, if at a given moment \(\text{FA}\) is a variable then

\[
\text{KEEP FA}
\]

is a valid KEEP statement. The execution of the above statement will cause a copy of the value of \(\text{FA}\) to be stored in the repository and be associated with the identifier \(\text{FA}\).

**TAKE STATEMENT:** The retrieval of items stored in the repository is achieved through TAKE statements. A TAKE statement is a construct of the form

\[
\text{TAKE P}_1, \text{P}_2, \ldots, \text{P}_n
\]
where \textit{TAKE} is a keyword and \textit{P1,P2,...Pn} are covariables. For example, if \textit{DEM} is a covariable, that is, it has been associated with a value in the repository by a previous \textit{KEEP} statement, then

\textit{TAKE DEM}

will be valid statement. The effect of this statement is to place a copy of the value of \textit{DEM} in the working memory and associate this value with the identifier \textit{DEM}. That is, the effect is to create a variable \textit{DEM}. This will not affect the covariable \textit{DEM} and its value in the statement, \textit{DEM} was a current variable then its previous value will be irrecoverably lost.

\textit{PRINT STATEMENT:} A PRINT statement is a construct of the form

\begin{verbatim}
PRINT E1,E2,...,En
\end{verbatim}

where PRINT is a keyword. In the basic form of the statement, each one of the entities \textit{E1,E2,...,En} is a variable and the execution of the statement will cause the values of these variables to be printed on the medium indicated by the current printing channel.

\textit{GIVE STATEMENT:} A give statement is a construct of the form

\begin{verbatim}
GIVE E1,E2,...,En
\end{verbatim}

where GIVE is a keyword and where the entities \textit{E1,E2,...,En} are as described for the \textit{PRINT} statement. The effect of a \textit{GIVE} statement is the same as that of a PRINT statement except that the output will appear on the screen of the terminal.

\textit{DRAW STATEMENT:} A DRAW statement is a construct of the form

\begin{verbatim}
DRAW E1,E2,...,En
\end{verbatim}
where DRAW is a keyword. In the basic form of the statement, each one of the
entities E1, E2, ..., En is either a formex variable or a string variable and the execution
of the statement will cause graphical representations of the formices together with
textual material to appear on the medium indicated by the current graphical output
channel, as will be discussed in the sequel.

**SHOW STATEMENT:** A show statement is a construct of the form

\[
\text{SHOW E1, E2, ..., En}
\]

where SHOW is a keyword and where the entities E1, E2, ..., En are as described for
a DRAW statement. The effect of the SHOW statement is the same as that of a
DRAW statement except that the output will appear on the graphical medium
associated with the terminal.

**SUBMIT STATEMENT:** A SUBMIT statement is a construct of the form

\[
\text{SUBMIT S1, S2, ..., Sn}
\]

where SUBMIT is a keyword and each one of the entities S1, S2, ..., Sn is a data
structure called a Plenix. 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. SUBMIT statements are further discussed in the following
chapter.

### 2.5.8 ORGANISATION STATEMENTS

There are five FORMIAN statements that are used for organisational purposes.
These are ERASE, USE, RECALL, EXIT and STOP statements.
**ERASE STATEMENT:** An ERASE statement is a construct of the form

\[ \text{ERASE A}_1, A_2, \ldots, A_n \]

where ERASE is a keyword and where every one of the entities \(A_1, A_2, \ldots, A_n\) is either a variable or a covariable enclosed in parentheses. The execution of the statement will cause the listed variables and covariables together with their values to be erased from the working memory or repository, respectively. For example, the execution of the statement

\[ \text{ERASE BAD, GRIG, (ROW)} \]

will result in the variables BAD and GRID and the covariable ROW together with their value to be erased.

**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 one of the entities \(A_1, A_2, \ldots, A_n\) is as a USE-item. There are many forms of USE-item employed for a variety of specifications. For instance, USE-item may be employed to specify current input-output channels. To elaborate, suppose that in a particular Formian installation the computing system has the channels listed in Table 2.12, where each channel has an identification number.

In this Formian installation, the USE statement

\[ \text{USE CH(8)} \]

will have the effect of specifying the laser plotter as the current graphical output channel (CH stands for channel).
**RECALL STATEMENT:** A RECALL statement is either of the form

\[
\text{RECALL } D
\]

or of the form

\[
\text{RECALL}
\]

where RECALL is a keyword and D is a string variable. The effect of a RECALL statement of the first type is to display an assignment statement on the screen of the terminal with the left-hand side of the statement being the identifier D and the right-hand side being the constant whose value is the same as that of D. The effect of RECALL statement of the second type is to display the statement which has been executed last.

**EXIT STATEMENT:** An EXIT statement is of the form

\[
\text{EXIT}
\]

where EXIT is a keyword. The effect of the statement is to leave the FORMIAN session temporarily and go to the operating system of the computer. When required activities within the operating system are performed, the user may return to Formian through an appropriate operating system command. On return to Formian, the working environment will be found to have remained unchanged from the moment of leaving the Formian session.

**STOP STATEMENT:** A STOP statement is of the form

\[
\text{STOP}
\]

where STOP is a keyword. The effect of the statement is to terminate the Formian
session and return the user to the operating system of the computer.

2.6 FORMIAN GRAPHICS

The objective of this section is to introduce the concepts and constructs through which images of formex plots may be created on graphical output media.

Consider a formex plot that consists of eight line segments forming a square based pyramid as shown in Fig 2.69. A three-dimensional plot is, of course, an abstraction and cannot be actually realised other than by, perhaps, a physical wire-model. One can, however create two-dimensional views of this plot in the manner described in the sequel.

A plot which is to be viewed is referred to as the object and the coordinate system relative to which the plot is produced is referred to as the object coordinate system. Also, the space in which the plot is imagined to be situated is referred to as the object space.

The point from which the object is to be viewed is referred to as the view point. The point which is directly viewed is referred to as the view centre and the line that passes though both the view centre is referred to as the view line, Fig 2.70.

The view point may be specified by a USE-item of the form

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

where the VP stands for view point and where \( x, y \) and \( z \) are numeric expressions whose values are the coordinates of the view point relative to the object coordinate system. Similarly, the view centre may be specified by a USE-item of the form

\[ \text{VC}(x, y, z) \]
Fig 2.69

Table 2.13

<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(0,0,0,100,100,10)</td>
</tr>
</tbody>
</table>
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.

It is assumed that there is a plane which is normal to the view line at the view centre. The plane is referred to as the trace plane, Fig 2.70. Also, it is assumed that there is a family of lines each of which passes through the view point and a point of the object. These lines are referred to as view rays. The collection of all the points at which the view rays intersect the trace plane creates an image of the object. This image is referred to as the trace of the object, Fig 2.70. The trace in Fig 2.70 is produced using a type of projection which is referred to as perspective projection. However, there is an alternative type of projection which is referred to as parallel projection. 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, Fig 2.71. A perspective projection results in an image which is a perspective view of the object and a parallel projection results in an image which is a parallel view of the object. The required type of view may be specified by a USE-item of the form

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

where VT stands for view type and where n is an integer expression whose value is either 1 or 2, specifying parallel view or perspective view, respectively.

The next stage involves the production of an image of the trace on the output medium of a device such as a VDU or a plotter. This image is referred to as the picture of the object. Furthermore, the plane in which the picture lies is referred to as the picture plane and the coordinate system of the output device is referred to as the device coordinate system, Fig 2.72. One may specify a rectangular frame in the picture plane, restricting the region for graphic production to the area enclosed by the frame. This frame is referred to as the view frame. The view frame may be
specified by a USE-item of the form

$$\text{VF}(p_1, p_2, q_1, q_2)$$

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 corners A1 and A2 of the view frame relative to the device coordinate system, Fig 2.72. The view frame and the device coordinate system are not graphically produced in the picture plane. To signify this fact, the view frame and the device coordinate system in Fig 2.72 are drawn in dotted lines and this convention of using dotted lines for entities that are not graphically produced is used henceforth.

In producing the picture of an object from its trace, it is necessary to have information about

(i) the required orientation of the picture,

(ii) the required position of the picture and

(iii) the required size of the picture.

The required orientation of the picture is determined using a vector which is referred to as the view rise. The view rise is defined relative to the object coordinate system and 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 2.73. The view rise is specified by a USE-item of the form

$$\text{VR}(x_1, y_1, z_1, x_2, y_2, z_2)$$

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

In cases when the starting point of the view rise is coincident with the view centre, it will be possible to specify the view point, the view centre and the view rise at the same time. This is achieved through the concept of view helm which is defined as a broken vector consisting of the view rise and the view centre, as shown in Fig 2.74.

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

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

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

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 nave, with the understanding that the picture will be positioned such that the image of the view base in the picture plane will coincide with the view nave, as shown in Fig 2.75. The view base may be specified by a USE-item of the form

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

where VB stands for view base and where \( x, y \) and \( z \) 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

\[ \text{VN}(p, q) \]
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.

The above 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

\[ \text{VS}(r) \]

where VS stands for view scale and where \( r \) is a numeric expression whose value controls the size of the picture. To elaborate, it is imagined that there is a circle in the trace plane whose centre is at the view centre and whose radius is a unit length (as used in relation to the object coordinate system). This circle is referred to as the unit circle, Fig 2.76. The scaling of the trace for production of the picture is chosen such that the radius of the image of the unit circle in the picture plane is a millimetres, Fig 2.76.

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

\[ \text{VG}(x_1, y_1, z_1, x_2, y_2, z_2, r) \]

where VG stands for view gauge and where \( x_1, y_1, z_1, x_2, y_2 \) and \( z_2 \) are numeric expressions whose values are the coordinates, relative to the object coordinate system, of the two points of a line segment in the object space which is referred to as the view gauge, Fig 2.77. The term \( r \) is a numeric expression whose value controls the size of the picture. To elaborate, the scaling of the trace for production of the picture is chosen such that the length of the image of the view gauge in the picture plane is in \( r \) millimetres, Fig 2.77.

The mode of controlling the position and the size of the picture through the concepts of view nave, view base and view scale (or view gauge) is referred to as the nave.
Fig 2.77
mode. The second method of controlling the size and the position of the picture is referred to as the range mode. This method allows automatic positioning and scaling of the picture. To elaborate, let the object and the view rise be as shown in Fig 2.78. Consider a rectangle in the trace plane where the sides of the rectangle are just touching the trace. This rectangle is referred to as the trace range. The range mode operates by choosing the position and the size of the picture such that the image of the trace range in the picture plane fits the view frame, as shown in Figs 2.78 and 2.79.

The third method of controlling the size and the position of the picture is referred to as the zone mode. This method allows zooming effects and involves the definition of a rectangular solid in the object space. This rectangular solid is referred to as the view zone with its facets being either parallel or perpendicular to the object coordinate axes, Fig 2.80. The zone mode operates by choosing the position and the size of the picture such that the image of the view zone in the picture fits the view frame. Furthermore, only those parts of the object that are inside the view zone will be considered for production of the picture, Fig 2.80. The view zone is specified by a USE-item of the form

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

where VZ stands for view zone and where \( x_1, y_1, z_1, x_2, y_2 \) 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

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

where VM stands for view zone and where \( n \) is an integer expression whose value is 1, 2 or 3, specifying nave mode, range mode or zone mode, respectively.
In viewing a scene in the real world, one would only be able to see those parts of the scene that fall within the field of vision. This situation 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 2.81. The view angle is chosen to be π radians for parallel projection and 3 radians for perspective projection. The effect of the view field is that only those parts of the object (or view zone) that are inside the view field are considered for picture creation.

With the exception of the view gauge, every one of the USE-items described above has a default setting. The default setting becomes effective at the beginning of every Formian session and remains current unless changed through USE statements. The default settings represent the most commonly used settings which are listed in Table 2.13.

The default method of determining the size of the picture is the nave mode is through the concept of view scale. Thus, unless a view gauge is specified through a USE statement, picture creation will be in terms of view scale and this is the reason for the view gauge not having a default setting.

Now, suppose that the following sequence of statements is entered at the beginning of a Formian session

\[
A1 = \{(1,1,1; 3,1,1), [1,1,1; 2,2,2]\}\\
A2 = \text{RINID}(9,9,2,2) | \text{ROS}(1,2,2,2) | A1\\
A3 = A2 | \text{RINID}(8,8,2,2) | \text{ROS}(1,2,3,3) | [2,2,2; 4,2,2]\\
A4 = \text{RINID}(3,3,4,4) | [6,6,2]\\
A = \text{LUX}(A4) | \text{PEX} | A3\\
\text{DRAW} \ A
\]
The first five assignment statements create a formex variable A which represents a double layer grid with 576 elements and the DRAW statement causes a plan view of the plot of A to be drawn on the current graphics output medium. The result is the configuration shown in Fig 2.82. The details relating to the coordinate axes in Fig 2.82 are added separately and are not drawn automatically through the DRAW statement.

The viewing specifications that give rise to the configuration of Fig 2.82 are provided by the default settings of the USE-items. One may, however, change these settings to obtain other aspects of the plot of A. For instance, the statements

```
USE VT(2), VP(10,-6,40), VC(10,10,2), VR(10,10,2,10,10,3)
DRAW A
```

will produce the perspective view shown in Fig 2.83. The first item in the above USE statement changes the view type from parallel to perspective and the subsequent items specify a view point, a view centre and a view rise. The above USE statement may equivalently be given as

```
USE VT(2), VH(10,-6,40,10,10,2,10,10,3)
```

where the view point, view centre and view rise are specified together through the concept of view helm.

A different aspect of the plot of A is obtained by entering the statements

```
USE VP(5,0,30)
DRAW A
```

The result is shown in Fig 2.84. The viewing specifications for the configuration of Fig 2.84 are the same as those for Fig 2.83 except for the view point which has been
Fig 2.72

Fig 2.82
changed as shown in the above USE statement.

To restore the default settings for the USE-items, one can specify the actual settings and enter

\[
\text{USE } \text{VT}(1), \text{ VP}(0,0,10000), \text{ VC}(0,0,0), \text{ VR}(0,0,0,0,1,0)
\]

Alternatively, one may enter the statement

\[
\text{USE } &\text{VT}, \ &\text{VP}, \ &\text{VC}, \ &\text{VR}
\]

The ampersand symbol implies default settings. Thus \&VT, \&VP,\ldots,etc, specify default settings for the view type, view point,\ldots,etc. Also, the ampersand symbol by itself as a USE-item implies the collection of all default settings. For instance, the statement

\[
\text{USE } \&, \text{ VM}(3), \text{ VZ}(0,0,0,11,8,2.5)
\]

restores all the default settings and then changes the view mode to the zone mode and sets the view zone as shown. At this point, the statement

\[
\text{DRAW A}
\]

will produce the configuration shown in Fig 2.85. This configuration is the plan view of the part of the plot of EI that is inside the view zone.

Attention is now turned to the description of Formian retrocords. Two classes of retrocords are employed in Formian. Firstly, there are a number of standard retrocords that are incorporated in the Formian Interpreter and can be put into effect through standard USE-items. The second class of retrocords are referred to as supplementary retrocords. These are defined by program segments that are added to
Fig 2.85

n=1  O  n=6  □
n=2  ●  n=7  ■
n=3  ○  n=8  □
n=4  ○  n=9  □
n=5  empty circle  n=10  empty square

Fig 2.86

Fig 2.87
the Formian Interpreter.

There are a variety of tenon and frond styles that can be obtained by different settings for USE-items as described in the sequel.

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

\[ LW(r) \]

where \( LW \) stands for line width and \( r \) is a numeric expression. The value of \( r \) specifies the line width in millimetres. The default setting for the line width USE-item is \( LW(0.3) \).

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

\[ TS(n,d) \]

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 in Fig 2.86.

The value of \( d \) is a non-negative number that specifies the dimension of the symbol in millimetres (diameter for \( n=1 \) to 5 and length of side for \( n=6 \) to 10). 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

\[ LS(p) \]

where \( LS \) stands for line style and \( p \) is an integer expression. The value of \( p \) may be from 1 to 4 representing the following patterns:
The default setting for the line style USE-item is LS(1).

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

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 in the picture plane are P1, P2,...,Ps, Pt, then different values of n have the following implications:

\[ n=1 \quad \text{The frond consists of t tenons and t-1 line segments} \]
\[ P1-P2, P2-P3,\ldots, Ps-Pt. \]

\[ n=2 \quad \text{The frond consists of t tenons and t line segments} \]
\[ P1-P2, P2-P3,\ldots, Ps-Pt, Pt-P1. \]

\[ n=3 \quad \text{The constitution of the frond is the same as for n=2, with any area(s)} \]
\[ \text{enclosed by the line segments being infilled. The colour of the infill may} \]
\[ \text{be the same or different from those of the line segments and /or tenons, as} \]
\[ \text{determined by the colour USE-item, subject to the limitations of the type of output} \]
\[ \text{device as discussed later.} \]

The tenons will not be encroached upon by the line segments and/or infill. In particular, when the tenon style is TS(5,d) or TS(10,d) then an empty circle or empty square will be present at every one of the points P1, P2,...,Pt. The default setting for the frond style USE-item is FS(2).
The fount of textual material on graphical output is determined by a USE-item of the form

\[ TF(t, h, w) \]

where TF stands for text fount, \( t \) is an integer expression and \( h \) and \( w \) are numeric expressions. The value of \( t \) is a nonzero positive integer that specifies the typeface of the characters and 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 fount USE-item for each 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

\[ TG(p, q, a) \]

where TG stands for text guide and where \( p \), \( q \) and \( a \) are numeric expressions. The values of \( p \) and \( q \) specify the position of the bottom left corner of the first character of the text in the picture plane relative to the device coordinate system. The value of \( a \), interpreted as degrees, determines the orientation of the text as shown in Fig 2.87. 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

\[ C(n, h) \]

where C stands for colour and where \( n \) and \( h \) are integer expressions. The value of \( n \) may be from 1 to 5, specifying:

\[ n = 1 \quad \text{line} \]
The value of \( h \) is a nonzero positive integer specifying a hue. In addition to hue, when \( n=3 \) or \( n=5 \), that is for infill and background, the value of \( h \) may specify various styles of half-tone and hatching. The USE-item has 5 default settings relating to line, tenon, infill, text and background. These settings for each Formian installation are chosen to suit the installation's environment.

The colour USE-item as defined above is intended for colour graphics displays and devices with comparable capabilities. Thus, for some output media only a subset of the specifiable features may be applicable. For example, for a monochrome graphics output medium, the only relevant specifications may be the styles of half-tone and hatching for infill and background.

Pen plotters are a special case. The choice of a colour for a pen plotter may only be achieved through the selection of a pen and this will also dictate the line width. Therefore, colour and line width USE-items are not applicable to pen plotters.

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

\[
PEN(n)
\]

where \( n \) is an integer expression. The value of \( n \) is a nonzero positive integer that identifies one of the pens of the pen plotter. The default setting for the pen USE-item is PEN(1).
2.7 DATA SUBMISSION IN FORMIAN

Consider the double layer grid whose plan is shown in Fig 2.88, where the top layer elements are drawn in full line, the bottom layer elements are drawn in dashed line and the web elements are drawn in dotted line. The dimensions of the grid together with the global coordinate system x-y-z and the normat U1-U2-U3 for formex formulation are shown in Fig 2.88.

The grid is supported at eight bottom layer nodes indicated by little circles in Fig 2.88. The grid is to be analysed as a pin-jointed linear structure. The computed program for the analysis is assumed to be based on the standard stiffness method using the banded solution technique for simultaneous equations.

There are three different element cross-sections, two different support types and one loading case as specified below:

Element type 1: top layer element with axial rigidity EA1 (indicated by ___________ ),

Element type 2: bottom layer element with axial rigidity EA2 (indicated by - - - - - - - ),

Element type 3: web element with axial rigidity EA3 (indicated by ..............),

Support type 1 (indicated by ⬤): translational constraints in x, y and z directions,

Support type 2 (indicated by O): translational constraints in z direction only,
4 divisions @ 2.5m = 10.0m

z and U3 are upward

U3=0, z=0 for bottom layer
U3=1, z=1m for top layer

Fig 2.88

Fig 2.90
Load case 1:

1) vertical downward point loads of 5.0 kN applied at the four top layer corner nodes,

2) vertical downward point loads of 7.5 kN applied at all the non-corner top layer nodes along the edges and

3) vertical downward point loads of 10.0 kN applied at all the top layer nodes that are not along the edges.

The complete formulation of data for the analysis is given in Fig 2.89. The formulation consists of a sequence of Formian statements that are enclosed in quote symbols. The axial rigidities $EA_1$, $EA_2$ and $EA_3$ are left as parameters in the formulation and are listed at the top, enclosed in colons. The construct that is enclosed in quote symbols in Fig 2.89 is referred to as a scheme in Formian. The scheme in Fig 2.89 is given the name GRID and six parts are identified in it.

```
GRID = ':EA1, EA2, EA3:
TOP=PEX|RINID(4,4,2,2)|ROSAD(1,1)|[0,0,1; 2,0,1]
Part 1 BOT=PEX|RINID(3,3,2,2)|ROSAD(2,2)|[1,1,0; 3,1,0]
WEB=RINID(4,4,2,2)|ROSAD(1,1)|[0,0,1; 1,1,0]
Part 2 S1={[1,3,0], [3,1,0]}
S2=LUX(S1)|ROSAD(4,4)|S1
Part 3 L1=ROSAD(1,1)|[0,0,1]
L2=ROSAD(4,4)|RIN(1,3,2)|[2,0,1]
L3=RINID(3,3,2,2)|[2,2,1]
Part 4 USE BT(2.5,2.5,1)
```
In part 1, formex variables TOP, BOT, WEB are created. These variables represent, the top layer elements, bottom layer elements and web elements, respectively.

Formex variables S1 and S2, representing support nodes of types 1 and 2, respectively, are created in Part 2 and formex variables L1, L2 and representing the loaded nodes for load case 1 are created in Part 3.

The formices in Parts 1 to 3 of the scheme of Fig 2.89 are formulated in terms of the normat U1-U2-U3 in Fig 2.88. However, in addition to the information provided by these formices, it is necessary to know the scale factors in the first, second and third directions, so that the actual x, y and z coordinates of the nodes can be determined. These scale factors are specified in Part 4 of the scheme.

The node numbering pattern of the structure is specified in Part 5 of the scheme. Here, the formex variable N represents all the nodal points of the grid given in the order in which the nodes are required to be numbered for the purposes of structural analysis. N is referred to as the numerant of the structure. The node numbering pattern represented by N is shown in Fig 2.90.

2.7.1 SUBMISSION PLENIX

The SUBMIT statement in Part 6 of the scheme of Fig 2.89 has the effect of
generating input data for the analysis program. The construct that follows the keyword SUBMIT is referred to as a submission plenix. A plenix (plural plenices) is a data structure that admits various mathematical entities as its components, where a component of a plenix is referred to as a panel. The submission plenix in Fig 2.89 consists of five principal panels as follows:

{codet panel, numerant panel, element panel, support panel, load panel}

Leaving the codet panel to be described later, the numerant panel, in the scheme of Fig 2.89 simply consists of the formex variable N that represents the numerant of the structure.

**2.7.2 ELEMENT PANEL**

The element panel in Fig 2.89 is given as:

{TOP, BOT, WEB [EA1; EA2; EA3]}

The panel itself is a plenix listing the formex variables that represent the elements of different types. These formex variables are followed by a panel that gives the axial rigidities of the elements, in the same order as the corresponding formex variables.

Input data for the analysis regarding element connectivity together with cross-sectional and material properties will be generated using the element panel in conjunction with the numerant panel. To elaborate, the formex variable N will be used as the parameter of the dictum function to transform the values of the formex variables TOP, BOT, WEB into element connectivity information in terms of the node numbering pattern of Fig 2.90. Also, the seviation function will be used to determine the band-width of the stiffness matrix of the structure.
It will not be necessary to explicitly specify the total number of elements or the number of different element types or the number of elements of each type, since these items of information may be deduced from the element panel. Also, explicit specification of the total number of nodes is not required, since this information may be deduced from the numerant panel.

The last panel of the element panel, that is

\[ \{E_{A1}; E_{A2}; E_{A3}\} \]

is referred to as element quantic panel. In general, the element quantic panel provides all the necessary information about the cross-sectional and material properties of the elements. In the present example, the grid is required to be analysed as a linear pin-jointed structure. In this situation, the axial rigidity of an element is the only necessary item of information regarding the cross-sectional and material properties of the element. Therefore, the element quantic panel in Fig 2.89 contains only axial rigidities.

### 2.7.3 SUPPORT PANEL

The fourth principal panel of the submission plenix of Fig 2.89 is the support panel. This panel is given as

\[ \{S_{1}, S_{2}, [1,1,1; 0,0,1]\} \]

Here, \(S_{1}\) and \(S_{2}\) are formex variables representing the support nodes of types 1 and 2, respectively. These variables will be used to generate information about the support nodes, relative to the numbering pattern of Fig 2.90 in the same manner as discussed previously in relation to element connectivity. The formex variables \(S_{1}\) and \(S_{2}\) are followed by the support quantic panel.
In general, a panel is referred to as a quantic panel provided that it contains quantitative information about such details as material properties, cross-sectional properties, constraint patterns and load magnitudes. The support quantic panel provides information about the constraints at the support nodes. Each row of this quantic panel has three entries, reflecting the fact that a node in a pin-jointed space structure has three degrees of freedom. In this context, a 1 is used to imply presence of constraint and a 0 is used to imply absence of constraint.

2.7.4 LOAD PANEL

The last principal panel of the submission plenix is the load panel which is given as

\[ \{L_1, L_2, L_3, [0, 0, -5; 0, 0, -7.5; 0, 0, -10]\} \]

Here the formex variables \(L_1, L_2\) and \(L_3\) represent the loaded nodes of the grid for load case 1. These variables will be used to generate information about the loaded nodes relative to the numbering pattern of Fig 2.90. The formex variable \(L_1, L_2\) and \(L_3\) are followed by the load quantic panel.

2.7.5 CONCEPT OF AN ALBUM

The execution of the SUBMIT statement will result in the creation of a collection of files in the repository. Such a collection of files is referred to as an album, where it is convenient to think of an album as being in the form of a plenix. Each album has an associated identifier that is referred to as its title.

The creation of an album involves two phases. During the first phase, the information provided by the submission plenix is used to create a plenix of the form

\[ \{\text{data panel, folio panel}\} \]
The data panel is a plenix itself and contains a complete set of data for the analysis of the structure under consideration. The information in the data panel is in a form that can be used directly for input to a structural analysis program. The folio panel contains information about the manner in which the data or the final analytical results may be outputted.

Once the data panel and the folio panel are formed, the first phase of the creation of the album is complete. The album at this stage is referred to as a phase 1 album.

The second phase of the creation of an album involves the input of the data panel to a structural analysis program and the acquisition of the analytical results. The album at this stage is in the form

\{\text{data panel, folio panel, result panel}\}

where the data panel and the folio panel are as described above and where the result panel is a plenix containing all the analytical results. The album at this stage is referred to as a phase 2 album.

### 2.7.6 CODET ALBUM

Returning to the description of the scheme of Fig 2.89, it is now possible to discuss the constitution of the first principal panel of the submission plenix. The panel is referred to as the codet panel and is of the form

\{\text{plenix codet, title codet, operation codet, destination codet, structure codet, analysis codet}\}

where the term codet is used to refer to an item of information that provides a guideline for processing of the submission plenix.
2.7.7 PLENIX CODET

The plenix codet is an integer expression whose value specifies the type of the submission plenix. To elaborate, there are many different forms of submission plenix each of which is suitable for a particular situation. The submission plenix in Fig 2.89 is intended for simple structural analysis and is designated as type 1.

2.7.8 TITLE CODET

The title codet is an identifier that acts as a title for the album which is to be created. The title codet in the submission plenix of Fig 2.89 is chosen to be GRIDA.

Album titles are unrelated to variables and covariables. Thus, a variable, a covariable and an album may use the same identifier simultaneously without any problem, since the context in which the identifier appears will always determine the type of entity it is representing without any ambiguity. However, if an album is to be created which has the same title as an existing album, then the new album will replace the old one.

2.7.9 OPERATION CODET

The third panel of the codet panel is the operation codet. This is an integer expression whose value is either 1 or 2. A value of 1 for the operation codet implies that it is required to create a phase 1 album and a value of 2 for the operation codet implies that it is required to create a phase 2 album. To elaborate, in some cases one would like to create the data to begin with and carry out the analysis at a later stage. For instance, one may like to examine the generated data before proceeding with the analysis. In such a case, one would give the value of the operation codet as 1. In some other cases, one may wish to complete the data generation and the analysis in
one go. In such a case, one would choose the value of the operation codet as 2. In the submission plenix of Fig 2.89, the operation codet is given as 2.

In a case when an existing phase 1 album is to be submitted for analysis, the following Formian statement may be used

\[
\text{SUBMIT} <\text{TITLE}>
\]

where, TITLE is expected to be an identifier that is the title of an existing phase 1 album. The effect of the execution of the above statement is to transform the given phase 1 album into a phase 2 album. The construct

\[
<TITLE>
\]

is referred to as an album denoter.

The deletion of an existing album is achieved through an ERASE-item which is in the form of an album denoter. Thus, if DOME is the title of phase 1 or phase 2 album, then the execution of the statement

\[
\text{ERASE} <\text{DOME}>
\]

will cause the album DOME to be deleted from the repository.

2.7.10 DESTINATION CODET

The fourth panel of the codet panel is the destination codet. This is an integer expression whose value specifies the program to which the data is to be submitted. The destination codet in the scheme of Fig 2.89 is given as 7 and this is assumed to designate the program which is intended to be employed for the analysis.
2.7.11 STRUCTURE CODET

The fifth panel of the codet panel is the structure codet. This is an integer expression whose value specifies the type of the structure which is to be analysed. For instance, the integer 2 that is chosen as the structure codet in the scheme of Fig 2.89 indicates that the structure is a pin-jointed space structure.

Some common structure types and their corresponding codets are listed below:

- pin-jointed plane structure 1
- pin-jointed space structure 2
- rigidly jointed plane structure 3
- grillage 4
- rigidly jointed space structure 5

2.7.12 ANALYSIS CODET

The last panel of the codet panel is the analysis codet. This is an integer expression whose value specifies the type of analysis to be performed. For instance, the integer 1 given as the analysis codet in the scheme of Fig 2.89 indicates that linear analysis is to be performed.

2.7.13 INDUCTION OF THE SCHEME

The discussion so far was concerned with a description of the component parts of the scheme of Fig 2.89. It is now necessary to describe the manner in which the scheme GRID may be used to initiate the actual process of data generation and analysis. The initiation is achieved by inducing the scheme. The scheme GRID is induced by a statement of the form
GRID(42000, 63000, 24000)

This is called an induction statement with the items listed in parentheses being referred to as induction parameters. The induction parameters here correspond to the nominal parameters EA1, EA2 and EA3 in the heading of the scheme of Fig 2.89. The effect of the above induction statement is to execute the assignment statements

\[ EA1=42000; \ EA2=63000; \ EA3=24000; \]

and then execute all the statements in the body of the scheme of Fig 2.89. As a result the variables in the scheme (such as TOP, BOT, ...) will be assigned values and the execution of the SUBMIT statement will create the album GRIDA.

2.7.14 OUTPUT OF INFORMATION

The output of information about data and analytical results may be achieved through lineation and listing functions. The lineation function is a mechanism that allows data and analysis results to be outputted in a graphical manner. The listing function is similar to the lineation function but is for textual output.

The lineation function is a construct of the form

\[ \text{LIN}(N) \]

where LIN is an abbreviation for lineation and where N is an integer expression whose value indicates the type of output required. Specifically,

\[ N=1 \text{ indicates node numbers} \]
\[ N=2 \text{ indicates element types and numbers,} \]
\[ N=3 \text{ indicates constraints,} \]
N=4 indicates loads,
N=5 indicates nodal displacements,
N=6 indicates element forces,
N=7 indicates element stresses and
N=8 indicates reactions.

A lineation function designator is a construct of the form

\[ \text{LIN}(N)|E \]

where E is a formex expression whose value represents the nodes or the elements for which output is required. The value of E is a formex which is relative to the normat in terms of which the formex formulation for the analysis is carried out. For instance, in relation to the example of Fig 2.88, the lineation function designator for stresses in the top layer elements will be

\[ \text{LIN}(7)|\text{TOP} \]

In a case when a particular type of output is required for the whole of the structure, then the argument of the lineation function designator may be omitted. For instance, if it is required to output nodal displacements for the whole of the structure of Fig 2.88, then the lineation function designator will take the simple form

\[ \text{LIN}(5) \]

A lineation function designator may only appear as a DRAW-item or a SHOW-item. The execution of a DRAW or SHOW statement that involves a lineation function designator will have the effect of producing the output indicated by N on the appropriate graphic output medium. For instance, in the case of the example of Fig 2.88, the statement
DRAW LIN(I)|MED|TOP

can give rise to the output shown in Fig 2.91. The term lineation may also be used to refer to a graphical output produced through a lineation function. Thus, the output shown in Fig 2.91 can be referred to as lineation.

2.7.15 CONCEPT OF FOLIO

The production of a lineation requires specification of a number of details. For instance, in producing the lineation of Fig 2.91, it is necessary to have information about the required size, style and positions of the node numbers. Also, the required style and thickness of the lines for the outline of the top layer elements need specification.

Specifications of this kind are made through folios. A folio is a collection of specifications for production of a particular type of output. The lineation function has eight associated folios each of which corresponds to one of the values of N described above.

During the phase 1 of the setting up of an album, a complete set of folios for various kinds of output is created. This set includes a common folio which contains some general specifications relating to all types of output. This set of folios is referred to as default folios, since the specifications in them are based on a number of default settings.

If the default folio for a particular output is satisfactory, then one can proceed with the production of the required lineation. However, one has the option of altering the default settings of a folio to suit a particular situation. This may be achieved through a TAKE-item of the form

<N>
Fig 2.91

Fig 2.92
where \( N \) is an integer expression whose value is greater than or equal to zero. When \( N > 0 \) then \(<N>\) denotes the folio relating to the \( N \)-th category of output as described for the lineation function above and when \( N = 0 \) then \(<N>\) denotes the common folio of the album. The effect of the appearance of \(<N>\) as a TAKE-item is to display the corresponding folio on the screen of the terminal. This folio may be modified, interactively, as required and the old version of the folio may then be replaced by the modified one. The construct

\(<N>\)

is referred to as a folio denoter.

For instance, in relation to the example of Fig 2.88 the statement

\[
\text{TAKE } <1>
\]

will result in displaying the folio for output of node numbers. One may then change the specifications by choosing a larger size for the node numbers and the dotted line style for the outline of the members and a subsequent statement

\[
\text{DRAW LIN(1)|MED|BOT}
\]

will produce the lineation shown in Fig 2.92.

2.7.16 OUTPUT OF ANALYTICAL RESULTS

The statement

\[
\text{DRAW LIN(5)|MED|TOP, LIN(6)|TOP}
\]

provides an example of the use of the lineation function for the output of analytical
results. The lineations shown in Figs 2.93 and 2.94 are the results of the execution of the above DRAW statement. Fig 2.93 gives the $z$ deflections of the top layer nodes of the grid of Fig 2.88.

If the absolute value of the maximum $z$ deflection for all the top layer nodes is denoted by $D$ and if the $z$ deflection of a typical node is denoted by $d$, then the integer number written near this node in Fig 2.93 is given by

$$1000\left(\frac{d}{D}\right)$$

This integer number is referred to as an indicant and represents the deflection as a permillage of the maximum deflection. Permillage is similar to percentage but it consists of parts out of thousand rather than parts out of hundred.

The number which is given as a factor below the lineation of Fig 2.93 has the value

$$D/1000$$

To obtain the actual value of a deflection it is necessary to multiply the corresponding indicant by the factor.

In the lineation of Fig 2.93, a positive indicant implies movement in the positive direction of the $z$ axis, that is, upward, and a negative indicant implies movement in the negative direction of $z$ axis, that is, downward. The method of presentation of the values of the deflections employed in Fig 2.93 is very convenient since the relative sizes of the deflections are deduced straight away and that one has the convenience of working with simple integers.

The lineation in Fig 2.93 also contains a graphical representation for the deflections. This is achieved by placing little circles at the nodal points with the sizes of the circles representing the relative value of the deflections. Here, a solid circle
Fig 2.93

Factor = 1.03648E-01

Fig 2.94

Factor = 3.9660E+01
represents a downward deflection and a hollow circle represents an upward deflection. The details regarding the choices for the size and the style of the indicants and the maximum size of the circles are made through the 5th folio, that is, the folio for nodal displacements. The folio is also used to specify the z component of displacement for output.

The concepts of indicant and factor are also employed in Fig 2.94 for the axial forces in the top layer elements of the grid in Fig 2.88. In the lineation of Fig 2.94, a positive indicant implies a tensile force and a negative indicant implies a compressive force. The information provided by the indicants is complemented by graphical representation of forces. This is achieved by placing a rectangle in the position of each member, where the width of the rectangle is a measure of the relative size of the force in the member. A tensile force is represented by a hollow rectangle and a compressive force is represented by a solid rectangle.

Indicants and factors may also be used for the representation of loads, reactions and stresses. For example, the DRAW statement

```
DRAW LIN(4)|MED|TOP, LIN(3)|(S1#S2)
```

can give rise to the lineation shown in Figs 2.95 and 2.96. The lineation of Fig 2.95 gives the external loads for the grid of Fig 2.88 and the lineation of Fig 2.96 gives the constraints.

In Fig 2.96 the sequence of zeros and ones placed near each node represents the constraint pattern for that node. Each one of the sequences consists of three digits, where the 1st, 2nd and 3rd digits refer to the degrees of freedom in the x, y and z directions, respectively. A digit one in the sequence indicates the presence of constraint and a digit zero indicates the lack of constraint.

The digit sequences in Fig 2.96 are accompanied by graphical representation of
Fig 2.95

Factor = 1.0000E+01

Fig 2.96
constraint patterns. This graphical representation consists of pie charts placed at the nodes. Each of these pie charts has three sectors. Starting from the twelve o’clock position of a pie chart and proceeding in a clockwise manner, the sectors of the chart refer to the degrees of freedom in the x, y and z directions, respectively. A solid sector implies the presence of constraint and a hollow sector implies the lack of constraint.

2.7.17 CURRENT ALBUM

In the above discussion, it is tacitly assumed that there is only one album in the repository, so that there is no ambiguity regarding the album which is to be used for output. However, in general, the repository contains many albums each of which relates to a particular project. Among the set of albums that are in the repository at any given time, there is one album that has the status of being the current album and information for output is always taken from the current album. When an album is created it acquires the status of being the current album and it retains this status until such time when

either another album is created, assuming the status of the current album

or another album is made current through a USE-item.

The USE-item which may be used to make an album current is simply an album denoted. Thus, if VAULT is the title of a phase 1 or phase 2 album in the repository, then the execution of the statement

USE <VAULT>

will have the effect of making VAULT the current album.
2.7.18 CURRENT LOAD CASE

The example of Fig 2.88 involves a single load case and the information displayed in Figs 2.92 and 2.93 relates to this load case. However, in general, the number of load cases may be more than one load case, then at any given time one of these has the status of the current load case and this is the load case that will be used for the output of information. The assumption is that the first specification is made to the contrary. Such a specification may be through the common folio of the current album.
Chapter 3

Tractation Retronorm

INTRODUCTION

In dealing with the formex formulation of a configuration, it is usual to begin by formulating a topological description of the configuration using formex functions. The next stage involves the employment of a transformation for associating geometric coordinates with nodes of the configuration. A transformation of this kind is referred to as a retronorm. Two categories of retronorms are employed in Formian. Firstly, there are ten standard retronorms that are incorporated in the Formian Interpreter. The second category of retronorms are referred to as supplementary retronorms.

A supplementary retronormic function is introduced through a program segment
which is supplied by the user in order to create a nonstandard retronorm. The program segment is linked to the body of the Formian Interpreter.

This study deals with the implementation of a supplementary retronormic function called the "tractation retronorm". The tractation retronorm enables a configuration to be projected on different types of surfaces such as spheres, ellipsoids, paraboloids, cylinders, hyperbolic paraboloids or planes. The term tractation is used to imply projection of a configuration on a surface or surfaces. Tractation is derived from the Latin word "tractus" meaning "drawing". The projection may be central, axial, parallel or radial and these types of projection will be discussed in due course.

3.1 AN EXAMPLE

The applications of the tractation retronorm may be described with the help of an example. Consider a single layer grid pattern which is to be projected on a sphere. The grid, together with the global coordinate system x-y-z and the normat U1-U2-U3 for the formex formulation are shown in Fig 3.1. The grid may be represented in terms of a formex F where

\[
F = \text{PEX} | \text{LUX} | \text{ROSAD}(10,10) | \text{GENID}(3,3,2,2,0,1) | [0,0] | ..
\]

\[
\text{RINID}(10,10,2,2) | \text{ROSAD}(1,1) | [[0,0;2,0],[0,0;1,1]]
\]

Figure 3.2 shows the plan view and elevation of the grid, together with the sphere on which the projection is to be carried out. For the specification of the sphere, the centre C(xc,yc,zc) and the radius R, have been defined. The centre of projection is specified as the point A(xa,ya,za).

The above specifications, together with another three parameters may be given by the statement
Fig 3.1

Fig 3.2

Fig 3.4
These three parameters may be the type of projection, the type of surface and the required solution. The first parameter is referred to as "projection specifier" and the second parameter is referred to as "surface specifier". The last parameter is referred to as the "selector" and will be described in detail in section 3.5.

TRAC in Fig 3.3 is an abbreviation for "tractation" and is followed by a sequence of parameters enclosed in parentheses. The construct that consists of TRAC together with the ensuing parameter list is referred to as the "tractation retronorm" and what is enclosed in parentheses is called the "descriptor". A general descriptor is of the form given in Fig 3.4. The significance of the values of the projection specifier P, the centre of projection, the axis or direction of projection, the selector and the coefficients of the surface are discussed in detail in the sequel. However, as far as the significance of the values of h and U these will be discussed in detail in section 3.11. A graphical representation of formex D, projected on the sphere is shown in Fig 3.5.

3.2 PROJECTION SPECIFIER

The projection specifier P in Fig 3.3 is given as 1. The centre of projection A(xa,ya,za) is specified by the coordinates 10, 10, and -10 relative to the global
coordinate system. The projection specifier $P$ is a nonzero positive integer defining the type of projection. The value of the projection specifier $P$ may be from 1 to 4, as follows

- $P=1$ indicating central projection
- $P=2$ indicating parallel projection
- $P=3$ indicating axial projection
- $P=4$ indicating radial projection

Now, the different types of projection may be described in terms of the mathematical approach as given below.

### 3.2.1 CENTRAL PROJECTION

The central projection of a point $T(U_1, U_2, U_3)$ in space onto a surface $S$, is the point $P(x, y, z)$ produced by the intersection of the line $AT$, with the given surface $S$, as shown in Fig 3.6. Where, $A$ is a fixed point, which it may be referred to as the centre of projection given by the coordinates $x_a$, $y_a$ and $z_a$. The equation of the straight line $AT$ may be given as

$$\frac{x-x_a}{U_1-x_a} = \frac{y-y_a}{U_2-y_a} = \frac{z-z_a}{U_3-z_a}$$

### 3.2.2 PARALLEL PROJECTION

The parallel projection of a point $T(U_1, U_2, U_3)$ in space onto a surface $S$, is the point $P(x, y, z)$ which is the intersection of the line that passes through $T$ and is parallel to the line $AB$, as shown in Fig 3.7. The line $AB$ defining the direction of projection is given by the points $A(x_a, y_a, z_a)$ and $B(x_b, y_b, z_b)$. 

204
A(xa, ya, za)

**Fig 3.6**

**Fig 3.7**

C is the centre of sphere

**Fig 3.8**

**Fig 3.9**

**Fig 3.10**
To begin with, the mathematical definition of two parallel lines has been given. Two lines in space are subsets of parallel lines, if and only if the direction numbers for one line are direction numbers for the other.

As mentioned earlier the direction of projection is defined by the straight line joining points A(xa, ya, za) and B(xb, yb, zb). The equation of line AB may be given as

\[
\frac{x-x_a}{xb-xa} = \frac{y-y_a}{yb-ya} = \frac{z-z_a}{zb-za} \ldots \land \ldots \land \begin{array}{c}
xb-xa \neq 0, yb-ya \neq 0, za-zb \neq 0
\end{array}
\]

and whose direction numbers are

\[
\begin{align*}
l_1 &= xb-xa \\
m_1 &= yb-ya \\
n_1 &= zb-za
\end{align*}
\]

Since AB must be parallel to the line passing through point T(U1, U2, U3) then the equation of this line may be given by

\[
\frac{x-U_1}{l_1} = \frac{y-U_2}{m_1} = \frac{z-U_3}{n_1}
\]

Substituting Eqns 1 in the above equation will give

\[
\frac{x-U_1}{xb-xa} = \frac{y-U_2}{yb-ya} = \frac{z-U_3}{zb-za}
\]

Now, the procedure for parallel projection becomes identical to the central projection. Each point belongs to the line that passes through point T(U1, U2, U3) may be considered as the centre of projection.

Consider the single layer grid pattern which is shown in Fig 3.1. Suppose that the grid is to be projected, using parallel projection, on a sphere. A Formian statement describing this operation may be given as
The projection specifier is given as 2, where 2 implies parallel projection. The direction of projection has been defined by the line joining points A(10,10,-10) and B(10,10,-30). Figure 3.8 shows the plan view and elevation of the grid together with the sphere on which the projection is to be carried out. A graphical representation of the grid pattern projected on the sphere using parallel projection is as shown in Fig 3.9.

3.2.3 AXIAL PROJECTION

The axial projection of a point T(U1,U2,U3) in space onto a surface S is the point P(x,y,z) which is the intersection of the line that passes through point T and is perpendicular to line AB at K(xk,yk,zk), as shown in Fig 3.10. The line AB is referred to as the axis of projection given by the points A(xa,ya,za) and B(xb,yb,zb).

To deal with axial projection the coordinates of point K(xk,yk,zk) have to be specified. To begin with, the mathematical definition of two perpendicular lines has been given.

Let two lines in space have direction numbers l1, m1, n1 and l2, m2, n2, respectively. These lines are perpendicular to each other if and only if their direction numbers satisfy the equation
Now, consider a straight line through points \(T(U_1, U_2, U_3)\) and \(K(x_k, y_k, z_k)\) whose equation is given by

\[
\frac{x - x_k}{U_1 - x_k} = \frac{y - y_k}{U_2 - y_k} = \frac{z - z_k}{U_3 - z_k}
\]

and whose direction numbers are

- \(l_1 = U_1 - x_k\)
- \(m_1 = U_2 - y_k\)

and

- \(n_1 = U_3 - z_k\)

As mentioned earlier the axis of projection is defined by the line joining \(A(x_a, y_a, z_a)\) and \(B(x_b, y_b, z_b)\). The equation of line \(AB\) may be given as

\[
\frac{x - x_a}{x_b - x_a} = \frac{y - y_a}{y_b - y_a} = \frac{z - z_a}{z_b - z_a}
\]

and whose direction numbers are

- \(l_2 = x_b - x_a\)
- \(m_2 = y_b - y_a\)

and

- \(n_2 = z_b - z_a\)

Since \(KT\) and \(AB\) must be perpendicular to each other then

\[
(U_1 - x_k)(x_b - x_a) + (U_2 - y_k)(y_b - y_a) + (U_3 - z_k)(z_b - z_a) = 0
\]

As \(K\) is on the line \(AB\), the coordinates of \(K\) will satisfy Eqn 2 and may be written as
\[
\frac{x_k-x_a}{x_b-x_a} = \frac{y_k-y_a}{y_b-y_a} = \frac{z_k-z_a}{z_b-z_a} \quad \ldots \ldots \quad 4
\]

from which the following equations may be obtained

\[
\frac{x_k-x_a}{x_b-x_a} = \frac{z_k-z_a}{z_b-z_a}
\]

and

\[
\frac{y_k-y_a}{y_b-y_a} = \frac{z_k-z_a}{z_b-z_a}
\]

After some manipulations, one obtains

\[
x_k = \frac{x_b-x_a}{z_b-z_a} \cdot (z_k-z_a) + x_a \quad \ldots \ldots \quad 4a
\]

and

\[
y_k = \frac{y_b-y_a}{z_b-z_a} \cdot (z_k-z_a) + y_a \quad \ldots \ldots \quad 4b
\]

Substituting Eqns 4a and 4b in Eqn 3 the coordinate \(z_k\) is obtained as

\[
z_k = \frac{a + za \cdot b}{c}
\]

where

\[
a = (x_b-x_a) \cdot (U_1-x_a) + (y_b-y_a) \cdot (U_2-y_a) + U_2 \cdot (z_b-z_a)
\]

\[
b = \frac{(x_b-x_a)^2 + (y_b-y_a)^2}{z_b-z_a}
\]

and

\[
c = b + z_b-z_a
\]
The values of $x_k$ and $y_k$ are then obtained as

$$x_k = \frac{xb-xa}{zb-za} \cdot (zk-za) + xa$$

and

$$y_k = \frac{yb-ya}{zb-za} \cdot (zk-za) + ya$$

The equation of the straight line perpendicular to the line AB may be given as

$$\frac{x-x_k}{U1-x_k} = \frac{y-y_k}{U2-y_k} = \frac{z-z_k}{U3-z_k}$$

If the coordinates of K are specified the procedure for axial projection becomes identical to procedure for central projection. The point K is now the centre of projection, given by the coordinates $x_k$, $y_k$ and $z_k$.

Consider the single layer grid pattern which is shown in Fig 3.1. Suppose that the grid is to be projected, using axial projection, on a sphere. A Formian statement describing this operation may be given as

```
D=TRAC(3,0,0,-10,20,20,-10,1,10,10,-5,20,13)|F
```

The projection specifier is given as 3, where 3 implies axial projection. The axis of projection has been defined by the line joining points A(0,0,-10) and B(20,20,-10). Figure 3.11 shows the plan view and elevation of the grid with the sphere on which

210
Fig 3.11

C is the centre of sphere
AB is the axis of projection

Fig 3.12

Fig 3.13

Fig 3.14
the projection is to be carried out. A graphical representation of the grid pattern projected on the sphere using axial projection is as shown in Fig 3.12.

3.2.4 RADIAL PROJECTION

Very often the centre of projection coincides with the centre of a surface and radial projection is used. The radial projection is a special case of central projection and is only applicable in relation to a surface S that has a centre such as a sphere or an ellipsoid. In radial projection the centre of projection A(xa,ya,za) coincides with the centre C(xc,yc,zc) of the given surface S, and the projection P(x,y,z) of a point T(U1,U2,U3) on S is obtained as the intersection with S of a straight line that connects T with the centre of the surface as shown in Fig 3.13. The equation of the straight line TC may be given as

\[
\frac{x-xc}{U1-xc} = \frac{y-yc}{U2-yc} = \frac{z-zc}{U3-zc}
\]

Consider the single layer grid pattern which is shown in Fig 3.1. Suppose that the grid is to be projected, using radial projection, on a sphere. A Formian statement may be given as

D=TRAC(4,1,10,10,-5,20,13)|F

The projection specifier is given as 4, where 4 implies radial projection. The centre of projection coincides with the centre of the sphere C(10,10,-5). Figure 3.14 shows the plan view and elevation of the grid, together with the sphere on which the projection is to be carried out. A graphical representation of the grid projected on
the sphere using radial projection is as shown in Fig 3.15.

3.3 SURFACE SPECIFIER

The surface specifier $S$ in Fig 3.3 is given as 1, where 1 stands for a sphere. The surface specifier $S$ is an integer expression whose value is a nonzero positive integer defining the surface of projection. The value of the surface specifier $S$ may be from 1 to 9 representing the following surfaces:

- $S=1$ Sphere
- $S=2$ Ellipsoid
- $S=3$ Elliptic Paraboloid
- $S=4$ Hyperbolic Paraboloid
- $S=5$ Circular Cylinder
- $S=6$ Elliptic Cylinder
- $S=7$ Parabolic Cylinder
- $S=8$ Plane specified by two points that define a vector normal to the plane
- $S=9$ Plane specified by three points

3.4 COEFFICIENTS OF A SURFACE

The description of a surface requires a number of coefficients that should follow the surface specifier. These are given below for different surfaces:

Sphere

the $xc$, $yc$ and $zc$ coordinates of the centre of the sphere followed by the radius $R$ of the sphere
Ellipsoid
the xc, yc and zc coordinates of the centre of the ellipsoid followed by three semiaxes a, b and c of the ellipsoid in the x, y and z directions

Elliptic Paraboloid
the xc, yc and zc coordinates of the centre of the elliptic paraboloid followed by three semiaxes a, b and c of the elliptic paraboloid in the x, y and z directions

Hyperbolic Paraboloid
the three semiaxes a, b and c of the hyperbolic paraboloid in the x, y and z directions

Circular Cylinder
the xc and yc coordinates of the centre of the directrix circle of the cylinder followed by the radius R of the directrix circle

Elliptic Cylinder
the xc and yc coordinates of the centre of the directrix ellipse of the cylinder followed by two semiaxes a and b of the directrix ellipse in the x and y directions

Parabolic Cylinder
the xc and yc coordinates of the centre of the directrix parabola of the cylinder

Plane specified by two points that define a vector normal to the plane
the x, y and z coordinates of two points N(n1,n2,n3) and T(t1,t2,t3) that define a vector normal to the plane

Plane specified by three points
the x, y and z coordinates of three points, (x1,y1,z1), (x2,y2,z2) and
3.5 SELECTOR

The projection of a point on a surface usually involves a quadratic equation. The value of the parameter $t$ specifies the course of action to be taken when the projection of a point cannot be determined uniquely and is referred to as "selector". To elaborate, in obtaining the projection $P$ of a point $T$ on a surface $S$, the following situations may arise:

i) $P$ is determinable uniquely, Fig 3.16
ii) $P$ is nonexistent, Fig 3.17
iii) there is more than one solution for $P$, Fig 3.18

Various possible courses of action in the case of nonexistent or multiple solutions for $P$, together with the corresponding section codes are listed in Table 3.1.

3.6 SPHERE

The sphere is a closed surface that is the locus of all points that are at a fixed distance $R$ (the radius) from a given point $C$ (the centre), Fig 3.19. In Cartesian coordinates, the equation of a sphere may be obtained as follows:

A. If the centre of the sphere is the origin, Fig 3.19a, the equation is given as

$$x^2 + y^2 + z^2 = R^2$$

B. If the centre of the sphere is the point $C(xc, yc, zc)$, Fig 3.19b, the equation is given as
### Table 3.1

<table>
<thead>
<tr>
<th>Selection Code</th>
<th>Indicated Solution</th>
<th>Selection Code</th>
<th>Indicated Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>greatest x component</td>
<td>21</td>
<td>greatest x component</td>
</tr>
<tr>
<td>-11</td>
<td>least x component</td>
<td>-21</td>
<td>least x component</td>
</tr>
<tr>
<td>12</td>
<td>greatest y component</td>
<td>22</td>
<td>greatest y component</td>
</tr>
<tr>
<td>-12</td>
<td>least y component</td>
<td>-22</td>
<td>least y component</td>
</tr>
<tr>
<td>13</td>
<td>greatest z component</td>
<td>23</td>
<td>greatest z component</td>
</tr>
<tr>
<td>-13</td>
<td>least z component</td>
<td>-23</td>
<td>least z component</td>
</tr>
<tr>
<td>14</td>
<td>greatest distance from the origin</td>
<td>24</td>
<td>greatest distance from the origin</td>
</tr>
<tr>
<td>-14</td>
<td>least distance from the origin</td>
<td>-24</td>
<td>least distance from the origin</td>
</tr>
<tr>
<td>15</td>
<td>greatest distance from T</td>
<td>25</td>
<td>greatest distance from T</td>
</tr>
<tr>
<td>-15</td>
<td>least distance from T</td>
<td>-25</td>
<td>least distance from T</td>
</tr>
</tbody>
</table>

- Ignore the cantile (a part of) which represents point T.
- Accept point T as the solution.

---

**Fig 3.20**

- Plan
  - U1, x
  - U2, y
  - U3, z (upwards)

**Fig 3.21**

- C is the centre of sphere
- R is the radius of sphere
- Centre of projection
- Grid
Consider a straight line in 3-space whose equation is given as:

\[
\frac{x-x_a}{U1-xa} = \frac{y-y_a}{U2-ya} = \frac{z-z_a}{U3-za} \quad \cdots U1-xa\neq0, U2-ya\neq0, U3-za\neq0 \cdots 1
\]

where the values \(x_a, y_a, z_a, U1, U2\) and \(U3\) are as explained before.

This will give rise to the equations:

\[
\frac{x-x_a}{U1-xa} = \frac{z-z_a}{U3-za} \quad \cdots 1a
\]

and

\[
\frac{y-y_a}{U2-ya} = \frac{z-z_a}{U3-za} \quad \cdots 1b
\]

This line will intersect a sphere whose equation is given by

\[(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 = R^2 \quad \cdots 2\]

in points, whose coordinates may be obtained as the solutions of Eqns 1 and 2.

Substitution of \(x\) and \(y\) from Eqns 1a and 1b in Eqn 2, gives

\[
\left[\frac{U1-xa}{U3-za} \ast (z-za) + xa-x_c\right]^2 + \left[\frac{U2-ya}{U3-za} \ast (z-za) + ya-y_c\right]^2 + (z-zc)^2 = R^2 \quad \cdots 3
\]

This is a quadratic equation giving two values of \(z\), the corresponding values of \(x\) and \(y\) may be obtained by substituting \(z\) into the Eqns 1a and 1b.

The two values of \(z\) are the roots of Eqn 3. These roots may be real and different,
real and coincident, or imaginary.

If the roots are real and different the line will intersect the sphere in two points whose coordinates are given as follows

\[
x = \frac{U1 - xa}{U3 - za} \cdot (z - za) + xa
\]

\[
y = \frac{U2 - ya}{U3 - za} \cdot (z - za) + ya
\]

\[
z = -\frac{\beta + \sqrt{\beta^2 - 4 \cdot \alpha \cdot \gamma}}{2 \cdot \alpha}
\]

and

\[
x = \frac{U1 - xa}{U3 - za} \cdot (z - za) + xa
\]

\[
y = \frac{U1 - ya}{U3 - za} \cdot (z - za) + ya
\]

\[
z = -\frac{\beta - \sqrt{\beta^2 - 4 \cdot \alpha \cdot \gamma}}{2 \cdot \alpha}
\]

where

\[
\alpha = \left( \frac{U1 - xa}{U3 - za} \right)^2 + \left( \frac{U2 - ya}{U3 - za} \right)^2 + 1
\]

\[
\beta = -2 \cdot zc - 2 \cdot za \cdot \left( \frac{U1 - xa}{U3 - za} \right)^2 + \left( \frac{U2 - ya}{U3 - za} \right)^2 + 2 \cdot \frac{U1 - xa}{U3 - za} \cdot (xa - xc) + \frac{U2 - ya}{U3 - za} \cdot (ya - yc)
\]

and
\[ \gamma = za^2 \cdot \left[ \left( \frac{U1-xa}{U3-za} \right)^2 + \left( \frac{U2-ya}{U3-za} \right)^2 \right] - R^2 + (xa-xc)^2 + (ya-yc)^2 + zc^2 - \]

\[ 2 \cdot za \cdot \left[ \frac{U1-xa}{U3-za} \cdot (xa-xc) + \frac{U2-ya}{U3-za} \cdot (ya-yc) \right] \]

If the roots are coincident the line is tangent to the sphere and the required point is the point of tangency whose coordinates are given as

\[ x = \frac{U1-xa}{U3-za} \cdot (z-za) + xa \]

\[ y = \frac{U2-ya}{U3-za} \cdot (z-za) + ya \]

\[ z = \frac{-\beta}{2 \cdot \alpha} \]

where the values of \( \alpha \) and \( \beta \) are as given above.

If the roots are imaginary, the line will not intersect the sphere at all.

### 3.6.2 EXAMPLES FOR PROJECTION ON A SPHERE

The configurations generated in this section are the results of projecting a grid on a sphere. To begin with, consider the grid shown in Fig 3.20. The interconnection pattern of the configuration with respect to the indicated normat may be formulated as

\[ \text{GRID} = \text{PEX} | \text{RINID}(10,10,2,2) | \text{ROSAD}(1,1) | \{[0,0; 2,0],[0,0; 1,1]\} \]

The viewing specifications that give rise to the configurations which are discussed in this section, are provided by the USE-items
USE VM(3), VT(1), VZ(-5,-5,5,25,25,20), VH(10,10,200,10,10,190,10,11,190)

The USE-items have been chosen such that a plan view is obtained.

Suppose that, this grid is to be projected on a sphere. For this purpose the tractation retronorm is used and the descriptor for tractation may be defined as

\[ [P, x_a, y_a, z_a, S, x_c, y_c, z_c, R, t] \]

where \( P \) is the projection specifier, \( x_a, y_a \) and \( z_a \) are the coordinates of the centre of projection with respect to the global coordinate system. \( S \) is the surface specifier, where \( S=1 \) implies a sphere, with its centre given by the coordinates \( x_c, y_c \) and \( z_c \) with respect to the global coordinate system and \( R \) is the radius of the sphere. The projection of a point on a surface such as sphere, ellipsoid, cylinder or a paraboloid may be represented as a quadratic equation. Therefore, there may be two solutions for the projection, Fig 3.18. The choice of either of the two solutions is determined by the parameter \( t \) which is referred to as the "selector". The value of \( t=13 \) has been used throughout this section and stands for the solution with the larger \( z \). The significance of the parameter \( t \) in the above descriptor is as described in section 3.5.

Suppose that, the descriptor is specified as

\[ [1,10,10,-10,1,10,10,-5,20,13] \]

where the first parameter \( P=1 \) implies central projection. To explain how the projection takes place the plan view and elevation of the grid together with the sphere on which the projection is to be obtained are shown in Fig 3.21. The dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut the sphere. This gives rise to the solution determined by \( t=13 \). The solution is shown in Fig 3.21 by a thick line. A graphical representation of formex GRID projected on the sphere is as shown in Fig 3.22a. As it has been mentioned above,
Fig 3.22
for the configurations which are discussed in this section the USE-items have been
chosen such that a plan view is always obtained. In plan view, the projection of the
grid, the grid itself and the centre of projection are on top of each other as shown in
Fig 3.23a. In order to see them clearly, the grid together with the centre of
projection are moved in the x-y plane, Figs 3.23b-3.23c. This convention of shifting
the grid and the centre of projection such that they can be seen clearly is used
henceforth.

As a further example of central projection consider the grid represented by the formex
GRID and let this be projected onto a sphere using a different radius and centre of
projection. The other parameters in the descriptor remain as before.

The descriptor in this case may be given by

\[ [1,10,10,25,1,10,10,-5,8,13] \]

Fig 3.24 shows the plan view and elevation of the grid together with the sphere. A
graphical representation of formex GRID projected on the sphere is shown in Fig
3.25.

In the descriptor given above the parameters have been chosen such as to bring out
an interesting feature of the tractation retronorm. In the descriptor the radius of the
sphere has been chosen as \( R=8 \). The length of the grid is equal to 20. Since the
sphere is smaller as compared to the grid, only a part of the grid is projected onto the
sphere. The rest of the grid is not projected and remains as before.

The above examples illustrate the concept of central projection. However, as
mentioned before in tractation retronorm the user is able to choose between different
types of projection, such as axial, parallel or radial. In certain applications the
required shape may be derived from axial or parallel projection.
Fig 3.23

(a) Centre of projection

(b) Projection

(c) Grid

Fig 3.24

Plan

A is the centre of projection

Elevation

C is the centre of sphere

Fig 3.25
In order to project the configuration represented by GRID on a sphere, using parallel projection, the descriptor has to be defined. The descriptor for parallel projection may take the form

\[ [2, xa, ya, za, xb, yb, zb, 1, xc, yc, zc, R, 13] \]

where the value of the first parameter, that is the projection specifier for parallel projection is equal to 2. The coordinates \( xa, ya, za, xb, yb \) and \( zb \) represent two points, in the global coordinate system, defining the direction of the projection AD. The centre of the sphere relative to the global coordinate system is given by the coordinates \( xc, yc \) and \( zc \). \( R \) is the radius of the sphere.

If the descriptor is specified as

\[ [2, 10, 10, -10, 10, 10, -20, 1, 10, 10, -5, 20, 13] \]

then a graphical representation of formex GRID projected on the sphere using parallel projection is as shown in Fig 3.22d. To explain how the projection takes place, the plan view and elevation of the grid, together with the sphere are shown in Fig 3.26. The dashed lines which are drawn parallel to the direction of projection pass through the edges of the grid and cut the sphere. This gives rise to the solution where \( t = 13 \). Therefore, in plan, a configuration with the same dimensions as the flat grid is obtained.

So far, central and parallel projections have been described. The following examples illustrate the concept of axial projection. To begin with, the corners of the grid shown in Fig 3.20 have been marked as KLMN for identification. Suppose that the grid is to be projected on a sphere using axial projection. The descriptor for such a projection may take the form

\[ [3, xa, ya, za, xb, yb, zb, 1, xc, yc, zc, R, 13] \]
Fig 3.26

C is the centre of sphere

Fig 3.27

C is the centre of sphere

Fig 3.28

C is the centre of sphere
where the value of the first parameter, that is the projection specifier for axial projection is equal to 3. Also, $xa, ya, za, xb, yb$ and $zb$ are the coordinates of two points with respect to the global coordinate system defining the axis of projection $AB$.

If the descriptor is specified as

$$[3,15,5,-10,5,15,-10,1,10,10,-5,20,13]$$

then a graphical representation of formex GRID projected on the sphere using axial projection is as shown in Fig 3.22b. To explain how the projection takes place the plan view and elevation of the grid, together with the sphere on which the projection is to be obtained are shown in Fig 3.27. The axis of projection $AB$ is drawn in the same figure. Once the axis $AB$ is defined the projection is obtained in a manner similar to the one described in section 3.2.3. It should be noticed that in plan, the configuration has the same length along LN which is in the direction parallel to the axis of projection $AB$. This results in the configuration being elongated in the direction perpendicular to the axis of projection, along the corners $KM$.

In the examples discussed in this section, which are obtained from axial projection the dimension of the grid which is parallel to the axis $AB$ always remains the same in the plan view. However, the length of the grid perpendicular to the axis $AB$ varies depending on the position of the axis of projection $AB$. The different effects which are obtained by changing the position of the axis $AB$ has been described in detail below. The position of the axis of projection in each case has been given in terms of its z-coordinate. That is, the $x$ and $y$ coordinates of the points $A$ and $B$ of the axis remaining the same and only the z-coordinates changing. Therefore points $A$, $B$ are given as $(15,5,z)$ and $(5,15,z)$.

To begin with consider the example of axial projection which has been given above, where the axis of projection $AB$ has $z=-10$. Now, suppose that the axis $AB$ is moved away from the grid by changing the $z$ coordinate, Fig 3.28. The configuration being
elongated in the direction perpendicular to the axis of projection. A graphical representation for \( z = -10 \) and \( z = -40 \) is shown in Figs 3.29a and 3.29b, respectively. It can be seen that as the axis moves away from the grid the configuration tends to be the same as the original grid, Fig 3.29b.

Suppose now, that the axis of projection AB is moved above the grid so that the z-coordinate changes from \( z = 5 \) to \( z = 10 \), \( z = 15 \), \( z = 20 \), \( z = 25 \), Fig 3.30. The length of the configuration along the corners KM decreases, Fig 3.31.

It must be noticed that for \( z = 5 \) and \( z = 10 \) the axis AB is inside the sphere and the effect is shown in Fig 3.32. It is observed that the length of the configuration along corners KM goes on decreasing as the axis AB moves towards the periphery of the sphere, until the corners KM come closer to each other as shown in Fig 3.32. When the axis AB is a tangent to the sphere, \( z = 15 \), the corners KM have the minimum distance between them, Fig 3.32.

As the axis goes away from the periphery, that is for \( z = 20 \), \( z = 25 \) the effect is as shown in Fig 3.32. The length of the configuration perpendicular to axis AB starts increasing.

Now suppose, that the axis of projection AB is rotated by 90 degrees as shown in plan in Fig 3.33. The descriptor may be specified as

\[
[3,5,5,-10,15,15,-10,1,10,10,-5,20,13]
\]

then a graphical representation of formex GRID projected on the sphere is as shown in Fig 3.22c. In this case the length of diagonal NL of the configuration varies.

As in the previous case, once the axis AB is defined the projection is obtained in a manner similar to the one described in section 3.2.3. Therefore, in plan, the configuration has the same length along the direction parallel to axis of projection.
Fig 3.29

(a)

Fig 3.30

(b)
Fig 3.31

Fig 3.32
Fig 3.33

Plan

axis of projection

C is the centre of sphere

Elevation

Fig 3.34

Plan

C is the centre of sphere

Elevation

Fig 3.36

Plan

AB is the axis of projection

Elevation
AB, along the corners KM. This results in the configuration being elongated in the
direction perpendicular to the axis AB along corners NL.

As before, the length of the grid perpendicular to the axis AB varies depending on
the position of the axis of projection. An approach similar to the previous example
has been followed. The position of the axis of projection AB in each case as
discussed before, has been given in terms of its z-coordinate. The axis AB is
moved from $z=-10$ to $z=-40$ as shown in Fig 3.34. As the axis AB moves away
from the grid, the configuration tends to be the same as the original grid, Fig 3.35b.
If the axis AB is brought nearer the grid an effect similar to the one in Fig 3.35a is
obtained.

Suppose that, the axis of projection is moved above the grid so that the z-coordinate
changes from $z=5$ to $z=10$, $z=15$, $z=20$, $z=25$, Fig 3.36. The length of the
configuration along corners NL decreases following the effects which have been
described before. That is, the length of the configuration along corners NL goes on
decreasing as the axis AB moves towards to the periphery of the sphere. The corners
NL come closer to each other as shown in Fig 3.37. For $z=15$ the axis AB is a
tangent to the sphere and the corners NL have the minimum distance between them,
Figs 3.37-3.38.

As the axis goes away from the periphery, that is for $z=20$, $z=25$ the effect is shown
in Fig 3.37. The length of the configuration along corners NL starts increasing.

Very often the centre of projection may coincide with the centre of a surface and
radial projection is used. The radial projection is a special case of central projection.
Radial projection may be used only for surface such as spheres, ellipsoids. The
descriptor in this case may take the form

$$[P,S,xc,yc,zc,R,t]$$

232
axis of projection
z = -10

axis of projection
z = -40

(a)

(b)

Fig 3.35
where the value of the projection specifier \( P \) for radial projection is equal to 4. Also, \( x_c, y_c, z_c \) are the coordinates of the centre of the sphere which coincides with the centre of projection, relative to the global coordinate system. \( S \) is the surface specifier, \( R \) is the radius of the sphere and \( t \) is the selector.

If the descriptor is specified as

\[ [4,1,10,10,-5,20,13] \]

then a graphical representation of formex GRID projected on the sphere using radial projection is as shown in Fig 3.39. The plan view and elevation of the grid together with the sphere on which the projection is to be obtained are shown in Fig 3.40.

As a projection of the grid on the sphere involves a quadratic equation, there may be two solutions as shown in Fig 3.41. So far, in the examples given above the value of the selector has been specified as \( t=13 \), which stands for the solution with larger \( z \). Now, suppose that the grid is projected on the sphere using central projection and the descriptor is given as

\[ [1,10,10,-10,1,10,10,-5,20,-13] \]

where \( t=-13 \) stands for the solution with smaller \( z \). Fig 3.41 shows the plan view and elevation of the grid together with the sphere. The dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut the sphere. This gives rise to the solutions \( t=13 \) and \( t=-13 \). Both of the solutions are drawn in Fig 3.41 with thick lines. A graphical representation of the formex GRID projected on the sphere is as shown in Fig 3.42. It can be seen that each one of the values of the parameter \( t \) gives rise to a different configuration. The choice of the configuration is entirely dependent on the application and the requirement of the user.
Fig 3.39

Fig 3.40

Fig 3.41
3.7 ELLIPSOID

An ellipsoid is a surface whose plane sections are all either ellipses or circles. An ellipsoid is symmetrical with respect to the three planes determined by three lines which are the principal axes of the ellipsoid. The largest axis is the major axis and the smaller axis is the minor axis. The third axis perpendicular to the other two is the mean axis. The intersection of these lines is called the centre C. Any chord through the centre is called a diameter, Fig 3.43. In Cartesian coordinates, the equation of an ellipsoid may be obtained as follows:

A. If the centre of the ellipsoid is the origin, Fig 3.43a, the equation is given as

\[
\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1
\]

B. If the centre of the ellipsoid is the point C(xc, yc, zc), Fig 3.43b, the equation is given as

\[
\frac{(x-xc)^2}{a^2} + \frac{(y-yc)^2}{b^2} + \frac{(z-zc)^2}{c^2} = 1
\]

The centre of an ellipsoid is the point of symmetry of the ellipsoid. This point is the intersection of the three principal planes of the ellipsoid. A line segment from the centre to the ellipsoid along one of the axis is called semiaxis. When two of the three semiaxes a, b and c are equal, the surface is an ellipsoid of revolution.

An ellipsoid of revolution, or spheroid, is an ellipsoid generated by revolving an ellipse about one of its axes which is called the axis of revolution. This is an ellipsoid whose sections by planes perpendicular to the axis of revolution are all circles.

3.7.1 INTERSECTION OF AN ELLIPSOID AND A STRAIGHT LINE

Consider a straight line in 3-space whose equation may be given as
Fig 3.43

Fig 3.44

Fig 3.45

A is the centre of projection

C is the centre of ellipsoid

a and c are the semioaxes of ellipsoid
where the values \( x_a, y_a, z_a, U_1, U_2 \) and \( U_3 \) are as explained above.

This will give rise to the equations

\[
\frac{x-x_a}{U_1-x_a} = \frac{z-z_a}{U_3-z_a} \quad ...1a
\]

and

\[
\frac{y-y_a}{U_2-y_a} = \frac{z-z_a}{U_3-z_a} \quad ...1b
\]

This line will intersect an ellipsoid whose equation is given by

\[
\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} + \frac{(z-z_c)^2}{c^2} = 1 \quad ...2
\]

in points, whose coordinates may be obtained as the solutions of Eqns 1 and 2.

Substitution of \( x \) and \( y \) from Eqns 1a and 1b in Eqn 2, gives

\[
\frac{(U_1-x_a)(z-z_a) + x_a-x_c}{a^2} + \frac{(U_2-y_a)(z-z_a) + y_a-y_c}{b^2} + \frac{(z-z_c)^2}{c^2} = 1 \quad ...3
\]

This is a quadratic equation giving two values of \( z \), the corresponding values of \( x \) and \( y \) may be obtained by substituting \( z \) into the Eqns 1a and 1b.

The two values of \( z \) are the roots of Eqn 3. These roots may be real and different, real and coincident, or both are imaginary.

If the roots are real and different the line will intersect the ellipsoid in two points...
whose coordinates are given as follows:

\[
x = \frac{U_1 - x_a}{U_3 - z_a} \cdot (z - z_a) + x_a
\]

\[
y = \frac{U_2 - y_a}{U_3 - z_a} \cdot (y - y_a) + y_a
\]

\[
z = \frac{-\beta + \sqrt{\beta^2 - 4 \cdot \alpha \cdot \gamma}}{2 \cdot \alpha}
\]

and

\[
x = \frac{U_1 - x_a}{U_3 - z_a} \cdot (z - z_a) + x_a
\]

\[
y = \frac{U_2 - y_a}{U_3 - z_a} \cdot (z - z_a) + y_a
\]

\[
z = \frac{-\beta - \sqrt{\beta^2 - 4 \cdot \alpha \cdot \gamma}}{2 \cdot \alpha}
\]

where

\[
\alpha = \left( \frac{U_1 - x_a}{U_3 - z_a} \right)^2 \cdot b^2 \cdot c^2 + \left( \frac{U_2 - y_a}{U_3 - z_a} \right)^2 \cdot a^2 \cdot c^2 + a^2 \cdot b^2
\]

\[
\beta = 2 \cdot b^2 \cdot c^2 \cdot (x_a - x_c) \cdot \frac{U_1 - x_a}{U_3 - z_a} + 2 \cdot a^2 \cdot c^2 \cdot (y_a - y_c) \cdot \frac{U_2 - y_a}{U_3 - z_a}
\]

\[
2 \cdot z_c \cdot a^2 \cdot b^2 - 2 \cdot z_a \cdot \left[ b^2 \cdot c^2 \cdot \left( \frac{U_1 - x_a}{U_3 - z_a} \right)^2 + a^2 \cdot c^2 \cdot \left( \frac{U_2 - y_a}{U_3 - z_a} \right)^2 \right]
\]

and

241
\[ y = za^2 \left[ \left( \frac{U1-xa}{U3-za} \right)^2 + b^2 \right] + \left( \frac{U2-ya}{U3-za} \right)^2 + a^2 \left( \frac{xa-xc}{za} \right)^2 + c^2 \left( \frac{ya-yc}{za} \right)^2 \]

If the roots are real and coincident the line is a tangent to the ellipsoid and the required point is the point of tangency whose coordinates are given as

\[ x = \frac{U1-xa}{U3-za} \left( z - za \right) + xa \]

\[ y = \frac{U2-ya}{U3-za} \left( z - za \right) + ya \]

\[ z = \frac{-\beta}{2\alpha} \]

where the values of \( \alpha \) and \( \beta \) are as given above.

If the roots are imaginary, the line will not appear to intersect the ellipsoid at all.

### 3.7.2 EXAMPLES FOR PROJECTION ON AN ELLIPSOID

The configurations generated in this section are the results of projecting a grid on an ellipsoid. To begin with, consider the grid shown in Fig 3.44. The interconnection pattern of the configuration with respect to the indicated normat may be formulated as

\[ \text{GRID} = \text{PEX} | \text{RINID}(10,10,2,2) | \text{ROSAD}(1,1) | \{ [0,0; 2,0], [0,0; 1,1] \} \]
The viewing specifications that give rise to the configurations which are discussed in this section are provided by the USE-items

USE VM(3), VT(1), VZ(-5,-5,5,25,25,15), VH(10,10,200,10,10,190,10,11,190)

The USE-items have been chosen such that a plan view is obtained.

Suppose that this grid is to be projected on an ellipsoid. For this purpose the tractation retronorm is used and the descriptor for tractation may be defined as

$$[P, x_a, y_a, z_a, S, x_c, y_c, z_c, a, b, c, t]$$

where $P$ is the projection specifier, $x_a$, $y_a$ and $z_a$ are the coordinates of the centre of projection with respect to the global coordinate system. $S$ is the surface specifier, where $S=2$ implies an ellipsoid, with its centre given by the coordinates $x_c$, $y_c$ and $z_c$ with respect to the global coordinate system and $a$, $b$ and $c$ are the semiaxes of the ellipsoid. The projection of a point on a surface such as a sphere, ellipsoid, cylinder or a paraboloid may be represented as a quadratic equation. Therefore there may be two solutions for the projection, Fig 3.18. The choice of either of the two solutions is determined by the parameter $t$ which is referred to as the "selector". The value of $t=13$ has been used throughout this section and stands for the solution with the larger $z$. The significance of the parameter $t$ in the above descriptor is as described in section 3.5.

Suppose that, the descriptor is specified as

$$[1,10,10,-10,2,10,10,-5,30,13,15,13]$$

where the first parameter $P=1$ implies central projection. To explain how the projection takes place the plan view and elevation of the grid together with the ellipsoid on which the projection is to be obtained are shown in Fig 3.45. The
dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut the ellipsoid. This gives rise to the solution determined by \( t = 13 \). The solution is shown in Fig 3.45 with a thick line. A graphical representation of formex GRID projected on the ellipsoid is as shown in Fig 3.46a. A view of the grid together with the global coordinate system and the centre of projection is drawn in the same figure. For the configurations which are discussed in this section the USE-items have been chosen such that a plan view is always obtained. In plan view the projection of the grid, the projection itself and the centre of projection are on top of each other as shown in Fig 3.23a. In order to see them clearly the grid together with the centre of projection are moved in the x-y plane, Figs 3.23b-3.23c. This convention of shifting the grid and the centre of projection such that they can be seen clearly is used henceforth.

As a further example of central projection consider the grid represented by the formex GRID and let this be projected onto an ellipsoid using a different radius and centre of projection. The other parameters in the descriptor remain as before.

The descriptor in this case may be given by

\[ [1,10,10,20,2,10,10,-5,9,13,10,13] \]

Fig 3.47 shows the plan view and elevation of the grid together with the ellipsoid. A graphical representation of formex GRID projected on the ellipsoid is shown in Fig 3.48.

In the descriptor given above the parameters have been chosen such as to bring out an interesting feature of the tractation retronorm. In the descriptor the semiaxes of the ellipsoid has been chosen as \( a = 9 \), \( b = 13 \), \( c = 10 \). The length of the grid is equal to 20. Since the ellipsoid is smaller as compared to the grid, only a part of the grid is projected onto the ellipsoid. The rest of the grid is not projected and remains as before.
Fig 3.46

(a) Centre of projection

(b) Axis of projection
Fig 3.47

C is the centre of ellipsoid
A is the centre of projection
Elevation

Fig 3.48

Fig 3.49
The above examples illustrate the concept of central projection. As mentioned before in tractation retronorm the user is able to choose between different types of projection, such as axial, parallel or radial. In certain applications the required shape may be derived from axial or parallel projection.

The following examples illustrate the concept of axial projection. To begin with, the corners of the grid shown in Fig 3.44 have been marked as KLMN for identification. Suppose that the grid is to be projected on an ellipsoid using axial projection. The descriptor for such a projection may take the form

\[ [3, x_a, y_a, z_a, x_b, y_b, z_b, 2, x_c, y_c, z_c, a, b, c, 13] \]

where the value of the first parameter, that is the projection specifier for axial projection is equal to 3. Also, \( x_a, y_a, z_a, \) \( x_b, y_b \) and \( z_b \) are the coordinates of two points with respect to the global coordinate system defining the axis of projection AB.

If the descriptor is specified as

\[ [3, 5, 5, -10, 15, 15, -10, 2, 10, 10, -5, 30, 13, 15, 13] \]

then a graphical representation of formex GRID projected on the ellipsoid using axial projection is as shown in Fig 3.46b. To explain how the projection takes place the plan view and elevation of the grid together with the ellipsoid on which the projection is to be obtained are shown in Fig 3.49. The axis of projection AB is drawn in the same figure. Once the axis AB is defined the projection is obtained in a manner similar to the one described in section 3.2.3. It should be noticed that in plan, the configuration has the same length along KM which is in the direction parallel to the axis of projection AB. This results in the configuration being elongated in the direction perpendicular to the axis of projection, that is corners LN.

In the examples discussed in this section, which are obtained from axial projection,
the dimension of the grid which is parallel to the axis AB always remains the same in the plan view. However, the length of the grid perpendicular to the axis AB varies depending on the position of the axis of projection AB. The different effects which are obtained by changing the position of the axis AB have been described in detail below. The position of the axis of projection in each case has been given in terms of its z-coordinate. That is, the x and y coordinates of the points A and B of the axis remain the same and only the z-coordinates change. Therefore points A, B are given as (5,5,z) and (15,15,z).

To begin with consider the example of axial projection which has been given above. In this case the axis of projection AB has z=-10 and the effect is shown in Fig 3.50a. The configuration is elongated in the direction perpendicular to the axis of projection. Now the axis AB is moved away from the grid by changing the z-coordinate to z=-40, Fig 3.51. It can be seen that as the axis moves away from the grid the configuration tends to be the same as the original grid, Fig 3.50b.

Suppose now, that the axis of projection AB is moved above the grid so that the z-coordinate changes from z=5 to z=10, z=15, z=20, Fig 3.52. The length of the configuration along the corners LN decreases, Fig 3.53.

It must be noticed that for z=5 the axis AB is inside the ellipsoid and the effect is shown in Fig 3.54. It is observed that the length of the configuration along corners LN goes on decreasing as the axis AB moves towards the periphery of the ellipsoid, until the corners LN come closer to each other as shown in Fig 3.54. When the axis AB is a tangent to the ellipsoid, z=10, the corners LN have the minimum distance between them, Fig 3.54.

As the axis goes away from the periphery, that is for z=15, z=20 the effect is as shown in Fig 3.54. The length of the configuration perpendicular to axis AB starts to increase.
Fig 3.50

(a) Axis of projection $z=-10$

(b) Axis of projection $z=-40$
Fig 3.51

Fig 3.52

Fig 3.53
Very often the centre of projection may coincide with the centre of a surface and radial projection is used. The radial projection is a special case of central projection. Radial projection may be used only for surfaces such as spheres, ellipsoids. The descriptor in this case may take the form

\[ [P, S, x_c, y_c, z_c, a, b, c, t] \]

where the value of the projection specifier \( P \) for radial projection is equal 4. Also, \( x_c, y_c, z_c \) are the coordinates of the centre of the ellipsoid which coincide with the centre of projection, relative to the global coordinate system. \( S \) is the surface specifier, \( a, b \) and \( c \) are the semiaxes of the ellipsoid and \( t \) is the selector.

If the descriptor is specified as

\[ [4, 2, 10, 10, -5, 30, 13, 15, 13] \]

then a graphical representation of formex GRID projected on the ellipsoid using radial projection is as shown in Fig 3.55. The plan view and elevation of the grid together with the ellipsoid on which the projection is to be obtained are shown in Fig 3.56.

As a projection of the grid on the ellipsoid involves a quadratic equation, there may be two solutions as shown in Fig 3.57. So far, in the examples given above the value of the selector has been specified as \( t=13 \), which stands for the solution with the larger \( z \). Now, suppose that the grid is projected on the ellipsoid using central projection and the descriptor is given as

\[ [1, 10, 10, -10, 2, 10, 10, -5, 30, 13, 15, -13] \]

where \( t=-13 \) stands for the solution with the smaller \( z \). Fig 3.57 shows the plan view and elevation of the grid together with the ellipsoid. The dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut.
Fig 3.55

Fig 3.56

A is the centre of projection

Fig 3.57

C is the centre of ellipsoid
through the ellipsoid. This gives rise to the solutions $t=13$ and $t=-13$. Both of the solutions are drawn in Fig 3.57 with thick lines. A graphical representation of the formex GRID projected on the ellipsoid is as shown in Fig 3.58. It can be seen that each one of the values of the parameter $t$ gives rise to a different configuration. The choice of the configuration is entirely dependent on the application and the requirement of the user.

3.8 PARABOLOID

The term paraboloid applied to the elliptic and hyperbolic paraboloids. The elliptic paraboloid is a surface such that its sections parallel to one of the coordinate planes are ellipses, and parallel to the other coordinate planes are parabolas. When the surface is in the position illustrated in Fig 3.59, with its axis along the $z$-axis, in Cartesian coordinates, the equation of an elliptic paraboloid may be obtained as follows:

A. If the centre is the origin, Fig 3.59a, then the equation is given as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z}{c} = 0$$

B. If the centre is the point $C(x_c, y_c, z_c)$, Fig 3.59b, then the equation is given as

$$\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} + \frac{z-z_c}{c} = 0$$

The cross sections perpendicular to the $z$ axis above the $xy$-plane are ellipses. The cross sections in the planes that contain the $z$-axis are parabolas.

The parabola of revolution or circular paraboloid is a special case of an elliptic paraboloid in which the ellipses are circles and is obtained by taking $b=a$ in the equations for the elliptic paraboloid. In Cartesian coordinates the equation of a circular paraboloid may be obtained as follows:
Fig 3.58
A. If the centre is the origin, then the equation is given as
\[ \frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z}{c} = 0 \]

B. If the centre is the point \( C(x_c, y_c, z_c) \), then the equation is given as
\[ \frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{a^2} + \frac{z-z_c}{c} = 0 \]

The cross sections of the surface by planes perpendicular to the z-axis are circles. The cross sections by planes containing the z axis are parabolas.

The hyperbolic paraboloid is a surface such that its sections parallel to one of the coordinate planes are hyperbolas and parallel to the other coordinate planes are parabolas. When the surface is in the position illustrated in Fig 3.60, the equation of a hyperbolic paraboloid may be given as
\[ \frac{y^2}{b^2} - \frac{x^2}{a^2} = \frac{z}{c} \]

The cross section perpendicular to the z axis above and below the xy-plane are hyperbolas. The cross sections in planes perpendicular to the other axes are parabolas.

3.8.1 INTERSECTION OF AN ELLIPTIC PARABOLOID AND A STRAIGHT LINE

Consider a straight line in 3-space whose equation may be given as
\[ \frac{x-x_a}{U_1-x_a} = \frac{y-y_a}{U_2-y_a} = \frac{z-z_a}{U_3-z_a} \]

where the values \( x_a, y_a, z_a, U_1, U_2 \) and \( U_3 \) are as explained before.

This will give rise to the equations
\[
\frac{x-x_a}{U1-x_a} = \frac{z-z_a}{U3-z_a} \quad \ldots \ldots 1a
\]

and
\[
\frac{y-y_a}{U2-y_a} = \frac{z-z_a}{U3-z_a} \quad \ldots \ldots 1b
\]

This line will intersect an elliptic paraboloid whose equation is given by
\[
\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} + \frac{z-z_c}{c} = 0 \quad \ldots \ldots 2
\]
in points, whose coordinates may be obtained as the solutions of Eqns 1 and 2. Substitution of \( x \) and \( y \) from Eqns 1a and 1b in Eqn 2, gives
\[
\left[ \frac{U1-x_a}{U3-z_a} \cdot (z-z_a) + x_a - x_c \right]^2 + \left[ \frac{U2-y_a}{U3-z_a} \cdot (z-z_a) + y_a - y_c \right]^2 + \frac{z-z_c}{c} = 0 \quad \ldots \ldots 3
\]
This is a quadratic equation giving two values of \( z \), the corresponding values of \( x \) and \( y \) may be obtained by substituting \( z \) into the Eqns 1a and 1b.

The two values of \( z \) are the roots of Eqn 3. These roots may be real and different, real and coincident, or both may be imaginary. If the roots are real and different the line will intersect the elliptic paraboloid in two points whose coordinates are given as follows

\[
x = \frac{U1-x_a}{U3-z_a} \cdot (z-z_a) + x_a
\]

\[
y = \frac{U2-y_a}{U3-z_a} \cdot (z-z_a) + y_a
\]

\[
z = -\beta + \sqrt{\beta^2 - 4 \cdot \alpha \cdot \gamma} \]
\[
2 \cdot \alpha
\]

258
and

\[ x = \frac{U_1-\alpha_a}{U_3-\alpha} * (z-\alpha_z) + \alpha_a \]

\[ y = \frac{U_2-\alpha_y}{U_3-\alpha} * (z-\alpha_z) + \alpha_y \]

\[ z = -\beta - \sqrt{\beta^2 - 4 \alpha \gamma} \]

\[ 2 \alpha \]

where

\[ \alpha = \left( \frac{U_1-\alpha_a}{U_3-\alpha} \right)^2 * b^2 * c + \left( \frac{U_2-\alpha_y}{U_3-\alpha} \right)^2 * a^2 * c \]

\[ \beta = 2 * c \left[ \frac{U_1-\alpha_a}{U_3-\alpha} * (\alpha_a-\alpha_x) * b^2 + \frac{U_2-\alpha_y}{U_3-\alpha} * (\alpha_y-\alpha_y) * a^2 \right] + a^2 * b^2 - 2 \alpha * \alpha_z \]

and

\[ \gamma = \alpha_z^2 * \alpha + (\alpha_a-\alpha_x)^2 * b^2 * c + (\alpha_y-\alpha_y)^2 * a^2 * c - \alpha_z^2 * b^2 * \alpha_z - \]

\[ 2 \alpha_z * c \left[ \frac{U_1-\alpha_a}{U_3-\alpha} * (\alpha_a-\alpha_x) * b^2 + \frac{U_2-\alpha_y}{U_3-\alpha} * (\alpha_y-\alpha_y) * a^2 \right] \]

If the roots are real and coincident the line is a tangent to the elliptic paraboloid and the required point is the point of tangency, whose coordinates are given as

\[ x = \frac{U_1-\alpha_a}{U_3-\alpha} * (z-\alpha_z) + \alpha_a \]

\[ y = \frac{U_2-\alpha_y}{U_3-\alpha} * (z-\alpha_z) + \alpha_y \]
\[ z = \frac{-\beta}{2 \alpha} \]

where the values of \( \alpha \) and \( \beta \) are as given above.

If the roots are imaginary, the line will not intersect the elliptic paraboloid at all.

### 3.8.2 INTERSECTION OF A HYPERBOLIC PARABOLOID AND A STRAIGHT LINE

Consider a straight line in 3-space whose equation may be given as

\[
\frac{x-x_a}{U1-x_a} = \frac{y-y_a}{U2-y_a} = \frac{z-z_a}{U3-z_a} \quad \text{...4a}
\]

where the values \( x_a, y_a, z_a, U1, U2 \) and \( U3 \) are as explained before.

This will give rise to the equations

\[
\frac{x-x_a}{U1-x_a} = \frac{z-z_a}{U3-z_a} \quad \text{.........4a}
\]

and

\[
\frac{y-y_a}{U2-y_a} = \frac{z-z_a}{U3-z_a} \quad \text{.........4b}
\]

This line will intersect a hyperbolic paraboloid whose equation is given by

\[
\frac{y^2}{b^2} - \frac{x^2}{a^2} - \frac{z}{c} = 0 \quad \text{.........5}
\]

in points, whose coordinates may be obtained as the solutions of Eqns 4 and 5. Substitution of \( x \) and \( y \) from Eqns 4a and 4b in Eqn 5, gives
This is a quadratic equation giving two values of $z$, the corresponding values of $x$ and $y$ may be obtained by substituting $z$ into the Eqns 4a and 4b.

The two values of $z$ are the roots of Eqn 6. These roots may be real and different, real and coincident, or both may be imaginary.

If the roots are real and different the line will intersect the hyperbolic paraboloid in two points whose coordinate are given as follows:

$$x = \frac{U_1 - xa}{U_3 - za} \cdot (z - za) + xa$$
$$y = \frac{U_2 - ya}{U_3 - za} \cdot (z - za) + ya$$
$$z = \frac{-\beta + \sqrt{\beta^2 - 4*a*y}}{2*a}$$
where

\[ \alpha = \left( \frac{U_2 - ya}{U_3 - za} \right)^2 \cdot a^2 \cdot c - \left( \frac{U_1 - xa}{U_3 - za} \right)^2 \cdot b^2 \cdot c \]

\[ \beta = 2 \cdot b^2 \cdot c \cdot za \cdot \left( \frac{U_1 - xa}{U_3 - za} \right)^2 - 2 \cdot a^2 \cdot c \cdot za \cdot \left( \frac{U_2 - ya}{U_3 - za} \right) + 2 \cdot \frac{U_2 - ya}{U_3 - za} \cdot ya \cdot a^2 \cdot c - 2 \cdot \frac{U_1 - xa}{U_3 - za} \cdot xa \cdot b^2 \cdot c - a^2 \cdot b^2 \]

and

\[ \gamma = a^2 \cdot c \cdot za^2 \cdot \left( \frac{U_2 - ya}{U_3 - za} \right)^2 - b^2 \cdot c \cdot za \cdot \left( \frac{U_1 - xa}{U_3 - za} \right)^2 - 2 \cdot za \cdot ya \cdot a^2 \cdot c \cdot \frac{U_2 - ya}{U_3 - za} + 2 \cdot \frac{U_1 - xa}{U_3 - za} \cdot za \cdot xa \cdot b^2 \cdot c + a^2 \cdot c \cdot ya^2 - b^2 \cdot c \cdot xa^2 \]

If the roots are real and coincident the line is a tangent to the hyperbolic paraboloid and the required point is the point of tangency, whose coordinates are given as

\[ x = \frac{U_1 - xa}{U_3 - za} \cdot (z - za) + xa \]

\[ y = \frac{U_2 - ya}{U_3 - za} \cdot (z - za) + ya \]

\[ z = -\frac{\beta}{2 \cdot \alpha} \]

where the values of \( \alpha \) and \( \beta \) are as given above.

If the roots are imaginary, the line will not intersect the hyperbolic paraboloid at all.
3.8.3 EXAMPLES FOR PROJECTION ON A PARABOLOID

The configurations generated in this section are the results of projecting a grid on a paraboloid. To begin with, consider the grid shown in Fig 3.61. The interconnection pattern of the configuration with respect to the indicated normal may be formulated as

\[
\text{GRID} = \text{PEX} | \text{RINID}(10, 10, 2, 2) | \text{ROSAD}(1, 1) | \{[0, 0; 2, 0], [0, 0; 1, 1]\}
\]

The viewing specifications that give rise to the configurations which are discussed in this section are provided by the USE-items

\[
\text{USE VM(3), VT(1), VZ(0,0,-5,15,15,50), VH(10,10,200,10,10,190,10,11,190)}
\]

The USE-items have been chosen such that a plan view is obtained.

Suppose that, this grid is to be projected on an elliptic paraboloid. For this purpose the tractation retronorm is used and the descriptor for tractation may be defined as

\[
[P, x_a, y_a, z_a, S, x_c, y_c, z_c, a, b, c, t]
\]

where P is the projection specifier, x_a, y_a and z_a are the coordinates of the centre of projection with respect to the global coordinate system. S is the surface specifier, where S = 3 implies an elliptic paraboloid, with its centre given by the coordinates x_c, y_c and z_c with respect to the global coordinate system and a, b and c are the semiaxes of the elliptic paraboloid. The projection of a point on a surface such as sphere, ellipsoid, cylinder or a paraboloid may be represented as a quadratic equation. Therefore there may be two solutions for the projection, Fig 3.18. The choice of either of the two solutions is determined by the parameter t which is referred to as the "selector". The value of t = 13 has been used throughout this section and stands for the solution with the larger z. The significance of the parameter t in the above
descriptor is as described in section 3.5.

Suppose that, the descriptor is specified as

\[ [1,10,10,-10,3,10,10,20,12,15,30,13] \]

where the first parameter \( P = 1 \) implies central projection. To explain how the projection takes place the plan view and elevation of the grid, together with the paraboloid on which the projection is to be obtained are shown in Fig 3.62. The dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut the paraboloid. This gives rise to the solution determined by \( t = 13 \). The solution is shown in Fig 3.62 with a thick line. A graphical representation of formex GRID projected on the elliptic paraboloid is as shown in Fig 3.63a. A view of the grid together with the global coordinate system and the centre of projection is drawn in the same figure. As mentioned before the configurations which are discussed in this section, the USE-items, have been chosen such that a plan view is always obtained. In plan view the projection the grid and the centre of projection are on top of each other as shown in Fig 3.23a. In order to see them clearly the grid together with the centre of projection are shifted in the x-y plane, Figs 3.23b-3.23c. This convention of shifting the grid and the centre of projection such that can be seen clearly is used henceforth.

As a further example of central projection consider the grid represented by the formex GRID and let this be projected onto a paraboloid using different values for the semiaxes \( b \) and \( c \) of the paraboloid. The other parameters in the descriptor remain as before.

The descriptor in this case may be given by

\[ [1,10,10,-10,3,10,10,20,12,30,15,13] \]
Fig 3.62

Fig 3.64
Fig 3.63
Fig 3.64 shows the plan view and elevation of the grid together with the paraboloid. A graphical representation of formex GRID projected on the paraboloid is shown in Fig 3.63b.

As the next example consider the grid represented by the formex GRID and let this be projected onto a hyperbolic paraboloid using central projection. For this purpose the tractation retronorm is used and the descriptor for tractation may be defined as

\[[P, xa, ya, za, S, a, b, c, t]\]

where $P$ is the projection specifier, $xa$, $ya$ and $za$ are the coordinates of the centre of projection with respect to the global coordinate system. $S$ is the surface specifier where $S=4$ implies a hyperbolic paraboloid and $a$, $b$ and $c$ are the semiaxes of the hyperbolic paraboloid. The value of $t=-13$ has been used and stands for the solution with the largest $z$.

Suppose that, the descriptor is specified as

\[[1,0,0,-45,4,15,20,25,-13]\]

where the first parameter $P=1$ implies central projection. The parameters 15, 20 and 25 defining the semiaxes of the hyperbolic paraboloid in the $x$, $y$ and directions, respectively. To explain how the projection takes place the plan view, the $yz$ elevation and $xz$ elevation of the grid together with the hyperbolic paraboloid on which the projection is to be obtained are shown in Fig 3.65. The dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut the hyperbolic paraboloid. This gives rise to the solution determined by $t=-13$. The solution is shown in Fig 3.65 with a thick line. A graphical representation of formex GRID projected on the hyperbolic paraboloid by using the tractation retronorm is as shown in Fig 3.66.
Fig 3.65

A is the centre of projection

a, b and c are the semiaxes of the hyperbolic paraboloid

Fig 3.66
3.9 CYLINDER

A cylindrical surface is generated by a straight line moving parallel to a given straight line, and intersecting a given curve, Fig 3.67. The moving line is called the generator or generatrix. The curve is called the directrix. The generatrix in any one fixed position is called an element. The cylinder may be circular, elliptic, parabolic,..., etc, according to the directrix being a circle, ellipse, parabola,..., etc. In Cartesian coordinates, the equations of circular, elliptic and parabolic cylinder with each of their elements parallel to the z axis, may be obtained as follows.

Circular Cylinder

A. If the centre of the circle is at the origin, then the equation is given as

\[ x^2 + y^2 = R^2 \]

where R is the radius of the circle, Fig 3.68a.

B. If the centre of the circle is at the point C(x_c, y_c), then the equation is given as

\[ (x-x_c)^2 + (y-y_c)^2 = R^2 \]

where R is the radius of the circle, Fig 3.68b.

Elliptic Cylinder

A. If the centre of the ellipse is at the origin, then the equation is given as

\[ \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \]

where a and b are the semi-axes of the ellipse, Fig 3.69a.
Fig 3.67

(a) The semiaxes of the ellipse.
(b) The semiaxes of the ellipse.

Fig 3.68

(a) A cylinder with axes.
(b) A cylinder with axes.

Fig 3.69

(a) A cylinder with axes and origin.
(b) A cylinder with axes and origin.

Fig 3.67

given line

directrix

generatrix

Fig 3.68

(a) A cylinder with axes.
(b) A cylinder with axes.

Fig 3.69

(a) A cylinder with axes and origin.
(b) A cylinder with axes and origin.
B. If the centre of the ellipse is at the point $C(x_c, y_c)$, then the equation is given as

$$\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} = 1$$

where $a$ and $b$ are the semiaxes of the ellipse, Fig 3.69b.

**Parabolic Cylinder**

A. If the centre is at the origin, Fig 3.70a, then the equation is given as

$$x^2 = y$$

B. If the centre is at the point $C(x_c, y_c)$, Fig 3.70b, then the equation is given as

$$(x-x_c)^2 = (y-y_c)$$

### 3.9.1 INTERSECTION OF A CIRCULAR CYLINDER AND A STRAIGHT LINE

Consider a straight line in 3-space whose equation is given as

$$\frac{x-x_a}{U1-x_a} = \frac{y-y_a}{U2-y_a} = \frac{z-z_a}{U3-z_a} \ldots \land U1-x_a \neq 0, U2-y_a \neq 0, U3-z_a \neq 0, 1$$

where the values $x_a, y_a, z_a$ of $U1, U2$ and $U3$ are as explained in section 3.2.1.

Eqn 1 will give rise to the equation

$$\frac{x-x_a}{U1-x_a} = \frac{y-y_a}{U2-y_a} \ldots 1a$$

The line described by Eqn 1 will intersect a circular cylinder whose equation is given by
\[(x-x_c)^2 + (y-y_c)^2 = R^2 \ldots \ldots \ldots \ldots 2\]

in points, whose coordinates may be obtained as the solutions of Eqns 1a and 2. Substitution of \(y\) from Eqn 1a in Eqn 2, gives

\[
(x-x_c)^2 \left( \frac{U_2-y_2}{U_1-x_a} \right) (x-x_a)+y_2-y_c = R^2 \ldots \ldots 3
\]

Eqn 3 is a quadratic equation giving two values of \(x\), the corresponding values of \(y\) may be obtained by substituting \(x\) into the Eqn 1a.

The two values of \(x\) are the roots of Eqn 3. These roots may be real and different, real and coincident, or both may be imaginary.

If the roots are real and different the line will intersect the circular cylinder in two points whose coordinates are given as follows

\[
x = \frac{-\beta + \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}
\]

\[
y = \frac{U_2-y_a}{U_1-x_a} \left( x-x_a \right) + y_a
\]

and

\[
x = \frac{-\beta - \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}
\]

\[
y = \frac{U_2-y_a}{U_1-x_a} \left( x-x_a \right) + y_a
\]

where
\[ \alpha = \left( \frac{U_2 - y_a}{U_1 - x_a} \right)^2 + 1 \]

\[ \beta = -2x + 2 \cdot \frac{U_2 - y_a}{U_1 - x_a} \cdot (y_a - y_c) - 2 \cdot \left( \frac{U_2 - y_a}{U_1 - x_a} \right)^2 \cdot x_a \]

and

\[ \gamma = x^2 + \left( \frac{U_2 - y_a}{U_1 - x_a} \right)^2 \cdot x_a^2 + (y_a - y_c)^2 - 2 \cdot \frac{U_2 - y_a}{U_1 - x_a} \cdot (y_a - y_c) \cdot x_a - R^2 \]

If the roots are coincident, the line is tangent to the circular cylinder and the required point is the point of tangency whose coordinates are given as

\[ x = \frac{-\beta}{2 \cdot \alpha} \]

\[ y = \frac{U_2 - y_a}{U_1 - x_a} \cdot (x - x_a) + y_a \]

where the values of \( \alpha \) and \( \beta \) are as given above.

If the roots are imaginary, the line will not intersect the circular cylinder at all.

### 3.9.2 Intersection of an Elliptic Cylinder and a Straight Line

Consider a straight line in 3-space whose equation is given as

\[ \frac{x - x_a}{U_1 - x_a} = \frac{y - y_a}{U_2 - y_a} = \frac{z - z_a}{U_3 - z_a} \ldots \]

where the values \( x_a, y_a, z_a, U_1, U_2 \) and \( U_3 \) are as explained in section 3.2.1.

Eqn 1 will give rise to the equation
The line described by Eqn 4 will intersect an elliptical cylinder whose equation is given by

\[
\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} = 1 \ldots .5
\]

in points, whose coordinates may be obtained as the solutions of Eqns 4a and 5. Substitution of y from Eqn 4a in Eqn 5, gives

\[
\frac{(x-x_c)^2}{a^2} + \left(\frac{U_2-y_a}{U_1-x_a} (x-x_a) + y_a - y_c\right)^2 \frac{(U_2-y_a)(x-x_a) + y_a - y_c}{b^2} = 1 \ldots .6
\]

Eqn 6 is a quadratic equation giving two values of x, the corresponding values of y may be obtained by substituting x into the Eqn 5a.

The two values of x are the roots of Eqn 6. These roots may be real and different, real and coincident, or may be imaginary.

If the roots are real and different the line will intersect the elliptic cylinder in two points whose coordinates are given as follows

\[
x = \frac{-b + \sqrt{b^2 - 4a + y}}{2a}
\]

\[
y = \frac{U_2-y_a}{U_1-x_a} (x-x_a) + y_a
\]

and
\[ x = \frac{-\beta - \sqrt{\beta^2 - 4\alpha \gamma}}{2\alpha} \]

\[ y = \frac{U_2 - y_a}{U_1 - x_a} \cdot (x - x_a) + y_a \]

where

\[ \alpha = b^2 \left( \frac{U_2 - y_a}{U_1 - x_a} \right)^2 \cdot a^2 \]

\[ \beta = -2b^2x_c - 2a^2x_a \left( \frac{U_2 - y_a}{U_1 - x_a} \right)^2 + 2a^2 \cdot \frac{U_2 - y_a}{U_1 - x_a} \cdot (y_a - y_c) \]

and

\[ \gamma = b^2x_c^2 + a^2x_a^2 \left( \frac{U_2 - y_a}{U_1 - x_a} \right)^2 + a^2 \cdot (y_a - y_c)^2 - 2a^2x_a \cdot \frac{U_2 - y_a}{U_1 - x_a} \cdot (y_a - y_c) - a^2b^2 \]

If the roots are coincident the line is tangent to the elliptic cylinder and the required point is the point of tangency whose coordinates are given as

\[ x = \frac{-\beta}{2\alpha} \]

\[ y = \frac{U_2 - y_a}{U_1 - x_a} \cdot (x - x_a) + y_a \]

where the values of \( \alpha \) and \( \beta \) are as given above.

If the roots are imaginary, the line will not intersect the elliptic cylinder at all.
3.9.3 INTERSECTION OF A PARABOLIC CYLINDER AND A STRAIGHT LINE

Consider a straight line in 3-space whose equation is given as

\[ \frac{x-x_a}{U_1-x_a} = \frac{y-y_a}{U_2-y_a} = \frac{z-z_a}{U_3-z_a} \ldots U_1-x_a \neq 0, U_2-y_a \neq 0, U_3-z_a \neq 0 \ldots \ldots 7 \]

where the values \( x_a, y_a, z_a \) and \( U_1, U_2, U_3 \) are as explained in section 3.2.1.

Eqn 7 will give rise to the equation

\[ \frac{y-y_a}{U_2-y_a} = \frac{x-x_a}{U_1-x_a} \ldots \ldots 7a \]

The line described by Eqn 7 will intersect a parabolic cylinder whose equation is given by

\[ (x-x_c)^2 = y-y_c \ldots \ldots 8 \]

in points, whose coordinates may be obtained as the solutions of Eqns 7a and 8. Substitution of \( y \) from Eqn 7a in Eqn 8 gives

\[ (x-x_c)^2 = \frac{U_2-y_a}{U_1-x_a} \cdot (x-x_a) + y_a-y_c \ldots \ldots 9 \]

Eqn 9 is a quadratic equation giving two values of \( x \), the corresponding values of \( y \) may be obtained by substituting \( x \) into the Eqn 7a.

The two values of \( x \) are the roots of Eqn 9. These roots may be real and different, real and coincident, or may be imaginary.

If the roots are real and different the line will intersect the parabolic cylinder in two points whose coordinates are given as follows
\[
x = \frac{-\beta + \sqrt{\beta^2 - 4\alpha \gamma}}{2\alpha}
\]

\[
y = \frac{U_2 - ya}{U_1 - xa} \ast (x - xa) + ya
\]

and

\[
x = \frac{-\beta - \sqrt{\beta^2 - 4\alpha \gamma}}{2\alpha}
\]

\[
y = \frac{U_2 - ya}{U_1 - xa} \ast (x - xa) + ya
\]

where

\(a = 1\)

\(\beta = -2 \ast x_c \ast \frac{U_2 - ya}{U_1 - xa}\)

and

\(\gamma = x_c^2 \ast x_a \ast \frac{U_2 - ya}{U_1 - xa} \ast (ya - yc)\)

If the roots are coincident the line is tangent to the parabolic cylinder and the required point is the point of tangency whose coordinates are given as

\[
x = \frac{-\beta}{2\alpha}
\]

\[
y = \frac{U_2 - ya}{U_1 - xa} \ast (x - xa) + ya
\]

where the values of \(\alpha\) and \(\beta\) are as given above.

If the roots are imaginary, the line will not intersect the parabolic cylinder at all.
3.9.4 EXAMPLES FOR PROJECTION ON A CYLINDER

The configurations generated in this section are the results of projecting a grid on a cylinder. The cylinder may be described as an infinitely long tube which may be circular, elliptic or parabolic with its generatrix parallel to the z axis, Figs 3.68-3.70. To begin with, consider the grid shown in Fig 3.71. The interconnection pattern of the configuration with respect to the indicated normat may be formulated as

\[
\text{GRID} = \text{PEX} | \text{RINIS}(8, 15, 2, 2) | \text{ROSAS}(1, 1) | \{[0, 1, 0; 2, 1, 0], [0, 1, 0; 1, 1, 1]\}
\]

Suppose that, this grid is to be projected on a circular cylinder. For this purpose the tractation retronorm is used and the descriptor for tractation may be defined as

\[
[P, x_a, y_a, z_a, S, x_c, y_c, R, t]
\]

where \( P \) is the projection specifier, \( x_a, y_a \) and \( z_a \) are the coordinates of the centre of projection with respect to the global coordinate system. \( S \) is the surface specifier, where \( S = 5 \) implies a circular cylinder. The centre of its circular directrix in the x-y plane is given by the coordinates \( x_c \) and \( y_c \) with respect to the global coordinate system. \( R \) is the radius of the circular directrix. The projection of a point on a surface such as a sphere, ellipsoid, cylinder or a paraboloid may be represented as a quadratic equation. Therefore, there may be two solutions for the projection as discussed before in section 3.5.

Suppose that, the descriptor is specified as

\[
[1, 8, -20, 15, 5, 8, 4, 12, 12]
\]

where the first parameter \( P = 1 \) implies central projection. To explain how the projection takes place, the plan view the elevation and side view of the grid together with the cylinder on which the projection is to be obtained are shown in Fig 3.72.
C is the centre of the directrix of the cylinder

A is the centre of projection
The dashed lines which are drawn from the centre of projection pass through the edges of the grid and cut through the cylinder. This gives rise to the solution determined by \( t=12 \). The value of \( t=12 \) has been used throughout this section and stands for the solution with larger \( y \). The solution is shown in Fig 3.72 by a thick line. A graphical representation of formex GRID projected on the circular cylinder is as shown in Fig 3.73a.

The viewing specifications that give rise to the configurations which are discussed in this section are provided by the USE-items

\[
\text{USE VM(3), VT(1), VZ(-10,-5,-15,26,21,45), VH(8,100,15,8,90,15,8,90,16)}
\]

The USE-items have been chosen such that a plan view is obtained.

In plan view, the projection of the grid, the grid itself and the centre of projection are on top of each other as shown in Fig 3.74a. In order to see them clearly the grid together with the centre of projection are moved in the x-z plane, Figs 3.74b-3.74c. This convention of shifting the grid and the centre of projection such that they can be seen clearly is used throughout this chapter.

Once again, consider the grid represented by the formex GRID and let this be projected onto a circular cylinder using a different radius and centre for the cylinder.

The descriptor in this case may be given by

\[ [1,8,-20,15,5,8,-1,5,12] \]

where the new centre and radius of the cylinder are given as \( C(8,-1) \) and \( R=5 \), respectively.

Fig 3.75 shows the plan view, elevation and side view of the grid together with the
Fig 3.73
Plan

circular cylinder

Side View

t=12
grid
circular cylinder

Elevation

t=12
grid
circular cylinder
Fig 3.74

Fig 3.76

Fig 3.77
circular cylinder. A graphical representation of formex GRID projected on the circular cylinder is shown in Fig 3.76.

In the descriptor given above the parameters have been chosen such as to bring out an interesting feature of the tractation retronorm. In the descriptor the radius of the circular directrix has been chosen as \( R = 5 \). The length of the grid is equal to 16. Since the circular cylinder is smaller as compared to the grid, only a part of the grid is projected onto the circular cylinder. The rest of the grid is not projected and remains as before.

Consider the grid shown in Fig 3.77a. A formex formulation for the configuration may be given by

\[
\text{CON= ' : ALPHA:} \\
\text{GRID=PEX | RINIS(8,15,2,2) | ROSAS(1,1) | \{(0,1,0;2,1,0),[0,1,0;1,1,1]\}} \\
\text{F1=TRAN(1,-8) | TRAN(3,-15) | GRID} \\
\text{F=VERAS(0,0,ALPHA) | F1} \\
\text{D=TRAC(1,0,-20,0,5,0,4,8,12) | F} \\
\text{USE VM(3), VT(1), VZ(-10,-30,-20,26,11,20), VH(0,100,0,0,90,0,0,90,1)} \\
\text{DRAW D'}
\]

The formulation is written as a Formian scheme in which formex variable \( F \) represents the grid relative to the indicated normat shown in Fig 3.77a and formex variable \( D \) the projected configuration. \( \text{ALPHA} \) is the angle shown in Fig 3.77b.

The scheme represents an infinite number of configurations, determined by various choices for the values of the parameter \( \text{ALPHA} \). The scheme is executed through induction statements listing the required values of the parameter. For example, the induction statement

\[
\text{CON(30)}
\]
corresponds to the configuration shown in Fig 3.78, the statement

CON(45)

corresponds to the configuration shown in Fig 3.79 the statement

CON(60)

corresponds to the configuration shown in Fig 3.80 and the statement

CON(90)

corresponds to the configuration shown in Fig 3.81.

To explain how the projection is obtained, the plan view, the elevation and the side view of the grid together with the cylinder are shown in Fig 3.82. The grid has been placed at an angle in the x-z plane. The value of the angle is given by ALPHA. In Fig 3.82 ALPHA is chosen as 45 degrees. The radius of the circular directrix has been chosen as R=8. The length of the grid is equal to 30. Since the cylinder is smaller than the grid, only a part of the grid is projected onto the cylinder. The rest of the grid is not projected and remains as before. Figs 3.78-3.81 show how different projections are obtained with different values for ALPHA.

As a further example of central projection consider the grid represented by the formex GRID and let this be projected onto an elliptic cylinder.

The descriptor in this case may be given by

\[ [P, x_a, y_a, z_a, S, x_c, y_c, a, b, t] \]

where P is the projection specifier, \( x_a, y_a \) and \( z_a \) are the coordinates of the centre of
A is the centre of projection

Plan

Side View

Elevation

Fig 3.82
projection with respect to the global coordinate system. $S$ is the surface specifier, where $S=6$ implies an elliptic cylinder. The centre of its elliptic directrix in the $x$-$y$ plane is given by the coordinates $xc$ and $yc$ with respect to the global coordinate system and $a$ and $b$ are the semiaxes of the ellipse.

Suppose that, the descriptor is specified as

$$[1,8,-20,15,6,8,4,12,18,12]$$

where $P=1$ implies central projection. The graphical representation of formex GRID projected on the elliptic cylinder is as shown in Fig 3.73b. Fig 3.83 shows the plan view and elevation of the grid together with the elliptic cylinder.

The above examples illustrate the concept of central projection. As mentioned before in tractation retronorm the user is able to choose between different types of projection, such as axial, parallel or radial. Each type of projection gives rise to a different configuration. Therefore, depending on the requirements, different types of projection may be used.

The following examples illustrate the concept of axial projection. To begin with, the corners of the grid shown in Fig 3.71 have been marked as KLMN for identification. Suppose that the grid is to be projected on a circular cylinder using axial projection. The descriptor for such a projection may take the form

$$[3,xa,ya,za,xb,yb,zb,S,xc,yc,R,12]$$

where the value of the first parameter, that is the projection specifier for axial projection is equal to 3. Also, $xa$, $ya$, $za$, $xb$, $yb$ and $zb$ are the coordinates of two points with respect to the global coordinate system defining the axis of projection $AB$.

If the descriptor is specified as
C is the centre of the directrix of the cylinder

a and b are the semiaxes elliptic directrix

A is the centre of projection

Fig 3.83
then a graphical representation of formex GRID projected on the circular cylinder using axial projection is as shown in Fig 3.84a. To explain how the projection takes place, the plan view, elevation and side view of the grid together with the circular cylinder on which the projection is to be obtained are shown in Fig 3.85. The axis of projection AB is drawn in the same figure. Once the axis AB is defined, the projection is obtained in a manner described in section 3.2.3. It should be noticed that in plan, the configuration has the same length along LN which is in the direction parallel to the axis of projection AB. In the direction perpendicular to the axis of projection, that is along corners KM, the length of the configuration varies depending on the position of the axis of projection. In the example given above, the configuration is elongated in the direction perpendicular to the axis of projection. Like projections on a sphere, ellipsoid or paraboloid, the axial projection for a cylinder gives rise to similar effects corresponding to different positions of the axis of projection.

In the example discussed above the axis of projection was in the x-z plane as shown in Fig 3.85. Now, let the axis be changed to a new position as shown in Fig 3.86. In this case the descriptor may be given as

\[ [3,16,-20,0,0,-20,30,5,8,4,12,12] \]

and a graphical representation of formex GRID projected on the circular cylinder is as shown in Fig 3.84b. Fig 3.86 shows the plan view, the elevation and the side view of the grid together with the cylinder.

As in the previous example the length of the configuration corresponds to the length of the flat grid in the direction which is parallel to the axis of projection. However, the width of the configuration can change depending on the position of the axis of projection. The position of the axis of projection in each case has been given in
Fig 3.84
**Fig 3.85**

C is the centre of the directrix of the cylinder

AB is the axis of projection

Elevation
C is the centre of the directrix of the cylinder

AB is the axis of projection

Fig 3.86
terms of its y-coordinate. That is, the x and z coordinates of the points A and B of the axis remaining the same and only the y-coordinates changing. Therefore, points A, B are given as (8, y, 0) and (8, y, 30). To begin with consider the example of axial projection which has been given above. In this case the axis of projection AB has y = -20 and the effect is as shown in Fig 3.87a. Now the axis AB is moved away from the grid by changing the y-coordinate to y = -40, Fig 3.88. It can be seen that as the axis moves away from the grid the configuration tends to be the same as the original grid, Fig 3.87b.

3.10 PLANE

A plane is a surface such that a straight line joining any two of its points lies entirely in the surface. The equation of a plane in three-dimensional Cartesian coordinates, is a polynomial equation of the first degree, which may be given as

\[ Ax + By + Cz + D = 0 \quad \ldots \ldots \ldots 1 \quad A^2+B^2+C^2! = 0 \]

This may be called the general form of the equation of a plane.

Three Point Form

The three point form of equation of the plane is expressed in terms of three points of the plane. Consider a plane M in space that is defined by the global coordinates of three points P1(x1, y1, z1), P2(x2, y2, z2) and P3(x3, y3, z3), Fig 3.89. To obtain the equation of the plane M, the parameters A, B, C and D have to be eliminated from Eqn 1. The three equations are obtained from Eqn 1 by the fact that P1, P2 and P3 lie on the plane M and may be given as follows:

\[ Ax_1 + By_1 + Cz_1 + D = 0 \]
\[ Ax_2 + By_2 + Cz_2 + D = 0 \]
Fig 3.87
C is the centre of the directrix of the cylinder.

AB is the axis of projection.

**Fig 3.88**
\[ Ax^3 + By^3 + Cz^3 + D = 0 \]

The result can be expressed in the determinant form

\[
\begin{vmatrix}
  x & y & z & 1 \\
  x_1 & y_1 & z_1 & 1 \\
  x_2 & y_2 & z_2 & 1 \\
  x_3 & y_3 & z_3 & 1 \\
\end{vmatrix} = 0
\]

elimination of the last column will give rise to determinant

\[
\begin{vmatrix}
  x-x_1 & y-y_1 & z-z_1 \\
  x_1-x_2 & y_1-y_2 & z_1-z_2 \\
  x_2-x_3 & y_2-y_3 & z_2-z_3 \\
\end{vmatrix} = 0
\]

The above determinant after manipulations will give rise to the equation

\[ Qx - Ry + Sz = E \ldots \ldots 2 \]

where

\[
Q = (y_1-y_2)(z_2-z_3) - (z_1-z_2)(y_2-y_3)
\]

\[
R = (x_1-x_2)(z_2-z_3) - (z_1-z_2)(x_2-x_3)
\]

\[
S = (x_1-x_2)(y_2-y_3) - (y_1-y_2)(x_2-x_3)
\]

and

\[ E = Qx_1 - Ry_1 + Sz_1 \]

Normal Form

Suppose that a nonzero vector is defined by the global coordinates of two points \( N(n_1,n_2,n_3) \) and \( T(t_1,t_2,t_3) \). Suppose \( M \) is a plane in space normal (perpendicular)
to the vector NT. Then plane M consists of all the points P(x,y,z) for which the vector PT is orthogonal to NT, Fig 3.90. That is, P lies in M if and only if

\[ NT*PT = 0 \]

or

\[ (n1-t1)*(x-t1) + (n2-t2)*(y-t2) + (n3-t3)*(z-t3) = 0 \]

This becomes

\[ (n1-t1)*x + (n2-t2)*y + (n3-t3)*z = (n1-t1)t1 + (n2-t2)t2 + (n3-t3)t3 \]

Rearrangement, will result in

\[ (n1-t1)*x + (n2-t2)*y + (n3-t3)*z = p \]

where

\[ p = (n1-t1)t1 + (n2-t2)t2 + (n3-t3)t3 \]

3.10.1 INTERSECTION OF A PLANE THAT IS SPECIFIED BY THREE POINTS AND A STRAIGHT LINE

Consider a straight line in 3-space whose equation is given as

\[ \frac{x-xa}{U1-xa} = \frac{y-ya}{U2-ya} = \frac{z-za}{U3-za} \ldots U1-xa\neq 0, U2-ya\neq 0, U3-za\neq 0 \ldots 3 \]

where the values xa, ya, za, U1, U2 and U3 are as explained in section 3.2.1.
Eqn 3 will give rise to the equations

\[
\frac{x-x_a}{U_1-x_a} = \frac{z-z_a}{U_3-z_a} \quad \ldots \ldots 3a
\]

and

\[
\frac{y-y_a}{U_2-y_a} = \frac{z-z_a}{U_3-z_a} \quad \ldots \ldots 3b
\]

The line described by Eqn 3 will intersect a plane whose equation is given by

\[
Qx - Ry + Sz = E \quad \ldots \ldots 4
\]

in points, whose coordinates may be obtained as the solutions of Eqns 3 and 4. Substitution of \(x\) and \(y\) from Eqns 3a and 3b in Eqn 4 gives

\[
Q* \left[ \frac{U_1-x_a}{U_3-z_a} * (z-z_a) + x_a \right] - R* \left[ \frac{U_2-y_a}{U_3-z_a} * (z-z_a) + y_a \right] + S*z = E \ldots \ldots 5
\]

Eqn 5 is a polynomial equation giving the value of \(z\), the corresponding values of \(x\) and \(y\) may be obtained by substituting \(z\) into the Eqns 3a and 3b. The coordinates may obtained as follows

\[
x = \frac{U_1-x_a}{U_3-z_a} * (z-z_a) + x_a
\]

\[
y = \frac{U_2-y_a}{U_3-z_a} * (z-z_a) + y_a
\]

and

\[
z = \frac{E - Q*x_a + R*y_a + \left[ Q* \frac{U_1-x_a}{U_3-z_a} - R* \frac{U_2-y_a}{U_3-z_a} \right] * z_a}{Q* \frac{U_1-x_a}{U_3-z_a} - R* \frac{U_2-y_a}{U_3-z_a} + S}
\]

where the values of \(Q\), \(R\) and \(S\) are as given above.
3.10.2 INTERSECTION OF A PLANE THAT IS SPECIFIED BY TWO 
POINTS DEFINING A VECTOR NORMAL TO THE PLANE AND A 
STRAIGHT LINE

Consider a straight line in 3-space whose equation is given as

\[
\frac{x-x_a}{U1-x_a} = \frac{y-y_a}{U2-y_a} = \frac{z-z_a}{U3-z_a} \ldots U1-x_a \neq 0, U2-y_a \neq 0, U3-z_a \neq 0 \ldots 6
\]

where the values \( x_a, y_a, z_a, U1, U2 \) and \( U3 \) are as explained before.

This will give rise to the equations

\[
\frac{x-x_a}{U1-x_a} = \frac{z-z_a}{U3-z_a} \ldots 6a
\]

and

\[
\frac{y-y_a}{U2-y_a} = \frac{z-z_a}{U3-z_a} \ldots 6b
\]

This line will intersect a plane whose equation is given by

\[
(n1-t1)*x + (n2-t2)*y + (n3-t3)*z = p \ldots \ldots 7
\]

in points, whose coordinates may be obtained as the solutions of Eqns 6 and 7. Substitution of \( x \) and \( y \) from Eqns 6a and 6b in Eqn 7 gives

\[
(n1-t1)*\left[\frac{U1-x_a}{U3-z_a}*(z-z_a)+x_a\right] +
(n2-t2)*\left[\frac{U2-y_a}{U3-z_a}*(z-z_a)+y_a\right] + (n3-t3)*z = p \ldots \ldots 8
\]

This is a polynomial equation giving the value of \( z \), the corresponding values of \( x \) and \( y \) may be obtained by substituting \( z \) into the Eqns 6a and 6b. The coordinates may be obtained as follows.
\[ x = \frac{U_1 - x_a}{U_3 - z_a} \ast (z - z_a) + x_a \]

\[ y = \frac{U_2 - y_a}{U_3 - z_a} \ast (z - z_a) + y_a \]

and

\[ z = \frac{p - (n_1 - t_1)x_a - (n_2 - t_2)y_a + z_a}{(n_1 - t_1)(U_1 - x_a) + (n_2 - t_2)(U_2 - y_a) + (n_3 - t_3)} \]

where the values \( n_1, n_2, n_3, t_1, t_2, t_3 \) and \( p \) are as given above.

### 3.11 MULTI-LAYER GRID DOMES

The above examples illustrate the use of the tractation retronorm to project a configuration on a single surface. It is also possible to project a configuration on several surfaces in which case different descriptors for each of the surfaces have to be defined. The general form of the tractation retronorm may be given as

\[ \text{TRAC}(D_1, D_2, \ldots, D_n) \]

where each one of the entities \( D_1, D_2, \ldots, D_n \) is a descriptor which describes type of projection, type of surfaces, and set of signets to be projected.

The parameters \( h \) and \( U \) are for specification of the set of signets that belong to different layers of a configuration. In the above examples the configuration had to be projected on only one surface hence the specification of the values of \( h \) and \( U \) were not necessary.

If \( F \) is the formex of the configuration to be projected then \( h \) is an integer number
specifying the uniple of a signet which has to be examined for projection. The value of \( U \) determines the signets of formex \( F \) which may be projected on the specified surface.

For example,

\[ [2,3,4,5] \]

and

\[ [7,8,9,1] \]

are two signets of grade four. Also, if \( h=3 \) then \( U \) may be 4 or 9. If \( U=9 \) the second signet will be projected.

Consider the double layer grid of Fig 3.91 whose bottom layer elements are to be projected on a sphere of radius 20 and whose top layer elements are to be projected on a sphere of radius 25. Therefore, two descriptors have to be defined. The formex formulation for the double layer grid may be given by

\[
GRID='::
\]

\[
TOP=PEX|RINID(10,10,2,2)|ROSAD(1,1)|[0,0,2; 2,0,2]
\]

\[
WEB=RINID(10,10,2,2)|ROSAD(1,1)|[0,0,2; 1,1,1]
\]

\[
BTM=PEX|RINID(9,9,2,2)|ROSAD(2,2)|{1,1,1; 3,1,1}
\]

\[
F=TOP#WEB#BTM
\]

\[
D=TRAC([3,1,4,1,10,10,-5,20,13],[3,2,4,1,10,10,-5,25,13])|F'
\]

From the definition of \( h \), in the formulation given above \( h \) may be 1, 2 or 3. In the formex formulation the third uniple of each signet determines the layer of the grid. Therefore, the value of \( h \) will be 3. In the formulation the third uniple has values 1 or 2 where 1 stands for the bottom layer elements and 2 stands for top layer elements. Hence, the value of \( U \) will either be 1 or 2, representing the bottom layer elements and top layer elements, respectively. As the values of \( h \) and \( U \) have been
defined all the signets whose third uniple is 1 will be projected on the sphere of \( R=20 \) and all the signets whose third uniple is 2 will be projected on the sphere of \( R=25 \). A graphical representation of formex variable \( D \) is shown in Fig 3.92. In Figs 3.91 and 3.92 the top elements are drawn in thick lines and the bottom layer elements as well as the web elements are drawn in thin lines.
Chapter 4
Polyhedric and Geodesic Forms

INTRODUCTION

Geodesic forms constitute an important family of structural systems. They are efficient and appealing and are employed frequently for dome structures. Data generation for geodesic forms has always been a difficult task, from the days of hand data generation using tables of coefficients to today’s specially written computer programs with many limitations and shortcomings. The objective of the present chapter is to introduce the concepts and constructs through which data generation for geodesic forms of all kind can be handled with ease and elegance.

The generation of geodesic forms is solved in two stages. Firstly, a function called the polyhedron function is used to generate a configuration modelled on a polyhedron. The resulting configuration is referred to as a "polyhedric configuration". The polyhedron function constitutes the kernel of the problem handling strategy for the data generation of geodesic forms.

In the next stage, the tractation retronorm is employed to obtain the projection of the
polyhedric configuration on one or more specified surfaces. All the surfaces available in the tractation retronorm may serve the purpose of shaping geodesic forms. Different types of projection such as central axial or parallel may also be used to generate interesting geodesic forms.

4.1 THE PLATONIC AND ARCHIMEDEAN POLYHEDRA

The first stage in the generation of a geodesic form is the creation of the polyhedric configuration. The approach presented in this chapter provides a methodology that allows polyhedric configurations of all kinds to be generated in a convenient manner. However, polyhedric configurations based on regular polyhedra (known as Platonic solids) or semiregular (Archimedean) polyhedra are used in this study to demonstrate the concepts and constructs through which polyhedric configurations may be created.

A "polyhedron" is a surface composed of plane polygonal surfaces, the "faces". The sides of the polygons joining two faces, are its "edges". The corners, where three or more faces meet, are its "vertices".

A "regular polyhedron" is a polyhedron whose faces are congruent regular polygons. Every vertex is to be congruent to every other vertex, that is, the faces must be arranged in the same order around each vertex. There are only five regular polyhedra, these are

1) Tetrahedron (4 faced)
2) Cube (Hexahedron) (6 faced)
3) Octahedron (8 faced)
4) Dodecahedron (12 faced)
5) Icosahedron (20 faced)

and are as shown in Figs 4.1-4.5
The Platonic solids were not discovered by Plato, but they have been so named because of their appearance in the Timaues (See Ref 2). In Timaues, Plato identifies four of the regular polyhedra with the four elements, the tetrahedron represents fire; the cube, earth; the octahedron, air and the icosahedron, water. The fifth and last regular polyhedron, the dodecahedron represents the quintessence of which the heavenly bodies are made and in which the four elements are impregnated.

Each Platonic polyhedron can be surrounded by a sphere so that all its vertices touch that sphere. This sphere which circumscribes a Platonic polyhedron is referred to as the "circumscribing sphere" or "circumsphere" of the polyhedron, Figs 4.6 and 4.7 (The centre of the polyhedron is the same as the centre of its circumsphere). Since the vertices of a Platonic polyhedron touch the circumsphere, its edges will be chords to that sphere. Since the chords are equal in length their midpoints are at a constant distance from the centre, so a smaller concentric sphere, called an "intersphere" can be constructed, Figs 4.6 and 4.8. This sphere touches the midpoints of each edge. The centre of all the faces of each Platonic polyhedron also are at a constant distance from the centre of the polyhedron, so a third smaller concentric sphere called an "insphere" can be constructed to touch every face, Figs 4.6 and 4.9. It should also be mentioned that all pairs of adjacent faces of a particular regular polyhedron meet at a constant angle. This angle called the "dihedral angle", Fig 4.10. The dihedral angle is the angle formed between the planes of two adjacent polygons, the angle taken in a plane perpendicular to the common edge and is less than 180 degrees. All the dihedral angles for each of the regular polyhedra are equal.

Plates 4.1 to 4.5 give some useful data for the Platonic polyhedra. In these tables each polyhedron in shown together with the global Cartesian x-y-z coordinate system. The origin of the coordinate system is at the centre of the polyhedron and is indicated with a large dot. The point where the positive side of the x-axis intersects the polyhedron is indicated by a little circle with an enclosed x. This point is referred to as the "x-point". The positions of the positive directions of y- and z- axes are indicated by arrows with z being always vertical. Each face of the polyhedron is
Circumsphere  Intersphere  Insphere

Fig 4.7  Fig 4.8  Fig 4.9

Fig 4.10  Fig 4.11

Fig 4.12  Truncated Tetrahedron  Fig 4.13  Cuboctahedron
### Plate 4.1

#### TETRAHEDRON

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 triangles</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: $(\sqrt{6/4})L$
- **Radius of Intersphere**: $(\sqrt{2/4})L$
- **Radius of Insphere**: $(\sqrt{6/12})L$
- **Dihedral Angle**: $\cos^{-1}(1/3)$

$L$ denotes the edge length.

---

**Face View**

**Edge View**

**Vertex View**
Hexahedron

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 squares</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

Radius of Circumsphere: $(\sqrt{3}/2)L$
Radius of Intersphere: $(\sqrt{2}/2)L$
Radius of Insphere: $0.5L$
Dihedral Angle: $\arccos 0$

$L$ denotes the edge length

Face View | Edge View | Vertex View
### OCTAHEDRON

**Faces**: 8 triangles  
**Edges**: 12  
**Vertices**: 6

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 triangles</td>
<td>12</td>
<td>6</td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: \((\sqrt{2}/2) \times L\)
- **Radius of Intersphere**: \(0.5 \times L\)
- **Radius of Insphere**: \((\sqrt{6}/6) \times L\)
- **Dihedral Angle**: \(\arccos(-1/3)\)

**L denotes the edge length**

---

**Face View**  
**Edge View**  
**Vertex View**
<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 pentagons</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Radius of Circumsphere</td>
<td>$\frac{3L}{\sqrt{15} - \sqrt{13}}$</td>
<td></td>
</tr>
<tr>
<td>Radius of Intersphere</td>
<td>$(1 + \cos 72) L$</td>
<td></td>
</tr>
<tr>
<td>Radius of Insphere</td>
<td>$(\frac{3L}{\sqrt{15} - \sqrt{13}}) \cdot \sqrt{((5 + 2\sqrt{5})/15)}$</td>
<td></td>
</tr>
<tr>
<td>Dihedral Angle</td>
<td>$180 - \arctan 2$</td>
<td></td>
</tr>
</tbody>
</table>

L denotes the edge length
### ICOSAHEDRON

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 triangles</td>
<td>30</td>
<td>12</td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: $\frac{5L}{\sqrt{10}\sqrt{5}}$
- **Radius of Intersphere**: $\cos 36^\circ L$
- **Radius of Insphere**: $\frac{L(1 + \sqrt{5})}{8\sqrt{113}}$
- **Dihedral Angle**: $180 - \arcsin \left( \frac{2}{3} \right)$

$L$ denotes the edge length

---

**Face View**  
**Edge View**  
**Vertex View**
identified with a "face code" which is given at one corner of the face. A face code consists of a number followed by a letter and possibly followed by an asterisk. The number in a face code is the identification number of the face. The letter in a face code determines the points A, B, C, ..., etc of the configuration that is to be placed on the indicated corner of the face. These letters for configurations corresponding to different shapes of polyhedral faces are as shown in Fig 4.11. If a face code has an asterisk, it implies that the configuration which is to be placed on the face is the reflection, with respect the x-y plane, of the given configuration.

The Archimedean or "semiregular polyhedra" are what is called "facially" regular polyhedra. This means that every face is a regular polygon though the faces are not all of the same kind. However, every vertex is to be congruent to every other vertex that is, the faces must be arranged in the same order around each vertex. The Archimedean polyhedra were discovered in ancient Greece and were described by Archimedes thus they are called the Archimedean polyhedra. His writings on the semiregular polyhedra were lost together with the knowledge of the figures. During the Renaissance they were gradually rediscovered and were described by such people as Pierro della Francesca and Albrecht Durer, though the description of the Archimedean polyhedra did not appear until Johannes Kepler's Harmonices Mundi was published in 1619. The Archimedean solids are

1) truncated tetrahedron
2) cuboctahedron
3) truncated cube
4) truncated octahedron
5) small rhombicuboctahedron
6) great rhombicuboctahedron (truncated cuboctahedron)
7) icosidodecahedron
8) truncated dodecahedron
9) truncated icosahedron
10) snub cube (right handed and left handed version)
11) small rhombicosidodecahedron
12) great rhombicosidodecahedron (truncated icosidodecahedron)
13) snub dodecahedron (right handed and left handed version)

The Archimedean polyhedra are shown in Figs 4.12-4.24. These polyhedra consist of various combinations of triangles, squares, pentagons, hexagons, octagons and decagons, Fig 4.11. The letter which is given in each vertex of a polyhedral face is a constituent of the face code and is as explained before. Ten of the Archimedean polyhedra utilize only two kinds of polygons and the remaining three utilize three kinds of polygons, as shown in Table 4.1. Some useful data for the Archimedean solids are given in Plates 4.6-4.18. These tables should be interpreted in a way similar to the one given for the Platonic polyhedra. The significance of the number, the letter and the asterisk which are given for each one of the faces of an Archimedean polyhedron in the tables is as explained before.

Eleven of the Archimedean polyhedra can be derived from the five Platonic polyhedra by truncations of vertices and/or edges, Figs 4.25 to 4.27. The construction of the last two semiregular polyhedra, the snub cube and the snub dodecahedron is still based on the truncation of the edges of a regular polyhedron. Both of them have faces which are located on the faces of the original regular polyhedron, comprising regular polygons with the same number of sides but smaller and rotated in a particular direction, Figs 4.26 and 4.27. This process is referred to as "snubbing". The rotation may be right handed or left handed. Therefore, the snub cube and snub dodecahedron occur in two versions, each version is the mirror image of the other as shown in Figs 4.28 and 4.29. The snub cube shown in Fig 4.28 cannot be rotated to make it coincide with the snub cube shown in Fig 4.29. It will always be a right handed (dextro) or a left handed (laevo) version of the polyhedron. Such polyhedra are called "enantiomorphic". The right handed versions of the snub cube and the snub dodecahedron have been used in this study, (ref 34).

The sheer length of many of the names of the Archimedean polyhedra may deter
Fig 4.14 Truncated Cube

Fig 4.15 Truncated Octahedron

Fig 4.16 Small Rhombicuboctahedron

Fig 4.17 Great Rhombicuboctahedron

Fig 4.18 Icosidodecahedron

Fig 4.19 Truncated Dodecahedron
Fig 4.20 Truncated Icosahedron

Fig 4.21 Snub Cube

Fig 4.22 Small Rhombicosidodecahedron

Fig 4.23 Great Rhombicosidodecahedron

Fig 4.24 Snub Dodecahedron

Fig 4.25 Truncated Tetrahedron
Table 4.1

<table>
<thead>
<tr>
<th>Polyhedron</th>
<th>Type and Number of Faces</th>
<th>Polyhedron</th>
<th>Type and Number of Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncated Tetrahedron</td>
<td><img src="image" alt="4 triangles" /> 4</td>
<td>Truncated Dodecahedron</td>
<td><img src="image" alt="20 triangles" /> 12</td>
</tr>
<tr>
<td>Cuboctahedron</td>
<td><img src="image" alt="8 triangles" /> 6</td>
<td>Truncated Icosahedron</td>
<td><img src="image" alt="12 pentagons" /> 20</td>
</tr>
<tr>
<td>Truncated Cube</td>
<td><img src="image" alt="8 triangles" /> 6</td>
<td>Snub Cube</td>
<td><img src="image" alt="32 triangles" /> 6</td>
</tr>
<tr>
<td>Truncated Octahedron</td>
<td><img src="image" alt="6 squares" /> 8</td>
<td>Small Rhombicosidodecahedron</td>
<td><img src="image" alt="20 triangles" /> 30 <img src="image" alt="12 squares" /></td>
</tr>
<tr>
<td>Small Rhombicuboctahedron</td>
<td><img src="image" alt="8 triangles" /> 18</td>
<td>Great Rhombicosidodecahedron</td>
<td><img src="image" alt="30 squares" /> 20 <img src="image" alt="12 squares" /></td>
</tr>
<tr>
<td>Great Rhombicuboctahedron</td>
<td><img src="image" alt="12 squares" /> 8 <img src="image" alt="6 squares" /></td>
<td>Snub Dodecahedron</td>
<td><img src="image" alt="80 triangles" /> 12</td>
</tr>
<tr>
<td>Icosidodecahedron</td>
<td><img src="image" alt="20 triangles" /> 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Plate 4.6

**TRUNCATED TETRAHEDRON**

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 triangles</td>
<td>18</td>
<td>12</td>
</tr>
<tr>
<td>4 hexagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: $\sqrt{\frac{11}{8}}L$
- **Radius of Intersphere**: $\frac{3L}{\sqrt{8}}$
- **Dihedral Angle 3-6**: $\cos^{-1}\left(-\frac{1}{3}\right)$
- **Dihedral Angle 6-6**: $\cos^{-1}\left(\frac{1}{3}\right)$

L denotes the edge length

---

**Face View**  **Edge View**  **Vertex View**
CUBOCTAHEDRON

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 triangles</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td>6 squares</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Radius of Circumsphere: \( L \)
Radius of Intersphere: \( \sqrt{3/2}L \)
Dihedral Angle: \( \cos(-\sqrt{13}/3) \)

L denotes the edge length

Face View  Edge View  Vertex View
### TRUNCATED CUBE

#### Faces
- 8 triangles
- 6 octagons

#### Edges
- 36

#### Vertices
- 24

#### Radii
- **Radius of Circumsphere**: $L \cdot \sqrt[3]{\frac{1}{(2 - \sqrt{2})^3} + \frac{1}{4}}$
- **Radius of Intersphere**: $L / (2 - \sqrt{2})$

#### Dihedral Angles
- **Dihedral Angle 3-8**: $\arccos\left(-\frac{\sqrt{13}}{3}\right)$
- **Dihedral Angle 8-8**: $\arccos\left(-\frac{\sqrt{2}}{2}\right)$

$L$ denotes the edge length

---

**Face View**

**Edge View**

**Vertex View**

---

323
Plate 4.9

**TRUNCATED OCTAHEDRON**

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 squares</td>
<td>36</td>
<td>24</td>
</tr>
<tr>
<td>8 hexagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Radius of Circumsphere: $\sqrt{10*L/2}$

Radius of Intersphere: $1.5*L$

Dihedral Angle 4-6: $\cos\left(-\frac{\sqrt{13}}{3}\right)$

Dihedral Angle 6-6: $\cos\left(-\frac{1}{3}\right)$

$L$ denotes the edge length

---

Face View | Edge View | Vertex View
---|---|---

324
**SMALL RHOMBICUBOCTAHEDRON**

![Small Rhombicuboctahedron Diagram]

<table>
<thead>
<tr>
<th>Face</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 triangles</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>18 squares</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: \( L \times \sqrt{\frac{5}{4} + \frac{1}{\sqrt{2}}} \)
- **Radius of Intersphere**: \( L \times \sqrt{\frac{2 - \sqrt{2}}{2}} \)
- **Dihedral Angle 3-4**: \( \text{ACOS}\left(\frac{-\sqrt{11}}{6}\right) \)
- **Dihedral Angle 4-4**: \( \text{ACOS}\left(\frac{-\sqrt{12}}{2}\right) \)

L denotes the edge length.

![Face View](image1)
![Edge View](image2)
![Vertex View](image3)
**Plate 4.11**

**GREAT RHOMBICUBOCTAHEDRON**

<table>
<thead>
<tr>
<th>Face</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 squares</td>
<td>72</td>
<td>48</td>
</tr>
<tr>
<td>8 hexagons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 octagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Radius of Circumsphere: \( L \times \sqrt{\frac{3}{2 - \sqrt{12}}} + \frac{1}{4} \)
- Radius of Intersphere: \( L \times \sqrt{\frac{31}{2 - \sqrt{12}}} \)
- Dihedral Angle 4-6: \( \text{ACOS} \left( -\sqrt{\frac{6}{3}} \right) \)
- Dihedral Angle 4-8: \( \text{ACOS} \left( -\sqrt{\frac{2}{2}} \right) \)
- Dihedral Angle 6-8: \( \text{ACOS} \left( -\sqrt{\frac{3}{3}} \right) \)

\( L \) denotes the edge length

---

**Face View**  | **Edge View**  | **Vertex View**
---|---|---
### ICOSIDODECAHEDRON

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 triangles</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>12 pentagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: $(\sqrt{5}+1)\frac{L}{2}$
- **Radius of Intersphere**: $\tan\frac{72}{2}\frac{L}{2}$
- **Dihedral Angle**: $\arccos\left(-\frac{\tan 54}{\tan 60}\right)$

$L$ denotes the edge length

---

**Face View**

**Edge View**

**Vertex View**
# TRUNCATED DODECAHEDRON

![Diagram of a truncated dodecahedron]

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 triangles</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>12 decagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Radius of Circumsphere: \( L \times \sqrt{\frac{1}{3/\sqrt{15} - 1} + \frac{1}{4}} \)

Radius of Intersphere: \( L / (3/\sqrt{15} - 1) \)

Dihedral Angle 3-10: \( \text{ACOS}\left(-\text{TAN}\left(\frac{54}{\text{TAN}\left(60\right)}\right)\right) \)

Dihedral Angle 10-10: \( 180 - \text{ATAN}(2) \)

L denotes the edge length
### TRUNCATED ICOSAHEDRON

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 pentagons</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>20 hexagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Radius of Circumsphere**: \( L \times \sqrt{\frac{9}{\sqrt{15} - 1} + \frac{1}{4}} \)
- **Radius of Intersphere**: \( 3 \times \frac{L}{\sqrt{5 - 1}} \)
- **Dihedral Angle 5-6**: \( \arccos\left(-\frac{\sqrt{54}}{\tan \frac{160}{2}}\right) \)
- **Dihedral Angle 6-6**: \( 180 + \arcsin\left(-\frac{2}{3}\right) \)

\( L \) denotes the edge length

---

Face View | Edge View | Vertex View
---|---|---
Plate 4.15

**SNUB CUBE**

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 triangles</td>
<td>60</td>
<td>24</td>
</tr>
<tr>
<td>6 squares</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Radius of Circumsphere: 1.3437*L
Radius of Intersphere: 1.2472*L
Dihedral Angle 3-3: 153°14’
Dihedral Angle 3-4: 142°59’

L denotes the edge length

---

**Face View**  **Edge View**  **Vertex View**

330
SMALL RIOMICOSIDODECAIHEDRON

<table>
<thead>
<tr>
<th>Face</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 triangles</td>
<td>30 squares</td>
<td>12 pentagons</td>
</tr>
</tbody>
</table>

Radius of Circumsphere: \( L \cdot \text{SQRT} \left( \text{TAN} \left( \frac{72}{2} + \frac{1}{4} \right) \right) \)

Radius of Intersphere: \( L \cdot \text{TAN} \left( \frac{72}{\text{SQRT} 2} \right) \)

Dihedral Angle 3-4: \( \text{ACOS} \left( -\left( 1 + \text{SQRT} 5 \right) / 2 \right) \cdot \text{SQRT} 3 \)

Dihedral Angle 4-5: \( 180 + \text{ATAN} \left( -\left( \text{SQRT} 5 - 1 \right) / 2 \right) \)

L denotes the edge length
GREAT RHOMBICOSIDODECAHEDRON

**Face Edges Vertices**

<table>
<thead>
<tr>
<th>Face</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 squares</td>
<td>180</td>
<td>120</td>
</tr>
<tr>
<td>20 hexagons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 decagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Radius of Circumsphere: \( L \cdot \text{SQRT}\left(\frac{3\cdot\tan^2 72}{2} + \frac{1}{4}\right) \)

Radius of Intersphere: \( L \cdot \tan \frac{72}{\sqrt{3}} \cdot \frac{2}{3} \)

Dihedral Angle 4-6: \( \text{ACOS}\left(-1 + \text{SQRT}\left|\frac{5}{2}\right| \cdot \text{SQRT}\left|3\right|\right) \)

Dihedral Angle 4-10: \( 180 + \text{ATAN}\left(-\left(\text{SQRT}\left|5\right|-1\right)\right) \)

Dihedral Angle 6-10: \( \text{ACOS}\left(-\text{TAN}\left|\frac{54}{\text{TAN}\left|60\right|}\right|\right) \)

L denotes the edge length

Face View

Edge View

Vertex View
**Plate 4.18**

**SNUB DODECAHEDRON**

<table>
<thead>
<tr>
<th>Faces</th>
<th>Edges</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 triangles</td>
<td>150</td>
<td>60</td>
</tr>
<tr>
<td>12 pentagons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Radius of Circumsphere: $2.1557^*L$
- Radius of Intersphere: $2.0969^*L$
- Dihedral Angle 3-3: $164^\circ11'$
- Dihedral Angle 3-4: $152^\circ56'$

*L denotes the edge length*
Fig 4.26

Snub Cube

Cube

Octahedron

Truncation (3)

Truncated Cube

Truncated Octahedron

TRUNCATION (2)

Cuboctahedron

TRUNCATION (3)

Small Rhombicuboctahedron

Great Rhombicuboctahedron

Snubbing

AFFINITY

AFFINITY
some people but they are descriptive and can be easy to remember. The cuboctahedron can be seen to have obvious relationships to both the cube and the octahedron. The icosidodecahedron has relationships to both the icosahedron and the dodecahedron. The word "truncated" in such names as truncated tetrahedron clearly describes a polyhedron, in this case a tetrahedron which has had its extremities removed or truncated. The great rhombicuboctahedron and great rhombicosidodecahedron are sometimes called the truncated cuboctahedron and the truncated icosidodecahedron. But their former names are used in this study. The prefixes "great" and "small" in the names of the rhombicuboctahedra and rhombicosidodecahedra differentiate between figures in terms of their sizes. The word "snub" is an old Norwegian word, a literal translation being "snub-nosed" or flat-nosed".

A Platonic polyhedron has a constant dihedral angle between its adjacent faces. However, of the semiregular polyhedra only the cuboctahedron and the icosidodecahedron each have constant dihedral angles but the other eleven Archimedean polyhedra have two or three different dihedral angles. Each Archimedean polyhedron can be circumscribed by one of the Platonic polyhedra so that all of its vertices lie evenly arranged on the faces or edges of the circumscribing figure as shown in Figs 4.30 and 4.31. In these figures a cuboctahedron and a great rhombicuboctahedron circumscribed by cubes. In each case the vertices of the inscribed polyhedron are at an equal distance from the centre of the faces of the circumscribing polyhedron. Since the centre of the faces of each Platonic polyhedron are at a constant distance from the centre of the polyhedron itself, all the vertices of the inscribed polyhedron must also be at a constant distance from it. Hence, like in a Platonic polyhedron an Archimedean polyhedron has a circumsphere touching all of its vertices. Since the edges of an Archimedean polyhedron are equal in length, they are also equal chords to its circumsphere and their midpoints are an equal distance from the centre of that circumsphere. Therefore, an Archimedean polyhedron can have an intersphere which touch all of its edges. The face centres of different types of face are not at a constant distance from the centre of the polyhedron, so, no
Fig 4.28  dextro

Fig 4.29  laevo

Fig 4.30

Fig 4.31

Fig 4.32

ALTERNATE METHOD
insphere can be constructed to touch every face of an Archimedean polyhedron.

There are many ways in which the faces of a polyhedron can be subdivided but two methods, the "Alternate Method" and the "Triacon Method", will be described at this stage. Both of the methods can only be used on faces which are triangles or have been triangulated. Hence, if a cube or a dodecahedron is to be used, its faces must first be triangulated, (Ref 31).

In the "Alternate Method" each triangulated face can be subdivided into a series of smaller triangles by lines running parallel to the original edges of the triangle, Fig 4.32. In the "Triacon Method" each triangulated face subdivided into small triangles by lines joining the vertices of each triangle to the midpoints of their opposite edges divided the triangle into six parts, Fig 4.33. Each of the lines defines an axis and higher frequency subdivisions can be achieved by drawing additional lines parallel to those lines. In both the subdivisions each face has been named according to the number of parts into which its original edges have been divided. That is, the original edges of a two frequency (2f) have been divided into two parts, the edge of a four frequency (4f) face into four parts and so on. Only low frequencies subdivisions are shown here, but the idea can be extended to much higher frequencies. Notice that in the Triacon Method the faces can only be divided into frequencies that are even numbers.

4.2 POLYHEDRON FUNCTION

Consider a polyhedron and let some given configurations be placed on its faces. The result is referred to as a "polyhedric configuration" or "polyhedric form". This term may also be used to refer to a portion of a polyhedric configuration. A polyhedron which is used as the basis for the creation of a polyhedric configuration or form is referred to as the "base polyhedron" of the polyhedric form. There is a formex function called "polyhedron function" that can be used to create formices
Fig 4.33

TRIACON METHOD

Fig 4.34

Fig 4.35

Fig 4.36

Plan View

Elevation
representing polyhedric configurations.

The applications of the polyhedron function may be described with the help of an example. Consider a single layer triangular configuration which will be referred to in the sequel as the configuration. The configuration together with the normat U1-U2-U3 for the formex formulation are shown in Fig 4.34. This configuration may be represented in terms of the formex variable

\[ E = \text{LIB(I=0,5)} | \text{RIN(1,6-I,2)} | \text{TRANID(I,I)} | \{(0,0;2,0),[2,0;1,1],[1,1;0,0]\} \]

Let it be required to map this configuration onto all the faces of a tetrahedron. A Formian statement describing this operation may be given as

\[ D = \text{PEX|POL(1,15,[0,0;12,0])|G} \]

where

\[ G = \text{BB(I,TAN 60)} | E \]

As has been stated in section 4.1 a tetrahedron has four equilateral triangular faces. The triangular configuration that is described here by formex E is not equilateral. Therefore, it will be necessary to scale the configuration using the appropriate scale factors. Formex variable G specifies the scale factors that are used in the first and second direction to obtain the equilateral triangular configuration of Fig 4.35. Formex variable G represents the actual node coordinates while formex variable E represents the corresponding normat coordinates of the configuration. A graphical representation of formex variable D is as shown in Fig 4.36. Also, in the figure the plan view and the elevation of the polyhedral configuration have been given together.
with the global Cartesian coordinate system.

The construct

\[
\text{POL}(1, 15, [0,0; 12,0])
\]

is a formex function representing a rule for transformation of a given formex \( G \) into a formex \( D \). The parameters 1, 15, \([0,0; 12,0]\), are parts of the rule defining the particulars of the transformation and are referred to as canonic parameters. The above function is referred to as a "polyhedron function". The polyhedron function can be used to create single layer or multi layer polyhedric configurations. The general form of the polyhedron function for single layer polyhedric configurations may be written as

\[
\text{POL}(P, R, [A_1, A_2; B_1, B_2] < <, \{F_1, F_2, \ldots, F_n\} > >)
\]

where the first canonic parameter \( P \) is referred to as the "polyhedron code". The polyhedron code specifies the type of polyhedron which is to be used as the basis for the operation. Table 4.2 lists the code numbers for the Platonic and Archimedean polyhedra. The integer 1 given as the polyhedron code in the above polyhedron function specifies a tetrahedron. The "radius specifier" determines the size of the polyhedron by specifying the radius of its circumsphere, that is, the sphere that contains all the vertices of the polyhedron. This parameter is given as 15 units of length. The "locator" specifies the manner in which a given configuration is to be mapped onto a face of the polyhedron. To elaborate, consider the configuration shown in Fig 4.35. Two corners of the configuration are denoted by the letters A and B. The configuration is intended to be placed on a face of the tetrahedron in such a way that AB fits an edge of the tetrahedron. This convention in conveyed by including the \( U_1-U_2 \) coordinates of A and B in the locator.

The last canonic parameter is referred to as the "face list". The role of the face list
is to specify those faces of the polyhedron onto which the configuration is to be mapped. The face list is enclosed in special brackets which are referred to as option brackets. Absence of the face list in the polyhedron function implies that the configuration is to be mapped onto all the faces of that polyhedron. However, one has the option of generating only a part of the polyhedric configuration by specifying the required face numbers through the face list. The face numbers for the Platonic and Archimedean polyhedra are given in Plates 4.1-4.5 and 4.6-4.18, respectively.

4.2.1 POLYHEDRON CODE

The polyhedron code $P$ is a nonzero integer defining the type of polyhedron. The value of the polyhedron code $P$ may be from 1 to 18 as shown in Table 4.2. In this table the right handed versions of the snub cube and snub dodecahedron have been assigned positive values whereas the left handed versions have been assigned negative values.

4.2.2 SYMMETRY OPERATIONS

This section is devoted to the description of the process which is used for the creation of the tetrahedral configuration in Section 4.2. This process is based on the formex functions which have been explained in Chapter 2.

To begin with consider the configuration shown in Fig 4.35. The polyhedron function for the creation of the tetrahedral configuration of edge length $L=24.5$ may be written as

$$D = \text{PEX}\{\text{POL}(1,15,[0,0;12,0])\}G$$

where $G$ is a formex variable representing the compret of the configuration of Fig
4.35 and where formex variable D represents the entire tetrahedral configuration of Fig 4.36. The edge length \( L = 24.5 \) is obtained from

\[
R = \sqrt{16/12} \cdot L \quad \text{or} \quad L = 24.5
\]

which is given in Plate 4.1 in Section 4.1.

Plate 4.1 in Section 4.1 includes information about the tetrahedron's face numbers. In the plate the faces that are numbered as 1, 2 and 3 constitute the top part of the tetrahedron and are referred to as the "non x-y faces". A non x-y face is a face which is not parallel to the x-y plane. The fourth face is at the bottom of the tetrahedron and below the origin of the global coordinate system of Fig 4.36. This fourth face is referred to as the "bottom x-y face". The bottom x-y face is the face which is parallel to the x-y plane and has the minimum z coordinate. It should be noted that the U1-U2-U3 axes are assumed, at this stage, to coincide with the x-y-z axes of the global Cartesian coordinate system.

The coordinates of the nodal points on each one of the first three faces of the tetrahedral configuration are obtained in three stages. Firstly, a formex variable \( P \) is obtained from the formex variable \( G \) representing the compret of the configuration of Fig 4.35 through the following Formian statement

\[
P = \text{VERAD}(0,0) \mid \text{DILID}(24.5/12,24.5/12) \mid \text{TRAN}(1,-6) \mid G
\]

where the translation of formex variable \( G \) places point \( K \) of the configuration, that is the midpoint of line AB, at the origin of the global Cartesian coordinate system, Fig 4.37. The next formex function in the above formulation, that is,

\[
\text{DILID}(24.5/12,24.5/12)
\]

scales the configuration by the dilatation factor 24.5/12, to obtain the required edge
length of the tetrahedral configuration, Fig 4.38.

Finally, the formex function

\[ \text{VERAD}(0,0) \]

in the above formulation places the line AB of the configuration on the y axis as shown in Fig 4.39. The resulting configuration represented by formex variable P, is referred to as the "prime configuration".

In the second stage, formex variable P is transformed into a formex variable Q through the following statement

\[ Q = \text{TRAN}(1,12.25\ast\tan30) \mid \text{VERAS}(0,0,-\arccos\left(\frac{1}{3}\right)) \mid P \]

where the formex function

\[ \text{VERAS}(0,0,-\arccos\left(\frac{1}{3}\right)) \]

rotates the prime configuration by the appropriate angle to obtain a non x-y face of the tetrahedral configuration, Fig 4.40. This angle is equal to the dihedral angle of the tetrahedron and is as given in Plate 4.1. This rotation is followed by a translation along the positive x axis which is given by the function

\[ \text{TRAN}(1,12.25\ast\tan30) \]

This translation places the configuration at the position shown in Figs 4.41 and 4.42.

Finally, the formex variable Q is transformed into a formex variable S through the following statement
\[ S = \text{ROSAD}(0,0,3,120) \mid \text{TRAN}(3,-12.25*\sqrt{16/12}) \mid Q \]

where formex variable \( S \) represents the three non x-y faces of the tetrahedral configuration in their correct position. The formex function

\[ \text{TRAN}(3,-12.25*\sqrt{16/12}) \]

in the above formulation translates the configuration represented by formex variable \( Q \) along the z axis by \(-12.25*\sqrt{16/12}\) units. This translation places the configuration at its correct position to form the first non x-y face of the tetrahedral configuration, Fig 4.43. Finally, the formex function

\[ \text{ROSAD}(0,0,3,120) \]

rotates the first non x-y face about the origin \((0,0)\) by an angle of 120 degrees to create the three non x-y faces of the tetrahedral configuration, Figs 4.44 and 4.45. The canonic parameters 3 in the above ROSAD function results in the creation of three faces and the canonic parameter 120 gives the angle of rotation.

Once the three non x-y faces of the tetrahedral configuration of Fig 4.36 are created the next step is to create the fourth face. The coordinates of the nodal points of the fourth face of the tetrahedral configuration of Fig 4.36 is obtained in three stages. In the first stage the formex variable \( P \) is obtained as explained above, that is, through the statement

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(24.5/12,24.5/12) \mid \text{TRAN}(1,-6) \mid G \]

where a plot of \( P \) is shown in Fig 4.39.

In the second stage the formex variable \( P \) is transformed into a formex variable \( Q \) through the following statement
\[ Q = \text{REF}(3,0) \mid \text{VERAS}(0,0, -90) \mid P \]

where the formex function

\[ \text{VERAS}(0,0, -90) \]

rotates the configuration by 90 degrees and places it perpendicular to the x axis, Fig 4.46. The function

\[ \text{REF}(3,0) \]

reflects the configuration with respect to the x-y plane and places it along the negative z axis, Figs 4.47 and 4.48.

Finally, the formex variable Q is transformed into a formex variable S through the following statement

\[ S = \text{TRANIS}(12.25\times\text{TAN} \mid 30, -12.25\times\text{SQRT} \mid 6/12) \mid \text{VERAS}(0,0, -90) \mid Q \]

where the formex function

\[ \text{VERAS}(0,0, -90) \]

rotates the configuration represented by Q by 90 degrees and places it parallel to the x-y plane, Figs 4.49-4.51. The subsequent translations along the x and z axes place the configuration at its correct position in the global Cartesian coordinate system for the bottom x-y face to form the tetrahedral configuration of Fig 4.36. These translations are given by the formex function

\[ \text{TRANIS}(12.25\times\text{TAN} \mid 30, -12.25\times\text{SQRT} \mid 6/12) \]
where the amounts of translations are given by $12.25 \times \tan(30\degree)$ and $-12.25 \times \sqrt{16/12}$ in the first and third directions, respectively, Figs 4.52-4.54. A graphical representation of the tetrahedral configuration is as shown in Fig 4.36. Plate 4.19 summarises the process which has been followed for the creation of the triangular faces of the tetrahedral configuration of Fig 4.36.

It should also be mentioned that in the formulation of formex variable D the PEX function has been used to remove all the doubly represented elements along the edges of the tetrahedral configuration, Fig 4.55.

Procedures similar to the ones which are described above are followed for the creation of all the Platonic polyhedra. Each one of the procedures consists of a series of Formian statements. Each of the Formian statements involves a number of formex functions. These formex functions are used to imply dilatation, translation, rotation or reflection of a given configuration. The values of the canonic parameters which are involved in the formex function are specified in relation to the x-y-z global Cartesian coordinate system, which is given in each one of the Plates 4.1-4.5. The procedure which has been followed for the creation of the three non x-y faces of the tetrahedral configuration is referred to as "Procedure 2.1" and its general form is given in the sequel. Also, the procedure which has been followed for the creation of the bottom face of the tetrahedral configuration is referred to as "Procedure 4" and its general form is also given in the sequel. However, for the creation of the Platonic polyhedra some other procedures are needed and their general description are also given below.

4.2.3 BASIC PROCEDURES FOR THE CREATION OF THE PLATONIC POLYHEDRA

For the creation of the Platonic polyhedra some basic procedures have been followed throughout this study. These are for the creation of a polyhedric configuration and
Table 4.2

<table>
<thead>
<tr>
<th>Polyhedron</th>
<th>Code</th>
<th>Polyhedron</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tetrahedron</td>
<td>1</td>
<td>Great Rhombicuboctahedron</td>
<td>11</td>
</tr>
<tr>
<td>Cube (Ihexahedron)</td>
<td>2</td>
<td>Icosidodecahedron</td>
<td>12</td>
</tr>
<tr>
<td>Octahedron</td>
<td>3</td>
<td>Truncated Dodecahedron</td>
<td>13</td>
</tr>
<tr>
<td>Dodecahedron</td>
<td>4</td>
<td>Truncated Icosahedron</td>
<td>14</td>
</tr>
<tr>
<td>Icosahedron</td>
<td>5</td>
<td>Snub Cube (dextro)</td>
<td>15</td>
</tr>
<tr>
<td>Truncated Tetrahedron</td>
<td>6</td>
<td>Snub Cube (laevo)</td>
<td>-15</td>
</tr>
<tr>
<td>Cuboctahedron</td>
<td>7</td>
<td>Small Rhombicosidodecahedron</td>
<td>16</td>
</tr>
<tr>
<td>Truncated Cube</td>
<td>8</td>
<td>Great Rhombicosidodecahedron</td>
<td>17</td>
</tr>
<tr>
<td>Truncated Octahedron</td>
<td>9</td>
<td>Snub Dodecahedron (dextro)</td>
<td>18</td>
</tr>
<tr>
<td>Small Rhombicuboctahedron</td>
<td>10</td>
<td>Snub Dodecahedron (laevo)</td>
<td>-18</td>
</tr>
</tbody>
</table>

Fig 4.55

Table 4.3

<table>
<thead>
<tr>
<th>Polyhedron</th>
<th>Procedure 1</th>
<th>Non x-y Faces</th>
<th>Bottom Face</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top Face</td>
<td>Top Part</td>
<td>Bottom Part</td>
</tr>
<tr>
<td></td>
<td>Procedure 2.1</td>
<td>Procedure 2.2</td>
<td>Procedure 3.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Polyhedron</th>
<th>Procedure 1</th>
<th>Non x-y Faces</th>
<th>Bottom Face</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top Face</td>
<td>Top Part</td>
<td>Bottom Part</td>
</tr>
<tr>
<td></td>
<td>Procedure 2.1</td>
<td>Procedure 2.2</td>
<td>Procedure 3.1</td>
</tr>
</tbody>
</table>

- T Triangle
- S Square
- P Pentagon

349
depend on the type of face to be created, that is, top x-y face, non x-y face and bottom x-y face. Six different procedures are followed in the creation of the three types of faces of the polyhedral configurations that are based on the Platonic polyhedra. Procedure 1 deals with the creation of the top x-y face of the polyhedral configuration. Procedures 2.1 and 2.2 deal with the creation of the non x-y faces of the top part of the polyhedral configuration. Procedures 3.1 and 3.2 deal with the creation of the non x-y faces of the bottom part of the polyhedral configuration and Procedure 4 deals with the creation of the bottom x-y face of the polyhedral configuration. Each one of the procedures consists of a series of Formian statements. Each of the Formian statements involves a number of formex functions. These formex functions are used to imply dilatation, translation, rotation or reflection of a given configuration.

Top x-y Face

The top x-y face is the face which is parallel to the x-y plane and has the maximum z coordinate. Of the Platonic polyhedra only the cube and the dodecahedron have a top x-y face (See Plates 4.2 and 4.4). Procedure 1 is adopted for the creation of the top x-y face of a polyhedral configuration.

Procedure 1

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx1,-Ty) \mid G \]

\[ Q = \text{TRANIS}(Tx2,Tz) \mid P \]

where the terms which are used in the above formulations are as follows:

- **G**: formex variable representing the compret of a given configuration
- **L**: edge length of the polyhedral configuration
- **Tx1, Tx2**: amount of translation along the x axis
- **Ty**: amount of translation along the y axis
- **Tz**: amount of translation along the z axis
Non x-y Faces, Top Part

A non x-y face is a face which is not parallel to the x-y plane. The top part of a polyhedric configuration is the part which is above the x-y plane. All the Platonic polyhedra have non x-y faces (See Plates 4.1-4.5). Procedures 2.1 and 2.2 are used for the creation of the non x-y faces of the top part of a Platonic polyhedron. Let the line AB represents an edge of the configuration that is to be mapped on a non x-y face of the top part. Procedure 2.1 is used for the creation of a non x-y face when the line AB is parallel to the y axis and lies on a plane parallel to the x-y plane. Procedure 2.2 is used for the creation of a non x-y face when the configuration which is mapped onto a face of the polyhedric form is to be reflected.

Procedure 2.1

\[
P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx_1,-Ty) \mid G
\]

\[
Q = \text{TRAN}(1,Tx_2) \mid \text{VERAS}(0,0,-Ry) \mid P
\]

\[
S = \text{ROSAD}(0,0,F,Rz) \mid \text{TRAN}(3,-Tz) \mid Q
\]

where the terms which are used in the above formulations are as follows:

- \(G\) formex variable representing the compret of a given configuration
- \(L\) edge length of the polyhedric configuration
- \(Tx_1, Tx_2\) amount of translation along the x axis
- \(Tz\) amount of translation along the z axis
- \(Ry\) angle of rotation about the y axis in degrees
- \(Rz\) angle of rotation about the z axis in degrees
- \(F\) number of faces to be created

Procedure 2.2

\[
P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx_1,-Ty) \mid G
\]

351
The terms which are given in the above formulations should be interpreted in a way similar to the ones given for Procedure 2.1. Procedure 2.2 is similar to Procedure 2.1 except that in addition to the other stages it involves a reflection with respect to the x-y plane.

Non x-y Faces, Bottom Part

Procedure 3.1
Some of the Platonic polyhedra have the same arrangement of non x-y faces in the top part as in the bottom part. In this case, the entire top part of the corresponding polyhedral configuration is reflected with respect to the x-y plane to obtain the bottom part of that polyhedral configuration. A Formian statement describing this operation may be given as

\[
BTM = \text{REF}(3,0) \mid \text{TOP}
\]

where formex variable TOP represents the top part of the polyhedral configuration and the formex variable BTM represents the bottom part.

Procedure 3.2
Two stages are identified in this procedure. The first stage is the same as Procedure 3.1. In the second stage, the reflection of the top part of the polyhedral configuration is followed by an anticlockwise rotation around the z axis by an appropriate angle to obtain the bottom part of the polyhedral configuration. A Formian statement describing Procedure 3.2 may be given as

\[
Q = \text{REF}(3,0) \mid \text{VERAS}(0,0,-Ry) \mid P
\]

\[
S = \text{ROSAD}(0,0,F,Rz) \mid \text{TRANIS}(Tx2,-Tz) \mid Q
\]
BTM = VERAD(0,0,Rz) | REF(3,0) | TOP

where the canonic parameter Rz defines the angle of rotation about the z axis.

**Bottom x-y Face**

The bottom x-y face is the face which is parallel to the x-y plane and has the minimum z coordinate. All the Platonic polyhedra except the octahedron and the icosahedron have a bottom x-y face (See Plates 4.1, 4.2 and 4.4). Procedures 4.1 and 4.2 are used for the creation of the bottom x-y face of a Platonic polyhedral configuration. Let the line AB represents an edge of a polyhedral configuration. Procedure 4.1 is used for the creation of the bottom x-y face when the line AB is parallel to the y axis and Procedure 4.2 is used when line AB is not parallel to the y axis.

**Procedure 4.1**

\[ P = \text{VERAD}(0,0) | \text{DILID}(L,L) | \text{TRANID}(-T_{x1},-T_{y}) | G \]

\[ Q = \text{REF}(3,0) | \text{VERAS}(0,0,-90) | P \]

\[ S = \text{TRANIS}(T_{x2},-T_{z}) | \text{VERAS}(0,0,-90) | Q \]

The terms which are used in the above formulations are as follows:

- **G**: formex variable representing the compret of a given configuration
- **L**: edge length of the polyhedral configuration
- **T_{x1}, T_{x2}**: amount of translation along the x axis
- **T_{y}**: amount of translation along the y axis
- **T_{z}**: amount of translation along the z axis
- **Rz**: angle of rotation about the z axis in degrees
Procedure 4.2

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx1,-Ty) \mid G \]

\[ Q = \text{REF}(3,0) \mid \text{VERAS}(0,0,-90) \mid P \]

\[ S = \text{TRANIS}(Tx2,-Tz) \mid \text{VERAS}(0,0,-90) \mid Q \]

\[ W = \text{VERAD}(0,0,Rz) \mid S \]

The terms which are given in the above formulations should be interpreted in a way similar to the ones given for Procedure 4.1. Procedure 4.2 is similar to Procedure 4.1 except that in addition to the other stages it involves a rotation around the z axis and is only applied when the bottom x-y pentagonal face of a dodecahedral configuration is to be created.

Table 4.3 summarises the process for the creation of each one of the Platonic polyhedra. The first column of Table 4.3 lists the Platonic polyhedra. The second column corresponds to Procedure I and deals with the creation of the top x-y face of a polyhedric configuration. Columns 3 to 6 deal with the creation of the non x-y faces of a polyhedric configuration. In particular, columns 3 and 4 deal with the non x-y faces that constitute the top part of a polyhedric configuration and correspond to Procedures 2.1 and 2.2, respectively. Columns 5 and 6 deal with the creation of the non x-y faces that constitute the bottom part of the polyhedric configuration and correspond to Procedures 3.1 and 3.2, respectively. The last two columns of Table 4.3 correspond to Procedures 4.1 and 4.2 and deal with the bottom x-y face of a polyhedric configuration. The number and the type of faces that are created in each of the procedures are given at the top and bottom of a cross, respectively.

Table 4.3 may be interpreted as follows: Let it be required to generate a polyhedric configuration based on a tetrahedron. For the creation of the three non x-y triangular faces Procedure 2.1 is applied and for the creation of the bottom x-y triangular face
Procedure 4 is applied. As the next example, suppose that a polyhedric configuration based on a dodecahedron is to be generated. The dodecahedron has a top x-y face, a bottom x-y face and ten non x-y faces where five of them constitute the top part and the other five constitute the bottom part of the dodecahedral configuration. For the creation of the top x-y pentagonal face Procedure 1 is applied and for the creation of the five non x-y pentagonal faces that constitute the top part of the dodecahedral configuration Procedure 2.2 is applied. Furthermore, Procedures 3.2 and 4 are applied for the creation of the five non x-y pentagonal faces of the bottom part and the bottom x-y face of the dodecahedral configuration, respectively.

To further clarify the approach for the creation of each of the Platonic polyhedra a flow chart has been given for each one of them. The main body of the flow chart consists of five rhombic boxes, where each box represents a Platonic polyhedron. The numbers that are given on the side of each box correspond to the ones in Table 4.2. Smaller flow charts with rectangular boxes have been given for each polyhedron describing the order of the procedures that have to be followed for the creation of a polyhedric configuration. For instance, suppose that is required to generate an icosahedron. It may be seen from Plate 4.5 that an icosahedron has twenty triangular non x-y faces and represents the same arrangement of faces in the top part as in the bottom part. For the creation of the top part of the icosahedral configuration Procedures 2.1 and 2.2 are applied. In particular, Procedure 2.1 is applied for the creation of the five triangular non x-y faces of the top part (5 T Faces) and Procedure 2.2 is applied for the creation of the other five non x-y reflected triangular faces (5 T Faces). Finally, Procedure 3.2 is applied for the creation of the ten triangular faces (10 T Faces) that constitute the bottom part of the icosahedral configuration. The letters T, S, or P which are given at the bottom of the rectangular boxes in the smaller flow charts represent a triangular, a square or a pentagonal polyhedral face, respectively.

As the next example, consider the configuration shown in Fig 4.56. This a view of a polyhedric configuration obtained by mapping the configuration of Fig 4.57 onto
1
Procedure 2.1
3 T Faces

Procedure 4.1
1 T Faces

END

2
Procedure 1
1 S Face

Procedure 2.1
4 S Faces

Procedure 4.1
1 S Face

END

configuration

Tetrahedron

YES → 1

NO

Cube

YES → 2

NO

Octahedron

YES → 3

NO

Dodecahedron

YES → 4

NO

Icosahedron

YES → 5

NO

ARCHIMEDEAN POLYHEDRA
three faces of a cube. The formulation of formex variable \( C_1 \) that represents the polyhedric form of Fig 4.56 may be given as

\[
C_1 = \text{POL}(2,10,[0,0;12,0],[1;3]) | P_1
\]

where the argument \( P_1 \) represents the configuration of Fig 4.57 and may be given as

\[
P_1 = \text{RINID}(6,6,2,2) | \text{ROSAD}(1,1) | [0,0;1,1]
\]

The new points to be noticed in this example are:

i) The type of polyhedron may be changed. A cube has been used as the base polyhedron (in Fig 4.56) instead of the tetrahedron of Fig 4.36.

ii) The part of the polyhedron on which the configuration mapped may be varied. The first, second and third face of the cube have been used to obtain the polyhedric form of Fig 4.56 whereas the entire tetrahedron has been create in Fig 4.36. Therefore, the face list has been used to specify the faces 1 to 3 of the cube onto which the configuration is to be mapped. The face numbers of the Platonic polyhedra are given in Plates 4.1-4.5.

In creating a polyhedric form, it is possible to use different patterns for different faces of the base polyhedron. For instance, in the case of the polyhedric configuration of Fig 4.58, the diagonal pattern of Fig 4.57 is place onto the second and third faces of a cube and the pattern of Fig 4.59 is placed onto the first face. The formulation of formex variable \( C_2 \) that represents the polyhedric form of Fig 4.58 is given as

\[
C_2 = \text{POL}(2,10,[0,0;12,0],[2;3]) | P_1 | \text{POL}(2,10,[0,0;12,0],1) | P_2
\]

where the argument \( P_2 \) represents the configuration of Fig 4.59 and may be given as
An important point to be noticed here is that the pattern of Fig 4.59 consists of "tile" elements whereas the pattern of Fig 4.57 consists of "line" elements. The difference between the element types is implied through the formex formulations. Namely, in the formulation of P1 an element is specified by the coordinates of its two end points, whereas in the formulation of P2 an element is specified by the coordinates of its four corner points. In general, there are no restrictions regarding the types of elements that may be involved in configurations for creation of polyhedric forms.

A configuration that is mapped onto a face of a polyhedron need not necessarily "fill" the face or "match" the boundaries of the face, as demonstrated by the polyhedric configuration shown in Fig 4.60. Here, the configuration of Fig 4.61 which has a rectangular shape of boundary is mapped onto the first face of the cube and the configuration of Fig 4.62 which has a triangular shape of boundary is mapped onto the second and third faces of the cube. The configuration that is placed onto the first face extends beyond the edges of the face and the configuration that is placed onto the second and third faces cover the faces only partially. The formex formulation of formex variable C1 that represents the polyhedric form of Fig 4.60 may be given as

\[ C1 = \text{POL}(2,10,[0,0;12,0],1) \# \text{POL}(2,10,[0,0;12,0],[2;3]) \# P3 \]

where the argument P3 represents the configuration of Fig 4.61 and may be given as

\[ P3 = \text{RINID}(8,6,2,2)[0,0;2,0;2,2;0,2] \]

and the argument P4 represents the configuration of Fig 4.62 and may be given as

\[ P4 = \text{JUN}([1,2,1],[2,13;6,5;14,13]) \# P1 \]

So far, the polyhedron function has been used to generate single layer polyhedric
configurations. However, the polyhedron function can be used to generate multi-layer polyhedric configurations. For example, consider the configuration shown in Fig 4.63. This is a view of a double layer polyhedric configuration which is based on the top five faces of two concentric icosahedra. The double layer configuration used for mapping is shown in Fig 4.64 together with the normat U1-U2-U3 for the formex formulation. A formex formulation for this double layer triangular configuration may be given as

\[ T = \text{GENID}(5, 5, 2, \text{SQRT}13, 1, -I) \text{IROSAD}(1, \text{SQRT}3/3, 3, 120) \]

\[ \{[0,0,2;2,0,2],[0,0,2;1,\text{SQRT}13/3,1]\} \]

\[ B = \text{GENID}(4, 4, 2, \text{SQRT}13, 1, -I) \text{IROSAD}(2, 2*\text{SQRT}13/3, 3, 120) \]

\[ [1,\text{SQRT}3/3,1;3,\text{SQRT}3/3,1] \]

In Figs 4.63 and 4.64 the top layer elements are drawn in thick lines and the bottom layer elements as well as the web elements are drawn in thin lines. The formulation of formex variable C4 that represents the polyhedric form of Fig 4.63 may be given as

\[ C4 = \text{PEX} \text{POL}(5,[10;12],[[1,\text{SQRT}3/3;9,\text{SQRT}3/3],[0,0;10,0]],[1;5]) \text{(T#B)} \]

The new points to be noticed in this example are:

i) In formulating the configuration of Fig 4.64 each layer has an identification number. The identification number of the bottom layer is 1 and that of the top layer is 2. These identification numbers appear as "third direction coordinates" in the formex formulations.

ii) The radius specifier in formex variable C4 contains two entries. The first entry,
Fig 4.64

Fig 4.65

Fig 4.66

Fig 4.67
that is, 10 gives the radius of the circumsphere of the base polyhedron for the first layer (bottom layer) and the second entry, that is, 12 gives the radius of the circumsphere of the base polyhedron for the second layer (top layer).

iii) The locator in C4 contains two parts. The first part of the locator specifies the coordinates of points A1 and B1 (in Fig 4.64) that should fit the end vertices of an edge of the base polyhedron of the first layer (bottom layer). The second part of the locator specifies the coordinates of points A2 and B2 (in Fig 4.64) that should fit the end vertices of an edge of the base polyhedron of the second layer (top layer). Situations when polyhedric configurations involve more than two layers are dealt with in an analogous manner. In such case, the general form of the polyhedron function may be written as

\[
\text{POL}(T, [R_1; R_2; \ldots; R_n], [(A_1; B_1), (A_2; B_2), \ldots, (A_n; B_n)] < < \{F_1, F_2, \ldots, F_n\} > )
\]

where the construct \([R_1; R_2; \ldots; R_n]\) lists the radii of the circumspheres of the base polyhedra for the \(n\) layers. The locator contains \(n\) cantles where each one of the cantles specifies the coordinates of points A and B that should fit the end vertices of an edge of the base polyhedron for each one of the \(n\) layers.

So far, polyhedric configurations based on the Platonic polyhedra have been generated. However, the polyhedron function can be used to create polyhedric configurations based on the Archimedean polyhedra. In the case of Archimedean polyhedra the faces are not all of the same shape. In spite of this, one may create a polyhedric form by mapping a single configuration on all or some of the faces of an Archimedean polyhedron. However, in creating polyhedric forms that are based on Archimedean polyhedra, normally one would use different configurations for different face shapes. For instance, the polyhedric configuration of Fig 4.65 is based on a cuboctahedron and is obtained by mapping the configurations of Figs 4.66 and 4.67 onto the triangular and square faces, respectively.

362
A Formian statement describing this operation may be given as

\[ C5 = \text{PEX}(\text{POL}(7,10,[0,0;12,0],[1;8]) \# \text{POL}(7,10,[0,0;12,0],[9;14])) \# \text{P6} \]

where P5 is a formex variable representing the compret of the configuration of Fig 4.66 and may be given as

\[ P5 = \text{GENID}(6,6,2,\sqrt{3},1,-1) \# \text{ROSAD}(1,\sqrt{3}/3,3,120)[0,0;2,0] \]

and P6 is a formex variable representing the compret of the configuration of Fig 4.67 and may be given as

\[ P6 = \text{LUX}([6,6]) \# \text{ROSAD}(6,6) \# \text{BB}(1,1/\sqrt{3}) \# \text{P5} \]

The process which has been followed for the creation of the cuboctahedral configuration is similar to the one followed for the creation of the tetrahedral configuration in Section 4.2.2. The cuboctahedron has a top x-y square face, eight triangular and four square non x-y faces and a bottom x-y square face (See Plate 4.7).

For the creation of the top and bottom face Procedures 1 and 4 are adopted, respectively (See Table 4.4 and flow chart number 7). To create the non x-y faces three different procedures are adopted, namely, Procedure 2.4, Procedure 3.1 and Procedure 5.1 (See Table 4.4 and flow chart number 7). Procedures 2.4 and 3.1 are used for the creation of the eight non x-y triangular faces and Procedure 5.1 for the creation of the four non x-y square faces. To begin with, the triangular faces are created and the square faces are created later. This convention for creating the faces with the least number of edges first and the faces with the maximum number of edges last will be followed henceforth.

The coordinates of the nodal points on each one of the first four triangular non x-y faces of the cuboctahedral configuration are obtained in four stages. Firstly, a formex
<table>
<thead>
<tr>
<th>Polyhedron</th>
<th>Truncated Tetrahedron</th>
<th>Cuboctahedron</th>
<th>Truncated Cube</th>
<th>Octahedron</th>
<th>Small Rhombicuboctahedron</th>
<th>Great Rhombicuboctahedron</th>
<th>Icosidodecahedron</th>
<th>Truncated Dodecahedron</th>
<th>Icosahedron</th>
<th>Snub Cube</th>
<th>Small Rhombicicosidodecahedron</th>
<th>Great Rhombicicosidodecahedron</th>
<th>Snub Dodecahedron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top x' y' z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top x y' z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top x z' y'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top y x' z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top y z' x'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z x' y'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top y x z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z y x'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z y z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z z x'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top x y y'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top x y z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z y y'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z y z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Top z z z'</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 4.4

1. Triangle
2. Square
3. Pentagon
4. Hexagon
5. Octagon
6. Decagon
ARCHIMEDEAN POLYHEDRA 16-18
variable $P$ is obtained from the formex variable $P_5$ representing the compret of the configuration of Fig 4.66 through the following Formian statement

$$P = \text{VERAD}(0,0) \mid \text{DILID}(10/12,10/12) \mid \text{TRAN}(1,-6) \mid P_5$$

where the translation of formex variable $P_5$ places point $K$ of the configuration, that is the midpoint of line $AB$, at the origin of the global Cartesian coordinate system, Fig 4.68. The next formex function in the above formulation, that is,

$$\text{DILID}(10/12,10/12)$$

scales the configuration by the dilatation factor $10/12$, to obtain the required edge length of the cuboctahedral configuration, that is $10$, Fig 4.69.

Finally, the formex function

$$\text{VERAD}(0,0)$$

in the above formulation places the line $AB$ of the configuration on the $y$ axis as shown in Figs 4.70 and 4.71. The resulting configuration represented by formex variable $P$, is referred to as the "prime configuration" of the triangular faces.

In the second stage, formex variable $P$ is transformed into a formex variable $Q$ through the following statement

$$Q = \text{REF}(3,0) \mid \text{VERAS}(0,0,-\text{ACOS}\left(-\sqrt{13/3}\right)) \mid P$$

where the formex function

$$\text{VERAS}(0,0,-\text{ACOS}\left(-\sqrt{3/3}\right))$$
rotates the prime configuration to obtain a non x-y face of the cuboctahedral configuration, Figs 4.72 and 4.73. The angle of rotation is equal to the dihedral angle of the cuboctahedron and is as given in Plate 4.7. This rotation is followed by a reflection, with respect to the x-y plane, which is given by the formex function

$$\text{REF}(3,0)$$

This reflection places the configuration at the position shown in Figs 4.74.

In the third stage, the formex variable Q is transformed into a formex variable S through the following statement

$$S = \text{VERAD}(0, 0, 45) \mid \text{TRANIS}(5, 10 \times 0.7071) \mid Q$$

where the formex function

$$\text{TRANIS}(5, 10 \times 0.7071)$$

translates the configuration represented by formex variable Q along the x and z axes by 5 and 10*0.7071 units, respectively, Figs 4.75-4.77. The formex function

$$\text{VERAD}(0, 0, 45)$$

rotates the translated configuration around the z axis by 45 degrees. This rotation places the configuration at its correct position to form the first non x-y triangular face of the cuboctahedral configuration, Figs 4.78 and 4.79.

Finally, the formex variable S is transformed into a formex variable W through the following statement

$$W = \text{ROSAD}(0, 0) \mid S$$
where formex variable \( W \) represents the first four non \( x-y \) triangular faces in their correct position. The formex function

\[
\text{ROSAD}(0,0)
\]

in the above formulation rotates the configuration represented by formex variable \( S \) about the origin \((0,0)\) to create the four non \( x-y \) triangular faces that constitute the top part of the cuboctahedral configuration, Figs 4.80-4.83.

Once the first four non \( x-y \) triangular faces of the cuboctahedral configuration of Fig 4.65 are created the next step is to create the four non \( x-y \) triangular faces that constitute the bottom part of the cuboctahedral configuration. Procedure 3.1 is used for this purpose. A Formian statement describing procedure 3.1 may be given as

\[
\text{BTM} = \text{REF}(3,0) \mid W
\]

where formex variable \( W \) represents the top part of the cuboctahedral configuration and formex variable \( \text{BTM} \) represents the bottom part of the cuboctahedral configuration in its correct position, Figs 4.84 and 4.85.

So far, all the triangular faces of the cuboctahedral configuration have been created. The next step is to create the remaining six square faces. To begin with, Procedure 1 is adopted to create the top \( x-y \) square face. The coordinates of the nodal points of the top face are created in three stages. Firstly, a formex variable \( P \) is obtained from the formex variable \( \text{P6} \) representing the compret of the configuration of Fig 4.67 through the following Formian statement

\[
\text{P} = \text{VERAD}(0,0) \mid \text{DILID}(10/12,10/12) \mid \text{TRAN}(1,-6) \mid \text{P6}
\]

where the translation of formex variable \( \text{P6} \) places point \( K \) of the configuration, that is the midpoint of line \( AB \), at the origin of the global Cartesian coordinate system,
Fig 4.86. The next formex function in the above formulation, that is,

\[ \text{DILID}(10/12,10/12) \]

scales the configuration by the dilatation factor 10/12, to obtain the required edge length of the cuboctahedral configuration, that is 10, Fig 4.87.

Finally, the formex function

\[ \text{VERAD}(0,0) \]

in the above formulation places the line AB of the configuration on the y axis as shown in Fig 4.88. The resulting configuration represented by formex variable P, is referred to as the "prime configuration" for the square faces.

In the second stage the formex variable P is transformed into a formex variable Q through the following statement

\[ Q = \text{TRANIS}(5,10*0.7071) \mid P \]

where the formex function

\[ \text{TRANIS}(5,10*0.7071) \]

translates the configuration by 5 and 10*0.7071 units along the x and z directions, respectively, Figs 4.89-4.90.

Finally, the formex variable Q is transformed into a formex variable S through the following statement

\[ S = \text{VERAD}(0,0,45) \mid Q \]
where formex variable $S$ represents the top $x$-$y$ square face of the cuboctahedral configuration, Fig 4.91.

For the creation of the four non $x$-$y$ square faces of the cuboctahedral configuration Procedure 5.1 is used. The coordinates of the nodal points on each one of the four non $x$-$y$ square faces of the cuboctahedral configuration are obtained in three stages. Firstly, a formex variable $P$ is obtained from the formex variable $P_6$ representing the compret of the configuration of Fig 4.67, through the Formian statement

$$P = \text{VERAD}(0,0) \mid \text{DILID}(10/12,10/12) \mid \text{TRAN}(1,-6) \mid P_6$$

where the plot of $P$ is shown in Fig 4.88.

In the second stage the formex variable $P$ is transformed into a formex variable through the following statement

$$Q = \text{TRANIS}(10\ast0.7071,-5) \mid \text{VERAS}(0,0,-90) \mid P$$

where the formex function

$$\text{VERAS}(0,0,-90)$$

rotates the prime configuration to obtain a non $x$-$y$ square face of the cuboctahedral configuration, Fig 4.92. The formex function

$$\text{TRANIS}(10\ast0.7071,-5)$$

translates the configuration along the $x$ and $z$ axes by $10\ast0.7071$ and 5 units, respectively, Figs 4.93 and 4.94.

Finally, the formex variable $Q$ is transformed into a formex variable $S$ through the
following statement

\[ S = \text{ROSAD}(0,0) \mid \text{VERAT}(0,0,45) \mid Q \]

where formex variable \( S \) represents the first four non x-y square faces in their correct positions. The formex function

\[ \text{VERAT}(0,0,45) \]

rotates the configuration by 45 degrees and places it at the position shown in Fig 4.95. The formex function

\[ \text{ROSAD}(0,0) \]

in the above formulation rotates the configuration represented by formex variable \( S \) about the origin \((0,0)\) to create the four non x-y square faces that constitute the middle part of the cuboctahedral configuration, Figs 4.96 and 4.97.

The coordinates of the nodal points of the bottom x-y face of the cuboctahedral configuration of Fig 4.65 is obtained in four stages. In the first stage the formex variable \( P \) is obtained as explained above, that is, through the statement

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(10/12,10/12) \mid \text{TRAN}(1,-6) \mid P6 \]

where a plot of \( P \) is shown in Fig 4.88.

In the second stage the formex variable \( P \) is transformed into a formex variable \( Q \) through the following statement

\[ Q = \text{REF}(3,0) \mid \text{VERAS}(0,0,-90) \mid P \]
where the formex function

\[ \text{VERAS}(0,0,-90) \]

rotates the configuration by 90 degrees and places it perpendicular to the x axis, Fig 4.98. The function

\[ \text{REF}(3,0) \]

reflects the configuration with respect to the x-y plane and places it along the negative z axis, Fig 4.99.

In the third stage, the formex variable Q is transformed into a formex variable S through the following statement

\[ S = \text{TRANIS}(5,-10\times0.7071) \mid \text{VERAS}(0,0,-90) \mid Q \]

where the formex function

\[ \text{VERAS}(0,0,-90) \]

rotates the configuration represented by Q by 90 degrees and places it parallel to the x-y plane, Fig 4.100. The formex function

\[ \text{TRANIS}(5,-10\times0.7071) \]

translates the configuration along the x and z axes by 5 and -10*0.7071 units, respectively, Figs 4.101 and 4.102.

Finally, the formex variable S is transformed into a formex variable W through the following statement
where the formex variable $W$ represents the bottom $x$-$y$ face of the cuboctahedral configuration, Fig 4.103. A graphical representation of the cuboctahedral configuration is as shown in Fig 4.65. Plates 4.20 and 4.21 summarize the process which has been followed for the creation of the triangular and square faces of the cuboctahedral configuration. In particular, Plate 4.20 deals with the creation of the triangular faces and Plate 4.21 deals with the creation of the square faces.

Procedures similar to the ones described above are used for the creation of all the Archimedean polyhedra. A general description of these procedures is given in the sequel.

4.2.4 BASIC PROCEDURES FOR THE CREATION OF THE ARCHIMEDEAN POLYHEDRA

For the creation of the Archimedean polyhedra some basic procedures have been followed throughout this study. These are for the creation of a polyhedric configuration and depend on the type of face to be created, that is, top $x$-$y$ face, non $x$-$y$ face or bottom $x$-$y$ face. Twelve different procedures are followed in the creation of the three types of faces for the polyhedric configurations that are based on the Archimedean polyhedra. Procedures 1.1 and 1.2 deal with the creation of the top $x$-$y$ face of the polyhedric configuration. Procedures 2.1 to 2.4 deal with the creation of the non $x$-$y$ faces of the top part of the polyhedric configuration. Procedures 3.1 and 3.2 deal with the creation of the bottom part of the polyhedric configuration and Procedures 4.1 and 4.2 deal with the creation of the bottom $x$-$y$ face of the polyhedric configuration. In order to create the non $x$-$y$ faces of the middle part of an Archimedean polyhedron Procedures 5.1 and 5.2 are used. These procedures are also employed in order to create the non $x$-$y$ faces of the top part of the snub cube and the snub dodecahedron. Each one of the procedures consists of a series of Formian
statements. Each of the Formian statements involves a number of formex functions. These formex functions are used to imply dilatation, translation, rotation or reflection of a given configuration.

Top x-y Face

The top x-y face is the face which is parallel to the x-y plane and has the maximum z coordinate. All the Archimedean polyhedra have a top x-y face (See Plates 4.6 to 4.18). Procedures 1.1 and 1.2 are used for the creation of the top x-y face of a polyhedric configuration that is based on an Archimedean polyhedron. Let the line AB represents an edge of the configuration that is to be mapped on the top x-y face of a polyhedric configuration. Procedure 1.1 is used for the creation of the top x-y face when the line AB is parallel to the y axis and Procedure 1.2 is used when the line AB is not parallel to the y axis.

Procedure 1.1

\[
P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx1,-Ty) \mid G
\]

\[
Q = \text{TRANIS}(Tx2,Tz) \mid P
\]

Procedure 1.2

\[
P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx1,-Ty) \mid G
\]

\[
Q = \text{TRANIS}(Tx2,Tz) \mid P
\]

\[
S = \text{VERAD}(0,0,Rz) \mid Q
\]

where the terms which are used in the above formulations are as follows:

- \( G \): formex variable representing the compret of a given configuration
- \( L \): edge length of the polyhedric configuration
**Tx1, Tx2** amount of translation along the x axis

**Ty** amount of translation along the y axis

**Tz1** amount of translation along the z axis

**Rz** angle of rotation about the z axis in degrees

Procedure 1.2 is similar to Procedure 1.1 except that it involves a rotation around the z axis.

**Non x-y Faces, Top Part**

A non x-y face is a face which is not parallel to the x-y plane. The top part of a polyhedral configuration is the part which is above the x-y plane. All the Archimedean polyhedra have non x-y faces (See Plates 4.6-4.18). Procedure 2.1 to 2.4 are used for the creation of the non x-y faces of the top part of an Archimedean polyhedron. Let the line AB represents an edge of the configuration that is to be mapped on a non x-y face of the top part of a polyhedral configuration. Procedures 2.1 and 2.2 are used for the creation of a non x-y face when the line AB is parallel to the y axis and lies on a plane parallel to the x-y plane. If the line AB lies on a plane parallel to the x-y plane but is not parallel to the y axis then Procedures 2.3 and 2.4 are used for the creation of a non x-y face of the top part. Furthermore, Procedure 2.2 and 2.4 are used for the creation of a non x-y face when the configuration which is mapped onto a face of the polyhedral form is to be reflected.

**Procedure 2.1**

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx1,-Ty) \mid G \]

\[ Q = \text{TRAN}(1,Tx2) \mid \text{VERAS}(0,0,-Ry) \mid P \]

\[ S = \text{ROSAD}(0,0,F,Rz) \mid \text{TRAN}(3,-Tz) \mid Q \]

The terms which are used in the above formulations are as follows:
G formex variable representing the compret of a given configuration
L edge length of the polyhedral configuration
Tx1, Tx2 amount of translation along the x axis
Tz amount of translation along the z axis
Ry angle of rotation about the y axis in degrees
Rz angle of rotation about the z axis in degrees
F number of faces to be created

Procedure 2.2
P = VERAD(0,0) | DILID(L,L) | TRANID(-Tx1,-Ty) | G
Q = REF(3,0) | VERAS(0,0,-Ry) | P
S = ROSAD(0,0,F,Rz) | TRANIS(Tx2,-Tz) | Q

The terms which are given in the above formulations should be interpreted in a way similar to the ones given for Procedure 2.1. Procedure 2.2 is similar to Procedure 2.1 except that in addition to the other stages it involves a reflection with respect to the x-y plane.

Procedure 2.3
P = VERAD(0,0) | DILID(L,L) | TRANID(-Tx1,-Ty) | G
Q = TRAN(1,Tx2) | VERAS(0,0,-Ry) | P
S = VERAD(0,0,Rz1) | TRAN(3,Tz) | Q
W = ROSAD(0,0,F,Rz2) | S

The terms which are used in the above formulations are as given before.
Procedure 2.4

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-Tx1,-Ty) \mid G \]

\[ Q = \text{REF}(3,0) \mid \text{VERAS}(0,0,-Ry) \mid P \]

\[ S = \text{VERAD}(0,0,Rz1) \mid \text{TRANIS}(Tx2,-Tz) \mid Q \]

\[ W = \text{ROSAD}(0,0,F,Rz2) \mid S \]

The terms which are used in the above formulations are as given for Procedure 2.1. Procedure 2.4 is similar to Procedure 2.3 except that in addition to the other stages it involves a reflection with respect to the x-y plane.

Non x-y Faces, Bottom Part

Procedure 3.1

Some of the Archimedean polyhedra have the same arrangement of non x-y faces in the top part as in the bottom part. In this case, the entire top part of the polyhedric configuration is reflected with respect to the x-y plane to obtain the bottom part. A Formian statement describing this operation may be given as

\[ \text{BTM}=\text{REF}(3,0)\mid \text{TOP} \]

where formex variable TOP represents the top part of the polyhedric configuration and the formex variable BTM represents the bottom part.

Procedure 3.2

Two stages are identified in this procedure. The first stage is the same as Procedure 3.1. In the second stage, the reflection of the top part of the polyhedric configuration is followed by an anticlockwise rotation around the z axis by an appropriate angle to
obtain the bottom part of the polyhedric configuration. A Formian statement describing this operation may be given as

\[
BTM = \text{VERAD}(O, O, Rz) \mid \text{REF}(3, 0) \mid \text{TOP}
\]

where the canonic parameter \( Rz \) defines the angle of rotation about the \( z \) axis.

**Bottom x-y Face**

The bottom x-y face is the face which is parallel to the x-y plane and has the minimum z coordinate. All the Archimedean polyhedra have a bottom x-y face (See Plates 4.6 to 4.18). Procedures 4.1 and 4.2 are used for the creation of the bottom x-y face of an Archimedean polyhedric configuration. Let the line AB represents an edge of the configuration that is to be mapped on the bottom x-y face of a polyhedric configuration. Procedure 4.1 is used for the creation of the bottom x-y face when the line AB is parallel to the y axis and Procedure 4.2 is used when line AB is not parallel to the y axis.

**Procedure 4.1**

\[
P = \text{VERAD}(0, 0) \mid \text{DILID}(L, L) \mid \text{TRANID}(-T_{x1}, -T_{y}) \mid G
\]

\[
Q = \text{REF}(3, 0) \mid \text{VERAS}(0, 0, -90) \mid P
\]

\[
S = \text{TRANIS}(T_{x2}, -T_{z}) \mid \text{VERAS}(0, 0, -90) \mid Q
\]

**Procedure 4.2**

\[
P = \text{VERAD}(0, 0) \mid \text{DILID}(L, L) \mid \text{TRANID}(-T_{x1}, -T_{y}) \mid G
\]

\[
Q = \text{REF}(3, 0) \mid \text{VERAS}(0, 0, -90) \mid P
\]

\[
S = \text{TRANIS}(T_{x2}, -T_{z}) \mid \text{VERAS}(0, 0, -90) \mid Q
\]
W = VERAD(0,0,Rz) | S

The terms which are used in the above formulations are as follows:

- **G**: formex variable representing the compact of a given configuration
- **L**: edge length of the polyhedral configuration
- **Tx1, Tx2**: amount of translation along the x axis
- **Ty**: amount of translation along the y axis
- **Tz**: amount of translation along the z axis
- **Rz**: angle of rotation about the z axis in degrees

**Non x-y Faces, Middle Part**

A non x-y face of an Archimedean polyhedral configuration may belong either to the top part or to the bottom part of the polyhedral configuration. However, in some cases a non x-y face does not belong completely to the top part or to the bottom part. In such a case this non x-y face may be regarded as belonging to the "middle part" of the Archimedean polyhedral configuration. The "middle part" of an Archimedean polyhedral configuration consists of those faces whose nodal points lie simultaneously above and below the x-y plane. Let the line AB represents an edge of the configuration that is to be mapped on a non x-y face of the middle part of the polyhedral configuration. Procedure 5.1 is used for the creation of the non x-y faces of the middle part of a polyhedral configuration when the line AB is not parallel to the y axis and does not lie on a plane parallel to the x-y plane. Procedure 5.2 is similar to Procedure 5.1 except that in addition to the other stages it involves a reflection with respect to the x-y plane. Procedure 5.2 is used when the configuration which is mapped onto a non x-y face of the middle part of a polyhedral form is to be reflected. Procedures 5.1 and 5.2 are also employed in order to create the non x-y faces of the top part of the snub cube and snub dodecahedron.
Procedure 5.1

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-T_{x1},-T_y) \mid G \]

\[ Q = \text{TRANIS}(T_{x2}, T_z) \mid \text{VERAS}(0,0,-R_y) \mid P \]

\[ S = \text{ROSAD}(0,0,F,R_{z2}) \mid \text{VERAD}(0,0,R_{z1}) \mid Q \]

The terms which are used in the above formulations are as follows:

- **G** formex variable representing the compret of a given configuration
- **L** edge length of the polyhedric configuration
- **T_{x1}, T_{x2}** amount of translation along the x axis
- **T_z** amount of translation along the z axis
- **R_y** angle of rotation about the y axis in degrees
- **R_{z1}, R_{z2}** angle of rotation about the z axis in degrees
- **F** number of faces to be created

Procedure 5.2

\[ P = \text{VERAD}(0,0) \mid \text{DILID}(L,L) \mid \text{TRANID}(-T_{x1},-T_y) \mid G \]

\[ Q = \text{REF}(3,0) \mid \text{VERAS}(0,0,-R_y) \mid P \]

\[ S = \text{VERAT}(0,0,R_{z1}) \mid \text{TRANIS}(T_{x2},-T_z) \mid Q \]

\[ W = \text{ROSAD}(0,0,F,R_{z2}) \mid S \]

The terms which are used in the above formulations are as given before.

Table 4.4 summarises the process for the creation of each one of the Archimedean polyhedra. The first column of Table 4.4 lists the Archimedean polyhedra. The second and third columns correspond to Procedures 1.1 and 1.2 and deal with the
creation of the top x-y face of a polyhedric configuration. Columns 4 to 9 deal with the creation of the non x-y faces of a polyhedric configuration. In particular, columns 4 to 7 deal with the non x-y faces that constitute the top part of a polyhedric configuration and correspond to Procedures 2.1, 2.2, 2.3 and 2.4, respectively. Columns 8 and 9 deal with the creation of the non x-y faces that constitute the bottom part of the polyhedric configuration and correspond to Procedures 3.1 and 3.2, respectively. Columns 10 and 11 deal with the creation of the bottom x-y faces of a polyhedric configuration and correspond to Procedures 4.1 and 4.2, respectively. Finally, the last two columns of Table 4.4 correspond to Procedures 5.1 and 5.2 and deal with the creation of the middle part of the Archimedean polyhedra and with the creation of the non x-y faces that constitute the top part of the snubs. The number and the type of faces that are created in each of the procedures are given at the top and bottom of a cross, respectively.

Table 4.4 may be interpreted as follows: Let it be required to generate a polyhedric configuration based on a cuboctahedron. For the creation of the eight non x-y triangular faces Procedures 2.4 and 3.1 are applied, respectively. For the creation of the top and bottom x-y square faces Procedures 1.2 and 4.2 are applied, respectively. Finally, in order to create the four non x-y square faces that constitute the middle part of the cuboctahedral configuration Procedure 5.1 is used.

To further clarify the approach for the creation of each of the Archimedean polyhedra a flow chart has been given for each one of them. The main body of the flow chart consists of thirteen rhombic boxes, where each box represents an Archimedean polyhedron. The numbers that are given on the side of each box correspond to the ones in Table 4.2. Smaller flow charts with rectangular boxes have been given for each polyhedron describing the order of the procedures that have to be followed for the creation of a polyhedric configuration which is based on an Archimedean polyhedron. For instance, suppose that is required to generate an icosidodecahedron. It may be seen from Plate 4.12 that an icosidodecahedron has a top x-y pentagonal face, a bottom x-y pentagonal face and thirty non x-y faces. Twenty of these non x-y
faces are triangular faces and ten are pentagonal faces. The icosidodecahedron represents the same arrangement of faces in the top part as in the bottom part. As mentioned above the faces with the least number of edges are created first and the faces with the maximum number of edges are created last. Therefore, the triangular faces of the icosidodecahedral configuration are created first. Procedures 2.1 and 2.2 are used for the creation of the triangular faces that constitute the top part of the icosidodecahedral configuration. In order to create the bottom part of the icosidodecahedral configuration the top part is reflected with respect to the x-y plane and Procedure 3.2 is used. Once the twenty triangular faces have been created the pentagonal faces are to be created. Procedure 1.1 is used for the creation of the top x-y pentagonal faces and Procedure 2.3 for the creation of the five non x-y pentagonal faces of the top part. In order to create the five non x-y pentagonal faces that constitute the bottom part of the icosidodecahedral configuration Procedure 3.2 is used. Finally, Procedure 4.2 is used for the creation of the bottom x-y pentagonal face. The letters T, S, P, H, O or D which are given at the bottom of the rectangular boxes in the smaller flow charts represent a triangular, a square, a pentagonal a hexagonal, an octagonal or a decagonal polyhedral face, respectively.

4.4 GEODESIC FORMS

Geodesic forms allow effective use of material and space and may be employed to create architecturally interesting and economic building structures. They are presently used in a number of specialised areas of construction such as domes for arenas, cultural centres, exhibition halls and Olympic facilities. Their widespread use has been obstructed by the difficulty in defining their geometry. This problem has presented a challenge for engineers and architects for decades. Many attempts have been made throughout the world to evolve techniques that deal with the data generation of geodesic forms. However, the approach presented in this work provides a methodology that allows data generation for geodesic forms of all kinds to be handled with ease and elegance.
Most of the existing geodesic domes have been obtained from the radial projection of the triangulated faces of an icosahedron on a sphere. However, in this study a geodesic dome may be obtained by projecting a polyhedral configuration on a surface. For the projection the tractation retronorm will be applied. All the surfaces and types of projections available in the tractation retronorm may be used to generate intersecting geodesic dome configurations.

4.4.1 AN EXAMPLE

Consider the configuration shown in Fig 4.104. This is obtained by projecting the polyhedral configuration of Fig 4.105 on a sphere which is concentric with the icosahedron using the centre of the sphere as the centre of projection. A Formian statement describing this operation may be given as

\[
D = \text{TRAC}(4,1,0,0,0,10,13)|P
\]

where the formex variable \( P \) represents the top five faces of the icosahedral configuration of Fig 4.105 and may be given as

\[
P = \text{PEX}|\text{POL}(5,10,[0,0; 18,01; [1; 5)] | E
\]

Formex variable \( E \) represents the compret of Fig 4.107 and may be given as

\[
E = \text{GENID}(9,9,2,\text{SQRT}|3,1,-1)|\{[0,0;2,0],[2,0;1,\text{SQRT}|3],[1,\text{SQRT}|3;0,0]\}
\]
Fig 4.104

Fig 4.105

Fig 4.107  z, U3 (upwards)

Fig 4.108

Fig 4.109

388
The configuration of Fig 4.104 is an example of a "geodesic form" or "geodesic configuration".

The constituent parts of formex variable D are as follows:

i) TRAC is an abbreviation for tractation and is followed by a sequence of parameters enclosed in parentheses. The construct that consists of TRAC together with the ensuing parameter list is referred to as a "tractation retronorm" and has been discussed in detail in Chapter 3. The "projection specifier" specifies the type of projection to be used. The projection specifier may have the value 1, 2, 3 or 4 indicating central, parallel, axial or radial projection, respectively (See Sections 3.2.1 to 3.2.4). The projection specifier in Fig 4.106 is given as 4, implying radial projection. The "surface specifier" specifies the type of surface on which projection is to be made, (See Section 3.3). The surface specifier for formex variable D is given as 1 implying a sphere and this is followed by the coordinates of the centre of the sphere (0,0,0) and the number 10 that specifies the radius of the sphere. The items that should follow the surface specifier in different cases are as given in Section 3.4. The selector specifies the course of action to be taken when the projection of a point cannot be determined uniquely (See Chapter 3, Table 3.1).

As the next example consider the configuration shown in Fig 4.108. This is a view of a geodesic form obtained by projecting the icosidodecahedral configuration of Fig 4.109 onto an ellipsoid using central projection. A Formian statement describing this operation may be given as

\[
D1 = \text{TRAC}(1,0,0,-10,2,0,0,-5,11,15,20,13)|P1
\]

where the formex variable P1 represents the polyhedral form of Fig 4.109 and may be given as

\[
P1 = \text{POL}(12,10,[0,0; 8,0; [1; 10])|E1 \# \text{POL}(12,10,[0,0; 8,0; [21; 26])|E2
\]
Formex variable $E_1$ represents the compret of the pattern on the triangular faces of the icosidodecahedral configuration, Fig 4.110. Formex variable $E_2$ represents the compret of the pattern on the pentagonal faces of the icosidodecahedral configuration, Fig 4.111.

A geodesic form may have more than one layer. For example, consider the configuration shown in Fig 4.112. This is a view of a double layer geodesic form obtained by projecting the polyhedral configuration of Fig 4.113 on two concentric spherical surfaces using radial projection. A Formian statement describing this operation may be given as

$$D_3 = \text{TRAC}([4,1,4,1,0,0,0,10,13],[4,2,4,1,0,0,0,12,13])|P_2$$

where the formex variable $P_2$ represents the polyhedral form of Fig 4.113 and may be given as

$$P_2 = \text{PEX|POL}(5,[10; 12],[1,\sqrt{3}/3; 11,\sqrt{3}/3],.. [0,0;2,0],[15])|(T#B)$$

Formex variables $T$ and $B$ represent the compret of the double layer configuration of Fig 4.114 and may be given as

$$T = \text{GENID}(6,6,2,\sqrt{3},1,-1)|\text{ROSAD}(1,\sqrt{3}/3,3,120)|.. \{[0,0,2;2,0,2],[0,0,2;1,\sqrt{3}/3,1]\}$$

$$B = \text{GENID}(5,5,2,\sqrt{13},1,-1)|\text{ROSAD}(2,2\sqrt{3}/3,3,120)|.. \{1,\sqrt{3}/3,1;3,\sqrt{3}/3,1\}$$

In Figs 4.112-4.114 the top layer elements are drawn in thick lines and the bottom
layer elements as well as the web elements are drawn in thin lines.

4.4 FURTHER INTERCONNECTION PATTERNS

The concepts and constructs presented in previous sections are combined in this section to represent the compretic and normic properties of a number of polyhedral forms and geodesic dome configurations. Configurations based on all the Platonic and Archimedean polyhedra will be described here in an attempt to explore the immense range of possible shapes and forms for polyhedral and geodesic forms.

To begin with, consider the polyhedral configuration shown in Fig 4.115. This is obtained by placing the triangular pattern of Fig 4.116 on the faces of a tetrahedron. A Formian statement describing this operation may be given as

\[ D = \text{PEX}\mid\text{POL}(1,8,[0,0; 12,0])\mid E \]

where \( D \) is a formex variable representing the entire tetrahedral configuration of Fig 4.115 and where formex variable \( E \) represents the compret of the configuration of Fig 4.116. This configuration is shown in Fig 4.116 together with the normat U1-U2-U3 for the formex formulation. The formulation of formex variable \( E \) may be given as

\[ E = E_1 \# E_2 \]

where

\[ E_1 = \text{GENID}(6,6,2,\sqrt{3},1,-1)\mid\{[0,0; 1,\sqrt{3}], [1,\sqrt{3}; 2,0]\} \]

\[ \text{ROSAD}(1,\sqrt{3}/3,3,120)\mid[0,0; 1,\sqrt{3}/3]\]

and

\[ E_2 \]

\[ = \text{GENID}(6,6,2,\sqrt{13},1,1)\mid\{[0,0; 1,\sqrt{13}], [1,\sqrt{13}; 2,0]\} \]

\[ \text{ROSAD}(1,\sqrt{13}/3,3,120)\mid[0,0; 1,\sqrt{13}/3]\]

\[ = \text{GENID}(6,6,2,\sqrt{13},1,-1)\mid\{[0,0; 1,\sqrt{13}], [1,\sqrt{13}; 2,0]\} \]
The letters A and B which are given in Fig 4.116 should be interpreted in a way similar to the one described in section 4.2.

The type of polyhedron and the configuration that has been mapped onto the faces of a polyhedron may be changed. For instance, the configuration of Fig 4.117 is an octahedral configuration. This is obtained by mapping the configuration of Fig 4.118 onto the faces of an octahedron. The configuration of Fig 4.119 is an example of a geodesic form. This is obtained by first mapping the configuration of Fig 4.120 on the top four faces of an octahedron, as in Fig 4.121 and then projecting the resulting configuration on a sphere which is concentric with the icosahedron, using the centre of the sphere as the centre of projection.

As the next example, consider the configuration of Fig 4.120. This is mapped on different parts of an icosahedron and gives rise to the polyhedric configurations shown in Figs 4.122 and 4.123. The polyhedric form shown in Fig 4.122 is obtained by mapping the configuration of Fig 4.120 on the top five faces of an icosahedron and the polyhedric form of Fig 4.123 is obtained by mapping the configuration of Fig 4.120 on one side of an icosahedron. Finally, the icosahedral configuration of Fig 4.124 is obtained by mapping the configurations of Figs 4.120, 4.125 and 4.126 on the top part of an icosahedron. The geodesic forms shown in Figs 4.127 to 4.129 are obtained by projecting the polyhedric configurations of Figs 4.122 to 4.124 on a sphere using radial projection.

Consider the configuration of Fig 4.130. This is a view of a polyhedric configuration obtained by mapping the configuration of Fig 4.131 onto the three faces of a cube. In creating a polyhedric configuration it is possible to use different configurations for different faces of the base polyhedron. For instance, in the case of Fig 4.132 the
configurations of Figs 4.133 to 4.135 are placed on the first, second and third face of the cube, respectively. As the next example consider the polyhedric configuration shown in Fig 4.136. This is obtained by mapping the configuration of Fig 4.137 onto the three faces of a cube.

A configuration that is mapped onto a face of a polyhedron need not necessarily "fill" the face or "match" the boundaries of the face as demonstrated by the polyhedric configuration of Fig 4.138. Here, the configuration of Fig 4.139 is placed onto the first face of the cube and the configuration of Fig 4.137 is placed onto the second and third faces of the cube. The configuration that is placed onto the first face extends beyond the edges of the face.

Consider the configuration of Fig 4.140. This is a view of a cuboctahedral configuration obtained by placing the configurations of Figs 4.141 and 4.142 on one side of a cuboctahedron. The configuration of Fig 4.141 is placed on the triangular faces and the configuration of Fig 4.142 is placed on the square faces. The geodesic form of Fig 4.143 is obtained by first mapping the configurations of Figs 4.141, 4.142 and 4.144 onto the triangular and the squares faces of a cuboctahedron as in Fig 4.145. In the case of the square faces the configuration of Fig 4.142 is mapped on the top x-y face whereas the configuration of Fig 4.144 is placed on the square faces that constitute the middle part. The configuration of Fig 4.145 is now projected on a sphere, using central projection, to create the geodesic dome of Fig 4.143.

The polyhedric configurations shown in Figs 4.146 to 4.148 are obtained by mapping the configurations of Figs 4.149 and 4.150 on different parts of a small rhombicuboctahedron. Furthermore, the geodesic forms of Figs 4.151 and 4.152 are obtained by projecting the rhombicuboctahedral configuration of Fig 4.146 on a circular and elliptical paraboloid, respectively, using central projection.

By placing the configurations of Figs 4.153 and 4.154 on the triangular and the square faces of the top part of the snub cube the polyhedric configuration of Fig
Fig 4.149
Fig 4.150
Fig 4.151
Fig 4.152
Fig 4.153
Fig 4.154
Fig 4.155
Fig 4.156
Fig 4.157
4.155 is obtained. In order to obtain the geodesic forms of Figs 4.156 and 4.157 the polyhedral configuration of Fig 4.155 is projected on an ellipsoid and a sphere, respectively.

The polyhedral forms of Figs 4.158 and 4.159 are based on a dodecahedron. These are obtained by placing the pattern of Fig 4.160 on different parts of a dodecahedron. Hence, the polyhedral form of Fig 4.158 is obtained by mapping the pattern of Fig 4.160 on the top part of a dodecahedron whereas the polyhedral form of Fig 4.159 is obtained by placing the pattern of Fig 4.160 on one side of a dodecahedron.

As the next example consider the polyhedral forms of Figs 4.161 and 4.162. These are obtained by mapping the configurations of Fig 4.163 on different parts of an icosidodecahedron. The geodesic forms of Figs 4.164 and 4.165 are obtained by projecting the icosidodecahedral configuration of Fig 4.162 onto an ellipsoid and a sphere, respectively. Consider the configuration of Fig 4.166. This is a view of a polyhedral configuration obtained by mapping the patterns of Figs 4.167 and 4.168 on part of a snub dodecahedron. The geodesic form of Fig 4.169 is obtained by projecting the polyhedral form of Fig 4.166 onto a sphere using radial projection.

By placing the patterns of Figs 4.167, 4.168 and 4.170 on different parts of a small rhombicosidodecahedron the polyhedral configurations of Figs 4.171 and 4.172 are obtained. In creating a polyhedral configuration it is possible to use different patterns for different faces of the base polyhedron. For instance, in the case of Figs 4.171 and 4.172 the patterns of Figs 4.167, 4.168 and 4.170 are placed on the triangular, pentagonal and square faces, respectively. By projecting the polyhedral form of Fig 4.172 on an ellipsoid the geodesic form of Fig 4.173 is obtained.

As the next example, consider the configuration of Fig 4.174. This is a view of a polyhedral configuration obtained by mapping the patterns of Figs 4.175 and 4.176 on the triangular and the hexagonal faces of a truncated tetrahedron, respectively. The polyhedral configurations of Figs 4.177 to 4.179 are obtained by placing the
patterns of Figs 4.180 to 4.182 on different parts of a truncated octahedron. For instance, the polyhedral forms of Fig 4.179 is obtained by placing the patterns of Figs 4.180 to 4.182 on one half of the truncated octahedron. The geodesic form of Fig 4.183 is obtained by projecting the polyhedral form of Fig 4.179 on a sphere using radial projection. The configuration of Fig 4.184 is based on a truncated cube. This is obtained by placing the patterns of Figs 4.185 and 4.186 on the octagonal and triangular faces of the truncated cube.

Consider the configuration of Fig 4.187. This is the plan view of a truncated icosahedral configuration obtained by placing the configurations of Figs 4.188 to 4.190 on the top part of a truncated icosahedron. The configuration of Fig 4.188 is placed on the pentagonal faces whereas the configuration of Figs 4.189 and 4.190 are placed on the hexagonal faces. The geodesic forms of Figs 4.191 and 4.192 are obtained by mapping the polyhedral form of Fig 4.187 on a sphere and an ellipsoid, respectively. Furthermore, the polyhedral configuration of Fig 4.193 is based on a great rhombicuboctahedron. This is obtained by placing the patterns of Figs 4.194 to 4.196 on the top part of a great rhombicuboctahedron. The geodesic form of Fig 4.197 is obtained by first mapping the patterns of Figs 4.198 and 4.199 on the top part of a truncated dodecahedron and then projecting this on an ellipsoid.

Finally, consider the polyhedral forms of Fig 4.200 and 4.201. These are based on a great rhombicosidodecahedron. The polyhedral form of Fig 4.200 is obtained by placing the patterns of Figs 4.202 and 4.205 on the square and the decagonal faces, respectively. In the case of the polyhedral form of Fig 4.201 the patterns of Figs 4.02 to 4.205 are mapped onto the square, hexagonal and decagonal faces, respectively. The geodesic form of Fig 4.206 is obtained by projecting the polyhedral form of Fig 4.201 on a sphere using central projection.
INTRODUCTION

In the present work, the configuration processing of grid domes and geodesic forms have been studied. The purpose of this chapter is to bring out the main points emerged and to draw conclusions regarding various aspects. Suggestions for future work are also included.

5.1 CONCLUSIONS

Configuration processing for space structures is tedious and time consuming without using automatic data generation techniques. Formex algebra is a mathematical system that provides a compact and concise means of describing a structural system. It complements the human imagination and allows configurations to be expressed in a
laconic and elegant manner. Using the formex algebra for complex structural systems simplifies the process of data generation. Data generation for grid domes and geodesic forms without a suitable conceptual tool is notoriously difficult.

An important aspect of the present work is the establishment of the concepts and constructs through which grid domes may be created. A grid dome is obtained by projection of a grid pattern on a surface. Therefore, one may write a formex representing the compret of a grid pattern and obtain the normic properties for the grid dome with the same formex in conjunction with the tractation retronorm.

The tractation retronorm enables a configuration to be projected on different types of surfaces such as spheres, ellipsoids, cylinders, hyperbolic paraboloids or planes. Also, in the tractation retronorm the user is able to choose between different types of projections, that is, central parallel, axial or radial projections. Keeping the compretic properties the same different configurations may be obtained by choosing the type of projection. Therefore, depending on the user's requirements different types of projections may be used. Furthermore, the tractation retronorm can be employed for the creation of multi-layer grid domes. In such a case a multi-layer grid configuration is projected on different surfaces. The general form of the tractation retronorm is given as

\[ \text{TRAC}(D_1, D_2, D_3, \ldots, D_n) \]

where each one of the entities \( D_1, D_2, D_3, \ldots, D_n \) is a "descriptor" which describes type of projection, type of surface and the set of signets to be projected.

The tractation retronorm provides a basis for the configuration processing of grid domes in a compact and readily understood manner. The tractation retronorm allows one to work with the same set of tools in all data generation problems for grid domes eliminating the need for the employment of an assortment of programs which are dealing with specific problems.
Geodesic domes constitute an important family of structural systems. They are presently used in a number of specialised areas of construction such as domes for sports arenas, cultural centres, exhibitions halls, Olympic facilities and radomes covering radar installations. Their widespread use has been hindered by the difficulty in defining their geometry.

In this study the configuration processing of geodesic domes is solved in two stages. The first stage in the generation of a geodesic dome is the creation of the polyhedric configuration. Polyhedric configurations based on the Platonic and Archimedean polyhedra are used in this study to demonstrate the concepts and constructs through which polyhedral configurations may be created. The scope of this work, however, is much wider than the applications in relation to Platonic and Archimedean polyhedra. In fact, the approach presented in this work provides a methodology that allows polyhedral configurations of all kinds to be generated in a convenient manner. There is a formex function called the "polyhedron function" that can be used to create formices representing polyhedral configurations. The polyhedron function can be used to create single layer or multi-layer polyhedral configurations.

The last stage in the generation of a geodesic form is the projection of the polyhedral configuration on a surface. For the projection the tractation retronorm will be applied. All the surfaces available in the tractation retronorm may serve the purpose of shaping geodesic domes.

The geodesic domes dealt with in this thesis has been confined to applications relating to structures. However, the scope of the ideas is not limited to the realm of structures and to the structural engineers. Structural chemists have also faced similar difficulties when investigating the geometry of large carbon molecules such as C60. The molecule of C60 is a geodesic form based on a truncated icosahedron.
5.2 SUGGESTIONS FOR FUTURE WORK

Much of the work done in this thesis is exploratory and a great amount of research remain to be carried out in the field of configuration processing of grid domes and geodesic forms. A number of possible directions are suggested as follows:

Many grid domes which are obtained by the projection of a grid on a surface such as sphere, ellipsoid, paraboloid or cylinders are introduced in this work. However, to explore the spectrum of different forms and shapes one may included new surfaces on the tractation retronorm such as hyperboloid, super ellipse (See Ref 11), cone or torus.

Polyhedric and geodesic forms based on the Platonic and Archimedean polyhedra are introduced in the present work. However, possibilities are not exhausted and one may explore further polyhedric and geodesic forms. For instance, one may create polyhedric and geodesic forms based on the duals of the Archimedean polyhedra.

In this study the polyhedral forms are obtained by mapping a configuration on the faces of a Platonic or an Archimedean polyhedron. However, polyhedral forms obtained by replacing the edges or the nodes of a polyhedron by a given configuration have yet to be investigated.
REFERENCES


