A FORMEX APPROACH TO
FINITE ELEMENT MESH GENERATION

BY
M H YASSAEE

A THESIS PRESENTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

CIVIL ENGINEERING DEPARTMENT
UNIVERSITY OF SURREY
FEBRUARY 1986
The increased popularity of the finite element analytical method due to the availability of high speed, large memory computers has led to the solution of many heretofore unsolvable problems. The creation of models for various finite element programs classically has been one of the most costly, error-prone and time-consuming elements in the modelling process.

Automatic mesh generation is an attempt at simplifying input data for finite element programs. The available methods for automatic generation of finite elements address specific classes of problems and thus their application are generally limited.

The main theme of this thesis is to introduce a new approach for finite element mesh generation which is based on the concepts of formex algebra. Formex algebra is a mathematical system that may be used to represent and process any kind of configuration, where the term 'configuration' refers to any collection of entities. Formex algebra provides a powerful and valuable tool for mesh generation and the generality of its mathematical framework allows unlimited scope for extension.

A selection of finite element meshes are used to serve in illustrating the underlying concepts of formex algebra and its applications in mesh generation. Various meshes have been formulated using appropriate formex functions. A new class of retronorms have been introduced which conveniently deal with complex mesh patterns.
To
my mother
and
memory of my father
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. H Nooshin, for his unfailing support, encouragement and advise throughout this work. I appreciate the opportunity to be able to work under his guidance.

Particular thanks are due to Mr P Disney for keeping FORMIAN working at all times, without which the practical application of formex algebra is impossible. His suggestions and help are immensly appreciated. Special thanks go to my colleague Mr El-labbar, for his help, advise and friendship throughout the period of my research. To these and other friends and staff at the space structure research centre of University of Surrey which made the period of my research a memorable one, I extend my sincere appreciation and thanks.

My special thanks go to my family for their moral and financial support. Finally, I would like to thank many friends whose understanding and encouragements made life easier in difficult moments.
1 INTRODUCTION
1.1 General Introduction
1.2 Review of the Finite Element Method
1.3 Mesh Generation and Formex Approach

2 FORMEX ALGEBRA
2.1 Introduction
2.2 Formices
2.3 Equality of Formices
2.4 Variant of a Formex
2.5 Sequation of a Formex
2.6 Composition of Formices
2.7 Libra Notation
2.8 Formex Graphics
2.8.1 Formex Plots
2.8.2 Retrocords
2.8.3 Retronorms
2.8.4 Plotting styles
2.8.5 Retrobases and Probases
2.8.6 Standard Retronorms
2.9 Formex Functions
2.9.1 Transflection Functions
2.9.1.1 Translation Functions
2.9.1.2 Rindle Functions
2.9.1.3 Reflection Functions
2.9.1.4 Lambda Functions
2.9.1.5 Vertition Functions
2.9.1.6 Rossette Functions
2.9.1.7 Projection Functions
2.9.1.8 Dilatation Functions
2.9.1.9 Gemination Functions
2.9.1.10 Triad Functions
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.9.2</td>
<td>Introflection Functions</td>
<td>75</td>
</tr>
<tr>
<td>2.9.2.1</td>
<td>Pexum Functions</td>
<td>76</td>
</tr>
<tr>
<td>2.9.2.2</td>
<td>Cordation Functions</td>
<td>77</td>
</tr>
<tr>
<td>2.9.2.3</td>
<td>Relection Functions</td>
<td>82</td>
</tr>
<tr>
<td>2.9.2.4</td>
<td>Brevic Notation</td>
<td>83</td>
</tr>
<tr>
<td>2.9.3</td>
<td>Node Numbering</td>
<td>86</td>
</tr>
<tr>
<td>2.9.3.1</td>
<td>Concept of a Catena</td>
<td>88</td>
</tr>
<tr>
<td>2.9.3.2</td>
<td>Dictum Functions</td>
<td>89</td>
</tr>
<tr>
<td>2.9.3.3</td>
<td>Redictum Functions</td>
<td>91</td>
</tr>
<tr>
<td>2.9.3.4</td>
<td>Seviation and Latitude Functions</td>
<td>93</td>
</tr>
<tr>
<td>2.9.3.5</td>
<td>Novation Functions</td>
<td>95</td>
</tr>
<tr>
<td>2.9.4</td>
<td>Formex Collation</td>
<td>99</td>
</tr>
<tr>
<td>2.9.4.1</td>
<td>Concept of Rapportance</td>
<td>99</td>
</tr>
<tr>
<td>2.9.4.2</td>
<td>Process of Rapportation</td>
<td>100</td>
</tr>
<tr>
<td>2.9.4.3</td>
<td>Rapported Sequation Functions</td>
<td>104</td>
</tr>
<tr>
<td>2.9.4.4</td>
<td>Rapported Variant Functions</td>
<td>105</td>
</tr>
<tr>
<td>2.9.5</td>
<td>Further Formex Functions</td>
<td>106</td>
</tr>
<tr>
<td>2.9.5.1</td>
<td>Tectrix Functions</td>
<td>106</td>
</tr>
<tr>
<td>2.9.5.2</td>
<td>Vinculum Functions</td>
<td>108</td>
</tr>
<tr>
<td>2.9.5.3</td>
<td>Pansion Functions</td>
<td>109</td>
</tr>
<tr>
<td>2.9.5.4</td>
<td>Depansion Functions</td>
<td>111</td>
</tr>
<tr>
<td>2.9.5.5</td>
<td>Medulla and Rapported Medulla Functions</td>
<td>112</td>
</tr>
<tr>
<td>3</td>
<td>FINITE ELEMENT MESH GENERATION TECHNIQUES</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>115</td>
</tr>
<tr>
<td>3.2</td>
<td>Automatic Meshing Techniques</td>
<td>115</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison of Mesh Generation Schemes</td>
<td>139</td>
</tr>
<tr>
<td>4</td>
<td>FORMEX APPROACH TO FINITE ELEMENT MESH GENERATION</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>144</td>
</tr>
<tr>
<td>4.2</td>
<td>Basic Approach</td>
<td>144</td>
</tr>
<tr>
<td>4.3</td>
<td>Formex Representation of Various Finite Elements</td>
<td>148</td>
</tr>
<tr>
<td>4.4</td>
<td>More Example of Formex Representation of Finite</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>Element Meshes</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Formex Formulation of Finite Element Idealization</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>With More Than One Type of Element</td>
<td></td>
</tr>
</tbody>
</table>
4.6 Formex Formulation of Three Dimensional Finite Element Meshes

4.7 Mesh Grading

4.8 Refinement of Finite Element Meshes

4.9 Local Mesh Modification

4.10 User's Defined Retronorms

5 DEXIANT RETRONORMS

5.1 Introduction

5.2 Simple Dexiant Retronorms

5.3 Further Example on Dexiant Retronorms

5.4 Tridirectional Dexiant Retronorms

5.5 Higher order dexiant retronorms

6 CONCLUSIONS

REFERENCES
CHAPTER 1
1 INTRODUCTION

1.1 GENERAL INTRODUCTION

Scientific advancements have created a need for accurate analysis of large and complex engineering systems. Geometrical complexity, use of composite materials, multiple loading and varying boundary conditions caused major hinderance for the extensive use of rigorous classical methods. These facts with the development of high speed electronic digital computers accelerated the developments of the new approach for analysis called the finite element method. The finite element techniques in conjunction with the electronic digital computers have enabled the numerical idealization and solution of complex engineering problems to be carried out in a systematic manner, and in effect have made possible the practical extension and application of the classical analysis procedures to very complex engineering systems.

Looking back over the development of the finite element method, it is interesting to study its dramatic increase in popularity and in the range of its applications during the past thirty years. It is now considered to be the standard analytical technique for nearly all the areas of engineering design. Of course, it is not coincidental that this growth has been concurrent with the vast developments that have occurred in the electronic computer industry, the finite element method is an analysis tool which is in concordance with the current state of computer hardware development.

The success of the finite element method is based largely on the basic finite element procedures used: the formulation of the problem in variational or weighted residual form, the finite element discretization of this formulation and the effective solution of the resulting finite element equations.
These basic steps are the same whichever problem is considered and provide a general framework and a natural approach to engineering analysis.

1.2 REVIEW OF THE FINITE ELEMENT METHOD

It is now over thirty years since the finite element method was first used in the solution of practical structural engineering problems, although the original concept dates back almost a century. The method was originally developed by engineers in the 1950s to analyse large structural systems for aircrafts. There have been many contributions to the development of the theory of finite element. Fundamental contributions have been made by pioneers such as Zienkiewicz, Turner, Clough and Argris [1.4,1.1,1.2,1.3]. The name 'finite element' was coined in the paper by Clough, in which the technique was presented for plane stress analysis.

The finite element method is essentially a process through which a continuum with infinite degrees of freedom can be approximated as an assemblage of subregions (or elements) each with a specified finite number of degrees of freedom. The finite element discretization involves the assumption of strain or stress field defined on a regional basis.

Today, the concept of the finite element method is a very broad one. Even when the analysis is restricted to the structural and solid mechanic problems, the method can be used in a variety of different ways. However, the approach which is widely used for the solution of practical problems, is the displacement-based finite element method. Practically all major general-purpose analysis programs have been written using this formulation, because of its simplicity, generality and good numerical properties.

The basic steps in the displacement-based finite element
method are as follows

(a) The continuum is separated by imaginary lines or surfaces into a number of 'finite elements'.

(b) A finite number of points are chosen as nodal points. The displacement of these nodal points will be the basic unknown parameters of the problem, just as in simple discrete structural analysis.

(c) A set of functions are chosen to define uniquely the state of the displacements within each 'finite element' in terms of the nodal displacements.

(d) The displacement functions define uniquely the state of strain within an element in terms of the nodal displacements. These strains, together with any initial strains and the constitutive properties of the material will define the state of the stress throughout the element and, hence, also on its boundaries.

(e) For each element the 'stiffness matrix' and 'applied load vector' are calculated. The stiffness matrix and load vector of each element are assembled to give respectively, the 'global stiffness matrix' and 'global load vector' for the complete structure. The resulting system of simultaneous equations is solved for the unknown nodal variables. Finally subsidiary quantities such as stress components are evaluated for each element.

1.3 MESH GENERATION AND FORMEX APPROACH

One of the most unattractive points in the practical use of the finite element method has been the great deal of work
involved in the generation of the mesh. This difficulty arises from the operation of subdividing a continuum into small elements, obtaining the geometrical data for each element and finally preparing a datafile containing appropriate information required by a finite element program.

There are many incentives for automating the mesh generation procedure. For example in Reference 1.5 it is stated that typically 50 percents (or more) of the total analysis time and cost is spent in generating and checking the input data.

To reduce the input cost and the probability of errors, it is necessary to develop mesh generators which automatically produce a finite element mesh of specified degree of refinement with a minimum amount of input data. Several researchers have developed a variety of automated schemes. A brief review of the different techniques can be found in a paper by Thacker [1.6]. However, no optimum technique has yet been found. Most of these schemes have been created for a special application and do not necessarily offer scope for extension.

A structural configuration may be viewed as a collection of interrelated entities, these entities may be physical, such as discrete structural members, joints and loads, or abstract such as 'finite elements' in a continuous structure, topological and geometrical properties, coordinate systems and degrees of freedom. Data preparation of finite element problems normally involves a straightforward recording of known facts but for large and complex systems the volume of information that has to be handled turns data preparation into a time consuming and enslaving task which is expensive and prone to errors.

An excellent and elegant solution for automated data generation is provided by a general system of configuration processing which has formex algebra as its basis. The basic
concepts of formex was originally conceived by H Nooshin in 1972-73. The early ideas went through substantial transformation during the years that followed and an up to date account of the concepts of formex algebra is given in Reference 1.7. Formex algebra is a mathematical system that consists of a set of abstract object, a set of relations and a set of operations and functions. Formex algebra may be used in various ways to overcome the difficulties of data generation. It allows networks of all kinds to be formulated and is a valuable tool in dealing with complex configurations. This general system in explained and expanded in the present work.

The material of the thesis is arranged in the following order:

In Chapter 2 detailed and rigorous description of the concepts of formex algebra is presented. This includes the definition of the concept of a formex and its various relations, operations and functions.

Chapter 3 contains a detailed survey of various mesh generation techniques. This review attempts to place in perspective the wide variety of methods which are available for generation of finite element meshes.

In Chapter 4 the author illustrates the power and capabilities of formex algebra as a mesh generation scheme. The generality of the mathematical framework of formex algebra provide a wide range of areas of application. Various meshes have been formulated and presented in this Chapter. Mesh irregularities, refinements and modification are dealt with using appropriate combinations of formex functions and retronorms.

In Chapter 5 the author introduces a new class of retronorms, namely dexiant retronorms. The complex patterns of
distribution of normat points are achieved through these new class of retronorms and the basis of the idea are introduced through examples. The concepts of generic formulation is also extended to include the changes allowable in the formal retronorms.

Finally, the conclusion of the present work is presented.
2.1 INTRODUCTION

Formex algebra is a mathematical system that consists of a set of abstract entities known as formices, and a number of rules through which these entities are manipulated. The concepts of formex algebra with the associated terminology and notation are described in the present Chapter.

2.2 - FORMICES

A 'formex' is a mathematical entity that consists of an arrangement of integers. The simplest type of a formex is a single integer and is referred to as a 'uniple'. For example 2, 10 and 3+i are uniples, where i is an integer variable.

A sequence of one or more uniples is referred to as a 'reglet'. The number of uniples that constitute a reglet is referred to as the 'grade' of the reglet. A uniple is a 'reglet of the first grade'. A reglet may be written down as

$$[U_1, U_2, \ldots, U_n]$$

where each one of the entities $U_1, U_2, \ldots, U_n$ is a uniple and where if the reglet is of the first grade (that is, if $n=1$), then the enclosing square brackets are omitted.

For example

$$[3, 7],$$
$$[5, 8, 7, -9]$$

and

$$10$$
are three reglets of grade two, four and one, respectively.

Next in the hierarchy are the class of formices that are referred to as maniples. A 'maniple' is a sequence of one or more reglets of the same grade where the grade of these reglets is referred to as the 'grade' of the maniple and the number of these reglets is referred to as the 'plexitude' of the maniple. A reglet is a maniple of the 1st plexitude and a uniple is a maniple of the 1st plexitude and 1st grade.

A maniple may be written in the form

\[ [U_{11}, U_{12}, \ldots, U_{1n}; U_{21}, U_{22}, \ldots, U_{2n}; \ldots; U_{m1}, U_{m2}, \ldots, U_{mn}] \]

where the sequence that are separated by semicolons are reglets that are written without their enclosing square brackets and where if the maniple is of the first grade and plexitude (that is, if \( m=n=1 \)), then the enclosing brackets are omitted.

For example

\[ [2,1; 1,6; 3,-7; 8,12] \]

is a maniple of the fourth plexitude and second grade,

\[ [3,2,5,8] \]

is a maniple of the first plexitude and forth grade and if \( i \) is an integer variable, then

\[ [i,-i,2i; 2,(i+2),(i-4); 4,5,6] \]

is a maniple of the third plexitude and third grade.

The general definition of a formex may now be given. A 'formex' is a sequence of zero or more maniples of the same
grade where the grade of these maniples is referred to as the 'grade' of the formex and the number of these maniples is referred to as the 'order' of the formex.

A formex may be written down using a construct of the form

\[[M_1, M_2, \ldots, M_r]\]

where each of the entities \(M_1, M_2, \ldots, M_r\) is a maniple and where if the formex is of the first order (that is, if \(r=1\)), then the enclosing curly brackets are omitted. A maniple is a formex of the first grade. Also, since a uniple or a reglet is a special case of a maniple, then a uniple or a reglet is, in turn, a special case of a formex.

For example

\[[[2, 3; 4, 6; 2, 8], [3, 8], [4, -10; 38, 7]]\]

is a formex of the third order and second grade,

\[[2, 4, 3, 1, 6, 8]\]

is a formex of the first order and sixth grade and

\[21\]

is a formex of the first order and grade. Also, if \(i\) is an integer variable, then

\[[i+3, 2i+4, 4i-20, 2, 1]\]

is a formex of the first order and fifth grade.

A formex that has no maniple is represented by

\[
\]
and is referred to as the 'empty formex'. The order of the empty formex is equal to zero but its grade is considered to be arbitrary.

Certain components part of a formex have special names. To elaborate, consider the formex

\[ M_1, M_2, \ldots, M_r \]

and let this formex be denoted by \( F \). Any one of the maniples \( M_1, M_2, \ldots, M_r \) is referred to as a 'cantle' of \( F \). Also, if \( F \) is of the \( n \)th grade then any reglet of the \( n \)th grade that is contained in \( F \) is referred to as a 'signet' of \( F \).

For instance, if \( E \) is the formex

\[
[ [4,6,8; 2,3,4], [8,2,1] ]
\]

then

\[
[4,6,8; 2,3,4]
\]

and

\[
[8,2,1]
\]

are the cantles of \( E \) and

\[
[4,6,8],
[2,3,4]
\]

and

\[
[8,2,1]
\]

are the signets of \( E \).

Thus the terms 'uniple', 'reglet', 'maniple' or 'formex' may be used to refer to

a formex which is a part of another formex
or: a formex which is not a part of another formex
or: a formex which is viewed as an independent whole, irrespective of it being or not being a part of another formex.

In contrast, the term 'signet' or 'cantle' is used exclusively to refer to a formex which is a component part of another formex. The sense of 'component' remains associated with the term 'signet' or 'cantle' even when a signet or a cantle happens to be the whole of a formex.

The serial position number of a cantle in a formex is referred to as the 'orderate' of that cantle, for example, the orderates of

\[ [3,4; 7,6] \]

and

\[ [33,24] \]

with respect to

\{ [12,16; 15,17], [3,4; 7,6], [33,24], [41,1; 3,15] \}

are 2 and 3 respectively.

A formex is said to be 'homogeneous' provided that all its cantles are of the same plexitude and is said to be 'nonhomogeneous' otherwise. A homogeneous formex whose cantles are of the mth plexitude is referred to as a homogeneous formex of the mth plexitude. A uniple or a reglet is a homogeneous formex of the first plexitude. Also, a maniple that consists of m signets is a homogeneous formex of the mth plexitude.

A homogeneous formex of the first plexitude is referred to as an 'ingot'. For instance,
is an ingot of the forth order and third grade and

\{2,1,4,11,8,15\}

is an ingot of the sixth order and first grade.

A uniple or a reglet is a special case of an ingot. Also, the empty formex may be regarded as a special case of an ingot.

2.3 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 example, formices

\[[12,3; 41,80], [12,32], [15,18; 19,31; 22,27]\]

and

\[[1,4; 12,17], [31; 61], [17,19; 15,14; 42,31]\]

are of the same constitution and so are the cantles of

\[[12,2,1; 3,4,5], [7,1,9; 15,12,11]\]

but

\[[4,11,15; 15,11,2], [2,11,7]\]

and

\[[2,11,7], [4,11,15; 15,11,2]\]

are not of the same constitution.

Two formices are said to be 'equal' provided that they are of the same constitution and that every uniple in one is equal to the corresponding uniple in the other. The conventional
equality symbol is used to indicate the relationship of equality between two formices. Thus,

\([i, j; m, n], [p, q]\) = \([4, 6, 12, 10], [2, 5]\)

implies that \(i=4, j=6, m=12, n=10, p=2\) and \(q=5\).

2.4 VARIENTS 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 other by a rearrangement of the position 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

- \(F_1 = ([3, 4; 12, 1], [3, 8])\)
- \(F_2 = ([12, 1; 3, 4], [3, 8])\)

then \(F_1\) and \(F_2\) are variants of each other and if

- \(E_1 = [2; 3; 4; 5]\),
- \(E_2 = [3; 2; 4; 5]\)
- \(E_3 = [4; 5; 2; 3]\)

then \(E_1\), \(E_2\) and \(E_3\) are variants of one another. Also if

\[H = ([2, 4; 7, 3; 6, 7], [6, 7; 7, 3; 2, 4], [2, 4; 6, 7; 7, 3])\]

then the cantles of \(H\) are variants of one another.
If $F$ is a uniple, a reglet, an ingot or the empty formex and if $E$ is a variant of $F$ then $E$ and $F$ are bound to be equal. A formex is said to be 'prolate' provided that it contains cantles that are variants of one another and is said to be 'nonprolate' otherwise. A uniple, a reglet or a maniple is nonprolate and so is the empty formex.

### 2.5 SEQUATIONS OF A FORMEX

Two formices are said to be 'sequations' of each other provided that 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

$$E_1 = \{[2,4], [3,21], [12,3; 4,5], [2,1]\},$$
$$E_2 = \{[2,1], [12,3; 4,5], [3,21], [2,4]\}$$

and

$$E_3 = \{[12,3; 4,5], [2,1], [3,21], [2,4]\}$$

then $E_1, E_2$ and $E_3$ are sequations of one another.

If $F$ is a uniple, a reglet, a maniple or the empty formex and if $E$ is a sequation of $F$ then $E$ and $F$ are bound to be equal.

### 2.6 COMPOSITION OF FORMICES

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 $F_2$, and the relationship between $F, F_1$
and F2 is written as

\[ F = F_1 \# F_2. \]

The symbol \( \# \) is referred to as the 'duplus symbol' and is read as 'duplus'. Thus,

\[ F_1 \# F_2 \]

is read as 'F1 duplus F2'.

For example, if

\[ F_1 = [[3,2,1; 3,3,2], [4,6,8]] \]
and
\[ F_2 = [[2,6,11], [2,12,13]] \]
then
\[ F_1 \# F_2 = [[3,2,1; 3,3,2], [4,6,8], [2,6,11], [2,12,13]] \]
and
\[ F_2 \# F_1 = [[2,6,11], [2,12,13], [3,2,1; 3,3,2], [4,6,8]]. \]

Also, if

\[ E = [[2,1], [3,2; 2,3]], \]
\[ F = [3,2; 2,-2; 8,-8] \]
and
\[ G = [4,22] \]
then
\[ E \# F = [[2,1], [3,2; 2,3], [3,2; 2,-2; 8,-8]], \]
\[ F \# G = [[3,2; 2,-2; 8,-8], [4,22]] \]
and
\[ G \# F \# E = [[4,22], [3,2; 2,-2; 8,-8], [2,1], [3,2; 2,3]]. \]

From the definition of the composition process it is clear
that the formices of different grades cannot be composed. In general formex composition has the following basic properties

(1) If E and F are two formices of the same grade then

\[ E \# F \neq F \# E. \]

That is, formex composition is not commutative.

(2) If E, F and G are formices of the same grade, then

\[ (E \# F) \# G = E \# (F \# G). \]

In other words, formex composition is associative.

(3) For any formex F

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

(4) If E and F are two formices of the same grade, then the formices

\[ E \# F \]

and

\[ F \# E \]

are seuation of each other.

2.7 LIBRA NOTATION

Let \( F_i \) denote a formex which is given in terms of an integer variable \( i \). For instance, \( F_i \) may have been given as

\[ \{[2,i; 3i,-i], [3,2i; 2i-3, i]\}. \]

Also, let \( F \) with a subscript other than \( i \), say \( j \), represent a
formex that is obtained by replacing every occurrence of \( i \) in \( F_i \) by \( j \). For instance, with respect to \( F_i \) as given above, \( F_3 \) will be equal to

\[
[[2,3; 9,-3], [3,6; 3,3]].
\]

Furthermore, let \( m \) and \( n \) be any two integers. The construct

\[
\prod_{i=m}^{n} F_i
\]
is a shorthand form of writing a formex that is the result of composition of a sequence of formices each of which is obtained by substituting a value for \( i \) in \( F_i \). More specifically, if \( m<n \) then

\[
\prod_{i=m}^{n} F_i = F_m \# F_{m+1} \# \ldots \ldots \# F_{n-1} \# F_n
\]

and if \( m=n \) then

\[
\prod_{i=m}^{n} F_i = F_m
\]

and if \( m>n \) then

\[
\prod_{i=m}^{n} F_i = F_m \# F_{m-1} \# \ldots \# F_{n+1} \# F_n.
\]

The symbol \( \prod \) is referred to as the 'libra symbol' and a construct such as

\[
\prod_{i=m}^{n} F_i
\]
is referred to as a 'libra operator' and is read as 'libra \( i=m \) to \( n \)'. The variable \( i \) is referred to as 'libra variable'. Furthermore, a construct such as

\[
\prod_{i=m}^{n} F_i
\]
is referred to as 'libra composition' and the notation used
in writing libra compositions is referred to as the 'libra notation'.

For sake of simplicity, in the present work a construct such as \( \text{lib}(i=m,n)Fi \) is used in place of \( \sum_{i=m}^{n} Fi \). The symbol \( : L=M \) is referred to as the 'rallus symbol' and is read as 'rallus' or 'of'.

For example, if \( Fi \) is given as

\[
\begin{align*}
\{[i,-1; -2i,2], [3i,-2; i-1,i]\}
\end{align*}
\]

then

\[
\text{lib}(i=3,4)Fi = \{[3,-1; -6,2], [9,-2; 2,3],
\quad [4,-1; -8,2], [12,-2; 3,4]\}
\]

and

\[
\text{lib}(k=1,5)k^2 = \{1,4,9,16,25\}.
\]

In describing the concept of a libra composition, it was assumed that the formex appearing to the right of a libra operator contains the corresponding libra variable. However this is not an essential requirement. Thus

\[
\text{lib}(i=2,4)\{[2,1; 3,4]\}
\]

is a valid construction and the formex it represents is simply

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

A libra composition may contain a sequence of libra operators and its general form may be written as

\[
\text{lib}(i1=ml,n1)\text{lib}(i2=m2,n2)....\text{lib}(ir=mr,nr)Fi
\]

where \( r>1 \) and where \( F \) is a formex which is normally given in terms of the libra variables \( i1,i2,....,ir \), although it is not absolutely necessary for \( F \) to contain all or any of the
variables $i_1, i_2, \ldots, i_r$. If $r=1$ then the libra composition is said to be 'simple' and if $r>1$ then the libra composition is said to be 'nested'. Also, a nested libra composition that involves $r$ libra operators may be referred to as a 'r-nested' libra composition. There are situations when one or more of the initial and/or terminal values of the libra variables in a nested libra composition are given in terms of the proceeding libra variables. For example the construct

$$\text{lib}(i=1,2)\|\text{lib}(j=i,i-2)\|\text{lib}(k=1,2j+i)\|[i,j; k,i]$$

which is a '3-nested' libra composition is an example of such a situation. The formex represented by a nested libra composition of this kind may be found by proceeding from the left and substituting for the libra variables in turn. Thus, one may start by substituting for $i$

$$\text{lib}(j=1,-1)\|\text{lib}(k=1,2j+1)\|[1,j; k,1] \#$$

$$\text{lib}(j=2,\emptyset)\|\text{lib}(k=1,2j+2)\|[2,j; k,2].$$

Then substituting for $j$ obtaining

$$\text{lib}(k=1,3)\|[1,1; k,1] \# \text{lib}(k=1,1)\|[1,\emptyset; k,1] \#$$

$$\text{lib}(k=1,-1)\|[1,-1; k,1] \# \text{lib}(k=1,6)\|[2,2; k,2] \#$$

$$\text{lib}(k=1,4)\|[2,1; k,2] \# \text{lib}(k=1,2)\|[2,\emptyset; k,2]$$

and this, in turn gives rise to

$$\{[1,1; 1,1], [1,1; 2,1], [1,1; 3,1], [1,\emptyset; 1,1],$$
$$[1,-1; 1,1], [1,-1; 0,1], [1,-1; -1,1], [2,2; 1,2],$$
$$[2,2; 2,2], [2,2; 3,2], [2,2; 4,2], [2,2; 5,2],$$
$$[2,2; 6,2], [2,1; 1,2], [2,1; 2,2], [2,1; 3,2],$$
$$[2,1; 4,2], [2,\emptyset; 1,2], [2,\emptyset; 2,2]\}.$$
2.8 FORMEX GRAPHICS

2.8.1 FORMEX PLOTS

A formex can be graphically represented and a graphical representation of a formex is known as a 'formex plot'. Also, every given geometric configuration may be represented by a formex.

Consider the formex

\[ F = ([1,1; 2,2], [2,2; 3,2], [3,2; 3,3], [3,3; 2,3], [2,3; 3,2], [3,2; 4,1]). \]

and let every distinct signet \([U_1, U_2]\) of \(F\) be represented by a little circle whose centre is given by the point

\[ x = U_1 \]
\[ y = U_2 \]

in a two dimensional Cartesian coordinate system.

Also, let every cantle of \(F\) be represented by the little circles that correspond to its signets, with a straight line joining these circles, where the line is labelled by the orderate of the cantle and an arrowhead is placed on the line to indicate the order of appearance of the signets in the cantle. The resulting configuration which is referred to as a 'plot' of \(F\) is displayed in Fig 2.1.

In general, any graphical representation of a formex is referred to as a 'plot' of that formex. A part of a plot which represents a signet is referred to as a 'tenon' and a part of a plot that represents a cantle of second or higher plexitude is referred to as a 'frond'. Every tenon is drawn relative to a point which is referred to as a 'pivot'. In
Fig 2.1, the centres of the little circles are the pivots. Each pivot relates to a signet and is located by specifying its coordinates with respect to a coordinate system.

In Fig 2.1, the pivots are located using equations

\[
\begin{align*}
x &= U_1 \\
y &= U_2
\end{align*}
\]

which specifies the coordinates of a pivot in terms of the uniples of the corresponding signet. Such equations are referred to as 'coordinate equations'.

2.8.2 RETROCORDS

A choice for an aspect of the shape of a tenon or a frond is referred to as a 'retrocord'. For instance, the choices of a circular shape for a tenons and a straight line for connecting the tenons in a frond are retrocords. The tenons and fronds may be drawn in an infinite number of ways. The choice for a particular retrocord is mainly dependent on the application of formex graphics.

2.8.3 RETRONORMS

A 'retronorm' is a set of rules through which the signets of a formex may be mapped into pivots of a plot. There are three types of retronorms:

1- FORMAL RETRONORMS :

A retronorm which is defined through mathematical formules and/or descriptive statements in a natural language is referred to as a 'formal retronorm'.

30
2- GRAPHICAL RETRONORMS:
A retronorm which is defined in terms of a graphical construction is referred to as a 'graphical retronorm'.

3- TABULAR RETRONORMS:
A retronorm which is defined in terms of a table is referred to as a 'tabular retronorm'.

A formal retronorm usually includes one or more coordinate equations. The general form of a coordinate equation is

$$\Omega = f(U_1, U_2, \ldots, U_n)$$

where $\Omega$ is a coordinate and $U_1, U_2, \ldots, U_n$ are the uniples of a typical signet of a formex. The entity $f(U_1, U_2, \ldots, U_n)$ should be a uniquely determined real number within the ranges of values for the uniples.

As an example consider the formex

$$E = \{[2,1; 3,1], [3,1; 4,1], [3,1; 3,2], [3,2; 3,3], [3,3; 2,2], [2,2; 3,2], [3,3; 4,4], [4,4; 5,3], [5,3; 3,3]\}.$$ 

A plot of $E$ with respect to a two dimensional Cartesian coordinate system is shown in Fig 2.2, whose pivots are obtained using the formal retronorm

$$x = U_1^2,$$
$$y = U_2^2.$$ 

It can be seen that $E$ is plotted in accordance with the
retrocords which were employed in the previous example.

Using the above coordinate equations an array of points indicating all the possible pivot positions in a region of interest can be plotted, say 1 to 6 for $U_1$ and 1 to 5 for $U_2$. This array of points is shown in Fig 2.3. In addition to these points two families of dotted lines are drawn that help to clearly define the array of points. Such an array of points, or such a grid of lines, is referred to as a 'normat'. Also, each one of the grid lines is referred to as a 'normat line' and each one of the grid points, is referred to as a 'normat point'. The term 'normat' is, in fact, an alternative name for a 'graphical retronorm'.

In the normat of Fig 2.3, each one of the vertical normat lines relates to a value for $U_1$ and each one of the horizontal normat lines relates to a value for $U_2$. These values are shown encircled and the direction of ascending values for $U_1$ and $U_2$ are indicated with arrows.

The formex $E_a$ a plot of which is shown in Fig 2.2, may now be re-plotted using the normat of Fig 2.3. The result is shown in Fig 2.4, where only the relevant parts of the normat are shown. It can be seen that in this style of representation the correlation between the formex $E$ and its plot is more clear.

Table 2.1 is an example of a tabular retronorm. A normat representing the above retronorm is shown in Fig 2.5. The normat points are uniquely determined in a two dimensional Cartesian coordinates system. But the normat lines are rather arbitrary. That is, a normat line could be of any shape provided it passes through the appropriate normat points.

The normat of Fig 2.5 is used to draw a plot of the formex
Table 2.1

<table>
<thead>
<tr>
<th></th>
<th>U1</th>
<th>U2</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>-0.8</td>
<td>-1.3</td>
<td>-2.0</td>
<td>-3.0</td>
<td>-3.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-4.0</td>
<td>-1.5</td>
<td>0.3</td>
<td>2.3</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>0.5</td>
<td>-0.5</td>
<td>-0.5</td>
<td>-1.0</td>
<td>-1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-3.2</td>
<td>-1.2</td>
<td>+0.5</td>
<td>3.0</td>
<td>4.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>1.2</td>
<td>0.5</td>
<td>1.0</td>
<td>1.2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-2.5</td>
<td>-0.8</td>
<td>0.7</td>
<td>3.8</td>
<td>5.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>2.0</td>
<td>1.6</td>
<td>2.2</td>
<td>3.0</td>
<td>3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-2.0</td>
<td>-0.3</td>
<td>1.0</td>
<td>3.9</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>3.5</td>
<td>3.0</td>
<td>3.7</td>
<td>4.3</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-3.0</td>
<td>-0.7</td>
<td>1.4</td>
<td>2.8</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>4.4</td>
<td>4.0</td>
<td>4.8</td>
<td>5.3</td>
<td>5.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-3.1</td>
<td>-1.3</td>
<td>0.3</td>
<td>1.8</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>5.0</td>
<td>5.3</td>
<td>5.7</td>
<td>6.3</td>
<td>6.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-3.1</td>
<td>-1.8</td>
<td>-0.4</td>
<td>-0.8</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>6.0</td>
<td>6.5</td>
<td>6.7</td>
<td>7.2</td>
<td>7.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-3.1</td>
<td>-2.2</td>
<td>-1.3</td>
<td>-0.1</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>6.6</td>
<td>7.3</td>
<td>7.8</td>
<td>8.5</td>
<td>9.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>-3.8</td>
<td>-3.0</td>
<td>2.2</td>
<td>-1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ F = \text{lib}(i=0,3)\mid\text{lib}(j=0,1)\mid[2+i, 2+j; 2+i, 3+j; 3+i, 3+j; 3+i, 2+j] \]

as shown in Fig 2.6.

A new retrocord is used here, a 4-plex cantle is shown by a shaded quadrilateral connecting the four tenons, where the order of appearance of the signets in the cantle is indicated by arrowheads but the edge that connects the first and the last tenons has no arrowhead. Also adjacent fronds of a 4-plex cantle are drawn with gaps in between.
Fig 2.5

Fig 2.6
2.8.4 PLOTTING STYLES

In plotting a formex, one may begin by specifying a retronorm through which the positions of the pivots are obtained. The next stage will involve the use of a set of appropriate retrocords in accordance with which the tenons and fronds can be drawn.

Various groups of retrocords, each of which suitable for a type of application can be defined. Three different plotting styles which have very common use have been defined here. Namely, the 'radix', 'normal' and 'Zygmunt' plotting styles.

A retrocord that specifies an aspect of the radix plotting style is referred to as a 'radix retrocord' and a plot that is obtained using the radix plotting style is referred to as a 'radix plot'. In a analogous manner, the terms 'natural retrocord' and 'natural plot' are used in relation to the natural plotting style and the terms 'Zygmunt retrocord' and 'Zygmunt plot' are used in relation to the Zygmunt plotting style. The terms 'R-plot', 'N-plot' and 'Z-plot' are shorthands for 'radix plot', 'natural plot' and 'Zygmunt plot', respectively.

The radix plotting style is the style that has been used for all the plots which have been presented so far. The characteristic feature of the radix plotting style is that it gives rise to plots that closely reflect the particulars of their respective formices.

As an example consider formex E, where

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

then the R-plot of E relative to a two dimensional Cartesian coordinate system with the coordinate equation
Fig 2.7

Fig 2.8
\[x = U1\]
\[y = U2\]

will be as shown in Fig 2.7.

The second plotting style is the natural plotting style and is introduced in terms of an example. An N-plot of the above formex E with respect to the retronorm of Fig 2.7 is shown in Fig 2.8.

The following retrocords are employed in drawing the plot of Fig 2.8

(1) The frond of an \(m\)-plex cantle, where \(m > 2\), is obtained by using the straight lines to connect the pivots relating to its signets in a sequential manner and, in addition if \(m > 3\) then the pivots relating to the first and last signets are connected by a straight line.

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

(3) No indication relating to the order of appearance of the signet in the cantle is included.

(4) No indication regarding the order of appearance of the cantles in the formex is included.

(5) A segment of the plot that involves overlapping parts is represented only once.

An N-plot in which one of the above retrocords is slightly modified is referred to as a 'quasi-natural plot'. In shorthand form it may be written as 'QN-plot'. For example
Fig 2.9

Fig 2.10
The Zygmunt plotting style is used in applications related to multi-layer configurations. To introduce this type of plotting style, let Fig 2.10 represent a plan view of a double layer grid. Fig 2.11 shows the same plan view, but the elements in the top layer are shown in full lines, the elements in the bottom layer are drawn in broken lines and the bracing elements are drawn in dotted lines. This plan view is referred to as a 'Zygmunt plan' or a 'Z-plan'. One may define a Zygmunt plot, or a Z-plot as a formex plot that represents a multi-layer configuration and in which, variations in fronds and/or tenons are used to indicate different layers.

It is convenient for particular applications to plot a formex in a way that its fronds are shown shrinked by a small prescribed amount. Thus, adjacent fronds of an m-plex cantle, where m>2 are drawn with gaps in between. If an N-plot is drawn by employing the above retrocord then the resulting plot may be referred to as a 'marginate natural plot' or 'MN-plot. The terms quasi and marginate can be used in conjunction with other plotting styles to specify special types of plots. For example, Fig 2.6 was in fact an MR-plot and Fig 2.12 shows an QMN-plot of formex E.

2.8.5 RETROBASES AND PROBASES

A 'retrobasis' is a set of rules through which a given formex may be plotted. Thus a retrobasis consists of two different types of rules. Firstly, there are the rules through which the positions of the pivots are obtained and, secondly, there are those rules through which the shapes of tenons and fronds are determined. That is, a retrobasis is a combination of a
retronorm and a collection of retrocords.

In contrast, a 'probasis' is defined as a set of rules through which a given geometric configuration may be represented by a formex. The rules that constitute a probasis may be divided into two different types. The first type of rules provide information regarding the correspondence 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 type of rules provides information about the values of the uniples in the formex and the combination of all the rules of this type is referred to as a 'pronorm'.

For example, consider the configuration shown in Fig 2.13. To represent the configuration by a formex, let the following procords be specified

1) Every one of the numbered squares in the configuration should be represented by a 4-plex cantle.

2) The cantles must appear in the order indicated by the numbers written in their corresponding squares.

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

4) The order of appearance of the signets in the cantles should be as indicated by the dotted line.

Also, suppose the pronorm is specified graphically in Fig 2.13 by two families of labelled lines providing the correspondence between the corners of the squares and the uniples of the required formex. Using the above pronorm and procords, the required formex may be written as follows

\[ \text{lib}(i=1,5)\text{lib}(j=1,3)\{[i,j; i+1,j; i+1,j+1; i,J+1]\}.\]
Fig 2.13

Fig 2.14
The concept of a probasis is the converse of that of a retrobasis. Similarly the concept of a procord is the converse of that of a retrocord and the concept of a pronorm is the converse of that of a retronorm. As in the case of a retronorm, a pronorm may be specified in three different ways. These three type of pronorms are referred to as 'formal', 'graphical' and 'tabular' pronorms.

Two formex plots are said to be 'homobasic' if they are produced using the same retrobasis and are said to be 'nonhomobasic' otherwise. Also, two formices are said to be 'homobasic' if they are obtained through the same probasis and are said to be 'nonhomobasic' otherwise.

2.8.6 STANDARD RETRONORMS

There are six categories of retronorms that relate to commonly used coordinate systems. Three of these are one, two and three dimensional Cartesian coordinate systems and the other three are polar, cylindrical and spherical coordinate systems. The terms 'unifect', 'bifect', 'trifect', 'polar', 'cylindrical' and 'spherical' are used to refer to the above systems, respectively.

Certain special cases of the above retronorms are classified as standard retronorms. There are three families of standard retronorms:

1) BASIANT RETRONORMS:

There are six types of 'basiant retronorms' and the particulars of these are given in table 2.2. Each of the entities b1, b2 and b3 is a coefficient and is referred to as a 'basifactor'. There are two types of basifactors. Namely, those that are associated with linear coordinates x, y, z and r are referred to as 'linear basifactors' and those that are
Table 2.2

<table>
<thead>
<tr>
<th>name</th>
<th>coordinate equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>basiunifect</td>
<td>( x = b_1 \ U_1 )</td>
</tr>
<tr>
<td>basibifect</td>
<td>( x = b_1 \ U_1 )</td>
</tr>
<tr>
<td></td>
<td>( y = b_2 \ U_2 )</td>
</tr>
<tr>
<td>basitrifect</td>
<td>( x = b_1 \ U_1 )</td>
</tr>
<tr>
<td></td>
<td>( y = b_2 \ U_2 )</td>
</tr>
<tr>
<td></td>
<td>( z = b_3 \ U_3 )</td>
</tr>
<tr>
<td>basipolar</td>
<td>( r = b_1 \ U_1 )</td>
</tr>
<tr>
<td></td>
<td>( \theta = b_2 \ U_2 )</td>
</tr>
<tr>
<td>basicylindrical</td>
<td>( r = b_1 \ U_1 )</td>
</tr>
<tr>
<td></td>
<td>( \theta = b_2 \ U_2 )</td>
</tr>
<tr>
<td></td>
<td>( z = b_3 \ U_3 )</td>
</tr>
<tr>
<td>basispherical</td>
<td>( r = b_1 \ U_1 )</td>
</tr>
<tr>
<td></td>
<td>( \theta = b_2 \ U_2 )</td>
</tr>
<tr>
<td></td>
<td>( \gamma = b_3 \ U_3 )</td>
</tr>
</tbody>
</table>

associated with angular coordinates \( \theta \) and \( \gamma \) are referred to as 'angular basifactors'.

For example, a basibifect retronorm with

\[ b_1 = 10 \text{ unit length} \]

and

\[ b_2 = 5 \text{ unit length} \]

would give rise to the normat a part of which is shown in
Fig 2.14. Also, a basicylindrical retronorm with

\[ b_1 = 10 \text{ unit length,} \]
\[ b_2 = \frac{\pi}{16} \]

and
\[ b_3 = 3 \text{ unit length} \]

would give rise to the normat a part of which is shown in Fig 2.15.

If \( b \) is an angular basifactor and if

\[ N \, b = 2\pi \]

then one may obtain a value for \( N \). For the above basicylindrical retronorm the value for \( N \) would be 32. This retronorm may be referred to as a '32-sect' retronorm. Thus one may specify \( b \) indirectly by giving the value for \( N \). This style of specification for angular basifactors is referred to as the 'sectorial notation' and may also be used in relation to basispherical retronorms.

2) PARIANT RETRONORMS:

The second family of standard retronorms to be introduced are referred to as 'pariant retronorms'. There are six types of pariant retronorms and the particulars of these are given in table 2.3.

It can be seen from the tabulated particulars, that the pariant retronorms are special cases of the basiant retronorms. In another word, pariant retronorms are obtained by letting every linear basifactor to be equal to one unit length. For example Fig 2.16 shows a 20-36-sect parispherical retronorm.

The term 'intrinsic retronorm' may be used to refer to a
Table 2.3

<table>
<thead>
<tr>
<th>name</th>
<th>coordinate equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>pariunifect</td>
<td>$x = U_1$</td>
</tr>
<tr>
<td>paribifect</td>
<td>$x = U_1$ $y = U_2$</td>
</tr>
<tr>
<td>paritrifect</td>
<td>$x = U_1$ $y = U_2$ $z = U_3$</td>
</tr>
<tr>
<td>paripolar</td>
<td>$r = U_1$ $\theta = b_2 \ U_2$</td>
</tr>
<tr>
<td>paricylindrical</td>
<td>$r = U_1$ $\theta = b_2 \ U_2$ $z = U_3$</td>
</tr>
<tr>
<td>parispherical</td>
<td>$r = U_1$ $\theta = b_2 \ U_2$ $\gamma = b_3 \ U_3$</td>
</tr>
</tbody>
</table>

'pariunifect', 'paribifect' or 'paritrifect' retronorm. Also, an intrinsic plot of a formex whose grade is the same as the dimension of the coordinate system with respect to which it is plotted may be referred to as a 'pertrinsic plot'.

3) METRIANT RETRONORMS:

The third family of standard retronorms are referred to as 'metriant retronorms'. The definitions of metriant
<table>
<thead>
<tr>
<th>name</th>
<th>coordinate equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>metriunifect</td>
<td>$x = b_1 \text{ met}(U_1, m_1)$</td>
</tr>
</tbody>
</table>
| metribifect         | $x = b_1 \text{ met}(U_1, m_1)$  
|                     | $y = b_2 \text{ met}(U_2, m_2)$  |
| metritrifect        | $x = b_1 \text{ met}(U_1, m_1)$  
|                     | $y = b_2 \text{ met}(U_2, m_2)$  
|                     | $z = b_3 \text{ met}(U_3, m_3)$  |
| metripolar          | $r = b_1 \text{ met}(U_1, m_1)$  
|                     | $\theta = b_2 \text{ met}(U_2, m_2)$  |
| metricylindrical    | $r = b_1 \text{ met}(U_1, m_1)$  
|                     | $\theta = b_2 \text{ met}(U_2, m_2)$  
|                     | $z = b_3 \text{ met}(U_3, m_3)$  |
| metrispherical      | $r = b_1 \text{ met}(U_1, m_1)$  
|                     | $\theta = b_2 \text{ met}(U_2, m_2)$  
|                     | $\gamma = b_3 \text{ met}(U_3, m_3)$ |

Retronorms involve a particular scalar function which is known as the 'metril function'. The metril function is based on the concept of geometric progression and is described as follows:

Let $U$ be an integer and $m$ be a nonzero positive real number. Let $R$ be obtained in accordance with the following rule

If $U > 0$ then
\[ R = 1 + m + m^2 + \ldots + m^{U-1} \]

and if \( U = 0 \) then

\[ R = 0 \]

and if \( U < 0 \) then

\[ R = -1 - m - m^2 - \ldots - m^{-U-1}. \]

Using the well established formula for the sum of the terms in a geometric progression the above rule may be given in the following form

If \( m = 1 \) or if \( U = 0 \) then

\[ R = U \]

and if \( m \neq 1 \) and \( U \neq 0 \) then

if \( U > 0 \) then

\[ R = \frac{1 - m^U}{1 - m} \]

and if \( U < 0 \) then

\[ R = \frac{1 - m^{-U}}{m - 1}. \]

The rule through which \( R \) is obtained from \( U \) and \( m \) is represented in terms of a function. This function is referred to as the 'metril function' and the abbreviation 'met' is used to symbolize it. The term \( R \) is referred to as the metril of \( U \) and \( m \) and the relation between \( U \), \( m \) and \( R \) is written as

\[ R = \text{met}(U, m). \]
The family of metriant retronorms may now be defined in terms of the metril function. There are six types of metriant retronorms and the particulars of these are given in table 2.4.

The terms b1, b2 and b3 in table 2.4 are the basifactors as described for the basiant retronorms. The terms m1, m2 and m3 which are nonzero positive coefficients are referred to as 'metrifactors'.

A metripolar retronorm is shown in Fig 2.17 and the coefficient b1, b2, m1 and m2 are respectively 1.0, 0.9, 0.8.

2.9 FORMEX FUNCTIONS

In scalar algebra, a relation such as

\[ y = x^2 - 2x + 3 \]

defines a rule which may be used to evaluate y for any given x. The terms 'independent' and 'dependent' variables are used to refer to x and y respectively and the relation is represented by

\[ y = f(x). \]

The term f is referred to as a 'function' and symbolizes the rule by which y is obtained from x.

In an analogous manner, formices may assume the role of dependent and independent variables. Thus a formex function, \( \Phi \), establishes a rule by which a formex F' may be obtained from another formex F, that is used in a manner compatible with the traditional concept of function, i.e.
\[ F' = \phi \downarrow F \]

where \( F \) assumes the role of independent variable and the symbol \( \downarrow \) is referred to as the 'rallus symbol' and is read as 'rallus' or 'of'.

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

1) If \( E \) and \( G \) are two formices and

\[ G = \phi \downarrow E \]

then it may happen that \( E \) is also expressible in terms of \( G \). In this case the function is said to have an 'inverse'. The inverse of a function \( \phi \) is denoted by \( \phi^{-1} \) and has the property that

\[ E = \phi^{-1} \downarrow G. \]

2) A composite function obtained from repeated application, say \( r \) times, of a function \( \phi \) is denoted by \( \phi^r \). For example

\[ \phi \downarrow \phi \downarrow E \text{ is written as } \phi^2 \downarrow E. \]

3) A composite function that consists of \( \phi^m \) and \( \phi^n \) is equivalent to \( \phi^{m+n} \).

4) The zeroth power of any function \( \phi \) is referred to as an 'identity function' and has the property that

\[ E = \phi^0 \downarrow E. \]

It often happens that the need for a particular way of processing a formex arises repeatedly. It would then be
convenient to standardize the process by turning it into a function and a number of frequently useful formex functions are described in the present work.

2.9.1 TRANSFLECTION FUNCTIONS

Firstly there is the family of formex functions that are referred to as 'transflection functions'. There are five basic classes of functions in this family and these are known as translation, reflection, vertition, projection and dilatation functions. In addition, any combination of these basic functions is referred to as a transflection.

2.9.1.1 TRANSLATION FUNCTIONS

Let E be a formex of the nth grade, q be any integer and h be a nonzero positive integer less than or equal to n. Let a formex G be obtained from E by replacing every signet

\[ [U_1, U_2, \ldots, U_n] \]

of E by

\[ [W_1, W_2, \ldots, W_n] \]

where for all values if \( i = 1, 2, \ldots, n \) except for \( i = h \)

\[ W_i = U_i \]

and where

\[ W_h = U_h + q. \]

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

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

and is referred to as 'translation function'. The formex G
is referred to as a translation of E and the relation between E and G is written as

\[ G = \text{tran}(h, q) \cdot E. \]

For example, if

\[ F_1 = \begin{bmatrix} [1,2; 1,3], [1,3; 3,3], \\ [3,1; 3,3], [2,1; 3,1] \end{bmatrix} \]

and if

\[ F_2 = \text{tran}(1,5) \cdot F_1, \quad F_3 = \text{tran}(2,5) \cdot F_1 \]

and

\[ F_4 = \text{tran}(1,5) \cdot \text{tran}(2,5) \cdot F_1. \]

Then \( F_2, F_3 \) and \( F_4 \) are found to be

\[ F_2 = \begin{bmatrix} [6,2; 6,3], [6,3; 8,3], \\ [8,1; 8,3], [7,1; 8,1] \end{bmatrix} \]

\[ F_3 = \begin{bmatrix} [1,7; 1,8], [1,8; 3,8], \\ [3,6; 3,8], [2,6; 3,6] \end{bmatrix} \]

and

\[ F_4 = \begin{bmatrix} [6,7; 6,8], [6,8; 8,8], \\ [8,6; 8,8], [7,6; 8,6] \end{bmatrix}. \]

Paribifect R-plots of \( F_1 \) to \( F_4 \) are shown in Fig 2.18 where the plot of \( F_i \) is denoted by \( P_i \). Examination of these plots reveals that if

\[ F_j = \text{tran}(h, q) \cdot F_i \]

then \( P_j \) is obtained by translating \( P_i \) parallel to the \( \text{Uh} \) axis by \( q \) units.

Further examples of translation functions are given below, where
F1 = [2, 1; 1, 3; 2, 5; 4, 5; 5, 3; 4, 1],
F2 = lib(i=2,4)\|\text{tran}(1,4i)\|F1,
F3 = lib(j=2,3)\|\text{tran}(2,4i)\|F2

and

F4 = lib(j=3,7)\|\text{tran}(2,2j)\|F2.

Parabifect N-plots of formices F1 to F4 are shown in Fig 2.19 where the plot of F1 is denoted by Pi.

Now consider the formulation for F2, it is obtained by translating a known formex F1. A formex playing a role such as F1 in a formex formulation is referred to as a 'generant'. Thus, in the above formulation F2 is used as generant in obtaining F3 and F4.

2.9.1.2 Rindle Functions

A particular construct involving the concepts of libra composition and translation functions occur frequently in formex formulations. The general form of this construct may be represented by

\text{lib}(v=0, s-1)\|\text{tran}(h, pv)\|F.

If h, p and F are independent of the libra variable v and if s>1, then the above libra composition may be written as

\text{rin}(h, s, p)\|F

where

\text{rin}(h, s, p)

is referred to as a 'rindle function' and where s and p are referred to as 'spread' and 'pace', respectively.

For example, if
**Fig 2.19**

**Fig 2.20**
\[
F = [1,1; 4,1; 3,3; 4,5; 1,5; 2,3]
\]
and if
\[
F_1 = \text{lib}(i=0,5) \text{|tran}(1,i) \text{|F}
\]
and
\[
F_2 = \text{lib}(i=0,4) \text{lib}(j=0,2) \text{|tran}(1,2i) \text{|tran}(2,4j) \text{|F}.
\]

Then \(F_1\) and \(F_2\) in terms of rindle functions, may be written as
\[
F_1 = \text{rin}(1,6,1) \text{|F}
\]
and
\[
F_2 = \text{rin}(1,5,2) \text{rin}(2,3,4) \text{|F}.
\]

Paribifect N-plots of \(F_1\) and \(F_2\) are shown in Figs 2.20 and 2.21.

2.9.1.3 REFLECTION FUNCTIONS

Let \(E\) be a formex of the nth grade and \(h\) be a nonzero positive integer less than or equal to \(n\). Also, let \(q\) be either an integer or a peninteger, where a 'peninteger' is defined as a rational number of the form \(M/2\) with \(M\) being an odd integer.

Let a formex \(G\) be obtained from \(E\) by replacing every signet \([U_1, U_2, \ldots, U_n]\) of \(E\) by
\[
[W_1, W_2, \ldots, W_n]
\]
where for all values of \(i = 1, 2, \ldots, n\) except for \(i=h\)

\[
W_i = U_i
\]
and where
\[
W_h = 2q - U_h.
\]
The rule by which $E$ is transformed into $G$ is symbolized in terms of a function. This function is denoted by

$$\text{ref}(h, q)$$

and is referred to as a 'reflection function'. The formex $G$ is referred to as a reflection of $E$ and the relation between $E$ and $G$ is written as

$$G = \text{ref}(h, q) \mid E.$$ 

For example if

$$F_1 = \begin{bmatrix} [1, 2; 1, 3], & [1, 3; 3, 3], & [3, 1; 3, 3], & [2, 1; 3, 1] \\ \end{bmatrix}$$

and if

$$F_2 = \text{ref}(1, 5) \mid F_1,$$

$$F_3 = \text{ref}(2, 5) \mid F_1$$

and

$$F_4 = \text{ref}(1, 5) \mid \text{ref}(2, 5) \mid F_1.$$

Then $F_2$, $F_3$ and $F_4$ are found to be

$$F_2 = \begin{bmatrix} [9, 2; 9, 3], & [9, 3; 7, 3], & [7, 1; 7, 3], & [8, 1; 7, 1] \\ \end{bmatrix},$$

$$F_3 = \begin{bmatrix} [1, 8; 1, 7], & [1, 7; 3, 7], & [3, 9; 3, 7], & [2, 9; 3, 9] \\ \end{bmatrix},$$

and

$$F_4 = \begin{bmatrix} [9, 8; 9, 7], & [9, 7; 7, 7], & [7, 9; 7, 7], & [8, 9; 7, 9] \\ \end{bmatrix}.$$ 

Paribifect R-plots of $F_1$ to $F_4$ are shown in Fig 2.22. Examination of these plots reveals that if

$$F_j = \text{ref}(h, q) \mid F_i$$

then $F_j$ is obtained as the mirror image of $F_i$ with respect
to a plane which is normal to the $U_h$ axis and intersects it at a point for which $U_h=q$.

Further examples of reflection functions are given below, where

$$F_1 = \begin{bmatrix} 1,1; 2,1; 2,2; 1,2 \end{bmatrix},$$
$$F_2 = \text{lib}(i=6,9)\text{ref}(1,i/2)\text{Fl},$$
$$F_3 = \text{lib}(j=6,9)\text{ref}(2,j/2)\text{F2}$$

and

$$F_4 = \text{lib}(j=3,5)\text{ref}(2,j)\text{F1}.$$

parabifect N-plots of $F_1$ to $F_4$ which are denoted by $P_1$ to $P_4$ are shown in Fig 2.23.

2.9.1.4 LAMBDA FUNCTIONS

The construct

$$\text{lib}(i=0,1)\text{ref}(h,q)^l\text{E}$$

occurs frequently in formex formulations. A convenient way of representing this construct is to write it down as

$$\text{lam}(h,q)^l\text{E}$$

where

$$\text{lam}(h,q)$$

is referred to as a 'lambda function' and represents

$$\text{lib}(i=0,1)\text{ref}(h,q)^l.$$

For example if

$$E = \begin{bmatrix} 1,1; 2,3 \end{bmatrix}$$
Fig 2.23

Fig 2.24
then instead of
\[ F = \text{lib}(i=0,1) \ref (1,2)^i \mid E. \]
one may write
\[ F = \text{lam}(1,2) \mid E \]
where \( F \) represents \( E \) and its reflection in the \( U_1 \) direction about 2.

As another example consider the formex
\[ E_1 = [2,1; 1,2] \]
then
\[ E_2 = \text{lam}(2,3) \mid \text{lam}(1,5) \mid \text{lam}(1,3) \mid \text{lam}(1,2) \mid \text{lam}(2,2) \mid E_1. \]

A paribifect \( N \)-plot of \( E_2 \) is shown in Fig 2.24.

2.9.1.5 VERTITION FUNCTIONS

Let \( E \) be a formex of the \( n \)th grade with \( n \) being greater than or equal to 2. Also, let \( h_1 \) and \( h_2 \), \( h_1 \neq h_2 \), be two nonzero positive integers less than or equal to \( n \). Furthermore, let \( q_1 \) and \( q_2 \) be either any two integers or any two penintegers. Thus, \( q_1=2, q_2=-4 \) and \( q_1=3/2, q_2=7/2 \) are acceptable pairs, but \( q_1=4, q_2=-21/2 \) and \( q_1=11/2, q_2=2 \) are not. Let a formex \( G \) be obtained from \( E \) by replacing every signet
\[ [U_1, U_2, \ldots, U_n] \]
of \( E \) by
\[ [W_1, W_2, \ldots, W_n] \]
where for all values of \( i=1,2,\ldots,n \) except for \( i=h_1 \) and \( i=h_2 \)
\[ W_i = U_i \]
and where
\[ \text{Wh}_1 = q_2 + q_1 - U_{h2} \]
and
\[ \text{Wh}_2 = q_2 - q_1 + U_{h1}. \]

The rule by which \( E \) is transformed into \( G \) is symbolized in terms of a function. This function is denoted by
\[ \text{ver}(h_1,h_2,q_1,q_2) \]
and is referred to as a 'vertition function'. The formex \( G \) is referred to as vertition of \( E \) and the relation between \( E \) and \( G \) is written as
\[ G = \text{ver}(h_1,h_2,q_1,q_2) \upharpoonright E. \]

For example, if
\[ F_1 = \{[1,2; 1,3], [1,3; 3,3], [3,1; 3,3], \]
\[ [2,1; 3,1], [1,2; 2,2] \} \]
and if
\[ F_2 = \text{ver}(1,2,4,4) \upharpoonright F_1, \]
\[ F_3 = \text{ver}(2,1,9/2,7/2) \upharpoonright F_1 \]
and
\[ F_4 = \text{ver}(1,2,1,7) \upharpoonright F_1. \]

Then \( F_2, F_3 \) and \( F_4 \) are found to be
\[ F_2 = \{[6,1; 5,1], [5,1; 5,3], [7,3; 5,3], \]
\[ [7,2; 7,3], [6,1; 6,2] \}, \]
\[ F_3 = \{[1,7; 2,7], [2,7; 2,5], [0,5; 2,5], \]
\[ [0,6; 0,5], [1,7; 1,6] \} \]
and
\[ F_4 = \{[6,7; 5,7], [5,7; 5,9], [7,9; 5,9], \]
\[ [7,8; 7,9], [6,7; 6,8] \}. \]

Parabifect R-plots of \( F_1 \) to \( F_4 \) are shown in Fig 2.25, where
the plot of $F_i$ is denoted by $P_i$.

Here, if

$$F_j = \text{ver}(h_1, h_2, q_1, q_2) \mid F_i$$

then $P_j$ is obtained by rotating $P_i$ through $\pi/2$ about an axis that is perpendicular to $Uhl-Uh2$ plane and intersects this plane at a point for which $Uhl=q_1$ and $Uh2=q_2$ and where the sense of the rotation is such that a rotation of $Uhl$ through $\pi/2$ about the origin will map the positive side of $Uhl$ onto that of $Uh2$.

Further examples of the use of vertition functions are given below, where

$$F_1 = \begin{bmatrix} 2,2; 2,3 \end{bmatrix}$$

and where $F_2$ to $F_4$ are obtained as

$$F_2 = \text{lib}(i=0,3) \mid \text{ver}(1,2,4,2) \mid \text{ver}(1,2,3,3) \mid F_1,$$

$$F_3 = \text{lib}(i=0,3) \mid \text{ver}(1,2,9,3) \mid \text{ver}(1,2,6,4) \mid F_2$$

and

$$F_4 = \text{lib}(j=0,1) \mid \text{ver}(1,2,3,8) \mid \text{lib}(i=0,3) \mid \text{ver}(1,2,2,7) \mid \text{ver}(2,1,6,5) \mid F_2.$$  

Paribifect N-plots of $F_1$ to $F_4$ are shown in Fig 2.26, where the plot of $F_i$ is denoted by $P_i$.

It should be mentioned that for any even power of a vertition function, the restriction that 'q1 and q2 must be both integers or both penintegers' is unnecessary and q1 and q2 may independently be integers or penintegers.
2.9.1.6 ROSETTE FUNCTIONS

The construct

$$\text{lib}(i=0,3)\mid \text{ver}(h_1,h_2,q_1,q_2)^i \mid E$$

occurs quite frequently in practice. A convenient way of representing this construct is to write it as

$$\text{ros}(h_1,h_2,q_1,q_2) \mid E$$

where

$$\text{ros}(h_1,h_2,q_1,q_2)$$

is referred to as a 'rosette function' and represents

$$\text{lib}(i=0,3)\mid \text{ver}(h_1,h_2,q_1,q_2)^i$$.

To exemplify the application of rosette functions, consider the configuration of Fig 2.27. Let it be required to write a formex representing the interconnection pattern of this configuration relative to the indicated basibifect retronorm shown. Such a formex may be written as

$$F = \text{ros}(1,2,7,7) \mid \text{ros}(1,2,4,4) \mid F_1$$

where

$$F_1 = \{[4,2; 1,1], [2,4; 1,1], [3,4; 4,3]\}.$$

2.9.1.7 PROJECTION FUNCTIONS

Let $E$ be a formex of the $n$th grade, $q$ be any integer and $h$ be a nonzero integer less than or equal to $n$. Let a formex $G$ be obtained from $E$ by replacing every signet

$$[U_1,U_2,\ldots,U_n]$$

of $E$ by

$$[W_1,W_2,\ldots,W_n]$$
where for all values of \( i = 1, 2, \ldots, n \) except for \( i = h \)

\[ W_i = U_i \]

and where

\[ W_h = q. \]

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

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

and is referred to as a 'projection function'. The formex \( G \) is referred to as a projection of \( E \) and the relation between \( E \) and \( G \) is written as

\[ G = \text{proj}(h, q) \vdash E. \]

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

For example, if

\[
F_1 = \{[[1, 2; 1, 3], [1, 3; 3, 3], [3, 1; 3, 3], [2, 1; 3, 1]]\}
\]

and if

\[
F_2 = \text{proj}(1, 6) \vdash F_1,
F_3 = \text{proj}(2, 5) \vdash F_1
\]

and

\[
F_4 = \text{proj}(1, 6) \vdash \text{proj}(2, 5) \vdash F_1
\]

then \( F_2, F_3 \) and \( F_4 \) are found to be

\[
F_2 = \{[[6, 2; 6, 3], [6, 3; 6, 3], [6, 1; 6, 3], [6, 1; 6, 1]],
F_3 = \{[[1, 5; 1, 5], [1, 5; 3, 5], [3, 5; 3, 5], [2, 5; 3, 5]]\}
\]
and

\[ F_4 = [[6,5; 6,5], [6,5; 6,5], [6,5; 6,5], [6,5; 6,5]]. \]

Parabifect R-plots of \(F_1\) to \(F_4\) are shown in Fig 2.28, where the plot of \(F_i\) is denoted by \(P_i\).

Two new radix retrocords are introduced in Fig 2.28. The first retrocord provides a style for drawing the frond of a 2-plex cantle whose signets are equal, as shown in plot of \(F_2\) and \(F_3\). The second retrocord provides a style for drawing coincident fronds of the above type, as shown for the plot of \(F_4\).

Examination of the plots of Fig 2.28 reveals that if

\[ F_j = \text{proj}(h, q)|F_i \]

then \(P_j\) is obtained by projecting \(P_i\) onto a plane that is perpendicular to the \(U_h\) axis and intersects it at a point for which \(U_h = q\).

2.9.1.8 DILATATION FUNCTIONS

Let \(E\) be a formex of the \(n\)th grade and \(h\) be a nonzero positive integer less than or equal to \(n\). Also, let \(q\) be a rational number such that if \(U_h\) denotes the \(h\)th uniple of a signet of \(E\), then for every signet of \(E\) the product \(qU_h\) is an integer. Let a formex \(G\) be obtained from \(E\) by replacing every signet

\[ [U_1, U_2, \ldots, U_n] \]

of \(E\) by

\[ [W_1, W_2, \ldots, W_n] \]

where for all values of \(i = 1, 2, \ldots, n\) except for \(i = h\)
\[ W_i = U_i \]

and where
\[ W_h = qU_h. \]

The rule by which \( E \) is transformed into \( G \) is denoted by
\[ \text{dil}(h,q) \]

and is referred to as a 'dilatation function'. The formex \( G \) is referred to as a dilatation of \( E \) and the relation between \( E \) and \( G \) is written as
\[ G = \text{dil}(h,q) \mid E. \]

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

For example, if
\[
F_1 = \begin{bmatrix} 1,2 & 1,3 \end{bmatrix}, \begin{bmatrix} 1,3 & 3,3 \end{bmatrix}, \begin{bmatrix} 3,1 & 3,3 \end{bmatrix}, \begin{bmatrix} 2,1 & 3,1 \end{bmatrix}
\]

and if
\[
F_2 = \text{dil}(1,4) \mid F_1,
F_3 = \text{dil}(2,4) \mid F_2
\]

and
\[
F_4 = \text{dil}(1,1/4) \mid F_3
\]

then \( F_2 \) to \( F_4 \) are found to be
\[
F_2 = \begin{bmatrix} 4,2 & 4,3 \end{bmatrix}, \begin{bmatrix} 4,3 & 12,3 \end{bmatrix}, \begin{bmatrix} 12,1 & 12,3 \end{bmatrix}, \begin{bmatrix} 8,1 & 12,1 \end{bmatrix}
\]
\[
F_3 = \begin{bmatrix} 4,8 & 4,12 \end{bmatrix}, \begin{bmatrix} 4,12 & 12,12 \end{bmatrix}, \begin{bmatrix} 12,4 & 12,12 \end{bmatrix}, \begin{bmatrix} 8,4 & 12,4 \end{bmatrix}
\]

and
\[
F_4 = \begin{bmatrix} 1,8 & 1,12 \end{bmatrix}, \begin{bmatrix} 1,12 & 3,12 \end{bmatrix}, \begin{bmatrix} 3,4 & 3,12 \end{bmatrix}, \begin{bmatrix} 2,4 & 3,4 \end{bmatrix}
\]
Paribifect R-plots of Fl to F4 are shown in Fig 2.29, where the plot of Fi is denoted by Pi.

Examination of these plots reveals that if

\[ F_j = \text{dil}(h,q) \vert F_i \]

then Pj is obtained by a stretching or contraction of Pi by a factor \(|q|\) in a direction parallel to the Uh axis. Furthermore, there is normally an accompanied translational displacement and if \(q<0\) then there will also be an additional reflective effect.

As another example consider the formex

\[ F = \text{lib}(j=1,4) \vert \text{tran}(2,3(j-1)) \vert \text{dil}(2,j) \vert F_1 \]

where

\[ F_1 = \text{lib}(i=1,4) \vert \text{tran}(2,3(i-1)) \vert \text{dil}(1,i) \vert F_2 \]

and where

\[ F_2 = \text{ros}(1,2,3,3) \vert \begin{bmatrix} 3,1; & 0,0; & 1,3; \\ 3,2; & 2,3 \end{bmatrix}. \]

A paribifect quasi-natural plot of F in which the tenons are shown by small circles is shown in Fig 2.30.

2.9.1.9 GEMINATION FUNCTIONS

A 'gemination function' is a transflection function that relates to two directions in accordance with the pattern to be discussed in the sequel. There are three categories of gemination functions and these are known as geminid, geminis and geminit functions.

A 'geminid function' relates to directions 1 and 2 in the order 1;2. For example the construct
tranid(q1,q2)

is equivalent to

\[ \text{tran}(2,q2) \mid \text{tran}(1,q1) \]

and is referred to as 'translation geminid function'.

The abbreviation for a geminid function is obtained by attaching the suffix 'id' to the abbreviated name of the function from which the geminid function is derived. There are a set of eight types of geminid functions which may be written as

\[ \text{tranid}(q1,q2), \]
\[ \text{rinid}(s1,s2,p1,p2), \]
\[ \text{refid}(q1,q2), \]
\[ \text{lamid}(q1,q2), \]
\[ \text{verid}(q1,q2), \]
\[ \text{rosid}(q1,q2), \]
\[ \text{projid}(q1,q2), \]

and finally

\[ \text{dilid}(q1,q2). \]

Analogously, there is a set of eight gemination functions that relate to directions 1 and 3 in the order 1,3. The name 'geminis function' and the suffix 'is' are used in relation to this set of gemination functions. There is also a set of eight gemination functions that relate to directions 2 and 3 in the order 2,3. The name 'geminit function' and the suffix 'it' are used to refer to this type of gemination functions.

2.9.1.10 TRIAD FUNCTIONS

A 'triad function' is a transflection function that relates
to directions 1, 2 and 3 in the order 1, 2, 3. The abbreviation for a triad function is obtained by attaching the suffix 'ad' to the abbreviated name of the function from which the triad function is derived.

For example, the construct

\[ \text{tranad}(q_1, q_2, q_3) \]

is equivalent to

\[ \text{tran}(3, q_3) \| \text{tran}(2, q_2) \| \text{tran}(1, q_1) \]

and is referred to as a 'translation triad function'.

There are six types of triad functions which are given below:

\[ \text{tranad}(q_1, q_2, q_3), \]
\[ \text{rinad}(s_1, s_2, s_3, p_1, p_2, p_3), \]
\[ \text{refad}(q_1, q_2, q_3), \]
\[ \text{lamad}(q_1, q_2, q_3), \]
\[ \text{projad}(q_1, q_2, q_3) \]
and finally
\[ \text{dilad}(q_1, q_2, q_3). \]

2.9.2 INTROFLECTION FUNCTIONS

Formex formulation of configurations has been accomplished so far by using the concepts of formex composition and transflection functions. A configuration may be formulated using the concept of formex composition only. The addition of transflection functions, increases the ability of dealing with the problems in a convenient manner. This ability may be further enhanced by using a family of formex functions that are known as introflection functions. These
functions allow formices to be curtailed in various ways and are of particular value when irregular interconnection patterns are being formulated.

Three basic classes of introflection functions are described in the present Section. These are known as pexum, cordation and relection functions. The class of cordation functions, in turn, consists of four subclasses of functions that are known as nexum, luxum, conexum and coluxum functions.

2.9.2.1 PEXUM FUNCTION

Consider a formex $E$ and let every cantle $C$ of $E$ that satisfies the following condition be deleted from $E$:

There are one or more cantles in $E$ that are variants of $C$ and whose orderates are less than that of $C$.

The resulting formex is referred to as the 'pexum' of $E$.

The rule by which a formex is transformed into its pexum is symbolized in terms of a function. This function is denoted by 'pex' and is referred to as the 'pexum function'. If $G$ is the pexum of a formex $E$, then the relation between $E$ and $G$ is written as

$$G = \text{pex}\{E\}.$$  

The pexum of the empty formex is considered to be the empty formex itself.

For example, if

$$F_1 = \{\langle 2,2; 4,4 \rangle, \langle 2,2; 4,4 \rangle, \langle 4,4; 2,2 \rangle\},$$
$$F_2 = \{\langle 1,6 \rangle, \langle 1,6 \rangle, \langle 1,6 \rangle, \langle 1,6 \rangle\},$$
\[ F_3 = ([3,5; 3,5; 3,5; 3,5]) \]

and

\[ F_4 = ([1,2; 2,1], [1,5; 6,1], [6,1; 2,1]) \]

then

\[ pex(F_1) = ([2,2; 4,4]), \]
\[ pex(F_2) = ([1,6]), \]
\[ pex(F_3) = ([3,5; 3,5; 3,5; 3,5]) \]

and

\[ pex(F_4) = ([1,2; 2,1], [1,5; 6,1], [6,1; 2,1]). \]

The pexum function has the following basic properties

(1) The pexum function has no inverse.

(2) If \( E \) is a nonprolate formex then

\[ pex(E) = E \]

(3) If \( k \) is nonzero positive integer then

\[ pex^k(E) = pex(E). \]

2.9.2.2 CORDATION FUNCTIONS

Let \( E \) and \( F \) be two formices of the same grade and let \( G_1 \) to \( G_4 \) be obtained from \( E \) in the following manner:

\( G_1 \) is obtained by deleting every cantle of \( E \) that includes one or more signets that are not in \( F \),

\( G_2 \) is obtained by deleting every cantle of \( E \) that includes one or more signet that are in \( F \),

\( G_3 \) is obtained by deleting every cantle of \( E \) that consists of signets all of which are in \( F \),

and
G4 is obtained by deleting every cantle of E that consists of signets none of which are in F.

G1 is referred to as the 'nexum' of E with respect to F and the relation between E and G1 is written as

\[ G_1 = \text{nex}(F) \upharpoonright E. \]

G2 is referred to as the 'luxum' of E with respect to F and the relation between E and G2 is written as

\[ G_2 = \text{lux}(F) \upharpoonright E. \]

G3 is referred to as the 'conexum' of E with respect to F and the relation between E and G3 is written as

\[ G_3 = \text{con}(F) \upharpoonright E. \]

G4 is referred to as the 'coluxum' of E with respect to F and the relation between E and G4 is written as

\[ G_4 = \text{col}(F) \upharpoonright E. \]

The functions \( \text{nex}(F) \), \( \text{lux}(F) \), \( \text{con}(F) \) and \( \text{col}(F) \) are referred to as 'nexum function', 'luxum function', 'conexum function' and 'coluxum function', respectively, and the term 'cordation function' is used to refer to any of these functions. Also, the construct

\[ \text{cord}(F) \]

is used to mean \( \text{nex}(F) \), \( \text{lux}(F) \), \( \text{con}(F) \) or \( \text{col}(F) \).

As an example consider the formices
\[ E = \{[[1,1; 7,1], [7,1; 4,6], [4,6; 1,1]], [1,1; 3,3], [7,1; 4,2], [4,6; 5,3], [3,3; 5,3], [3,3; 4,2], [4,2; 5,3]\} \]

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

and let
\[ G_1 = \text{nex}(F)\upharpoonright E, \]
\[ G_2 = \text{lux}(F)\upharpoonright E, \]
\[ G_3 = \text{con}(F)\upharpoonright E \]

and
\[ G_4 = \text{col}(F)\upharpoonright E. \]

The formices \( G_1 \) to \( G_4 \) are found to be
\[ G_1 = [[3,3; 5,3], [3,3; 4,2], [4,2; 5,3]], \]
\[ G_2 = [[1,1; 7,1], [7,1; 4,6], [4,6; 1,1]], \]
\[ G_3 = [[1,1; 7,1], [1,1; 3,3], [7,1; 4,2], [4,6; 5,3]] \]

and
\[ G_4 = [[1,1; 3,3], [7,1; 4,2], [4,6; 5,3], [3,3; 5,3], [3,3; 4,2], [4,2; 5,3]]. \]

Parbifect N-plots of \( E \) and \( F \) are shown in Figs 2.31 and 2.32, respectively. Also, paribifect N-plots of different cordation functions of \( E \) with respect to \( F \) (i.e. \( G_1 \) to \( G_4 \)) are shown in Figs 2.33 to 2.36, respectively.

Some basic properties of cordation function are as follows

(1) A cordation function does not have an inverse.

(2) For any formex \( E \)

\[ \text{nex}([ ])\upharpoonright E = [ ], \]
\[ \text{lux}([ ])\upharpoonright E = E, \]
\[ \text{con}([ ])\upharpoonright E = E \]

and
(3) For any formex $E$

$$\text{nex}(E) \mid E = E,$$
$$\text{lux}(E) \mid E = \emptyset,$$
$$\text{con}(E) \mid E = \emptyset$$

and

$$\text{col}(E) \mid E = E.$$

(4) If $E$ and $F$ are any two formices of the same grade then, if all the cantles of $E$ that constitute $\text{nex}(F) \mid E$ are removed from $E$ then the remaining formex is $\text{con}(F) \mid E$ and vice versa. Similarly, if all the cantles of $E$ that constitute $\text{lux}(F) \mid E$ are removed from $E$ then the remaining formex is $\text{col}(F) \mid E$ and vice versa. Hence,

$$\text{nex}(F) \mid E \neq \text{con}(F) \mid E,$$
$$\text{con}(F) \mid E \neq \text{nex}(F) \mid E,$$
$$\text{lux}(F) \mid E \neq \text{col}(F) \mid E$$

and

$$\text{col}(F) \mid E \neq \text{lux}(F) \mid E$$

are sequations of $E$.

(5) If $E$ and $F$ are any two formices of the same grade, then

$$\text{nex}(F) \mid \text{con}(F) \mid E = \emptyset,$$
$$\text{con}(F) \mid \text{nex}(F) \mid E = \emptyset,$$
$$\text{lux}(F) \mid \text{col}(F) \mid E = \emptyset$$

and

$$\text{col}(F) \mid \text{lux}(F) \mid E = \emptyset.$$

(6) If $k$ is a nonzero positive integer, then

$$\text{cord}(F)^k \mid E = \text{cord}(F) \mid E.$$
2.9.2.3 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

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

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

For example, consider the formex

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

and let \( P \) be specified as follows

\( P \) is true provided that the first uniple of every signet is equal to the second uniple of that signet and \( P \) is false otherwise.

The relection of \( F \) with respect to \( P \) is found to be

\[
\text{rel}(P)|_{F} = [[1,1; 2,2], [3,3; 5,5]].
\]
A condition of the type used in conjunction with reflection function is referred to as a 'predicant'. In general, a predicant is defined as a Boolean function which has one or more formices as arguments. Thus, the canonical variable of a reflection function is a Boolean entity.

A reflection function has no inverse.

The reflection functions are the most general of all introflection functions. Predicants for reflection functions may be described in a mixture of mathematical formulae and statements in a natural language. However, in some cases, predicants may be written in a very convenient notation which is discussed in the next Section.

2.9.2.4 BREVIC NOTATION

There is a shorthand notation that may be conveniently employed in writing certain simple types of commonly used predicants. Namely, those predicants that are expressible in terms of the uniples of a single manipule or a pair of manipules. The notation is referred to as the 'brevic' notation and is described in the sequel.

Let $M_a$ and $M_b$ be two manipules which may or may not be of the same plexitude and of the same grade. With reference to these manipules, the symbols that constitute the brevic notation together with their meanings are given in Table 2.5.

The following simplifications are allowed:

(1) If $j$ is an integer variable consisting of a single letter or if $j$ is a single digit integer number, then $EU(j)$, $EW(j)$, $AU(j)$ and $AW(j)$ may be written as $EUj$, $EWj$, $AUj$ and $AWj$, respectively. For example,
Table 2.5

<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>

EU(3)

that is, 3rd uniple of every signet of Ma, may be written as

EU3

and

AW(r)

that is, rth uniple of any signet of Mb, may be written as

AWr

but

Au(25)

that is, 25th uniple of any signet of Ma, and

EW(n-8)
that is, \((n-8)\)th uniple of every signet of \(Mb\), cannot be simplified.

(2) If \(i\) and \(j\) are single letter integer variables or single digit integer numbers, then \(U(i,j)\) and \(W(i,j)\) may be written as \(U_{ij}\) and \(W_{ij}\), respectively. For example,

\[U(3,2)\]

that is, the 2nd uniple of the 3rd signet may be written as

\[U_{32}\].

(3) When dealing with maniples of the first plexitude, then \(U(1,j)\) and \(W(1,j)\) may be written as \(U(j)\) and \(W(j)\), respectively. Furthermore, if \(j\) is a single letter integer variable or a single digit integer number, then \(U(j)\) and \(W(j)\) may be written as \(U_j\) and \(W_j\), respectively.

(4) When dealing with maniples of the first grade, then \(U(i,1)\) and \(W(i,1)\) may be written as \(U(i)\) and \(W(i)\), respectively. Furthermore, if \(i\) is a single letter integer variable or a single digit integer number, then \(U(i)\) and \(W(i)\) may be written as \(U_i\) and \(W_i\), respectively.

(5) When dealing with maniples of the first plexitude and grade, then \(U(1,1)\) and \(W(1,1)\) may be written as \(U\) and \(W\), respectively.

The above description of the brevic notation is a general one, covering the case when a perdicant is expressed in terms of the uniples of a single maniple as well as the case when a perdicant is given in terms of the uniples of a
pair of maniples.

As an example consider the configuration of Fig 2.37, which is a finite element mesh for a flat plate. Let it be required to write a formex formulation for the interconnection pattern of this configuration in terms of the indicated pronorm. Taking a superpansive approach, let

\[
G = [0,0; 1,0; 0,1], \\
G_1 = G \# \text{refid}(1/2,1/2); G
\]

then

\[
F_1 = \text{rinid}(10,10,2,2) \# \text{rosid}(1,1) \# G_1
\]

where \(F_1\) represents a configuration whose \(N\)-plot is given in Fig 2.38. The superfluous elements may be deleted by a reflection function. Thus, one may write

\[
F_2 = \text{rel}((\text{EU1} \leq 5 \text{ OR EU1} \geq 10 \text{ OR EU2} \leq 4 \text{ OR EU2} \geq 11) \text{ AND} \\
(\text{EU1} \leq 2 \text{ OR EU1} \geq 5 \text{ OR EU2} \leq 15 \text{ OR EU2} \geq 18) \text{ AND} \\
(\text{EU1} \leq 15 \text{ OR EU1} \geq 18 \text{ OR EU2} \leq 4 \text{ OR EU2} \geq 7))
\]

A paribifect \(N\)-plot of \(F_2\) would give rise to a configuration identical to the one shown in Fig 2.37.

2.9.3 NODE NUMBERING

In analysing a structural system by a computer, it is necessary to prepare a complete description of the system providing information about the interconnection pattern, geometric particulars, material properties, external loads and support conditions.

The task of providing the data for a small structure is simple and straightforward. It involves inputting known facts. However, for large systems, the amount of information to be input makes the whole process time
Fig 2.37

Fig 2.38
consuming and error prone. Formex algebra may be used in various ways to overcome these problems.

The objective of this Section is to show that a formex describing the disposition of the elements, loads or supports may be turned into a formex which is relative to an implied sequence of node numbers. This eliminates the need for explicit node numbering and provides a suitable way of data preparation.

2.9.3.1 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

\[ F = \{[1,2; 2,2], [2,2; 2,2], [1,1], [3,4; 3,3]\}, \]
\[ T_1 = \{[1,2], [2,2], [1,1], [3,4], [3,3]\} \]

and

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

then both T1 and T2 are catenas of F.

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 example T1 in the above example is an exclusive catena of F.

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 F.
If T is a catena of E then so is every sequation of T. Also, if T is an exclusive catena of E then so is every sequation of T and if T is an inclusive catena of E then so is every sequation of T.

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.

2.9.3.2 DICTUM FUNCTION

Consider a formex E and let T be a catena of E. Let a formex G be obtained from E by replacing every signet S of E by the orderate, with respect to T, of the first occurrence of S in T.

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

\[ \text{dic}(T) \]

and is referred to as a 'dictum function'. The formex G is referred to as the dictum of E with respect to T and the relation between E and G is written as

\[ G = \text{dic}(T) \upharpoonright E. \]

For example, let

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

and

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

Now, if every signet of E is replaced by its orderate with respect to T, the result is a formex of the first grade which is the dictum of E with respect to T and is given by
\[
\text{dic}(T)\vert_E = \{[1; 2], [3; 2; 4], [1; 4]\}.
\]

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

\[
\text{dic}(T)\vert[] = [].
\]

The practical application of dictum function is best illustrated through an example:

Consider a finite element mesh, an N-plot of which is shown in Fig 2.39 and let a formex \(F\) representing this interconnection pattern be written as

\[
F = F_1 \# F_2
\]

where

\[
F_1 = \text{rinid}(8,2,1,1)\vert G,
\]
\[
F_2 = \text{rinid}(4,3,1,1)\vert \text{tranid}(2,2)\vert G
\]

and

\[
G = [0,0; 1,0; 1,1; 0,1].
\]

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

\[
T = \text{rinid}(9,3,1,1)\vert[0,0] \# \text{rinid}(5,3,1,1)\vert[2,3].
\]

The orderate of every signet of \(T\) is written near the node represented by it in Fig 2.39.

Now, let the dictum of \(F\) with respect to \(T\) be denoted by \(H\). That is,

\[
H = \text{dic}(T)\vert F.
\]
The Formex H is found to be of the form

\[
[[1; 2; 11; 10], [2; 3; 12; 11], [3; 4; 13; 12], \ldots, [34; 35; 40; 39], [35; 36; 42; 41]].
\]

The sequence of the cantles of H provides a complete description of the interconnection pattern of the finite element mesh in terms of the node numbering scheme of Fig 2.39, where each cantle of H represents one of the elements of the mesh. This node numbering scheme is not affected by the order of appearance of the cantles in F but it is dictated by the disposition of the signets in T.

2.9.3.3 REDICTUM FUNCTION

Let E be a formex of the first grade and let T be an ingot of any grade and of order \( r \). Also, let a formex \( F \) be obtained from \( E \) by deleting every cantle of \( E \) that contains a uniple \( U \) such that either \( U \leq \emptyset \) or \( U > r \). Let a formex \( G \) be obtained from \( F \) by replacing every uniple \( U \) of \( F \) by the signet of \( T \) whose orderate is equal to \( U \).

The rule through which \( G \) is obtained from \( E \) is symbolized in terms of a function. This function is denoted by

\[ \text{red}(T) \]

and is referred to as a 'redictum function'. The formex \( G \) is referred to as the redictum of \( E \) with respect to \( T \) and the relation between \( E \) and \( G \) is written as

\[ G = \text{red}(T)\langle E. \]

The redictum of the empty formex with respect to any ingot is the empty formex. That is, if \( E \) is a formex of the first grade and \( T \) is any ingot, then
Fig 2.39

Fig 2.40
\text{red}(T)\mid \emptyset = \emptyset.

In general, neither a dictum function nor a redictum function has an inverse. However, if \( E \) is a formex and \( T \) is a catena of \( E \) and if

\[ G = \text{dic}(T)\mid E \]

then

\[ E = \text{red}(T)\mid G. \]

2.9.3.4 SEVIATION AND LATITUDE FUNCTIONS

Let \( E \) be a formex of the first grade and let

\[ [U_1; U_2; \ldots; U_m] \]

be a typical cantle of \( E \). Let the difference between all possible pairs of uniples

\[ U_1, U_2, \ldots, U_m \]

be compared and the greatest of these differences be denoted by \( \delta \), where if \( m=1 \) then \( \delta \) is considered to be equal to zero. Finally, let \( \delta \)'s for all the cantles of \( E \) be compared and the greatest of these be denoted by \( \Delta \).

The rule through which the integer \( \Delta \) for a formex of the first grade \( E \) is obtained is symbolized in terms of a function. This function is denoted by

\text{sev}

and is referred to as the 'seviation function'. The integer is referred to as the seviation of \( E \) and the relation between \( E \) and \( \Delta \) is written as
\[ \Delta = \text{sev}' \mathcal{E}. \]

The seviation function has no inverse. The seviation of the empty formex is considered to be zero. That is,

\[ \text{sev}' \emptyset = \emptyset. \]

For example, if

\[ \mathcal{E} = \{[12; 14; 17], [10; 5], [11; 21], [1]\} \]

then \( \delta \)'s for the cattles of \( \mathcal{E} \) are 5, 5, 10 and 0, respectively, and

\[ \text{sev}' \mathcal{E} = 10. \]

Let \( \mathcal{E} \) be a formex and \( \mathcal{T} \) be a catena of \( \mathcal{E} \). Also, let

\[ \Delta = \text{sev}' \text{dic}(\mathcal{T}) \mathcal{E}. \]

The rule through which \( \Delta \) is obtained from \( \mathcal{E} \) and \( \mathcal{T} \) is represented by a function. This function is denoted by

\[ \text{lat}(\mathcal{T}) \]

and is referred to as a 'latitude function'. The integer \( \Delta \) is referred to as the latitude of \( \mathcal{E} \) with respect to \( \mathcal{T} \) and the relation between \( \Delta \) and \( \mathcal{E} \) is written as

\[ \Delta = \text{lat}(\mathcal{T}) \mathcal{E}. \]

A latitude function has no inverse.

The latitude of the empty formex with respect to any ingot is zero. That is, for any ingot \( \mathcal{T} \).
\[ \text{lat}(T)\{\} = \emptyset. \]

The practical significance of the latitude function can be illustrated in terms of the example of the previous Section. One will find that

\[ \text{lat}(T)\{F = 10. \]

This number is, in fact, the greatest difference between the node numbers for an element in the configuration of Fig 2.39. This in fact is a measure of the band width of the stiffness matrix of the structure under consideration.

2.9.3.5 NOVATION FUNCTIONS

Consider a formex E and let F be a 2-plex formex of the same grade as E. Let the first cantle of F be represented by

\[ [S_1; S_2] \]

where \( S_1 \) and \( S_2 \) are the first and the second signets of the cantle, respectively. Let E be modified by replacing every signet of it which 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 represented by a function. This function is denoted by

\[ \text{nov}(F) \]

and is referred to as a 'novation function'. The formex G is referred to as the novation of E with respect to F and the relation between E and G is written as
\[ G = \text{nov}(F)_E. \]

The novation of the empty formex with respect to any 2-plex formex is considered to be the empty formex itself. Also, the novation of any formex \( E \) with respect to the empty formex is considered to be the formex \( E \) itself. A novation function, in general, has no inverse.

As an example, let a formex \( F \) be given as

\[ F = \text{rinid}(10,20,1,1)_G \]

where

\[ G = [5,0; 6,0; 6,1; 5,1]. \]

\( N \)-plots of \( F \) with respect to a basipolar retronorm with \( b_1=1.2 \) and \( b_2=18 \) and a paribifect retronorm are shown in Figs 2.40 and 2.41, respectively.

Let it be required to use a dictum function and transform \( F \) into a formex that describes the interconnection pattern of the finite element mesh of Fig 2.40 in terms of a node numbering system. In attempting to achieve this end, one would encounter problem. Namely, one finds that both

\[ 5,0 \]

and

\[ 5,20 \]

in \( F \) represent the nodal point which is shown encircled in Fig 2.40. As a result one would not be able to associate a unique number with this node through a dictum function. Examination of formex \( F \) reveals that in the cantle relating to the element indicated by \( A \) in Fig 2.40, the encircled node is represented by the signet

\[ 5,0 \]
and in the cantle relating to the element indicated by B in Fig 2.40, the encircled node is represented by the signet [5,20].

A similar problem arises in relation to the other nodes that are situated on the normat line for $U_2=0$ and $U_2=20$ as specified in Fig 2.42. This implies that as far as the above formex F is concerned, there are no connection between the elements above and the elements below the nodal point under consideration. To signify this fact these nodal points are shown with gaps in Fig 2.42.

From a purely graphical point of view, one may regard F as representing the interconnection pattern of the finite element mesh of Fig 2.40. However, if F is to serve as data describing the interconnection pattern of the mesh for the purpose of structural analysis, then one must modify F to take account of the connections between the elements of the above mentioned nodes correctly. Modification of this type may be achieved through a novation function. Thus, one may write

$$E = \text{nov}(H) \mid F$$

where

$$H = \text{rin}(1,11,1) \mid [5,20; 5,0].$$

The formex E provides a description for the interconnection pattern of the mesh of Fig 2.40 and is free from any problem regarding discontinuities at nodes.
2.9.4 FORMEX COLLATION

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. Also, it may be required to rearrange the cantles of \( F \) to suit the frontal solution technique.

Operations of this type may be performed using the concept of rapportance and the objective of the present Section is to explore the parts of formex algebra that relate to this concept.

2.9.4.1 CONCEPT OF RAPPORTANCE

Let \( E \) and \( F \) be two formices and let \( P \) be a perdictant that may be evaluated with respect to \( E \) and \( F \). Let the perdictant \( P \) be evaluated twice with respect to \( E \) and \( F \), where \( E \) and \( F \) are taken once in the order \( E,F \) and then in the order \( F,E \). Four possibilities arise and these are shown in Table 2.6.

In the first case

- \( E \) is said to be of a 'higher rapportance' than \( F \) with respect to \( P \)
- \( E \) is said to be 'more rapportant' than \( F \) with respect to \( P \).

In the second case
Table 2.6

<table>
<thead>
<tr>
<th>Case</th>
<th>Values of P with respect to E and F in the order of E,F</th>
<th>Values of P with respect to E and F in the order of E,F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>2</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>3</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>4</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

E is said to be of a 'lower rapportance' than F with respect to P

or

E is said to be 'less rapportant' than F with respect to P.

In the third or forth case

E and F are said to be of the 'same rapportance' with respect to P

or

E and F are said to be 'equally rapportant' with respect to P.

Various relationships between E and F may be symbolically represented as shown in Table 2.7.

2.9.4.2 PROCESS OF RAPPORTATION

Consider a sequence S of n formices where n>2 and let there be a perdicant P which may be evaluated with respect to any two elements of the sequence.
Table 2.7

<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 a 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>

Let the phrase '(i,j) forward seriation' be used to refer to the following procedure

For k=i, i+1, i+2, ..., j-1
the kth and (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 '(j,i) backward seriation' be used to refer to the following procedure
For \( k = j, j-1, j-2, \ldots, i+1 \)
the \( k \)th and \((k-l)\)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.

In relation to both forward and backward seriations, it is
understood that when reference is made to an element of \( S \),
say the \( k \)th element, then it is meant the element which is
in the \( k \)th position of the sequence at the time of the
reference rather than the element which was in the \( k \)th
position of the sequence initially.

The term 'seriation' may be used to refer to either a
forward seriation or a backward seriation.

Let the sequence \( S \) be subjected to the following process

For \( k = 1, 2, \ldots \)
a \((k, n-k+1)\) forward seriation is carried out and this
is 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'. The sequence \( S \) may be said to have been
subjected to the process of rapportation with respect to \( P \)
or the sequence may be said to have been rapported with
respect to \( P \).

For example consider the sequence of uniples

\[ 2, 3, 5, 4, 1, 3, 6, 7 \]

and let a perdicant \( P \), in brevic notation, be given by
To rapport the above sequence with respect to \( P \), one begins by applying a forward seriation involving all the elements of the sequence, with the process being from left to right. This seriation may be referred to as a \((1,8)\) forward seriation. The resulting sequence will be

\[ 2,3,4,1,3,5,6,7 \]

As the next step, an \((7,1)\) backward seriation is applied. This seriation will involve the elements 1 to 7 of the sequence, with the process being from right to left. The result is the sequence

\[ 1,2,3,4,3,5,6,7 \]

The next seriation to be applied is a \((2,7)\) forward seriation involving the elements 2 to 8 of the sequence. This will transform the sequence into

\[ 1,2,3,3,4,5,6,7 \]

Applying a \((6,2)\) backward seriation leaves the sequence unchanged which means that the process of rapportation is completed.

With respect to a sequence of formices \( S \), a perdicant \( P \) is said to be 'transitive' provided that if \( S \) is transformed into a sequence \( R \) by the process of rapportation with respect to \( P \) and if \( R_i \) is the \( i \)th element of \( R \) and \( R_j \) is the \( j \)th element of \( R \), with \( i \) being less than \( j \), then for all \( i \)'s and \( j \)'s

\[ R_i [\triangleright P] R_j \]

If the above condition is not satisfied then the perdicant
P is said to be 'nontransitive' with respect to S.

2.9.4.3 RAPPORTED SEQUATION FUNCTIONS

Consider a 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 let 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

\[ G = \text{ras}(P) \rvert E. \]

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

As an example, consider

\[ F = \{[9,3], [3,3], [5,8], [9,5], [8,6]\} \]

and let P in brevic notation be given by

\[ U1 < W1. \]

Then the rapported sequation of F with respect to P is found to be
\[ \text{ras}(P) \mid F = \{[3,3], [5,8], [8,6], [9,3], [9,5]\}. \]

### 2.9.4.4 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 a '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

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

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

For example, consider formex \( F \)

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

and let \( P \) in brevic notation be given as

\[ (U_1 > W_1) \ OR \ (U_1 = W_2 \ AND \ U_2 > W_2). \]

Then the rapported variant of \( F \) with respect to \( P \) is found to be
ras(P)\{F = \{[2,2; 2,1; 1,2; 1,1], [3,1; 2,2; 1,1]\}.

2.9.5 FURTHER FORMEX FUNCTIONS

In this Section a number of formex functions are defined. These are used to enhance the capabilities in dealing with formex formulation.

2.9.5.1 TECTRIX FUNCTIONS

Let \(E\) be a formex and suppose that a sequence of reglets

\[R_1, R_2, \ldots, R_t\]

is constructed from all the signets of \(E\), where the order of appearance of the reglets in \(S\) is exactly the same as in \(E\). Thus, if

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

then the sequence \(S\) will be

\[[1,1], [2,3], [3,3], [2,2], [3,3], [1,2], [1,1], [4,4].\]

Consider an \(m\)-plex formex \(F\) a typical cantle of which consists of the \(m\)-tuple of reglets

\[R_i, R_j, R_k, \ldots, R_p, R_q\]

where \(R_i, R_j, R_k, \ldots, R_p\) and \(R_q\) are reglets from \(S\) with subscripts indicating serial position numbers and with

\[i < j < k < \ldots < p < q\]
and where

\[ i \text{ varies from } 1 \text{ to } t-m+1 \]

and for every value of \( i \):

\[ j \text{ varies from } i+1 \text{ to } t-m+2 \]

and for every value of \( j \):

\[ k \text{ varies from } j+1 \text{ to } t-m+3 \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

and for every value of \( p \):

\[ q \text{ varies from } p+1 \text{ to } t \]

and where the order of appearance of the cantles in \( F \) is governed by the rule that for every two adjacent subscripts, say \( j \) and \( k \), all the integers in the range of \( k \) are employed in the natural ascending order, before \( j \) is incremented by unity. Thus, if \( S \) is given by

\[ [3,4], [6,1], [8,2], [4,3], [3,4] \]

and if \( m=2 \) then \( F \) is given by

\[
[[3,4; 6,1], [3,4; 8,2], [3,4; 4,3], [3,4; 3,4],
[6,1; 8,2], [6,1; 4,3], [6,1; 3,4],
[8,2; 4,3], [5,2; 3,4], [4,3; 3,4]].
\]

Also, with the above sequence \( S \), if \( m=1 \) then \( F \) is given by

\[
[[3,4], [6,1], [8,2], [4,3], [3,4]].
\]

Now, consider a perdicant \( P \) which may be evaluated for every cantle of \( F \) and obtain a formex \( G \) as

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

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

\[ \text{tec}(m, P) \]
and is referred to as a 'tectrix function'. The formex $G$ is referred to as the tectrix of $E$ with respect to $m$ and $P$ and the relation between $E$ and $G$ is written as

$$G = \text{tec}(m, P)|_{E}.$$  

Any tectrix function of the empty formex is considered to be the empty formex itself. Also, if the total number of signets in a formex $E$ is less than $m$, then

$$\text{tec}(m, P)|_{E}$$

is considered to be the empty formex. A tectrix function has no inverse.

2.9.5.2 VINCULUM FUNCTIONS

Let $E$ be a formex of the $n$th grade and let a formex $G$ be obtained as

$$G = \text{tec}(2, P)|_{E}$$

where $P$ is given by

$$B_1 \leq M \leq B_2$$

and where $B_1$ and $B_2$ are real numbers and are referred to as a 'lower bound' and 'upper bound', respectively, and $M$ in brevic notation is given by

$$M = |(\sum_{i=1}^{n} (U_{2i} - U_{1i})^2)^{1/2}|.$$  

The uniples in the above relation belong to a typical cantle $C$ of $E$ and $M$ is referred to as the 'metrum' of $C$.  

108
The rule by which $E$ is transformed into $G$ is symbolized in terms of a function. This function is denoted by

$$\text{vin}(r_1, r_2)$$

and is referred to as a 'vinculum function', where $r_1$ and $r_2$ are real numbers and if $r_1 \leq r_2$ then $r_1$ and $r_2$ are interpreted as being the lower and upper bounds $B_1$ and $B_2$, respectively, and if $r_1 > r_2$ then $r_1$ is interpreted as being the upper bound $B_2$ and $r_2$ is interpreted as being the lower bound $B_1$. The formex $G$ is referred to as the vinculum of $E$ with respect to $r_1$ and $r_2$ and the relation between $E$ and $G$ is written as

$$G = \text{vin}(r_1, r_2) \mid E.$$ 

Any vinculum of the empty formex is considered to be the empty formex itself. Also, if $E$ is a reglet, then

$$\text{vin}(r_1, r_2) \mid E$$

is considered to be the empty formex. A vinculum function has no inverse.

2.9.5.3 PANSION FUNCTIONS

Let $E$ be a formex of the $n$th grade and let $q$ be any integer. Also, let $h$ be an integer such that

$$1 \leq h \leq n+1.$$ 

Let a formex $G$ be obtained from $E$ by replacing every signet

$$[U_1, U_2, \ldots, U_n]$$

of $E$ by a reglet $R$ of the $(n+1)$th grade, where
if \( h=1 \) then \( R = [q, U_1, U_2, \ldots, U_n] \)
and if \( h=2 \) then \( R = [U_1, q, U_2, \ldots, U_n] \)
and if \( h=3 \) then \( R = [U_1, U_2, q, \ldots, U_n] \)

and if \( h=n \) then \( R = [U_1, U_2, \ldots, q, U_n] \)
and if \( h=n+1 \) then \( R = [U_1, U_2, \ldots, U_n, q] \).

The rule through which \( G \) is obtained from \( E \) is represented by a function. This function is denoted by

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

and is referred to as a 'pansion function'. The formex \( G \) is referred to as a pansion of \( E \) and the relation between \( E \) and \( G \) is written as

\[ G = \text{pan}(h, q)\downarrow E. \]

Any pansion of the empty formex is the empty formex. Every pansion function has an inverse, where if

\[ G = \text{pan}(h, q)\downarrow E \]
then

\[ E = \text{pan}(h, q)^{-1}\downarrow G. \]

For example, if

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

\[ \text{pan}(1,8)\downarrow F = \{[8,4,4; 4,3], [8,2,3], [8,2,1; 8,1,2; 8,3,1]\} \]
and

\[ \text{pan}(3,8)\downarrow F = \{[4,4,8; 4,3,8], [2,3,8], [2,1,8; 1,2,8; 3,1,8]\}. \]
2.9.5.4 DEPANSION FUNCTIONS

Let $E$ be a formex of the $n$th grade and let $h$ be an integer such that

$$1 \leq h \leq n.$$  

Let a formex $G$ be obtained from $E$ by removing the $n$th uniple of every signet of $E$. The rule through which $G$ is obtained from $E$ is represented by a function. This function is denoted by

$$\text{dep}(h)$$

and is referred to as a 'depansion function'. The formex $G$ is referred to as a depansion of $E$ and the relation between $E$ and $G$ is written as

$$G = \text{dep}(h) \upharpoonright E.$$  

Any depansion of the empty formex is the empty formex itself. Also, if $F$ is a formex of the first grade, then

$$\text{dep}(1) \upharpoonright F$$

is considered to be the empty formex.

A depansion function has no inverse. However, the concept of depansion functions is the converse of that of pansion functions and

$$\text{pan}(h, q)^{-1} \upharpoonright E = \text{dep}(h) \upharpoonright E.$$  

As an example, consider formex $F$

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

then
dep(2) \{ F = \{ [1; 4], [3; 2], [5; 1] \} \}.

2.9.5.5 MEDULLA AND RAPPORTED MEDULLA FUNCTIONS

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. Then let an ingot G be obtained as

\[ G = \text{pex}^\text{?} T. \]

The rule through which G is obtained from E is represented by a function. This function is denoted by

\[ \text{med} \]

and is referred to as the medulla of E and the relation between E and G is written as

\[ G = \text{med}^\text{?} E. \]

For example, if E is given as

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

then T will be

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

and

\[ G = \text{med}^\text{?} E = \{ [2,1], [1,1], [2,2], [3,3], [4,3], [2,3] \}. \]

The medulla of the empty formex is considered to be the empty formex itself. The medulla has no inverse.

Now, Let E be a formex and let an ingot G be obtained as
G = ras(P)\text{med}_{E}.

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

\text{ram}(P)

and is referred to as a 'rapported medulla function'. The ingot G is referred to as the rapported medulla of E with respect to P and the relation between E and G is written as

G = \text{ram}(P)\downarrow E.

Any rapported medulla of the empty formex is considered to be the empty formex itself. A rapported medulla has no inverse.
CHAPTER 3
3 FINITE ELEMENT MESH GENERATION TECHNIQUES

3.1 INTRODUCTION

Although the finite element method may be used for complex engineering systems, the manual modelling of a structural system directly with finite elements has traditionally been a laborious, time consuming and error prone process. Automation of the finite element generation process provides an easier, faster and more accurate method. The method was, however in use successfully long before an appropriate theory and procedure was given. During the last ten years significant progress has been made in the techniques and computational implementation.

Most major advances in finite element mesh generation techniques have resulted in the past from efforts to solve a particular class of problems and for this matter their application are mainly limited. It is appropriate at this stage of development that detail studies be made of the mathematical bases of the methods so that they may be placed on a fundamental foundation. The tools, procedures and guidelines illustrated by various researchers should enable us to construct mathematical procedures tailored for most complex structures.

3.2 AUTOMATIC MESHING TECHNIQUES

During the past thirty years, the development of the finite element methods has been one of the major achievements of engineering researchers. Unfortunately for the analyst in this field, it has in many instances made data preparation a tedious, error prone and time consuming task. In an attempt to overcome this drudgery and to minimize data errors, considerable efforts have been expended in the developing mesh
generation schemes.

Mesh generators are schemes which automatically subdivide a domain into elements of the required type, size and density for the minimum amount of data input by the user. They are expected to be economical with respect to both the computer time and manual effort. No error beyond the discretization error inherent to the chosen finite element model should be introduced by the meshing process.

The efficiency and flexibility of a scheme are heavily dependent on the form of the mathematical procedures used to automate the mesh generation process. Several different approaches have been proposed and implemented by various researchers.

One of the earliest and most commonly employed mesh generation procedures for arbitrary geometries and element layout was the Laplacian scheme. This procedure, apparently originally conceived by Wilson, is described in details in ref[3.1]. Consider the small grid shown in Fig 3.1. Wilson's concept was to locate the n interior nodes, not directly specified by the user such that

\[
X_i = (X_{i1} + X_{i2} + X_{i3} + X_{i4})/4 \quad i = 1,n \quad (3.1)
\]

\[
Y_i = (Y_{i1} + Y_{i2} + Y_{i3} + Y_{i4})/4 \quad i = 1,n \quad (3.2)
\]

Subsequently, it was observed that equations (3.1) and (3.2) can be interpreted as Laplacian finite difference operators for the unknown \( X_i \) and \( Y_i \), thus the name 'Laplacian generation scheme'.

The above equations were found to be limited to meshes that were transformable to rectangular meshes. This problem can easily be remedied by positioning the interior nodes so that the position vector, \( P_i \) of each interior node \( i \) satisfies the following equation
Fig 3.1

Fig 3.2 - Neighbourhood of Node i
a) portion of none rectangular grid
b) detailed view of element n
\[ \Pi_i = \frac{1}{2} \sum_{n=1}^{N_i} (P_{nj} + P_{nm}) \]  \hspace{1cm} (3.3)

where, \( N_i \) is the number of elements framing into node \( i \); \( P_{nj} \) and \( P_{nm} \) are the position vectors of adjacent nodes in the neighbouring element \( n \) as shown in Fig. 3.2. Meshes produced by this technique will tend to have fairly uniformly shaped elements [3.5]. A set of equations with \( \Pi_i \) as unknown is then made up. Solution of these non-linear simultaneous equations is best obtained via an iterative technique such as Gauss-Siedel or Jacobi iteration, which represents a relatively large computation for mesh generation. The other undesirable feature of the Laplacian generation scheme is far more serious, i.e., the Laplacian scheme is not sufficiently sensitive to information contained in the specified boundary node locations concerning mesh spacing and boundary curvature (see Fig. 3.3).

To overcome these drawbacks Herrmann [3.2], investigated a number of alternative generation schemes; and developed the Laplacian-isoparametric scheme. Herrmann's equation represents the position vector for the origin of the parametric coordinates system in an eight-node isoparametric finite element

\[ \Pi_i = \left( \frac{1}{N_i (2-w)} \right) \sum_{n=1}^{N_i} (P_{nj} + P_{nm} - w P_{nk}) \]  \hspace{1cm} (3.4)

Different values of \( w \) produce a family of schemes called Laplacian-isoparametric schemes. A value of \( w = 1 \) yields the 'pure-isoparametric' Laplacian scheme, and \( w = 0 \) yields the Laplacian scheme. The pure isoparametric method produces well conditioned mesh in which the interior adequately reflects the boundary geometry. Intermediates value of \( w \) yield mixture of the two methods. Herrmann [3.2] reported that as \( w \) tends to its optimum value of 1, the number of iterations to convergence increases greatly.
Fig 3.3 - illustration of undesirable feature of laplacian mesh
(a)-location of boundary; (b)-desired meshes; (c)-laplacian meshes

logical grid

voids

zone diagram

Fig 3.4
Denayer [3.3] also presented a method for satisfying equation (3.3) without resorting to iterative solution methods. The method involves a mapping between an imaginary domain defined by an idealized mesh composed entirely of regular polygons and the actual domain to be meshed. The idealized mesh is constructed from boundary curve information, including the number of elements connecting to each boundary node. Once the mesh topology is created, the location of the interior nodes may be selected so that equation (3.3) is satisfied. The mapping between the idealized mesh and the true mesh is first assumed to be of the form

$$P = \sum_{m=1}^{M} P_m \, \varnothing_m (u, v)$$  \hspace{1cm} (3.5)

where $P_m$'s are now the known position vectors of $M$ boundary nodes, $\varnothing_m$'s are assumed shape functions over the region to be meshed, and $u$ and $v$ are the parametric coordinates of the idealized mesh. If the mapping from the idealized mesh to the real mesh is harmonic, equation (3.3) will be satisfied. This is obtained if the function

$$W = \int_{\Omega} \left[ (dp/du)^2 + (dp/dv)^2 \right] \, d\Omega$$  \hspace{1cm} (3.6)

is minimized, where $\Omega$ is the domain to be meshed. A finite element procedure based on the representation of equation (3.5) is used to extremize equation (3.6). This leads to direct matrix equations for the unknown nodal coordinates.

A natural outgrowth of the use of isoparametric mappings to represent curved finite elements is the use of the same mapping to mesh the entire region of a structure. Zienkiewicz and Phillips [3.4] have shown how to construct curvilinear coordinates, whose constant coordinate lines induce a natural rectilinear parametrization of domains, whose boundaries are defined as low-order parametric polynomials. The essence of their scheme is the use of...
'isoparametric' mappings to generate generalized coordinate systems on quadrilateral zones. The scheme generates meshes of triangular elements in plane and curved surfaces. Depending on geometric and material variations, the domain to be meshed is divided into a number of four-sided zones. Each zone is identified by the coordinates of four up to eight points on the boundary of the zone. Then a logical mesh is constructed each of whose elements corresponds to a zone in the actual domain (the elements of the logical mesh specify the relative position of zones with respect to one another). Then having decided on the number of divisions, automatically, each element of the logical mesh is subdivided into regularly spaced meshes which are then mapped into actual domain, through the interpolation functions.

Figure 3.4 illustrates an example of the subdivision of a region with four-sided zones and the corresponding logical grid. The specified points are indicated by dots.

Gordon and Hall extended the work of Zienkiewicz and co-workers. The main drawback of the mapping and generation suggested by Zienkiewicz was the fact that the original surface or volume boundaries are approximated by simple parabolas and a geometric error can be developed there. To overcome this difficulty another form of mapping, originally developed by Coons [3.6], for representation of complex motor car body shapes, was adopted by Gordon and Hall [3.7]. They use the term 'transfinite' to describe the general class of interpolation schemes. The general 'transfinite' method describes an approximate surface or volume which will match a desired, or true, surface or volume at a non-denumerable number of points. It is this property which gave rise to the term 'transfinite mapping'. This property contrasts with the isoparametric mappings which match the true surface at only a finite number of points, i.e. the points used for interpolation.
To describe the transfinite mapping it is necessary to introduce the concept of a projector $P$. The discussion here will be limited to mapping describing three-dimensional surfaces.

A projector is any idempotent linear operator which maps a true surface to an approximate surface, subject to certain interpolatory constraints. The surface is represented by a vector valued function $f$. The approximate surface is represented by a second vector valued function $U$. The interpolatory constraints are such that the true surface and the approximate surface will match exactly along certain preselected curves. Thus the function $U$ is defined by a specific projector operating on $f$.

The simplest projectors are the unidirectional projectors which interpolate in one direction only. These projectors form the basis for more complex schemes which blend interpolation in more than one direction.

Consider four curves shown in Fig 3.5 and let it be required to define a surface between them. Using local coordinates $r,s$, the final surface is denoted by $f(r,s)$ and consequently the boundary curves are written as $f(0,s)$, $f(1,s)$, $f(r,0)$ and $f(r,1)$. Note that, it is required that each pair of opposite boundary curves to be identically parametrized.

Now, two unidirectional operators $P_1$, $P_2$ are defined by

\[
P_1f = f(0,s)\ L_{10}(r) + f(1,s)\ L_{11}(r)
\]
\[
P_2f = f(r,0)\ L_{10}(s) + f(r,1)\ L_{11}(s)
\]

where $L_{li}$ are linear Lagrange polynomials. Their defining property is

\[
L_{ni}(uk) = \begin{cases} 
1, & i = k \\
0, & i \neq k.
\end{cases}
\]
Fig 3.5 Surface Definition

Fig 3.6 Bilinearly blended Coon's patch Boolean sum
Both $P_1$ and $P_2$ interpolate two opposite boundary curves; the resulting surfaces $P_1f$ and $P_2f$ are ruled surfaces, while $P_1(P_2f)$ is a doubly ruled surface (see Fig. 3.6). Coons superposed these three surfaces by forming the 'Boolean sum' 

$$U = Pf = (P_1 \ominus P_2)f = (P_1 + P_2 - P_1P_2)f$$

of $P_1$ and $P_2$. It is easy to check that $Pf$ does indeed interpolate to all four boundary curves. Note that $P_1P_2f=P_2P_1f$ if four boundary curves are compatible, i.e. they intersect at the four corners.

The main drawback of the bilinearly blended Coons patch $Pf$ is the fact that in general two adjacent patches will be $C^0$ only, even if their boundary curves form a $C^1$ network. This flaw is to be blamed on the 'blending functions' $L_{li}$. Coons realized that the cubic Hermite polynomials $H_{30}$ and $H_{33}$ could be used as blending functions as well [3.6].

Gordon surfaces are a straight generalization of the Coons technique: Suppose one is given a network of compatible (i.e. mutually intersecting) curves

$$f(u_i, v), \ i=0,1,\ldots,n; \quad f(u, v_k), \ k=0,1,\ldots,m$$

as shown in Fig 3.7.

A surface is to interpolate to all curves. With univariate Lagrange polynomials $L_{ni}(u)$ and $J_{mk}(v)$ two unidirectional projectors may be defined as follows

$$P_1f = \sum_{i=0}^{n} f(u_i, v) L_{ni}(u)$$
$$P_2f = \sum_{k=0}^{m} f(u, v_k) J_{mk}(v).$$

Both $P_1$ and $P_2$ interpolate to one family of given curves, and
Fig 3.7 Data for Gordon's surface

Fig 3.8 Boundary of a triangular patch rst-plane
their Boolean sum

\[ U = (P_1 \oplus P_2) = P_1f + P_2f - P_1P_2f \]

interpolate to the whole network. Note that for \( m=n=1 \) one obtains the bilinearly blended coons patch.

The intrinsic curvilinear co-ordinate system used in the transfinite mapping provides a basis for the automatic generation of the element topology within a region. One distinct advantage of the transfinite mapping is that the generated surface description will match the entire boundary of the region exactly without placing any restriction on the form of the boundary curves.

The key idea behind the Coons-Gordon surfaces is that a bivariate interpolant can be constructed from an adequate superposition of univariate interpolation schemes. This idea was carried over to triangular patches by Barnhill, Birkoff and Gordon [3.20]. The analogoue to the bilinearly blended Coons patch is developed as follows: suppose that three boundary curves \( f(0,s,t), f(r,0,t) \) and \( f(r,s,0) \) of a triangular patch are given as shown in Fig 3.8, where \( r,s,t \) are local coordinates. Let an operator \( P_1f \) be defined as follows

\[ P_1f = f(r,0,t) L_{10}(\alpha ) + f(r,s,0) L_{11}(\alpha ) \]

where \( \alpha = s/(s+t) \) is a new local parameter of the line \( r=\text{constant} \). Note that the blending functions \( L_{ii}(\alpha ) \) are also linear in \( s \) because \( s+t=1-r=\text{constant} \). The operators \( P_2f \) and \( P_3f \) are defined analogously. The Boolean sum of any two of these operators interpolates to all three boundary curves, e.g.

\[ P_{12}f = (P_1 \oplus P_2)f. \]
The discussed scheme is only $C^0$; two adjacent triangles will not join smoothly. If one has derivative information prescribed around a triangular patch, it is possible to formulate $C^1$ transfinite interpolants: e.g. a cubic Hermite operator may be defined as

\[
Q_1f = f(r,\theta,t) H_30(\alpha) + D_\alpha f(r,\theta,t) H_31(\alpha)
\]

\[
D_\alpha f(r,s,\theta) H_32(\alpha) + f(r,s,\theta) H_33(\alpha)
\]

where $D_\alpha$ denotes derivatives with respect to $\alpha=s/(1-r)$. Similarly one defines $Q_2$ and $Q_3$. The Boolean sum of any two distinct $Q_i$ interpolates to the whole boundary information.

Cook [3.8] describes surface or volume nodal point generators using 'edge functions' in terms of local coordinates. These edge functions describe the way nodes are placed along the main boundary. Cook uses the boundary nodal positions to distribute the internal nodal points of a 2-Dimensional surface or domain. Figure 3.9 shows a mesh where the positions of internal nodes were calculated with surface generator. Similarly his volume nodal point generator does the same in 3-Dimensions. Cook's 'edge functions' are based upon the linearly blended interpolation formula of Coons [3.6]. Blending function method involves the smooth 'blending' together of information given along curves.

R.Harber et al [3.5] describes a scheme utilizing the transfinite mapping method proposed by Gordon and Hall. Two gross classes of representations for the boundary of a region may be identified: continuous forms and discrete forms.

The continuous forms represent the position vector to a boundary curve as a function of some parametric coordinate. The parameterization may be in any of a number of forms, such as arc length, radians, Cartesian coordinates, etc. The continuous representations allow the curve position to be evaluated at all points along the curve.
The discrete forms of representation consists of finite lists of points located on the curve, with a unique coordinate associated with each point in the list. The curve position can only be evaluated at the points contained in the list and is undefined elsewhere. The discrete form of representation is entirely general and may be used on any curve form.

The continuous forms of representation give a more complete curve definition but are not general. This scheme presents transfinite mapping based on discrete boundary curve descriptions which provides an effective basis for automatic generation of finite element meshes. This method is general and simple to implement. Local mesh modifications may be accomplished using the mesh editor display page. Elements may be deleted from the mesh and new elements may be added and position of an existing node may easily be changed.

The principles of the methods presented here are mostly based on such geometrical principle as the methods of mapping from a regular mesh pattern, dividing, merging or interpolating nodes or manual input by aid of graphic displays and so on. What is advantageous in these methods is the ease of generating the mesh pattern, though less flexible in controlling the shape and the number of elements, since they are considerably affected by the configuration of the boundary curve.

In an early attempt to overcome this inflexibility, Suhara and Fukuda [3.9] tried to determine automatically the location of each node in the given domain of any shape by making use of random numbers and to select the node which satisfy some geometrical conditions to get favourable form for each triangular element. In this method the main domain is divided into a number of regions, each of which is assigned a specified value for the element density. Nodes are then generated either randomly or according to a regular pattern until they fill the domain to the required density.
The information on the shape of the boundary curves of a given domain should be accepted in the form of coordinates of sufficient number of points to describe the given boundary curve or any geometrical information presented in the forms of mathematical expressions by part of the boundary curves.

For node generation a rectangle is constructed around the domain. This rectangle is then subdivided into smaller equal width square of side $r$ as shown in Fig 3.10. For each square zone one point is decided within that square zone by making two uniform random numbers between 0 and 1 and letting these correspond to Cartesian coordinates $(x, y)$ along both side of each square, where $x$ and $y$ are normalized respectively. The generated node is retained if its localised coordinates lies within the domain and if it is more than a prescribed distance $r$ from any internal nodal points already obtained and any nodal points initially designated on the external perimeter. Furthermore, it would not be retained if it lies within a zone running around the boundary of the region to prevent excessively acute-angled triangles being formed when connection are made to nearby boundary nodes.

The process followed so far is repeated several times by letting random number come up again, and if a point lying farther than from any other node is obtained, that is taken as final one in that square zone. In each case when such a point is not obtained even by repeating the process several times, that square zone will be passed on without defining any nodal point, to resume the same process in the next square zone. When the determination of internal nodal points is done on all the square zones, then these are connected one to one to form triangular elements. First being subjected to a series of tests to determine their eligibility and to reject invalid connections. Because of the random positioning of the nodes, it is also necessary to check that the choice of a particular node will not give rise to poorly shaped elements later in the generation procedure (i.e.
elements which are acute-angled to such an extent that the accuracy of the analysis will be significantly reduced). These are avoided by calculating the size length of each potential element and deriving a 'shape' value which indicates how near it is to an equilateral triangle.

Cavendish [3.10] modified Suhara-Fukuda's algorithm, which is a series of procedural steps for the solution of problem of mesh generation, so that the user may specify any fixed interior node points chosen. The suitability criteria are also modified so that a triangle which satisfies these criteria do not lead to generation of poor elements at a later stage in the decomposition. The algorithm was also improved by applying a post-processing smoothing technique to the generated element nodes. Thus each interior node of the triangulation is replaced by the centroid of the polygon composed of those triangles which surround that node.

Shaw [3.11] uses an improved version of Suhara-Fukuda method. In this method more control is exercised when generating the nodes which produces a more evenly balanced mesh. This produces an optimized node positions and the subsequent triangulation into elements can be carried out by simpler method with fewer checks. Thus, poorly shaped elements are avoided not by checking element which will form later in the generation procedure, but by ensuring that the node position cannot give rise to such elements. To do this, nodes are generated along the boundaries in addition to the interior area as shown in Fig 3.11.

Nodes are interpolated along the boundary between two regions, so that, they are spaced at distances compatible with the node densities of these regions. Where two regions of different densities share a common boundary segment, the spacing along this segment should be an intermediate value (i.e. \( r' = \sqrt{r_1 \cdot r_2} \)). The next modification is that it imposes a rectangular zone of \( r \times \sqrt{3} \) (\( r \) is the nodes density). Then
Fig 3.11 Generation of Nodes

Fig 3.12 Node Levels

boundary layer I

boundary layer II

contour I

contour II

contour III

node with level = 1

node with level = 2

node with level = 3
two nodes are generated in each grid zone. In the triangulation stage, the nodes are connected together working inward from the boundaries. Each section of boundary between adjacent nodes is retained as a base for a triangle. Then the choice of the third point is determined by calculating the apex angle at that point which must be maximum for the selected node. Because of the more even distribution of nodes, it is unnecessary to check the consequence of using sides of the triangle under test as a base side in a latter stage of element generation. In this method smoothing is applied, so that each node is moved to centroidal position of the polygon formed by the triangles which meet at that node.

Nelson [3.12] approaches the problem in a similar manner but his suitability criteria is based upon the construction of a circum-circle. The most suitable triangle is one in which its circum-circle when drawn about the given base line has its centre nearest to the base line. The resulting mesh is finally smoothed by an appropriate algorithm.

One approach in automating mesh generation is to use computer to systematically subdivide a coarse mesh which has been prepared by hand. George [3.13] starts with a gross triangulation of the domain and then partitions each subdomain into triangular elements. Points are generated one at a time, working inward from the boundary, until the entire subdomain is covered by triangular elements.

Bykat [3.14] and Schoofs [3.15] have developed schemes that start by automatically partitioning the domain into convex subregions, which are regions with internal angles less than 180°, and then continue by subdividing these subregions into elements. Based on a set of objective criteria, the algorithm of Bykat consists conceptually of two steps. In the first step the region is subdivided into convex subregions; in the second step these convex subregions are further subdivided into triangular elements. Due to the
design of the procedure, the shape of element is governed not only by the shape of the region but also by the node spacing in the boundary of a convex subregion under consideration. The boundary is automatically preconditioned by inserting additional nodes so that the lengths of any two consecutive segments in the boundary do not differ by more than a factor of two. The technique of varying the density of elements to accurately capture the behaviour of the solutions is referred to as local mesh refinement (i.e. grading). An automatic grading facility has been built into the algorithm and can be requested by user to satisfy the geometric features of the domain (i.e. re-entrant corners, etc). Starting with an initial point on the boundary of the domain, a convex subregion is then cut off from the main domain by establishing a barrier. If this subregion contains a node for grading, then grading is produced. Then with a series of suitability criteria, which optimize the shape of the elements, this subregion is triangulated by the best form of triangular elements (i.e. well shaped, near to equilateral). It then returns to the main domain, cut off the next convex subregion and repeats the process.

Sadek [3.16] describes a simple algorithm which can be used to divide an arbitrary domain into reasonably well-shaped triangular (and/or quadrilateral) elements. Given a certain domain with a number of nodes around its boundary, the nodes on original boundary are considered to be of level equal to unity. In the case of a domain with a number of inside holes, the nodes around the boundary of the holes are also considered of level equal unity. Subdivisioning on this level at a corner point, at which the boundary angle $\theta$ shown in Fig 3.12 is not equal to 180, generates nodes of level equal two, etc.

The subdivision of a domain is performed to cut a continuous layer around the boundary of the domain. Then starting with a nodes on this level, a new set of nodes of level equal
three are generated and the process is repeated until the whole region is triangulated. The nodes on a new level are obtained such that the triangles generated are as close to equilateral as possible. This is carried out through a series of simple tests.

The method proposed by S.F. Yeung and M.B. Hsu [3.17] tackles the problem of automatic mesh generation with an interesting algorithm based upon the set theory. A series of objective criteria are defined and then computer is used to scan through the geometric idealization of a boundary to form regions for mesh generation. This procedure is supported by a rigorously built mathematical model which ascertains that the logic used is consistent with the criteria and that the procedure is unique. A cyclic set is defined according to the ordering of segments on a boundary. Corresponding to this is a quadrilateral set whose elements represent the partitioning of the geometry into regions for mesh generation.

As an example, consider an engineering problem. This is an idealization of an intake nozzle for an intermediate heat exchanger. The outline polygon has 10 sides as shown in Fig 3.13a. Referring to the side P1P2 as initial side there are 10 possible combinations of quadrilaterals which have five pairs of equivalent cases. These cases are shown in Fig 3.13b-f. The computer program based on the set theory evaluates and arrives at a solution which is shown in Fig 3.13e. Subsequent automatic mesh generation is given in Fig 3.13g.

A new and interesting approach to mesh generation is that of Park [3.18] which can be applied to problems in which there is a cross-sectional similarity. The basis of this method which is referred to as 'drag method' is a pattern set of 0-dimension (point), 1-dimension (line) and 2-dimension (area) generatrix elements and a set of displacement (step or
Fig 3.13 Idealization of an intake nozzle and its automated mesh generation
increment) control vectors. Starting with its original position, the generatrix set is dragged through a sequence of positions as determined by the succession of displacement control vectors and a mesh is generated in this way.

For example if a triangular element with three nodes is the generatrix and is moved twice, then two prism elements will be formed with 6 nodes each as shown in Fig 3.14.

The key to the simplicity and power of the drag mesh generation method is the generatrix. First, there is no dependence and hence no restrictions on the geometry of the generatrix. Further, there is no dependence on the numbering of nodes in the generatrix. Each generatrix is treated completely separately, so that multiple finite element types can be created simultaneously by including multiple generatrix types, even of different dimensions, in the pattern set.

A recent scheme suggested by Pissanetzky [3.19] has a basis different from the above mentioned schemes. It is based on construction of structures by assembly of modules. Modules are independent units, three dimensional assemblies of nodes and elements. The number of modules to be used, type of the modules and the number of elements comprising each module depend on the particulars of the problem under consideration. In summary the mesh generation involves two major steps:

1- A set of modules are generated independently with respect to a three dimensional integer coordinate system, where every node may be referred to by its integer coordinates. Each module has its own mesh inside. Then, the modules are placed in contact to generate the complete topology of the structure under consideration.

2- The shape and size of the bodies are associated with the topology of the structure. The modules are 'tailored' to the
Fig 3.11 Prism generation

Fig 3.15 Logical Modules  (a)- A parallelepiped module with $S_1 = 4$, $S_2 = 2$ and $S_3 = 3$. The integer coordinates of $A$ are $(1,1,1)$; those of $B$ are $(4,2,2)$. This is said to be a $4 \times 2 \times 3$ sliced module. The numbering of the faces of the module is also shown.  (b)- A peripheral module with two layers of quadrilaterals, in two dimensions. The core is a $2 \times 2$ parallelepiped module. $P$ is the origin of the projection line $PQ$. 

138
bodies or domains associated with the given problem. Various transformations map the integer coordinates of the nodes to geometric coordinates.

Only two types of modules are defined: Parallelepiped (or sliced) modules and peripheral (layers of cuboids, wrapped around a core) modules. A parallelepiped module and a peripheral module are shown in Fig 3.15. Depending on the manner in which the elements (cuboids) are connected, a large variety of shapes can be generated out of these type of modules and most practical requirement can be met.

3.3 COMPARISON OF MESH GENERATION METHODS

There is no 'one-best' scheme available that would satisfy all the users of the finite element method. The main criterion for a mesh generation scheme will be considered to be its applicability to finite element applications. Finite element domains commonly have holes, curved boundaries, areas of special interest, where the solution will vary greatly and zones where materials change. Merits of different methods can be evaluated by their adequacy to cater for physical features of the domain in a computationally efficient way with the minimum discretization error.

Any reliable mesh generation scheme should possess the following features:

1- Precise modelling of boundaries: Boundary nodes should lie precisely on the boundary of the structure. There should be no limitations on the forms of boundary curves that can be accurately modelled.

2- Good correlation between the interior mesh and information on the mesh boundary: The curvatures and node spacings on the boundaries of the region should be well represented in
the interior of the mesh.

3- Minimal input effort: To reduce the time and effort required of the user to set up the finite element model, the amount of input data required should be reduced as far as possible.

4- Broad range of applicability: To minimize the user's learning time, program development time and program size, it is desirable to use a small set of mesh generation techniques that can be applied to a broad range of structural problems.

5- General topology: The method of meshing should not impose any restriction on the topology of the mesh within a region.

6- Automatic topology generation: The method of meshing should create element connectivity without user intervention.

7- Mesh refinement: The user should be able to control element density in any part of the domain.

8- Favourable element shapes: The elements produced by the automatic method process should possess shapes that will not produce ill-conditioning in the finite element model.

9- Optimal numbering patterns: The numbering of nodes and elements within the structure should be arranged such that favourable conditions are obtained for equation solution.

10- Computational efficiency: The method of mesh generation should make efficient use of computer resources to minimize expenses.

The schemes that generate meshes using a simple algorithm (such as George, Bykat, etc.) are not really suitable for the present need of the finite element analyst. Most of these methods cannot properly deal with complex boundaries.
Although they may all produce well-conditioned elements and a regular mesh but they are more geared to the use of triangular elements, which have become less popular with the introduction of new elements (curvilinear elements, etc) and now are classical.

The mesh generation schemes based on Laplacian schemes are let down by their inability to reflect boundary information concerning grid spacing and boundary curvature in the interior of the mesh. The Laplacian-isoparametric schemes can be made suitable but at the expense of extensive computation purely to generate the mesh. The Laplacian schemes also do not possess an inherent method for generating element connectivity within a region except that of Denayer [3.3].

The mesh generation schemes which are based on the construction of curvilinear coordinate systems within a domain are quite powerful. These generators require some initial form of gross partitioning of the structure into simpler sub-blocks. Then using a mapping from a canonical domain (for example a cube) onto a sub-block creates a natural coordinization within it. The constant coordinate lines provide a basis for the automatic generation of the element topology within a region. This technique although useful but places restriction on the element topology and only topology for line, triangular and rectangular element can automatically be generated.

Formex algebra the concepts of which are described in the previous Chapter provides a fundamental foundation for mesh generation schemes. The generality of its mathematical framework provides a wide range of areas of application. Formex algebra with its functions and operations provides an ideal approach for description of interconnection patterns of configurations. The description of interconnection pattern of a configuration is inherent in the formex representing...
that configuration. No restriction is placed on the topology of elements constructing that configuration.

The separation of topological characteristics of a configuration and its geometric shape is an important feature in the concepts of formex algebra. Indeed a formex is, basically, an algebraic representation of the topological characteristic of a configuration and is quite independent of such geometric concepts as 'length' and 'angle'. The geometric information relating to a configuration may normally be obtained from a formex representation of the configuration through simple transformations. Formex algebra can benefit from various advances which have been made in surface/solid modelling techniques by incorporating useful points from these approaches within its well-defined framework. All the accepted surface/solid modelling techniques which are being used in computer aided design provide useful tools in defining retronorms for a wide range of configurations.

The concepts of formex algebra may also be used in conveying information relating to external loads, support arrangements, etc. These points enhance the application range of the formex algebra as a mesh generation scheme. As a whole, the concepts of formex algebra provide a natural base for a simple and efficient finite element mesh generation scheme of considerable versatility. The simplicity of formex approach to finite element mesh generation will be illustrated in the subsequent Chapters.
CHAPTER 4
4 FORMEX APPROACH TO FINITE ELEMENT MESH GENERATION

4.1 INTRODUCTION

The purpose of this Chapter is to explore the ways in which the concepts of formex algebra may be used to generate finite element meshes. The examples in this Chapter will serve to illustrate the basic ideas involved in the formex approach to the finite element mesh generation. The technique adopted in describing the concepts is to start from simple cases and gradually proceed to more complex meshes.

4.2 BASIC APPROACH

To begin with, consider a deep cantilever beam under a uniformly distributed load as shown in Fig 4.1a. This is a plane stress problem and can be solved using the finite element mesh, shown in Fig 4.1b. Brebbia [4.1] employs a mesh similar to the one shown. As may be seen, the mesh consists of simple triangular elements with 4 and 6 divisions in the x and y directions, respectively.

The configuration of Fig 4.1b may be interpreted in two different ways. Firstly as an interconnection pattern of triangular finite elements and, secondly as a formex plot. An immediate consequence is that a suitably constituted formex that has such a plot is capable of representing this interconnection pattern.

In order to represent the configuration shown in Fig 4.1b by a formex, consider a typical triangular element of the mesh which is shown in Fig 4.2. Every node of this element is uniquely determined with respect to the reference system shown. Thus every node may be associated with a signet of the second grade \([U_1, U_2]\). Furthermore, every element of the
Fig 1.1

Fig 1.2
configuration connecting the three nodes i, j and k may be represented by a cantle of the third plextitude

\[ [U_{i1}, U_{i2}; U_{j1}, U_{j2}; U_{k1}, U_{k2}] \].

Then, a homogenous formex of the third plextitude F, in which each cantle relates to an element of the mesh, would represent the interconnection pattern of Fig 4.1b.

Such a formex may be written as

\[ F = \text{rinid}(4,6,1,1) \mid GF \]

where

\[ GF = G \# \text{refid}(1/2,1/2) \mid G \]

and

\[ G = \begin{bmatrix} 0 & 0; 0,1; 1,1 \end{bmatrix}. \]

The geometric coordinates of the configuration of Fig 4.1b is related to its formex representation F through a set of coordinate equations specified as

\[ \begin{align*}
x &= 9 \ U_1 \\
y &= 13 \ U_2.
\end{align*} \]

Figure 4.3 shows the plot of F, drawn with respect to a basibifect retronorm with \( b_1 = 9.0 \) and \( b_2 = 13.0 \) units, respectively.

Now, consider the structure of Fig 4.1a again, but this time with a different finite element idealization as shown in Fig 4.4. The basic 4-noded rectangular element of the mesh can be represented by a four-plex cantle of the form

\[ [U_{i1}, U_{i2}; U_{j1}, U_{j2}; U_{k1}, U_{k2}; U_{m1}, U_{m2}] \].

Then a formex of the fourth plextitude E, in which each cantle relates to an element of the mesh, could represent the
Fig 1.3

Fig 1.4
mesh of Fig 4.4. For example E may be written as

\[ E = \text{rinid}(4,6,1,1) \big| G \]

where

\[ G = \begin{bmatrix} 0,0; & 1,0; & 1,1; & 0,1 \end{bmatrix}. \]

An N-plot of E with respect to a basibifect retronorm with basifactors \( b_1=9.0 \) and \( b_2=13.0 \) units is shown in Fig 4.4.

### 4.3 FORMEX REPRESENTATION OF VARIOUS FINITE ELEMENTS

In previous examples, the mesh generation was confined to the use of simple triangular or rectangular elements. There are many different types of elements available for finite element idealizations. In this Section the flexibility of formex algebra in representation of different finite elements is illustrated through a number of examples.

Consider the quadratic 8-noded rectangular element shown in Fig 4.5. Every node of this element may be associated with a signet of the second grade \([U_1,U_2]\). Thus this element may be represented by an 8-plex cantle

\[ [U_{11},U_{12}; U_{21},U_{22}; U_{31},U_{32}; \ldots ; U_{81},U_{82}]. \]

The following retrocords are employed in plotting Fig 4.5

(a) The tenon of a signet is represented by a small circle whose centre is at the pivot.

(b) The frond of a cantle of the eight plextitude is represented by a configuration which is obtained by connecting all the tenons together, in a sequential manner, each one to its previous one and the last one to the first one.
Fig 1.5

Fig 1.6
Now, consider the finite element shown in Fig 4.6. Again every node of this element may be associated with a signet of the second grade \([U_1, U_2]\). Thus this element may be represented by a cantle of the fourth plextitude

\[ [U_{i1}, U_{i2}; U_{j1}, U_{j2}; U_{k1}, U_{k2}; U_{m1}, U_{m2}] . \]

The retrocords in this case are as follows

(a) As above

(b) The frond of a cantle of the fourth plextitude is obtained by connecting the tenons of the first three signets of the cantle together to form a triangle and the fourth signet is considered as a single signet represented by a small circle.

As the third example, consider the simplest element from the tetrahedron family of finite elements, i.e., the 4-noded tetrahedron shown in Fig 4.7. Every node of this element may be associated with a signet of the third grade \([U_1, U_2, U_3]\). Thus this element may be represented by a cantle of the fourth plextitude

\[ [U_{i1}, U_{i2}, U_{i3}; U_{j1}, U_{j2}, U_{j3}; U_{k1}, U_{k2}, U_{k3}; U_{m1}, U_{m2}, U_{m3}] . \]

The retrocords in this case are

(a) As before

(b) A cantle of the fourth plextitude is represented by a tetrahedron by connecting together the tenons of the four signets.
Fig 1.7

Fig 1.8
4.4 MORE EXAMPLES ON FORMEX REPRESENTATION OF FINITE ELEMENT MESHES

In the initial example, the domain under consideration was very simple. In this Section, domain with irregular boundaries is considered and the power of formex algebra in representation of finite element meshes for such domains is illustrated through a number of examples.

Mesh 1:

Consider the configuration shown in Fig 4.8. To formulate this mesh, let the element of the mesh denoted by an asterisk be represented by

\[ G_1 = [0,18; 2,18; 1,20]. \]

Then, one may proceed by writing

\[ G_2 = \text{ref}(2,19)\text{tran}(1,1)G_1, \]
\[ E_1 = \text{lib}(j=0,8)\text{lib}(i=j,12)\text{tranid}(2i-j,-2j)G_1, \]
\[ E_2 = \text{lib}(j=0,8)\text{lib}(i=j,11)\text{tranid}(2i-j,-2j)G_2, \]
\[ E_3 = E_1 \# E_2 \]

and

\[ F = \text{lam}(2,18)E_3 \]

where \( F \) represents the mesh of Fig 4.8.

Mesh 2:

Consider the configuration of Fig 4.9. This configuration may be considered to consist of two overlapping trapezoids. Each being the reflection of the other. To formulate this mesh, consider one of these trapezoidal regions, let say the upper one (i.e. the one pointing toward the top of the page). A formex \( E_5 \) representing the upper trapezoid may be obtained as
\[ E_5 = E_3 \# E_4 \]

where
\[ E_3 = \text{lib}(j=0,10) \mid \text{lib}(i=j,14) \\mid \text{tranid}(2i-j,2j) \mid E_1, \]
\[ E_4 = \text{lib}(j=0,10) \mid \text{lib}(i=j,13) \\mid \text{tranid}(2i-j,2j) \mid E_2 \]

and where
\[ E_2 = \text{ref}(2,7) \mid \text{tran}(1,1) \mid E_1 \]

and
\[ E_1 = [0,6; 2,6; 1,8]. \]

Then
\[ E_6 = \text{ref}(2,14) \mid E_5 \]

would represent the lower trapezoid. Thus
\[ E_7' = E_6 \# E_5 \]

represents the mesh with some of its elements doubly represented. Then
\[ E = \text{pex} \mid E_7 \]

is a complete description of the configuration of Fig 4.9 and it does not include any double representation of elements.

Mesh 3:

As the third example, consider an octagonal domain with a central square hole. The finite element idealization of this domain is shown in Fig 4.10. The configuration may be partitioned into two trapezoidal regions and two rectangular regions.

Formex formulation of the upper trapezoidal region may be given as
\[ E_6 = E_3 \# E_4 \# E_5 \]
where
\[ E_3 = \text{lib}(j=0,4) \mid \text{lib}(i=j,15-j) \mid \text{tranid}(2i,2j) \mid E_1, \]
\[ E_4 = \text{lib}(j=0,4) \mid \text{lib}(i=j,14-j) \mid \text{tranid}(2i,2j) \mid E_2, \]
\[ E_5 = \text{ref}(1,17) \mid \text{lib}(i=0,4) \mid \text{tranid}(2i,2j) \mid E_1 \]
and where
\[ E_2 = \text{refid}(2,25) \mid E_1 \]
and
\[ E_1 = [0,24; 2,24; 2,26]. \]

The left rectangular region may be formulated as
\[ H_4 = \text{rinid}(5,7,2,-2) \mid H_3 \]
where
\[ H_3 = H_1 \# H_2 \]
and where
\[ H_1 = \text{ref}(2,24) \mid E_1 \]
and
\[ H_2 = \text{refid}(1,23) \mid H_1. \]

Then
\[ F = \text{lam}(1,17) \mid H_4 \# \text{lam}(2,17) \mid E_6 \]
would represent the mesh of Fig 4.10.

Mesh 4:

Let it be required to write a formex formulation for the finite element mesh shown in Fig 4.11. This configuration may be partitioned into six triangular regions. Formex formulation for the bottom left region may be obtained as
\[ F = F_1 \# F_2 \]
where
\[ F_1 = \text{lib}(j=0,5) \mid \text{lib}(i=j,5) \mid \text{tranid}(2i-j,2j) \mid G_1, \]
\[ F_2 = \text{lib}(j=0,4) \mid \text{lib}(i=j,4) \mid \text{tranid}(2i-j,2j) \mid G_2 \]
and where
Fig 4.11

Fig 4.12
\[ G_1 = [0,0; 2,0; 1,2] \]

and
\[ G_2 = \text{tran}(1,1) \cdot \text{ref}(2,1) \cdot G_1. \]

Then the formulation of the complete mesh may be given as
\[ E = E_1 \# E_2 \]
where
\[ E_1 = \text{lamid}(12,12) \cdot F \]
and
\[ E_2 = \text{lam}(2,12) \cdot \text{ref}(2,6) \cdot \text{tran}(1,6) \cdot F. \]

Mesh 5:

Consider mesh of Fig 4.12 which represent a finite element model of the cross-section of a 10 cm thick reinforced concrete shear wall. Murazumi [4.2] employed a mesh similar to the one shown. A formex formulation of the interconnection pattern of the mesh, relative to the indicated pronorm, may be written as
\[ C = C_{FB} \# C_W \# C_{FT}. \]

The elements of the bottom flange may be represented as
\[ C_{FB} = C_5 \# C_6 \]
where
\[ C_5 = \text{lam}(1,29) \cdot C_4, \]
\[ C_6 = \text{rin}(1,9,2) \cdot \text{tran}(1,20) \cdot \text{dil}(1,1/2) \cdot C_1 \]
and where
\[ C_4 = C_2 \# C_3, \]
\[ C_3 = \text{rin}(1,2,16) \cdot C_1, \]
\[ C_2 = \text{rin}(1,2,6) \cdot \text{tran}(1,4) \cdot \text{dil}(1,3/2) \cdot C_1 \]
in which
\[ C_1 = [0,0; 2,0; 4,0; 4,6; 4,12; 2,12; 0,12; 0,6] \]
is an 8-plex maniple representing the element which is
situated at the left bottom corner of the mesh.

The elements of the web may be represented as

\[ CW = C_9 \# C_{10} \# C_{12} \]

where

\[ C_9 = \text{rin}(1,2,22)\text{itr}(16,20)\text{dil}(2,1/6)\text{Cl}, \]
\[ C_{10} = \text{rin}(1,4,4)\text{itr}(22,20)\text{dil}(1/2,1/6)\text{Cl}, \]
\[ C_{12} = \text{lam}(2,21)\text{Cl} \]

and where

\[ C_{11} = C_7 \# C_8, \]
\[ C_7 = \text{rinid}(2,2,22,4)\text{itr}(16,12)\text{dil}(2,1/3)\text{Cl} \]

and

\[ C_8 = \text{rinid}(9,2,2,4)\text{itr}(20,12)\text{dil}(1/2,1/3)\text{Cl}. \]

Finally the elements of the top flange may be represented by

\[ C_{FT} = C_{13} \# C_{17} \]

where

\[ C_{13} = \text{tran}(2,30)\text{dil}(2,2/3)\text{Cl}, \]
\[ C_{17} = \text{lam}(1,29)\text{Cl} \]

and where

\[ C_{16} = C_{15} \# C_{14}, \]
\[ C_{15} = \text{itr}(16,30)\text{dil}(2,2/3)\text{Cl} \]

and

\[ C_{14} = \text{itr}(10,30)\text{dil}(3/2,2/3)\text{Cl}. \]

A MN-plot of C is shown in Fig 4.13.

Mesh 6:

Figure 4.14 shows a finite element model of a reinforced concrete shear wall with two openings. Murzumi [4.2] used a mesh similar to the one shown for the analysis. Let it be required to write a formex for the interconnection pattern of this configuration. First it is assumed that there is no opening and a formex CC representing such a mesh can be
Fig 1.13

Fig 1.14
arrived at as follows

\[ C_1 = [0,0; 3,0; 6,0; 6,4; 6,8; 3,8; 0,8; 0,4], \]
\[ C_2 = \text{tran}(1,6) \text{dil}(1,2/3) | C_1, \]
\[ C_3 = \text{lam}(1,8) | C_1 \text{ ref}(1,11) | C_1, \]
\[ C_4 = \text{tran}(1,22) \text{dil}(1,5/3) | C_1, \]
\[ C_5 = C_2 \# C_3 \# C_4, \]
\[ C_6 = \text{tran}(2,8) \text{dil}(2,1/2) | C_5, \]
\[ C_7 = \text{rin}(2,2,6) | \text{tran}(2,12) \text{dil}(2,7/9) | C_5, \]
\[ C_8 = C_5 \# C_6 \# C_7 \]

and

\[ CC = \text{lamid}(32,24) | C_8. \]

To remove the extra 8 elements, let

\[ G = \{[16,30],[48,18]\}. \]

Then

\[ CB = \text{lux}(G) | CC \]

would be a formex representing the mesh of Fig 4.14. An MN-plot of CB is shown in Fig 4.15.

Mesh 7:

Consider the mesh pattern of finite element model of a vertical slope shown in Fig 4.16. Nimitchi Snitbham, et al [4.3] who investigated the problem of large deformations of slopes employed a finite element similar to the one shown. A formex formulation for this mesh relative to the indicated pronorm, may be written as follows

\[ G_1 = [0,0; 5,0; 5,6; 0,6], \]
\[ G_2 = \text{tran}(1,5) \text{dil}(1,3/5) | G_1, \]
\[ G_3 = G_2 \# G_1, \]
\[ G_4 = \text{lam}(1,14) | G_3 \text{ tran}(1,28) | G_1, \]
\[ G_5 = \text{tran}(1,8) | \text{dil}(1,1/5) | G_1, \]
\[ G_6 = \text{rin}(1,12,1) | G_5. \]
Fig 4.15

Fig 4.16
G7 = G6 # G4,
G8 = tran(2,6)\{dil(2,5/6)\}G7,
G9 = tran(2,11)\{dil(2,1/6)\}G7,
G10 = rin(2,6,1)\{G9,
G11 = tranid(14,17)\{dilid(1/5,1/6)\}G1,
G12 = rinid(6,6,1,1)\{G11,
G13 = ref(1,14)\{G3 # tran(1,28)\}G1,
G14 = tran(2,17)\{dil(2,1/6)\}G13,
G15 = rin(2,6,1)\{G14,
FVS = G15 # G12 # G10 # G8 # G7

where FVS is a formex representing the mesh pattern of the vertical slope of Fig 4.16.

Mesh 8:

As another example, consider the modelling of the saw-cutting process by finite element method. During the cutting or slicing process the tool is subjected to a travelling load. Because of the rotation it is travelling around the circumference of the blade. Tonishoff and Jendryschik [4.4] who investigated this problem used a finite element model consisting of 256 finite 3-noded plate elements. Figure 4.17 shows a mesh similar to that used by Tonishoff. To formulate this mesh, let a formex F be formulated as

\[ F = rinid(8,16,1,1)\{G2 \]

where

\[ G2 = G1 # refid(2.5,0.5)\{G1 \]

and where

\[ G1 = [2,0; 2,1; 3,0]. \]

A paribifect N-plot of F is shown in Fig 4.18. An N-plot of F with respect to a basipolar retronorm with the following basifactors \( b_1=2.0 \) and \( b_2=22.5 \) would represent the mesh shown in Fig 4.17. From a purely graphical point of view, one may regard F as representing the interconnection pattern of the
finite element model of the saw shown in Fig 4.17. However, if $F$ is to serve as data describing the interconnection pattern of the finite element model of the saw for structural analysis, then one must modify $F$ to take account of the connection between the elements indicated by asterisks correctly. Modification of this type can be achieved as follows

\[ E = \text{nov}(H)^{\dagger} F \]

where

\[ H = \text{rin}(1,1,9)^{\dagger}[[0,16; 0,0]]. \]

The formex $E$ provides a description for the interconnection pattern of the finite element model shown in Fig 4.17 and is free from any discontinuities at nodes.

Mesh 9:

Consider the mesh shown in Fig 4.19 which is a finite element model of a flat plate with three rectangular openings. Bykat [4.5] uses a mesh similar to the one shown. A formex $E$ representing the interconnection pattern of the mesh, relative to the indicated pronorm may be obtained as follows

\begin{align*}
E_1 &= [\emptyset, 0; 4, 0; 0, 4], \\
E_2 &= E_1 \# \text{refid}(2,2)^{\dagger} E_1, \\
E_3 &= \text{rinid}(14,3,4,4)^{\dagger} E_2, \\
E_4 &= \text{rinid}(3,3,4,4)^{\dagger} \text{tran}(2,12)^{\dagger} E_2, \\
E_5 &= \text{lam}(1,24)^{\dagger} \text{tranid}(12,20)^{\dagger} E_2, \\
E_6 &= \text{tranid}(44,20)^{\dagger} E_2, \\
E_7 &= \text{rin}(1,2,4)^{\dagger} \text{tranid}(20,12)^{\dagger} E_2, \\
E_8 &= \text{tranid}(52,12)^{\dagger} E_2, \\
E_9 &= \text{rinid}(2,3,4,4)^{\dagger} \text{tranid}(36,12)^{\dagger} E_2, \\
E_{10} &= \text{lamid}(56,24)^{\dagger} \text{lib}(i=3,9)^{\dagger} E_i
\end{align*}

where $E_{10}$ represents all the elements of the mesh except those at the corners of the openings. These can be
Fig 1.19

Fig 1.20
formulated as follows

\[ E_{11} = [\emptyset, \emptyset; 4, \emptyset; \emptyset, 4], \]
\[ E_{12} = \text{lam}(2,15) \mid E_{11}, \]
\[ E_{13} = \text{ref}(2,14) \mid \text{tranid}(-16,-16) \mid \text{dilid}(2,2) \mid E_{11}, \]
\[ E_{14} = [15,15; 14,16; 14,14], \]
\[ E_{15} = E_{14} \# E_{13} \# E_{12}, \]
\[ E_{16} = E_{15} \# \text{ver}(2,1,14,14) \mid \text{ref}(1,14) \mid E_{15}, \]
\[ E_{17} = \text{tranid}(-16,-16) \mid \text{dilid}(2,2) \mid \text{ver}(2,1,15,15) \mid E_{14}, \]
\[ E_{18} = E_{17} \# \text{ver}(2,1,14,14) \mid E_{17}, \]
\[ E_{19} = E_{17} \# E_{16}, \]
\[ E_{20} = \text{lam}(2,16) \mid E_{19} \# \text{ref}(1,16) \mid E_{19}, \]
\[ E_{21} = \text{lam}(1,24) \mid E_{20} \# \text{tran}(1,32) \mid E_{20}, \]
\[ E_{22} = \text{lamid}(56,24) \mid E_{21} \]

and a formex representing the whole mesh may be written as
\[ E = E_{10} \# E_{22}. \]

An MN-plot of \( E_{19} \) representing the mesh at left bottom corner of the leftmost opening is shown in Fig 4.20.

Mesh 10:

As another example consider a thin plate embedded with a centre crack subjected to an applied stress \( \sigma \) at both upper and lower boundary (see Fig 4.21). Because of the axisymmetrical nature of the specimen's geometry and loading, only a quarter of the specimen needs to be discretized. A typical finite element idealization is shown in Fig 4.22 whereas Fig 4.23 depicts detailed element construction near the crack tip. Chow and Jilin [4.6] used a mesh similar to the one shown for the analysis of crack propagation through a plate.

A formex \( F \) representing the interconnection pattern of Fig 4.22 can be formulated as follows

\[ F = F_1 \# F_2 \]
Fig 1.21

Fig 1.22
where \( F_1 \) represents the elements at the crack tip whose parabifect N-plot is shown in Fig 4.23 and \( F_2 \) represent the remaining elements. \( F_1 \) can be formulated as follows

\[
\begin{align*}
A_1 &= [0,0; 16,0; 8,16], \\
A_2 &= \text{tranid}(8,16)\mid A_1, \\
A_3 &= \text{tran}(1,8)\mid \text{ref}(2,24)\mid A_2, \\
A_4 &= \text{lam}(1,32)\mid [24,16; 32,16; 32,32], \\
A_5 &= \text{rin}(1,3,16)\mid A_4, \\
A_6 &= \text{tran}(1,16)\mid \text{rin}(1,5,8)\mid \text{dilid}(0.5,0.5)\mid A_4, \\
A_7 &= \text{tran}(1,24)\mid \text{rin}(1,9,4)\mid \text{dilid}(0.25,0.25)\mid A_4, \\
A_8 &= A_5 \# A_6 \# A_7, \\
A_9 &= [[16,0; 8,16; 20,8], [8,16; 20,8; 24,16]], \\
A_{10} &= A_1 \# A_2 \# A_3 \# A_9, \\
A_{11} &= \text{tran}(1,16)\mid \text{dilid}(0.5,0.5)\mid A_{10}, \\
A_{12} &= \text{tran}(1,24)\mid \text{dilid}(0.25,0.25)\mid A_{10}, \\
A_{13} &= A_{10} \# A_{11} \# A_{12}, \\
A_{14} &= \text{lam}(1,48)\mid A_{13}, \\
A_{15} &= \text{tran}(1,16)\mid \text{rin}(1,2,16)\mid A_3, \\
A_{16} &= \text{tran}(1,16)\mid \text{dilid}(0.5,0.5)\mid \text{rin}(1,4,16)\mid A_3, \\
A_{17} &= \text{tran}(1,24)\mid \text{dilid}(0.25,0.25)\mid \text{rin}(1,8,16)\mid A_3, \\
A_{18} &= A_{17} \# A_{16} \# A_{15}, \\
A_{19} &= \text{lib}(j=0,1)\mid \text{lib}(i=0,19-j)\mid \text{tranid}(2i+j,2j)\mid [28,0; 30,0; 29,2] \\
A_{20} &= \text{lib}(j=0,1)\mid \text{lib}(i=0,18-j)\mid \text{tranid}(2i+j,2j)\mid [29,2; 31,2; 30,2]
\end{align*}
\]

and where

\[
F_1 = A_{20} \# A_{19} \# A_{14} \# A_{13} \# A_{8}.
\]

The formex \( F_2 \) can be obtained as follows

\[
\begin{align*}
R_1 &= [0,96; 32,160; 0,160], \\
R_2 &= \text{lam}(1,160)\mid \text{rin}(2,4,128)\mid \text{lam}(2,160)\mid R_1, \\
R_3 &= [0,96; 32,160; 64,96], \\
R_4 &= \text{lam}(2,160)\mid R_3, \\
R_5 &= \text{rinid}(5,4,64,128)\mid R_4,
\end{align*}
\]
\[ R6 = \text{lam}(2,160) \text{tran}(1,32) \text{ref}(2,128) \text{R3}, \]
\[ R7 = \text{rinid}(4,4,64,128) \text{R6} \]
\[ R8 = R7 \# R5 \# R2 \]
\[ R9 = [[0,0; 0,32; 8,16], [0,32; 16,32; 8,16], \]
\[ [88,16; 96,0; 116,32], [80,32; 116,32; 96,64], \]
\[ [80,32; 116,32; 88,16], [128,96; 96,64; 116,32]], \]
\[ R10 = [0,32; 0,64; 16,32], \]
\[ R11 = \text{lam}(1,32) \text{tran}(1,32) \text{R10}, \]
\[ R12 = \text{lam}(1,48) \text{R11}, \]
\[ R13 = \text{rin}(1,3,32) [16,32; 0,64; 32,64], \]
\[ R14 = R13 \# R12 \# R10, \]
\[ R15 = [0,64; 32,64; 0,96], \]
\[ R16 = R15 \# \text{lam}(1,64) \text{tran}(1,64) \text{R15}, \]
\[ R17 = \text{lam}(1,64) [0,96; 32,64; 64,96], \]
\[ R18 = [[128,96; 116,32; 160,48], \]
\[ [116,32; 96,0; 136,0], \]
\[ [116,32; 136,0; 160,48]], \]
\[ R19 = [128,96; 160,48; 192,96], \]
\[ R20 = \text{rin}(1,3,64) \text{R19}, \]
\[ R21 = \text{rin}(1,2,64) \text{tran}(1,32) \text{ref}(2,72) \text{R19}, \]
\[ R22 = \text{lam}(2,48) [320,0; 288,48; 320,48], \]
\[ R23 = [[160,48; 192,0; 224,48], \]
\[ [192,0; 224,48; 224,0]], \]
\[ R24 = \text{lam}(1,224) \text{R23}, \]
\[ R25 = [[136,0; 160,48; 192,0], \]
\[ [288,48; 256,0; 320,0]], \]
\[ R26 = R25 \# R24 \# R16 \# R17 \# R18 \# R20 \# R21 \# R22 \]
and
\[ F2 = R26 \# R14 \# R8 \# R9. \]

4.5 FORMEX FORMULATION OF FINITE ELEMENT IDEALIZATION
WITH MORE THAN ONE TYPE OF ELEMENT

Up to now, various meshes have been considered in which only one type of finite element was employed. As mentioned in section 4.3, there is no limits on the types of finite element which can be represented through the concepts of
formex algebra. In this Section various finite element meshes which are constructed from more than one type of element will be considered and formulation will be given in each case. The formulation will be carried out in a manner similar to the previous examples, the only difference being that the formex representing the mesh would be a non-homogeneous formex.

Mesh 11:

Consider Fig 4.24, it can be seen that this configuration consists of both rectangular and triangular elements.

Let

\[ E_1 = [0,0; 2,0; 2,2; 0,2] \]
and

\[ E_2 = \text{rinid}(7,7,2,2) \mid E_1 \]

where \( E_1 \) represents a rectangular element and \( E_2 \) represents a square mesh consisting of these rectangular elements. To remove the five central elements. Let

\[ E_3 = \text{pex} \mid \text{rosid}(7,7) \mid \text{rosid}(5,5) \mid [4,6]. \]

Then

\[ E_4 = \text{con}(E_3) \mid E_2 \]

where \( E_4 \) represents all the rectangular elements with the unwanted central ones removed.

Now let a triangular element be represented by

\[ E_5 = [6,4; 8,4; 7,5]. \]

Then

\[ E_6 = \text{rosid}(7,7) \mid \text{rosid}(7,5) \mid E_5 \]
represents the triangular elements and

\[ F = E_4 \# E_6 \]
represents the whole of the mesh.
Mesh 12:

Mesh of Fig 4.25 may be formulated in a similar manner. Let

\[
E_7 = \text{rosid}(7,5) \cdot [6,4; 8,4; 7,5],
\]
\[
E_8 = \{[7,5], [6,6], [8,6]\},
\]
\[
E_9 = \text{con}(E_8) \cdot E_7,
\]

and

\[
E_{10} = \text{rosid}(7,7) \cdot E_9
\]

where \(E_{10}\) represents all the triangular elements of the mesh. Then a formex \(H\) representing the mesh of Fig 4.25 can be written as

\[
H = E_{10} \# E_4
\]

where \(E_4\) is the formex obtained in the formulation of mesh 11.

Mesh 13:

As another example, consider Fig 4.26. This mesh consists of two different types of finite elements. The formulation for the outer triangulated region is the same as that of mesh 3 which is given by \(F\) (see formulation for mesh 3). The formulation for the square grid is similar to that of mesh 11 (or 12). Thus a formex representing this part of the configuration may be given by

\[
G_1 = \text{tranid}(10,10) \cdot E_4
\]

where \(E_4\) is the formex obtained in the formulation of mesh 11 (or 12). Then a formex \(G_5\) representing the inner triangulated region may be obtained as

\[
G_5 = G_4 \# \text{tran}(2,2) \cdot G_3
\]

where
\[ G_4 = \text{rosid}(17,17) \{ G_3, \]
\[ G_3 = \text{rosid}(17,15) \{ G_2 \]

and where
\[ G_2 = [16,14; 18,14; 17,15]. \]

The complete description of the configuration of Fig 4.26 may be given by

\[ E = F \# G_1 \# G_5. \]

Mesh 14:

As another example in this Section consider the finite element model of a tanker web frame as shown in Fig 4.27. Mukhopadhyay [4.7] used a mesh similar to the one shown. The model consists of triangular and rectangular elements. Let a rectangular and a triangular element be represented by the maniples

\[ S_1 = [0,0; 1,0; 1,1; 0,1] \]

and

\[ S_2 = [3,5; 4,6; 3,6], \]

respectively.

Thus a formex formulation for the interconnection pattern of Fig 4.27, relative to the indicated pronorm, may be written as follows

\[ S_3 = \text{rinid}(3,20,1,1) \{ S_1, \]
\[ S_4 = \text{rinid}(20,2,1,1) \{ \text{tranid}(3,8) \{ S_4, \]
\[ S_5 = \text{lan}(2,14) \{ S_4, \]
\[ S_6 = \text{rinid}(2,20,1,1) \{ \text{tran}(1,23) \{ S_1, \]
\[ S_7 = \text{tranid}(3,7) \{ S_1, \]
\[ S_8 = \text{ref}(1,4) \{ S_7 \# \text{lan}(2,7) \{ S_6, \]
\[ S_9 = \text{lan}(2,9) \{ S_8 \# \text{tran}(2,10) \{ S_8, \]
\[ S_{10} = \text{tranid}(22,7) \{ S_1, \]
\[ S_{11} = \text{lan}(2,9) \{ S_{10} \# \text{tran}(2,10) \{ S_{10}, \]
where SR represents all the rectangular elements of the mesh. The triangular elements can be generated as follows

\[ S_{13} = \text{lib}(i = 0,2)\{\text{tran}(i,i)\}|S_2, \]
\[ S_{14} = \text{lam}(2,9)|S_{13} \# \text{tran}(2,10)|S_{13}, \]
\[ S_{15} = \text{lamid}(13,14)\{\text{refid}(7.5,8)\}|S_2, \]
\[ S_{16} = \text{tran}(2,1)\{\text{ref}(1,13)\}|S_2, \]
\[ S_{17} = \text{tranid}(-1,1)\{S_{16} \# S_{16}, \]
\[ S_{18} = \text{lam}(2,9)|S_{17} \# \text{tran}(2,10)|S_{17}, \]
\[ ST = S_{14} \# S_{15} \# S_{18}. \]

Thus a formex representing the interconnection pattern of Fig 4.27 can be written as

\[ S = SR \# ST. \]

Mesh 15:

As the final example in this Section consider the buckling of an I-beam. El-Ghazaly [4,8] who investigated this problem assumed a linear axial strain distribution on the central span. This distribution is approximated by the step-wise variation in the vertical direction. The flanges are then modelled by one dimensional beam elements having cross-section A and the web is modelled by triangular plate elements having thickness w. A mesh similar to the one used by El-Ghazaly is shown in Fig 4.28. A formex formulation for this mesh can be written as follows

\[ T_1 = [0,0; 1,0; 0,1], \]
\[ T_2 = T_1 \# \text{refid}(0.5,0.5)|T_1, \]
\[ T = \text{rinid}(20,15,1,1)|T_2 \]

where T represents all the triangular elements of the mesh.
Fig 1.28

Fig 1.29
The beam elements are represented by a formex $B$ which is formulated as

$$B_1 = [0,0; 1,0]$$
$$B = \text{rinid}(20,2,1,15)\backslash B_1.$$  

Then an $N$-plot of a formex $F$ where

$$F = B \# T$$

drawn with respect to a basibifect retronorm with the basifactors $b_1=1.0$ and $b_2=0.75$ would represent the mesh shown in Fig 4.28.

4.6 FORMEX FORMULATION OF THREE DIMENSIONAL FINITE ELEMENT MESHES

Up to now all the finite element meshes which were considered were two dimensional. Many finite element applications require three dimensional meshes in the form of surfaces or solids. There are a wide range of finite element meshes which can easily be generated using the formex algebra. The basic procedure used in generation of two dimensional finite element meshes can be applied to the generation of three dimensional meshes. Since the purpose of this Chapter is to show the formex approach to finite element mesh generation a number of three dimensional meshes are formulated and presented in this Section.

Mesh 16:

Figure 4.29 shows a finite element mesh of a pressure vessel under internal pressure. A formex $F$ representing the above mesh can be written as follows

$$F = \text{lam}(3,7)\backslash( F_1 \# F_2)$$
where
\[
F_1 = \text{rinit}(9,3,1,1)\mid G, \\
F_2 = \text{rinit}(5,4,1,1)\mid \text{tranit}(4,3)\mid G
\]

and where
\[
G = \begin{bmatrix}
1,0,0; 1,0,1; 1,1,1; 1,1,0; \\
2,1,0; 2,1,1; 2,0,1; 2,0,0
\end{bmatrix}.
\]

A plot of \( F \) with respect to a basicylindrical retronorm would represent the mesh of Fig 4.29.

Mesh 17:

As another example, let it be required to calculate the post-buckling strength of a thin walled member. The thin-walled member under consideration is a hat-section beam, a finite element idealization of which is shown in Fig 4.30. Lee and Harris [4.9] employed a mesh similar to the one shown for the analysis. The formex formulation of the mesh shown is as follows

\[
\begin{align*}
H_1 &= \text{lam}(1,10)\mid [0,0,32; 10,0,32; 10,5,32; 0,5,32], \\
H_2 &= [20,0,32; 22,0,30; 22,5,30; 20,0,32], \\
H_3 &= \text{rin}(3,3,10)\mid [22,0,0; 35,0,0; 35,5,0; 22,5,0], \\
H_4 &= [35,0,0; 35,5,0; 35,5,5; 35,0,5], \\
H_5 &= H_1 \# H_2 \# H_3 \# H_4, \\
H_6 &= \text{lam}(2,5)\mid H_5, \\
H_7 &= \text{tran}(2,10)\mid \text{dil}(2,2)\mid H_5, \\
H_8 &= \text{tran}(2,20)\mid \text{dil}(2,3)\mid H_5, \\
H_9 &= \text{tran}(2,35)\mid \text{dil}(2,4)\mid H_5, \\
H_{10} &= H_7 \# H_8, \\
H_{11} &= \text{lam}(2,45)\mid H_{10}, \\
H_{12} &= H_6 \# H_{11} \# H_9, \\
H &= \text{lamid}(0,0)\mid H_{12}.
\end{align*}
\]

An plot of \( H \) with respect to a basitrifect retronorm with the following basifactors \( b_1=1.2, b_2=1.5 \) and \( b_3=1.2 \) would give rise to the mesh shown in Fig 4.30.
Consider Fig 4.31 which shows a finite element model of a ribbed shell dome. The finite element mesh shown consists of rectangular plate and beam elements. A formex formulation for the mesh shown can be written as

\[ F_3 = F_1 \# F_2 \]

where \( F_1 \) and \( F_2 \) represent the plate and beam elements, respectively and are defined as follows

\[ F_1 = \text{rinit}(18,7,1,1)|G_1, \]
\[ F_2 = \text{rinit}(9,7,2,1)|G_2 \]

and where

\[ G_1 = [10,0,0; 10,1,0; 10,1,1; 10,0,1] \]

and

\[ G_2 = [10,1,0; 10,1,1]. \]

Form a purely graphical point of view, one may regard \( F_3 \) as representing the interconnection pattern of the finite element model of the dome of Fig 4.31. However, if \( F_3 \) is to serve as data describing the interconnection pattern of the finite element model of the dome for structural analysis, then one must modify \( F_3 \) to take account of the connection between the elements indicated by asterisks correctly. Modification of this type can be achieved as follows

\[ F = \text{nov}(H)|F_3 \]

where

\[ H = \text{rinit}(3,1,8)|[10,18,0; 10,0,0] \]
\[ \# \text{lib}(i=0,17)|[9,i,0; 9,0,0]. \]

An marginal natural plot of \( F \) with respect to a basispherical retronorm with the basifactors \( b_1=1.0, b_2=20.0 \) and \( b_3=10.0 \) would represent the mesh of Fig 4.31.
Fig 1.32

Fig 1.33
MESH 19:

As another example consider Fig 4.32. It shows a finite element idealization of a component of a train gear. Wilsworth [4.10] employed a mesh similar to the one shown. A formex formulation of the mesh shown can be obtained as follows

\[ F = F_1 \# F_2 \# F_3 \]

where

\[ F_1 = \text{rinit}(20,7,1,1)\mid G, \]
\[ F_2 = \text{rinit}(20,2,1,1)\mid \text{tran}(1,1)\mid G, \]
\[ F_3 = \text{rinit}(20,3,1,1)\mid \text{tran}(1,3)\mid G \]

and where

\[ G = [4,0,0; 4,1,0; 4,1,1; 4,0,1; 5,0,0; 5,1,0; 5,1,1; 5,0,1]. \]

An N-plot of \( F \) with respect to a basicylindrical retromorph with the basifactor \( b_1=1.0, b_2=18.0 \) and \( b_3=1.0 \) would represent the mesh shown in Fig 4.32.

MESH 20:

As a further example consider another component of the train gear, a finite element idealization of which is shown in Fig 4.33 [4.10]. A formex \( E \) representing this mesh can be formulated as follows

\[ E = \text{lib}(i=1,5)\mid E_i \]

where

\[ E_1 = \text{rinit}(20,15,1,1)\mid G, \]
\[ E_2 = \text{rinit}(20,4,1,1)\mid \text{tranis}(1,6)\mid G, \]
\[ E_3 = \text{rinid}(7,20,1,1)\mid \text{tranis}(-4,5)\mid G, \]
\[ E_4 = \text{rinid}(10,20,1,1)\mid \text{tranis}(-4,20)\mid G, \]
\[ E_5 = \text{rinad}(2,20,7,1,1,2)\mid \text{tranis}(-4,6)\mid G \]
An N-plot of $F$ with respect to a basicylindrical retronorm with the basifactors $b_1=1.0$, $b_2=18.0$ and $b_3=1.0$ would represent the mesh shown in Fig 4.33.

4.7 MESH GRADING

As it is well known, the computational efficiency of a finite element solution of a problem can be significantly improved by using large elements in those region where the solution is expected to change slowly and small elements where rapid change is anticipated. The mesh generation problem becomes more difficult when a variation in element density from region to region is required in the finite element model. Formex approach to finite element mesh generation permits a large degree of control over such aspect as mesh grading, mesh refinement, element proportion and element topology. Using the concepts of formex algebra, the mesh grading of a finite element model can be carried out in two distinct ways.

In the first approach a formex is formulated in a way that it contains the necessary grading information. Any finite element mesh can be represented by a formex whose variant or basiant N-plot is both topologically and geometrically identical to it. Meshes 5, 6, 7, 10 and 17 in the previous sections were formulated in this way. This approach is ideal when dealing with finite element meshes in which the mesh refinement needs to be local and does not effect the rest of the mesh. Local high density modelling around a point of stress concentration can easily be performed using various formex functions. Although this approach is convenient for some types of mesh grading, it makes the formulation process lengthy when dealing with meshes in which the size of elements of the mesh are required to increase or decrease.
successively.

In the second approach, one proceeds with the formex formulation of a finite element mesh without any worry regarding the grading of the mesh. The formex obtained in this way only represents the mesh topologically. The geometric particular and thus the grading of the mesh is then dictated by the particulars of the retronorm employed for drawing the mesh.

**MESH 21:**

As an example consider the mesh shown in Fig 4.34. This shows a finite element mesh used for estimating the stresses at the interface of a waste container sleeve and adjacent salt. William and Pariseau [4.11] who investigated this problem used a similar mesh. Let it be required to write a formex formulation for this mesh. A finite element model which is topologically identical to the above mentioned mesh is shown in Fig 4.35. A formex representing this configuration may be formulated as

\[
E = \text{lam}(2,0)\mid \text{lib}(i=1,5)\mid E_i
\]

where

\[
E_1 = \text{lam}(1,3)\mid \text{rin}(2,3,2)\mid G_2,
E_2 = \text{rin}(2,2,2)\mid \text{tranid}(2,2)\mid G_2,
E_3 = \text{rinid}(5,3,2,2)\mid \text{tran}(2,6)\mid G_2,
E_4 = \text{rin}(1,5,2)\mid G_5,
E_5 = \text{rinid}(5,5,2,1)\mid G_7
\]

and

\[
G_2 = \text{lamid}(1,1)\mid (G_1 \# \text{refid}(1/2,1/2)\mid G_1),
G_4 = \text{lux}(G_3)\mid \text{tran}(2,12)\mid G_2,
G_5 = G_4 \# [1,13; 2,14; 0,14],
G_7 = G_6 \# \text{refid}(1,14.5)\mid G_6,
G_1 = [0,0; 1,1; 1,0],
G_3 = [1,14]
\]
and where

\[ G_6 = [0,14; 0,15; 2,15]. \]

An \( N \)-plot of formex \( E \) with respect to a metribifect retronorm with the following factors \( b_1=0.3, b_2=1.0, m_1=1.0 \) and \( m_2=1.15 \) would represent the mesh of Fig 4.34. It can be seen that the mesh grading is dictated by the retronorm employed in drawing the formex \( E \) and this produces the successively increasing size of the elements of the mesh.

MESH 22:

As a second example consider the mesh shown in Fig 4.36 which is a finite element model of a vertical slope. Snitbham [4.3] who investigated the problem of large deformation in slopes used a mesh similar to the one shown. Now, let a formex \( F \) be formulated as follows

\[ F = F_1 \# \text{lam}(1,0)|F_2 \]

where

\[ F_2 = \text{ref}(2,0)|F_1, \]
\[ F_1 = \text{rinid}(10,10,1,1)|G \]

and where

\[ G = [0,0; 1,0; 1,1; 0,1]. \]

An \( N \)-plot of \( F \) with respect to a paribifect retronorm is shown in Fig 4.37. An \( N \)-plot of \( F \) with respect to a metribifect retronorm with \( b_1=1.0, b_2=1.0, m_1=1.10 \) and \( m_2=1.10 \) would represent the mesh shown in Fig 4.36.

4.8 REFINEMENT OF FINITE ELEMENT MESHES

In using the finite element method, it is often necessary to increase the density of the overall mesh in order to obtain a more accurate solution for the problem under consideration. Formex algebra solves the problem of mesh refinement in an
elegant way. One may write the formulation of a number of meshes in terms of one or more parameters. Then a series of meshes ranging from coarse to fine can be generated by specifying values for the parameters. Each of these meshes would be drawn with respect to a retronorm. The basifactors/metrifactors of these retronorms could also be defined in terms of parameters.

MESH 23:

As an example consider the rectangular plate with an opening as shown in Fig 4.38. Let it be required to write a formex formulation for a number of finite element meshes for this structure. One can start by writing a generic formulation as follows:

\[ \text{MRP}(n1, n2, n3, n4) = \text{lux}(E1) | F1 \]

where

\[ F1 = \text{rinid}(n1, n2, 1, 1) | G, \]
\[ E1 = \text{pex}_{\text{lamid}}(0, 0) | \text{rinid}(n3, n4, 1, 1) | [0, 0] \]

and where

\[ G = [0, 0; 1, 0; 1, 1; 0, 1]. \]

MRP is the name given to the formulation (standing for Meshes for Rectangular Plate). For instance MRP(5, 4, 2, 1) would represent the mesh of Fig 4.39 drawn with respect to a basibifect retronorm with \( b1=8.0 \) and \( b2=8.0 \). Similarly MRP(10, 8, 4, 2) would represent the mesh of Fig 4.40 drawn with respect to a basibifect retronorm with \( b1=4.0 \) and \( b2=4.0 \) and finally MRP(15, 12, 6, 3) would represent the mesh of Fig 4.41 drawn with respect to a basibifect retronorm with \( b1=2.0 \) and \( b2=2.0 \).

MESH 24:

As another example consider the laminated composite shown in Fig 4.41. Let it be required to calculate the interlaminar stresses at the tab-specimen interface under the loading
Fig 4.38

Fig 4.39
shown. A generic formulation for a number of finite element meshes for this problem can be written as

\[ \text{FEMLC}(n_1, n_2, n_3, n_4, n_5) = F_1 \# F_2 \# F_3 \# F_4 \]

where

\[ F_1 = \text{rinid}(n_1, 2n_5, 3, 2)' \cdot G_1, \]
\[ F_2 = \text{rinid}(n_2, 2n_5, 2, 2)' \cdot \text{tran}(1, 3n_1)' \cdot G_2, \]
\[ F_3 = \text{rinid}(n_3, 2n_5, 2, 2)' \cdot \text{tran}(1, 3n_1+2n_2)' \cdot G_3, \]
\[ F_4 = \text{rinid}(n_4, n_5, 2, 2)' \cdot \text{tran}(1, 3n_1+2n_2+n_3)' \cdot G_2 \]

and where

\[ G_1 = [0, 0; 3, 2; 3, 2; 0, 2], \]
\[ G_2 = [0, 0; 2, 0; 2, 2; 0, 2] \]

and

\[ G_3 = [0, 0; 1, 0; 1, 2; 0, 2]. \]

FEMLC is the name given to the formulation (standing for Finite Element Meshes for Laminated Composite) and the parameter list specifies the order in which the parameters are to be given. For instance, FEMLC(5, 14, 10, 4, 9) would represent the mesh of Fig 4.43 and FEMLC(5, 9, 20, 4, 9) would represent the mesh of Fig 4.44 and finally FEMLC(400, 800, 800, 100, 5) would represent the mesh of Fig 4.45.

It can be seen that the generic formulation approach not only provides an efficient way for changing the overall density of the mesh but also it can be used to change the density of a part of a mesh.

4.9 LOCAL MESH MODIFICATION

Formex approach to finite element mesh generation provide the user with convenient methods for defining and manipulating the overall form of a finite element mesh. In some cases it become necessary to make local modification to the mesh to meet the requirement of a particular design. Various formex
Fig 1.12

Fig 1.13
Fig 1.11

Fig 1.15
meet the requirement of a particular design. Various formex functions and operations may be used for local mesh modification without changing the overall mathematical description of the whole mesh. It enables the user to delete or reposition existing nodes, to create new nodes and to add or delete individual elements.

MESH 25:

As an example consider Fig 4.31 and let it required that the following modification be carried out to the mesh shown

a) An opening is to be created at the top of the dome.

b) The dome is required to be reinforced by two ring beam at the top and bottom.

A formex E representing the above dome can be obtained by modifying the formex F which represents the dome of Fig 4.31 and is obtained as follows

\[ E = E_1 \# E_2 \]

where

\[ E_1 = \text{lux}(G)\|F, \]
\[ E_2 = \text{rinit}(2,7,1,6)\|[10,0,1; 10,1,1] \]

and

\[ G = [10,0,0]. \]

An MN-plot of E with respect to a basispherical retronorm with the following basifactors b1=1.0, b2=20.0 and b3=10.0 would represent the above mentioned dome and is shown in Fig 4.46.

4.10 USER'S DEFINED RETRONORMS

In the previous examples all the meshes were plotted with respect to one of the 18 categories of retronorms which were defined in Chapter two. However the availability of these
Fig 1.16

Fig 1.17
retronorms does not place any restriction on the use of special coordinate equations. The user can define any form of coordinate equations suitable for the problem under consideration.

MESH 26:

As an example consider the toroidal surface shown in Fig 4.47. Such a surface is best defined in terms of a cylindrical coordinate system through the following coordinate equations

\[
\begin{align*}
    r &= R + a \cos(b_1 \theta_1) \\
    \theta &= b_2 \theta_2 \\
    z &= a \sin(b_1 \theta_1).
\end{align*}
\]

Now consider a formex F which is formulated as follows

\[
F = \text{nov}(E)|F_1
\]

where

\[
\begin{align*}
    F_1 &= \text{rinid}(20,20,1,1); [0,0; 1,0; 1,1; 0,1] \\
    E &= \text{rin}(1,20,1); [0,20; 0,0] \\
    \# &= \text{rin}(2,20,1); [20,0; 0,0]
\end{align*}
\]

A view of a plot of F with respect to the above defined retronorm with \(b_1=\pi/10\), \(b_2=\pi/20\), \(R=50.0\) and \(a=20.0\) is shown in Fig 4.48.

MESH 27:

As another example, consider the mesh shown in Fig 4.49. It shows a pressure vessel which consists of a toroidal part and a cylindrical part. Yates [4.12] used a mesh similar to the one shown. This mesh can easily be defined in a cylindrical coordinate system through the following coordinate equations
\[
\begin{align*}
R &= 50 + 30 \cos \left( \frac{\pi U_1}{20} \right) \\
\text{For } U_2 < 10 &\Rightarrow \theta = \frac{\pi U_2}{40} \\
Z &= 30 \sin \left( \frac{\pi U_1}{20} \right) \\
R &= 50 \\
\text{For } U_2 > 10 &\Rightarrow \theta = \frac{\pi U_2}{40} \\
Z &= 30 + 4(U_1 - 10).
\end{align*}
\]

Using such coordinate equations a formex \( F \) which is formulated as follows

\[ F = \text{nov}(E) | F_1 \]

where

\[ F_1 = \text{rinid}(40,40,1,1) | [0,0; 1,0; 1,1; 0,1] \]

and

\[ E = \text{rin}(1,40,1) | [0,40; 0,0] \]

can be transformed into a geometrical shape representing the configuration of Fig 4.49.

MESHES 28 and 29:

As another example consider Figs 4.50 and 4.51. It shows two finite element meshes which were used by Baroonian [4.13] for analysing the behaviour of fibre reinforced corrugated sheets. Now let two formices \( F_1 \) and \( F_2 \) be formulated as follows

\[ F_1 = \text{rinid}(60,3,2,1) | [0,0; 2,0; 2,1; 0,1] \]

\[ \# \text{rinid}(60,9,2,2) | [0,3; 2,3; 2,5; 0,5] \]

and

\[ F_2 = \text{lam}(2,9) | \text{rinid}(60,3,2,1) | [0,0; 2,0; 2,2; 0,2] \]

\[ \# \text{rinid}(60,6,2,1) | [0,6; 2,6; 2,7; 0,7]. \]

Plots of formices \( F_1 \) and \( F_2 \) with respect to a trifect retronorm using the following coordinate equations
are shown in Figs 4.50 and 4.51, respectively, where \( b_1=30.0 \), \( b_2=50.0 \), \( b_3=0.125 \) and \( b_4=15.0 \).

**MESH 30:**

As another example consider the mesh shown in Fig 4.52. Let a formex \( F \) be formulated as follows:

\[
F = \text{nov}(E) ; \text{rinid}(30,36,1,1) ; [0,0 ; 1,0 ; 1,1 ; 0,1] \\
\text{where} \\
E = \text{rin}(2,36,1) ; [30,0 ; 0,0] \ # \ lib(i=0,30) ; [0,i ; 0,0].
\]

A plot of \( F \) with respect to a cylindrical coordinate system with the following coordinate equations:

\[
r = 0.3 \ U_1 \\
o = 10.0 \ U_2 \\
z = 10.0 \ \text{SIN}(\pi \ U_1/6)
\]

is shown in Fig 4.52.

**MESH 31:**

As the final example in this Chapter consider the elliptical dome shown in Fig 4.53. Let a formex \( F \) be formulated as follows:

\[
F = F_1 \ # F_2 \ # F_3 \\
F_1 = \text{nov}(E_1 \ # E_2) ; \text{rinit}(20,20,1,1) ; \\
[10,0,0 ; 10,0,1 ; 10,1,1 ; 10,1,0], \\
F_2 = \text{nov}(E_1 \ # E_2) ; \text{rinit}(10,20,2,1) ; [10,0,0 ; 10,0,1].
\]
\[ F_3 = \text{nov}(E_1) \cdot \text{rin}(2, 20, 1) \cdot [10, 0, 20; 10, 1, 20] \]

where
\[ E_1 = \text{rin}(3, 20, 1) \cdot [10, 20, 0; 10, 0, 0] \]

and
\[ E_2 = \text{lib}(i=0, 20) \cdot [10, 0, 0; 10, 0, 0]. \]

A view of a plot of F with respect to a spherical coordinate system relative to the following coordinate equation

\[
\begin{align*}
r &= r_1 / (\sin \gamma + r_1 \cos \gamma) \\
r_1 &= 1 / (4.0 \sin \Theta + \cos \Theta) \\
\Theta &= 18.0 \text{ U2} \\
\gamma &= 4.0 \text{ U3}
\end{align*}
\]

is shown in Fig 4.53.
CHAPTER 5
5 DE XIANT RETRONORMS

5.1 INTRODUCTION

The three classes of retronorms which were defined in the previous Chapter are limited in usage for practical problems. Sometimes, it is required to define a normat in a way that the distribution of normat points and thus the spacing of normat lines/surfaces vary over a given region in a more complex pattern than that which can be produced using the basiant or metriant retronorms. Complex patterns of distribution of normat points can be achieved through a new class of retronorms that are referred to as 'dexiant retronorms'. The idea of dexiant retronorms is best illustrated through examples. The basis of the idea is first introduced using a series of simple examples and gradually more complex forms of dexiant retronorms are considered.

5.2 SIMPLE DE XIANT RETRONORMS

Consider the compound normat of Fig 5.1, it consists of three connected regions as shown. Each is paved by a basiant normat with a pair of basifactors b1 and b2. For the first region which is bounded by the normat lines U1=I1, U1=I2, U2=J1 and U2=J2 the basifactors are b11 and b2. For the second region which is bounded by the normat lines U1=I2, U1=I3, U2=J1 and U2=J2 the basifactors are b12 and b2. Finally for the third region which is bounded by the normat lines U1=I3, U1=I4, U2=J1 and U2=J2 the basifactors are b13 and b2.

The coordinates of normat points are defined as
Fig 5.1

Fig 5.2
If $I_1 \leq U_1 \leq I_2$ then
\[
\begin{align*}
  x &= x_1 + b_{11}(U_1 - I_1) \\
  y &= y_1 + b_{22} U_2
\end{align*}
\]

If $I_2 \leq U_1 \leq I_3$ then
\[
\begin{align*}
  x &= x_2 + b_{12}(U_1 - I_2) \\
  y &= y_1 + b_{22} U_2
\end{align*}
\]

If $I_3 \leq U_1 \leq I_4$ then
\[
\begin{align*}
  x &= x_3 + b_{13}(U_1 - I_3) \\
  y &= y_1 + b_{22} U_2.
\end{align*}
\]

where $x_1, x_2$ and $x_3$ are the $x$-coordinates at the starting border of each region (parallel to $y$ axis) and $y_1$ is the $y$ coordinate at the starting border of all the regions (parallel to $x$-axis) and where $x_2$ and $x_3$ in terms of $x_1$ are given as follows

\[
x_2 = x_1 + b_{11}(I_2 - I_1)
\]

and

\[
x_3 = x_2 + b_{12}(I_3 - I_2).
\]

Each one of the normat lines which is parallel to the $y$-axis relates to a value of $U_1$ and each one of the normat lines which is parallel to the $x$-axis relates to a value of $U_2$. As stated earlier, each region is bounded by two pairs of normat lines. One pair is parallel to the $x$-axis and the other pair is parallel to the $y$-axis. For the first region the normat lines at the borders of the region relate to the values of $I_1$ and $I_2$ in the $U_1$ direction and to the values of $J_1$ and $J_2$ in the $U_2$ direction as shown in Fig 5.1. These four values are referred to as the 'border values' of the first region. Similarly the values of $I_2, I_3, J_1$ and $J_2$ are referred to as the border values of the second region, etc.

In specifying the basifactors one may provide them directly or indirectly by giving the $x$ and $y$ coordinates of the normat
lines at the borders of each region.

As an example of indirect specification of the basifactors, for the first region one can write

\[ b_{11} = \frac{(x_2 - x_1)}{(I_2 - I_1)} \]

and

\[ b_2 = \frac{(y_2 - y_1)}{(J_1 - J_2)}. \]

Thus by providing the x and y coordinates of the borders of each region and the border values of each region one can obtain the basifactors for a given region.

As an example, let formex E whose paribifect N-plot is shown in Fig 5.2 be given as follows

\[ E = \text{lam}(1,10) \mid F \]

where

\[ F = F_1 \# F_2 \# F_3 \# F_4 \]

and

\[ F_1 = \text{rinid}(4,12,1,1) \mid G, \]
\[ F_2 = \text{rin}(2,10,1) \mid \text{tranid}(4,1) \mid G, \]
\[ F_3 = \text{rinid}(5,8,1,1) \mid \text{tranid}(5,2) \mid G, \]
\[ F_4 = \text{lam}(2,6) \mid \{[4,0; 5,1; 4,11,4,1, 5,1; 6,2; 5,2]\} \]

and

\[ G = [0,0; 1,0; 1,1; 0,1]. \]

Let formex E be plotted with respect to the above normat which consists of the following three regions:

1st region is a zone for which \(0 < U_1 < 15\) and \(0 < U_2 < 20\),
2nd region is a zone for which \(6 < U_1 < 15\) and \(0 < U_2 < 20\),
and finally
3rd region is a zone for which \(15 < U_1 < 20\) and \(0 < U_2 < 20\).

The normat also has the following basifactors
Fig 5.3

Fig 5.4
A plot of $E$ with respect to this normat is shown in Fig 5.3.

As another example, let the formex $E$ be plotted with respect to the normat of Fig 5.1 but with the following basifactors

\begin{align*}
\text{1st region} & \quad b_1 = 0.9 \\
\text{2nd region} & \quad b_1 = 2.4 \\
\text{3rd region} & \quad b_1 = 1.3 \\
& \quad b_2 = 1.5
\end{align*}

A plot of $E$ with respect to this normat is shown in Fig 5.3.

As another example, let again the formex $E$ be plotted with respect to the normat of Fig 5.1 but this time let the basifactors of the retronorm be given as follows

\begin{align*}
\text{1st region} & \quad b_1 = 2.5 \\
\text{2nd region} & \quad b_1 = 1.2 \\
\text{3rd region} & \quad b_1 = 0.55 \\
& \quad b_2 = 0.9
\end{align*}

A plot of $E$ with respect to this retronorm is shown in Fig 5.4.

As another example, consider the normat of Fig 5.6. It consists of three basipolar retronorms. The first retronorm has basifactors $b_1$ and $b_2$ and paves the region bounded by normat lines $U_1=I_1$, $U_1=I_2$, $U_2=J_1$ and $U_2=J_2$. The second region is paved by a basipolar retronorm with the basifactors $b_1$ and $b_2$. This region is bounded by the normat lines $U_1=I_1$, $U_1=I_2$, $U_2=J_2$ and $U_2=J_3$. Finally the third region is bounded by the normat lines $U_1=I_1$, $U_1=I_2$, $U_2=J_3$ and $U_2=J_4$ and is paved by a basipolar retronorm with the basifactors $b_1$ and $b_2$. 

A plot of $E$ with respect to this normat is shown in Fig 5.5.
The coordinates of the normat points are defined as

\[
\begin{align*}
\text{J2} &\Rightarrow \text{U2} \Rightarrow \text{J1} & r &= b_1 U_1 \\
& & \theta &= \theta_1 + b_{21} (U_2 - J_1) \\
\text{J3} &\Rightarrow \text{U2} \Rightarrow \text{J2} & r &= b_1 U_1 \\
& & \theta &= \theta_2 + b_{22} (U_2 - J_2) \\
\text{J4} &\Rightarrow \text{U2} \Rightarrow \text{J3} & r &= b_1 U_1 \\
& & \theta &= \theta_3 + b_{23} (U_2 - J_2)
\end{align*}
\]

where \(\theta_1, \theta_2\) and \(\theta_3\) are polar angles at the starting border of each region with respect to the coordinate system shown and are as follows

\[
\begin{align*}
\theta_2 &= \theta_1 + b_{21} (J_2 - J_1) \\
\theta_3 &= \theta_2 + b_{22} (J_3 - J_2)
\end{align*}
\]

Let a formex E whose paribifect N-plot is shown in Fig 5.7 be formulated as follows

\[
E = E_1 \# E_2
\]

where

\[
\begin{align*}
E_1 &= \text{rinid}(10, 30, 1, 1) \| \text{tran}(1, 10) \| G \\
E_2 &= \text{rin}(2, 3, 12) \| \text{rinid}(5, 6, 1, 1) \| G
\end{align*}
\]

and

\[
G = [5, 0; 6, 0; 6, 1; 5, 1].
\]

Now, let formex E be plotted with respect to the normat of Fig 5.6 for which the border values are specified as follows

\[
I_1 = 0, I_2 = 20, J_1 = 0, J_2 = 12, J_3 = 18 \text{ and } J_4 = 30
\]

and the basifactors are given as
bl = 2.5, b2l = 7.0, b22 = 18.0 and b23 = 3.0.

A plot of E with respect to this retronorm is shown in Fig 5.8.

As another example, let formex E be plotted with respect to the retronorm of Fig 5.6 but with the following basifactors

\[ b1 = 3.0, b2l = 20.0, b22 = 3.0 \text{ and } b23 = 8.0. \]

A plot of E with respect to this retronorm is shown in Fig 5.9.

As a further example, let formex E be plotted with respect to the retronorm of Fig 5.6, keeping the basifactors as specified above but redefining the border values as follows

\[ I1 = 0, I2 = 20, J1 = 0, J2 = 10, J3 = 15 \text{ and } J4 = 30. \]

A plot of E with respect to this retronorm is shown in Fig 5.10.

Let a formex F whose paribifect N-plot is shown in Fig 5.11 be defined as follows

\[ F = \lambda_{mid(17,16)}|{(E1 \# E2)} \]
\[ E1 = \lambda_{lib(i=0,6)}|{lib(j=i,15-i)}|{tranid(i,j)}|G1 \]
and
\[ E2 = \lambda_{am(2,8)}|{lib(i=0,5)}|{tranid(i,i)}|G2 \]

where
\[ G1 = [10,0; 11,0; 11,1; 10,1] \]
and
\[ G2 = [11,0; 11,1; 12,1]. \]

Figure 5.12 shows a plot of F with respect to the normat of Fig 5.6 for which
b1 = 3.0, b21 = 5.0, b22 = 19.0, b23 = 10.0,
and
I1 = 0, I2 = 24, J1 = 0, J2 = 20 and J3 = 32.

As the third example consider the compound normat of Fig 5.13. It consists of 9 regions. The first region which is paved by a metriant retronorm is bounded by the normat lines U1=I1, U1=I2, U2=J1 and U2=J2. It has basifactors b11 and b21 and its metrifactors are m11 and m21, respectively. The second region is bounded by normat lines U1=I2, U1=I3, U2=J1 and U2=J2. Its basifactors are b12 and b21 and its metrifactors are m12 and m21, respectively, etc. The values of the basifactors and metrifactors for all the regions are given in Table 5.1.

Table 5.1

<table>
<thead>
<tr>
<th>REGION</th>
<th>BASIFACTORS</th>
<th>METRIFACTORS</th>
<th>BORDER VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>b11, b21</td>
<td>m11, m21</td>
<td>I1, I2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J1, J2</td>
</tr>
<tr>
<td>2</td>
<td>b12, b21</td>
<td>m12, m21</td>
<td>I2, I3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J1, J2</td>
</tr>
<tr>
<td>3</td>
<td>b13, b21</td>
<td>m13, m21</td>
<td>I3, I4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J1, J2</td>
</tr>
<tr>
<td>4</td>
<td>b11, b22</td>
<td>m11, m22</td>
<td>I1, I2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J2, J3</td>
</tr>
<tr>
<td>5</td>
<td>b12, b22</td>
<td>m12, m22</td>
<td>I2, I3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J2, J3</td>
</tr>
<tr>
<td>6</td>
<td>b13, b22</td>
<td>m13, m22</td>
<td>I3, I4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J2, J3</td>
</tr>
<tr>
<td>7</td>
<td>b11, b23</td>
<td>m11, m23</td>
<td>I1, I2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J3, J4</td>
</tr>
<tr>
<td>8</td>
<td>b12, b23</td>
<td>m12, m23</td>
<td>I2, I3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J3, J4</td>
</tr>
<tr>
<td>9</td>
<td>b13, b23</td>
<td>m13, m23</td>
<td>I3, I4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>J3, J4</td>
</tr>
</tbody>
</table>

As an example let a formex F, a paribifect N-plot of which is shown in Fig 5.14 be given as

\[ F = E1 \# E2 \]

where
Fig 5.13

Fig 5.14
\[ E_1 = \text{rinid}(10,10,3,3); G_1, \]
\[ E_2 = \text{rinid}(9,9,3,3); G_2, \]
\[ G_1 = [1; 0; 2; 0; 3; 1; 3, 2; 2, 3; 1; 3; 0; 2; 0, 1] \]

and
\[ G_2 = [3, 2; 4, 3; 3, 4; 2, 3]. \]

Also, let the factors in Table 5.1 be specified as follows
\[ b_{11} = 2.5, \quad b_{21} = 2.0, \]
\[ b_{12} = 1.5, \quad b_{22} = 1.5, \]
\[ b_{13} = 4.0, \quad b_{23} = 3.0, \]
\[ m_{11} = 0.95, \quad m_{21} = 1.1, \]
\[ m_{12} = 1.0, \quad m_{22} = 1.0, \]

and
\[ m_{13} = 1.1, \quad m_{23} = 0.9 \]

and the the border values be given as follows
\[ I_1 = 0, \quad I_2 = 12, \quad I_3 = 24, \quad I_4 = 30, \quad J_1 = 0, \quad J_2 = 10, \quad J_3 = 20 \quad \text{and} \quad J_4 = 30. \]

A plot of \( F \) with respect to the above specified normat is shown in Fig 5.15.

As another example, now let the basifactors be defined as follows
\[ b_{11} = 1.2, \quad b_{21} = 1.6, \]
\[ b_{12} = 4.0, \quad b_{22} = 1.8, \]
\[ b_{13} = 1.4, \quad b_{23} = 1.1, \]
\[ m_{11} = 1.05, \quad m_{21} = 0.95, \]
\[ m_{12} = 0.95, \quad m_{22} = 1.05 \]

and
\[ m_{13} = 1.15, \quad m_{23} = 0.85 \]

but keeping the border values as before, a plot of \( F \) with respect to this retronorm is shown in Fig 5.16.
The above defined retronorms are three examples from a class of retronorms referred to as 'dexiant retronorms'. One can simply state that the dexiant retronorms are compound retronorms and are obtained by mixing various basiant and/or metriant retronorms together to pave a given region.

The retronorms of Figs 5.1 and 5.13 were defined in terms of a two dimensional Cartesian coordinate system and are referred to as 'dexibifect retronorms'. The retronorm of Fig 5.6 was defined in terms of a polar coordinate system, thus is referred to as a 'dexipolar retronorm'.

In order to distinguish between various types of dexiant retronorms, one can refer to a retronorm by the number of zones present in each of the directions in which the retronorm is defined followed by its standard name as explained above. For example, the first retronorm defined above, may be referred to as a 3x1 or 3 by 1 dexibifect retronorm. It consists of three zones in the first direction and one in the second direction. The second retronorm is referred to as a 1x3 dexipolar retronorm and finally the third one is referred to as a 3x3 dexibifect retronorm.

In order to distinguish between various basifactors (or metrifactors) of a dexiant retronorm the following notation is employed:

A basifactor (or metrifactor) in general, may be written as follows

\[ b(i)(j)(k)(n) \]

where the integer variable i refers to the direction along which the basifactor is specified. Integer variables j, k and n specify the position of the region (for which the basifactor is being specified) with respect to the 1st, 2nd and 3rd directions, respectively. For a dexiunifect
retronorm a basifactor may be written as \( b(i)(j) \) and for a dexibifect retronorm it may be written as \( b(i)(j)(k) \).

If the integer variable consists of a single letter or if it is a single integer number then the parentheses may be removed. Thus for a dexitrifect retronorm a basifactor may simply be written as \( bijkn \). For example

\[ b1(12)14 \]

refers to the basifactor along the first direction of a region which is situated on the 12th zone along the first direction and on the 1st zone along the second direction and 4th zone along the third direction. For a dexibifect retronorm a basifactor may be written as \( bijk \). For example

\[ b241 \]

refers to the basifactor along the second direction for a region situated on the 4th zone along the first direction and 1st zone along the second direction, respectively.

The above notation is general and any basifactor or metrifactor can be referred to using the above notation.

An convenient way in which the information such as those shown in Table 5.1 can be provided is through an array which is termed 'dexiarray'. A dexiarray can be partitioned into three parts as shown below

\[
\begin{bmatrix}
B \\
M \\
N
\end{bmatrix}
\]

Part \( B \) contains the basifactors, part \( M \) contains the metrifactors and part \( N \) contains the border values of the regions. The number of rows in \( B \) (or \( M \)) depends on the
number of basifactors (or metrifactors) of each region. If the region is paved by a unifect retronorm then $B$ (or $M$) has one row. If the region is paved by a bifect or polar retronorm, then $B$ (or $M$) has two rows. Finally, if the region is paved by a trifect, cylindrical or spherical retronorm then $B$ (or $M$) has three rows. The number of columns is equal to the number of regions and each column of the dexiarray refers to a particular region. Thus, the dexiarray for the first example may be written as follows:

$$
\begin{bmatrix}
0.9 & 2.4 & 1.3 \\
1.5 & 1.5 & 1.5 \\
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
0 & 6 & 15 \\
6 & 15 & 20 \\
0 & 0 & 0 \\
20 & 20 & 20 \\
\end{bmatrix}
$$

The above array contains all the information necessary to construct the normat for the Fig 5.3. Every element of the above matrix is referred to as a dexifactor. Subarrays $B$ and $M$ are real numbers but $N$ are integer numbers. Each column of the dexiarray relates to a region. The dexiarray for the normat of Fig 5.1 which is a 3x1 dexibifect retronorm, consists of three columns. Its first two rows contain the basifactors. There are six basifactors, two for each region. As there is only one zone in the second direction, the basifactors along the second direction for all three region are equal. The metrifactors are given in rows 3 and 4 and all are considered to be 1.0.

In the case of Fig 5.4, for which only the basifactors differ from that of Fig 5.3 the subarray $B$ is as follows.
Also, for Fig 5.5 the subarray B is as follows

\[
\begin{bmatrix}
1.75 & 0.5 & 2.5 \\
1.8 & 1.8 & 1.8
\end{bmatrix}
\]

The dextraarray for the normat of Fig 5.8 may be written as follows

\[
\begin{bmatrix}
2.5 & 2.5 & 2.5 \\
7.0 & 18.0 & 3.0 \\
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
0 & 0 & 0 \\
20 & 20 & 20 \\
0 & 12 & 18 \\
12 & 18 & 30
\end{bmatrix}
\]

For Fig 5.9 the metri factors and border values are as above but the subarray B is as follows

\[
\begin{bmatrix}
3.0 & 3.0 & 3.0 \\
20.0 & 3.0 & 8.0
\end{bmatrix}
\]

The dextraarray for Fig 5.10 which defines a 1 by 3 dextripolar retronorm is as follows.
The dexiarray for Fig 5.12 which defines a 1 by 3 dexipolar retronorm may be written as follows:

\[
\begin{bmatrix}
4.0 & 4.0 & 4.0 \\
20.0 & 3.0 & 8.0 \\
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
0 & 0 & 0 \\
20 & 20 & 20 \\
0 & 12 & 15 \\
10 & 15 & 30 \\
\end{bmatrix}
\]

The information shown in Table 5.1 which relate to Fig 5.15 can be written in terms of a dexiarray (defining a 3 by 3 dexibifect retronorm) as follows:

\[
\begin{bmatrix}
3.0 & 3.0 & 3.0 \\
5.0 & 19.0 & 10.0 \\
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
0 & 0 & 0 \\
24 & 24 & 24 \\
0 & 10 & 20 \\
10 & 20 & 32 \\
\end{bmatrix}
\]
The dexiarray of Fig 5.16 is the same as that of Fig 5.15, except for the subarrays B and M which are as follows:

\[
\begin{bmatrix}
2.5 & 1.5 & 4.0 & 2.5 & 1.5 & 4.0 & 2.5 & 1.5 & 4.0 \\
2.0 & 2.0 & 2.0 & 1.5 & 1.5 & 1.5 & 3.0 & 3.0 & 3.0 \\
0.95 & 1.0 & 1.1 & 0.95 & 1.0 & 1.1 & 0.95 & 1.0 & 1.1 \\
1.1 & 1.1 & 1.1 & 1.0 & 1.0 & 1.0 & 0.9 & 0.9 & 0.9 \\
0 & 12 & 24 & 0 & 12 & 24 & 0 & 12 & 24 \\
12 & 24 & 36 & 12 & 24 & 36 & 12 & 24 & 36 \\
0 & 0 & 0 & 10 & 10 & 10 & 20 & 20 & 20 \\
10 & 10 & 10 & 20 & 20 & 20 & 30 & 30 & 30 \\
\end{bmatrix}
\]

One can extend the above idea to cover all the standard retronorms introduced in the previous Chapter. For example, Fig 5.17 shows a plot of a formex F with respect to a 3x3x3 dexicylindrical retronorm with the following dexisubarray B, M and N, where:

\[
\begin{bmatrix}
1.2 & 2.8 & 1.4 & 1.2 & 2.8 & 1.4 & 1.2 & 2.8 & 1.4 \\
1.6 & 1.6 & 1.6 & 1.8 & 1.8 & 1.8 & 1.1 & 1.1 & 1.1 \\
1.05 & 0.95 & 1.1 & 1.05 & 0.95 & 1.1 & 1.05 & 0.95 & 1.1 \\
0.95 & 0.95 & 0.95 & 1.05 & 1.05 & 1.05 & 0.8 & 0.8 & 0.8 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1.5 & 2.0 & 2.0 & 1.5 & 2.0 & 2.0 & 1.5 & 2.0 & 2.0 \\
1.5 & 1.5 & 1.5 & 2.0 & 2.0 & 2.0 & 2.5 & 2.5 & 2.5 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
.85 & .90 & 1.0 & .85 & .90 & 1.0 & .85 & .90 & 1.0 \\
1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
.90 & .90 & .90 & 1.0 & 1.0 & 1.0 & 1.5 & 1.5 & 1.5 \\
\end{bmatrix}
\]
Fig 5.17

Fig 5.18
and where formex $F$ is given by

$$F = \text{rinad}(7, 12, 20, 1, 1, 1) \upharpoonright G$$

where

$$G = \begin{bmatrix}
10, 0, 0; 11, 0, 0; 11, 0, 1; 10, 0, 1; \\
10, 1, 0; 11, 1, 0; 11, 1, 1; 10, 1, 1.
\end{bmatrix}$$

In the case of Fig 5.18, for which the basifactors and metrifactors differ from that of Fig 5.17 the subarray $B$ and $M$ are as follows

$$B = \begin{bmatrix}
1.5 & 4.0 & 6.0 & 1.5 & 4.0 & 6.0 & 1.5 & 4.0 & 6.0 \\
7.0 & 7.0 & 7.0 & 15. & 15. & 15. & 7.0 & 7.0 & 7.0 \\
8.0 & 8.0 & 8.0 & 1.5 & 1.5 & 1.5 & 4.0 & 4.0 & 4.0
\end{bmatrix}$$

$$M = \begin{bmatrix}
1.15 & 1.1 & 1.0 & 1.15 & 1.1 & 1.0 & 1.15 & 1.1 & 1.0 \\
1.1 & 1.1 & 1.1 & 1.0 & 1.0 & 1.0 & 0.9 & 0.9 & 0.9 \\
0.9 & 0.9 & 0.9 & 1.0 & 1.0 & 1.0 & 1.1 & 1.1 & 1.1
\end{bmatrix}$$

O'Donoghue et al [5.1] used meshes similar to the ones shown for analysis of pressure vessels with multiple surface cracks.

The simple dextrant retronorms discussed so far can be used to solve the problem of mesh generation for a large number of standard problems in finite elements.

For example consider a typical stress discontinuity problem in a semi-infinite plate under uniform pressure $p$ over the region $-a < x < a$ as shown in Fig 5.19. Stress discontinuities
exists at the ends, A and A', of the loaded region. Figure 5.20 shows a finite element mesh for this problem which is similar to that used by Whitcomb et al [5.2].

Now let a formex $F$ be formulated as follows

$$F = \text{rinid}(20,20,2,2) \setminus G$$

where

$$G = [0,0; 1,0; 2,0; 2,1; 2,2; 1, \bar{y}; 0,2; 0,1]$$

represents an 8-noded isoparametric finite element.

A paribifect N-plot of $F$ is shown in Fig 5.21. A plot of this formex with respect to a 2x1 dexibifect retronorm with the following dexiarray

$$\begin{bmatrix}
2.0 & 0.6 \\
3.0 & 3.0 \\
0.8 & 1.2 \\
0.92 & 0.92 \\
0 & 14 \\
14 & 40 \\
0 & 0 \\
40 & 40
\end{bmatrix}$$

would represent the mesh shown in Fig 5.20.

Now, let the above formex be plotted with respect to a 1x2 dexipolar retronorm. A different mesh pattern is obtained in this case. F. Medina et al [5.3] used such a mesh in their analysis of this type of stress discontinuity problem. A plot of $F$ with respect to a 1x2 dexipolar retronorm with the following dexiarray
would represent such a mesh and is shown in Fig 5.22.

As another example consider the problem of interlaminar stress singularities at a straight free edge in composite laminates. Details of the specimen under investigation is shown in Fig 5.23. Figure 5.23(a) shows a long, symmetric laminate loaded in the z-direction. The laminate has a width of $2a$ and has four piles, each of thickness $h$. Because of steep stress gradients near the free edge a stress singularity exists at the interface corner. Because of the symmetries in the problem, only the shaded region ($0 \leq x \leq a; 0 \leq y \leq 2h$) shown in Fig 5.23(b) of an $z = \text{constant}$ plane was considered. Whitcomb [5.2] used three mesh patterns ranging from coarse mesh to fine mesh similar to those shown in Figs 5.24-5.26.

One can use the generic formulation approach here and write down a formulation for the above three meshes in terms of one or more parameters. Such a formulation may be written as follows:

$$
MCL(m,n) = F1 \# F2
$$

where

$$
F1 = \text{rinid}(m,n,2,2) | G,
$$

$$
F2 = \text{tran}(1,n) | F1
$$

and
Fig 5.23

(a)-Four-ply laminate

(b)-An z=constant plane

region under investigation
free edge

interface

Fig 5.23

Fig 5.24
Fl and F2 are formex formulations of finite element meshes in the composite laminates and MCL is the name given to the formulation, standing for Mesh for Composite Laminate. Subsequent to the above generic formulation being defined, a series of meshes ranging from coarse to fine can be generated by specifying various values for m and n. Each of these meshes would be drawn with respect to a retronorm. For each retronorm one has to provide a dexiarray. Here again the generic approach can be used and the dexiarrays can be defined generically in terms of one or more parameters.

The dexiarray in general, for a j by k bifect retronorm can be written generically as follows

$$G = [0,0; 1,0; 2,0; 2,1; 2,2; 1,2; 0,2; 0,1].$$

$$DGF(b_111,b_121,\ldots,b_{1j1},
b_{211},b_{212},\ldots,b_{21k},
m_{111},m_{121},\ldots,m_{1j1},
m_{211},m_{212},\ldots,m_{21k},
N_1,N_2,\ldots,N(j+1),M_1,M_2,\ldots,M(k+1)) =$$

$$\begin{bmatrix}
b_{111} & b_{121} & b_{1j1} & b_{111} & b_{121} & b_{1j1} \\
b_{211} & b_{211} & b_{212} & b_{212} & b_{212} & b_{21k} & b_{21k} \\
m_{111} & m_{121} & m_{1j1} & m_{111} & m_{121} & m_{1j1} & m_{111} & m_{121} & m_{1j1} \\
m_{211} & m_{211} & m_{212} & m_{212} & m_{212} & m_{21k} & m_{21k} & m_{21k} \\
N_1 & N_2 & N(j) & N_1 & N_2 & N(j) & N_1 & N_2 & N(j) \\
N_2 & N_3 & N(j+1) & N_2 & N_3 & N(j+1) & N_2 & N_3 & N(j+1) \\
M_1 & M_2 & M(k) & M_1 & M_2 & M(k) & M_1 & M_2 & M(k) \\
M_2 & M_3 & M(k+1) & M_2 & M_3 & M(k+1) & M_2 & M_3 & M(k+1)
\end{bmatrix}$$

in which

$$b_{111},b_{121},\ldots,b_{1j1}$$

and
ml11, ml21, ..., mlj1
represent the basifactors and metrifactors along the first
direction. The basifactors and metrifactors along the second
direction are represented by
    b211, b212, ..., b21k
and
    m211, m212, ..., m21k
respectively. Finally
    N1, N2, ..., N(j+1)
and
    M1, M2, ..., M(k+1)
represent the borders values for the first and second
directions, respectively. DGF is the name given to the
dexiarray (standing for the Dexiarray in Generic Form) and the
parameter list specifies the order in which the parameters
are to be given.

For instance
    MCL(4, 2)
would represent the mesh shown in Fig 5.24, plotted with
respect to a 3x3 dexibifect retronorm defined through a
dexiarray specified as follows

DCL(15.0, 7.5, 3.5,
    7.5, 3.5, 7.5,
    1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
    0, 4, 12, 16, 0, 2, 6, 8).

The mesh patterns for MCL(8, 4) and MCL(32, 8) are shown in
Figs. 5.25 and 5.26, respectively. These are drawn with
respect to 3x3 dexibifect retronorms. As all the
metrifactors are equal to 1.0, they are not needed to be
specified in the parameter list. Thus the dexiarrays in
these cases are, respectively, specified in a simplified form
as follows
DCL(8.0,4.0,2.0,4.0,2.0,4.0,0,6,22,32,0,4,12,16)

and

DCL(5.0,2.5,0.75,2.0,1.0,2.0,0,10,32,64,0,8,24,32).

As another example of the generic formulation approach, consider a deep beam with cutout as shown in Fig 5.27. Mohr and Milner [5.4] who investigated this problem assumed an elastic plane state of stress for the web of the beam and an elastic uniaxial state of stress for the flanges. The web of the beam was then idealized using rectangular plane stress elements with thickness of 0.31 in. The flanges were modelled as bar elements having a cross-sectional area of 4.34 in².

Figures 5.29 and 5.30 show two finite element models similar to those used by Mohr and Milner. The generic formulation of a number of meshes for this problem may be written as follows

\[ \text{DBC}(n_1, n_2, n_3, n_4) = F_1 \# F_2 \# F_3 \]

where DBC is the name given to the formulation and stands for Deep Beam with Cutout.

The axial bar elements which are represented by \( F_1 \) are formulated as follows

\[ F_1 = \text{rinid}(n_1, 2, 1, 2n_2)[0, 0; 1, 0] \]

where the cantle \([0, 0; 1, 0]\) represents a bar element.
Fig 5.27

Fig 5.28

U2

U1

flange

cutout

web

237
The rectangular elements of the mesh are represented by $F_2$ and $F_3$ and are formulated as

\[
F_2 = \text{rinid}(n_1,n_3,1,1); G_1,
\]
\[
F_3 = \text{tran}(2,n_3); \text{rinid}(n_4;n_2,1,1); G_1
\]

and

\[
G_1 = [0,0; 1,0; 1,1; 0,1]
\]

where $G_1$ represents a rectangular plane stress element.

Now, let a formex $E_1$ be defined as

\[
E_1 = \text{DBC}(34,20,10,25).
\]

$E_1$ represents a mesh consisting of 68 axial bar elements and 1180 rectangular plane stress elements. A paribifect $N$-plot of $E_1$ is shown in Fig 5.28. The generic form of the dexiarray in this case is similar to the above example except that it has 12 columns. Figure 5.29 shows a plot of $E_1$ drawn with respect to a 3x4 dexibifect retronorm defined through a dexiarray as follows

\[
\text{DDBC}(4.0,3.0,0.4,3.0,0.4,0.4,1.0,0.85,1.3,0.8,1.25,0.83,1.25,0,15,25,34,0,10,20,30,40)
\]

where DDBC stands for the Dexiarray for Deep Beam with Cutout.

Now, let a formex $E_2$ be defined as

\[
E_2 = \text{DBC}(43,22,11,31).
\]

$E_2$ represents a mesh consisting of 86 axial bar elements and 1628 rectangular plane stress elements. Figure 5.30 shows a plot of $E_2$ with respect to a 3x4 dexibifect retronorm defined
Fig 5.29

Fig 5.30
through a dexiarray which is specified as follows

\[
\text{DDBC}(3.5,2.5,0.3, 2.5,0.3,2.5,0.3, 1.0,0.85,1.3,0.8,1.25,0.83,1.25, 0,18,31,43,0,11,22,33,44).
\]

As another example consider the problem of flow in a cavity with an obstruction. Figure 5.31 shows a rectangular cavity with a square obstruction. Mizukami \[5.5\] used a mesh similar to that shown in Fig 5.32 employing linear triangular elements. This mesh can be formulated as follows

\[
\begin{align*}
G &= [5,0; 6,0; 6,1] \\
G_1 &= G \# \text{refid}(5,5,.5); G, \\
F_1 &= \text{rinid}(26,7,1,1); G_1, \\
F_2 &= \text{rinid}(12,23,1,1); \text{tran}(2,7); G_1, \\
F_3 &= \text{rinid}(7,7,1,1); \text{tranid}(19,7); G_1, \\
F_4 &= \text{rinid}(14,16,1,1); \text{tranid}(12,14); G_1, \\
F_5 &= \text{lam}(1,18); \text{rinid}(5,4,1,1); \text{tranid}(-5,26); G_1
\end{align*}
\]

and finally

\[
F = F_1 \# F_2 \# F_3 \# F_4 \# F_5.
\]

A plot of \( F \) with respect to a 9x7 dexibifect retronorm defined through a dexiarray which is specified as follows

\[
\text{DCO}(2.0,1.0,2.0,1.0,2.0,1.0,2.0,1.0,2.0, 1.0,2.0,1.0,2.0,1.0,2.0,1.0, 0,3,7,15,19,22,26,29,33,36,0,2,5,9,12,16,24,30)
\]

would represent the mesh shown in Fig 5.32, where DCO stands for the Dexiarray of Cavity with Obstruction. As all the metrifactors are equal to 1.0, they are not included in the parameter list.
5.3 FURTHER EXAMPLE ON DEXIANT RETRONORMS

In all the dexiant retronorms considered so far, the basifactors were defined for a given region and were constant within that region. For example if there is a 3x1 dexibifect retronorm six basifactors are required to be specified, a pair for each region. There are situations when within a given region the basifactors are not constant. As an example, Consider two non-parallel line L11 and L12. Let it be required to define a normat between these two lines, where L11 and L12 are considered to be normat lines. Let it be assumed that along L11 U1=IA and along L12 U1=IB, where IA and IB are integer values. Interpolating linearly in the U1 direction between L11 and L12 will give rise to a series of normat lines shown dotted in Fig 5.33.

Now, consider a region of this normat which is bounded by two normat lines L21 and L22 parallel to x axis (see Fig 5.34). Let it be assumed that along L21 U2=IC and along L22 U2=ID. Interpolating linearly in the U2 direction between L21 and L22 will give rise to a series of normat lines which are parallel to the x axis. This creates a bidirectional normat in the region bounded by the lines L11 to L22. The same procedure may be carried out if L21 and L22 were parallel to the y axis.

The dexiarray in this type of retronorm is similar to that of the simple dexiant retronorms. The only difference being that the subarray B instead of containing the basifactors, contains the information describing the borders of the region under consideration. As in the example under consideration the borders are straight lines only the coordinates of the four corners of the region are required to be specified.

In the case of the above example the information necessary to carry out the linear interpolation can be provided through a dexiarray which its transpose is
\[ [x_1 \ x_2 \ x_3 \ x_4 \ \ y_1 \ y_2 \ y_3 \ y_4 \ 1.0 \ 1.0 \ \ IA \ IB \ IC \ ID] \]

where \( x_1 \) to \( x_4 \) and \( y_1 \) to \( y_4 \) are the coordinates of the four corners of the region, \( IA \) to \( ID \) are border values of the region and metrifactors in this example are considered to be 1.0.

Consider a formex \( F \), a paribifect N-plot of which is shown in Fig 5.35. A plot of \( F \) with respect to the normat of Fig 5.34 is given in Fig 5.36.

Now let it be assumed that there are a series of lines \( L_{11}, L_{12}, L_{13}, L_{14} \) and \( L_{15} \) which are intersected by two lines \( L_{21} \) and \( L_{22} \) parallel to the x axis (see Fig 5.37). Using the above approach and linearly interpolating between each pair of consecutive lines would produce a series of normat lines between \( L_{11} \) and \( L_{15} \) matching \( L_{12}, \ L_{13} \) and \( L_{14} \) exactly. Furthermore, interpolating linearly between \( L_{21} \) and \( L_{22} \) creates a series of lines parallel to the x axis. Thus a normat is defined between lines \( L_{11} \) to \( L_{15} \) and \( L_{21} \) to \( L_{22} \) where \( L_{11} \) to \( L_{15} \) and \( L_{21} \) to \( L_{22} \) are normat lines.

As an example, consider a formex \( E \) a paribifect N-plot of which is shown in Fig 5.38. A plot of \( E \) with respect to the above defined normat is given in Fig 5.39.

As another example consider the mesh shown in Fig 5.40. Let a formex \( F \) be formulated as follows

\[
F = \text{rinid}(16,10,1,1)\mid G
\]

where

\[
G = G_1 \ # \ \text{refid}(1/2,1/2)\mid G_1
\]

and

\[
G_1 = [0,0; 1,0; 1,1].
\]

A plot of \( F \) with respect to a 3-zoned dexibifect retronorm
Fig 5.37

Fig 5.38
with the following dextrarray

\[
\begin{bmatrix}
 x_{c1} & x_{c2} & x_{c3} \\
 y_{c1} & y_{c2} & y_{c3} \\
 1.0 & 1.0 & 1.0 \\
 1.0 & 1.0 & 1.0 \\
 0 & 6 & 10 \\
 6 & 10 & 16 \\
 0 & 0 & 0 \\
 10 & 10 & 10 \\
\end{bmatrix}
\]

would represent the mesh of Fig 5.40. The entities \( x_c \) and \( y_c \) represent the coordinates of the corners of each region and as the mesh density is constant within each region the metrifactors are considered to be 1.0.

Now consider a region bounded by four non-parallel lines \( L_{11}, L_{12}, L_{21} \) and \( L_{22} \), as shown in Fig 5.41. Here again by interpolating linearly between each pair of opposite lines (i.e. \( <L_{11}, L_{12}> \) and \( <L_{21}, L_{22}> \)) a normat can be defined inside the bounded region. Each of the lines \( L_{11}, L_{12}, L_{21} \) and \( L_{22} \) are considered to be normat lines (i.e. along \( L_{11} \), \( U_1=I_A \), \( I_C \leq U_2 \leq I_D \), along \( L_{12} \), \( U_1=I_B \), \( I_C \leq U_2 \leq I_D \), along \( L_{21} \), \( U_2=I_C \), \( I_A \leq U_1 \leq I_B \) and along \( L_{22} \), \( U_2=I_D \), \( I_A \leq U_1 \leq I_B \)).

A formex whose parabifect N-plot is shown in Fig 5.42 is plotted with respect to the above normat and its plot is shown in Fig 5.43.

As an extension of the above example, consider the regions given in Fig 5.44. It consists of 9 regions formed from intersection of 8 broken lines. Each of these lines is considered to be a normat line. Line \( L_{11} \) is considered to be a normat line along which \( U_1=I_1 \) and \( J_1 \leq U_2 \leq J_2 \), etc.
Each region is bounded by four straight line segments. A normat can be defined within each region by interpolating between its border lines. Performing the interpolation outlined in the above example within each region, covering all the regions one by one, a normat may be defined over all the regions.

Now, consider a formex $G$, whose paribifect $N$-plot is shown in Fig 5.45. A plot of $G$ with respect to the above normat is shown in Fig 5.46.

In the type of dexiant retronorm introduced above, the basifactors are not specified for a given region. Instead the coordinates of the borders of each region are provided. It can be seen that a retronorm specified in this way can have any arbitrary shape. In other words the above idea can be extended to specify a retronorm in a way that its normat lines match given borders exactly. Also, in order to specify a normat for a complex region, one can break it into a few simple regions and then define a retronorm within each region.

As another example consider the region shown in Fig 5.47, which is a square region with a central square hole. Let it be assumed that it is required to produce a finite element mesh for the above region. The region can be divided to 4 zones as shown in Fig 5.47 by the thick dotted lines. A normat can then be defined within each zone by interpolating between its borders as shown in Fig 5.47.

The dexiaarray defining the above normat can be written as
where $x_{cl}$ and $y_{cl}$ represent the $x$ and $y$ coordinates of the corners of the first region, etc.

Now consider a formex $F_l$, where

$$F_l = \text{rinid}(16,4,2,2) \upharpoonright G$$

and where

$$G = \text{rosid}(1,1) \upharpoonright [0,0; 2,0; 1,1].$$

From a purely graphical point of view, one may regard the plot of $F_l$ drawn with respect to the normat of Fig 5.47 as representing the interconnection pattern of the finite element mesh shown in Fig 5.48. However if $F_l$ is to serve as data describing the interconnection pattern of the finite element mesh shown, then one must modify $F_l$ to take account of the connection between the elements indicated by asterisks correctly. Modification of this type can be achieved as follows

$$F = \text{nov}(E) \upharpoonright F_l$$

where

$$E = \text{rin}(2,5,1) \upharpoonright [[32,0; 0,0]].$$

A plot of $F$ with respect to the above defined normat is shown in Fig 5.48. It represents a finite element mesh which is free from discontinuities at nodes. Fujii et al [5.6] used a
mesh similar to the one shown.

As another example consider a typical crack tip problem. The specimen under consideration here is a square plate with a side crack. Due to symmetry only half of the plate is considered. Around the crack tip where large stress gradients are expected very small elements are needed, whereas in the region where the stress gradients are not so high larger elements are employed. Firstly the structure is divided into a number of zones. An appropriate pattern for the normal lines is as shown in Fig 5.49, where a set of normal lines are needed to cross through the crack tip. A 10 regioned dextibifect retronorm is defined for the structure and is shown in dotted lines in Fig 5.49. The elements used in this example are 4-noded rectangular elements. These elements at the crack tip degenerate to 4-noded triangular elements.

Now a formex F whose paribifect N-plot is shown in Fig 5.50 may be written as follows

\[
G = [0,0; 1,0; 1,1; 0,1],
\]

\[
F_1 = \text{rinid}(16,8,1,1)\mid G,
\]

\[
F_2 = \text{rinid}(8,4,1,1)\mid \text{tranid}(8,8)\mid G
\]

and \[ F = F_1 \# F_2. \]

A plot of F with respect to the above defined retronorm is shown in Fig 5.51. Hammel, et al [5.7] used a mesh similar to the one shown in elastic-plastic finite element analysis of crack-tip fractured specimen. One can use a higher order element such as 8-noded isoparametric element in the formulation by simply changing the formex G (which represents a typical element of the mesh).

In all the dextiant retronorms considered so far all the region were divided into zones which were bounded by straight
Fig 5.49

Fig 5.50
Fig 5.51

Fig 5.52
lines. One can use the same approach and deal with curved regions. In defining a dexiant retronorm for a region with curved borders the first subarray of the dextr array (i.e. B) should provide the information necessary to describe the borders of that region.

An acceptable and relatively easy way to describe the borders of a region is by parabolic functions. One can provide three points on a parabola at appropriate locations to describe a particular geometry. These points are referred to as 'key normat points'. The subarray B contains the coordinates of these key normat points.

Describing border curves by quadratic polynomial means that points of inflection and slope discontinuities cannot be incorporated in the borders of a single zone. Thus in order to model a complex region multiple zones are required.

Interpolating between two pairs of border curves of a zone produces a series of normat lines defining a normat within that region. If the borders of a region cannot be exactly described by quadratic polynomials then these normat lines would match the borders at only a finite number of points, i.e. key normat points and a geometric error is introduced by this type of interpolation.

As an example consider the problem of steady-state conduction in a cylinder with eccentric circular hole. Let it be assumed that the temperature at the inner and outer surfaces of the cylinder are prescribed to be at 1000 K and 0 K. Noor et al [5.8] carried out a two-dimensional steady state thermal analysis of this problem. A finite element model using nine-noded isoparametric thermal elements similar to that used by Noor et al is shown in Fig 5.53. In order to model the cross-section of the cylinder with a mesh the structure is divided into 4 zones as shown in Fig 5.52. The boundaries of each zone are then modelled by parabolic curves. The
Fig 5.53

Fig 5.54
coordinates of the points which are shown by solid circles are used to describe the borders and the subsequent interpolation.

Now consider a formex F1 where

\[ F1 = \text{rinid}(10, 40, 2, 2) \mid G \]

and

\[ G = [0, 0; 1, 0; 2, 1; 2, 2; 1, 2; 0, 2; 0, 1; 1, 1] \]

and where G represents a nine-noded isoparametric element.

The information necessary to transform the formex F1 into a mesh suitable for the above example is specified through the dextarray

\[
\begin{bmatrix}
X_{knpl} & X_{knp2} & X_{knp3} & X_{knp4} \\
Y_{knpl} & Y_{knp2} & Y_{knp3} & Y_{knp4} \\
1.0 & 1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 & 1.0 \\
0 & 0 & 0 & 0 \\
10 & 10 & 10 & 10 \\
0 & 20 & 40 & 60 \\
20 & 40 & 60 & 80 \\
\end{bmatrix}
\]

where X_{knp} and Y_{knp} represent the x and y coordinates of the key normal points of the regions 1 to 4.

From a purely graphical point of view, one may regard F1 as representing the interconnection pattern of the finite element model shown in Fig 5.53. However if F1 is to serve as data describing the interconnection pattern of this finite element model, then one must modify F1 to take account of the connection between the elements indicated by asterisks correctly. Modification of this type can be achieved as
follows

\[ F = \text{nov}(E) \mid F_1 \]

where

\[ E = \text{rin}(2,20,1) \mid \{[80,0; 0,0] \}. \]

A plot of \( F \) with respect to a retronorm defined by the above dexiarray is shown in Fig 5.53 and is free from discontinuities at nodes.

As another example consider a square plate with a central circular hole under direct tensile force. Due to symmetry only quarter of the plate is needed to be considered and this is shown in Fig 5.54. The plates is divided into three zones as shown. The circular edge is modelled more accurately by two parabolic curve segments than one. Interpolating between the border curves of each zone creates a normat within that region. This interpolation can be carried out using the dexiarray

\[
\begin{bmatrix}
X_{knpl} & X_{knp2} & X_{knp3} \\
Y_{knpl} & Y_{knp2} & Y_{knp3} \\
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
0 & 0 & 20 \\
20 & 20 & 30 \\
20 & 0 & 0 \\
40 & 20 & 20
\end{bmatrix}
\]

Thus a 3 zoned dexibifect retronorm is defined for this problem.

Now, consider a formex \( F \), where

\[ F = F_1 \# F_2 \]
Fig 5.55

Fig 5.56
and where
\[ F_1 = \text{rinid}(10,20,1,1)' G, \]
\[ F_2 = \text{rinid}(5,10,1,1)' \text{tran}(1,20)' G \]
and
\[ G = \begin{bmatrix} 0 & 0; 1 & 0; 2 & 0; 2 & 1; 2 & 2; 1 & 2; 0 & 2; 0 & 1 \end{bmatrix}. \]

A paribifect N-plot of \( F \) is shown in Fig 5.55 and a plot of \( F \) with respect to a normat defined through the above dexiarray is shown in Fig 5.56. Goffe [5.9] used a mesh similar to the one shown.

As another example consider a gear tooth subjected to a concentrated load. The region under consideration is divided into 5 zones as shown in Fig 5.57. Interpolating between the borders of each zone creates a normat within that zone. The interpolation can be carried out using the dexiarray

\[
\begin{bmatrix}
X_{kn1} & X_{kn2} & X_{kn3} & X_{kn4} & X_{kn5} \\
Y_{kn1} & Y_{kn2} & Y_{kn3} & Y_{kn4} & Y_{kn5} \\
1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
2 & 2 & 2 & 0 & 26 \\
26 & 26 & 26 & 2 & 28 \\
11 & 6 & 0 & 0 & 0 \\
17 & 11 & 6 & 6 & 6 \\
\end{bmatrix}
\]

Now, let a formex \( F \) whose paribifect N-plot is shown in Fig 5.58 be defined as follows

\[ F = R \# T \]

where \( R \) represents all the rectangular elements and is formulated as follows
\[ R_1 = [0,0; 1,0; 1,2; 0,2], \]
\[ R_2 = \text{rinid}(2,2,1,2)|R_1, \]
\[ R_3 = \text{tran}(2,4)|R_1, \]
\[ R_4 = \text{lam}(1,14)|\{ R_1 \# R_2 \}, \]
\[ R_5 = \text{rinid}(4,2,6,2)|[2,0; 8,0; 8,2; 2,2], \]
\[ R_6 = [2,6; 2,7; 5,7; 5,6], \]
\[ R_7 = \text{rinid}(18,4,3,1)|R_6, \]
\[ R_8 = \text{rinid}(6,4,4,1)|[2,11; 6,11; 6,12; 2,12], \]
\[ R_9 = \text{rin}(1,8,3)|\text{tran}(2,10)|R_6, \]
and
\[ R = R_9 \# R_8 \# R_7 \# R_5 \# R_4. \]

The triangular elements are represented by \( T \) and are formulated as follows

\[ T_1 = [1,4; 2,5; 1,6], \]
\[ T_2 = \text{lam}(2,5)|[1,4; 2,4; 2,5], \]
\[ T_3 = \text{lam}(1,14)|\{ T_1 \# T_2 \}, \]
\[ T_4 = \text{lam}(1,8)|[[2,4; 8,4; 6,5], [2,4; 2,5; 6,5]] \]
\[ T_5 = T_3 \# T_4 \# [8,4; 6,5; 10,5], \]
\[ T_6 = [[2,5; 2,6; 5,6], [2,5; 5,6; 6,5], \]
\[ [6,5; 5,6; 8,6]], \]
\[ T_7 = \text{lam}(1,8)|T_6 \# [6,5; 8,6; 10,5], \]
\[ T_8 = [[10,5; 11,6; 14,5], [8,6; 10,5; 11,6], \]
\[ [14,4; 14,5; 20,4], [20,4; 18,5; 14,5], \]
\[ [18,5; 22,5; 26,4], [22,5; 26,4; 26,5], \]
\[ [18,5; 20,4; 26,4]], \]
\[ T_9 = [[14,6; 14,5; 18,5], [18,5; 22,5; 17,6], \]
\[ [22,5; 26,5; 20,6], [14,6; 17,6; 18,5], \]
\[ [20,6; 23,6; 26,5], [23,6; 26,6; 26,5], \]
\[ [17,6; 22,5; 20,6]], \]
\[ T_{10} = T_9 \# T_8 \# T_7 \# T_5, \]
\[ T_{11} = T_9 \# T_7, \]
\[ T_{12} = \text{tran}(2,9)|T_{11}, \]
\[ T_{13} = \text{refid}(14,8)|T_{12}, \]
and
\[ T = T_{10} \# T_{12} \# T_{13}. \]
A plot of $F$ with respect to the normat defined through the above dexiarray is shown in Fig 5.59. Mc Comber, et al [5.10] used a mesh similar to the one shown.

As another example consider the region shown in Fig 5.60 [5.11]. It is divided into 14 zones. Now, let a formex $F$ a paribifect N-plot of which is shown in Fig 5.61 be formulated as follows

$$G = [0, 2; 1, 2; 1, 3; 0, 3],$$

$$G_1 = \text{rin}(52, 4, 1, 1)\{G,$$

$$G_2 = \text{rin}(4, 2, 1, 1)\{G,$$

$$G_3 = \text{rin}(1, 2, 2)\{\text{tran}(22, -2)\{G_2,$$

$$G_4 = \text{rin}(4, 4, 1, 1)\{G,$$

$$G_5 = \text{rin}(1, 4, 9)\{\text{tran}(9, 4)\{G_4$$

and

$$F_1 = G_5 \# G_3 \# G_1.$$ 

From a purely graphical point of view, one may regard a plot of $F_1$ drawn with respect to a 14 zoned dexibifect retronorm as representing the interconnection pattern of the finite element mesh of Fig 5.62. However if $F_1$ is to serve as data describing the interconnection pattern of this finite element model, then one must modify $F_1$ to take account of the connection between the elements indicated by asterisks correctly. Modification of this type can be achieved as follows

$$F = \text{nov}(E)\{F_1$$

where

$$E = \text{rin}(2, 4, 1)\{[[52, 2; 0, 2]].$$

A plot of $F$ with respect to the 14 zoned dexibifect retronorm is a finite element mesh as shown in Fig 5.62.

While all the examples considered so far in this Section are confined to 2-dimensional meshes, the above idea can easily
be extended to handle three dimensional situations. Indeed a dexiant retronorm can easily be defined for a three dimensional mesh using a dexiarray in which the subarray B contains the x, y and z coordinates of the borders of that mesh. The subarrays M and N are as before.

As a first example consider the family of hyperbolic paraboloid shell as shown in Fig 5.63. Now let a formex F be formulated as follows

\[ F = \text{rinid}(20,20,1,1)[0,0,0; 1,0,0; 1,1,0; 0,1,0]. \]

In the case of hyperbolic paraboloid shell of Fig 5.63(a), one can define a 1 zoned dexitrifact retronorm by interpolating linearly between the corners of the shells. The transpose of the dexiarray in this case is

\[
\begin{bmatrix}
X_c & Y_c & Z_c & 1.0 & 1.0 & 0 & 20 & 0 & 20
\end{bmatrix}
\]

where \(X_c\), \(Y_c\) and \(Z_c\) represent the x, y and z coordinates of the corners of the shell.

A plot of F with respect to this normat is shown in Fig 5.64. In the case of shells of Fig 5.63(b-d) one can define a 4 zoned dexitrifact retronorm for each one of them. Plots of F with respect to these retronorms are shown in Fig 5.65 to Fig 5.67.

As another example consider the folded plate of Fig 5.68. This example demonstrates a problem in which the actual flat finite element representation is physically exact. As the plate is flat, then the representation of it by triangular plate elements does not introduces any geometrical error in the modelling process. The plate is modelled with triangular plate elements and the frame stiffness is included in the analysis by suitable beam elements.
Fig 5.67

Fig 5.68
Now let a formex \( F \) be formulated as follows

\[
F = F_1 \# F_2
\]

where \( F_1 \) represents all the triangular plate elements and \( F_2 \) represents all the beam elements and are as follows

\[
G = [0, 0, 0; 1, 0, 0; 1, 1, 0]
\]

\[
F_1 = \text{rinid}(8, 17, 1, 1) \{ G \# \text{refid}(0.5, 0.5) \} G
\]

and

\[
F_2 = \text{rinid}(2, 17, 8, 1) \{ [0, 0, 0; 0, 1, 0] \}
\]

The information necessary to transform the formex \( F \) into a mesh suitable for analysis is specified through a dexiarray which defines a 3 zoned dexiant retronorm

\[
\begin{bmatrix}
X_{c1} & X_{c2} & X_{c3} \\
Y_{c1} & Y_{c2} & Y_{c3} \\
Z_{c1} & Z_{c2} & Z_{c3}
\end{bmatrix}
\begin{bmatrix}
1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 \\
0 & 0 & 0 \\
8 & 8 & 8 \\
0 & 4 & 13 \\
4 & 13 & 17
\end{bmatrix}
\]

where \( X_c, Y_c \) and \( Z_c \) represent the \( x, y \) and \( z \) coordinates of the corners of each zone. A plot of \( F \) with respect to the above defined retronorm is shown in Fig 5.69. Zienkiewicz [5.12] used a mesh similar to the one shown.

Figure 5.70 Shows the intersection of a cylinder with a plate. Let it be required to define a mesh for this example. In order for the parabolic representation of the circular shape of the cylinder does not introduce excessive geometric
errors, the structure is divided into 8 zones as shown. Hamilton et al [5.13] used a mesh similar to the one shown in Fig 5.71.

Now let a formex F be formulated as follows

\[ F = \text{rinid}(40,20,1,1); [0,0,0; 1,0,0; 1,1,0; 0,1,0]. \]

A plot of F with respect to an 8 zoned dexitrifect retronorm defined through the following dextiarray

\[
\begin{bmatrix}
X_{knpl} & X_{knp2} & X_{knp3} & X_{knp4} & X_{knp5} & X_{knp6} & X_{knp7} & X_{knp8} \\
Y_{knpl} & Y_{knp2} & Y_{knp3} & Y_{knp4} & Y_{knp5} & Y_{knp6} & Y_{knp7} & Y_{knp8} \\
Z_{knpl} & Z_{knp2} & Z_{knp3} & Z_{knp4} & Z_{knp5} & Z_{knp6} & Z_{knp7} & Z_{knp8}
\end{bmatrix}
\]

\[
\begin{bmatrix}
1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
0 & 10 & 20 & 30 & 0 & 10 & 20 & 30 \\
10 & 20 & 30 & 40 & 10 & 20 & 30 & 40 \\
0 & 0 & 0 & 0 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 20 & 20 & 20 & 20
\end{bmatrix}
\]

is shown in Fig 5.71.

As a final example in this Section consider Fig 5.72 which shows a part of a roof of a petrol station. Pissanestzky [5.14] used a mesh similar to the one shown. To formulate a mesh for this finite element model, let a formex be defined as

\[ F = \text{rinid}(10,20,1,1); G \]

and

\[ G = [0,0,0; 1,0,0; 1,1,0; 0,1,0; 0,0,1; 1,0,1; 1,1,1; 0,1,1]. \]

where G represents a hexahedral finite element with corner
nodes.

A plot of $F$ with a 2-zone dexitrifect retronorm defined through the following dexiarray

$$
\begin{bmatrix}
X_{knpl} & X_{knp2} \\
Y_{knpl} & Y_{knp2} \\
Z_{knpl} & Z_{knpl}
\end{bmatrix}
\begin{align*}
1.0 & 1.0 \\
1.0 & 1.0 \\
1.0 & 1.0 \\
0 & 0 \\
10 & 10 \\
0 & 10 \\
10 & 20 \\
0 & 0 \\
1 & 1
\end{align*}
$$

will be as shown in Fig 5.72.

5.4 TRIDIRECTION DEXIANT RETRONORMS

The uni- and bidirectional interpolation described in the previous Section are best suited to create normats within a four-sided region. Although a degenerate form of these mappings may be used to create a normat in three-sided regions, the resulting normat has a set of its normat lines converging at a corner of the region (see Fig 5.49). If the region is then paved with finite elements this will creates a condition where several elements are connected to a single node. In some cases this is not suitable and results in poor elements aspect ratio and an unfavorable nodal bandwidth. These problems are best overcome by using a triangular coordinate system to create a normat within a triangular
Fig 5.73

Fig 5.74
Consider the triangular region shown in Fig 5.73. A tridirectional dexiant retronorm can be defined within the region by interpolating in three directions using a triangular coordinate system. Here, L1, L2 and L3 represent a convenient set of coordinates for the triangle. The linear relationship between these and the Cartesian coordinate system are

\[ x = L1 \cdot x1 + L2 \cdot x2 + L3 \cdot x3 \]
\[ y = L1 \cdot y1 + L2 \cdot y2 + L3 \cdot y3 \]
\[ L1/(IA-IB) + L2/(IC-ID) + L3 = 1. \]

A tridirectional normat can be defined within the triangular region using the above defined coordinate system and is shown by dotted lines.

As an example consider a quarter of a circular plate as shown in Fig 5.74 and let it be required to pave this region with a uniform mesh consisting of triangular elements. A dexiarray which defines a tridirectional dexibifect retronorm for the region can be defined as

\[
\begin{bmatrix}
X_{knp} \\
Y_{knp} \\
1.0 \\
1.0 \\
0 \\
10 \\
0 \\
10
\end{bmatrix}
\]

where X_{knp} and Y_{knp} contain the information necessary to describe the three borders of the region (i.e. the
coordinates of the corners and mid-side point of the curved border).

Now, let a formex $F$ be formulated as follows

$$ F = F_1 \# F_2 $$

where

$$ F_1 = \text{lib}(J=0,9) \mid \text{lib}(I=0,9-J) \mid $$
$$ \text{tranid}(I,J) \mid [0,0; 1,0; 0,1] $$

and

$$ F_2 = \text{lib}(J=0,8) \mid \text{lib}(I=0,8-J) \mid $$
$$ \text{tranid}(I,J) \mid [0,1; 1,0; 1,1]. $$

A paribifect N-plot of $F$ is shown in Fig 5.75. A plot of $F$ with respect to the above defined tridirectional dexibifect retronorm is shown in Fig 5.76. Figure 5.77 shows a circular region paved with triangular elements using a tridirectional dexibifect retronorm.

As a three dimensional example, consider the conical surface shown in Fig 5.78. Let it be required to model this structure with triangular shell elements. Ghassemi [5.15] uses a mesh similar to that shown in Fig 5.79, produced by an automatic mesh generation scheme. In order to pave the conical region with finite elements, the cone is divided into 4 zones as shown in Fig 5.78. A 4 zone dexiarray defining a tridirectional dexitrifect retronorm for the region is
where Xknp, Yknp and Zknp contain the information necessary to describe the borders of each zone. Now, let a formex Fl be formulated as

\[ Fl = F2 \# F3 \]

where

\[ F2 = \text{rin}(1,4,10); \text{lib}(J=0,9); \text{lib}(I=0,9-J); \text{tranid}(I, J); [0,0; 1,0; 0,1] \]

\[ F3 = \text{rin}(1,4,10); \text{lib}(J=0,8); \text{lib}(I=0,8-J); \text{tranid}(I, J); [1,0; 1,1; 0,1]. \]

A plot of Fl with respect to the above defined tridirectional dextrifitect retronorm is shown in Fig 5.79.

As another example consider Fig 5.80 which is a part of a three dimensional shell. Let it be required to pave this region with triangular plate elements. Harber et al [5.16] used a mesh similar to the one shown in Fig 5.81. In order to pave this region by triangular finite elements let a 1-zone tridirectional dextrifitect retronorm be defined as
Now, let a formex $F$ be formulated as

$$ F = F_1 \# F_2 $$

where

$$ F_1 = \text{lib}(j=0,9) \mid \text{lib}(i=0,9-j) \mid \text{tranid}(i,j) \mid [0,0; 1,0; 0,1] $$

and

$$ F_2 = \text{lib}(j=0,8) \mid \text{lib}(i=0,8-j) \mid \text{tranid}(i,j) \mid [0,1; 1,0; 1,1]. $$

A plot of $F$ with respect to the above tridirectional dextrifect retronorm will represent the mesh of Fig 5.81.

As another example consider a square plate with a central circular hole under direct tensile forces. Due to symmetry a quarter of the plate is considered and is shown in Fig 5.82. The plate is divided into 7 zones as shown. Only one of the zones is triangular, the rest are quadrilateral, thus the retronorm defined for this example is composed of both tridirectional and bidirectional dextrifect retronorms.

Let a formex $F$ be formulated as follows

$$ F = F_1 \# F_2 \# F_3 \# F_4 \# F_5 \# F_6 $$

where
Fl = lam(1,6); rin(2,3,2); lam(2,2); [0,1; 1,2; 0,2],
F2 = lamid(6,4); [0,0; 1,0; 0,1],
F3 = rinid(6,4,2,2); lam(2,1); [0,1; 1,0; 2,1],
F4 = rinid(5,4,2,2); lam(2,1); [1,0; 2,1; 3,0],
F5 = rin(1,3,2); [3,8; 5,8; 3,10] #
       rin(1,2,2); [3,10; 5,8; 5,10]
and
F6 = [[3,10; 5,10; 14,3], [5,10; 7,10; 3,14]].

A paribifect N-plot of F is shown in Fig 5.83.

A plot of F with respect to a dexiarray which is defined as

\[
\begin{array}{cccccccc}
Xknpl & Xknp2 & Xknp3 & Xknp4 & Xknp5 & Xknp6 & Xknp7 \\
Yknpl & Yknp2 & Yknp3 & Yknp4 & Yknp5 & Yknp6 & Yknp7 \\
1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
0 & 4 & 8 & 0 & 3 & 9 & 3 \\
4 & 8 & 12 & 3 & 9 & 12 & 9 \\
0 & 0 & 0 & 7 & 7 & 7 & 8 \\
7 & 7 & 7 & 8 & 8 & 8 & 14 \\
\end{array}
\]

represents a finite element mesh for quarter of the plate and is shown in Fig 5.84. Zienkiewicz [5.17] used a mesh similar to the one shown.

5.5 HIGHER ORDER DEXIANT RETRONORMS

In all the examples considered so far, in order to define a normat for a region it was divided into zones for which the borders were defined by low degree polynomials. As explained in Chapter 3, the accepted surface/solid modelling techniques can provide useful tools in defining retronorms for a wide range of configurations. For instance transfinite mappings
which are well documented by Gordon et al [5.18-5.22] can be used to define normats for complex surfaces and volumes.

It should be mentioned that these mappings are only used to define a normat within the region under consideration. The topology and thus the connectivity of the mesh is generated using the concepts of formex algebra. In finite element mesh generation by transfinite mappings the constant curvelinear coordinate systems produced by this technique are used to generated the elements topology. This procedure places a restriction on the element topology. In formex approach the topological characteristics of a configuration are separated from its geometrical shape. This removes the restriction placed by the mapping technique on the element topology.

As an example consider the mesh shown in Fig 5.85. If one tries to pave this region by a dexitrifect retronorm using the approach which was defined in Section 5.3 the region must be divided into 12 zones (for accurate geometric modelling) as shown in Fig 5.86. One can use a different approach and divide the region into 4 zones as shown in Fig 5.87. One may then pave each region by a normat obtained by interpolation between the borders of each zone.

Now let a formex \( F \) be formulated as follows

\[
F = \text{nov}(E) \mid \text{rinid}(30,36,1,1) \mid G
\]

where

\[
G = [0,0; 1,0; 1,1; 0,1]
\]

and

\[
E = \text{rin}(1,30,1) \mid [0,36; 0,0].
\]

A plot of \( F \) with respect to the above defined dexitrifect retronorm would represent the mesh of Fig 5.85. As another example consider the mesh shown in Fig 5.88. In order to pave this region by a dexicylindrical retronorm the structure is divided into 3 zones as shown in Fig 5.89. Then a normat
is defined within each region by interpolating between its borders.

Now let a formex \( F \) be formulated as follows

\[
F = \text{nov}(E) \text{rinid}(20,20,1,1) \text{gin}(0,0,0,0,0,1,1,0,1)
\]

where

\[
G = [0,0; 1,0; 1,1; 0,1]
\]

and

\[
E = \text{rin}(1,20,1) [0,20; 0,0].
\]

A plot of \( F \) with respect to the above defined dexicylindrical retronorm would represent the mesh of Fig 5.89.
CHAPTER 6
The success of the finite element method in solving practical structural mechanics problems has lead to its increase use in other disciplines of engineering. But the users are sometimes dismayed by the enormous amount of input required for a realistic analysis. In many cases, the success of the method relies on a suitable discretization of the given domain. In view of this situation, a number of mesh generation algorithms with varying degrees of automation have been proposed by many researchers. However, these computer-assisted data generation schemes have been restricted to specific types of structures. On the other hand, the use of these techniques within their domain of application require only a small number of parameters to be specified by the user, allowing the bulk of data to be generated automatically.

In this work a new approach to mesh generation has been proposed which has formex algebra as its basis. The formex approach to mesh generation is somewhat different from other generation schemes in that it provides a framework with the basic facilities and the possibility of simple implementation of many others.

Formex approach to mesh generation is performed in two distinct steps, namely a topological step and a geometrical step.

Firstly, a formex which basically is an algebraic representation of the topological characteristics of the configuration is formulated. The process of formulation can be carried out using various formex functions which act as conceptual tools in dealing with complex configurations. As there are no topological restrictions, practically any type of finite element can be defined. During the formulation process one has to take into account the connectivity of the
points/edges/faces of the mathematical model which will be in contact after the geometric transformation.

Structural analysis programs normally require the interconnection data to be in terms of node numbers. The next stage in the topological step is to turn the formex describing the disposition of the elements into a form which is relative to an implied sequence of node numbers. This eliminates the need for explicit node numbering and provides a suitable way of data preparation for use in conjunction with the existing structural analysis programs. This is achieved through a family of formex functions which are defined for node numbering purpose. These formex functions are used in a similar fashion to express other types of structural data such as loads or supports in terms of a node numbering scheme. If the material properties of the elements of the mesh vary then the formulation of the mesh is represented by a set of formices, that is each group of the elements are represented by a separate formex.

Next, the geometric step is carried out. Its purpose is the specification of all the nodal coordinates of the mesh. The procedure ranges from the most obvious and simple one of specifying the coordinates of every node to the more sophisticated ones giving the coordinates of selected nodes and then using some suitable interpolation rule to calculate the remaining nodal coordinates.

Standard retronorms provide a suitable means for defining the geometry of a mesh with considerable ease. For more complex geometries one can employ coordinate equations (defined by the user) in terms of the standard retronorms.

At this point it is useful to put forward some recommendation for future work:

In order to model complex structures higher order dexitant
retronorms must be defined. All the accepted surface/solid modelling techniques which are being used in computer aided design provide useful tools in defining retronorms for a wide range of configurations.

In the present work an old fashioned key board method was used to input the geometrical data of the mesh. This inputing technique must be substituted by screen cursor, joystick, tracker ball, tablet digitizer or light pen for user convenience.

In present work all the meshes were generated by defining the elements first and then creating the mesh through coordinate transformation. Another method which is worth investigating is generating nodal points first (which must be inserted within and on the boundary of the structures to be meshed). Then these nodal points are automatically connected together to form a network of well-proportioned elements.
BLANK IN ORIGINAL
CHAPTER 1


CHAPTER 3


3.16- Sadek, E.A., 'A Scheme for the automatic generation of triangular finite elements', International Journal for


CHAPTER 4


4.4- Tonishoff, H.K., Jendryschik, J.,'Behaviour of disk like rotating tools', ENGINEERING FRACTURE MECHANICS, 1985,pp


4.9- Lee, H.P. and Harris, P.J., 'Post-buckling strength of thin-walled members', COMPUTERS and STRUCTURES, vol 10, 1979, pp 689-702.


4.13- Baroonian, A., 'Experimental and theoretical

CHAPTER 5


5.3- Francisco Medina et al, INTERNATIONAL JOURNAL OF NUMERICAL METHODS IN ENGINEERING, vol. 17-(1981) PP 1177-1185


5.6- Fujii, T. and Yuki, T., 'Automated data generation and matrix operation', THEORY AND PRACTICE IN FINITE ELEMENT STRUCTURAL ANALYSIS-(1973) PP 671-686


5.8- Noor,A.K., Balch, C.D. and Shibut, M.A., 'Reduction

5.9- Goffe, I. and Yassaee, M.H., 'Solutions of Elasticity Problems Using the Finite Element Method', Final year BSc project, University of Birmingham, Civil Engineering Department, 1980.


5.16- Harber, R. and Abel, J.F., 'Discrete transfinite mappings for the description and meshing of three-dimensional