5-1-1982

Computer programs for transformations of coordinates in two (2) and three (3) dimensional Euclidean geometry

Melony Vanessa Sanks
Atlanta University

Follow this and additional works at: http://digitalcommons.auctr.edu/dissertations

Part of theApplied Mathematics Commons, Computer Sciences Commons, and theMathematics Commons

Recommended Citation
Sanks, Melony Vanessa, "Computer programs for transformations of coordinates in two (2) and three (3) dimensional Euclidean geometry" (1982). ETD Collection for AUC Robert W. Woodruff Library. Paper 2238.

This Thesis is brought to you for free and open access by DigitalCommons@Robert W. Woodruff Library, Atlanta University Center. It has been accepted for inclusion in ETD Collection for AUC Robert W. Woodruff Library by an authorized administrator of DigitalCommons@Robert W. Woodruff Library, Atlanta University Center. For more information, please contact cwiseman@aucr.edu.
COMPUTER PROGRAMS FOR
TRANSFORMATIONS OF COORDINATES
IN TWO (2) AND THREE (3) DIMENSIONAL
EUCLIDIAN GEOMETRY

A THESIS
SUBMITTED TO THE FACULTY OF ATLANTA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE

BY
MELONY VANESSIA SANKS

DEPARTMENT OF MATHEMATICAL SCIENCES

ATLANTA, GEORGIA
MAY, 1982

\[
R = \sin T = \frac{89}{99}
\]
ABSTRACT

MATHEMATICAL SCIENCES

SANKS, MELONY V. B.S. FORT VALLEY ST. COLLEGE, 1979

COMPUTER PROGRAMS FOR TRANSFORMATION OF COORDINATES IN TWO (2) AND THREE (3) DIMENSIONAL EUCLIDEAN GEOMETRY.

Advisor: Dr. Johnny L. Houston

Thesis dated December, 1981

The main objective in preparing this thesis is to develop computer routines to do transformations of coordinates in two and three dimensional Euclidean Geometry. The author superficially explores transformations of coordinates geometrically and algebraically in order to discover the transformation equations for use in developing the computer routines. The routines are then developed and tested.
ACKNOWLEDGEMENT

In appreciation for their understanding and long hours of work, I would like to thank my advisor, Dr. Johnny L. Houston and my typist, Mrs. Lorene McRae. I would also like to thank my family for their patience and Mr. Lewis Wooten for his suggestions.
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1</td>
<td>The relationship between various types of geometries and Incidence geometry</td>
<td>6</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Translation of axes</td>
<td>22</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Graph of $y = x^2 - 6x + 5$ and $y' = x'^2$</td>
<td>22</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Rotation of Axes</td>
<td>25</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Rotation of Axes</td>
<td>25</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Rotation through an Angle Theta</td>
<td>27</td>
</tr>
<tr>
<td>2.3.1</td>
<td>3-Dimensional Coordinate System</td>
<td>30</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Right-handed and Left-handed System</td>
<td>32</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Change of Origin</td>
<td>35</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Rotation of Axes</td>
<td>37</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Flowchart of PROGRAM TRACOR2</td>
<td>44</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Flowchart of PROGRAM TRACOR3</td>
<td>52</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Two Types of Matrices Used in Graphic Applications</td>
<td>74</td>
</tr>
<tr>
<td>5.2.2</td>
<td>The Untransformed View of a Coffee Cup</td>
<td>78</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Coffee Cup Expanded Along the $y$-axis</td>
<td>80</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Coffee Cup Expanded Along the $x$-axis</td>
<td>81</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Coffee Cup Rotated $90^\circ$ and Expanded Along the $z$-axis</td>
<td>82</td>
</tr>
</tbody>
</table>
5.2.6 Coffee Cup Magnified ..................... 83
5.2.7 Coffee Cup Magnified, Expanded and Rotated ............................................. 87
5.2.8 Coffee Cup Rotated but not Sheared ..... 87
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1</td>
<td>The Unit Matrix</td>
<td>78</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Expansion along the y-axis by a factor $K$</td>
<td>79</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Expansion along the x-axis by a factor of $K$</td>
<td>81</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Expansion along the z-axis by a factor of $K$</td>
<td>82</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Magnification by a factor $K$</td>
<td>83</td>
</tr>
<tr>
<td>5.2.6</td>
<td>Rotation in the x,y and y,z planes sequentially</td>
<td>84</td>
</tr>
<tr>
<td>5.2.7</td>
<td>Translation matrix</td>
<td>86</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................ iii
LIST OF FIGURES ................................................ iv
LIST OF TABLES .................................................... vi

Chapter

I. Introduction ................................................. 7
   1.1 Geometries ................................................. 7
   1.2 Incidence Geometry ................................. 11
   1.3 Euclidean Geometry ............................ 16
   1.4 Coordinate Euclidean Geometry for
        n-Dimensions .................................. 18

II. Transformation of Coordinates  
     (Geometrically and Algebraically) .......... 20
   2.1 Transformations ....................................... 20
   2.2 Transformation of Coordinates in
        Two-Dimensions ................................ 21
   2.3 Cartesian Coordinates in
        Three-Dimensions ................................ 29
   2.4 Transformation of Coordinates  
        in Three-Dimensions .......................... 34

III. PROGRAM TRACOR2 ...................................... 41
   3.1 General Organization .............................. 41
   3.2 Flowchart of PROGRAM TRACOR2 ............ 44
   3.3 Code for PROGRAM TRACOR2 .................. 46
IV. PROGRAM TRACOR3 .......................... 49
  4.1 General Organization ...................... 49
  4.2 Flowchart of PROGRAM TRACOR3 ........... 52
  4.3 Code for PROGRAM TRACOR3 ............... 54

V. Applications .................................... 57
  5.1 Sample Runs of TRACOR2 and TRACOR3 . 58
  5.2 Graphic Manipulations Using Matrices ................. 72
CHAPTER I

INTRODUCTION

1.1 Geometries

The full extent to which geometrical and numerical relationships were understood and used in ancient times is not known. The beginnings of geometry dates back to the time of the Babylonians and Egyptians. The word "geometry" is derived from the Greek root "geo" meaning earth and from "metrein" meaning measure. Geometry was important to early peoples in handling problems involving land areas, building volumes, shapes for utensils, weapons, etc. Until about 100 BC all mathematical knowledge consisted of numerous formulas or procedures that were obtained by experimenting rather than from any formal logical system.

It is not certain who first had the idea of trying to prove a mathematical rule by reasoning rather than by testing it. Although both Thales (640-546 BC) and Pythagoras (born 582 BC) have been given credit for the idea, some historians believe that geometry was introduced
to Greece by Thales and that he produced the first proof of a geometric generality or theorem.¹

Over the next three hundred years, the mathematics of geometry grew rapidly. Greek geometers continued to prove theorems and gradually built a body of geometric knowledge. The mathematician Euclid (365-275 BC) unified the mass of geometric findings by constructing the first definite formal system of mathematics. His work was called "The Elements".²

Following Euclid, Greek geometry continued to flourish, with such great mathematicians as Archimedes and Apollonius, until around 100 BC. Western mathematics went into a decline with the rise of the Roman Empire and had reached a standstill by the time of the fall of Rome early in the fifth century.

In the period up to the twelfth century, the Hindu-Arabic numbering system was devised and the development of Algebra had begun. During the 13th and 14th centuries, Moslem scholars introduced Hindu-Arabic mathematics to Western Europe. Euclid's geometry did not become a part


²Ibid., p. 16.
of Western Europe's knowledge until the latter part of the 15th century.

By the end of the 16th century, Euclidean geometry was widely understood and Italian and French mathematicians had made accomplishments in Algebra. By the early 17th century Algebra was combined with geometry in the making of analytic geometry by René Descartes.

The Greek method of doing mathematics within a clearly stated system, with all reasoning referred to definite axioms was restricted to elementary geometry. The idea that all mathematics had the same systematic make-up and could and should be developed in the same manner did not materialize until the end of the 19th century.

The Greek and the succeeding mathematicians felt that the initial rules of abstract geometry were dictated by the simple facts of physical experience. They thought of these rules or axioms as self-evident truths.³

One fact of experience that Euclid formulated as an axiom was the parallel postulate: If a point p is not in a line t, then, in the plane of p and t, there is

³Ibid., p. 17.
exactly one line which passes through p and does not intersect t. Because Euclid developed many theorems before he made use of this postulate, many mathematicians thought the theorems could be proved from other axioms.

In 1829 Nikolai Lobachevsky developed a system of geometry by starting in the same manner as Euclid and by developing the same theorems until the parallel postulate was needed. Then, instead of making use of Euclid's parallel postulate, Lobachevsky made the following assumption: If a point p is not in a line t, then in the plane of p and t, there are always at least two different lines that pass through p and do not intersect t. Using this new postulate, Lobachevsky constructed and developed a geometry different from Euclid's but one that was built based on logical reasoning.

This non-Euclidean geometry was found to be just as logically consistent as that of Euclid.\(^4\) Other non-Euclidean geometries were then constructed.

By 1930, J. Bolyai (1802-1860), a Hungarian Army officer, N. I. Lobachevsky (1793-1856), a Russian professor of Mathematics at the University of Kazan, and Gauss

\(^4\)Ibid., p. 18.
had developed theories of geometry based on a contradiction of Euclid's parallel postulate. They assumed that there is more than one line parallel to a given line through an external point. Bolyai and Lobachevsky are usually given credit as the discoverers of the new theory. Later, in 1857, the German mathematician B. Riemann (1826-1866) introduced a different non-Euclidean theory of geometry based on the assumption that there are no parallel lines.

1.2 Incidence Geometry

An incidence geometry is a system involving three primitive ideas; point, line and plane, which essentially satisfy Hilbert's postulates of incidence.

All the familiar kinds of geometry with the exception of spherical geometry can be characterized as incidence geometries which satisfy one or more additional postulates concerning parallelism, order, congruence, and continuity. The following chart indicates the relation of various types of geometry to incidence geometry.

The Hilbert Postulates are listed below.

HILBERT'S AXIOMS

AXIOM 1. There is one and only one line passing through any two given (distinct) points.

AXIOM 2. Every line contains at least two points and given any line there is at least one point not
Figure 1.1.1 The relationship between various types of geometry and incidence geometry
on it.

AXIOM 3. If a point B lies between the pts. A and C, then A, B, and C all lie on the same line, and B lies between C and A, and C does not lie between B and A, and A does not lie between B and C.

AXIOM 4. Given any two (distinct) points A and C, there can always be found a pt. B which lies between A and C, and a pt. D such that C lies between A and D.

AXIOM 5. If A, B, C are (distinct) points on the same line, one of the three pts. lies between the other two.

Definition. The segment (or closed interval) AC consists of the pts. A and C and of all pts. which lie between A and C. A pt. B is said to be on the segment AC if it lies between A and C, or is A or C.

Definition. Two lines, a line and a segment, or two segments are said to intersect each other if there is a pt. which is on both of them.

Definition. The triangle A B C consists of three segments AB, BC and CA (called the sides of the triangle), provided A, B and C (called vertices) are not on the same line.
AXIOM 6. A line which intersects one side of a triangle and does not pass through any of the vertices must also intersect one other side of the triangle.

AXIOM 7. If A and B are (distinct) pts. and A' is a pt. on a line l, there exist two and only two (distinct) pts. B' and B'' on l such that the pair of pts. A', B' is congruent to the pair A, B and the pair of pts. A', B'' is congruent to the pair of pts. A', B', moreover A' lies between B' and B''.

AXIOM 8. Two pairs of pts. congruent to the same pair of pts. are congruent to each other.

AXIOM 9. If B lies between A and C and B' lies between A' and C', and A, B is congruent to A', B', and B, C is congruent to B', C', then A, C in congruent to A', C'.

Definition. Two segments are congruent if their endpoints are congruent pairs of points.

Definition. The ray A C consist of all points B which lies between A, C, the point C itself, and all points D such that C lies between A and D. The ray A C is said to be from the point A.

Definition. The angle BAC consists of the pt. A and the two rays A B and A C.

Definition. If ABC is a triangle, the three angles
B A C, A C B, C B A are called the angles of the triangle. Moreover the angle BAC is said to be included between the sides A B and A C of the triangle.

AXIOM 10. If B A C is an angle whose sides do not lie in the same line, and B' and A' are (distinct) pts., there exists two and only two (distinct) rays, A'C' and A'C'', from A' such that the angle B'A'C' is congruent to the angle B A C, and the angle B'A'C'' is congruent to the angle B A C. Moreover E' is any point on the ray A'C', and E'' is any pt. on the ray A'C'', the segment E'' intersects the line A' B'.

AXIOM 11. Every angle is congruent to itself.

AXIOM 12. If two sides and the included angle of one triangle are congruent respectively to two sides and the included angle of another triangle, then the remaining angles of the first triangle are congruent each to the corresponding angle of the second triangle.

AXIOM 13. Through a given pt. A not on a given line 1, there passes at most one line which does not intersect 1.

AXIOM 14. If A, B, C, D are (distinct) pts., there exists on the ray A B a finite set of (distinct) pts.
A₁, A₂,..., Aₙ such that (1) each of the pairs A₁, Aₙ₋₁ is congruent to the pairs C, D, and (2) B lies between A and Aₙ.

AXIOM 15. The pts. of a line form a system of pts. such that no new pts. can be added to the space and assigned to the line without causing the line to violate one of the first eight axioms of Axiom 14.

Given the fifteen axioms, all further propositions of Euclidean plane geometry can be derived from them by a process of inference without further appeal to intuition.

1.3 Euclidean Geometry

By Euclidean geometry we mean a system of mathematical geometry that serves as the same model of reality as Euclid's did; that is a geometry that corresponds to the way we are most accustomed to thinking about physical space.⁵

The topic of Greek geometry was the description and investigation of different figures and their properties in the plane and in space. Some of the figures were en-

⁵Ibid., p. 19.
countered as shapes of material objects; others were produced by mechanical devices, and their properties were suggested by and derived from experience with physical phenomena.

It was the aim of Greek geometers to proceed logically. This made it necessary for them to deduce geometric propositions from those previously established without reference to physical phenomena. Out of the gathered knowledge, Euclid drew up a set of definitions and first propositions called axioms or postulates, and from these all other propositions by logical processes without reference to physical intuition can be derived.

It was found that Euclid's first set of principles were in some ways incomplete and redundant in others. It was incomplete in the sense that the results obtained involved unannounced axioms. On the other hand, some of Euclid's definitions do not serve any mathematical purpose. He never refers to them in subsequent definitions or theorems. It seems preferrable to begin with enumeration of certain mathematical objects by name and then state axioms describing properties of these objects. A modern set of axioms enumerates undefined terms and axioms containing these terms. One such example was the axioms proposed by Hilbert at the end of the 19th century.
1.4 Coordinate Euclidean Geometry for n-Dimensions

Descartes was the first mathematician to make extensive use of coordinates in geometry. The Ruler Postulate and the Ruler-placement Postulate could be called the coordinate postulates. They enable us to establish a rectangular coordinate system for the plane or for the 3-dimensional Euclidean space. In the 3-dimensional Euclidean space (or 3-space) this gives us a one-to-one correspondence between the pts. of 3-space and the set of all ordered triples of real numbers.

It was not until the mid 19th century that mathematicians actually identified pts. in the plane with ordered pairs of real numbers and pts. of 3-space with ordered triples of real numbers. We can construct geometry from the real-number system, rather than from the undefined concepts of point, line, plane, etc., by defining a pt. to be an ordered triple of real numbers, a plane to be a solution set \((x y z)\) of the linear equation \(ax + by + cz + d = 0\), and a line to be the solution set of a pair of linear equations.

Some advantages of the geometry of ordered triples is that it makes available a number of algebraic techniques. It reduces the problem of showing the consistency of Euclidean geometry to the problem of showing the consistency of the real-number system, and it can easily be
generalized to higher dimensions.

**Definition of** $E^n$

N-dimensional Euclidean space, $E^n$ is the set of all ordered n-tuples of real numbers $(x_1,\ldots,x_n)$ together with a distance defined on $R^n$ as follows: If $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$, then the distance between $x$ and $y$ is $xy = \sqrt{(x - y)^2 + \ldots + (x - y)^2}$. The Elements of $R^n$ are called points of $E^n$.

**Ruler Postulate** - The pts. of a line can be placed in correspondence with the real numbers in such a way that

1. To every pt. of the line there corresponds exactly one real number.
2. To every real number there corresponds exactly one pt. of the line.
3. The distance between two points is the absolute value of the difference of the corresponding numbers.

The Ruler-placement postulate - Given two pts. $p$ and $q$ of a line, the coordinate system can be chosen in such a way that the coordinates of $p$ is zero and the coordinates of $q$ is positive.
2.1 Transformations

Let's concern ourselves with relations between two sets of variables. Suppose \( u \) and \( v \) are the variables of one set and

\[
\text{Eq. 2.1.1 } x = f(u,v), \quad y = g(u,v)
\]

where \( f \) and \( g \) are certain functions. Then \( x, y \) are the variables of a second set. Equation 2.1.1 defines a transformation. A transformation is a certain kind of function whereby to certain pairs of numbers \( (u,v) \) there corresponds certain pairs of numbers \( (x,y) \). The transformation is a function whose values are pairs of real numbers. The number of variables in each set may be three instead of two. e.g.

\[
\text{Eq. 2.1.2 } x = f(u,v,w), \quad y = g(u,v,w), \quad z = h(u,v,w)
\]

The number of variables in either set might be any positive integer, and the number in the two sets may differ.

The most important cases in actual common application of transformations in which the number of variables in
each set is the same are those in which the number of variables is either two or three. These transformations are of type 2.1.1 and 2.1.2. Also the equations relating polar and rectangular coordinates in the plane, or spherical and rectangular coordinates in 3-dimensional space are of these types.

2.2 Transformation of Coordinates in TWO DIMENSIONS

In defining rectangular coordinates in a plane, we choose a pt. of the plane for the origin, one line through it for the x-axis and the line perpendicular to it for the y-axis. We define the x- and y- coordinates of any point of the plane to be the directed distances of the point from the y-axis and x-axis respectively. A plane does not have a set of pre-determined axes. If two different sets of axes are set up, then it is evident that a given point of the plane will have different coordinates with respect to the two sets of axes.

By the transformation of coordinates in two dimensions, we mean a shift from a rectangular coordinate system xy with origin 0 to another rectangular coordinate system x'y' with origin 0', both systems having the same unit of distance and both systems having the same orientation (i.e. both being right-handed or left-handed).

Given the equation of a curve with respect to an initial set of x and y axes, the equation of this same
Figure 2.2.1 Translation of Axes

Figure 2.2.2 Graph of $y = x^2 - 6x + 5$ and $y' = x'^2$
curve with respect to another specified set of axes may be found in two ways. If the two coordinate systems have different origins with the x and y axes of the two systems being parallel and having the same positive directions, then the transformation from which the new equation is found is called a translation of axes. If the new origin is identical to the original one and the new axes are obtained by revolving the original axes about this origin through some specified angle, the transformation is called a rotation of axes.

O is the origin of the system of coordinates x and y and O' of the system of coordinates x' and y'. A translation of axes is made such that the coordinates of O' in the xy-system are \( x_0 \) and \( y_0 \).

Let \( p \) be any point on a curve. Let its coordinates be \( (x, y) \) relative to the original axes and \( (x', y') \) relative to the new ones. For the representative point \( p \) of coordinates \( x, y \) and \( x', y' \) in the respective systems we have
\[
  x = TS + SP = x_0 + x', \quad y = RQ + QP = y_0 + y'.
\]
See figure 2.2.1

If in the original equation of the curve \( x \) is replaced by \( x_0 + x' \) and \( y \) is replaced by \( y_0 + y' \), the resulting relation between \( x' \) and \( y' \) is the equation of the curve relative to the new set of axes.

Given an equation of a curve \( y = x^2 - 6x + 5 \), to find the equation of the curve relative to the x' axis and y' axis
that passes through the point (3, -4), we would make the following substitution:

\[ x_0 = 3 \text{ and } y_0 = -4. \]

In \( y = x^2 - 6x + 5 \) replace \( x \) by \( x' + 3 \) and \( y \) by \( y' - 4 \).

\[
\begin{align*}
y' - 4 &= (x' + 3)^2 - 6(x' + 3) + 5 \\
y' - 4 &= x'^2 + 6x' + 9 - 6x' - 18 + 5 \\
y' - 4 &= x'^2 - 4 \\
y' - 4 &= x'^2
\end{align*}
\]

(See figure 2.2.2).

When the equation \( x = x' + x_0 \) and \( y = y' + y_0 \) are solved for \( x' \) and \( y' \), we obtain \( x' = x - x_0, y' = y - y_0 \). This is called the inverse transformation. Given the equation \( y' = x'^2 \), to find the equation of the curve relative to the \( x \) and \( y \) axes, the following substitution would be made:

\( x' \) would be replaced by \( x - 3 \) and \( y' \) by \( y + 4 \).

\[
\begin{align*}
y' &= x'^2 \\
y' &= (x - 3)^2 \\
y + 4 &= x^2 - 6x + 9 \\
y &= x^2 - 6x + 5
\end{align*}
\]

We next consider the general situation when the two sets of axes are not parallel; that is when the new set of axes are obtained by rotating the original axes through an angle \( \theta \). (Figures 2.2.3 and 2.2.4)

In each case of figures 2.2.3 and 2.2.4 the direction cosines of the \( x' \)-axis relative to the \( xy \)-system are
Figure 2.2.3 Rotation of Axes

Figure 2.2.4 Rotation of Axes
cosθ and sinθ, and an equation of the x'-axis is

\[ x - x_0 = \frac{y - y_0}{\cos \theta} \text{ or } (y - y_0) \cos \theta - (x - x_0) \sin \theta = 0 \]

and the equation of the y'-axis is

\[ (y - y_0) \sin \theta + (x - x_0) \cos \theta = 0. \]

Solving these equations for \( x - x_0 \) and \( y - y_0 \), the result is

\[ \text{Eq. 2.2.1} \quad x = x' \cos \theta - y' \sin \theta + x_0 \]
\[ y = x' \sin \theta + y' \cos \theta + y_0 \]

These are the set of rotation equations.

Hence any equation of any line or curve with respect to the xy system is transformed into an equation in the x'y'-system when the expressions in 2.2.1 are substituted for \( x \) and \( y \) in the given equation.

When the origin \( 0' \) coincides with \( 0 \) (Figure 2.2.5) then

\[ \text{Eq. 2.2.2} \quad x = x' \cos \theta + y' \sin \theta \]
\[ y' = x' \sin \theta + y' \cos \theta \]

The equations in 2.2.2 can be solved inversely for \( x' \) and \( y' \). If the x'y' system is obtained from the xy system by rotating through an angle \( \theta \), then the xy system is obtained from the x'y' system by a rotation through an angle \(-\theta\). The inverse equations can be obtained by replacing \( \theta \) with \(-\theta\) and shifting the primes to unprimed letters and vice versa. Since \( \cos (-\theta) = \cos \theta \) and \( \sin (-\theta) = -\sin \theta \), this gives
Figure 2.2.5 x and y axes rotated through an angle \( \theta \) to obtain new x' and y' axes.
2.2.3 \( x' = x \cos \theta + y \sin \theta \)
\( y' = -x \sin \theta + y \cos \theta \)

An alternate method for solving for the inverse may be used. The equations in 2.2.3 may be written as

2.2.4 \( \begin{bmatrix} x \\ y \end{bmatrix} = A \begin{bmatrix} x' \\ y' \end{bmatrix} \)

where \( \begin{bmatrix} x \\ y \end{bmatrix} \) and \( \begin{bmatrix} x' \\ y' \end{bmatrix} \)

and the operator \( A \) is a matrix operator defined by:

\[
A = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix}
\]

Equation (2.2.4) may be written as

\[
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}.
\]

Also

2.2.5 \( \begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} \).

The matrix \( A \) is an orthogonal matrix, that is \( A^{-1} = A^t \) and \( \det A = 1 \). We can write eq. (2.2.5) as

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},
\]

and \( x' = x \cos \theta + y \sin \theta \)
\( y' = -x \sin \theta + y \cos \theta \).

The equation of a curve depends on the location of the \( x \) and \( y \) axes relative to the curve. Given an equation with respect to a set of axes, the equation of the same curve with respect to a different set of axes may be found. This second equation may be obtained from the first one by transformations of translation and rotation. For each
original transformation there is an inverse transformation.

A transformation may be indicated by a set of equations. The equations and their inverses for translation and rotation are summarized below:

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Equations for Transformation</th>
<th>Equation for inverse Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation of axes to origin (h,k)</td>
<td>(x = h + x') (x' = x - h) (y = k + y') (y' = y - k)</td>
<td></td>
</tr>
<tr>
<td>Rotation of axes through angle (\theta)</td>
<td>(x = x'\cos\theta - y\sin\theta) (x' = x\cos\theta + y\sin\theta) (y = x'\sin\theta + y\cos\theta) (y' = -x\sin\theta + y\cos\theta)</td>
<td></td>
</tr>
</tbody>
</table>

2.3 Cartesian Coordinates in Three Dimensions

Three dimensional coordinate systems enable us to study graphs of equations in three variables. Three mutually perpendicular coordinate lines in space enable the introduction of coordinates in space.

In a plane the position of a point \(p\) is determined by two coordinates \(x\) and \(y\) referred to two straight lines \(Ox\), \(Oy\), the coordinate axes.

To fix the position of a point in space, three planes are needed. The planes have a point \(O\), the origin, in common and intersect in pairs in three lines \(x' O x\), \(y' O y\) and \(z' O z\). The three lines are called coordinate axes and the planes are called coordinate planes. The axes are assumed to have the same scale and to meet at their origins.
Figure 2.3.1 3-D Coordinate System
We define the coordinates of the point \( p \) as the three lengths

\[ O L' = x, \quad O M' = y, \quad O N' = z. \]

To every point \( p \) there corresponds uniquely a set of three numbers \((x, y, z)\), and conversely to every set of three numbers, positive or negative, there corresponds a unique point. The three numbers \((x, y, z)\) uniquely identifying a point \( p \) are called cartesian coordinates of the point \( p \).

A. Convection of Signs and Rectangular Systems

Let the positive direction along the axes of \( x \) and \( y \) be defined by \( Ox \) and \( Oy \); then in the plane \( xOy \) we may pass from \( Ox \) to \( Oy \) by a rotation through an angle \( xOy \) less than two right angles. Viewed from one side of the plane, the rotation is clockwise, and counter-clockwise from the other. That side of the plane from which the rotation appears counter-clockwise is defined as the positive side of the plane. The positive direction of the \( z \) axis is defined to be that which lies on the positive side of the \( xOy \) plane. This relation holds for each of the axes. This is called a right-handed system of cartesian coordinates. Figure 2.3.2 shows a right-handed and left-handed system. This paper deals exclusively with right-handed systems.

When the planes are mutually perpendicular, we call it a rectangular system, otherwise it is oblique. We shall
Figure 2.3.2

Left-Handed System

Right-Handed System
concern ourselves with a rectangular system.

B. Direction Angles and Direction Cosines

OP is called the radius vector of P, denoted by \( r \). Since \( PN \) is perpendicular to the plane \( xOy \). ON is the orthogonal projection of OP on the plane \( xOy \).

Let the angles which OP makes with the positive direction of the axes be \( \alpha, \beta, \gamma \), then

\[
\begin{align*}
x &= r \cos \alpha, \\
y &= r \cos \beta, \\
z &= r \cos \gamma.
\end{align*}
\]

The position of a point \( P \) is determined by the angles \( \alpha, \beta, \gamma \) and the radius vector \( r \), for these then determine \( x, y, z \). The angles alone determine only the direction of the line OP. We call them the direction-angles of the line OP. The cosines of these angles occur frequently which makes it convenient to call them the direction cosines and are denoted by \( l, m, n \).

C. Properties of the Direction Cosines

The direction angles are not unique since each is indeterminate to an added multiple of \( \pi \). However the direction-cosines are unique. With \( \theta \) always being positive, the direction-cosines \( l, m, n \) are uniquely defined as

\[
\begin{align*}
l &= x/r, \\
m &= y/r, \\
n &= z/r.
\end{align*}
\]

The following example shows how to find the direction-cosines of the line joining the origin to the point \( (1, 2, 2) \). Here \( r^2 = 1^2 + 2^2 + 2^2 = 1 + 4 + 4 = 9 \), hence \( r = 3 \).

Then \( l = 1/3 \), \( m = 2/3 \), \( n = 2/3 \).
To determine the radius vector in terms of the rectangular coordinates we make use of the Pythagorean Theorem.

\[ OP^2 = ON^2 + NP^2 = OL'^2 + LN^2 + NP^2 \]

Hence

\[ r^2 = x^2 + y^2 + z^2 \]

Putting \( x = r \cos \theta, \quad y = r \cos \theta, \quad z = r \cos \theta \), dividing by \( r^2 \) gives

\[ \cos^2 \theta + \cos^2 \theta + \cos^2 \theta = 1 \]

2.4 Transformation of Coordinates in Three Dimensions

A. Translation of axes

Let a new coordinate system be constructed with origin \( O' = [x', y', z'] \) and coordinate planes parallel to the old ones. Let the coordinates of a point \( P \) referred to the two systems be \( [x, y, z] \) and \( [x', y', z'] \). Draw through \( P \) a line parallel to the \( x \)-axis cutting the planes \( yOz \) and \( y'Oz' \) in \( L \) and \( L' \) and let the plane \( y'O'z' \) cut \( Ox \) in \( k \). Then since the parallel planes \( yOz \), \( y'O'z' \) intercept equal segments on parallel lines, \( LL' = Ok \). But \( LP = x, \quad L'P = x', \quad OK = x \), hence

\[ x = x' + X \]
\[ y = y' + Y \quad \text{and} \quad y' = y - Y \]
\[ z = z' + Z \quad \text{and} \quad z' = z - Z \]

Eq. 2.4.2

Eq. 2.4.1

\( x', y', z' \) are the coordinates of \( P \) relative to \( O = [X, Y, Z] \).

The transformation expressed in 2.4.1 (and its inverse expressed in 2.4.2) is called a translation of the original.
Figure 2.4.1 Change of Origin
If two lines $l_1, m_1, n_1$ and $l_2, m_2, n_2$ are perpendicular or orthogonal, their angle $\Theta = \frac{\pi}{2}$, and $\cos \Theta = 0$, then

$$l_1 l_1 + m_1 m_2 + n_1 n_2 = 0$$

or

$$\Sigma \cos \alpha_1 + \cos \alpha_2 = 0.$$

Two lines are parallel when they have the same direction-angles. It follows then that $\sin \Theta = 0$ and $\Theta = 0$.

Conversely, if $\Theta = 0$, $\sin \Theta = 0$ and $\Sigma (m_1 n_2 - m_2 n_1)^2 = 0$. If $l_1, m_1, n_1$ and $l_2, m_2, n_2$ are the actual direction-cosines we have

$$l_1^2 + m_1^2 + n_1^2 = I = l_2^2 + m_2^2 + n_2^2.$$

Putting each of the equal ratios equal to $t$ and substituting $l_1 = t l_2$, $m_1 = t m_2$, $n_1 = t n_2$, we get $t^2 (l_2^2 + m_2^2 + n_2^2) = l_2^2 + m_2^2 + n_2^2$.

Hence $t = \pm I$.

If $t = + I$, the direction cosines are identical; if $t = -I$, they are equal but have opposite signs.

B. Rotation of the Axes

Let's consider the transformation to new axes with the same origin. Let the direction-cosines of the new axes of $x', y', z'$ with respect to the old be

$$l_1, m_1, n_1,$$

$$l_2, m_2, n_2,$$

$$l_3, m_3, n_3$$

respectively and assume both sets of axes are orthogonal. Let $p$ be a point with $[x, y, z]$ as old coordinates and $[x', y', z']$ as the new ones. Projecting
Figure 2.4.2 Transformation to new axes with same origin.
Op on each of the original axes in succession, we have, since the projection of OP is equal to the sum of projections of $x'$, $y'$, and $z'$.

Eq. 2.4.3

\[
\begin{align*}
x &= l_1 x' + l_2 y' + l_3 z' \\
y &= m_1 x' + m_2 y' + m_3 z' \\
z &= n_1 x' + n_2 y' + n_3 z'.
\end{align*}
\]

Also since $[l_1, l_2, l_3]$ are the direction-cosines of Ox with respect to the new axes

Eq. 2.4.4

\[
\begin{align*}
x' &= l_1 x + m_1 y + n_1 z \\
y' &= l_2 x + m_2 y + n_2 z \\
z' &= l_3 x + m_3 y + n_3 z
\end{align*}
\]

Equations (2.4.3) and (2.4.2) represent inverse transformations and can be represented by

\[
\begin{bmatrix}
x' \\ y' \\ z'
\end{bmatrix} =
\begin{bmatrix}
x \\ y \\ z
\end{bmatrix}
\begin{bmatrix}
l_1 & l_2 & l_3 \\
m_1 & m_2 & m_3 \\
n_1 & n_2 & n_3
\end{bmatrix}
\]

The coefficients are connected by a number of equations. Any element in the square is the cosine of the angle between the axes in whose row and column it lies. Since $[l_1, m_1, n_1]$ are a set of direction cosines, we have

\[
\begin{align*}
l_1^2 + m_1^2 + n_1^2 &= 1 \\
l_2^2 + m_2^2 + n_2^2 &= 1 \\
l_3^2 + m_3^2 + n_3^2 &= 1
\end{align*}
\]

38
Since the new axes are mutually at right angles,
\[ l_2l_3 + m_2m_3 + n_2n_3 = 0 \]
\[ l_3l_1 + m_3m_1 + n_3n_1 = 0 \]
\[ l_1l_2 + m_1m_2 + n_1n_2 = 0. \]
Considering the old axes in terms of the new, we have
\[ l_1^2 + l_2^2 + l_3^2 = I \quad m_1n_1 + m_2n_2 + m_3n_3 = 0 \]
\[ m_1^2 + m_2^2 + m_3^2 = I \quad n_1l_1 + n_2l_2 + n_3l_3 = 0 \]
\[ n_1^2 + n_2^2 + n_3^2 = I \quad l_1m_1 + l_2m_2 + l_3m_3 = 0. \]
This transformation is equivalent to a rotation through a definite angle about a definite line. The angle \( \phi \) and the ratios of direction-cosines \( \lambda, \mu, \nu \) of the axis of rotation form the three independent constants, and the equations of transformation could be expressed in terms of these. When the center \( O' \) of the \( x'y'z' \)-system coincides with the origin \( O \) of the \( xyz \)-system, the equations of transformations are
\[ x = \lambda_1 x' + \lambda_2 y' + \lambda_3 z' \]
\[ y = \mu_1 x' + \mu_2 y' + \mu_3 z' \]
\[ z = \nu_1 x' + \nu_2 y' + \nu_3 z'. \]
The transformation is referred to as a rotation of the original axes. The inverse of a transformation is
\[ x' = \lambda_1 x + \mu_1 y + \nu_1 z \]
\[ y' = \lambda_1 x + \mu_1 y + \nu_1 z \]
\[ z' = \lambda_1 x + \mu_1 y + \nu_1 z. \]
Transformation of coordinates in three-dimensions may
be summarized as follows: To transform to parallel axes through a new origin, whose coordinates referred to the old axes, are \( x', y', z' \), the formulas of transformations are

\[
x = X + x', \quad y = Y + y', \quad z = Z + z'.
\]

Let the angles made by the new axes of \( x, y, z \) with the old axes by \( \alpha, \beta, \gamma; \alpha', \beta', \gamma' \); \( \alpha'', \beta'', \gamma'' \) respectively. If we project the new coordinates on one of the old axes, the sum of the three projections will be equal to the projection of the radius vector, which is the corresponding old coordinate. Thus we get the three equations

\[
x = X \cos \alpha + Y \cos \alpha' + Z \cos \alpha''.
\]
\[
y = X \cos \beta + Y \cos \beta' + Z \cos \beta''.
\]
\[
z = X \cos \gamma + Y \cos \gamma' + Z \cos \gamma''.
\]

The new coordinates expressed in terms of the old are

\[
X = x \cos \alpha + y \cos \beta + z \cos \gamma
\]
\[
Y = x \cos \alpha' + y \cos \beta' + z \cos \gamma'
\]
\[
Z = x \cos \alpha'' + y \cos \beta'' + z \cos \gamma''.
\]
3.1 General Organization

TRACOR2 is a FORTRAN IV program that does two-dimensional coordinate transformations. It was designed and written to run on the PDP-11-40 computer.

The user has the option of having the program to perform a translation, a rotation, or both. The choice must be indicated on the first line of the data file. If the two coordinate systems that are being used have different origins with the axes of both systems being parallel, then the user should select option 1, a translation. If the new origin is the same as the original one and the new axes are obtained by revolving the original axes about this origin through some specified angle, then option 2, a rotation should be chosen. If the two coordinate systems have different origins and non-parallel axes, then option 3, a translation and rotation should be chosen.

The program will transform a maximum of ten (10)
vectors per system call. To execute the program, a data file containing only the specified data for the program must have been created. The format of the data file is given below:

Line 1 - contains an option of 1, 2, or 3.
   "1" indicates that a translation is to be done.
   "2" indicates that a rotation is to be done.
   "3" indicates that a translation and a rotation will be done.

Line 2 - Option 1 - the new origin \((x_o, y_o)\).
   Option 2 - the number \((NPT)\) of points that are to be transformed.
   Option 3 - the new origin \((x_o, y_o)\).

Line 3 - Option 1 - the number \((NPT)\) of points that are to be transformed.
   Option 2 - "1" or "2" to indicate that the angle will be entered in degrees (1) or radians (2).
   Option 3 - the number \((NPT)\) of vectors that are to be transformed.

Line 4 - 13 - Option 1 - the point, \(x(k), y(k)\) (relative to the original axes) that are to be transformed.
Ex.

Line 4  \( x(1), y(1) \)
Line 5  \( x(2), y(2) \) etc. for \( k=1,NPT \)

Line 4  - Option 2 - the angle (\( \text{THETA} \)).

Line 4  - Option 3 - "1" or "2" to indicate that the angle will be entered in degrees (1) or radians (2).

Line 5  - 14 - Option 2 - the points, \( x(k), y(k) \) (relative to the original axes) that are to be transformed.

Line 5  - Option 3 - the angle (\( \text{THETA} \)).

Line 6  - 15 - the points, \( x(k), y(k) \) (relative to the original axes) that are to be transformed.

Once the proper data file has been created, then the user may execute the program by typing the command "RUN TRACOR2". The program will ask for the name of an input file that contains the data and for an output file to which the results will be written. The program then does the necessary calculations and writes the results in the output file. The execution is terminated and the user may look at the results that are stored in the specified output file.

43
3.2 Flowchart of PROGRAM TRACOR2

Figure 3.2.1
READ ANGLE IN DEGREES

CONVERT ANGLE IN DEGREES TO ANGLE IN RADIANS

READ ANGLE IN RADIANS

READ INPUT VECTORS

CALCULATED TRANSFORMED VECTORS

WRITE OUTPUT IN FILE

STOP
3.3 Code for PROGRAM TRACOR2

PROGRAM TRACOR2
C THIS PROGRAM CALCULATES TRANSFORMED VECTORS IN TWO DIMENSIONS. THE USER PROVIDES A SET OF POINTS RELATIVE TO THE SET OF ORIGINAL AXES, A NEW ORIGIN AND AN ANGLE OF ROTATION.
REAL X(10), Y(10), XN(10), YN(10)
INTEGER N, NPT, OPT, OPT1, RESP

INSTRUCTIONS FOR THE USER

TYPE 5
5 FORMAT (IX, 'THIS PROGRAM DOES A TRANSFORMATION OF COORDINATES IN TWO DIMENSIONS. IT WILL CALCULATE A MAXIMUM OF TEN (10) VECTORS PER CALL."

TYPE 15
15 FORMAT (IX, ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW, LINE 1 - CONTAINS ALL OPTION OF 1,2, OR 3,
1 'INDICATES THAT A TRANSLATION IS TO BE DONE, '2' INDICATES THAT A ROTATION IS TO BE DONE, '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
2 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
3 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
4 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
5 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
6 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
7 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
8 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
9 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
10 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
11 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
12 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
13 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
14 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
15 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
16 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
17 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
18 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
19 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
20 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
21 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
22 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
23 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
24 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
25 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
26 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
27 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
28 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
29 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
30 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
31 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
32 ' TO RUN THE PROGRAM, YOU MUST HAVE CREATED A DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW,
33 'LINE 1 - INDICATES THAT A TRANSLATION IS TO BE DONE. '2' INDICATES THAT A ROTATION IS TO BE DONE '3' INDICATES THAT A TRANSLATION AND A ROTATION WILL BE DONE.
TYPE 25

25 FORMAT ('DO YOU HAVE A DATA FILE WITH THE PRECISE',/,
  • ' DATA IN IT? TYPE 1 FOR YES, 2 FOR NO.'
) READ (5,8) RESP
8 FORMAT (IX, IX)
C IF RESPONSE IS NO, END RUN
IF (RESP .EQ. 2) GO TO 200
DATA BLANK / ' ' /

TYPE 35

35 FORMAT (IX, 'INPUT FILE', $)
CALL ASSIGN (2, ' ', -1, 'RDO', 'NC', 1)

TYPE 45

45 FORMAT (IX, 'OUTPUT FILE', $)
CALL ASSIGN (2, ' ', -1, 'NEW', 'CC', 1)
READ (2, *) X0, Y0

READ (2, *) NPT
IF (OPT .EQ. 1) GO TO 40
READ (2, *) OPT 1
IF (OPT 1 .EQ. 2) GO TO 30

C OPT 1 - ANGLE IN DEGREES
READ (2, *) THETA
C CONVERT ANGLE IN DEGREES TO ANGLE IN RADIANS
CF = 0.0174539
THETA = THETA * CF
GO TO 40
C OPT 2 - ANGLES IN RADIANS
30 READ (2, *) THETA
C READ THE INPUT VECTORS (RELATIVE TO THE ORIGINAL
C AXES) THAT ARE TO BE TRANSFORMED.
40 DO 100 I = 1, NPT
READ (2, *) X(I), Y(I)
100 CONTINUE
IF (OPT .EQ. 2) GO TO 300
IF (OPT .EQ. 3) GO TO 450
C PERFORM THE TRANSLATION
DO 200 I = 1, NPT
XY(I) = X(I) - X0
YY(I) = Y(I) - Y0
200 CONTINUE
GO TO 700
C PERFORM THE ROTATION
300 DO 400 I = 1, NPT
\[
\begin{align*}
X_H(I) &= X(I) \cdot \cos(\Theta) + Y(I) \cdot \sin(\Theta) \\
Y_H(I) &= -X(I) \cdot \sin(\Theta) + Y(I) \cdot \cos(\Theta)
\end{align*}
\]

400 CONTINUE
GO TO 700

C PERFORM TRANSLATION AND ROTATION

DO 500 I = 1, NPT
\[
\begin{align*}
X_N(I) &= X(I) \cdot \cos(\Theta) + Y(I) \cdot \sin(\Theta) - X_0 \\
Y_N(I) &= -X(I) \cdot \sin(\Theta) + Y(I) \cdot \cos(\Theta) - Y_0
\end{align*}
\]
500 CONTINUE

700 TYPE 55

55 FORMAT ('TRANSFORMATION COMPLETE')
WRITE (5,55)

50 FORMAT (1,3X, 'POINT NO.', 10X, 'ORIGINAL POINTS', 13X
* ' TRANSFORMED POINTS', /
* ' 2DX, (X', 13X 'Y', 14X, 'XN', 13X, 'YN', /, 70 (1H -1)
DO 800 I = 1, NPT
WRITE (5,50) I, XN(I), YN(I), X(I), Y(I)

60 FORMAT (6X, 14, 4F15.4)
800 CONTINUE
STOP
RUN
4.1 General Organization

TRACOR3 is a FORTRAN IV program that calculates transformed vectors in three dimensions. It was designed to run on the PDP 11-40 computer.

The program will calculate a maximum of ten (10) vectors per system call. The user must provide an angle triplet; yaw, pitch, roll respectively or a direction cosine matrix in addition to the vectors that are to be transformed. To run the program, a data file containing only the data necessary for the execution of the program must have been created. The format of the data file is given below:

Line 1 - Contains an option of 1, 2, or 3.
   "1" indicates an angle triplet in degrees.
   "2" indicates an angle triplet in radians.
   "3" indicates that the ij component of a direction cosine matrix will be used.
Line 2 - The number (N) of vectors that will be transformed.

Line 3 - Option 1 - Angle triplet in degrees
Option 2 - Angle triplet in radians
Lines 3-5 - Option 3 - Direction cosine matrix M
Ex.
\[ M(1,1), M(1,2), M(1,3) \]
\[ M(2,1), M(2,2), M(2,3) \]
\[ M(3,1), M(3,2), M(3,3) \]

Lines 4-13 - Vector triplets, x(k), y(k), z(k)
or 6-15 for \( k = 1,N \)
Ex.
\[ x(1), y(1), z(1) \]
\[ x(2), y(2), z(3) \]
\[ x(3), y(3), z(3) \]

Once the proper data file has been created, the user may execute the program by typing the command "RUN TRACOR3". The program will ask for name of an input file that contains the data and for an output file to which the results will be written. Using system functions and Euler's formulas (see below), the program does the necessary calculations and writes the results in the chosen output file. The execution is terminated and the user may look at the results that are stored in the output file.
The more practical version of Euler's Equations of Transformation is given in the theorem below.

Theorem: Let \((x, y, z)\) be any point in space and consider an angle triplet

\(\text{Yaw, Pitch and Roll} = A, B, C\) respectively;

then

\[
x' = x(\cos C \cdot \cos A - \sin C \cdot \cos B \cdot \sin A) + y(\cos C \cdot \sin A + \sin C \cdot \cos B \cdot \cos A) + z(\sin C \cdot \sin B)
\]

\[
y' = -x(\sin C \cdot \cos A + \cos C \cdot \cos B \cdot \sin A) + y(-\sin C \cdot \sin A + \cos C \cdot \cos B \cdot \sin A) + z(\cos C \cdot \sin B)
\]

\[
z' = x(\sin B \cdot \sin A) - y(\sin B \cdot \cos A) + z \cdot \cos B
\]

are the equations of a transformation. These equations are known as Euler's formulas.
4.2 Flowchart of PROGRAM TRACOR3

START

INSTRUCTIONS FOR THE USER

INPUT FILE?

OUTPUT FILE?

READ OPTION

GPT=1,2,or 3?

READ ANGLES IN DEGREES

A

CONVERT ANGLES IN DEGREES TO ANGLES IN RADIANS

READ ANGLES IN RADIANS

CALCULATE THE COSINES OF THE ANGLES

CALCULATE THE SINES OF THE ANGLES

CALCULATE THE TRANSFORMED MATRIX M

READ THE I-J COMPONENT OF THE MATRIX M

B

Figure 4.2.1

52
READ THE NUMBER OF VECTORS

PRINT THE MATRIX M

READ THE INPUT VECTORS

CALCULATE THE TRANSFORMED VECTORS

WRITE THE TRANSFORMED VECTORS

STOP
THIS FORTRAN PROGRAM CALCULATES TRANSFORMED VECTORS IN
THREE DIMENSIONS. THE USER MUST PROVIDE AN ANGLE TRIPLET;
YAW, PITCH, ROLL RESPECTIVELY OR A DIRECTION COSINE MATRIX IN
ADDITION TO THE VECTORS THAT ARE TO BE TRANSFORMED.

INTEGER RESF, N, K, OPT
REAL X(Io), Y(Io), Z(Io), YAW, PITCH, ROLL
REAL CY, CR, SY, SF, SR

INSTRUCTIONS FOR THE USER

TYPE 1
1 FORMAT(IX, 'THIS PROGRAM DOES A TRANSFORMATION OF ',
2 ' COORDINATES IN THREE DIMENSIONS. IT WILL CALCULATE ',
3 ' A MAXIMUM OF TEN (10) VECTORS PER CALL.

TYPE 2
2 FORMAT('TO RUN THE PROGRAM, YOU MUST HAVE CREATED A',/
1 ' DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS',/
2 ' PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW',/
3 ' LINE 1 - CONTAINS AN OPTION OF 1, 2 OR 3',/
4 ' 1 - INDICATES THAT YOUR ANGLE TRIPLET IS IN DEGREES',/
5 ' 2 - ANGLE TRIPLET IN RADIANS',/
6 ' 3 - YOU WILL ENTER THE 3 COMPONENTS OF A DIRECTION',/
7 '.cosine matrix')

TYPE 15
15 FORMAT(IX, 'LINE 2 - THE NUMBER (N) OF VECTORS THAT ',
1 ' WILL BE ENTERED', /
2 ' LINE 3 - OPT 1 - ANGLE TRIPLET IN DEGREES', /
3 ' LINE 3 - OPT 2 - ANGLE TRIPLET IN RADIANS', /
4 ' LINES 3-5 - OPT 3 - DIRECTION COSINE MATRIX M; EX', /
5 ' M(1,1), M(1,2), M(1,3) ', /
6 ' M(2,1), M(2,2), M(2,3) ', /
7 ' M(3,1), M(3,2), M(3,3) ', /
8 ' LINES 4-13 OR 6-15 VECTOR TRIPLETS X(K), Y(K), Z(K) ', /
9 ' FOR K = 1 TO N; EX',)

TYPE 25
25 FORMAT(IX, 'X(1), Y(1), Z(1)', /
1 ' X(2), Y(2), Z(2)', /
2 ' X(3), Y(3), Z(3)')

TYPE 3
3 FORMAT('DO YOU HAVE A DATA FILE WITH THE PRECISE', /
* 'DATA IN IT? TYPE 1 FOR YES, 2 FOR NO.')

READ(5,10) RESF

10 FORMAT(IX, 11)

IF RESPONSE IS NO, END RUN
IF (RESF .EQ. 2) GO TO 600
DATA BANK: '/

TYPE 4
4 FORMAT(IX, 'INPUT FILE?', 8)
CALL ASSIGN(3,'', -1, 'PDO', 'NC', 1)
TYPE 5

FORMAT(IX,'OUTPUT FILE?'$,8)
CALL ASSIGN(7,'',-1,'NEW', 'CC',1)
READ(3,*)OPT

IF(OPT .EQ. 2) GO TO 200
IF(OPT .EQ. 3) GO TO 300

C
OPT 1 - ANGLES IN DEGREES
READ(3,*) YAW, PITCH, ROLL

C CONVERT ANGLES IN DEGREES TO ANGLES IN RADIANS
CF = 0.01745329561
YAW = YAW*CF
PITCH = PITCH*CF
ROLL = ROLL*CF
GO TO 250

C
C OPT 2 - ANGLES IN RADIANS
200 READ(3,*) YAW, PITCH, ROLL

C CALCULATE THE COSINES OF ANGLES YAW, PITCH, AND ROLL.
250 CY = COS(YAW)
CP = COS(PITCH)
CR = COS(ROLL)

C CALCULATE THE SINES OF ANGLES YAW, PITCH, AND ROLL.

C
SY = SIN(YAW)
SP = SIN(PITCH)
SR = SIN(ROLL)

C CALCULATES THE TRANSFORMED MATRIX M

ML1 = CR*CY-SR*CF*SY
ML2 = CR*CY+SR*CF*CY
ML3 = CR*SP
M21 = -SR*CY-CR*CF*SY
M22 = -SR*CY+CR*CF*CY
M23 = CR*SP
M31 = SP*SY
M32 = SP*CY
M33 = CF
GO TO 300

C
C OPT 3 - IJ COMPONENT OF THE MATRIX M.
300 READ(3,*) ML1, ML2, ML3
READ(3,*) M21, M22, M23
READ(3,*) M31, M32, M33

C
READ THE NUMBER OF VECTORS
400 READ(3,*)N

55
PRINT THE MATRIX K

WRITE(7, 20)
FORMAT(1X, 'DIRECTION COSINE MATRIX')

WRITE(7, 30) M11, M12, M13
FORMAT(1X, F10.4, 5X, F10.4, 5X, F10.4)

WRITE(7, 40) M21, M22, M23
FORMAT(1X, F10.4, 5X, F10.4, 5X, F10.4)

WRITE(7, 50) M31, M32, M33
FORMAT(1X, F10.4, 5X, F10.4, 5X, F10.4)

READ THE INPUT VECTORS.

WRITE(7, 120)
FORMAT(1X, 'VECT NO.', 5X, 'TRANVECT', 15X, 'INPUT VECT')
DO 500 K = 1, N
READ(3, *) X(K), Y(K), Z(K)
V111 = X(K)
V121 = Y(K)
V131 = Z(K)

CALCULATE THE TRANSFORMED VECTORS V2 AND WRITE IN FILE.

V211 = K11*V111 + K12*V121 + M13*V131
V221 = M21*V111 + M22*V121 + M23*V131
V231 = M31*V111 + M32*V121 + M33*V131

WRITE(7, 60) K, V211, V221, V231, V111, V121, V131
500 CONTINUE

WRITE(7, 70)
FORMAT(1X, 'ANGLES ENTERED')

WRITE(7, 80) YAW
FORMAT(1X, 'YAW = ', F10.4)

WRITE(7, 90) PITCH
FORMAT(1X, 'PITCH = ', F10.4)

WRITE(7, 100) ROLL
FORMAT(1X, 'ROLL = ', F10.4)

WRITE(7, 110) N
FORMAT(1X, 'NO. OF VECTORS ENTERED', 4X, I3)

GO TO 700

DATA FILE HAS NOT BEEN CREATED

600 TYPE 8
FORMAT('THIS PROGRAM HAS BEEN TERMINATED', /,
'CREATE A DATA FILE AND RUN AGAIN.')

700 STOP
END
Chapter V contains sample runs of PROGRAM TRACOR2 and PROGRAM TRACOR3. The user has three options when executing either program. Two sample runs of each option will be presented. The input files containing the data and the output files containing the results will also be presented.

For TRACOR2 the input files are: for option 1, NOS11.DAT and NOS12.DAT; for option 2, NOS21.DAT and NOS22.DAT; and for option 3, NOS31.DAT and NOS32.DAT. The corresponding output files are: for option 1, NOS11.OUT and NOS12.OUT; for option 2, NOS21.OUT and NOS22.OUT; and for option 3, NOS31.OUT and NOS32.OUT.

For TRACOR3 the input files are: for option 1, NUM11.DAT and NUM12.DAT; for option 2, NUM21.DAT and NUM22.DAT; and for option 3, NUM31.DAT and NUM32.DAT. The corresponding output files are: for option 1, NUM11.OUT and NUM12.OUT; for option 2, NUM21.OUT and NUM22.OUT; and for option 3, NUM31.OUT and NUM32.OUT.
5.1 Sample Runs of TRACOR2 and TRACOP3

This program does a transformation of coordinates in two dimensions. It will calculate a maximum of ten (10) vectors per call. To run the program, you must have created a data file with only the data necessary to run this program. The data must be as specified below:

Line 1 - contains an option of 1, 2, or 3:
1 indicates that a translation is to be done.
2 indicates that a rotation is to be done.
3 indicates that a translation and a rotation will be done.

Line 2 - Option 1 - the new origin (Xo, Yo).
Option 2 - the number (Npt) of points that are to be transformed.
Option 3 - the new origin (Xo, Yo).

Line 3 - Option 1 - the number (Npt) of points that are to be transformed.
Option 2 - 1 or 2 to indicate that the angle will be entered in degrees (1) or radians (2).
Option 3 - the number (Npt) of vectors that are to be transformed.

Lines 4-13 - Option 1 - the points, X(K), Y(K) (relative to the original axes) that are to be transformed.
Ex.
Line 4 X(1), Y(1)
Line 5 X(2), Y(2) etc. for K=1, Npt
Line 4 - Option 2 - the angle (Theta).
Line 4 - Option 3 - 1 or 2 to indicate that the angle will be entered in degrees (1) or radians (2).
Line 5-14 - Option 2 - the points, X(K), Y(K) (relative to the original axes) that are to be transformed.
Line 5 - Option 3 - the angle (Theta).
Line 6-15 - the points, X(K), Y(K) (relative to the original axes) that are to be transformed.

Do you have a data file with the precise data in it? Type 1 for yes, 2 for no.

1

58
**LOOK NOS11.OUT**

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>ORIGINAL POINTS</th>
<th>TRANSFORMED POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>6.0000</td>
<td>8.0000</td>
</tr>
<tr>
<td>3</td>
<td>10.0000</td>
<td>-20.0000</td>
</tr>
<tr>
<td>4</td>
<td>5.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.0000</td>
<td>9.0000</td>
</tr>
<tr>
<td>6</td>
<td>-1.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

**NOS11.DAT**

1,
3,4
6
2,1
6,8
10,-20
5,-
0,9
-1,0
LOOK NOS12.OUT

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>ORIGINAL POINTS</th>
<th>TRANSFORMED POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>10.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>-5.0000</td>
<td>7.0000</td>
</tr>
<tr>
<td>3</td>
<td>3.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>4</td>
<td>8.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>5</td>
<td>3.0000</td>
<td>2.0000</td>
</tr>
</tbody>
</table>

NOS12.DAT

1
0.5
5
10.2
-5.7
-3.4
8.2
3.3
LOOK NOS21.OUT

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>ORIGINAL POINTS</th>
<th>TRANSFORMED POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>2</td>
<td>5.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>3</td>
<td>1.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>4</td>
<td>-2.0000</td>
<td>-5.0000</td>
</tr>
</tbody>
</table>

NOS21.DAT

2
4
1
45
2,3
5,4
1,2
-2,-5
LOOK NOS22.OUT

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>ORIGINAL POINTS</th>
<th>TRANSFORMED POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>2.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>2</td>
<td>5.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>3</td>
<td>1.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>4</td>
<td>-2.0000</td>
<td>-5.0000</td>
</tr>
</tbody>
</table>

NOS22.DAT

2
4
2
.7853
2,3
5,4
1,2
-2,-5
LOOK NOS31.OUT

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>ORIGINAL POINTS</th>
<th>TRANSFORMED POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>XN</td>
<td>YN</td>
</tr>
<tr>
<td>1</td>
<td>4.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td></td>
<td>0.4640</td>
<td>-3.5360</td>
</tr>
<tr>
<td>2</td>
<td>3.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td></td>
<td>1.0980</td>
<td>-1.9020</td>
</tr>
<tr>
<td>3</td>
<td>8.0000</td>
<td>8.0000</td>
</tr>
<tr>
<td></td>
<td>2.9280</td>
<td>-5.0720</td>
</tr>
</tbody>
</table>

NOS31.DAT
3
5,5
3
1
30
4,4
3,3
8,8
LOOK NOS32.OUT

<table>
<thead>
<tr>
<th>POINT NO.</th>
<th>ORIGINAL POINTS</th>
<th>TRANSFORMED POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>4.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>2</td>
<td>3.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>3</td>
<td>3.0000</td>
<td>3.0000</td>
</tr>
</tbody>
</table>

NOS32.DAT

3
5,5
3
2
.5235
4,4
3,3
6,8
THIS PROGRAM DOES A TRANSFORMATION OF
COORDINATES IN THREE DIMENSIONS. IT WILL CALCULATE
A MAXIMUM OF TEN (10) VECTORS PER CALL.
TO RUN THE PROGRAM, YOU MUST HAVE CREATED A
DATA FILE WITH ONLY THE DATA NECESSARY TO RUN THIS
PROGRAM. THE DATA MUST BE AS SPECIFIED BELOW
LINE 1 - CONTAINS AN OPTION OF 1, 2 OR 3
1 - INDICATES THAT YOUR ANGLE TRIPLET IS IN DEGREES
2 - ANGLE TRIPLET IN RADIANS
3 - YOU WILL ENTER THE IJ COMPONENT OF A DIRECTION
COSINE MATRIX
LINE 2 - THE NUMBER (N) OF VECTORS THAT
WILL BE ENTERED
LINE 3 - OPT 1 - ANGLE TRIPLET IN DEGREES
LINE 3 OPT 1 - ANGLE TRIPLET IN RADIANS
LINES 3-5-OPT 3 - DIRECTION COSINE MATRIX M; EX.
M(1,1), M(1,2), M(1,3)
M(2,1), M(2,2), M(2,3)
M(3,1), M(3,2), M(3,3)
LINES 4-13 or 6-15 VECTOR TRIPLETS, X(K), Y(K), Z(K)
FOR K = 1 TO N; EX,
X(1), Y(1), Z(1)
X(2), Y(2), Z(2)
X(3), Y(3), Z(3)
DO YOU HAVE A DATA FILE WITH THE PRECISE
DATA IN IT? TYPE 1 FOR YES, 2 FOR NO.
1
NUM11.DAT

1
3
35,78,102
-1,-8,-9
34,67,-90
-234,456,-341

NUM11.OUT

DIRECTION COSINE MATRIX
-0.0286  0.0473  -0.2033
-0.6819  -0.7313  -0.2033
0.5610   0.8012   0.2079

VECT NO   TRANS VECT       INPUT VECT
    1    1.48    -7.00     -5.09    -1    -8    9
    2   20.50   -53.82     54.64     34    67   -90
    3  133.24  -105.56    165.56   -234    456  -341

ANGLES ENTERED

YAW = 0.6108
PITCH = 1.3613
ROLL = 1.7802

NO. OF VECTORS ENTERED 3
NUM12.DAT

1
3
30,50,60
35,65,85
-70,-40,-50
-80,-100,-20

NUM12.OUT

DIRECTION COSINE MATRIX
0.2782 0.7919 0.3830
-0.8239 -0.1545 0.3800
0.3330 0.6633 0.5000

<table>
<thead>
<tr>
<th>VECT NO</th>
<th>TRANS VECT</th>
<th>INPUT VECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89.89</td>
<td>35 65 85</td>
</tr>
<tr>
<td>2</td>
<td>67.90</td>
<td>78.34 -70 -40 -50</td>
</tr>
<tr>
<td>3</td>
<td>3.17</td>
<td>-2 3 h</td>
</tr>
</tbody>
</table>

ANGLES ENTERED
YAW = 0.5235
PITCH = 0.8726
ROLL = 1.0471

NO. OF VECTORS ENTERED 3
NUM21.DAT

2
3
.52,.69,.89
1,2,3
4,5,6
5,6,7

NUM21.OUT

DIRECTION COSINE MATRIX

<table>
<thead>
<tr>
<th></th>
<th>0.2632</th>
<th>0.8294</th>
<th>0.4130</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.8027</td>
<td>0.0433</td>
<td>0.4130</td>
<td></td>
</tr>
<tr>
<td>0.3213</td>
<td>0.5565</td>
<td>0.7660</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VECT NO</th>
<th>TRANS VECT</th>
<th>INPUT VECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.16</td>
<td>0.52</td>
</tr>
<tr>
<td>2</td>
<td>7.67</td>
<td>-1.31</td>
</tr>
<tr>
<td>3</td>
<td>9.18</td>
<td>-0.86</td>
</tr>
</tbody>
</table>

ANGLES ENTERED

YAW = 0.5235
PITCH = 0.6981
ROLL = 0.7826

NO. OF VECTORS ENTERED 3
NUM22.DAT

2
2
.52,.87,1.04
1  2  3
4 -5  6

NUM22.OUT

DIRECTION COSINE MATRIX
0.2782  0.7319  0.3830
-0.8239 -0.1543  0.3830
  0.3830  0.6633  0.5000

VECT NO  TRANS VECT   INPUT VECT
1   2.89  0.00  3.20  1  2  3
2  -0.80 -0.24  1.21  4  -5  6

ANGLES ENTERED

YAW = 0.5235
PITCH = 0.8726
ROLL = 1.0471

NO. OF VECTORS ENTERED 2
NUM31.DAT

3
2
0.2782, 0.7319, 0.3830
-0.8239, -0.1548, 0.3830
0.3830, 0.6633, 0.5000
1, 2, 3
4, -5, 6

NUM31.OUT

DIRECTION COSINE MATRIX
0.2782 0.7319 0.3830
-0.8239 -0.1548 0.3830
0.3830 0.6633 0.5000

VECT NO  TRANS VECT  INPUT VECT
1 2.89  0.00  3.20  1 2 3
2 -0.80  -0.24  1.21  4  -5  6

ANGLES ENTERED
YAW = 0.0000
PITCH = 0.0000
ROLL = 0.0000

NO. OF VECTORS ENTERED 2
NUM32.DAT

3
3
0.2632, 0.8294, 0.4130
-0.8027, 0.0433, 0.4130
0.3213, 0.5565, 0.7660
1, 2, 3
4, 5, 6
5, 6, 7

NUM32.OUT

VECT NO | TRANS VECT | INPUT VECT
-------|------------|----------
1       | 3.16       | 3.73     | 1 2 3
2       | 7.67       | -1.31    | 4 5 6
3       | 9.18       | -0.86    | 5 6 7

ANGLES ENTERED

YAW = 0.0000
PITCH = 0.0000
ROLL = 0.0000

NO. OF VECTORS ENTERED 3
5.2 Graphic Manipulations Using Matrices

A. Introduction

According to Webster dictionary, a definition of graphics is a picture, map, or graph used for illustration or demonstration. Graphics may also be defined as a means to convert data into clear or vivid information. A small part of graphics is the conversion of data representing the position in space of the surface of an object into a three-dimensional picture of the same object.

The relation between the position of a point on an object, specified in three dimensions x, y and z, and the apparent translation of the point as the object is rotated is expressed by standard equations. These equations can be written in an organized form using matrices. These matrices are called coordinate transform matrices. They separate the mathematics associated with the angle of observation from the data describing the surface of an object. The small computers such as the APPLE II and such terminals as the TEKTRONIX Graphics Terminal make it easy to convert surface data to pictorial information using matrices.

B. Definitions

The Cartesian coordinate system will be used in this discussion of graphic manipulations. On a computer graphic
screen, \( x \) is defined as the horizontal dimension, \( y \) is the vertical dimension, and \( z \) is depth into the screen. Positive values are to the right on the \( x \)-axis, up on the \( y \)-axis, and away from the viewer on the \( z \)-axis. The origin will be in the center of the screen. A plane is described by any two axes. The \( x,y \) plane is the surface of the computer screen, the \( y,z \) plane is seen edge on as the \( y \) axis, and the \( x,z \) plane is seen edge on as the \( x \) axis.

Two kinds of matrices will be used. The first kind of matrix will represent the coordinate of a point in space, described by \( x,y \), and \( z \). This matrix, \( C \), is a column matrix, with one column and three rows as shown in figure 5.2.1. It will be used to hold the value for a point on the surface, both before and after transformation. All drawing commands will use the \( x \) and \( y \) from this matrix.

The second kind of matrix contains the standard equations which relate the transformed values for \( x,y \), and \( z \) to the original values. This matrix, \( M \), has three rows and three columns as shown in figure 5.2.1. The numbers at each position in this matrix are derived from the standard equations for some specified coordinates transformation type, such as a rotation. If the angle
Figure 5.2.1: Two types of matrices used in graphics applications. \( C \) represents a point in space. The matrix \( M \) contains equations to relate the transformed values of \( x, y, \) and \( z \) to the original values.
of observation of the object is arrived at by two successive operations, such as a rotation about each of two axes, then the matrix, $M$, which controls this view may have numbers in it that are derived from two matrix equations. The procedure which combines several operations and produces a single matrix, is called matrix multiplication. Matrix multiplication is used to produce the numbers and the $3 \times 3$ matrix, $M$, and then to apply the numbers to derive a transformed $3 \times 1$ matrix, $C$, which gives the apparent position of some part of the object's surface.

C. Computing Procedure

Given the ability to do matrix multiplication, the procedure for computing transformations is outlined as follows:

1. Generate an array of data consisting of the $x,y$ and $z$ coordinates of the object to be drawn.
2. Define the viewpoint. A matrix is generated for each motion required to arrive at the desired point of view. The matrices are then multiplied together to produce a single $3 \times 3$ matrix to be used in the main routine.
3. Write a program to draw the object in its untransformed state.
4. When the untransformed picture is accurate, then
the picture is reoriented. The original $x,y,z$ coordinates of the spot on the object are transformed into a new set of $x,y,z$ numbers representing the spot seen from the new viewpoint. Each set of coordinates from Step 1 is multiplied by the matrix generated in Step 2.

Since the computer screen is actually only two-dimensional, only the $x$ and $y$ elements are used in Step 3. After transformation, the numbers are shifted about and contain depth information. The drawing made in this way is a projected view rather than a three-dimensional drawing.

D. Types of Transformations

There are other operations, in addition to shifting and rotating a picture, possibly using coordinate transform matrices. Some of the more extremely effectual changes that may be made to the picture are: magnification, shear, stretching in one direction, and mapping onto a surface.

The following figures and tables show examples of several coordinate transformations. Each operation in represented by the matrix, the equations for the output matrix elements, and a figure to show how a picture of an object is changed by the operation.
1. Unit Matrix

Table 5.2.1 shows the unit matrix. This transformation reproduces the picture of the object in its original view as defined in section 5.3. Figure 5.2.2 shows a side view of a coffee cup.

2. Scaling

The object may be defined with some shape which is not the final version, but which uses easily verified dimensions.

Selective magnification along any axis may be used to alter the proportions of the object to the shape that may be nearer that required by the user. Thus, table 5.2.2 and figure 5.2.3 are appropriate for users with large capacities: table 5.2.3 and figure 5.2.4 produce a wide cup, while table 5.2.4 and figure 5.2.5 produce a large cup.

If the proportions are correct but the user wants to create several different sizes, table 5.2.5 will allow the same magnification to take place along all the axes. Figure 5.2.6 shows one such magnified cup.

3. Combinations

Table 5.2.6 shows an example of the sequential application of two rotations using matrix
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
= \begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix}
\]

\[
X' = (X \cdot 1 + Y \cdot 0 + Z \cdot 0)
\]

\[
Y' = (X \cdot 0 + Y \cdot 1 + Z \cdot 0)
\]

\[
Z' = (X \cdot 0 + Y \cdot 0 + Z \cdot 1)
\]

Table 5.2.1: The unit matrix.

Figure 5.2.2: The standard (untransformed) view of a coffee cup.
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & K & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
=
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix}
\]

\[
X' = (X \cdot 1 + Y \cdot 0 + Z \cdot 0) \\
Y' = (X \cdot 0 + Y \cdot K + Z \cdot 0) \\
Z' = (X \cdot 0 + Y \cdot 0 + Z \cdot 1)
\]

Table 5.2.2: Expansion along the y-axis by a factor K.
Figure 5.2.3: Expanded along the y axis by a factor of 1.5.
\[
\begin{array}{ccc|ccc}
K & 0 & 0 & X & X' \\
0 & 1 & 0 & Y & Y' \\
0 & 0 & 1 & Z & Z' \\
\end{array}
\]

\[
X' = (X \times K + Y \times 0 + Z \times 0)
\]

\[
Y' = (X \times 0 + Y \times 1 + X \times 0)
\]

\[
Z' = (X \times 0 + Y \times 0 + Z \times 1)
\]

Table 5.2.3: Expansion along the x axis by a factor of K.

Figure 5.2.4: Expanded along the x axis by a factor of 1.5.
Table 5.2.4: Expansion along the z axis by a factor K.

<table>
<thead>
<tr>
<th>1 0 0</th>
<th>X</th>
<th>X'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0</td>
<td>Y</td>
<td>Y'</td>
</tr>
<tr>
<td>0 0 K</td>
<td>Z</td>
<td>Z'</td>
</tr>
</tbody>
</table>

\[
X' = (X \times 1 + Y \times 0 + Z \times 0)
\]
\[
Y' = (X \times 0 + Y \times 1 + Z \times 0)
\]
\[
Z' = (X \times 0 + Y \times 0 + Z \times K)
\]

Figure 5.2.5: Rotated 90° and expanded along z axis by a factor of 1.5.
\[
\begin{array}{ccc|ccc}
K & 0 & 0 & X & X' \\
0 & K & 0 & Y & Y' \\
0 & 0 & K & Z & Z' \\
\end{array}
\]

\[
X' = (X \times K + Y \times 0 + X \times 0) \\
Y' = (X \times 0 + Y \times K + Z \times 0) \\
Z' = (X \times 0 + Y \times 0 + Z \times K)
\]

Table 5.2.5: Magnification by a factor K.

Figure 5.2.6: Magnified by a factor of 0.5.
Table 5.2.6: Rotation in the x,y and y,z planes sequentially.
multiplication to define the elements. Figure 5.2.7 combines the three rotations, a magnification and two scale changes along individual axes. Figure 5.2.8 applies a rotation.

4. Translation

Moving the object sideways is accomplished by adding the translation value to the original value for each axis as shown in table 5.2.7.

5. Mapping

All of the examples shown thus far have applied the same operation to every point on the surface of the object. The coordinate transform matrix contains the same numbers for all parts of the object. A more complicated operation results when the value (x,y,z) associated with a point on the surface of the object is replaced by a function of x,y, and z at each point. The result maps the object onto some surface. Thus if x is replaced by sin(x), the object will experience a change in scale which changes with x, and the output position of a point on the object which has a value of x equal to 0, and an arbitrary y and z is the same as a point with the value of x equal to 180° or 360°.
\[
\begin{bmatrix}
X + k \\
Y + m \\
Z + n
\end{bmatrix}
= 
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix}
\]

\[
X' = X' + k \\
Y' = Y' + m \\
Z' = Z' + n
\]

Table 5.2.7: Translation matrix.
Figure 5.2.7: Magnified expanded, and rotated.

Figure 5.2.8: Rotated but not sheared.
In order to make a three-dimensional picture of the mapped figure, the other two dimensions must be specified too. The depth must also be mapped by setting $z$ equal to $\cos(x)$. The new surface is a cylinder, thus $y$ remains unchanged.

F. Mapping Uses

Mapping may be used to compress the picture of an object to fit into a particular space or to emphasize some part while compressing the rest. It may also be used to picture some complicated configuration which is more easily defined in a rectangular coordinate system.


