## NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.



## AVIS

La qualité de cette microforme dépe[illegible] grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

THE UNIVERSITY OF ALBERTA

# The Effects of Craniofacial Morphology on Human Temporomandibular Joint Loading

BY

Steven Patrick McEvoy

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

EDMONTON, ALBERTA, CANADA

SPRING 1990

ISBN 0-315-60265-1

December 17, 1989

To Whom it May Concern:

This is a letter of permission giving Steven Patrick McEvoy the rights to reprint any portion of my 1985 University of Alberta Masters Thesis **A Mechanical Analysis of the Temporomandibular Joint.**

Andrew Stuart Hay

Permanent Address:
Apt. #2701
8210-111 Street
Edmonton, ALberta
T6G 2C7

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Steven Patrick McEvoy

TITLE OF THESIS: The Effects of Craniofacial Morphology on Human Temporomandibular Joint Loading

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1990



Steven P. McEvoy

11420 - 33A -Avenue
Edmonton, Alberta, Canada
T6J 3H3

Date: February 20, 1990

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **The Effects of Craniofacial Morphology on Human Temporomandibular Joint Loading** submitted by **Steven Patrick McEvoy** in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in MECHANICAL ENGINEERING.

Gary Faulkner

Dr. M.G. Faulkner

Kenneth Glover

Dr. K. Glover

Dr. D. Budney

Date: December 21, 1989

To my Mom and Dad

## Abstract

The purpose of this study was to develop a numerical modelling system capable of investigating the effects of craniofacial morphologies on human temporomandibular joint loading. Seventy human skulls were digitized for spatial information required to conduct the study. The skulls were classified as Angle's skeletal Class I, II or III for comparative analysis of their TMJ loading patterns. Classes were compared by the levels of resultant ipsilateral and contralateral condylar load, occlusal load magnitude and the level of angular misalignment of the ipsilateral condylar load and the articular eminence it is incident upon.

Experimental evidence showed that (1) Angle's Class IIs exhibit moderately higher levels of resultant TMJ loads while producing lower levels of occlusal load compared to the Class IIIs; (2) The Class IIs TMJ loads display higher levels of variability than the Class IIIs; (3) Class II morphologies function with a consistently higher level of resultant TMJ load misalignment than either Class I or Class III. This combination of factors may be a significant factor towards explaining the higher levels of TMJ dysfunction in the Class II population. In contrast Angle's Class IIIs showed significantly lower levels of resultant condylar loads while developing the highest levels of occlusal load. Class IIIs exhibit the lowest level of load variability, and most importantly the alignment of the TM joint loads is nearest to "ideal" of the three morphologies.

**Acknowledgement**

The author would like to express his deepest appreciation to the multitude of people who contributed towards this work, and without whose help this investigation would not have been possible, thanks to:

-Drs. Faulkner, Hatcher and Glover from the University of Alberta whose timely guidance, contributions and encouragement kept me going.

-Dr. Kirk Chambers for enduring a free trip to San Francisco, editorial contributions and his infinite patients digitizing the 70 sample skulls.

-Dr. Bonnie Chambers for letting Kirk go, and her editorial help.

-Dr. Mitchell Day (alias Dr. Detroit) and the University of the Pacific for their help and use of the Atkinson Cranio-Osteological collection.

-Mr. Andrew Hay for helping me pick the project up.

-Joanne White for the countless hours and effort she contributed to produce the anatomical drawings.

-Special thanks to Karen McKelvie B.Sc. for enduring the experience and seeing me through the tough times.

and final thanks to Babs, Doris, Gail, Helen, Rob, Rich, Ray, Shamim, Victor and anyone else that I missed that helped me through.

# Table of Contents

# List of Tables

# List of Figures

# List of Plates

# List of Nomenclature

**Angle's Classification:** a classification system developed by Edward H. Angle based on the relationship of the maxilla and the mandibular first permanent molar. It was later adapted to further describe the relationship of the two denture bases to each other.

**Attrition Facet:** a small smooth area on a bone or other firm structure, like a worn spot on a tooth produced by mastication.

**Balancing Side:** the opposite side of the mandible from which the occlusion is occurring.

**Bilateral:** affecting both sides.

**Bolus:** a small round lump or mass.

**Compact Bone:** densely packed bone consisting largely of concentric lamellar osteons and interstitial lamellae.

**Condyle:** the rounded process at the end of a bone that articulates within a fossa.

**Contralateral:** refers to the opposite side of the mandible on which occlusion is occurring.

**Cranial:** pertaining to the skull.

**Curve of Spee:** anatomical curvature of the occlusal alignment of teeth beginning at the tip of the lower cuspid following the buccal cusps of the natural bicuspids and molars, continuing to the anterior border of the ramus.

**Cusp:** the elevations on the chewing surface of the tooth.

**Dentition:** relating to the arrangement of teeth in the mouth.

**Digitized:** an informal term describing that an items' three dimensional spatial information was measured electronically and then stored.

**Electromyography:** the electrical response of a muscle tissue to nerve stimulation.

**Eminence:** a raised area of bone.

**Foramen Magnum:** the opening at the base of the skull through which the spinal cord passes.

**Force Vector:** a vector which defines both the direction and magnitude of the force.

**Fossa:** a cavity or small hollow.

**Frankfort Plane:** A horizontal plane passing through the most inferior point in the margin of the left and right orbit and the midpoint of a line connecting the most superior points on the margins of the left and right external auditory meatus.

**Incisal:** articular portion of the incisal teeth.

**Incisors:** referring to the front teeth between the canines in either jaw.

**Insertion:** the point of attachment of a muscle to the part that it moves.

**Interdigitation:** the mutual interlocking of toothed processes (i.e. upper cusp to lower fossa relationship).

**Ipsilateral:** refers to the same side of the mandible on which occlusion is occurring.

**Ligament:** a band of tough tissue connecting bones which limits travel.

**Mandible:** the lower jaw.

**Mastication:** chewing or grinding food.

**Maxilla:** the upper jaw.

**Moment:** the product of a force and its distance from a reference point.

**Numerical Model:** the process of simulating a real life situation with mathematical techniques.

**Occlusal Angle:** the angle between a vector normal to the occlusal plane and the line of action of the occlusal force.

**Occlusal Plane:** the plane created by using the maxillary first molars and first premolars (see section 3.4 for a more detailed description).

**Occlusion:** the way in which the upper and lower teeth fit together.

**Orbit:** orbit cavity, the bony cavity containing the eyeball and its adnexa.

**Origin:** the least movable of the two points of attachment of a muscle, usually to the more rigid part of the skeleton.

**Orthogonal:** at right angle.

**Osseous:** bony.

**Parasagittal Plane:** a plane parallel to the midsagittal plane.

**Process:** a projection or outgrowth of bone.

**Reaction:** an opposing force.

**Sagittal Plane:** a plane orientated inferosuperior that passes through the anterior and posterior nasal spines.

**Statically Determinate:** a structural numerical problem in which the unknowns may be solved for by using the equations of statics.

**Synovial Fluid:** clear lubricating fluid secreted by the membranes of the joint cavities.

**Trabecular Bone:** bone in which trabeculae are interconnected to form a latticework with interstices filled with connective tissue or bone marrow.

**Unilateral:** one side only.

**Unit Vector:** a vector whose length is unity.

**Working Side:** the side of the mandible on which unilateral mastication is occurring.

# 1.0 Introduction

## 1.1 Introduction of the Problem

In recent years there has been an increased awareness of Craniomandibular disorders or Temporomandibular Joint (TMJ) dysfunction in the general populace, thus stimulating research to better understand the complex function and dysfunction of these joints.

Of particular interest in joint dysfunction is the issue of TMJ failure which can be defined as a degenerative response of the articular tissues of the TMJs when the adaptive ability (functional range of these tissues) is exceeded. The joint tissues demand function in order to develop the specific properties required to maintain that function. When the tissues do not develop the characteristics necessary to maintain a level of function commensurate with the demands placed upon them, these tissues may be damaged and progress to joint failure. The factors involved in predisposing or leading to joint failure are an intense area of interest in current TMJ research.

There are many factors that effect the health of the articular tissues. Some of these factors include the tissues' adaptive potential and the characteristics of the loads incident on these tissues. The loading characteristics include direction, magnitude, frequency and duration of the appositional forces on the articular surfaces during mandibular function.

It has been observed that some joint surfaces demonstrate a change in form, such as faceting or flattening, when experiencing joint dysfunction. This flattening suggests that these articular surfaces have altered their form (osseous remodelling) to reduce the load per unit area (stress) by distributing the applied load over a larger surface area.

Examination of the structure and morphology of the joint tissues suggest that these tissues can maintain a higher functional demand when the loading forces are primarily compressive (loads applied perpendicular to the joint surfaces) [41]. The idealization towards compressive loads during function implies that the line of action for the load should be perpendicular to each of the opposing surfaces. This can be best achieved when the facets of the opposing articular surfaces for the condyle and eminence are parallel to each other with an articular disc interposed.

The objective of the present study was to determine the effects that variations in craniofacial morphology have on the specific TMJ loading characteristics of magnitude and direction. It was hypothesized that certain differences in facial morphology, based upon Angle's classification, lead to variations in resultant load direction and magnitude, specifically that there are higher levels of posteriorly directed shear loads in Class II's. Finally, it is proposed that comparison of the these loading characteristics to the facial morphologies will suggest that certain morphological types are more predisposed to TM joint failures than others.

## 1.2 Anatomy of the TMJ

The following sections on anatomy contains excerpts from Mr. A. Hay's 1985 masters thesis "A Mechanical Analysis of the Temporomandibular Joint" [10]. These have been augmented to assist in explaining the details of the current study.

### 1.2.1 Structure

The description of the features of the skull and mandible is accomplished by reference to various standard anatomical planes and directions (see Figure #1.1). The human TMJ includes the condylar process, an articular disc, the articular eminence and glenoid fossa of the temporal bone, and associated ligaments (Figure #1.2). Compact bone covers the outer surfaces of the fossa, eminence, and condyle. The outer surfaces are supported by an inner network of trabecular bone and are covered externally by a thin layer of fibrous connective tissue. The mandible (see Figure #1.3) is similarity constructed from an outer layer of compact bone and an inner network of trabecular bone. One of its functions is to support the teeth and the occlusal loads imposed on them. The teeth are supported in the mandible in bony sockets by periodontal ligaments that join the root surface to the underlying bone. During forceful occlusal contact, the forces are transferred from the teeth by the periodontal ligaments into the supporting structure of the trabecular bone and then into the dense compact bone of the mandible [35, 36].

The articular disc is a pad of fibrous connective tissue which is interposed between the condyle and the eminence/fossa structure, separating the joint into upper and lower compartments. Lubrication, nutrition and protection of the joint surfaces is provided by synovial fluid. The joint and associated structures are encapsulated by a capsular ligament with other ligaments, the lateral ligament, the sphenomandibular ligament and the stylomandibular ligament (Figure #1.4) acting to stabilize and control movements within tolerable limits. Because of the unique structure of the joint, the condyle can be translated forward onto the eminence, rotated in the sagittal plane, and rotated in the coronal plane (Figure #1.5). The mobility of this joint makes it unique within the body, however it also means that the conditions which lead to dysfunctions are extremely complex.

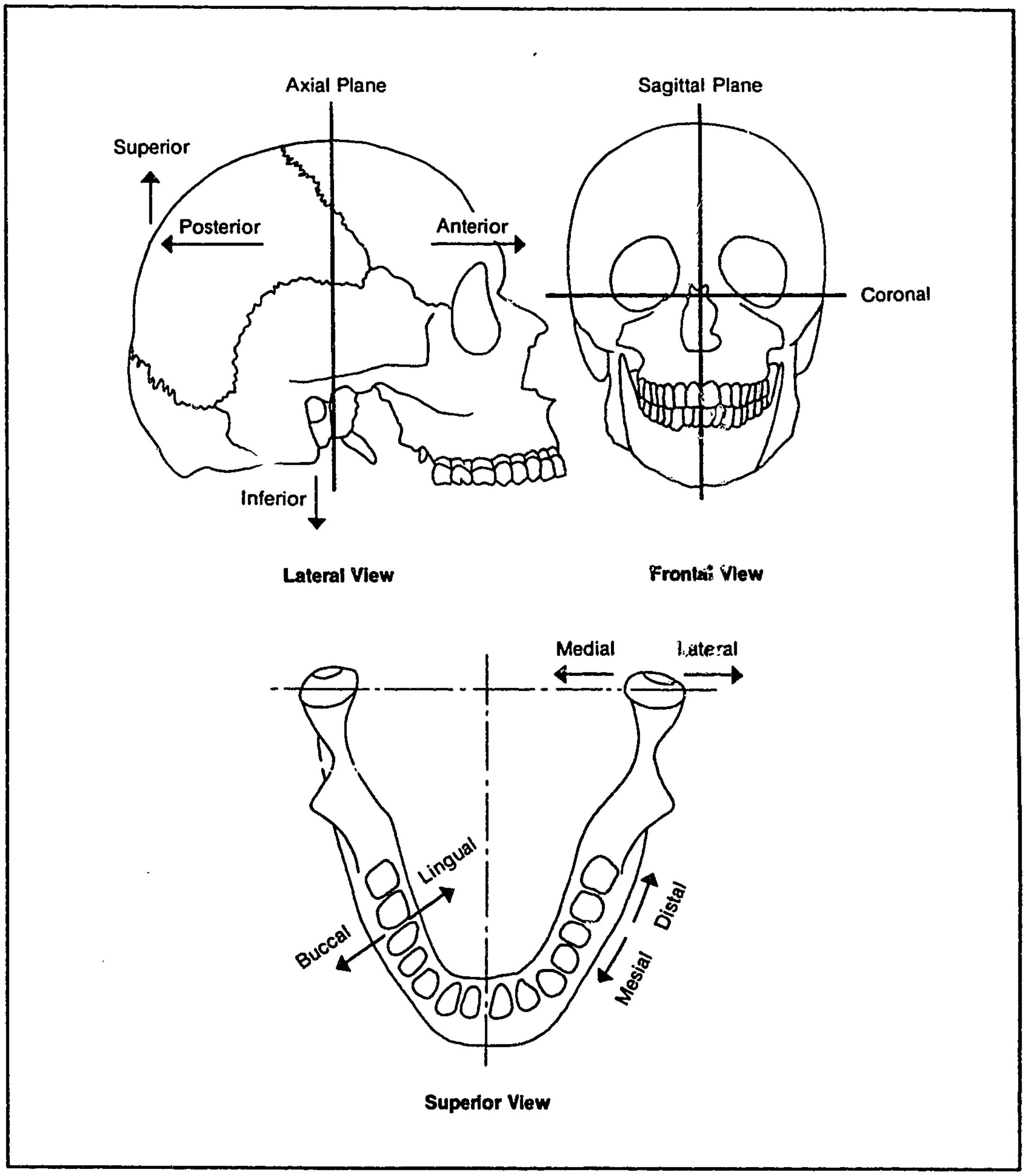



**Figure #1.1**
Anatomical reference planes and directions

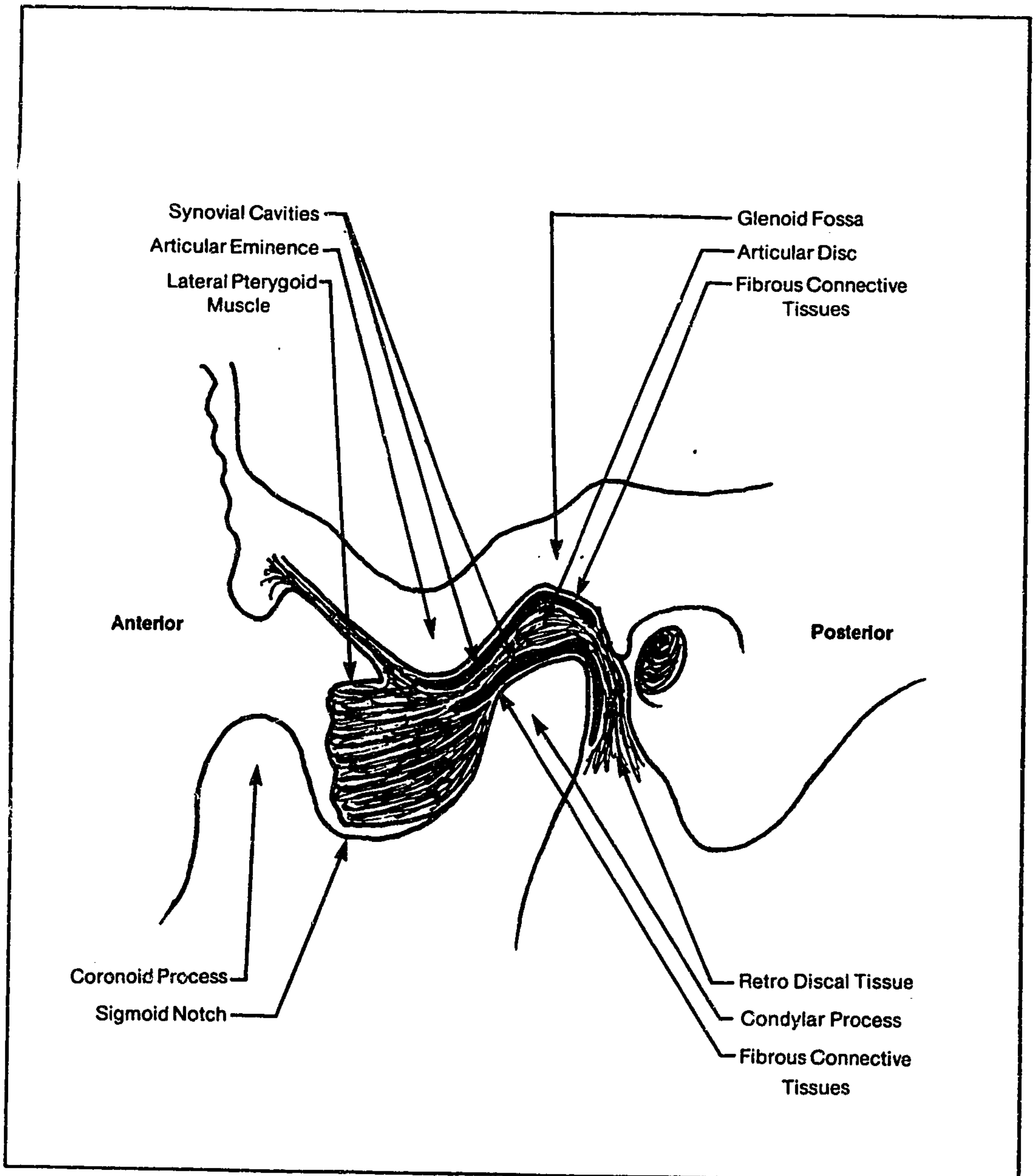



**Figure #1.2**
Lateral view of the temporomandibular joint

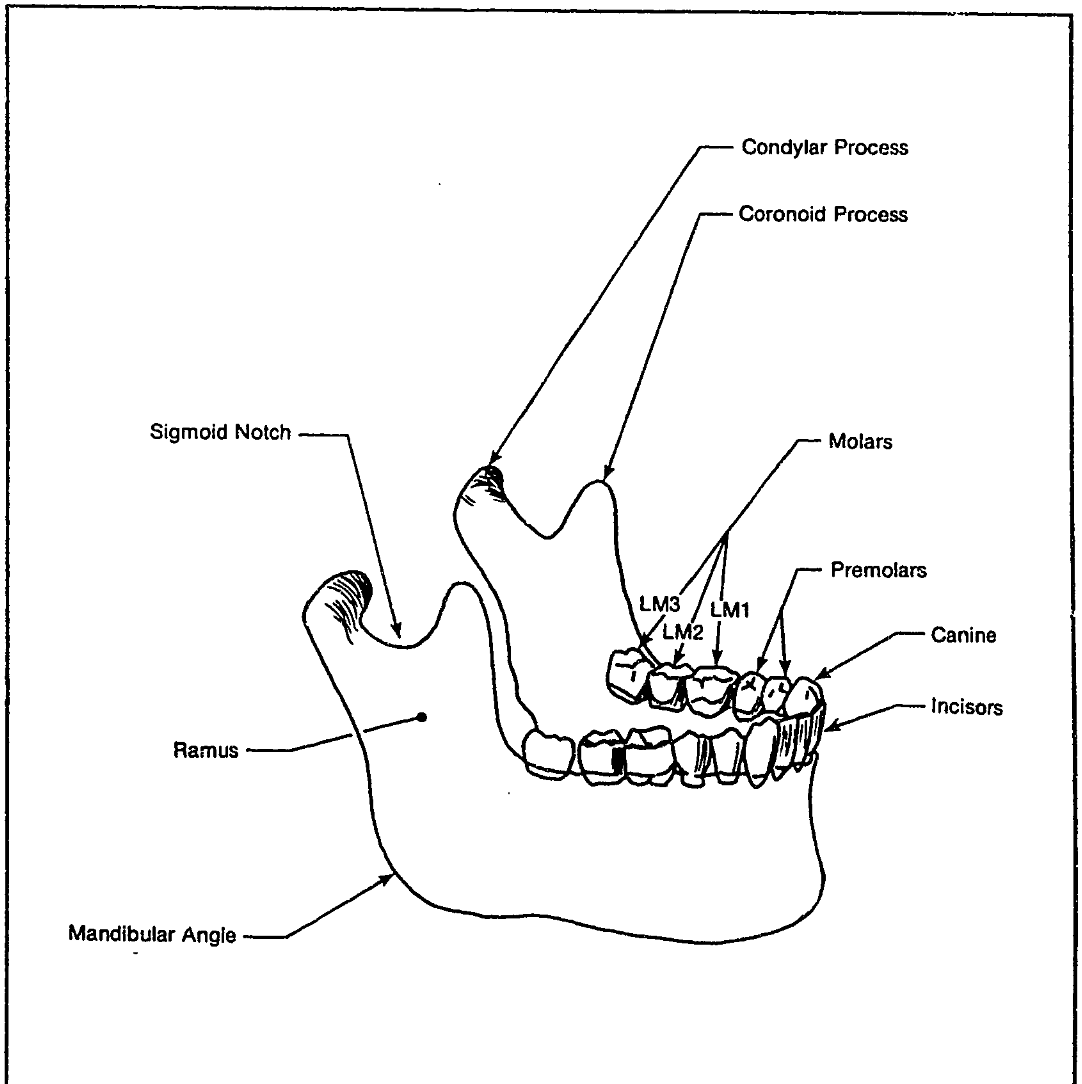



**Figure #1.3**
The mandible

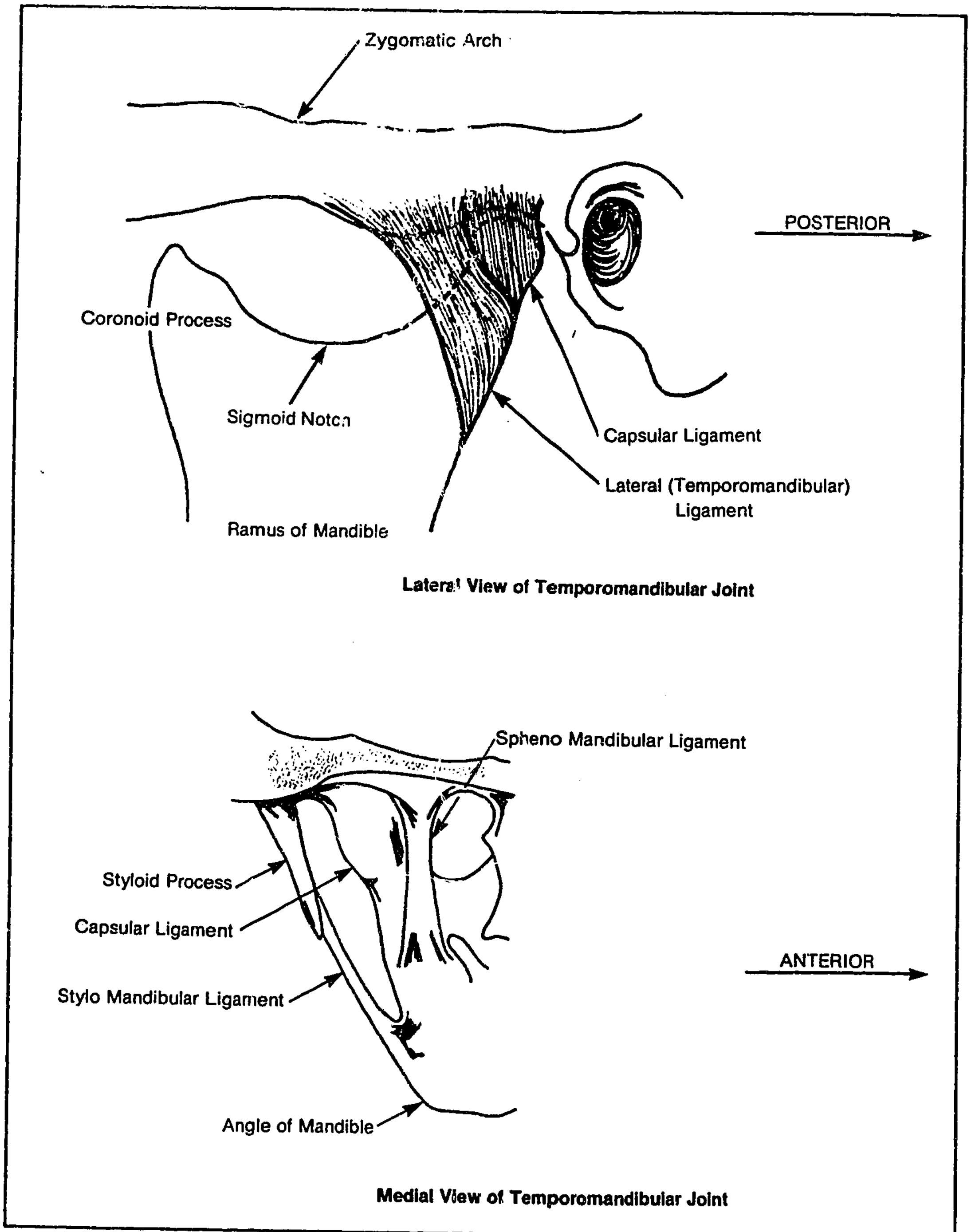



**Figure #1.4**
Ligaments

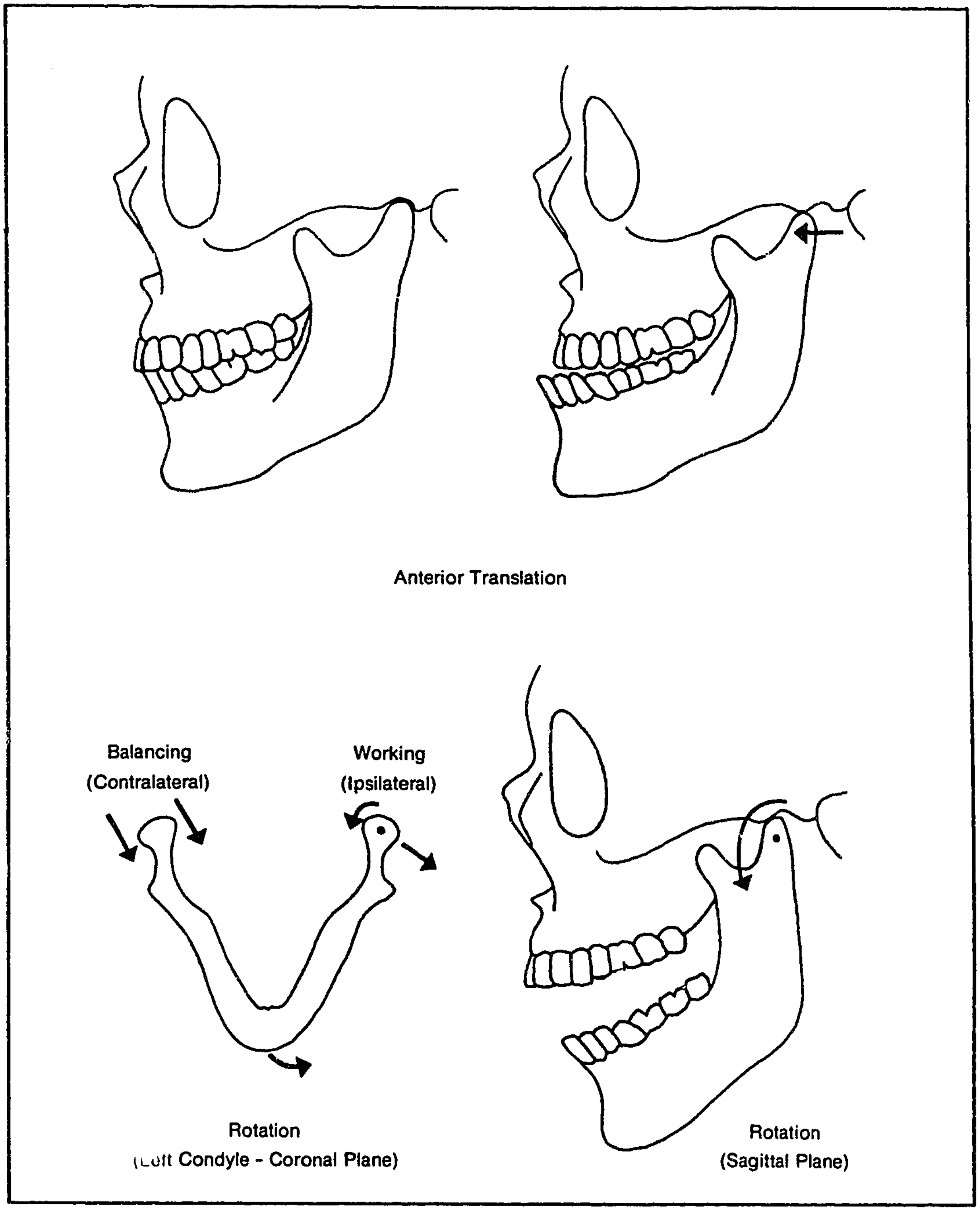



**Figure #1.5**
Condylar movements

### 1.2.2 Movements

The movements of the mandible can, in general, be categorized into two divisions: free movements and masticatory movements. Free movements include opening, closing, protrusion, retrusion, and lateral movements while masticatory movements are those which involve forceful occlusal contact, such as incisal biting and grinding movements (mastication). In the present study, only the forces developed in mastication are examined.

Mastication can be subdivided into three basic movements: opening, closing, and power strokes (Figure #1.6). In the opening stroke, the molar teeth on the occluding side (working or ipsilateral side) of the mandible are moved laterally and inferiorly. To accomplish this the ipsilateral condyle is rotated and shifted laterally while the contralateral (balancing side) condyle is translated anteroinferiorly and medially onto the anterior slope of the articular eminence. The closing stroke follows, in which the occluding molar teeth are moved superiorly and medially. To facilitate this movement, the ipsilateral condyle rotates in its fossa while the contralateral condyle is translated posterosuperiorly and laterally back towards the depth of the fossa. The power stroke then occurs, in which the occlusal contact takes place with a relatively large force, crushing and grinding the bolus.

### 1.2.3 Muscles

The muscles involved in the power stroke of mastication are the deep and superficial masseter, the temporalis, the medial pterygoid, and the inferior head of the lateral pterygoid. The temporalis muscle is fan-shaped and originates from the lateral surface of the temporal bone (Figure #1.7). The muscle fibres converge and pass between the lateral surface of the skull and the root of the zygoma, inserting primarily into the medial side of the coronoid process along its anterior, superior, and posterior borders.

The deep and superficial masseter muscles both originate at the anterior portion of the zygomatic arch and insert on the lateral external surface of the mandibular ramus (Figure #1.8). The superficial masseter muscle is lateral to the deep masseter muscle, and originates anterior to the deep masseter muscle.

The medial pterygoid muscle originates from the pterygoid plate of the sphenoid bone and inserts on the medial side of the mandibular angle and ramus, rising superiorly to the mandibular foramen (Figure #1.9). This muscle is the medial side counterpart of the masseter muscles and its direction in the sagittal plane is somewhat between the deep and superficial masseter muscles.

The lateral pterygoid is composed of two heads. The larger inferior head has its origin at the lateral pterygoid plate, while the superior head originates from the greater sphenoid wing. The superior head attaches to the anterior portion of the articular disc and condyle, and the inferior head inserts into the medial and anterior surfaces of the neck and condylar process (Figure #1.10). During the power stroke, the inferior head exerts a stabilizing force, helping to hold the condyle against the articular eminence [15].

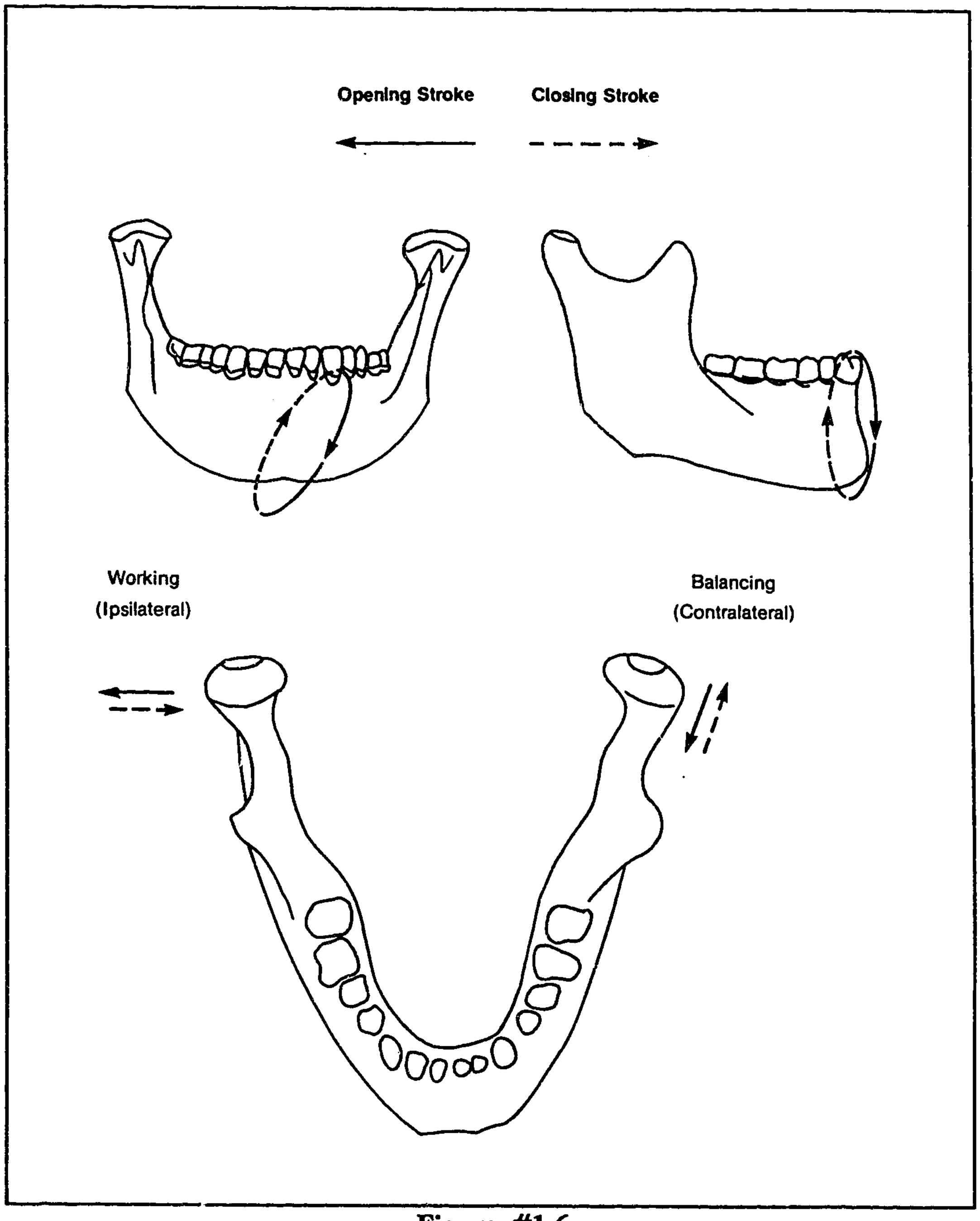



**Figure #1.6**
Opening and closing strokes of mastication

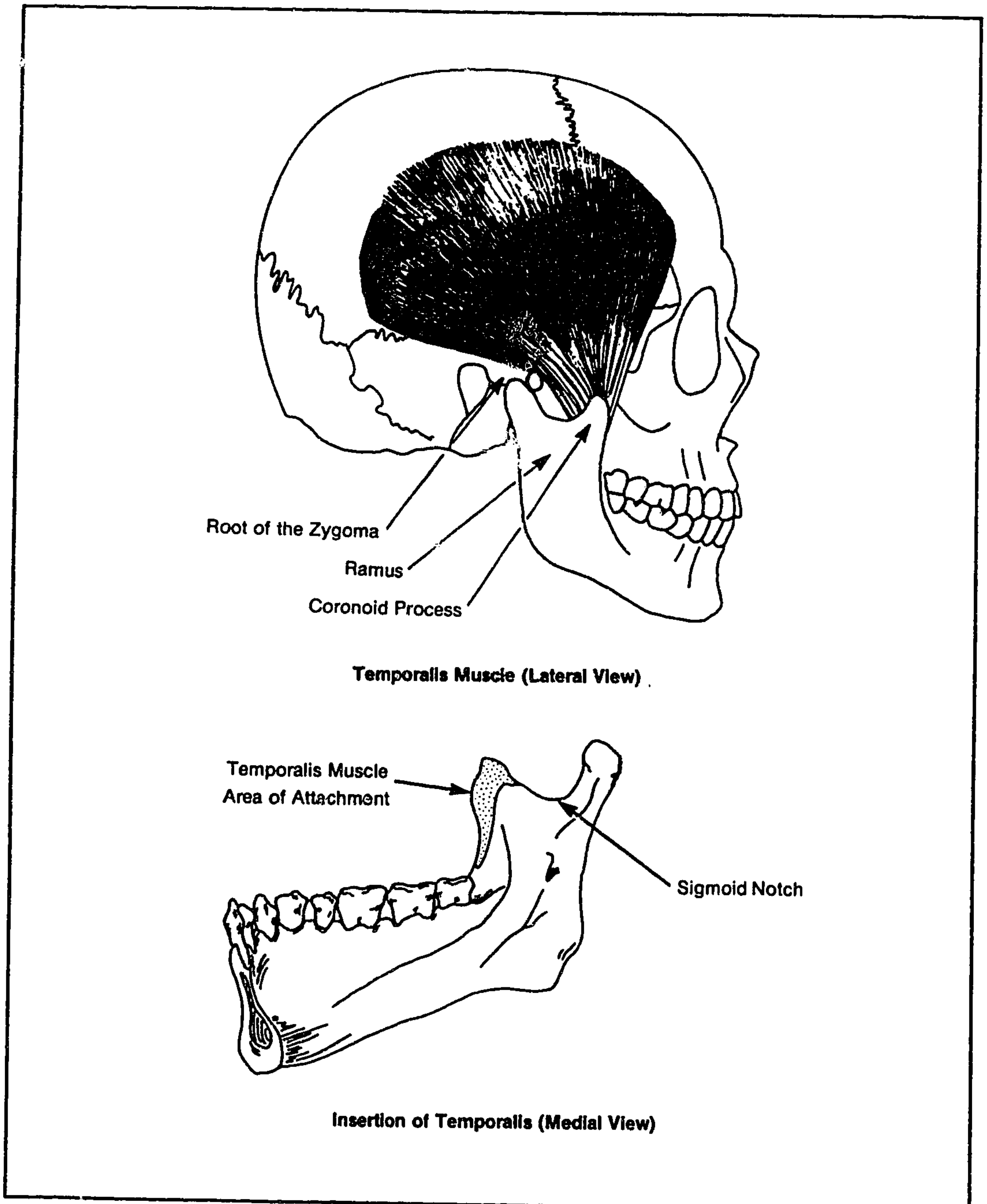



**Figure #1.7**
Temporalis muscle

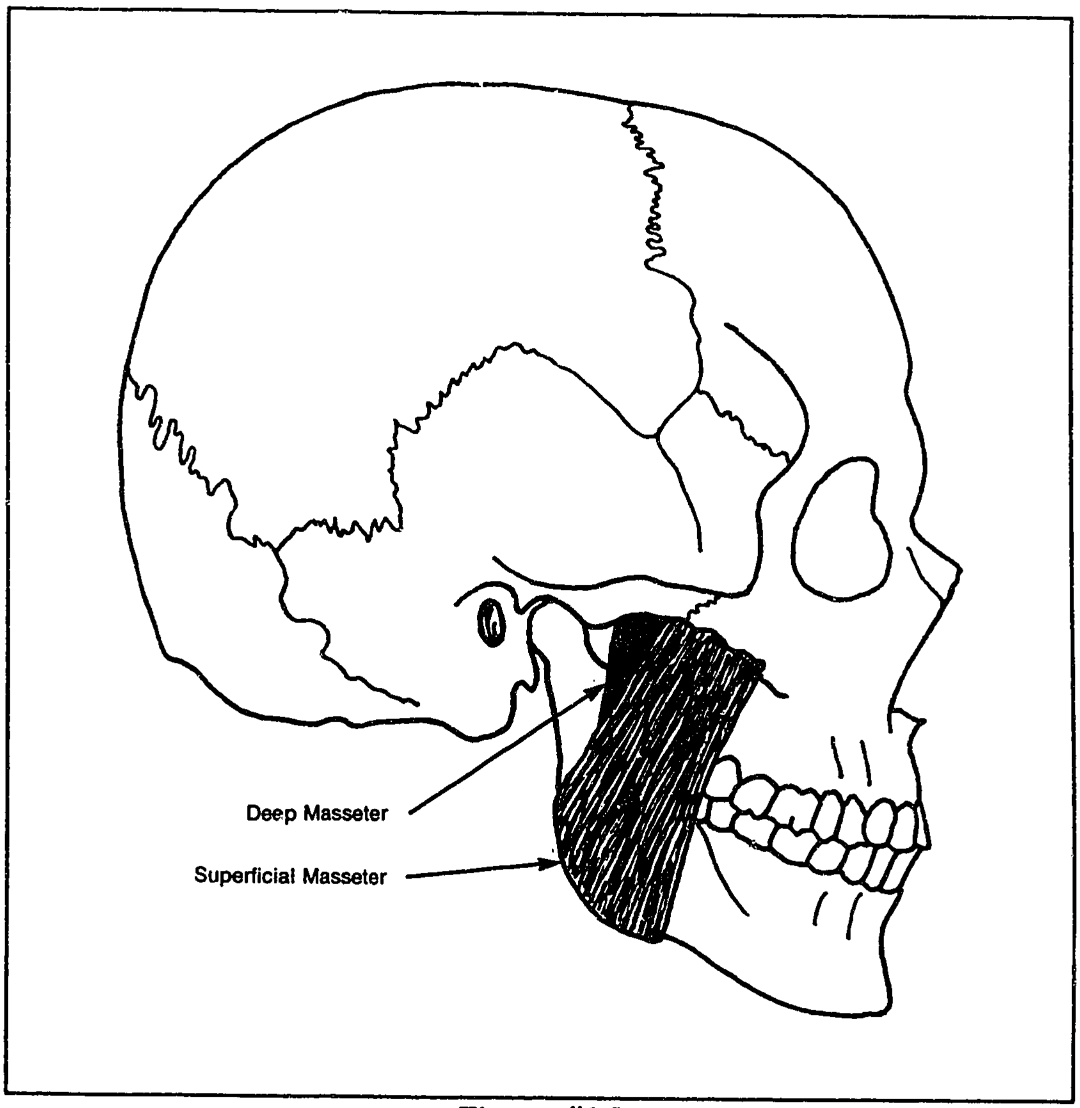
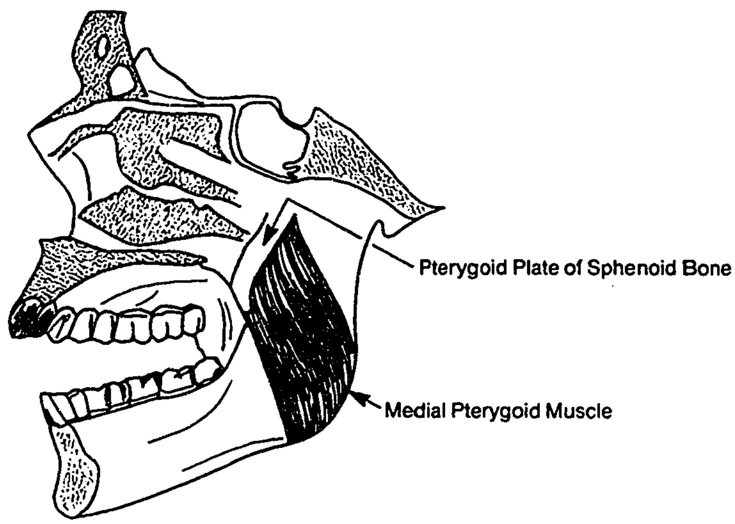
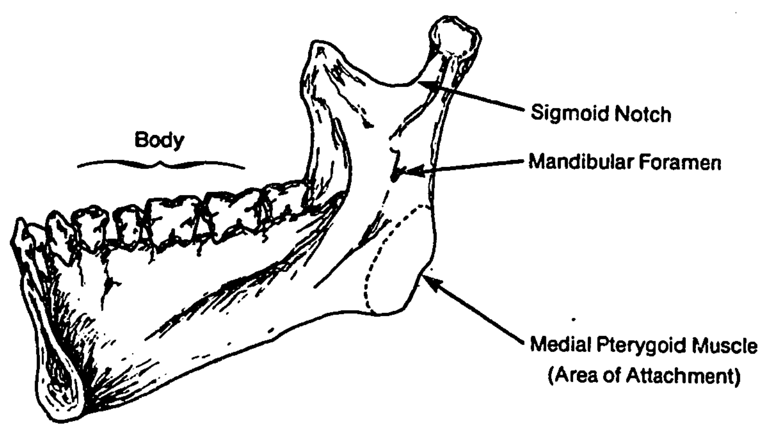



**Figure #1.8**
Masseter muscle




**Medial Pterygoid Muscle - (Medial View)**




**Medial Pterygoid Muscle Insertion - (Medial View)**

**Figure #1.9**
Medial pterygoid muscle

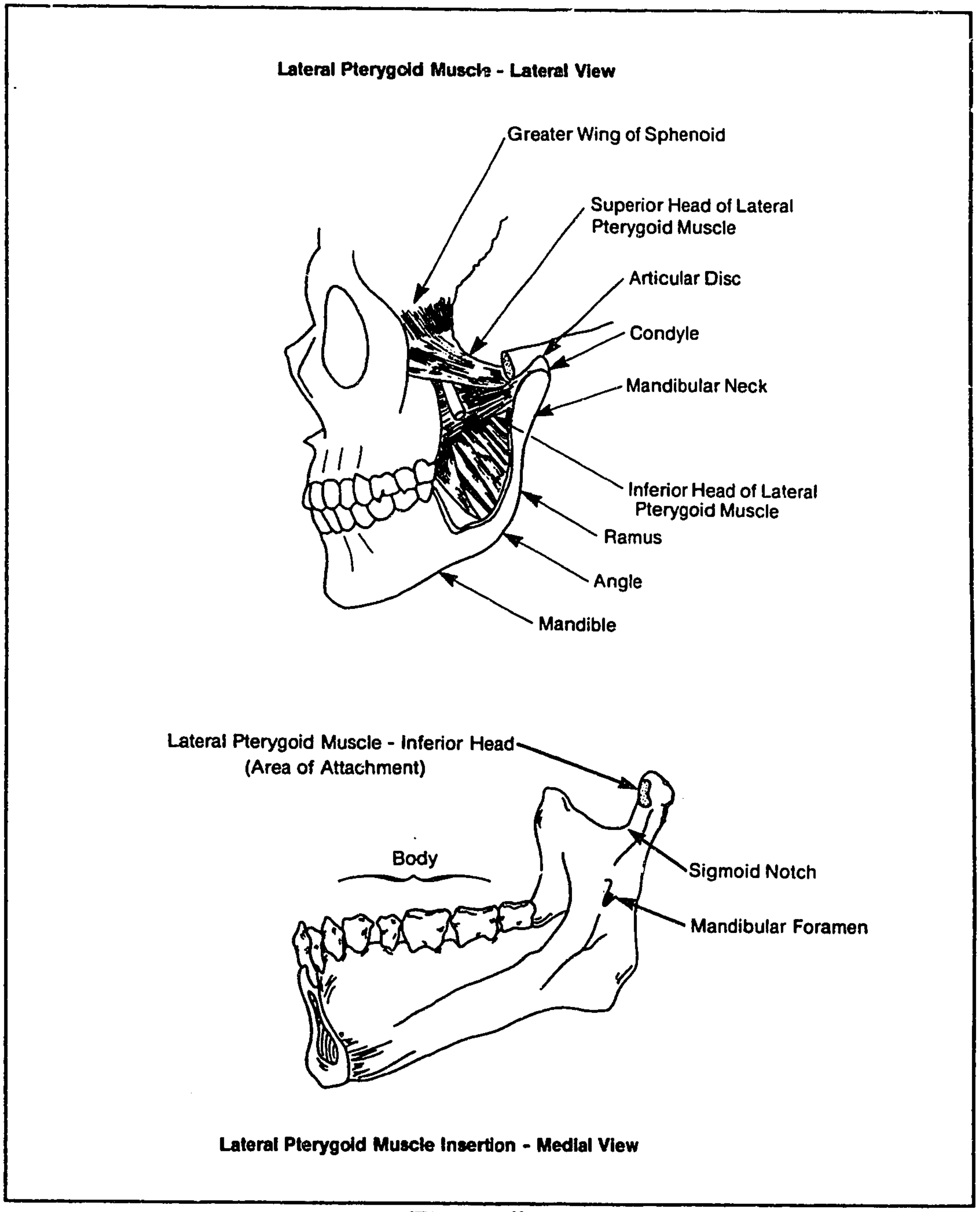



**Figure #1.10**
Lateral pterygoid muscle

### 1.2.4 Angle's Classification

To demonstrate a consistent distinction in the skeletal morphologies it was necessary to adopt a classification system. One terminology to effectively differentiate anterior-posterior dental relationship was proposed by Edward H. Angle in the 1890's. While his system is only unidimensional, it has provided dental personnel with a means of assessing the interdental relationship between the maxilla and the mandible. The key to Angle's classification is the maxillary first permanent molar which, according to him, was rarely misplaced. With this tooth as the benchmark, various combinations of lower teeth interdigitation are possible. In a Class I occlusion, considered a normal occlusion, the mesial lingual cusp of the maxillary first permanent molar is in line with the central fossa of the mandibular first permanent molar (see Figure #1.11). A Class II molar relation has the mandibular first permanent molar more than half a tooth width distal to the Class I relation (see Figure #1.12). Finally a Class III molar relationship has mandibular first permanent molar more than one half tooth width mesial to the Class I relationship (see Figure #1.13).

While Angle's classification system was directed only at the dentition, later investigations expanded his concept to describe the skeletal relationship of the maxilla and mandible to the anterior cranial base as well as each other [28]. The anterior cranial base is defined cephalometrically as a line joining a highly stable, reproducible point, sella (S), and a plane from sella to the intersection of the frontal and nasal bone, nasion (N) (refer to Figure #1.11). These structures mature very early in life, and while some growth changes occur, they are along the same plane with exceedingly small, if any, vertical changes.

The most retruded point of the inner curvature of the anterior maxilla and mandible are described as 'A' and 'B' points respectively. Therefore the angle SNA relates the anterior-posterior position of the maxilla to the anterior cranial base. Similarily the angle SNB relates the anterior-posterior position of the mandible to the anterior cranial base. The derived angle ANB (SNA minus SNB) is used to determine the relation of the maxilla to the mandible. A Class I skeletal pattern has an ANB angle between 0° and 5°, while a Class II skeletal pattern has an angle greater than +5°. A Class III skeletal pattern has an ANB angle less than 0°. The limits of each of these classes are arguable, so whenever possible the study will attempt to show the spectrum of ANB values individually rather than in groups.

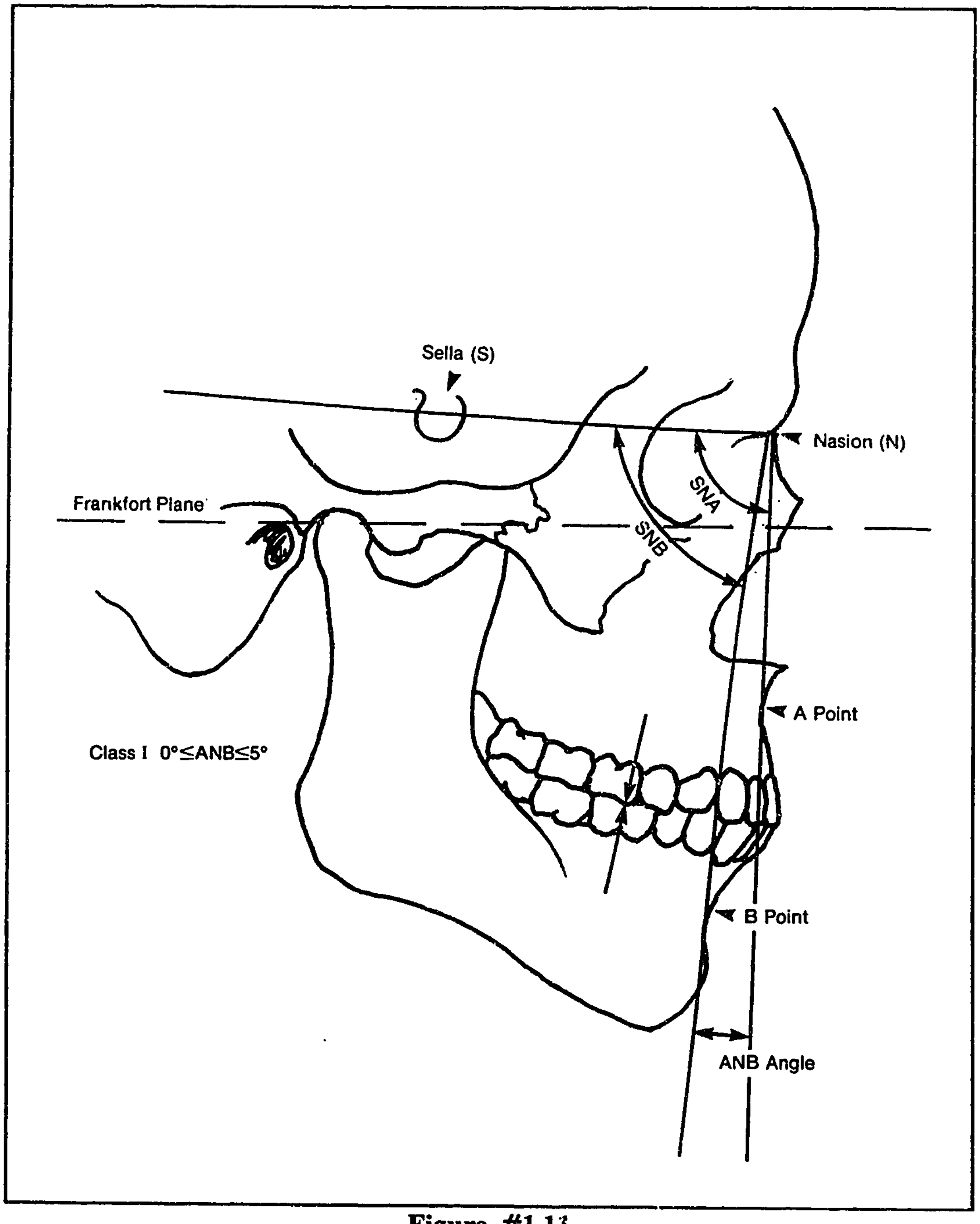



**Figure #1.11**
Lateral view of Angle's Class I skeletal

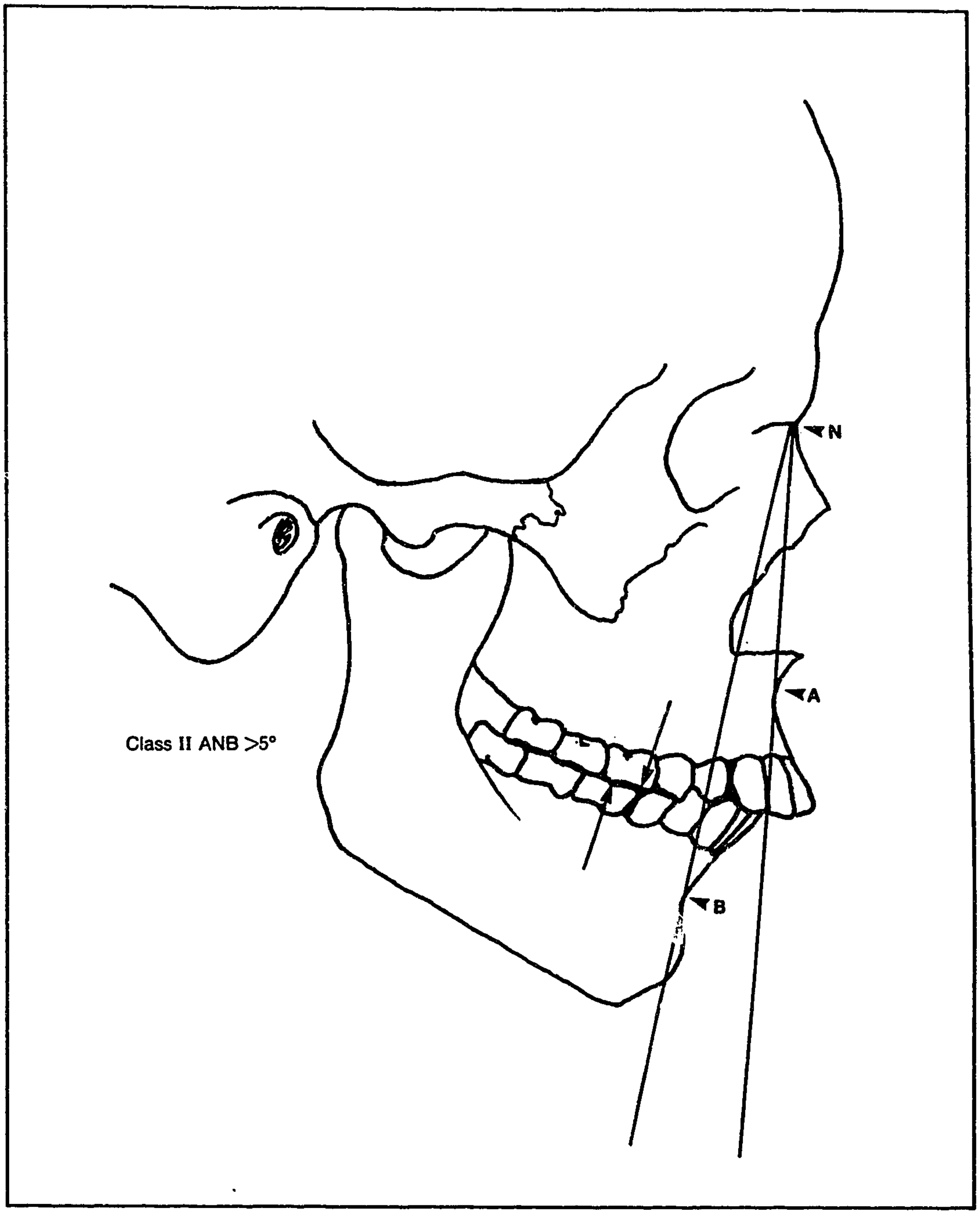



**Figure #1.12**
Lateral view of Angle's Class II skeletal

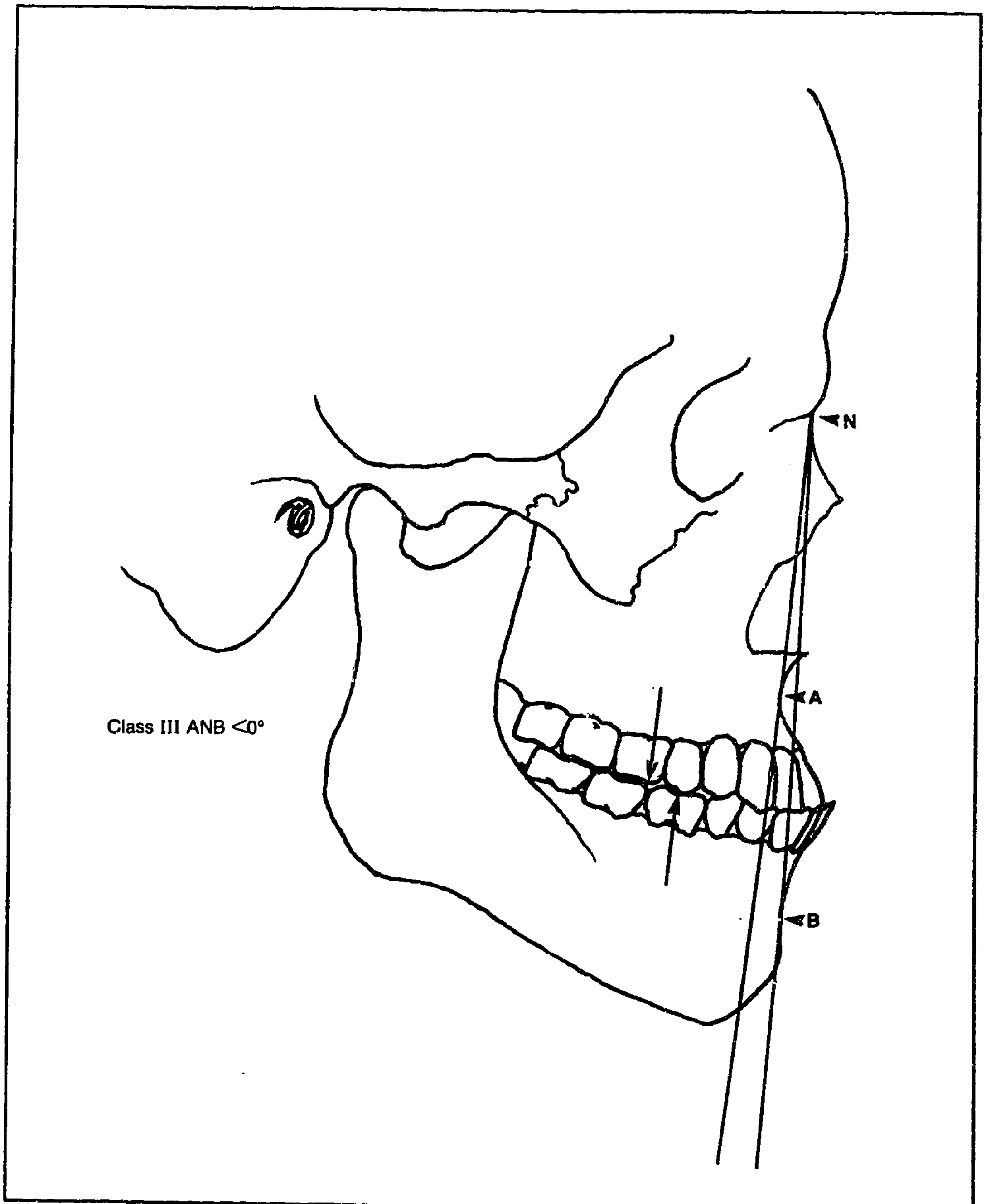



**Figure #1.13**
Lateral view of Angle's Class III skeletal

## 1.3 Literature Review

The temporomandibular joint is unlike any other joint in the human body, and has been the topic of much discussion. It's function and materials differ a great deal from those of other joints in the body. It is clearly evident that other joints (such as the knee or elbow) in the human body are loaded during function, but is the human TMJ loaded during function? For years there has been considerable debate as to whether the mandible functions as a lever (where the condyles are loaded) or a link (where the condyles are unloaded) [7, 12].

The proponents of the link theory [30, 38, 42] had what at first was a seemingly convincing argument that, based primarily on the fact that the roof of the glenoid fossa is extremely thin, the TMJ was poorly suited to bearing a load. Evidence presented by Scott [32] and Steinhardt [37] also described that, based on observation of other load bearing joints, the absence of load bearing tissue that appears in other joints suggested that the TMJ should not be capable of bearing a significant load. An overwhelming amount of recent research by supporters of the lever theory has shown that the TMJ is substantially loaded during function [1, 2, 3, 7, 9, 10, 12, 20, 21, 23, 26, 33, 34, 39, 40], and that the load is placed upon the articular eminence rather than the roof of the glenoid fossa [12]. Barbenel [3] and Smith [33] were able to show that the TMJ could be unloaded during occlusion in a few cases. They were also quick to dismiss this conclusion since the condition was only possible under highly unrealistic circumstances.

Current researchers of the TMJ have employed sophisticated three dimensional numerical techniques that compute the joint loads based on skeletal information. Due to the inability to accurately know the values of all the variables involved, different numerical strategies are employed to solve for the unknowns. The three most prominent strategies are: solution by minimizing the levels of the joint load used by Smith [33]; solution by minimizing the amount of energy expended by the musculature used by Osborn [23]; and finally using the pragmatic approach of providing estimates of all the applied muscle magnitudes. The last of these approaches was employed by Hay [10] and is used in the current study.

In brief there are three groups of information necessary to solve for the unknown joint loads in a numerical model (these are described in detail in Chapter 3). First, the spatial information about a skull is required (location of muscle origins and inserts, etc.); this is common to all three approaches. Second is the level of information on the occlusal load. All three strategies specify the position and direction, and all but the third specify the occlusal load magnitude as well. The final, and most controversial, information is the relative levels of muscle activation used to produce the occlusal load.

Rather than to try and specify the muscle magnitudes, Smith's and Osborn's models employ linear programming techniques based upon two different assumptions. Smith's model assumed that the masticatory system tries to minimize the magnitudes of the TMJ loads, and searches the possible combinations of muscle magnitudes to find the minimizing combination. Osborn's model is based upon the assumption that the masticatory system tries to be most efficient, that is to produce the required occlusal load with the minimum amount of muscle energy input. To this end Osborn's model also searches through all the possible combinations of muscle levels capable of producing the specified occlusal load, and chooses the combination with the minimum total amount of muscle activation.

Although each of these approaches has desirable and convincing attributes, in the author's opinion, the pragmatic approach is best suited to the current task of evaluating joint loading patterns in varying morphologies. By constructing an evaluation tool that does not seek to minimize or maximize variables, but rather evaluates each separate skull under the same conditions, it should more readily shows the strengths and weaknesses of the various morphologies with respect to each other. Comparative evaluation of the results of each of the different models is difficult due to the usage of different muscle groups within each of the models and the form their results take. If all three strategies were to employ standard spatial and muscle values, it may be revealed that all three approaches may lead to the same conclusions.

The task of defining the individual levels of muscle force during unilateral occlusion is a difficult one because these forces cannot be directly measured. Recently electromyography (EMG) has provided researchers with a better understanding of muscle activity. By inserting electrodes into the muscle fibers, or on the surface of the skin directly above it, the electrical activity of the muscle can be monitored. The greater the electrical activity of the muscle, the greater the force the muscle is exerting [14]. The masseter and temporalis muscles have been studied in much detail, but due to their inaccessibility studies on the pterygoid muscles are few. While EMG is inaccurate at predicting the exact force a muscle is exerting [27], it is a good predictor of when a muscle is active and can give some measure of the degree of activity.

A combination of EMG information [5, 18, 19, 24] and knowledge of muscle cross-sectional areas [8] was used by Hay [10] to develop his so-called Type II muscle magnitudes, and it is essentially these values are used in the current study. By applying these muscle levels consistently in the current study, an objective tool to evaluate the TMJ loading differences in varying morphologies is achieved.

Previously studies of TMJ loading have been primarily concerned with the magnitude of the resultant load within the TMJ, based on the premise that abnormally higher levels of joint load, levels that exceed the articular tissues adaptive potential, would be at greater risk of TMJ dysfunction than those with normal levels.

Substantial research has been done in the area of joint remodelling and degeneration. There are several ways the TMJ remodels itself in response to increased joint loads: the shape of the condyle changes through bone decomposition and resorption in order to distribute the load over a larger area to reduce the stress levels; the thickness of the fibrous connective tissues increase; and the qualities of these tissues improve (resiliency is increased) [16].

As the functional loads exceed the adaptive capabilities of the articular osseous and soft tissues, degeneration begins to occur. The thickness of the fibrous connective tissues decrease, their properties deteriorate, and a great deal of bone resorption occurs [22]. Failure to adapt to excessive loads has been shown to lead to degenerative joint disease (osteoarthritis) [11, 13, 25].

If the human masticatory system can be thought of as a mechanical system, an engineering evaluation of the all the different stages of degenerative joint disease indicates that it is unlikely that they are all as a result of one individual cause. In 1974 Barbenel [3] noted the importance of the direction of the resultant load within the TMJ with relation to the load bearing surfaces in his numerical investigation of TMJ loading. Recent works by Blaustein and Scapino [4,31] investigating the glycosaminoglycan content of articular discs removed from patients with TMJ dysfunction indicated that the region where the retro discal tissues and the posterior band of the articular disc join had remodelled due to abnormal loading. Although they noted that these findings must be limited to their study due to the minimal sample size (9 discs), they provide some evidence that the condylar loads in these individuals may have been misdirected posterosuperiorly onto the posterior band of the articular disc.

It is postulated that an "ideal" TMJ loading relationship is one where the reaction load required at the condyle would be aligned perpendicularly to the load bearing surface of the corresponding articular eminence. It is believed that this situation of a pure compression load would be the healthiest for the loaded tissues interposed between these surfaces. Further it is expected that evidence will show significantly higher levels of posterosuperiorly directed shear loads, created as a result of load misdirection, in morphologies with higher levels of TMJ dysfunction.

## 1.4 A Guide to the Thesis

The chapters to follow explain the methods, numerical techniques, and finding of the current study. Chapter 2 discusses in detail the methods used to gather the spatial information for the seventy skulls used in the study. Chapter 3 explains the concepts and techniques of the current numerical model and how they compare to previous models. Chapter 4 is a presentation of the computed results and a discussion of their implications. Finally Chapter 5 discusses the conclusions to be drawn from the findings as well it addresses areas which merit further investigation.

# 2.0 Data Collection

## 2.1 Introduction

Seventy skulls representing a distribution of craniofacial morphological types were selected from the osteological collections at the University of the Pacific in San Francisco, California, U.S.A. and the University of Alberta. These skulls were digitized, photographed and radiographed to provide an accurate three dimensional spatially detailed information base for numerical modelling.

## 2.2 The Sample

In gathering the spectrum of morphologies necessary two separate skull collections were used. Seventy skulls were selected from the population of skulls contained in the Atkinson Cranio - Osteological Collection at the University of the Pacific in San Francisco, California and the educational collection at the University of Alberta. The total sample was comprised of thirty one Angle's skeletal Class I's, twenty six Angle's Class II's with the remaining thirteen as Angle's Class III's. Based on caucasian population norms there are approximately 65% Class I cases, 32% Class II cases and approximately 3% Class III cases.

The skulls were randomly selected from each collection until a sufficient number of each class was collected to allow for meaningful inter-class comparison. The additional criteria for selection were: intact dentition (missing no more than one posterior tooth); and vertical or transverse asymmetries that were not greater than normal limits.

## 2.3 Point Selection

A digitizing regime was developed for acquiring the spatial information from the skulls. The mathematical model developed (described in Chapter 3) required three dimensional coordinates for each muscle's origin and insertion as well as for each of the condyles. In order to properly locate and apply the occlusal loads, information on the occlusion was also required. Coordinate information about other landmarks was needed so that the spatial information could be referenced relative to accepted anatomical planes (i.e. Frankfort, Occlusal , Midsagittal, etc.). In total 112 points were selected on each skull; twenty six skeletal landmarks, fifty four points on the dentition, twenty four points describing the fourteen muscle vectors, and eight points defining the condyles and fossae. Of these 112 points only fifty five were required to supply the information needed for the numerical model. These points are described in detail below.

### 2.3.1 Muscle Origins and Inserts:

Temporalis Origin: The temporalis muscle was defined by the extent of the visible muscle scar. This area was then divided into anterior, middle and posterior sections as seen in Figure #2.1. The distance between the anterior and posterior extent of the muscle scar, parallel with the zygomatic arch, was used to make the distinctions. 1/6 of the total distance along the line located the **anterior** origin, 1/2 way for the **middle** origin, and the distal aspect of the scar for the **posterior** origin. All point were placed 1/6 of the antero-posterior distance toward the coronoid process.

Temporalis Insert: The insertion for all three sections of the temporalis muscle was recorded on the medial aspect of the coronoid process equidistant from the sides as well as between the tip and base (see Figure #2.2).

Masseter Origins: The muscle scar where the masseter muscle attaches to the zygomatic bone and zygomatic arch was first delineated then arbitrarily divided into an anterior and posterior component at the zygomatic temporal suture. The **superficial** masseter origin was recorded at the midpoint of the anterior component, with the **deep** masseter origin recorded at the midpoint of the posterior aspect of the muscle scar (see Figure #2.3).

Masseter Inserts: The muscle scar on the lateral aspect of the angle of the mandible was first delineated and then divided into thirds in a superior-inferior direction. The **superficial** masseter insertion was recorded at the junction of the inferior and middle thirds and centered in the muscle scar in an anteroposterior direction. The **deep** belly of the masseter insertion was recorded at the junction of the middle and superior thirds and was also centered anteroposteriorly in the muscle scar (see Figure #2.4).

Lateral and Medial
Pterygoid Origins: The origins of the lateral and medial pterygoid muscles were recorded at the center of the sphenoid bone. The medial pterygoid muscle origin was recorded on the medial aspect of this bone, and the lateral pterygoid muscle origin was recorded on the lateral aspect. No distinction was made in the origins of the two bellies of the lateral pterygoid muscle (see Figure #2.5).

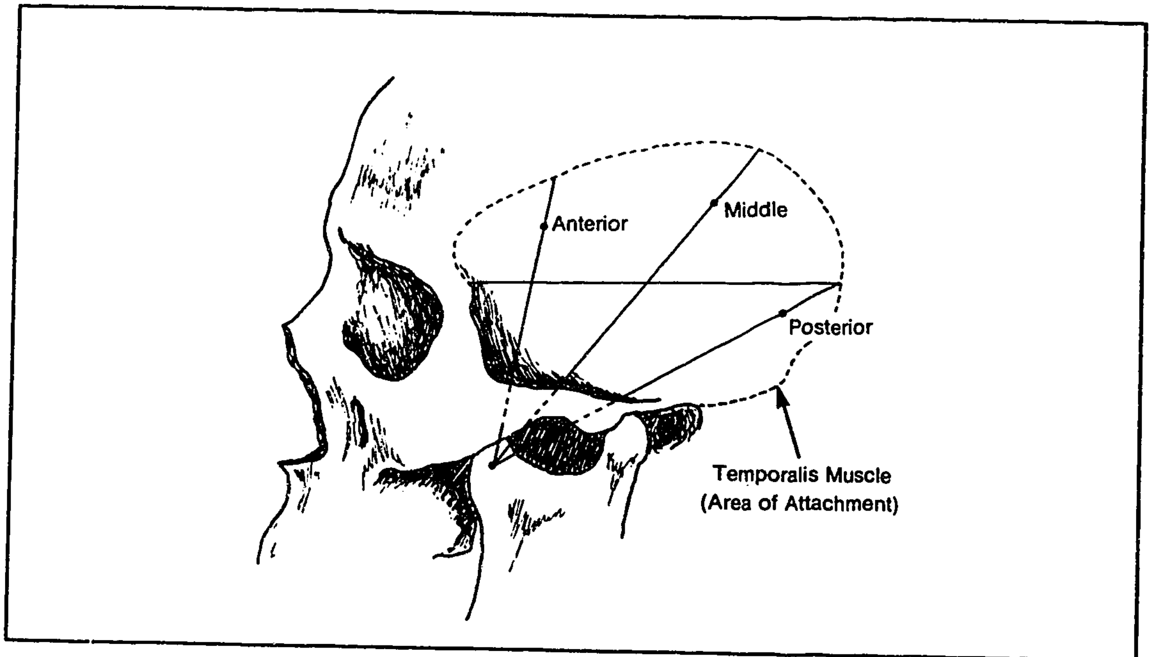



**Figure #2.1**
Temporalis muscle origins

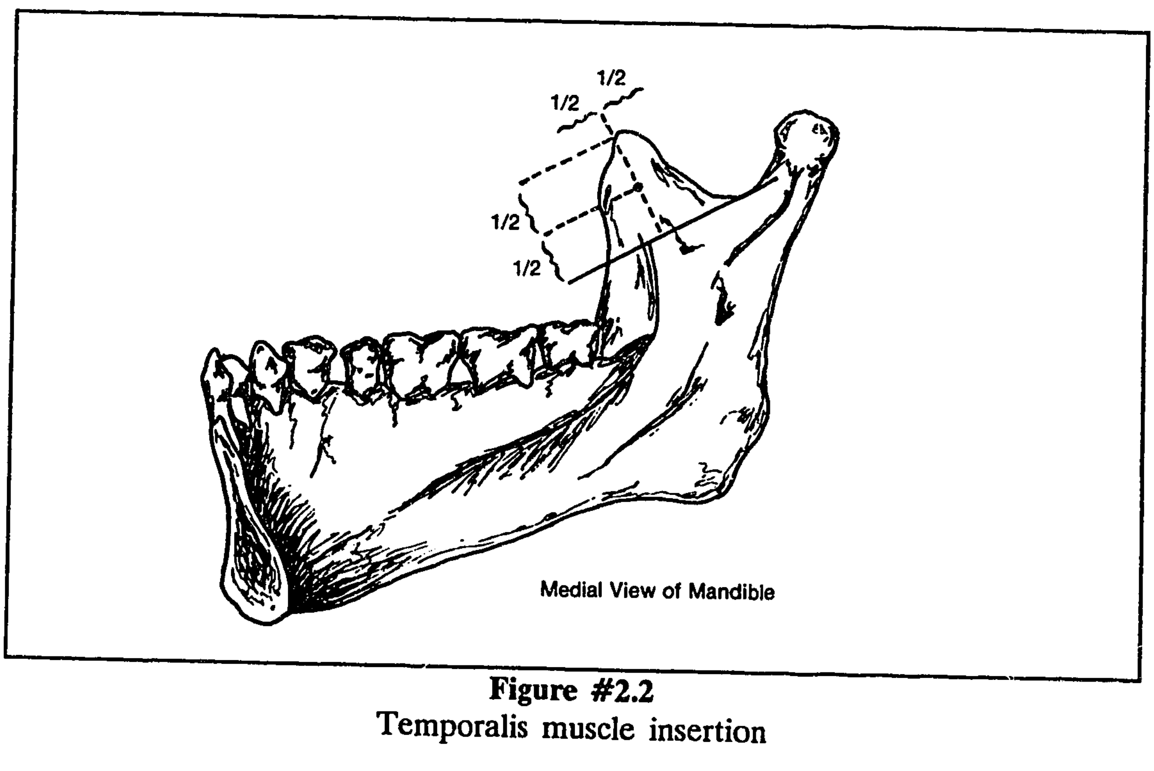



**Figure #2.2**
Temporalis muscle insertion

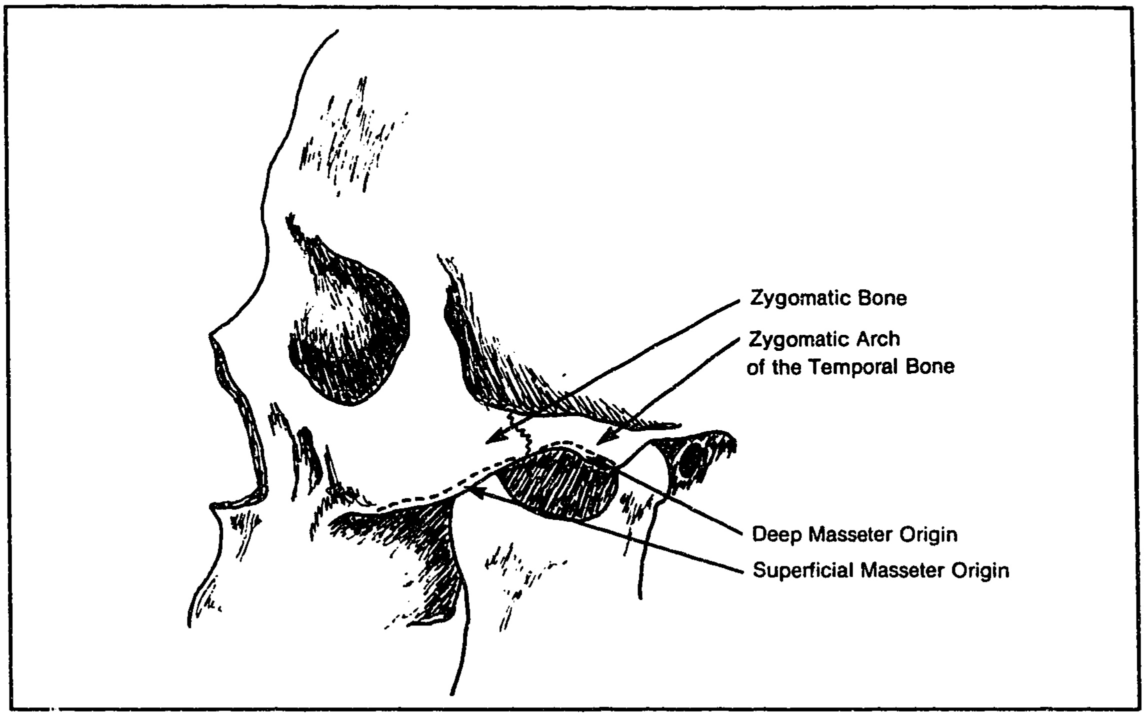



**Figure #2.3**
Masseter muscle origins

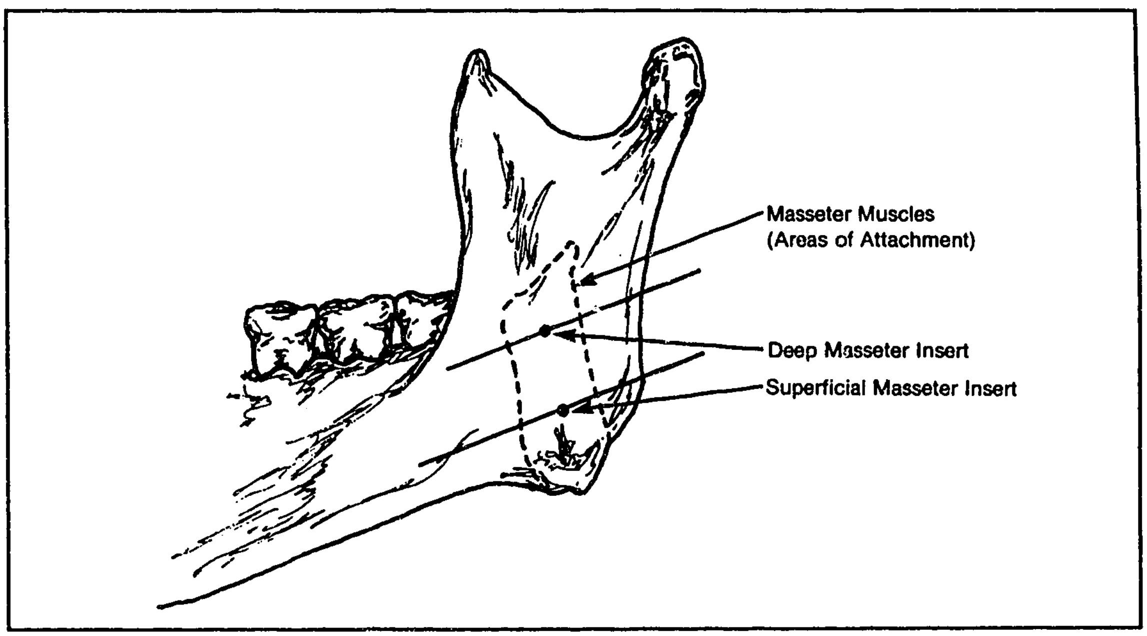



**Figure #2.4**
Masseter muscle insertions

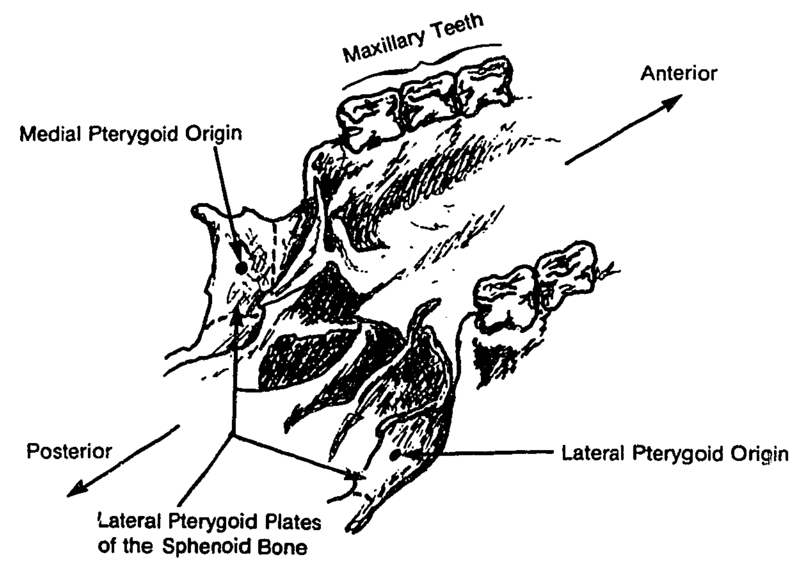



**Figure #2.5**
Lateral and medial pterygoid muscle origins

Lateral Pterygoid Insertion: The lateral pterygoid insertion was recorded on the anterior aspect of the head of the condyle, slightly medial to the centerline as depicted in Figure #2.6.

Medial Pterygoid Insertion: The medial pterygoid insertion was recorded in the center of the muscle scar on the medial surface of the mandible (see Figure #2.7).

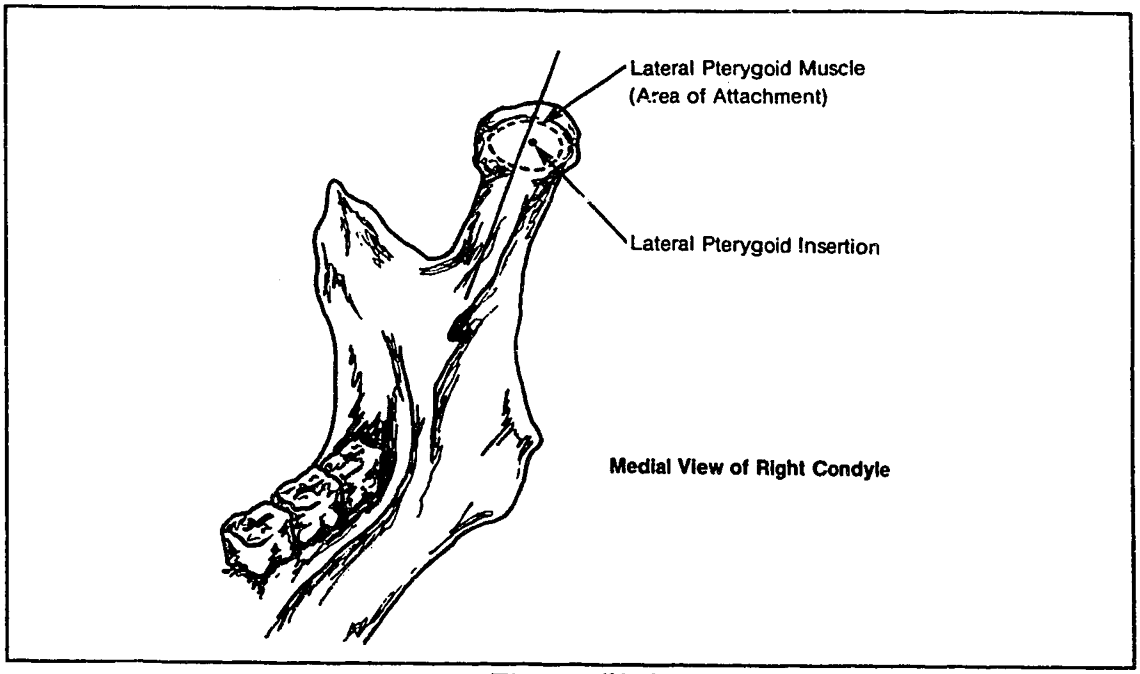



**Figure #2.6**
Lateral pterygoid muscle insertion

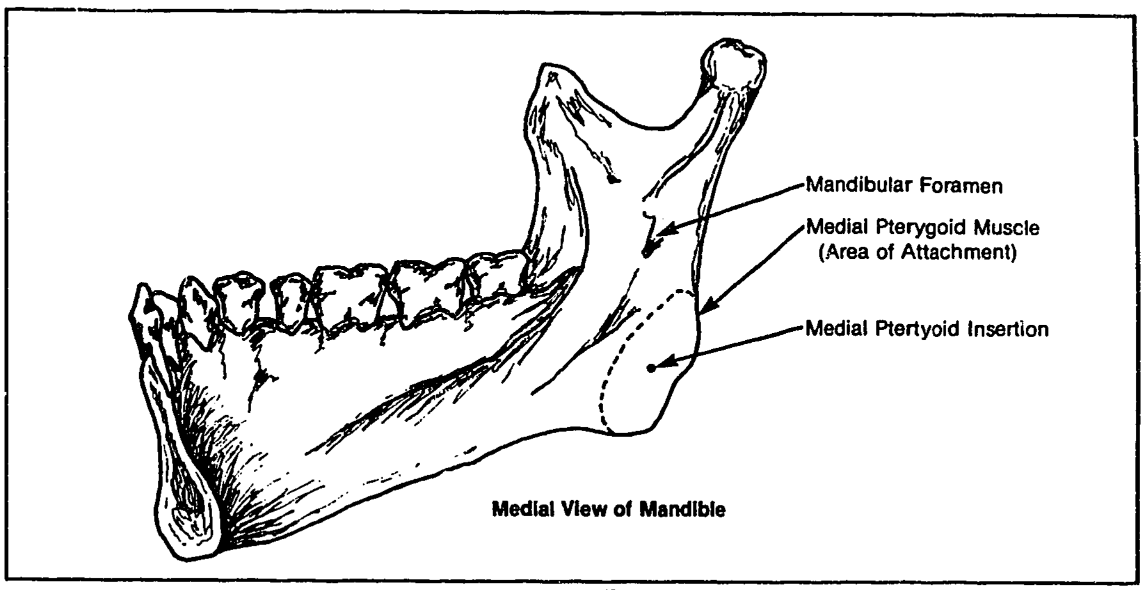



**Figure #2.7**
Medial pterygoid muscle insertion

## 2.3.2 Dentition

A total of five points were recorded on each of the three molars and two premolars on each side of the maxillary arch. The objective of taking these points was two fold: to establish the shape of each occlusal surface, and, to determine the relationship of these surfaces to themselves and other structures. It is important to note that data points recorded on the teeth were always recorded in positions where they best represented any attrition facets on the occlusal surfaces. In cases where no attrition was present the central fossa point was recorded in the position vertically which best represented the angulation of the functioning aspect of the cusps (see Figure #2.8). If a tooth was absent from the arch or was not in occlusion with the mandibular teeth, an estimate of its ideal occlusal position was made and its absence was noted.

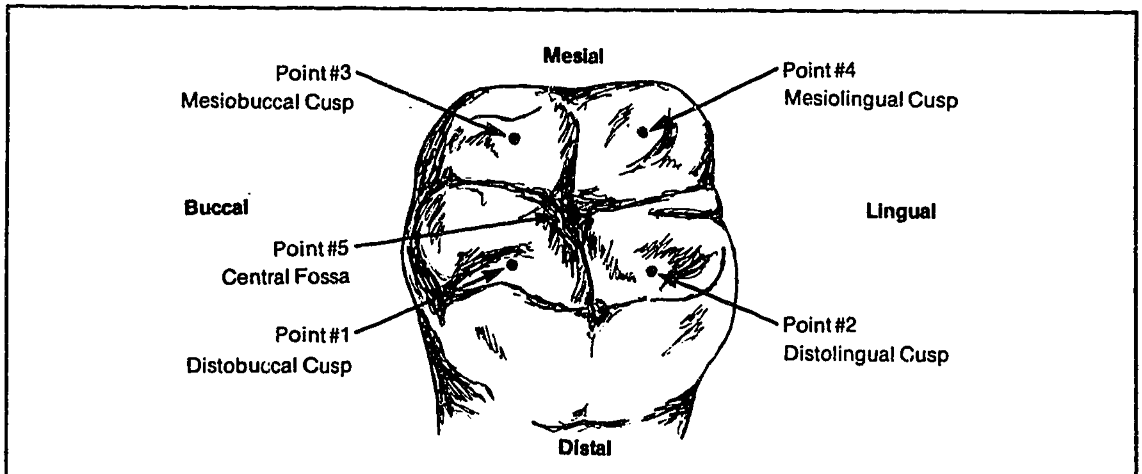



**Figure #2.8**
Digitized Points on the Left First Molar

Molars: The first four points were used to record the cusp tips with the fifth point recording the central fossa (Figure #2.8). On the maxillary first molar a sixth point was recorded at the mesial marginal ridge. This position was considered roughly equivalent to the distobuccal cusp of the mandibular first molar, which was required in order to establish the occlusal plane routinely used in orthodontic measurements. The distolingual cusp of maxillary molars is often not present, or too small to contribute to the occlusal surface of the tooth. On teeth where this was the case two points were used to demarcate the mesiolingual cusp.

Premolars: The first two points recorded were to identify the buccal and lingual cusp tips, and the third and fourth points recording the distal and mesial marginal ridges. A fifth point was taken at the mid point of the tooth in the development groove.

### 2.3.3 Glenoid Fossae

The anterior aspect of each glenoid fossa was recorded using four data points. The specific positions recorded were selected to best represent the discernable articulating surfaces (see Figure #2.9).

### 2.3.4 Landmarks

The following points were taken to precisely record the same positions as seen on a lateral cephalograph X-ray taken of the same skull (see Figure #2.10).

A point: Located at the location of innermost curvature from the maxillary nasal spine to the crest of the maxillary alveolar. Used in calculating SNA and ANB angles.

B point: Located at the location of innermost curvature on the anterior bony curvature of the mandible. Used in calculating SNB and ANB angles.

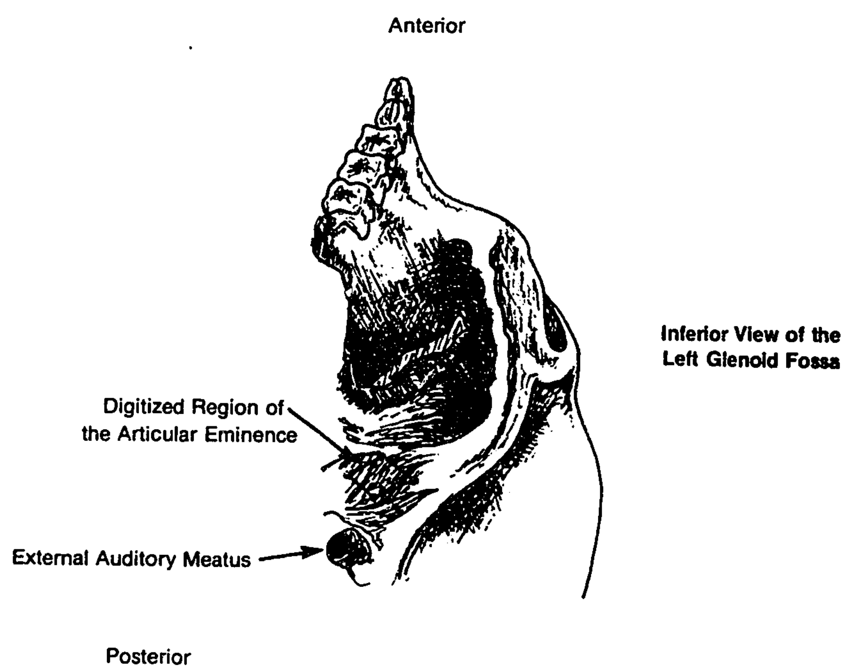



**Figure #2.9**
Digitized Region of the Glenoid Fossa

Pogonion: (Left/Central/Right) The most anterior portion of the bony contour of the chin.

Menton: (Left/Central/Right) The most inferior point of the mandibular symphysis.

Gonion: (Left/Right) A point representing the middle of curvature at the junction of the ramus and the body of the mandible at its posterior inferior aspect.

Anterior Nasal Spine: The anterior tip of the sharp bony process of the maxilla at the lower margin of the anterior nasal opening.

Posterior Nasal Spine: The posterior spine of the palatine bone constituting the hard palate.

Nasion: The junction of the frontal and nasal bones at the most posterior location along the curvature of the bridge of the nose. Used in calculating the SNA, SNB and ANB angles.

Superior Border of the Auditory Meatus: (Left/Right) Recorded at the most superior-lateral border of the external auditory meatus. Used in defining the Frankfort plane.

Inferior Border of Orbit: (Left/Right) Recorded at the inferior anterior aspect of the bony margin of each orbit. Used in defining the Frankfort plane.

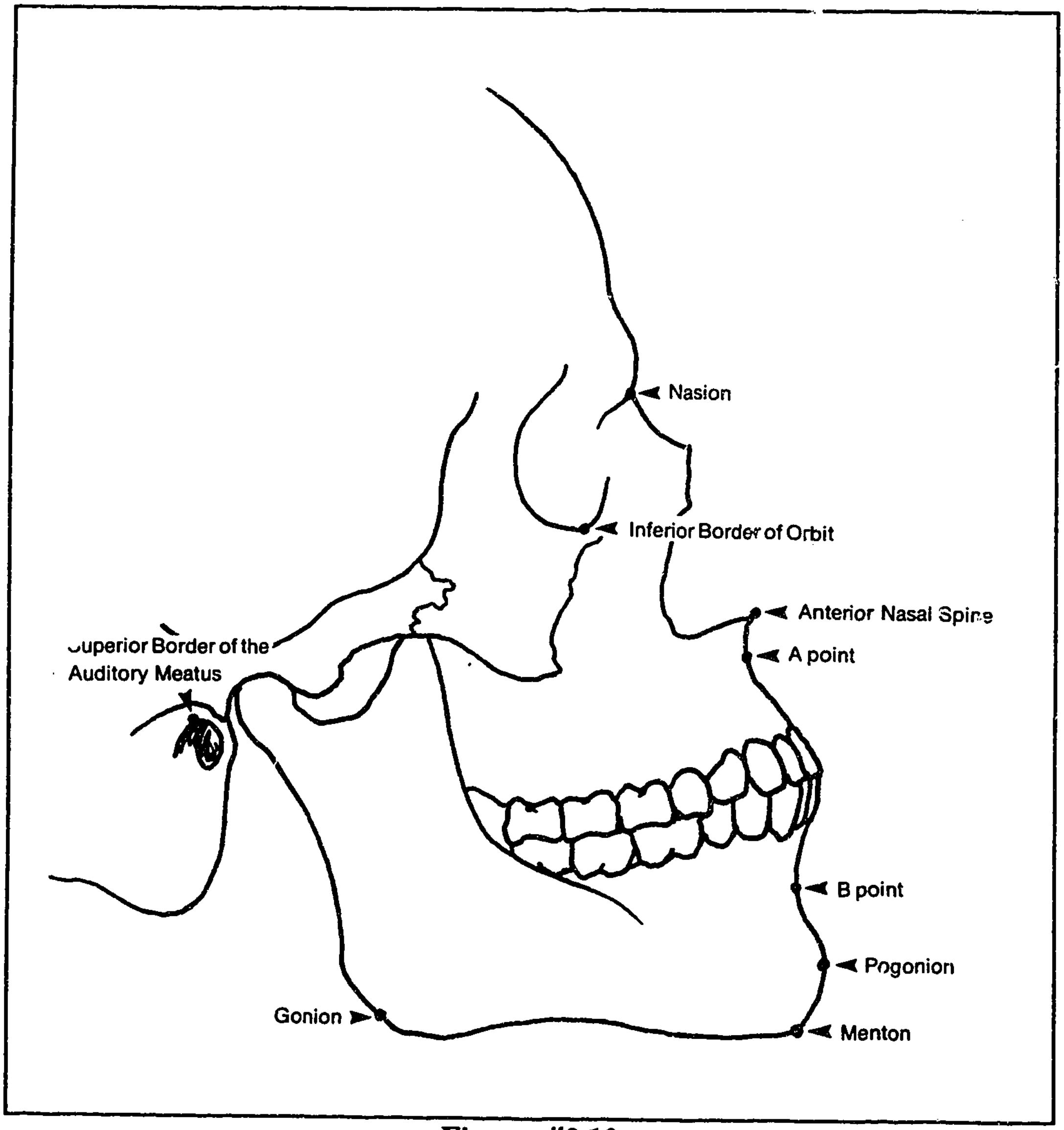



**Figure #2.10**
Lateral view showing landmarks

## 2.4 Digitization Hardware

Techniques were developed to collect, store and manipulate the three dimensional digital spatial information. The system had to provide accurate and reproducible results, and store these results in a computer datafile to allow easy retrieval for use in the mathematical model. To accomplish this a system was built from the following commercial components:

1) Summagraphics MM1200 two dimensional digitizing tablet with optional RS-232 interface

2) Mitutoyo 192-65 twelve inch digital height gauge

3) Mitutoyo MUX-10 serial multiplexor

4) Zenith Z183-92 laptop computer

A custom built frame was used to maintain the spatial alignment of the components and to act as a skull fixation device. Finally a serial multiplexor was used to combine the two dimensional information from the tablet with the one dimensional information from the height gauge into a single serial signal readable by the computer. A simplified system is depicted in Figure #2.11.

The unit incorporated a cartesian XYZ coordinate system with the tablet providing the X and Y coordinates, and the height gauge providing the Z coordinate. A pointer which was attached to the height gauge was aligned with the center point of the 2D tablet's puck mounted in the height gauge base (see Figure #2.12). This configuration allowed complete freedom of movement for the vertical tower within the 12 x 12 inch area of the tablet, yielding a one cubic foot digitization volume. Within this volume the measurement resolution was one thousandth of an inch for each direction.

A frame was built to hold the tablet and act as the base platform. A 1/4 inch glass plate was fitted on top of the tablet providing a smooth, level surface which allowed easy travel and positioning of height gauge. To mount the skull in a fixed but accessible location an adjustable bracket was attached to a vertical post at the back of the platform (see Plate #2.1). With this bracket it was possible to adjust the angle of the skull in either the X-Z plane, the Y-Z plane or in its vertical position. The skull was attached by means of a collapsible wing clasp inserted through the foramen magnum.

A "triggered" serial multiplexor was built to combine the two individual serial signals from the tablet and height gauge into a single serial signal for the computer. When the operator 'triggered' it, the multiplexor would send a signal to the tablet and forward the reply to the computer. Immediately following the reply it would then send a signal to the height gauge and forward the reply. This process was completed very quickly, and it would appear to the computer that it was receiving a single signal of the XYZ coordinates.

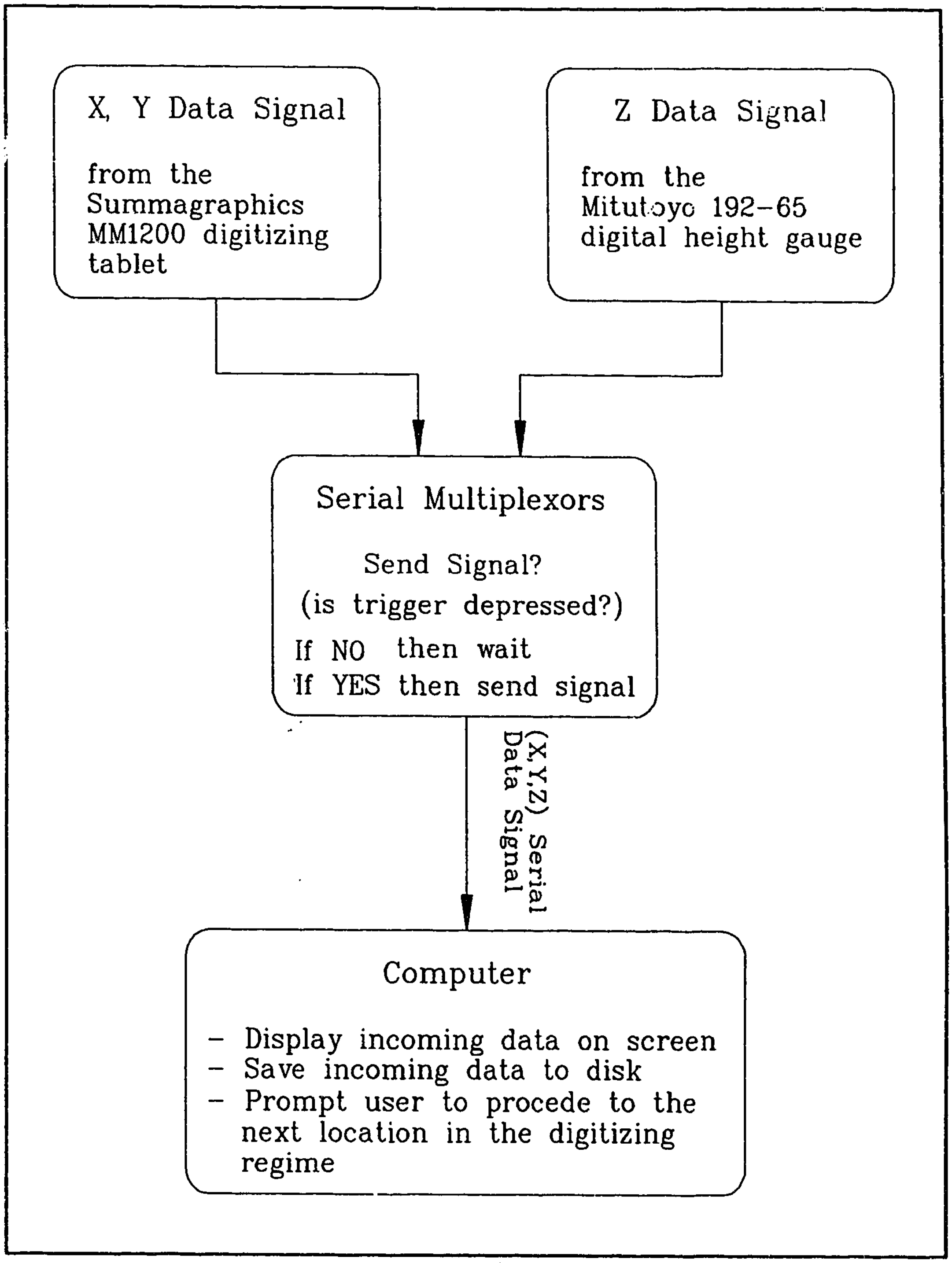



**Figure #2.11**
Digitizer system schematic

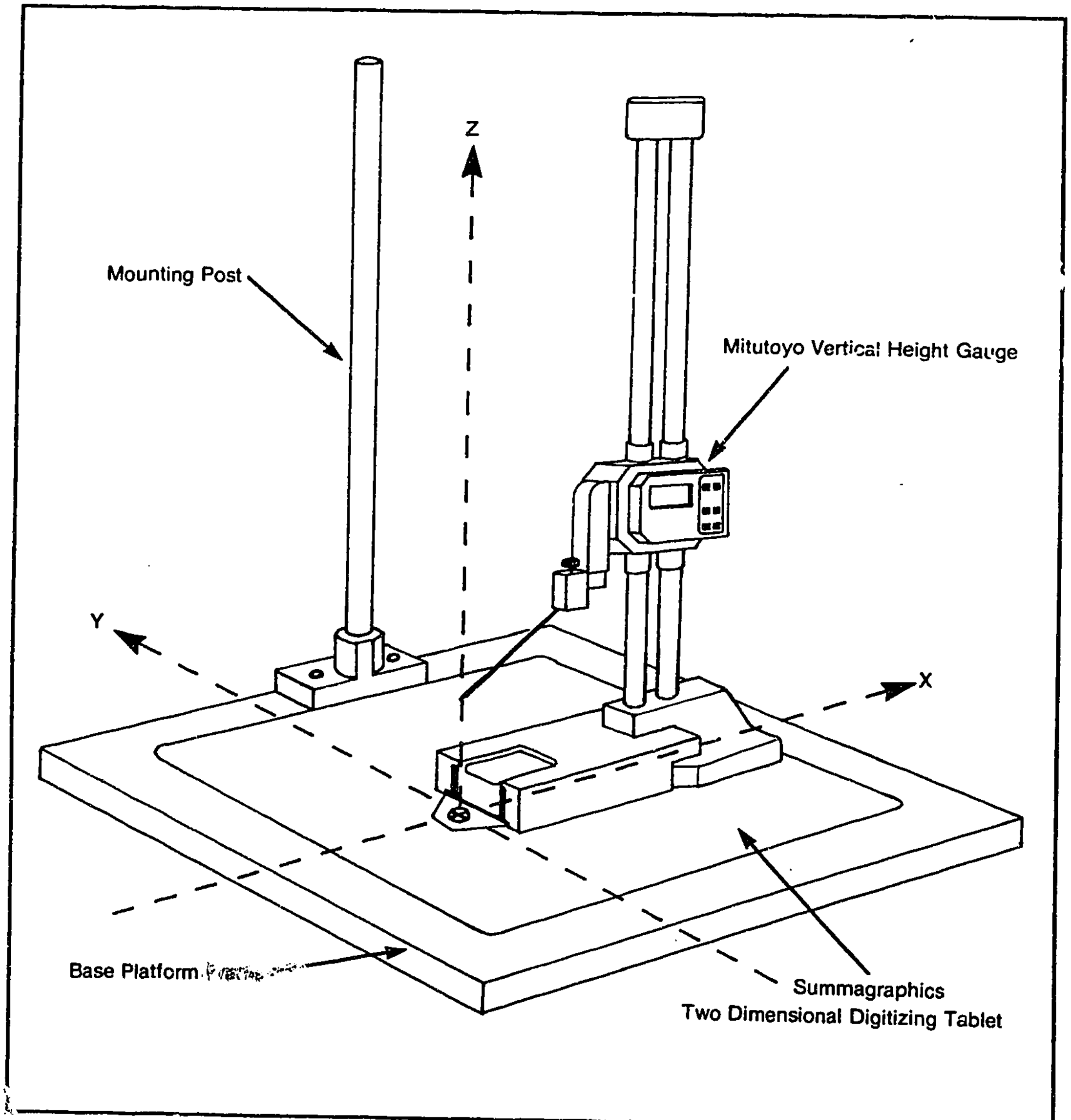
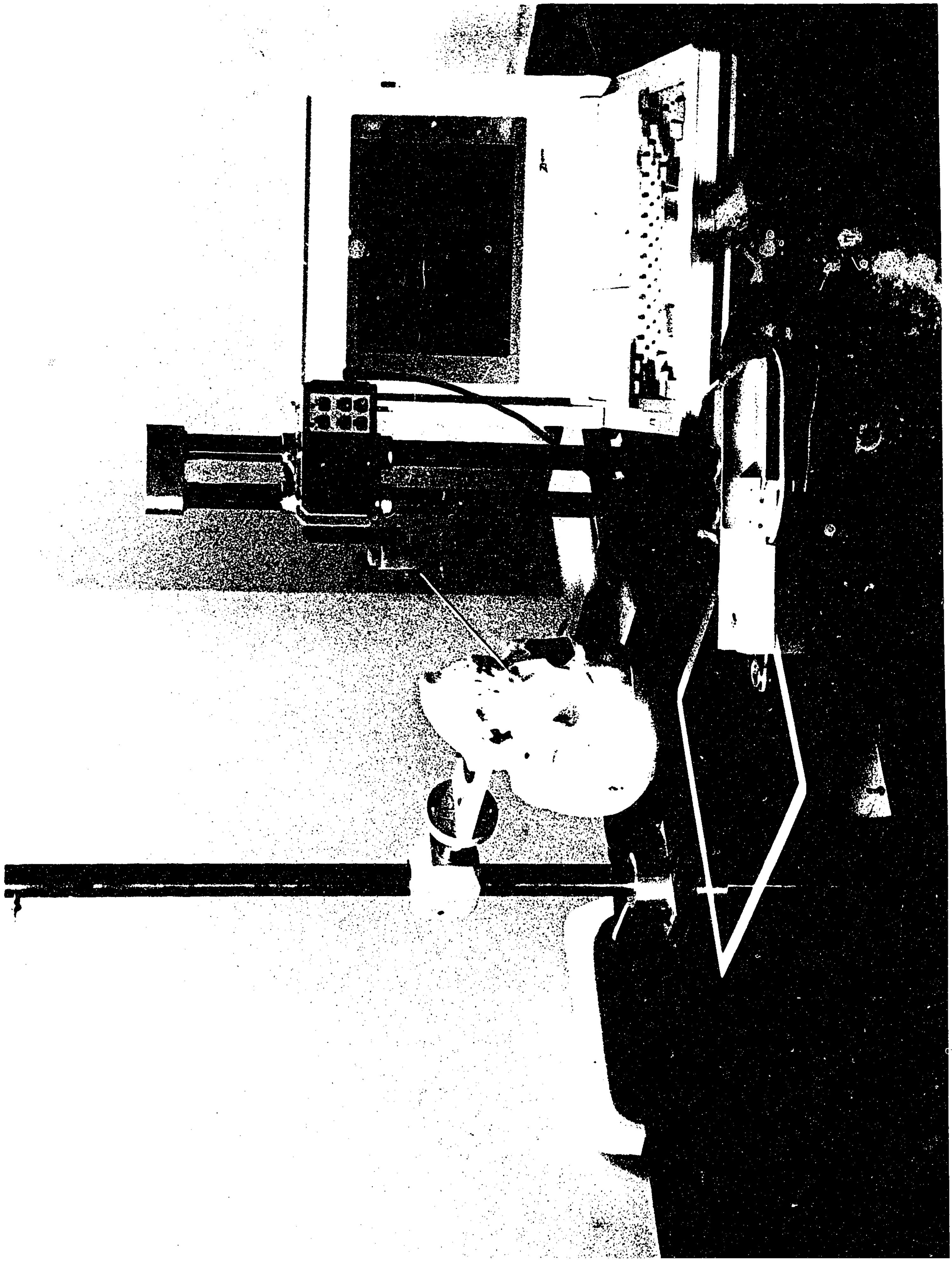



**Figure #2.12**
Perspective view of digitizer components



## 2.5 Digitization Software

To complement the hardware system a software program was written to systemize the data acquisition and produce the formatted output files. The program provided a user interface that would prompt the user for information about the skull as well as inform the operator on the order of the digitizing regime. The program was written in Microsoft QuickBasic version 4.5 and the source listing is included in Appendix #1 for completeness. The logic inherent in the program flow is outlined in Figure #2.13. The program's primary function was to acquire the data and save it in a specified output format for later use. Editing and commenting features were included to provide a means to check uncertain locations and correct mistakes when necessary.

Included in every datafile along with each point's three coordinates and comments was an abbreviated location name so that the files would be readily understandable. Each file was also given an information header containing the skull's class as well as the date and time that it was digitized. An excerpt of a datafile is included in Figure #2.14.

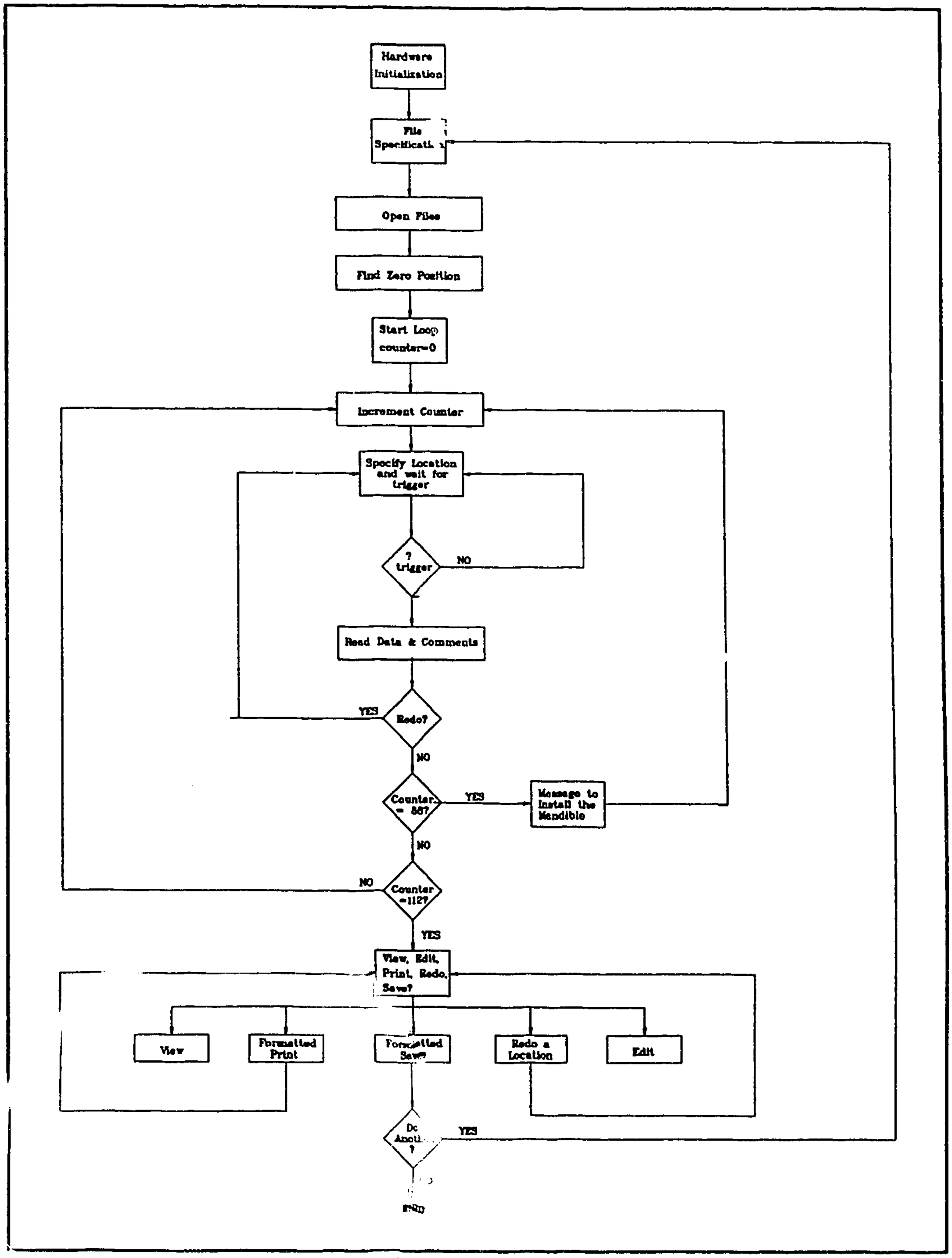



**Figure #2.13**
Digitizer software flow chart

```
1, 3
10:17:27
03-18-1989
LOCATION #  1 X=  -0.056 Y=  +0.272 Z=  +7.094 Z.P. NO COMMENT
LOCATION #  2 X=  -0.076 Y=  +0.412 Z=  +6.868 A NO COMMENT
LOCATION #  3 X=  -0.078 Y=  +0.252 Z=  +6.743 A.N.S. NO COMMENT
LOCATION #  4 X=  -0.042 Y=  +0.452 Z=  +4.997 NAS. NO COMMENT
LOCATION #  5 X=  +1.100 Y=  +0.822 Z=  +6.119 R.INF.ORB. NO COMMENT
LOCATION #  6 X=  +1.[illegible]36 Y=  +1.678 Z=  +4.458 R.ANT.TEMP.o NO COMMENT
LOCATION #  7 X=  +2.5[illegible]  +3.618 Z=  +3.869 R.MID.TEMP.o NO COMMENT
LOCATION #  8 X=  +2.5[illegible] Y=  [illegible]3.952 Z=  +5.127 R.POST.TEMP.o NO COMMENT
LOCATION #  9 X=  +2.0[illegible]  [illegible]1.550 Z=  +6.521 R.SUP.MAS.o NO COMMENT
LOCATION # 10 X=  +2.37[illegible]  +2.434 Z=  +6.250 R.DEEP.MAS.o NO COMMENT
LOCATION # 11 X=  +0.710 Y=  +2.318 Z=  +6.719 R.LAT.PT.o NO COMMENT
LOCATION # 12 X=  -0.582 Y=  +2.356 Z=  +6.680 L.MED.PT.o NO COMMENT
LOCATION # 13 X=  -0.090 Y=  +2.228 Z=  +6.954 P.N.S. NO COMMENT
LOCATION # 14 X=  +2.186 Y=  +3.004 Z=  +6.304 R.F.P.1 NO COMMENT
LOCATION # 15 X=  +1.846 Y=  +2.918 Z=  +6.360 R.F.P.2 NO COMMENT
LOCATION # 16 X=  +1.528 Y=  +3.186 Z=  +6.346 R.F.P.3 NO COMMENT
LOCATION # 17 X=  +1.842 Y=  +3.180 Z=  +6.182 R.F.P.4 NO COMMENT
LOCATION # 18 X=  +2.020 Y=  +3.782 Z=  +6.133 R.SUP.AUD.M. NO COMMENT
LOCATION # 19 X=  +1.952 Y=  +3.662 Z=  +6.359 R.ANT.AUD.M.SUF. NO COMMENT
LOCATION # 20 X=  +1.472 Y=  +3.520 Z=  +6.368 R.ANT.AUD.M.DP. NO COMMENT
LOCATION # 21 X=  +1.080 Y=  +1.968 Z=  +7.445 RM31 NO COMMENT
LOCATION # 22 X=  +1.110 Y=  +1.752 Z=  +7.475 RM32 NO COMMENT
```

**Figure #2.14**

Excerpt of datafile TMJ58.DAT

(i.e. location #8 had the XYZ coordinates of (+2.508, +3.952, +5.127). The abbreviated location was for the Right Posterior Temporalis Origin.)

## 2.6 Standard Radiograph Projections

A standard 60 inch midobject focal distance posteroanterior (P/A) and lateral cephalometric projection were taken of each skull. The lateral projection was taken with the skull positioned and stabilized in a cephalostat with the Frankfort plane in a horizontal orientation (parallel to the floor) and the sagittal plane perpendicular to the film and tube. The right side of the skull was placed closest to the film. The P/A projection was also taken with the Frankfort plane aligned horizontally and the axial plane perpendicular to the film and tube. The anterior portion of the skull was placed closest to the film.

These radiographs provided the morphological information necessary to calculate the SNA, SNB and the ANB angle, which was then used to designate the skeletal classification. The radiographs would provide an additional source of spatial information that could be used to verify the digitized coordinates. In addition the lateral film also rendered an excellent image of the dentition.

## 2.7 Photographs

A standard series of five photographs were also taken of each of the 53 skulls from the Atkinson collection. The photographs provided a supplemental source of spatial and morphological information. It was thought that scaled measurements from the photos could be used if it was found necessary to check the validity of any digitized spatial data. The series consisted of (see Plate #2.2):

1) - Full frontal view similar to the P/A cephalogram.

2) - Left lateral view similar to the lateral cephalogram.

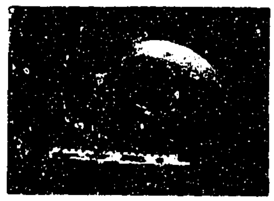
3) - Submental vertex view, looking superior. This view recorded information about the dentition in the maxilla and the fossae. Focal plane was aligned with the plane of the dentition.

4) - Inferoposterior view of the mandible. This view was taken to capture information about the condyles. The focal plane of the camera was through both condyles.

5) - Inferior view of the mandible. This view was taken to capture information about the dentition. The focal plane was approximately set to the occlusal plane.

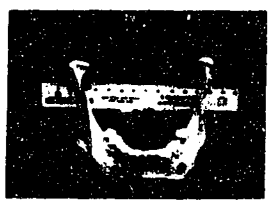
Besides the standard photos, pictures were taken of any unusual or interesting features (abnormalities), such as worn or remodelled condyles, asymmetries, etc.

#1 Frontal View



#2 Lateral View

#3 Submental View



#4 Condylar View

#5 Occlusal View

**Plate #2.2**
The different views of the photographic series

# 3.0 The Numerical Model

## 3.1 Overview

The mathematical model used in the current study was derived from an earlier model developed by Mr. A. Hay at the University of Alberta [10]. This model is based on a static unilateral occlusion loading situation, under maximum occlusal loading conditions. It was believed that this situation represents the conditions under which the joint is most heavily loaded and therefore most responsible for generating situations that are detrimental to the joint.

The core of this model requires three groups of data: the definition of the location, direction and magnitude of active muscle force vectors; the spatial location of the reaction forces within the left and right TMJ; and finally the direction and location of the occlusal contact. The model will then provide the values of the resultant reaction forces at each condyle and the resultant occlusal load magnitude.

## 3.2 Muscle Directions and Magnitudes

The original spatial information gathered by the digitizer for each skull was pre-processed to reorient the digitized coordinate locations with respect to an arbitrary cartesian co-ordinate system. This coordinate system was centered on the Frankfort plane at the intersection of a line extended superiorly from the anterior nasal spine (see Figure #3.1). The X-Y plane corresponds to the Frankfort plane and similarly the Y-Z plane conforms to the mid-sagittal plane as defined in the current model. Once the original data was reorientated the pre-processing step calculated each of the 14 normalized unit vectors that described the direction of each muscle vector to be used by the model (see Figure #3.2). These unit vectors and the reorientated digitized coordinates were then stored for later use.

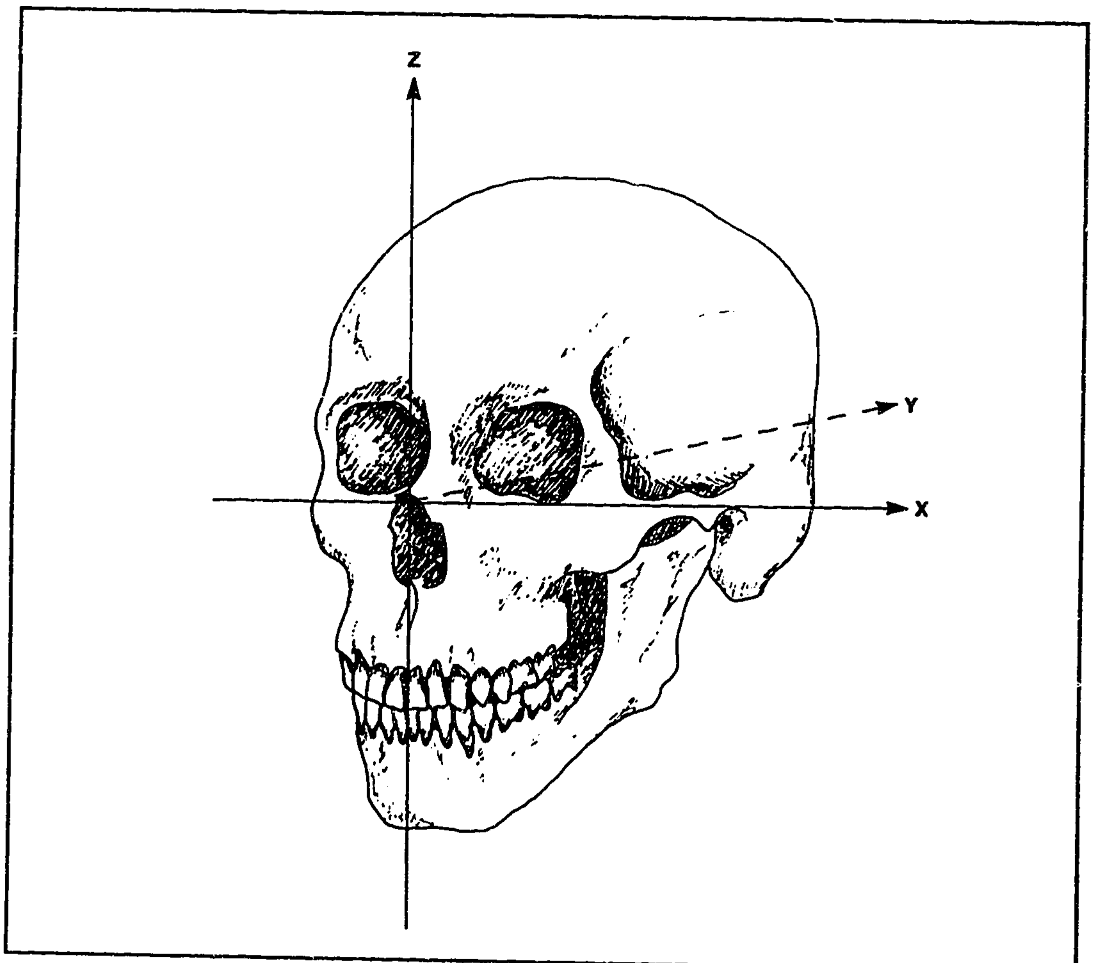



**Figure #3.1**
Coordinate system orientation

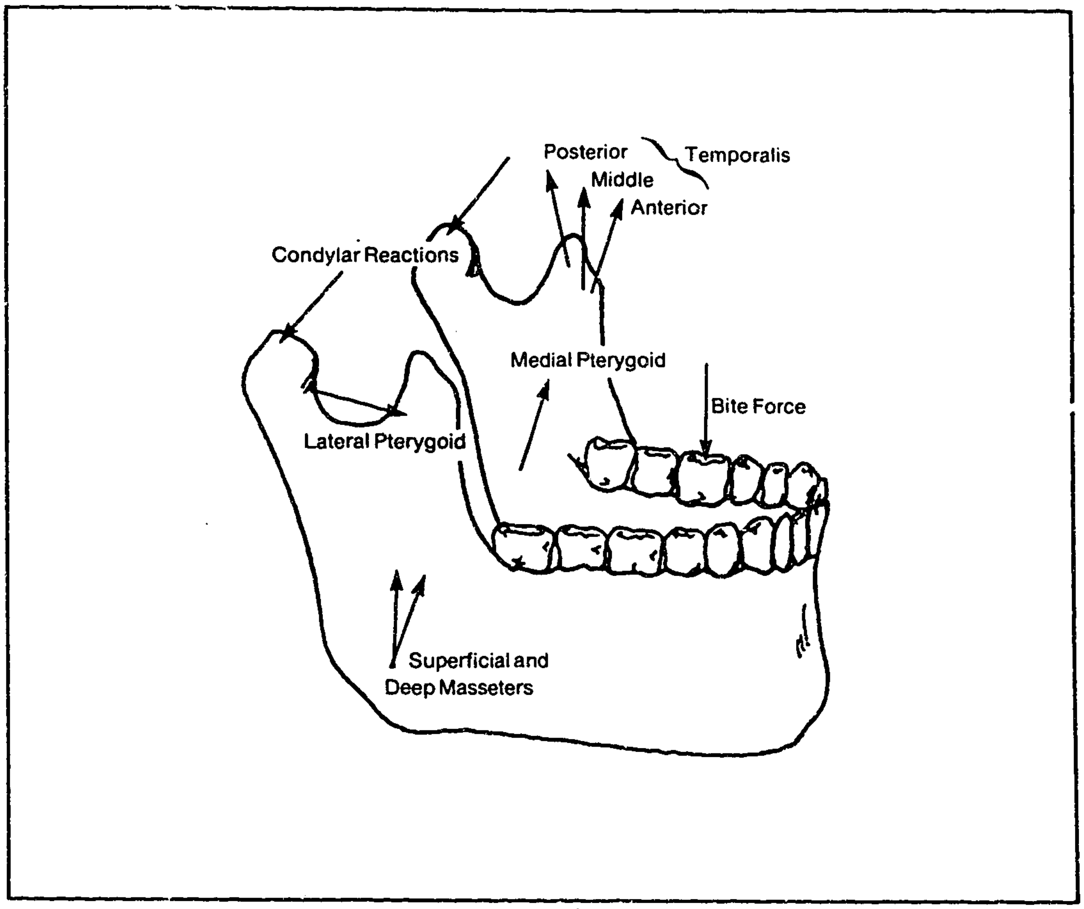



**Figure #3.2**
Simplified free body diagram showing the forces applied to a loaded mandible.

The magnitudes of the muscle forces applied to each muscle vector used in the present model are nearly the same as the Type II muscle forces described in Hay's study. These Type II magnitudes were derived from knowledge of the muscle's cross-sectional area and measured electromyographical (EMG) data. Application of the muscle forces was based on the total combined muscle input equalling 100 arbitrary load units, with each of the 14 muscles comprising a percentage of the total.

Hay's Type II muscle forces for the temporalis muscle's anterior and posterior components were modified for use with the present model's anterior, middle and posterior temporalis muscle components. The temporalis has a large fan shaped origin (see Figure #1.7) which encompasses a wide directional range. The temporalis was represented by three components to obtain an improved approximation compared with the two components used by Hay. This was done by using a weighted average method in which one third of each of Hay's two temporalis muscle value components were taken and assigned to the middle temporalis muscle in the current model (see Figure #3.3). To demonstrate the effects of this change the resultant loads using the original and modified muscle forces were calculated on one representative skull from each class (see Table #3.1). The resultant ipsilateral condylar loads all decreased, with a maximum reduction of 6.3 percent occurring for the Class III example. The resultant contralateral condylar loads increased similarly, with a maximum increase of 3.3 percent for the Class I example. The occlusal load magnitudes were least effected by the change with minimal decreases of up to 2.8 percent for the Class III case.

The muscle magnitudes selected are certainly not exactly correct for each skull in the study (that would require *in vivo* information for all the skulls which was unavailable), but rather it is their relative proportions that are considered appropriate. The majority of the force comes from the masseters (35.5% of total

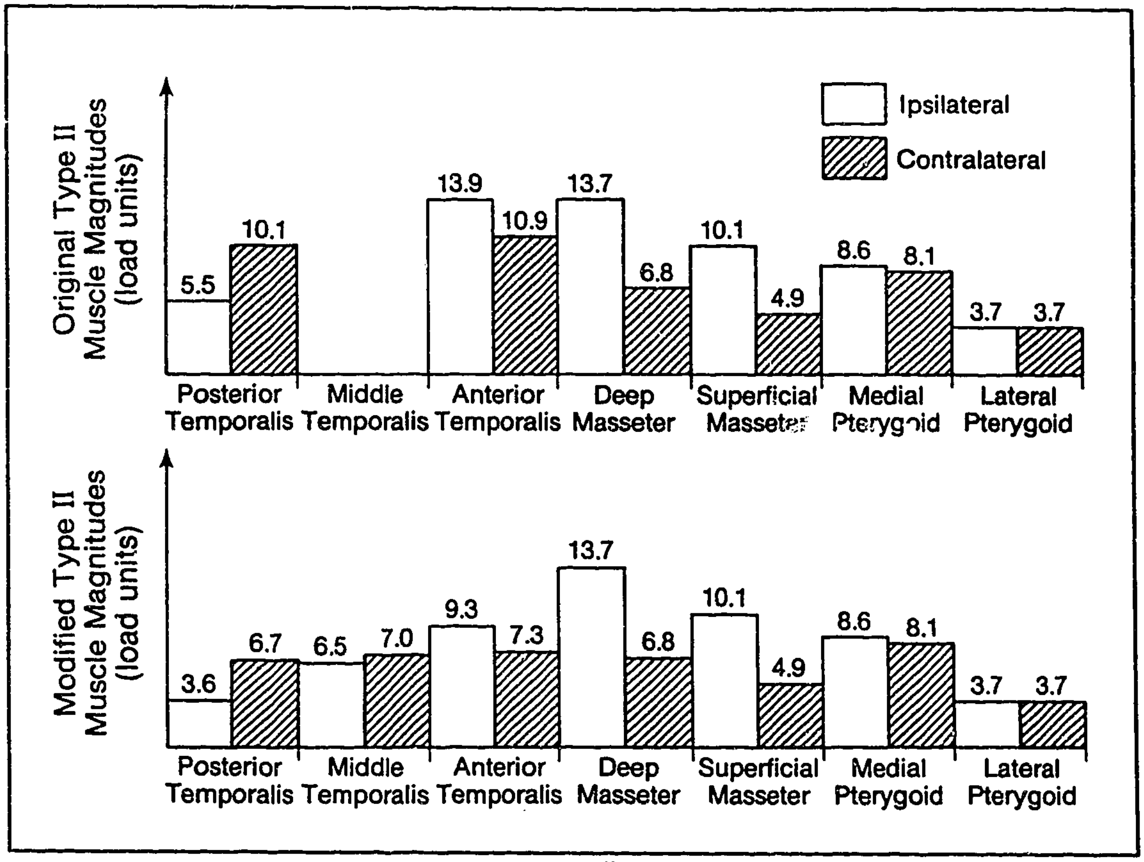



**Figure #3.3**
Original and Modified Type II muscle magnitudes

muscle force) and the temporalis muscles (40.4% of total muscle force), and the balancing side contributes less force than the working side (44.5% compared to 55.5%, respectively). The lateral pterygoid is considered to act primarily as a stabilizing force.

As the values used in this model are not precise, an effort was made to determine how sensitive the results computed by the model would be to alterations in these muscle magnitudes. To evaluate this, each individual muscle magnitude was increased by 25% of its original value, while the 13 other muscles were decreased proportionally to maintain a total muscle load of 100 load units (see Table #3.2). Table #3.3 and Figure #3.4 show the effects of these variations on the ipsilateral, contralateral and resultant occlusal loads of skull #18, selected as an average representation of Class I. Also shown in Table #3.3 are the ipsilateral and contralateral parasagittal inclines of the resultant condylar loads (see Figures #3.5 and #3.6).

**Table #3.1**
Effects of modified muscle magnitudes on the models resultant loads.

| | | Class I | Class II | Class III |
|---|---|---|---|---|
| Ipsilateral Condyle | (Original) | 19.4 | 20.3 | 20.8 |
| | (Modified) | 18.8 | 20.0 | 19.5 |
| | (% Change) | -3.1 | -1.5 | -6.3 |
| Contralateral Condyle | (Original) | 21.4 | 17.8 | 22.3 |
| | (Modified) | 22.1 | 18.3 | 22.8 |
| | (% Change) | +3.3 | +2.8 | +2.2 |
| Resultant Occlusal Load | (Original) | 38.9 | 38.0 | 42.8 |
| | (Modified) | 38.7 | 37.8 | 41.6 |
| | (% Change) | -0.5 | -0.5 | -2.8 |

## Table #3.2
Muscle magnitudes for model sensitivity analysis. 25% individual increases.

| Run # | Posterior Temporalis Ipsi. | Posterior Temporalis Contra. | Middle Temporalis Ipsi. | Middle Temporalis Contra. | Anterior Temporalis Ipsi. | Anterior Temporalis Contra. | Deep Masseter Ipsi. | Deep Masseter Contra. | Superior Masseter Ipsi. | Superior Masseter Contra. | Medial Pterygoid Ipsi. | Medial Pterygoid Contra. | Lateral Pterygoid Ipsi. | Lateral Pterygoid Contra. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1* | 3.6 | 6.7 | 6.5 | 7.0 | 9.3 | 7.3 | 13.7 | 6.8 | 10.1 | 4.9 | 8.6 | 8.1 | 3.7 | 3.7 |
| 2 | 4.5 | 6.6 | 6.4 | 6.9 | 9.2 | 7.2 | 13.6 | 6.8 | 10.0 | 4.9 | 8.5 | 8.0 | 3.7 | 3.7 |
| 3 | 3.5 | 8.4 | 6.4 | 6.9 | 9.1 | 7.2 | 13.5 | 6.7 | 9.9 | 4.8 | 8.4 | 8.0 | 3.6 | 3.6 |
| 4 | 3.5 | 6.6 | 8.1 | 6.9 | 9.1 | 7.2 | 13.5 | 6.7 | 9.9 | 4.8 | 8.5 | 8.0 | 3.6 | 3.6 |
| 5 | 3.5 | 6.6 | 6.4 | 8.8 | 9.1 | 7.2 | 13.4 | 6.7 | 9.9 | 4.8 | 8.4 | 8.0 | 3.6 | 3.6 |
| 6 | 3.5 | 6.5 | 6.3 | 6.8 | 11.6 | 7.1 | 13.4 | 6.6 | 9.8 | 4.9 | 8.4 | 7.9 | 3.6 | 3.6 |
| 7 | 3.5 | 6.6 | 6.4 | 6.9 | 9.1 | 9.1 | 13.4 | 6.7 | 9.9 | 4.9 | 8.4 | 7.9 | 3.6 | 3.6 |
| 8 | 3.5 | 6.4 | 6.3 | 6.7 | 8.9 | 7.0 | 17.0 | 6.5 | 9.7 | 4.7 | 8.3 | 7.8 | 3.6 | 3.6 |
| 9 | 3.5 | 6.6 | 6.4 | 6.9 | 9.1 | 7.2 | 13.5 | 8.5 | 9.9 | 4.8 | 8.4 | 8.0 | 3.6 | 3.6 |
| 10 | 3.5 | 6.5 | 6.3 | 6.8 | 9.0 | 7.1 | 13.3 | 6.6 | 12.6 | 4.8 | 8.4 | 7.9 | 3.6 | 3.6 |
| 11 | 3.6 | 6.6 | 6.4 | 6.9 | 9.2 | 7.2 | 13.4 | 6.7 | 10.0 | 6.1 | 8.5 | 8.0 | 3.7 | 3.7 |
| 12 | 3.5 | 6.5 | 6.4 | 6.8 | 9.1 | 7.1 | 13.4 | 6.6 | 9.9 | 4.8 | 10.8 | 7.9 | 3.6 | 3.6 |
| 13 | 3.5 | 6.6 | 6.4 | 6.8 | 9.1 | 7.1 | 13.4 | 6.7 | 9.9 | 4.8 | 8.4 | 10.1 | 3.6 | 3.6 |
| 14 | 3.7 | 6.6 | 6.4 | 6.9 | 9.2 | 7.2 | 13.6 | 6.7 | 10.0 | 4.9 | 8.5 | 8.0 | 4.6 | 3.7 |
| 15 | 3.7 | 6.6 | 6.4 | 6.9 | 9.2 | 7.2 | 13.6 | 6.7 | 10.0 | 4.9 | 8.5 | 8.0 | 3.7 | 4.6 |

* denotes the normally assigned muscle magnitudes.
Shaded value indicates muscle with the 25% increase.

## Table #3.3
Resultant loads and angles from sensitivity tests. 25% increase in magnitude.

| Run # | Ipsilateral Resultant Condylar Load (Load Units) | Contralateral Resultant Condylar Load (Load Units) | Resultant Occlusal Load (Load Units) | Ipsilateral Parasagittal Incline (Degrees) | Contralateral Parasagittal Incline (Degrees) |
|---|---|---|---|---|---|
| 1* | 18.4 | 22.0 | 36.0 | 26.4 | 4.7 |
| 2 | 18.3 | 21.7 | 35.7 | 23.8 | 5.1 |
| 3 | 17.9 | 22.6 | 35.5 | 26.7 | 0.8 |
| 4 | 18.6 | 21.6 | 35.9 | 22.6 | 4.8 |
| 5 | 17.6 | 23.0 | 35.9 | 27.2 | 1.7 |
| 6 | 19.0 | 21.2 | 36.6 | 24.7 | 5.5 |
| 7 | 17.3 | 23.2 | 36.4 | 27.8 | 4.2 |
| 8 | [illegible] | 20.2 | 35.6 | 24.0 | 5.3 |
| 9 | 17.5 | 23.5 | 35.8 | 27.2 | 5.2 |
| 10 | 19.6 | 20.7 | 36.7 | 29.0 | 5.0 |
| 11 | 17.4 | 22.6 | 36.3 | 27.7 | 6.6 |
| 12 | 18.8 | 21.8 | 36.0 | 28.1 | 4.8 |
| 13 | 18.1 | 22.5 | 36.0 | 26.4 | 6.5 |
| 14 | 18.0 | 21.7 | 35.7 | 28.5 | 5.0 |
| 15 | 18.2 | 21.3 | 35.7 | 26.3 | 6.6 |

* denotes results computed with the normally assigned muscle magnitudes

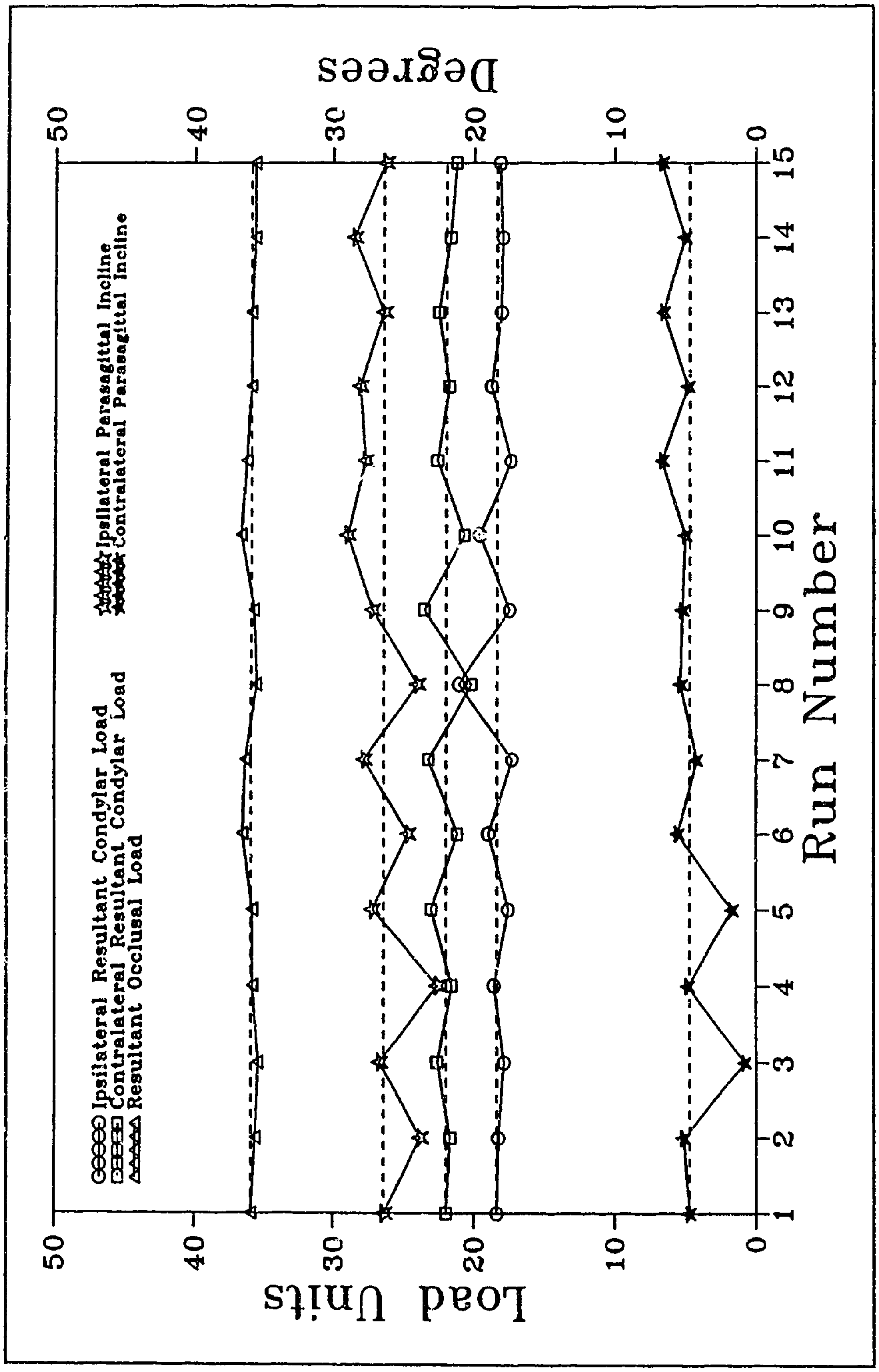



**Figure #3.4**
Variations in resultant loads and angles for sensitivity tests.

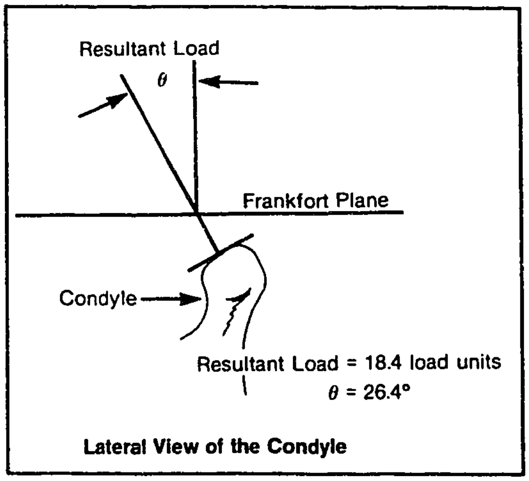



**Figure #3.5**
Ipsilateral loading pattern for Run #1.

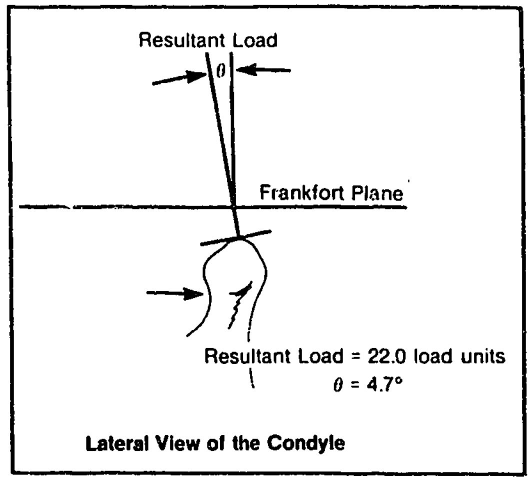



**Figure #3.6**
Contralateral loading pattern for Run #1.

The ipsilateral resultant condylar loads had an average absolute deviation of 0.7 load units, a 4.0% variation. The contralateral resultant condylar loads behaved similarly with an average absolute deviation of 0.8 load units, a 3.6% variation. The resultant occlusal load was largely unaffected by the changes in muscle magnitudes with an average absolute deviation of 0.3 load units, a minimal 0.8% variation. The Occlusal load had a maximum deviation of +0.7 load units (1.9%) for the 25% increase of the ipsilateral superficial masseter. The ipsilateral and contralateral condylar loads reacted predictably, with the maximum deviations occurring in conjunction with the largest single muscle increase. When the ipsilateral deep masseter was increased 25% (a 3.3 load unit increase) the ipsilateral and contralateral resultant condylar loads changed +14.7% and -8.2% (+2.7 and -1.8 load units) respectively.

The ipsilateral parasagittal incline was insensitive to variations in the contralateral muscle magnitudes. All angles increased slightly with deviations less than 1.4° (except for Run #15 which decreased 0.1°). Deviations due to increases in the ipsilateral muscle loads reached a maximum of -3.8° for the middle temporalis.

The contralateral parasagittal incline was similarly insensitive to variations in the ipsilateral muscle magnitudes (all angles increased less than 0.8°). It was slightly more sensitive to changes in the contralateral muscle magnitudes with a maximum deviation of -3.9° for the change in the posterior temporalis muscle. This is consistent since the posterior temporalis is orientated with a largely posterior directed component, and its increase will reduce the level of the anterior component in the resultant load.

From this it was clear that the current model is relatively insensitive to the large individual increases in muscle forces. This agrees with the conclusions of Throckmorton in his 1985 paper [39] on the importance of the magnitude of jaw muscle forces. For this reason only the results of the model using the modified muscle magnitudes were used for all comparisons.

## 3.3 Joint Force Locations

The spatial location of the reaction forces on each condyle are needed for the model. It was desired to place each joint reaction force at the center of the wear facet on the condyle, but it was not possible to directly digitize this location with the mandible in place on the skull. As a result the joint reaction forces were positioned midpoint along a vector connecting the anteroinferior and posterosuperior two points of the four points that were digitized within each glenoid fossa. This vector was representative of the flat midline portion of the articular eminence. In all cases the desired area on the condyle was in contact with this point.

## 3.4 Occlusal Load Direction

Initially the unit vector describing the direction of the occlusal load was to be orientated perpendicular to the occlusal plane of the dentition, such that a line extended from the vector would pass through the central fossa of the selected tooth. The occlusal plane was determined from the dentition (see Figure #3.7). Bisecting a vector that connects the distal-lingual and mesial-lingual digitized cusps on both the left and right first molars provides two of the three points necessary to define the occlusal plane. The third point is derived by first bisecting a vector between the lingual cusp tip and the distal marginal ridge on both the left and right first premolars to provide two intermediate points. The third point was finally attained by bisecting a vector which joins the two intermediate points.

Any one of the left and right molars or premolars could be selected by this process, and once the unit vector perpendicular to the occlusal plane was calculated there was an option to tip the vector anteriorly or posteriorly in the

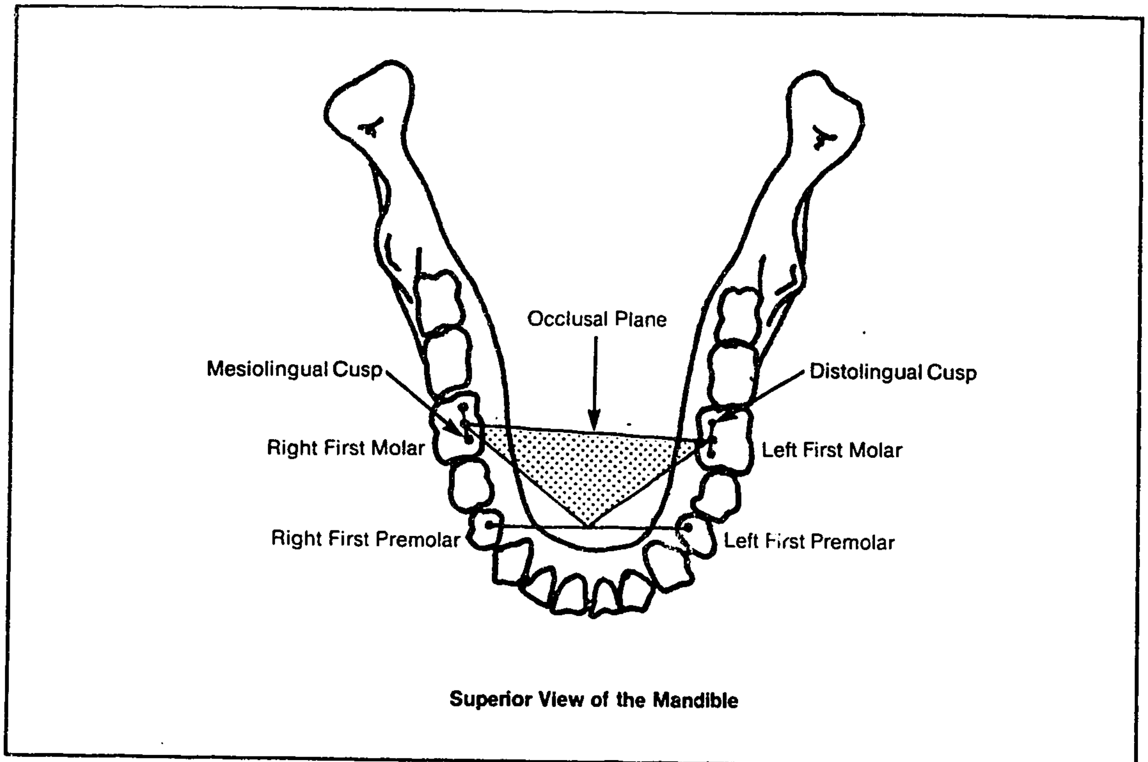



**Figure #3.7**
Definition of the occlusal plane

parasagittal plane.

## 3.5 Computer Model Details

The key to the current computer model is that it was developed with the flexibility to accommodate three dimensional spatial data from any number of skulls, rather than basing its findings on information from a single skull as used in previous studies [6, 10, 23, 33].

Once all the muscle magnitudes, vectors and locations are known along with the rest of the spatial and directional information the problem is reduced to a simple statics model. It is required that equilibrium exist if the mandible is not moving: therefore the sum of the forces in the X, Y and Z directions must equal zero and, the sum of all moments about each of the X, Y and Z axes must equal zero as well. To simplify some of the calculations the origin of the cartesian coordinate system was relocated to the right condyle's force application point. These conditions provide the six classical equations of static equilibrium:

$\Sigma F_x = 0$ (Sum of the forces in the X direction equal zero)

$\Sigma F_y = 0$ (Sum of the forces in the Y direction equal zero)

$\Sigma F_z = 0$ (Sum of the forces in the Z direction equal zero)

$\Sigma M_x = 0$ (Sum of the moments about the X axis equal zero)

$\Sigma M_y = 0$ (Sum of the moments about the Y axis equal zero)

$\Sigma M_z = 0$ (Sum of the moments about the Z axis equal zero)

Unknowns in the model are the three reactions at each condyles force application point (one reaction component in each of the X, Y and Z directions) and the occlusal load magnitude. These combine for a total of seven unknowns.

It was assumed that the X-direction (medial-lateral) reaction on the left condyle would be the same as the right condyle's. This assumption was necessary to provide a seventh equation so that the problem would reduce to a statically determinant one.

$R_x = L_x$ (Right condylar X-direction reaction equals the Left condylar X-direction reaction)

This assumption has been shown in previous studies not to significantly effect the results of the model [10].

Each muscles unit vector was then used to break its assigned muscle magnitude down into individual components in the X, Y and Z directions. Moment arms relative to the relocated origin were computed for each of the muscle magnitude components.

These component magnitude and moment arms were then combined with the information for the occlusal load to form the seven equations described previously. These formulae were then converted into an augmented matrix form and solved using a Gausian elimination technique. This provided the magnitudes of the seven unknown forces.

The current model was developed on an IBM PC-AT using Microsoft Quickbasic version 4.5, and the source code is contained in Appendix #2 for completeness. The test procedure was developed to be an inter-active, menu driven environment. The model followed the methodology outlined in Figure #3.8. At each stage the operator was asked to make a choice from a list of available options. The model was able to process one or all of the available skulls under the desired conditions. Once the selection process was complete, the program would proceed.

The program would then calculate the resultant occlusal and condylar loads for each skull specified under the current loading situation. These results were then automatically grouped by class, and all the information for the current run was written to an output file that was compatible for use with Lotus 123 for later analysis.

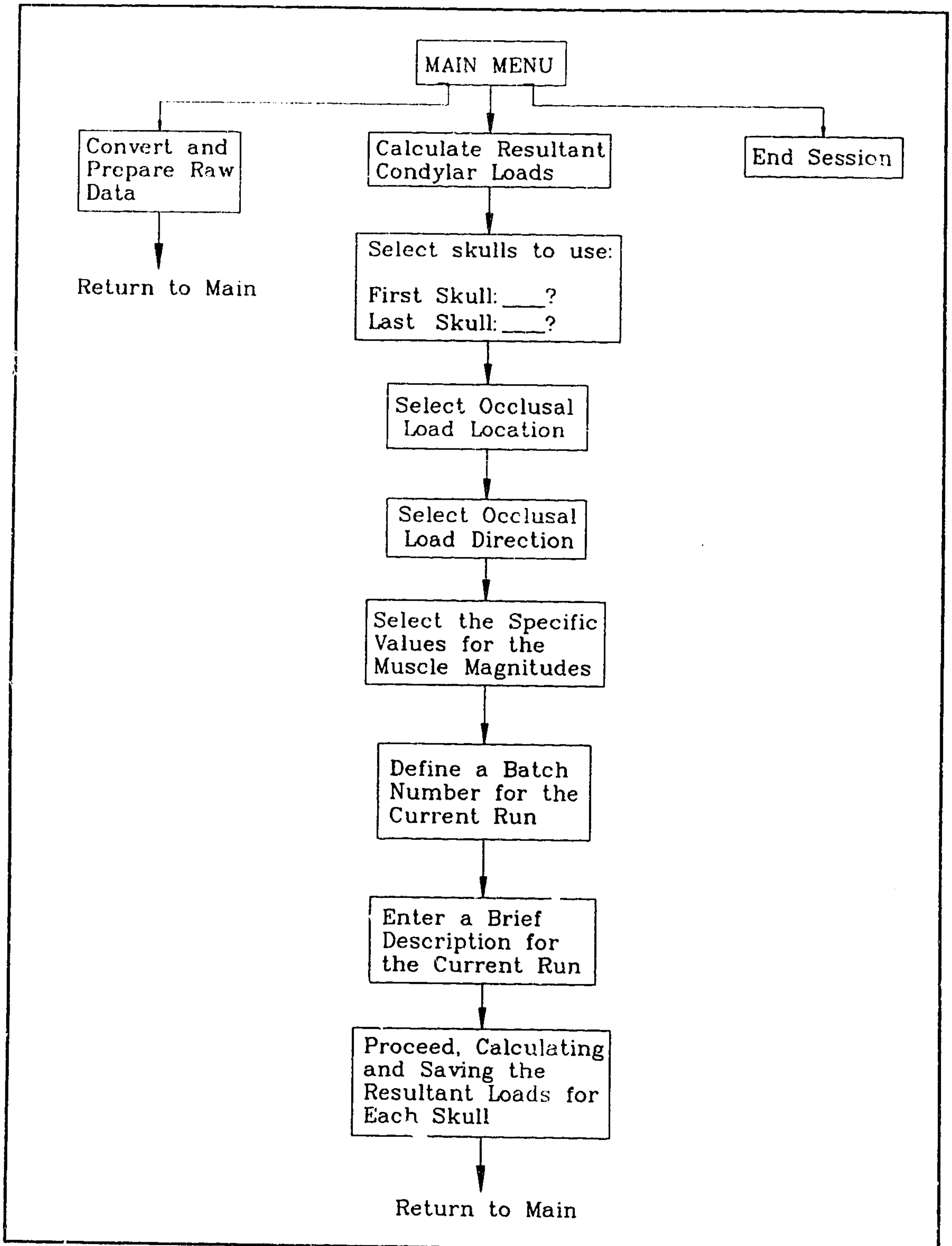



**Figure #3.8**
Flow chart of the numerical model

## 3.6 Model Comparisons

As a final test of the new model it was run under the conditions described by Faulkner et al and Hay [6, 10]. The studies used spatial information from a single Class I skull. Their investigations provided the necessary information on muscle inserts and vectors as well as the muscle magnitudes used (see Table #3.4). Tests were conducted for unilateral occlusion at the left first molar, whose spatial location was also listed. The occlusal load was orientated entirely in the Z direction.

**Table #3.4**

Muscle and occlusal parameters extracted from Hay's 1985 masters thesis on Mechanical Analysis of the TMJ [10].

| Muscle | Position Vector (inches) | Unit Vector |
|---|---|---|
| Right Posterior Temporalis | +0.18i -1.35j+0.00k | -0.10i +0.76j+0.64k |
| Right Middle Temporalis | +0.00i +0.00j+0.00k | +0.00i +0.00j+0.00k |
| Right Anterior Temporalis | +0.20i -1.48j -0.26k | -0.07i +0.34j+0.94k |
| Right Superficial Masseter | -0.05i -1.25j -1.75k | -0.27i -0.15j+0.95k |
| Right Deep Masseter | +0.00i -1.25j -1.75k | -0.27i +0.18j+0.94k |
| Right Medial Pterygoid | -0.05i -0.88j -1.75k | +0.32i -0.03j+0.94k |
| Right Lateral Pterygoid | +0.00i -0.25j -0.25k | +0.25i -0.94j -0.25k |
| Left Posterior Temporalis | +3.42i -1.35j+0.00k | +0.10i +0.76j+0.64k |
| Left Middle Temporalis | +0.00i +0.00j+0.00k | -0.00i +0.00j+0.00k |
| Left Anterior Temporalis | +3.40i -1.48j -0.26k | +0.07i +0.34j+0.94k |
| Left Superficial Masseter | +3.65i -1.25j -1.75k | +0.27i -0.15j+0.95k |
| Left Deep Masseter | +3.60i -1.25j -1.75k | +0.27i +0.18j+0.94k |
| Left Medial Pterygoid | +3.35i -0.88j -1.75k | -0.32i -0.03j+0.94k |
| Left Lateral Pterygoid | +3.60i -0.25j -0.25k | -0.25i -0.94j -0.25k |
| Left First Molar | +2.74i -2.88j -0.95k | +0.00i +0.00j+1.00k |
| Right Condylar Force Application Point | +0.00i +0.00j+0.00k | *not applicable* |
| Left Condylar Force Application Point | +3.60i +0.00j+0.00k | *not applicable* |

Under all conditions the models were in exact agreement (see Table #3.5). As will be shown in the following chapter, the loads shown here are in reasonable agreement with those of the group sample of Class I's.

**Table #3.5**
Comparison of results for different models.

| | Hay's 1985 Model | Current Model |
|---|---|---|
| Ipsilateral Resultant Condylar Load (load units) | 23.5* | 23.4 |
| Contralateral Resultant Condylar Load (load units) | 28.5* | 28.3 |
| Resultant Occlusal Load (load units) | 38.0* | 38.0 |

* Value interpreted from graphical presentation

# 4.0 Results

## 4.1 Introduction

Since individuals with a Class II craniofacial morphology have been clinically postulated to have a significantly higher percentage of TMJ dysfunction [29], then there should be some evidence of why this occurs in the joint loading patterns of this group of individuals. Previously studies have centered primarily on the magnitudes and ratios of the loads within the TMJ, with little emphasis directed towards understanding the implications of the directions in which these loads are applied. In the current study it is postulated that poor joint loading situations will be suggested by some combination of load magnitude and load misdirection, rather than either one exclusively.

To examine the loading differences between the three classes all seventy of the skulls were evaluated (using the model described previously) at six of the available occlusal positions (3 left molars and 3 right molars). At each of the occlusal locations the vector representing the direction of the occlusal load was varied in 5° increments parallel to the sagittal plane for a ±15° deviation from the normal to the occlusal plane.

It should be understood that although the current model produces precise numerical results, these results should only be interpreted using trends to assist in understanding the relationships rather than for quantitative differences. Until methods are developed to gain more accurate *in vivo* information on each individual within a sample there is considerable uncertainty in any model of this type. Therefore, even though the quantitative differences are described here, the conclusions drawn from trends and differences are more significant rather than the numbers themselves.

## 4.2 Investigation of Resultant Loads

In the current study it is postulated that the "better" of two loading patterns is the one that simultaneously produces the lowest condylar load and the highest levels of occlusal load. Conversely a "poorer" loading pattern exhibits higher levels of condylar loading while producing a lower level of occlusal load.

Initial examination of the results showed that there was little difference in the trends exhibited when the occlusal load was positioned on either the right or the left side, therefore only results for occlusion on the left side are presented in this study. As well, even though the results for occlusion on the third molar are presented and generally follow the prevailing trends, they are not discussed in detail as the third molar is generally considered to play a minor role in the mastication cycle (the third molar was not present in a number of skulls used in this study).

### 4.2.1 Loads Perpendicular to Occlusal Plane

The average magnitudes of the resultant occlusal, ipsilateral and contralateral condylar loads for occlusion on the three left molars are shown in Figure #4.1 for each of the three skeletal classes (see the statistical analysis in section 4.3.1 for information on the derivation of these average values and their significance). In this case, the direction of the occlusal load was perpendicular to the occlusal plane. Figure #4.2 shows the corresponding levels of variance in these values.

The magnitudes of the occlusal loads were as expected, that is increasing with a posterior shift of the occlusal position along the dentition. This was anticipated since a posterior shift shortens the moment arm of the occlusal vector, and hence a higher occlusal magnitude is required to provide the same balancing moment. The levels of variance also increased with the posterior shift of the occlusal vector, but it is uncertain whether this increase is actually due to a less stable craniofacial morphology or a manifestation of the inaccuracies of the model combining with an increasing occlusal load to produce proportionally higher levels of variance.

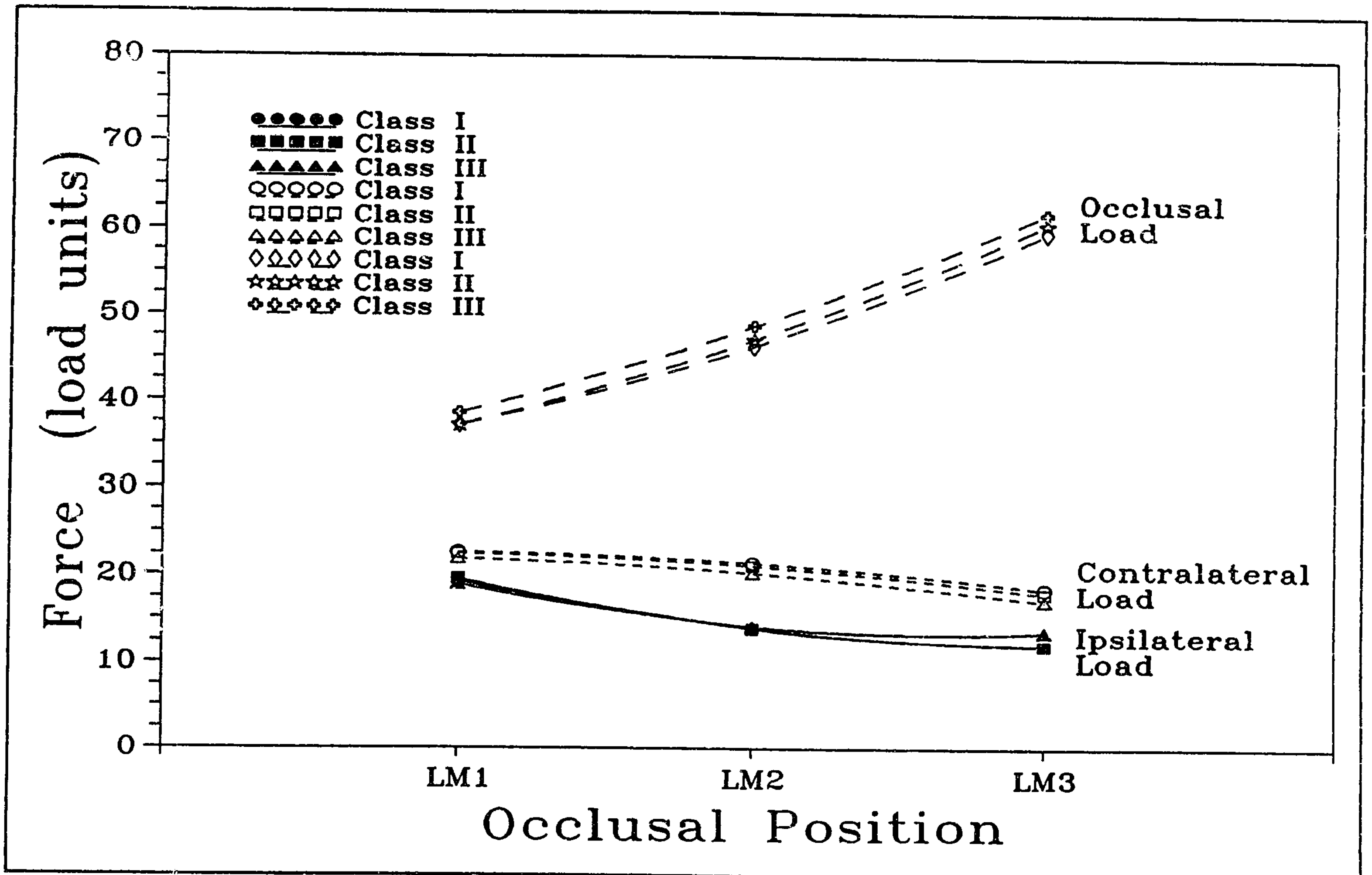



**Figure #4.1**
Resultant occlusal and joint loads with the occlusal load directed perpendicular to the occlusal plane.

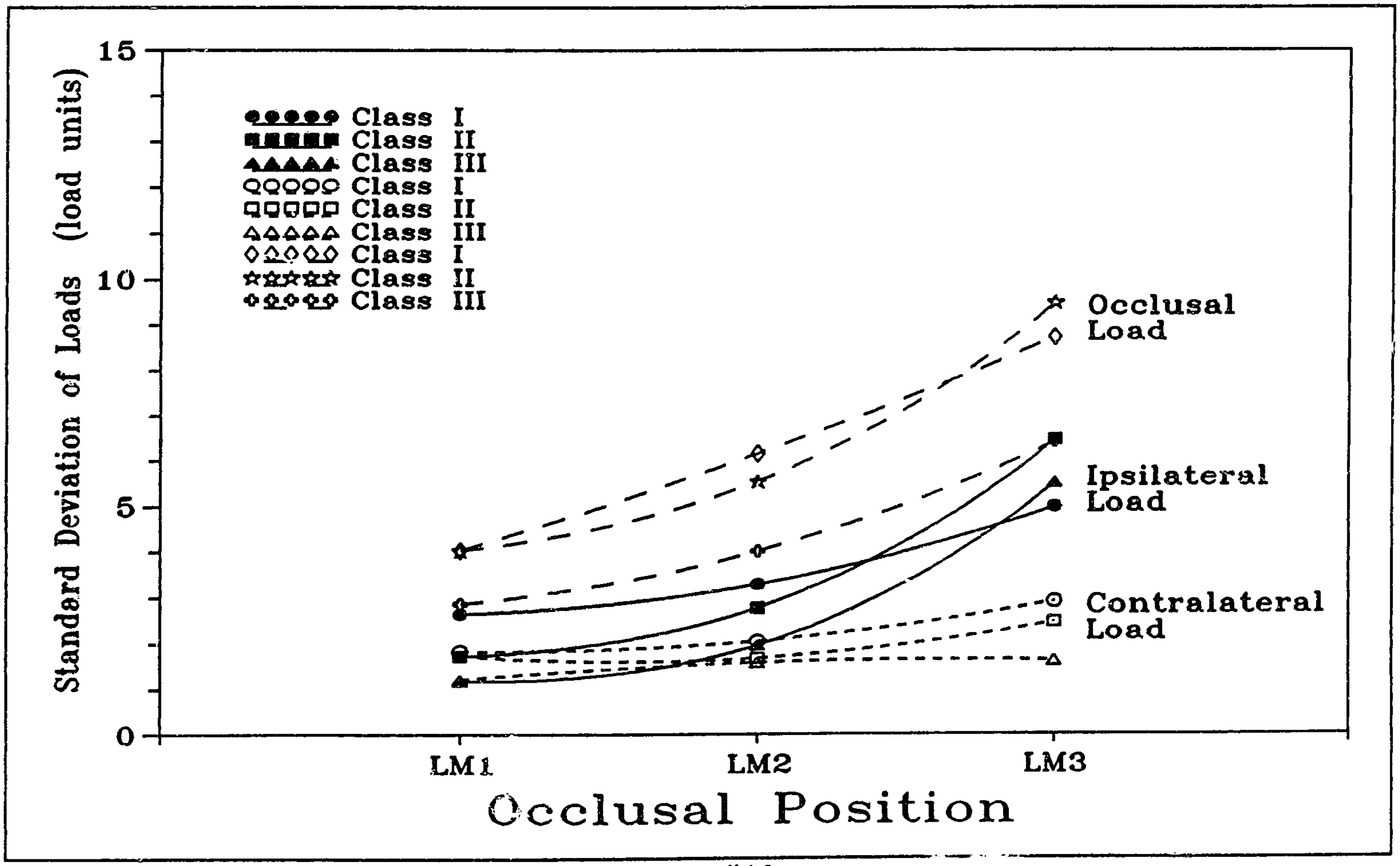



**Figure #4.2**

Standard deviations of resultant occlusal and joint loads with the occlusal load directed perpendicular to the occlusal plane.

Examination on the basis of skeletal class shows the Class IIIs with a consistently higher level of occlusal load (at an 85% confidence level) than the Class Is or IIs, with no significant difference apparent between the Class Is and IIs. The Class IIIs also had the lowest levels of variance, a minimum of 27% lower than either the Class Is and IIs (6.39 versus 8.71 load units for the Class Is at LM2 in the minimum case). In general it will be seen that the Class IIIs have a significantly lower level of variance, and the Class IIs have either the same or a slightly lower level than the Class Is in most cases.

The magnitudes of the resultant contralateral condylar loads decrease only slightly as the occlusal load was moved posteriorly from LM1. The Class IIIs had a consistent significantly lower level (at an 85% confidence level) of loading than the Class Is or IIs which showed no significant difference between them. On the whole the level of variations for the contralateral condylar loads were low, near a level of two load units, however again the Class IIIs showed the lowest level of variation amongst the three. The resultant ipsilateral condylar loads decreased at a slightly faster rate than the contralateral loads with a posterior shift of the occlusal position. The ipsilateral loads were consistently lower than the contralateral loads, with an average contralateral/ipsilateral load ratio of 1.2:1 at LM1, and 1.5:1 at LM2. These levels are notably lower than the 2:1 ratio reported by Hay in 1985 [10] for a single skull, however it should be noted that individual skulls showed variations as large as a 2:1 ratio.

Further examination of the ipsilateral resultant condylar loads showed that there were no statistically significant class differences in magnitude at LM1 or LM2, but the group averages were again order III-I-II from lowest to highest ipsilateral condylar loads. Again the level of variances increased with a posterior shift in occlusal position, with the classes ranking III-II-I from lowest to highest levels of variance. These levels were generally higher than those of the contralateral side.

Another way to examine this information was to normalize all the results to a constant occlusal load level (see Figure #4.3), and is analogous to determining the resultant condylar loads when performing the same task at LM1, LM2 or LM3 for each class. The loads were scaled proportionally to achieve an occlusal magnitude generated by each class at each occlusal position equal to the occlusal load generated by a Class I occluding at LM1. This occlusal load normalization shows more clearly the reductions of ipsilateral and contralateral resultant condylar loads with the posterior shift in occlusal position. Contralateral condylar loads were reduced approximately 25% by shifting the occlusal load from LM1 to LM2, while at the same time a greater (40%) reduction occurred for the ipsilateral loads.

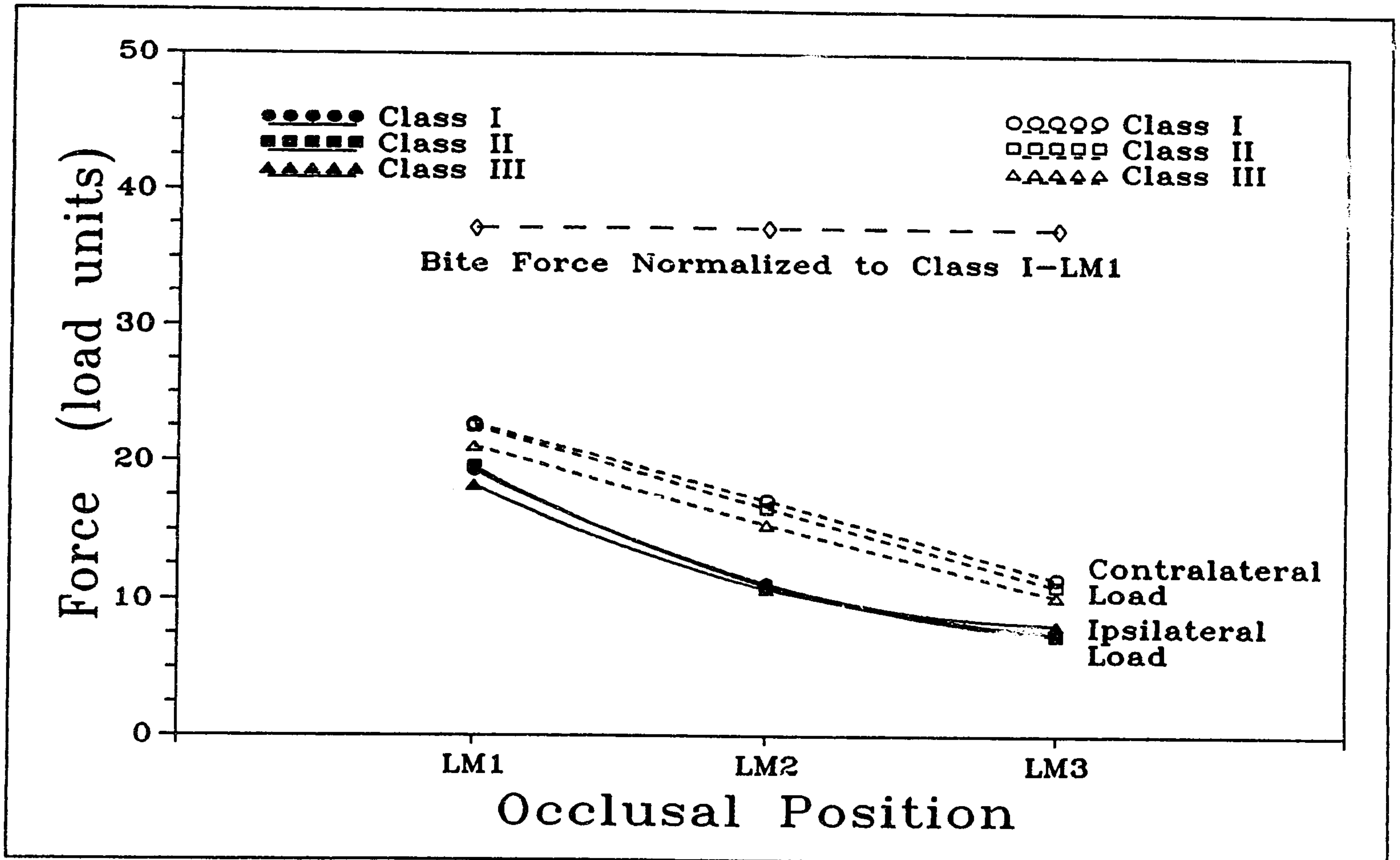



**Figure #4.3**

Normalized condylar loads as a result of setting all occlusal load magnitudes equal to a Class I occluding at LM1, perpendicular to the occlusal plane.

Several conclusions were evident when the occlusal load was orientated perpendicular to the occlusal plane. The contralateral TMJ is more heavily loaded than the ipsilateral TMJ by a ratio of 1.2:1 at the LM1 and 1.5:1 at LM2. The higher levels of variance in the ipsilateral TMJ compared to the contralateral imply that the ipsilateral joint reacts more markedly to the geometric differences. Class IIIs have demonstrated a "better" or more effective loading pattern than the Class Is and IIs by developing a greater occlusal load with simultaneously lower joint loads and overall lower levels of variance. There was no significant difference between the Class Is and IIs.

### 4.2.2 Effects of Occlusal Angle Variation

To investigate the effects of occlusal loads not directed perpendicular to the occlusal plane a series of tests were conducted with non-perpendicular orientations. The occlusal load vector was varied in 5° increments parallel to the sagittal plane for a $\pm 15°$ deviation from perpendicular to the occlusal plane. A positive occlusal angle is interpreted to mean that the upper end of the occlusal force vector was tipped anteriorly, giving the vector a small posteriorly directed component (when applied to the mandible). Conversely a negative occlusal angle means that the superior end of the occlusal force vector was tipped posteriorly, giving the vector a small anteriorly directed component. As before, the results showed similar trends for occlusion on either the left or the right side, as well as for variations in occlusal position, therefore only the results for occlusion at LM1 are discussed here.

The effects of this occlusal angle variation on the ipsilateral resultant condylar loads are shown in Figure #4.4. All three classes displayed similar parabola-like curves. The Class III curve had a condylar loading minima centered approximately at a bite angle of zero degrees. The Class Is and IIs had similar curves with a minima at an occlusal angle of approximately -5°. Of particular interest is the III-I-II order of lowest to highest condylar load levels for all occlusal angles greater than -5°. At angles of zero or greater there is a statistically significant difference at a 65% confidence level between Class III and Class I, and between Class I and II. The implications of these curves is that the ipsilateral condylar loads will increase rapidly as the occlusal angle is shifted away from zero. It would seem logical then to assume that the "better" occlusal angle would be one near perpendicular to the occlusal plane, therefore minimizing the level of the ipsilateral loading.

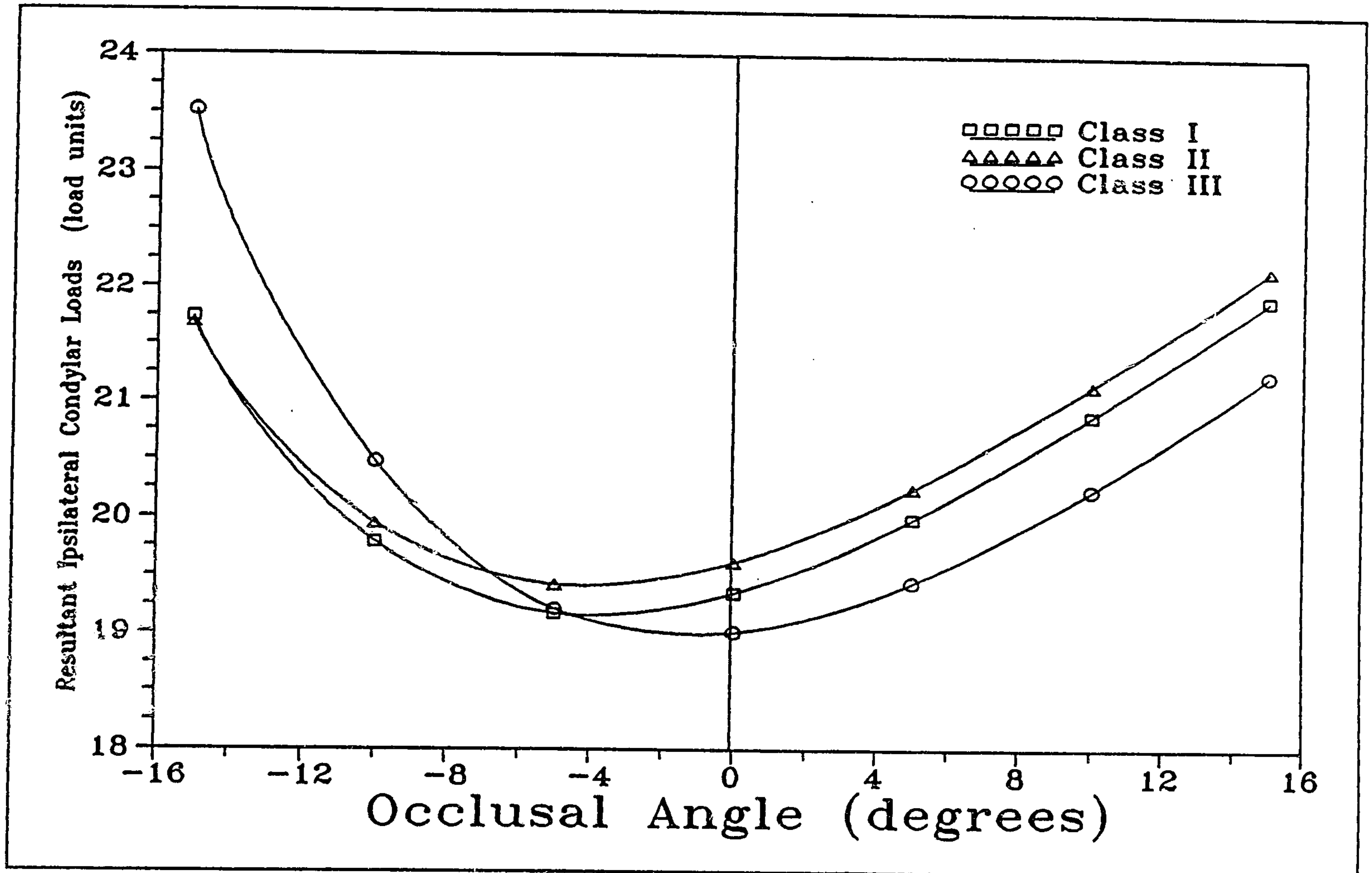



**Figure #4.4**

Variations of the resultant ipsilateral condylar loads with changes of the occlusal load angle, occluding at LM1.

The contralateral resultant condylar loads did not exhibit the same trends as the ipsilateral (see Figure #4.5), rather they followed a monotonically increasing trend as the occlusal angle was varied from negative to positive. They ranked III-II-I from lowest to highest contralateral condylar load at all angles, and the Class IIIs loads were consistently lower than those of the Class Is and IIs at a minimum 85% confidence level. There was no significant difference in the average loads between the Class Is and IIs. Clearly the contralateral (balancing side) TMJ is playing a different role than the ipsilateral TMJ. It is considered that the contralateral TMJ and the occlusal load play an inter-related role during mastication. As the resultant contralateral condylar loads increase (see Figure #4.5) the resultant occlusal loads decrease (see Figure #4.6) as if to maintain some constant combined force level.

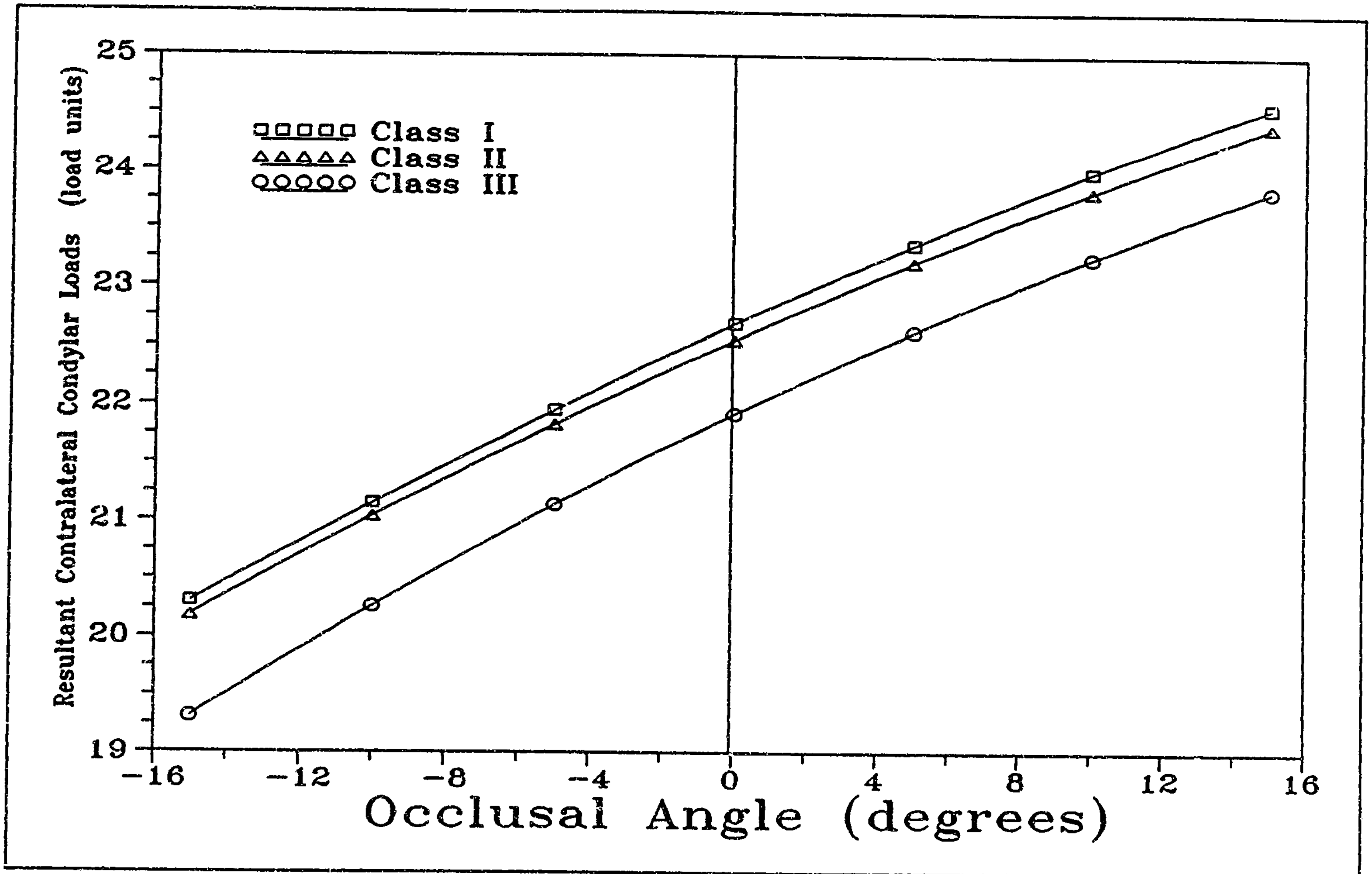



**Figure #4.5**

Variations of the resultant contralateral condylar loads to change in the occlusal load angle, occluding at LM1.

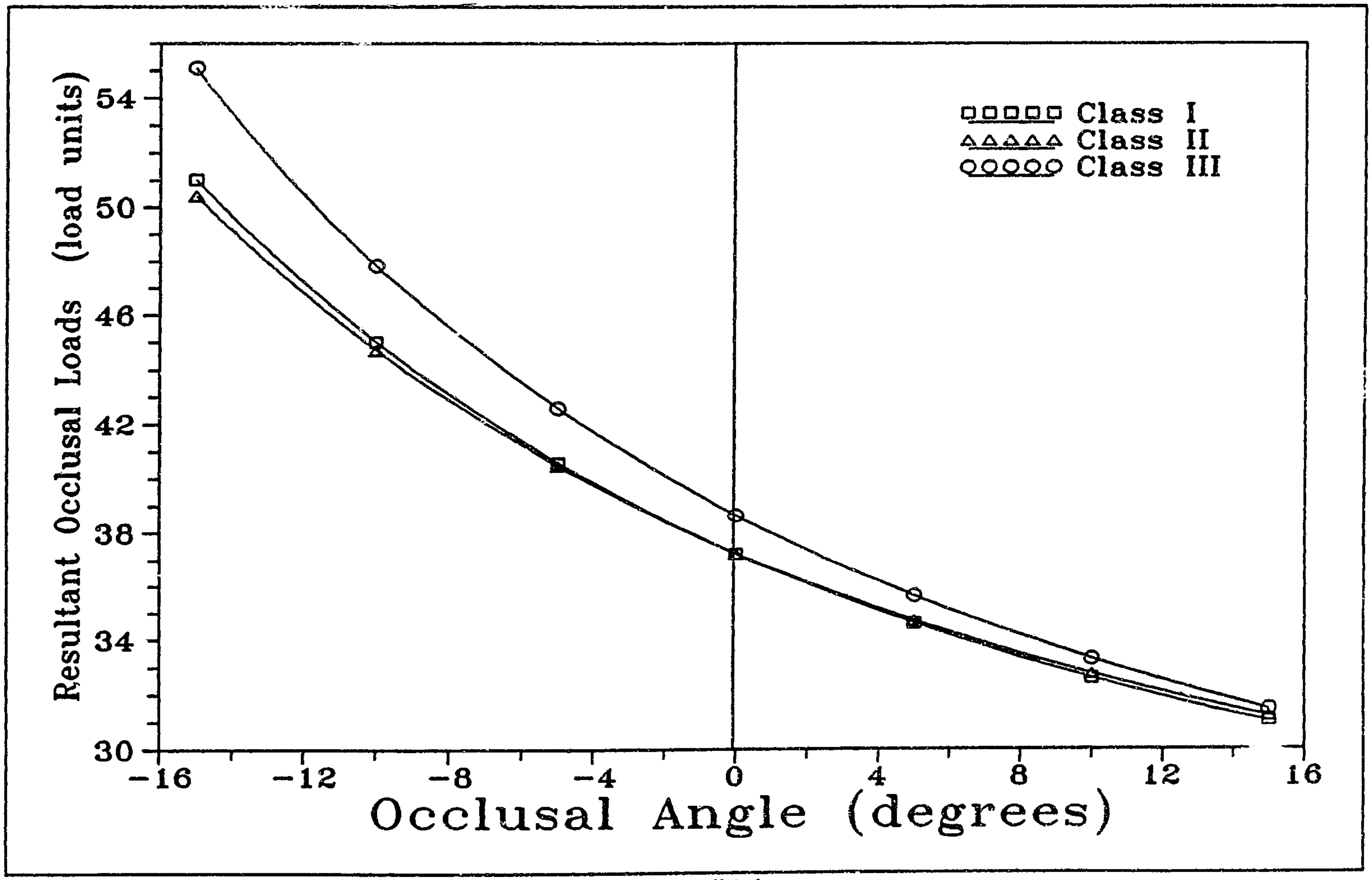



**Figure #4.6**

Variations of the resultant occlusal load magnitudes to changes in the occlusal load angle, occluding at LM1.

The resultant occlusal load magnitudes showed a monotonically decreasing trend as the occlusal angle was increased within the range shown (see Figure #4.6). The occlusal angle variation was limited to ±15° as variations beyond this angle produced unrealistic occlusal force magnitudes. It is believed the body certainly alters the levels of muscle magnitude to provide a more effective muscle combination if such an occlusal force direction was required. In the range of occlusal angle examined the Class IIIs maintained a higher level of generated occlusal load than the Class Is and IIs, but the value of this difference decreased steadily with a positive increase in occlusal angle.

Class IIIs have demonstrated a "better" or more effective loading pattern than the Class Is and IIs by developing greater occlusal loads with simultaneously lower joint loads throughout the occlusal angle range examined. There was no statistically significant difference in the load levels of the Class Is and IIs.

## 4.3 Investigation of Resultant Load Angles

A second and equally significant factor in joint loading patterns is the angle the resultant condylar load forms with respect to the articular eminence it is reacting against. Previously no investigations of this type have seriously examined the angular agreement between the reaction forces in the TMJ and their load bearing surfaces. In 1974 Barbenel noted the importance of the resultant load angles in the parasagittal plane in his numerical investigation of TMJ loading [3]. As stated earlier recent works by Blaustein and Scapino [4, 31] indicated that the region where the retro discal tissues and the posterior of the articular disc join had remodelled due to abnormal loading in patients with TMJ dysfunction. Although they noted that these findings must be limited to their study due to the minimal sample size (9 discs) they provide some evidence that the condylar loads in these individuals may have been misdirected posterosuperiorly onto the posterior band of the articular disc and the retro discal tissues.

It is postulated that an "ideal" TMJ loading relationship is one where the reaction load required at the condyle would be aligned perpendicularly to the load bearing surface of the corresponding articular eminence. It is believed that a purely compressive load would be the healthiest for the loaded tissues interposed between these surfaces [41]. Further it is expected that evidence will show significantly higher levels of posterosuperiorly directed shear loads, created as a result of load misdirection, in patients with higher levels of TMJ dysfunction.

Situations in the current model where anteroinferiorly directed shear loads are predicted are believed to be unrealistic. Such a loading situation during maximum occlusal load is believed to be counter productive as the shear component of the resultant condylar load would not be directed to oppose the occlusal load.

The functional conditions of the TM joint requires that small amounts of shear forces are acceptable. This is necessary to accommodate the translation of the condyle and articular disc along the articular eminence during opening and closing motions. High levels of shear load, developed during forceful occlusal contact by misdirected condylar loads, may exceed the loaded tissues adaptive abilities and predispose these articular tissues to damage.

The level of resultant load misdirection was evaluated by examining the angular misalignment of the resultant ipsilateral condylar loads on a class basis at different occlusal load angles. Only the ipsilateral side can be discussed as information about the slope of the articular eminence where the contralateral loads are incident was not recorded. This is unfortunate since the contralateral loads are generally more stable and would likely exhibit less variable trends than the ipsilateral side.

As stated earlier an "ideal" loading pattern has a resultant condylar load perpendicular to the loading surface of the articular eminence, or a 0° misalignment (see Figure #4.7). Loads with a posterosuperiorly directed shear load component are represented as a positive angular misalignment. Conversely, loads with an anteroinferiorly directed shear load component are represented as a negative angular misalignment.

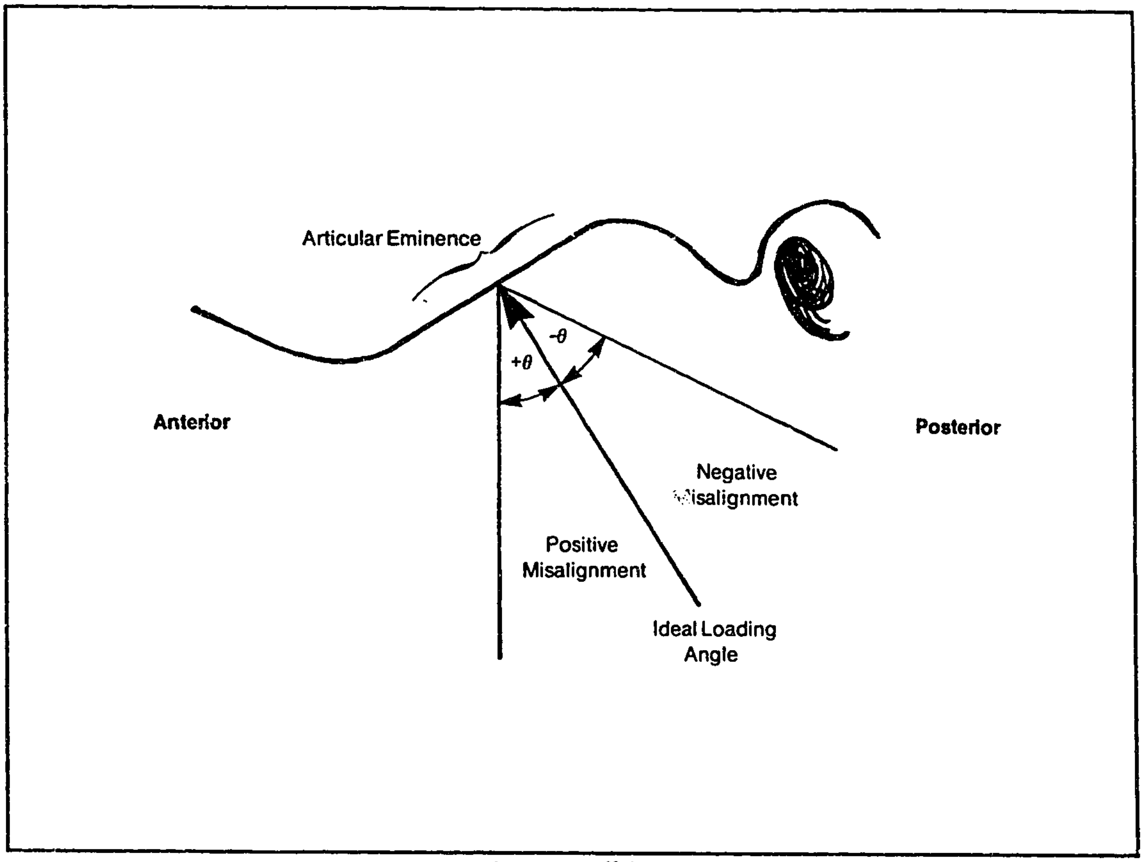



**Figure #4.7**
Representation of the "ideal" loading pattern

### 4.3.1 Description of Statistical Methods

In order to first evaluate the correlation between skeletal morphology and possible misdirection of the resultant condylar loads, Figure #4.8 compares the ANB angle to the difference between the computed resultant load direction and the perpendicular to the measured articular eminence slope. This plot considers the correlation for the ipsilateral condyle assuming the occlusal load is applied perpendicular to the

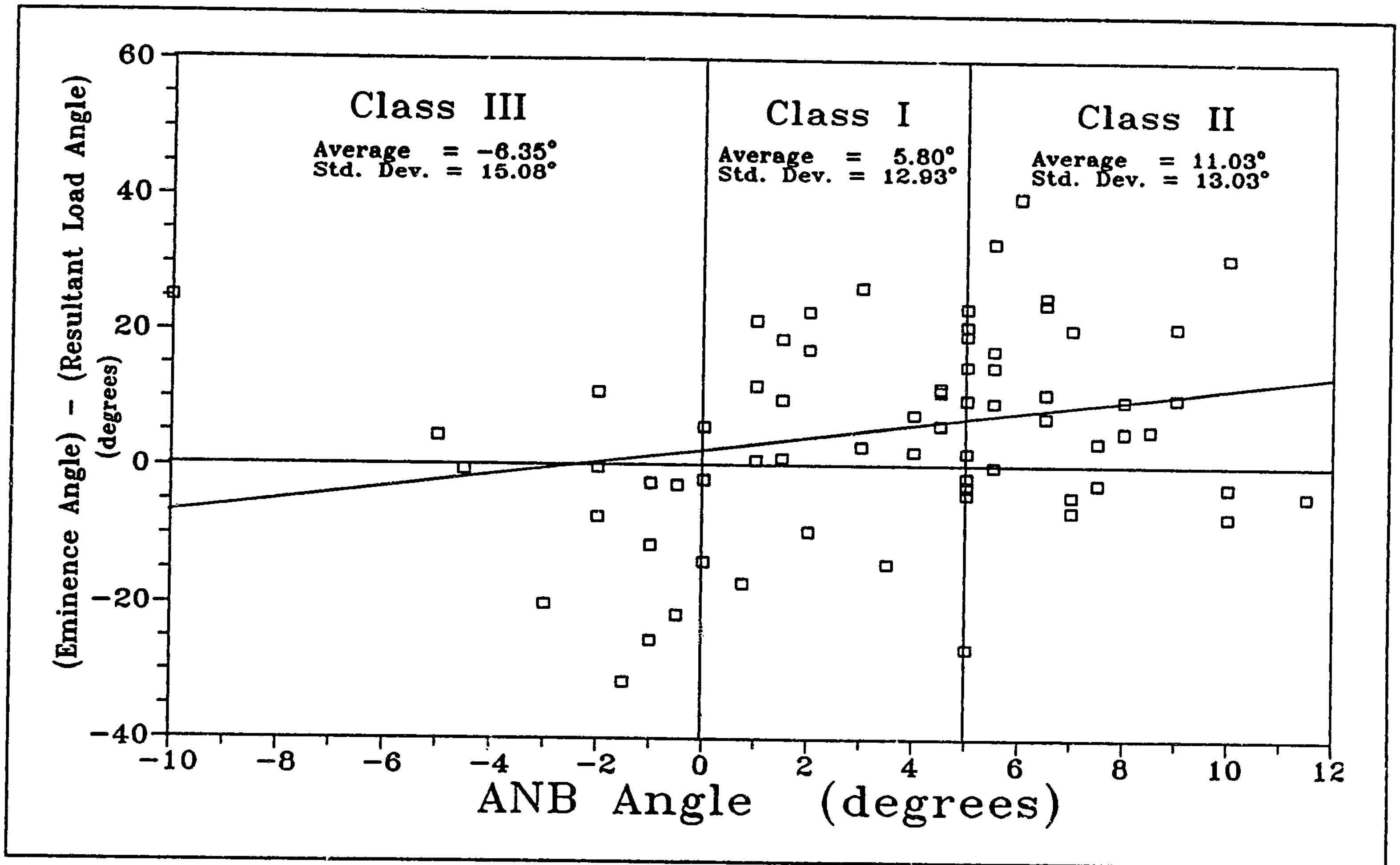



**Figure #4.8**

Scattergram showing the individual differences in the computed resultant load and measured eminence angle for the entire sample.

**Table #4.1**
Statistical data related to Figure #4.4

**General Information:**

| | | |
|---|---|---|
| $\Theta_1$ = 5[illegible] | $\Theta_2 = 11.03°$ | $\Theta_3 = -6.34°$ |
| $n_1$ = [illegible] | $n_2 = 24$ | $n_3 = 13$ |

Po[illegible]d Deviation = 13.71°

**Confidence In[illegible]**

$+0.98° \leq \Theta_1 \leq +10.62°$ at a 95% confidence level.

$+5.54° \leq \Theta_2 \leq +16.52°$ at a 95% confidence level.

$-13.79° \leq \Theta_3 \leq +1.11°$ at a 95% confidence level.

**Test for Difference of Means:**

| Significance Level | 99% | 95% | 90% |
|---|---|---|---|
| is $\Theta_2 > \Theta_1$? | No | No | Yes |
| is $\Theta_1 > \Theta_3$? | Yes | Yes | Yes |
| is $\Theta_2 > \Theta_3$? | Yes | Yes | Yes |

**Quantitative Difference of Means:**

| Significance Level | 99% | 95% | 90% |
|---|---|---|---|
| $\Theta_2 > \Theta_1$ by: | n/a | n/a | 0.45° |
| $\Theta_1 > \Theta_3$ by: | 1.60° | 4.69° | 6.33° |
| $\Theta_2 > \Theta_3$ by: | 6.39° | 9.60° | 11.32° |

occlusal plane at the LM1. As before, the occlusion on the right side produced trends nearly identical to the left side, and therefore only the left side results will be discussed here. Because of the scatter shown, statistical analysis of this data included several tests to evaluate differences between the three skeletal classes. The results of these are shown in Table #4.1. The averages ($\Theta_1$, $\Theta_2$ and $\Theta_3$) and standard deviations were used first in an F-test to determine if at least two of the class averages are different. In situations in which the F-test showed that at least two of the means were different then a confidence interval was computed for each of the means at a 95% confidence level. A number of paired T-tests were done to determine a confidence level for differences in the means for the three possible combinations. In this circumstance, all three pairs of means were different at a 90% confidence level, but only two pairs were at 95% and 99% ($\Theta_1 > \Theta_3$ and $\Theta_2 > \Theta_3$). Finally another series of paired T-tests were done to determine the quantitative difference between the pairs of means at a particular confidence level as a measure of their difference. These tests showed that the Class IIs had a 0.45° greater mean than the Class Is at a 90% level.

A linear regression of the scatterplot is also shown, but its value as a predictor is limited by the relatively high levels of variance. This study was primarily interested in the T-test for significant difference of means in small samples.

### 4.3.2 Loads Perpendicular to the Occlusal Plane

The effect of occlusal position on the average TMJ loading pattern alignments are shown in Figure #4.9 for the three classes at the three left molars. Looking closer at the averages for LM1 it is seen that Class IIIs have a slightly anteroinferiorly directed alignment (-6.35°), and that the Class Is have a nearly equal (+5.80°), but oppositely directed, posterosuperiorly directed resultant condylar load. Lastly the Class IIs exhibit the highest level of posterosuperiorly directed joint load misalignment (+11.03°).

The results demonstrated at LM2 and LM3 are at first misleading. Upon initial observation it appears that the direction of the resultant joint loads shifts anteroinferiorly as the occlusal load is moved distally. As discussed earlier it is felt that this would be inefficient and unrealistic. An explanation for this is found in an understanding of the curve of Spee [17] and the way the occlusal plane is calculated in this model.

The occlusal load was orientated perpendicular to the occlusal plane in Figure #4.9. The calculation of the occlusal plane in the current model relies on the first molars (see earlier discussion in section 3.4) and therefore is parallel to their crown. Aligning the occlusal load perpendicular to the occlusal plane is therefore roughly equivalent to aligning it perpendicular to the crown of the first molar (which is desired). At the second molar, having the occlusal load perpendicular to the occlusal plane does not produce an occlusal load perpendicular to the tooth's crown. This is due to the curve of Spee.

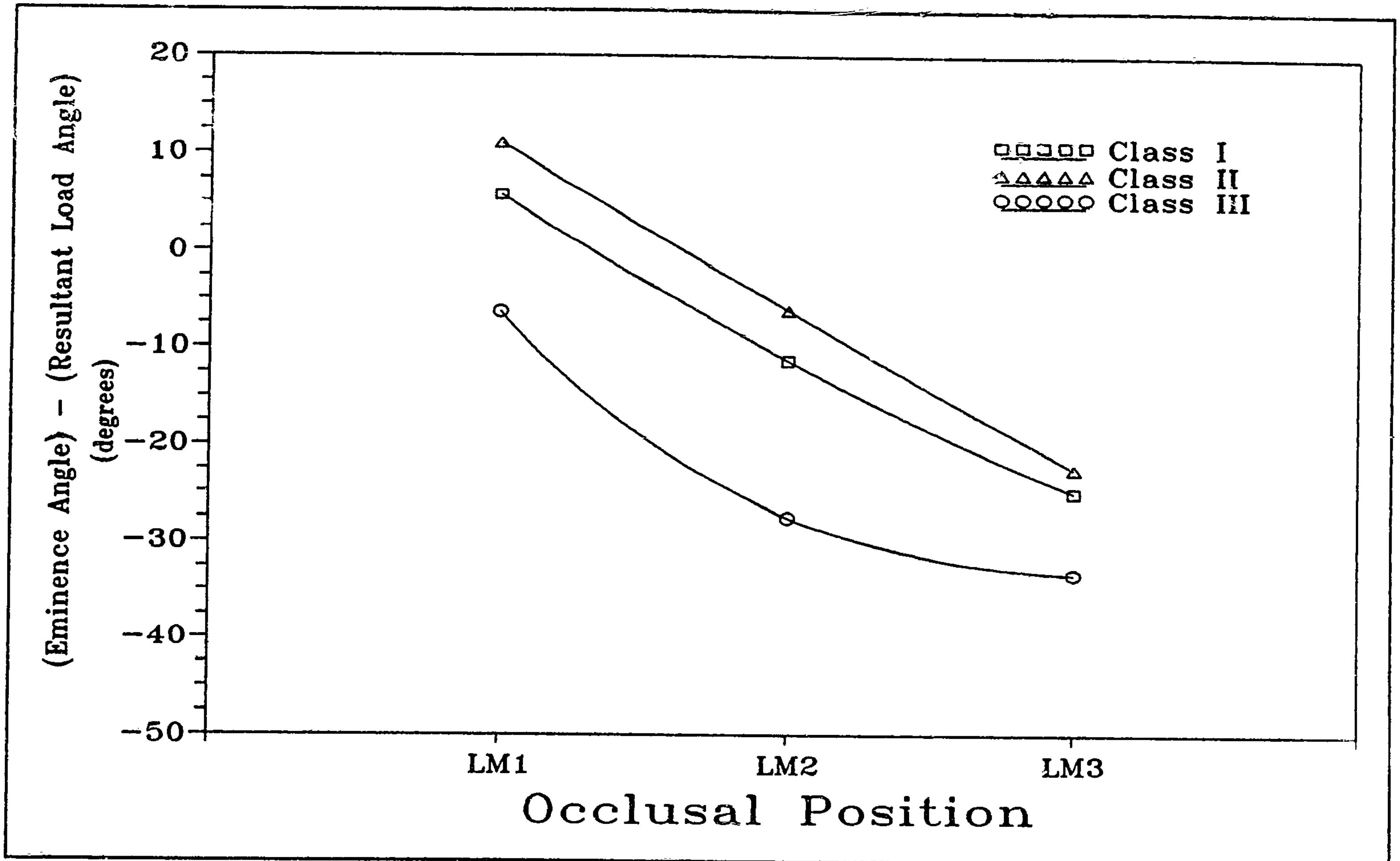



**Figure #4.9**

Agreement of average measured eminence and computed resultant load angles with the occlusal load perpendicular to the occlusal plane.

It is usual that the mandibular second molar is tipped slightly more mesially than the first molar with relation to the occlusal plane. An examination of the lateral cephalometric films confirmed this. By applying the occlusal load perpendicular to the occlusal plane at the second molar rather than the crown, the applied load is misaligned approximately -5°.

The effects of occlusal position on the average loading pattern alignments with the occlusal load tipped +5° from perpendicular to the occlusal plane is shown in Figure #4.10. This +5° tip of the occlusal load is a rough approximation to an application of the occlusal load perpendicular to the crown of the left second molar. The levels of resultant joint load misalignment are now much better and in close agreement with the levels for LM1 in Figure #4.9. Again the Class IIIs exhibit a slight anteroinferior directed load (-9.72°) with the Class Is displaying a posterosuperiorly inclined load (+4.67°). As before the Class IIs had the highest level of posterosuperiorly directed load at 9.16°.

This is very interesting in fact as it demonstrates the direct relationship between the curve of Spee and TMJ loading patterns. In order to maintain near "ideal" TM joint loading patterns as the occlusal load moves distally, the occluding surface of the mandibular dentition must tip mesially.

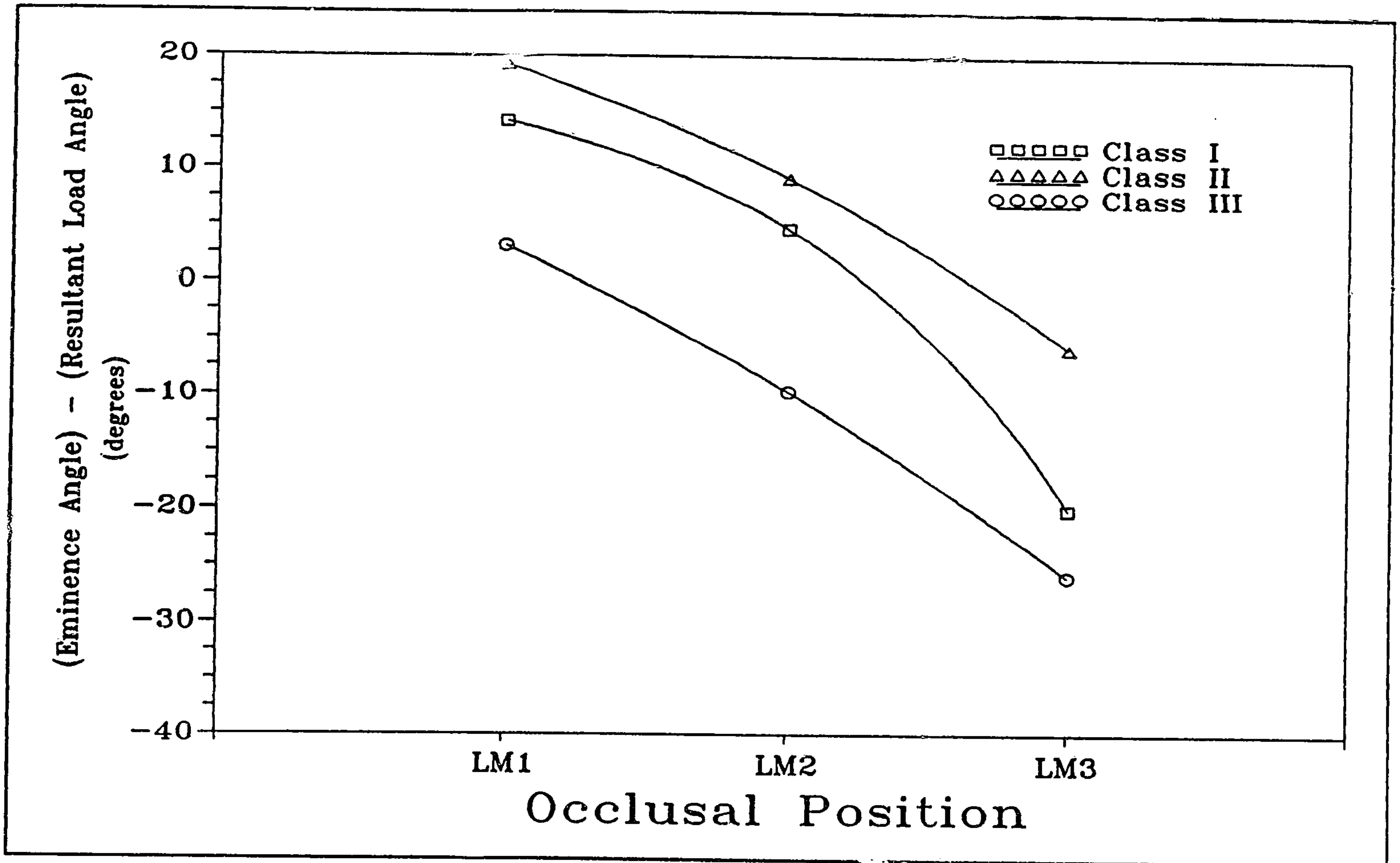



**Figure #4.10**
Agreement of the average measured eminence and computed resultant load angles with the occlusal load tipped +5° from perpendicular to the occlusal plane.

### 4.3.3 Effects of Occlusal Angle Variation

To investigate the effects that variations in the occlusal angle have on the loading pattern alignment the occlusal load angle was again varied ±15° in 5° increments. Figure #4.11 shows the resultant ipsilateral loading misalignment at LM1 (trends were similar at LM2 and LM3). The plot shows the load misalignment monotonically increasing at nearly the same rate for all classes as the occlusal angle is increased. As described above, it is considered that values significantly below 0° are believed to be unrealistic. At all positive occlusal angles the Class IIIs showed the most "ideal" loading arrangement with consistently lower levels of misalignment than the Class Is and IIs (a minimum of 5.02° below the Class I averages at a 90% significance level). The Class IIs showed the highest levels of misalignment at all positive occlusal angles, a minimum of 1.23° higher than the Class Is at 90% significance.

Figure #4.11 clearly shows that the Class I and, even more so that, Class III skeletal morphologies have a distinct advantage over a Class II by having lower levels of posterosuperiorly directed shear load than the Class II. An application of the information gained from this plot would suggest that if the occlusal plane for a particular craniofacial morphology is altered in a way such that the occlusal loads act in a negatively shifted direction it may be possible to reduce the level of angular misalignment within a joint, improving the TMJ loading pattern. For example, if the occlusal plane for the average Class IIs was altered by -5° its average level of misalignment will drop from +11.03° to a nearly "ideal" level of +1.5°.

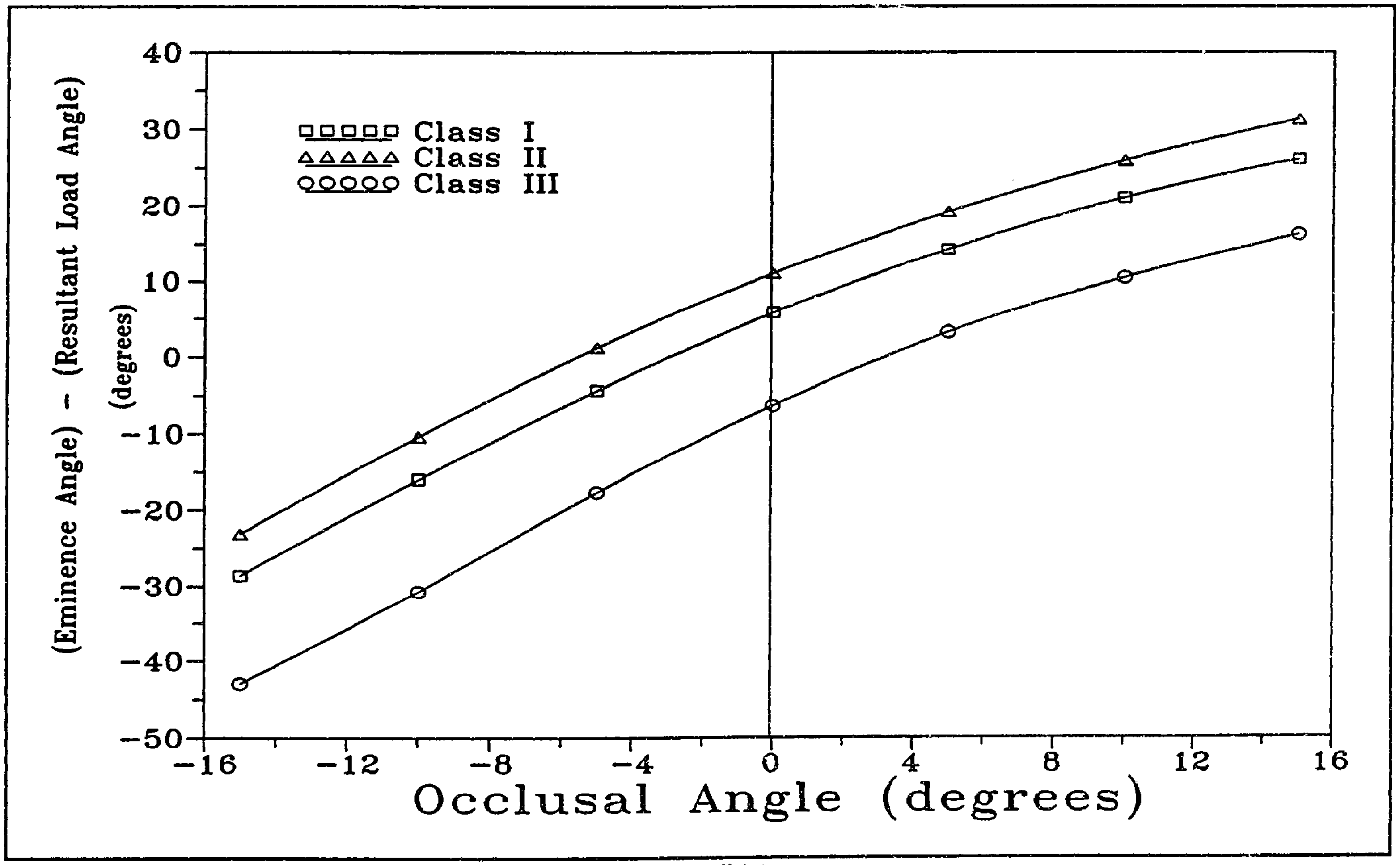



**Figure #4.11**

Variations of the ipsilateral agreement between the eminence and resultant load angles to changes in the occlusal load angle, occluding at LM1.

# 5.0 Conclusions

## 5.1 Final Discussion

A three dimensional numerical model was developed to compare the joint loading patterns of 70 human skulls of varying morphologies, namely Angle's skeletal Class I, II and III. The model utilizes a pragmatic approach towards the numerical solution, using specified Type II muscle magnitude data, measured spatial information and a specified occlusal load position and direction. The models results are in agreement with previous studies.

A two part investigation was conducted consisting of: examination of the resultant ipsilateral and contralateral condylar loads, the occlusal load magnitude, and the agreement of the ipsilateral resultant condylar load and articular eminence angles for static unilateral occlusion perpendicular to the occlusal plane at different occlusal position (LM1, LM2 and LM3); further, examination of the resultant ipsilateral and contralateral condylar loads, the occlusal load magnitude, and the agreement of the ipsilateral resultant condylar load and articular eminence angles for stationary unilateral occlusion at LM1 while the occlusal load direction was varied ±15° in the sagittal plane from perpendicular to the occlusal plane.

The primary conclusions are as follows:

- Angle's Class IIs exhibit moderately higher levels of resultant TM joint loads while producing lower levels of occlusal load compared to Class IIIs.
- The Class IIs joint loads display higher levels of variability than Class IIIs.

- Most significantly, Class II morphologies function with a consistently higher level of resultant TMJ load misalignment than either Class I or Class III.

These three factors combine to produce the most unfavorable TMJ loading pattern. Larger, less stable, joint load magnitudes combined with the greatest component of posterosuperiorly directed shear load produce the highest level of unhealthy shear force on the joint tissues and load the posterior band of the articular disc. This combination of factors may be a significant factor towards explaining the higher levels of TMJ dysfunction in the Class II population.

In contrast Angle's Class IIIs show significantly lower levels of resultant condylar loads while developing the highest levels of occlusal load. They exhibit the lowest level of load variability, and most importantly the alignment of their joint loads is nearest to "ideal" of the three morphologies. This combination of conditions should form the most favorable (healthiest) TMJ loading pattern, producing little or no shear force on the joint tissues.

## 5.2 Areas for Continued Evaluation

Although this study represents a fairly advanced numerical modelling technique, there are several areas that merit further attention. A more detailed measurement of the ipsilateral and contralateral eminence angles (while in their respective loaded positions) would provide a better level of information to evaluate the directional agreement of TMJ loads. Incorporation of information on jaw tracking to better locate the mandible with relation to the maxilla would lead to more accurate results. Also to this end, a further refinement of the applied muscle magnitudes, in particular development of 3 sets of magnitudes which accurately characterize the musculature of each morphology, is necessary. Finally, advancement of the model to a clinical level for *in vivo* evaluation of a subjects loading pattern may provide useful diagnostic information. This may be possible by utilizing the Ct number information from Computer Axial Tomography (CAT scan) or computer assisted three dimensional reconstruction of spatial information from two dimensional cephalometric projections.

# References

1) - Baragar, F.A., Osborn J.W. "Efficiency as a predictor of Human Jaw Design in the sagittal plane" JOURNAL OF BIOMECHANICS Vol. 20(1987):pp. 447-457

2) - Barbenel, J.C. "The Biomechanics of the Temporomandibular Joint: a Theoretical Study" JOURNAL OF BIOMECHANICS Vol. 5(1972):pp. 251-256

3) - Barbenel, J.C. "The Mechanics of the Temporomandibular Joint: a Theoretical and Electromyographical Study" JOURNAL OF ORAL REHABILITATION Vol. 1(1974):pp. 19-27

4) - Blaustein D.I.; Scapino R.P. "Remodelling of the Temporomandibular Joint Disk and Posterior Attachment in Disc Displacement Specimens in Relation to Glycosaminoglycan Content" PLASTIC AND RECONSTRUCTIVE SURGERY Vol. 78(1986):pp. 756-764

5) - Carlsoo, S. "Nervous Coordination and Mechanical Function of the Mandibular Elevators" ACTA ODONTOL SCAND. Vol. 10(1952):Suppl. 11

6) - Faulkner, M.G.; Hatcher, D.C.; Hay, A.S. "A Three-Dimensional Investigation of Temporomandibular Joint Loading" JOURNAL OF BIOMECHANICS Vol. 20(1987):pp. 997-1002

7) - Gingerich, P. "The Human Mandible: Lever, Link, or Both?" AMERICAN JOURNAL OF PHYSICAL ANTHROPOLOGY Vol. 51(1979):pp. 135-138

8) - Grant, P. "Biomechanical Significance of the Instantaneous Center of Rotation: The Human Temporomandibular Joint" JOURNAL OF BIOMECHANICS Vol. 6(1973):pp. 109-113

9) - Gysi, A. "Studies on the Leverage Problems of the Mandible" THE DENTAL DIGEST Vol. 27(1921):pp. 74-84, 144-150, 203-208

10) - Hay, A.S. "A Mechanical Analysis of the Temporomandibular Joint" M.Sc. Thesis, University of Alberta, 1985

11) - Helkimo, M. "Studies on Function and Dysfunction of the Masticatory System" SWEDISH DENTAL JOURNAL Vol. 67(1974):pp. 101-121

12) - Hylander, W. "The Human Mandible: Lever or Link" AMERICAN JOURNAL OF PHYSICAL ANTHROPOLOGY Vol. 43(1975):pp. 227-242

13) - Jurmain, R. "Stress and Etiology of Osteoarthritis" AMERICAN JOURNAL OF PHYSICAL ANTHROPOLOGY Vol. 46(1978):pp. 353-366

14) - Kawazoe, Y.; Kotani, H.; Hamada, T. "Relation Between Integrated Electromyographic Activity and Biting Force During Voluntary Isometric Contration in Human Masticatory Muscles" JOURNAL OF DENTAL RESEARCH Vol. 58(1979):pp. 1440-1449

15) - McNamara, J. Jr. "The Independent Functions of the Two Heads of the Lateral Pterygoid Muscle" AMERICAN JOURNAL OF ANATOMY Vol. 138(1970):pp. 197-206

16) - Meikle, M. "Remodelling" in "The TMJ: A Biological Basis for Clinical Practice." Sarnat, B. & Laskin, D., Eds. Charles C. Thomas, Springfield, Illinois, U.S.A. pp. 205-226

17) - Mohl, N.D.; Zarb, G.A.; Carlsson, G.E.; Rugh, J.D. "A Textbook of Occlusion" QUINTESSANCE PUBLISHING COMPANY, INC. Chicago, Illinois, U.S.A. (1988)

18) - Moller, E. "Computer Analysis of Electromyographic Data in Clinical Studies of Oral Function" SCAND. JOURNAL OF DENTAL RESEARCH Vol. 78(1970):pp. 411-416

19) - Mushimoto, E.; Mitani, H. "Bilateral Coordination Pattern of Masticatory Muscle Activities During Chewing in Normal Subjects" THE JOURNAL OF PROSTHETIC DENTISTRY Vol. 48(1982):pp. 191-197

20) - Nickel, J.C.; McLachlan, K.R.; Smith, D.M. "Eminence Development of the Postnatal Human Temporomandibular Joint" JOURNAL OF DENTAL RESEARCH Vol. 67(1988):pp. 896-902

21) - Nickel, J.C.; McLachlan, K.R.; Smith, D.M. "A Theoretical Model of Loading and Eminence Development of the Postnatal Human Temporomandibular Joint" JOURNAL OF DENTAL RESEARCH Vol. 67(1988):pp. 903-910

22) - Ogus, H. "Degenerative Disease of the Temporomandibular Joint in

Young Persons" BRITISH JOURNAL OF ORAL SURGERY Vol. 17(1979):pp. 17-26

23) - Osborn, J.W.; Baragar, F.A. "Predicted Pattern of Human Muscle Activity During Clenching Derived from a Computer Assisted Model: Symmetric Vertical Bite Forces" JOURNAL OF BIOMECHANICS Vol. 18(1985):pp. 599-612

24) - Pruim, G.; De Jongh, H.; Ten Bosch, J. "Jaw Muscle EMG-Activity and Static Loading of the Mandible" JOURNAL OF BIOMECHANICS Vol. 11(1978):pp. 389-395

25) - Radin, E.; Paul, I.; Rose. "Role of Mechanical Factors in Pathogenesis of Primary Osteoarthritis" THE LANCET (March 4, 1972):pp. 519-521

26) - Ralph, J.; Caputo, A. "Analysis of Stress Patterns in the Human Mandible" JOURNAL OF DENTAL RESEARCH Vol. 54(1975):pp. 814-821

27) - Ralston, H. "Uses and Limitations of Electromyography in the Quantitative Study of Skeletal Muscle Function" AMERICAN JOURNAL OF ORTHODONTICS Vol. 47(1961):pp. 521-530

28) - Riedel, R.A. "The relation of maxillary structures to cranium in malocclusion and in normal occlusion" JOURNAL OF ANGLE ORTHODONTICS Vol. 22(1952):pp. 142-145

29) - Riolo, M.L.; Brandt, D.; TenHave, T.R. "Associations between occlusal characteristics and signs and symptoms of TMJ dysfunction in children and young adults" AMERICAN JOURNAL OF ORTHODONTICS Vol. 92(1987):pp. 467-477

30) - Robinson, M. "The Temporomandibular Joint: Theory of Reflex Controlled Non-Lever Action of the Mandible" JOURNAL OF AMERICAN DENTAL ASSOCIATION Vol. 33(1946):pp. 1260-1271

31) - Scapino, R.P. "Histopathology associated with malposition of the human temporomandibular joint disc" JOURNAL OF ORAL SURGERY Vol. 55(1983):pp. 382-397

32) - Scott, J.H. "A contribution to the study of the mandibular joint function" BRITISH DENTAL JOURNAL Vol. 94(1955):pp. 345

33) - Smith D.M.; McLachlan K.R., McCall W.D. Jr. "A Numerical Model of Temporomandibular Joint Loading" JOURNAL OF DENTAL RESEARCH Vol. 65(1986):pp. 1046-1053

34) - Standlee, J.; Caputo, A.; Ralph, J. "The Condyle as a Stress-Distributing Component of the Temporomandibular Joint" JOURNAL OF ORAL REHABILITATION Vol. 8(1981):pp. 391-400

35) - Standlee, J.; Caputo, A.; Ralph, J. "Stress Trajectories Within the Mandible Under Occlusal Loads" JOURNAL OF DENTAL RESEARCH Vol. 56(1977):pp. 1297-1302

36) - Standlee, J.; Caputo, A.; Ralph, J. "Stress Transfer to the Mandible During Guidance and Group Function Eccentric Movements" JOURNAL OF PROSTHETIC DENTISTRY Vol. 41(1979):pp. 35-39

37) - Steinhardt, G. "Anatomy and function of the temporomandibular joint" INTERNATIONAL DENTAL JOURNAL Vol. 8(1958):pp. 155

38) - Tattersall, I. "Cranial Anatomy of Archaeolemurinae (Lemuroidea Primates)" ANTHROP. PAP. AMER. MUS. NAT. HIST. Vol. 52(1973):pp. 1-110

39) - Throckmorton, G.S.; Throckmorton, L.S. "Quantitative Calculations of Temporomandibular Joint Reaction Forces - I. The Importance of the Magnitude of the Jaw Muscle Forces" JOURNAL OF BIOMECHANICS Vol. 18(1985):pp. 445-452

40) - Throckmorton, G.S.; Throckmorton, L.S. "Quantitative Calculations of Temporomandibular Joint Reaction Forces - II. The Importance of the Direction of the Jaw Muscle Forces" JOURNAL OF BIOMECHANICS Vol. 18(1985):pp. 453-461

41) - Townsend, P.R.; Raux, P.; Rose, R.M.; Miegel, R.E.; Radin, E.L. "The Distribution and Anisotropy of the Stiffness of Cancellous Bone in the Human Patella" JOURNAL OF BIOMECHANICS Vol.8(1975):pp. 363-368

42) - Wilson, G. "The Anatomy and Physics of the Temporomandibular Joint" JOURNAL OF THE NATIONAL DENTAL ASSOCIATION Vol. 7(1920):pp. 414-420

# Appendix #1: Digitizing Source Code

```
DECLARE SUB COMMENTING (COMMENTS$(), COUNT!)
DECLARE SUB HEADER (TITLE$, ATTRIBUTE!)
DECLARE SUB CLEARBUFFER ()
DECLARE SUB CENTER (STATEMENT$, LYNE!, WRAPTEXT!)
DECLARE SUB GETNAMES (FULNAM$(), CCODE$())

'*********************************************************************
'*
'*     This program is developed for use with the Digitizer and an IBM
'*     PC with a serial communications port. It is written in Quick-
'*     Basic Ver 4.0. It's sole purpose is to record a digital image of all
'*     muscle, molar and joint orientation of a human skull for use in
'*     the research of Temporomandibular Joint loading.
'*
'*                                          Version 2.4
'*                                          March 1989
'*                                          by Steve McEvoy
'*
'*          Any reproduction or unauthorized use of these materials
'*     would be unethical, dishonest, and unprofessional. It is also
'*     illegal.  Please don't test [illegible] friendship.
'*
'********************************* [illegible] ************************************

'****
'*     DIMENSIONING ALL VARIABLES
'****

       CLEAR , , 12000
5      DIM FULNAM$(120), CCODE$(120), A$(120)
       DIM X(120), Y(120), Z(120), COMMENTS$(120)
       NUMNAMES = 112                       'NUMBER OF LOCATIONS
       REDOFLAG = 0                         'FLAG TO REPEAT CURRENT RUN
       OFFSET.X = 0                         'X dir ZERO REFERENCE
       OFFSET.Y = 0                         'Y dir ZERO REFERENCE
       F = 0                                'CURSOR PAD IDENT. NUMBER
       CONST WHITE = 15
       CONST BLACK = 0
       CONST BLUE = 1
       CONST RED = 4
       CONST YELLOW = 14
       COLOR WHITE, BLUE, BLACK
       CLS

'****
'*     SETTING OUTPUT FORMATS
'****

       FORMAT$ = "& +##.### & +##.### & +##.###"
       DISKFORM$ = "& ### & +##.### & +##.### & +##.### & &"

'****
'*     TURN OFF KEYS 6 TO 10
'****

       FOR J = 6 [illegible] 10
         KEY J, "
         [illegible] J

'****
'*     [illegible]IONS TO THE DIGITIZER
'****

       [illegible] RS,CS0,DS0,CD0" FOR RANDOM AS #1    'OPEN COMPORT #1
       [illegible]                                      'SET PAD AUTOBAUD
       F[illegible] = TIMER
       WH[illegible] WEND                               'RESET PAD
       PR[illegible]                                    'SET RESOLUTION 1000 LPI & POINT MODE

'****
'*     [illegible]UT DATA STORAGE FILENAME
'****

       HEADER "Output File Definition", YELLOW
       CENTER "DATA WILL BE STORED IN THE HARD DRIVE AND ON DISK IN DRIVE A:", 10, 0
       CENTER "FILENAME FOR DATA STORAGE WILL BE AUTOMATICALLY MADE", 13, 0
       COLOR WHITE
       LOCATE 15, 20
       INPUT "PLEASE ENTER THE SKULL NUMBER (i.e. 1):", FILENUM$
       FILE$ = "TMJ" + FILENUM$ + ".TMP"
```

```
          HEADER "Opening the Output Files", YELLOW
          CENTER "PLACE YOUR FORMATTED DATA DISKETTE IN DRIVE A:", 11, 0
          CENTER "PRESS ANY KEY WHEN READY", 12, 0
          DO WHILE INKEY$ = "": LOOP
          OPEN "\QB\TMJ\STORAGE\" + FILE$ FOR OUTPUT AS #2          'OPEN COM TO DRIVE C
          OPEN "A:" + FILE$ FOR OUTPUT AS #3                        'OPEN COM TO DRIVE A
          FOR I = 1 TO NUMNAMES
            READ FULNAM$(I), CCODES$(I)
          NEXT I

'****
'*        ENTERING SKULL PARTICULARS
'****

          HEADER "Skull Particulars", YELLOW
          COLOR WHITE
50        LOCATE 13, 30
          INPUT "SKULL CLASS (1,2 or 3): ", CLASS
          IF CLASS <> 1 AND CLASS <> 2 AND CLASS <> 3 GOTO 50

'****
'*        ESTABLISHING A ZERO REFERENCE
'****

55        HEADER "Locating the ZERO position", YELLOW
          LOCATE 10, 23
          PRINT "STEP #1:    Locate the x,y,z origin... "
          LOCATE 12, 23
          PRINT "STEP #2:    Zero the height gauge... "
          COLOR RED
          LOCATE 18, 25
          PRINT "and press the RED trigger button."
          CLEARBUFFER
          ON COM(1) GOSUB ZEROSET: COM(1) ON
60        GOTO 60

'****
'*        ENTERING LOCATION COORDINATES
'****

70        ON KEY(1) GOSUB REPEAT:              KEY(1) ON
          ON KEY(2) GOSUB REDOPREVIOUS1:       KEY(2) ON
          ON KEY(3) GOSUB MISSINGTOOTH:        KEY(3) ON
          ON KEY(4) GOSUB RESTART:             KEY(4) ON
          ON KEY(5) GOSUB EXITTODOS:           KEY(5) ON
          COUNT = 1
80        IF COUNT > NUMNAMES THEN 140
          HEADER "Entering the Location Coordinates", YELLOW
          COLOR RED
          CENTER "PRESS RED BUTTON TO ENTER X,Y,Z VALUES", 7, 0
          IF COUNT = 88 THEN GOSUB MANDIBLEON
          COLOR YELLOW
          LOCATE 23, 14
          PRINT "LOCATION #"; COUNT; ": "; FULNAM$(COUNT)
          CENTER "F1=REPEAT  F2=REDO A POINT  F3=MISSING TOOTH  F4=RESTART  F5=EXIT TO  DOS",24,1
          CLEARBUFFER
          ON COM(1) GOSUB INDATA: COM(1) ON                         'ON INPUT FROM TABLET
          CLEARBUFFER
90        GOTO 90                                                    'WAIT HERE FOR COM1 INPUT
100       LOCATE 7, 2
          PRINT SPACE$(78)
          LOCATE 10, 23
          COLOR WHITE
          PRINT USING FORMAT$;"X=";(X(COUNT)-OFFSET.X)/1000;"Y=";(Y(COUNT)-OFFSET.Y)/1000;"Z=";Z(COUNT) LOCATE 12, 25
          PRINT "PRESS THE ";
          COLOR RED
          PRINT "ENTER KEY ";
          COLOR WHITE
          PRINT "TO PROCEED..."
110       LOCATE 14, 21
          PRINT "OR PRESS THE ";
          COLOR RED
          PRINT "F1 ";
          COLOR WHITE
          INPUT "KEY TO REDO THIS LOCATION.", A$
          IF A$ = "" THEN
            COMMENTING COMMENT$(), COUNT
          ELSE
            GOTO 110
          END IF
          COUNT = COUNT + 1
          GOTO 80
```

```
'****
'*     PAPERWORK WITH CURRENT DATA
'****

140     FOR I = 1 TO 5
          KEY(I) OFF
        NEXT I
        COM(1) OFF
        GOSUB SAVING
150     HEADER "Verification of Current Dataset", YELLOW
        CENTER "F1=LIST F2=REDO A POINT F3=CHANGE FILENAME F4=PRINTOUT F5=RERUN F10=EXIT",24,1
        ON KEY(1) GOSUB LISTTOSCREEN:          KEY(1) ON
        ON KEY(2) GOSUB REDOPREVIOUS2:         KEY(2) ON
        ON KEY(3) GOSUB FILENAME:              KEY(3) ON
        ON KEY(4) GOSUB PRNTOUT:               KEY(4) ON
        ON KEY(5) GOSUB RERUN:                 KEY(5) ON
        ON KEY(10) GOSUB FINISHED:             KEY(10) ON
160     GOTO 160                                                'WAIT FOR CHOICE
        END

'*********************************************************************
'*
'*     SYSTEM SUBROUTINES
'*
'*********************************************************************

ZEROSET:
        COM(1) OFF
        INPUT #1, MITUTOYO$
        INPUT #1, OFFSET.X, OFFSET.Y, F
        OFFSET.X = OFFSET.X * 2
        OFFSET.Y = OFFSET.Y * 2
RETURN 70

INDATA:                                                         'WHEN TRIGGER BUTTON PUSHED
        COM(1) OFF
        GOSUB ZHEIGHT
        INPUT #1, X(COUNT), Y(COUNT), F
        X(COUNT) = X(COUNT) * 2
        Y(COUNT) = Y(COUNT) * 2
RETURN 100

ZHEIGHT:                                                        'VERTICAL READING FROM COM(1)
        INPUT #1, D$
        D$ = MID$(D$, 4, 8)
        Z(COUNT) = VAL(D$)
RETURN

MANDIBLEON:                                                     'REMINDER TO INSTALL MANDIBLE
        LOCATE 6, 19
        BEEP
        PRINT "INSTALL THE MANDIBLE BEFORE YOU PROCEED..."
        BEEP
RETURN

REPEAT:                                                         'TO REDO CURRENT LOCATION
        LOCATE 10, 2: PRINT SPACE$(78)
        LOCATE 12, 2: PRINT SPACE$(78)
        LOCATE 13, 2: PRINT SPACE$(78)
        LOCATE 14, 2: PRINT SPACE$(78)
        COLOR RED
        CENTER "PRESS RED BUTTON TO ENTER X,Y,Z VALUES", 7, 0
        CLEARBUFFER
        ON COM(1) GOSUB INDATA: COM(1) ON
        KEY(1) ON
RETURN 90

REDOPREVIOUS1:                                                  'REDO A PREVIOUS LOCATION VER.1
        COM(1) OFF
        HEADER "Change a Previous Location", YELLOW
        LOCATE 12, 26
        PRINT "CURRENT LOCATION IS #"; COUNT
        COLOR WHITE
        LOCATE 14, 15
170     INPUT "ENTER NUMBER OF LOCATION TO REDO:", LOWCAL
        IF LOWCAL < 1 OR LOWCAL > COUNT THEN 170
        LOCATE 12, 2: PRINT SPACE$(78)
        LOCATE 14, 2: PRINT SPACE$(78)
        CENTER "Readings at the " + FULNAM$(LOWCAL) + " point were:", 7, 0
        LOCATE 10, 25
        PRINT USING FORMAT$;"X=";(X(LOWCAL)-OFFSET.X)/1000;"Y=";(Y(LOWCAL)-OFFSET.Y)/1000;"Z=";Z(LOWCAL)
        LOCATE 13, 25
        INPUT "Do You Wish To Change This (y/n)"; CHNG$
```

```
        IF UCASE$(LEFT$(CHNG$, 1)) = "Y" THEN GOTO 180 ELSE GOTO 210
180     LOCATE 13, 2
        PRINT SPACE$(78)
        COLOR RED
        CENTER "PRESS THE RED TRIGGER BUTTON TO ENTER THE NEW LOCATION", 13, 0
        COLOR WHITE
        CLEARBUFFER
        ON COM(1) GOSUB INDATA2: COM(1) ON
190     DO WHILE INKEY$ = "": LOOP'GOTO 190
200     LOCATE 15, 25
        PRINT USING FORMAT$;"X=";(X(LOWCAL)-OFFSET.X)/1000;"Y=";(Y(LOWCAL)-OFFSET.Y)/1000;"Z=";Z(COUNT) GOSUB COMMENTING2
210     CENTER "PRESS THE SPACE BAR TO RETURN TO CURRENT LOCATION", 24, 1
220     B$ = INKEY$: IF B$ <> " " THEN 220
        KEY(2) ON
        GOTO 80
RETURN

INDATA2:                                                    'ON COM1 INPUT VER.2
        COM(1) OFF
        GOSUB ZHEIGHT2
        INPUT #1, X(LOWCAL), Y(LOWCAL), F
        X(LOWCAL) = X(LOWCAL) * 2
        Y(LOWCAL) = Y(LOWCAL) * 2
RETURN 200

ZHEIGHT2:                                                   'VERTICAL READING FROM COM1 VER.2
        INPUT #1, D$
        D$ = MID$(D$, 4, 8)
        Z(LOWCAL) = VAL(D$)
RETURN

COMMENTING2:                                                'INPUT OF CHARACTERISTIC COMMENT
        LOCATE 18, 25
        INPUT "Comments About This Location"; COMMENT$(LOWCAL)
        IF COMMENT$(LOWCAL) = "" THEN COMMENT$(LOWCAL) = "NO COMMENT"
RETURN

MISSINGTOOTH:                                               'FAST ADVANCE FOR MISSING TOOTH
        FOR I = 1 TO 4
          X(COUNT) = OFFSET.X
          Y(COUNT) = OFFSET.Y
          Z(COUNT) = 0
          COMMENT$(COUNT) = "MISSING TOOTH"
          COUNT = COUNT + 1
        NEXT I
        KEY(3) ON
RETURN 80

RESTART:                                                    'RESTART THE DATA LOOP, ALL LOST
        HEADER "Restart", YELLOW
        COLOR WHITE
        LOCATE 13, 15
        PRINT "ARE YOU SURE YOU WISH TO RESTART COMPLETELY? (y/n)"
225     A$ = INKEY$: IF A$ = "" THEN GOTO 225
        IF UCASE$(A$) = "Y" THEN GOTO 55
        KEY(4) ON
RETURN 80

EXITTODOS:                                                  'EXIT TO DOS, ALL SAVED
        HEADER "Exit to DOS", YELLOW
        COLOR WHITE
        CENTER "ARE YOU SURE YOU WISH TO QUIT HERE AND EXIT TO DOS?(y/n)", 13, 1
228     A$ = INKEY$: IF A$ = "" THEN GOTO 228
        IF UCASE$(A$) <> "Y" AND UCASE$(A$) <> "N" THEN GOTO 228
        IF UCASE$(A$) = "N" THEN RETURN 80
        CLOSE
        SYSTEM
RETURN

LISTTOSCREEN:                                               'LIST CURRENT LOCATIONS TO SCREEN
        NUM = 1
230     CLS
        HEADER "Current Location Values", YELLOW
        COLOR WHITE
        LOCATE 6, 2
        FOR I = 1 TO 112
          LOCATE , 2
          PRINT USING DISKFORM$;"LOCATION #";I;"X=";(X(I)-OFFSET.X)/1000;"Y=";(Y(I)-OFFSET.Y)/1000;"Z=";Z(I);CCODE$(I); COMMENT$(I)
          NUM = NUM + 1
          IF NUM = 19 THEN
                COLOR RED
                CENTER "PRESS ANY KEY TO CONTINUE...", 24, 1
                COLOR WHITE
```

```
                DO WHILE INKEY$ = "": LOOP
                LOCATE 6, 2
                FOR J = 1 TO 18
                  LOCATE , 2
                  PRINT SPACE$(78)
                NEXT J
                LOCATE 6, 2
                NUM = 1
                GOTO 240
          END IF
240       NEXT I
          COLOR RED
          CENTER "PRESS ANY KEY TO CONTINUE...", 24, 1
          DO WHILE INKEY$ = "": LOOP
RETURN 150

REDOPREVIOUS2:                                                    'REDO A PREVIOUS LOCATION VER.2
          HEADER "Change a Previous Location", YELLOW
          CENTER "CURRENT LOCATION IS # 112, SKULL COMPLETED", 12, 0
          LOCATE 14, 15
285       INPUT "ENTER NUMBER OF LOCATION TO REDO:", LOWCAL
          IF LOWCAL < 1 OR LOWCAL > 112 THEN 285
          LOCATE 12, 2: PRINT SPACE$(78)
          LOCATE 14, 2: PRINT SPACE$(78)
          CENTER "Readings at the " + FULNAM$(LOWCAL) + " point were:", 7, 0
          LOCATE 10, 25
          PRINT USING FORMAT$; "X="; (X(LOWCAL) - OFFSET.X) / 1000; "Y="; (Y(LOWCAL) - OFFSET.Y) / 1000; "Z="; Z(LOWCAL)
          COLOR WHITE
          LOCATE 13, 25
          INPUT "Do You Wish To Change This (y/n)"; CHNG$
          IF UCASE$(LEFT$(CHNG$, 1)) = "Y" THEN GOTO 286 ELSE RETURN 150
286       LOCATE 13, 2
          PRINT SPACE$(78)
          CENTER "PRESS THE RED TRIGGER BUTTON TO ENTER THE NEW LOCATION", 13, 0
          CLEARBUFFER
          ON COM(1) GOSUB INDATA3: COM(1) ON
287       DO WHILE INKEY$ = "": LOOP'GOTO 287
288       LOCATE 15, 25
          PRINT USING FORMAT$; "X="; (X(LOWCAL) - OFFSET.X) / 1000; "Y="; (Y(LOWCAL) - OFFSET.Y) / 1000; "Z="; Z(COUNT)
          GOSUB COMMENTING2
289       COLOR RED
          CENTER "PRESS THE SPACE BAR TO RETURN", 24, 1
          COLOR WHITE
290       B$ = INKEY$: IF B$ <> " " THEN 290
          GOTO 150
RETURN

INDATA3:                                                          'ON INPUT FROM COM1 VER.3
          COM(1) OFF
          GOSUB ZHEIGHT2
          INPUT #1, X(LOWCAL), Y(LOWCAL), F
          X(LOWCAL) = X(LOWCAL) * 2
          Y(LOWCAL) = Y(LOWCAL) * 2
RETURN 288

RERUN:                                                            'REDO THE CURRENT SKULL COMPLETELY
          HEADER "Quit and Rerun WITHOUT Saving!", YELLOW
          COLOR WHITE
          CENTER "ARE YOU SURE YOU WISH TO RESTART COMPLETELY WITHOUT SAVING THIS DATA (y/n)",13,1 295 E$ = INKEY$
          IF UCASE$(E$) <> "Y" AND UCASE$(E$) <> "N" THEN GOTO 295
          IF UCASE$(E$) = "Y" THEN
            CLOSE
            RETURN 5
          END IF
RETURN 150

FILENAME:                                                         'CHANGE CURRENT FILENAME
          HEADER "New Filename", YELLOW
          COLOR WHITE
          CENTER "CURRENT FILENAME IS: " + FILE$, 13, 0
          LOCATE 15, 34
          INPUT "NEW FILENAME"; FILE$
RETURN 150

PRNTOUT:                                                          'SEND CURRENT FILE TO PRINTER
          HEADER "Print the Current Datafile", YELLOW
          COLOR RED
          CENTER "MAKE SURE PRINTER IS ONLINE AND PRESS ANY KEY TO CONTINUE...", 13, 0
          COLOR WHITE
          DO WHILE INKEY$ = "": LOOP
          CENTER "...PRINTING IN PROGRESS", 15, 0
          LPRINT : LPRINT "FILENAME IS "; FILE$: LPRINT
          FOR I = 1 TO NUMNAMES
```

```
        LPRINT USING DISKFORM$; "LOCATION #"; I; "X="; (X(I) - OFFSET.X) / 1000; "Y="; (Y(I) - OFFSET.Y) / 1000; "Z="; Z(I); CCODE$(I);
        COMMENT$(I)
      NEXT I
      LPRINT "    END OF FILE"
      LPRINT CHR$(12)
      BEEP
      CENTER "PRINTING COMPLETED, RETURNING TO PROGRAM.", 20, 0
      FOR I = 1 TO 1000: NEXT I
RETURN 150

FINISHED:                                                    'FINISHED, SAVE, RESTART
      CLOSE
      KILL "\QB\TMJ\STORAGE\" + FILE$
      KILL "A:" + FILE$
      LETTERS = LEN(FILE$)
      FILE$ = LEFT$(FILE$, (LETTERS - 3))
      FILE$ = FILE$ + "DAT"
      OPEN "\QB\TMJ\STORAGE\" + FILE$ FOR OUTPUT AS #2       'OPEN COM TO DRIVE C
      OPEN "A:" + FILE$ FOR OUTPUT AS #3                     'OPEN COM TO DRIVE A
      PRINT #2, CLASS
      PRINT #2, TIME$
      PRINT #2, DATE$
      PRINT #3, CLASS
      PRINT #3, TIME$
      PRINT #3, DATE$
      FOR I = 1 TO NUMNAMES
        PRINT #2, USING DISKFORM$; "LOCATION #"; I; "X="; (X(I) - OFFSET.X) / 1000; "Y="; (Y(I) - OFFSET.Y) / 1000; "Z="; Z(I); CCODE$(I);
        COMMENT$(I)
        PRINT #3, USING DISKFORM$; "LOCATION #"; I; "X="; (X(I) - OFFSET.X) / 1000; "Y="; (Y(I) - OFFSET.Y) / 1000; "Z="; Z(I); CCODE$(I);
        COMMENT$(I)
      NEXT I
      CLOSE
      RUN "TMJDATA.EXE"
RETURN

SAVING:
      PRINT #2, CLASS
      PRINT #2, TIME$
      PRINT #2, DATE$
      PRINT #3, CLASS
      PRINT #3, TIME$
      PRINT #3, DATE$
      FOR I = 1 TO NUMNAMES
        PRINT #2, USING DISKFORM$; "LOCATION #"; I; "X="; (X(I) - OFFSET.X) / 1000; "Y="; (Y(I) - OFFSET.Y) / 1000; "Z="; Z(I); CCODE$(I);
        COMMENT$(I)
        PRINT #3, USING DISKFORM$; "LOCATION #"; I; "X="; (X(I) - OFFSET.X) / 1000; "Y="; (Y(I) - OFFSET.Y) / 1000; "Z="; Z(I); CCODE$(I);
        COMMENT$(I)
      NEXT I
RETURN

'**********************************************************************
'*
'*    FULL AND ABBREVIATED DIGITATION LOCATIONS
'*
'*          WITH MANDIBLE 'OFF'
'*
'**********************************************************************

      DATA ZERO POINT MAXILLARY MIDLINE,Z.P.
      DATA A,A
      DATA ANTERIOR NASAL SPINE,A.N.S.
      DATA NASION,NAS.
      DATA RIGHT INFERIOR BORDER OF ORBIT,R.INF.ORB.
      DATA RIGHT ANTERIOR TEMPORALIS ORIGIN,R.ANT.TEMP.o
      DATA RIGHT MIDDLE TEMPORALIS ORIGIN,R.MID.TEMP.o
      DATA RIGHT POSTERIOR TEMPORALIS ORIGIN,R.POST.TEMP.o
      DATA RIGHT SUPERFICIAL MASSETER ORIGIN,R.SUP.MAS.o
      DATA RIGHT DEEP MASSETER ORIGIN,R.DEEP.MAS.o
      DATA RIGHT LATERAL PTERYGOID ORIGIN,R.LAT.PT.o
      DATA LEFT MEDIAL PTERYGOID ORIGIN,L.MED.PT.o
      DATA POSTERIOR NASAL SPINE,P.N.S.
      DATA RIGHT FOSSA POINT 1,R.F.P.1
      DATA RIGHT FOSSA POINT 2,R.F.P.2
      DATA RIGHT FOSSA POINT 3,R.F.P.3
      DATA RIGHT FOSSA POINT 4,R.F.P.4
      DATA RIGHT SUPERIOR BORDER OF AUDITORY MEATUS,R.SUP.AUD.M.
      DATA RIGHT ANTERIOR BORDER OF AUDITORY MEATUS SURFACE,R.ANT.AUD.M.SUF.
      DATA RIGHT ANTERIOR BORDER OF AUDITORY MEATUS DEEP,R.ANT.AUD.M.DP.
      DATA RIGHT THIRD MOLAR FIRST,RM31
      DATA RIGHT THIRD MOLAR SECOND,RM32
      DATA RIGHT THIRD MOLAR THIRD,RM33
      DATA RIGHT THIRD MOLAR FOURTH,RM34
      DATA RIGHT THIRD MOLAR FIFTH,RM35
```

```
DATA RIGHT SECOND MOLAR FIRST,RM21
DATA RIGHT SECOND MOLAR SECOND,RM22
DATA RIGHT SECOND MOLAR THIRD,RM23
DATA RIGHT SECOND MOLAR FOURTH,RM24
DATA RIGHT SECOND MOLAR FIFTH,RM25
DATA RIGHT FIRST MOLAR FIRST,RM11
DATA RIGHT FIRST MOLAR SECOND,RM12
DATA RIGHT FIRST MOLAR THIRD,RM13
DATA RIGHT FIRST MOLAR FOURTH,RM14
DATA RIGHT FIRST MOLAR FIFTH,RM15
DATA RIGHT FIRST MOLAR MEASIAL MARGINAL RIDGE,RM1.MMR
DATA RIGHT SECOND PREMOLAR FIRST,RP21
DATA RIGHT SECOND PREMOLAR SECOND,RP22
DATA RIGHT SECOND PREMOLAR THIRD,RP23
DATA RIGHT SECOND PREMOLAR FOURTH,RP24
DATA RIGHT SECOND PREMOLAR FIFTH,RP25
DATA RIGHT FIRST PREMOLAR FIRST,RP11
DATA RIGHT FIRST PREMOLAR SECOND,RP12
DATA RIGHT FIRST PREMOLAR THIRD,RP13
DATA RIGHT FIRST PREMOLAR FOURTH,RP14
DATA RIGHT FIRST PREMOLAR FIFTH,RP15
DATA LEFT INFERIOR BORDER OF ORBIT,L.INF.ORB.
DATA LEFT ANTERIOR TEMPORALIS ORIGIN,L.ANT.TEMP.o
DATA LEFT MIDDLE TEMPORALIS ORIGIN,L.MID.TEMP.o
DATA LEFT POSTERIOR TEMPORALIS ORIGIN,L.POST.TEMP.o
DATA LEFT SUPERFICIAL MASSETER ORIGIN,L.SUP.MAS.o
DATA LEFT DEEP MASSETER ORIGIN,L.DEEP.MAS.o
DATA LEFT LATERAL PTERYGOID ORIGIN,L.LAT.PT.o
DATA RIGHT MEDIAL PTERYGOID ORIGIN,R.MED.PT.o
DATA LEFT FOSSA POINT 1,L.F.P.1
DATA LEFT FOSSA POINT 2,L.F.P.2
DATA LEFT FOSSA POINT 3,L.F.P.3
DATA LEFT FOSSA POINT 4,L.F.P.4
DATA LEFT SUPERIOR BORDER OF AUDITORY MEATUS,L.SUP.AUD.M.
DATA LEFT ANTERIOR BORDER OF AUDITORY MEATUS SURFACE,L.ANT.AUD.M.SUF.
DATA LEFT ANTERIOR BORDER OF AUDITORY MEATUS DEEP,L.ANT.AUD.M.DP.
DATA LEFT THIRD MOLAR FIRST,LM31
DATA LEFT THIRD MOLAR SECOND,LM32
DATA LEFT THIRD MOLAR THIRD,LM33
DATA LEFT THIRD MOLAR FOURTH,LM34
DATA LEFT THIRD MOLAR FIFTH,LM35
DATA LEFT SECOND MOLAR FIRST,LM21
DATA LEFT SECOND MOLAR SECOND,LM22
DATA LEFT SECOND MOLAR THIRD,LM23
DATA LEFT SECOND MOLAR FOURTH,LM24
DATA LEFT SECOND MOLAR FIFTH,LM25
DATA LEFT FIRST MOLAR FIRST,LM11
DATA LEFT FIRST MOLAR SECOND,LM12
DATA LEFT FIRST MOLAR THIRD,LM13
DATA LEFT FIRST MOLAR FOURTH,LM14
DATA LEFT FIRST MOLAR FIFTH,LM15
DATA LEFT FIRST MOLAR MEASIAL MARGINAL RIDGE,LM1.MMR
DATA LEFT SECOND PREMOLAR FIRST,LP21
DATA LEFT SECOND PREMOLAR SECOND,LP22
DATA LEFT SECOND PREMOLAR THIRD,LP23
DATA LEFT SECOND PREMOLAR FOURTH,LP24
DATA LEFT SECOND PREMOLAR FIFTH,LP25
DATA LEFT FIRST PREMOLAR FIRST,LP11
DATA LEFT FIRST PREMOLAR SECOND,LP12
DATA LEFT FIRST PREMOLAR THIRD,LP13
DATA LEFT FIRST PREMOLAR FOURTH,LP14
DATA LEFT FIRST PREMOLAR FIFTH,LP15

'****
'*      WITH MANDIBLE 'ON'
'****

DATA RIGHT TEMPORALIS INSERT,R.TEMP.i
DATA RIGHT POSTERIOR CONDYLE,R.POST.CON.
DATA RIGHT DISTO INFERIOR POINT OF RAMUS,R.DIST.INF.PT.RAM.
DATA RIGHT LATERAL PTERYGOID INSERT,R.LAT.PT.i
DATA RIGHT GONION,R.GON.
DATA RIGHT SUPERFICIAL MASSETER INSERT,R.SUP.MAS.i
DATA RIGHT DEEP MASSETER INSERT,R.DEEP.MAS.i
DATA LEFT MEDIAL PTERYGOID INSERT, L. MED. PT. i
DATA B,B
DATA RIGHT POGONION,R.POGO.
DATA CENTRAL POGONION,CENT.POGO.
DATA LEFT POGONION,L.POGO.
DATA LEFT MENTON,L.MNT.
DATA CENTRAL MENTON,CENT.MNT.
DATA RIGHT MENTON,R.MNT.
DATA INCISOR POINT (UPPER),I.P.U.
```

```
        DATA INCISOR POINT (LOWER),I.P.L.
        DATA LEFT TEMPORALIS INSERT,L.TEMP.i
        DATA LEFT POSTERIOR CONDYLE,L.POST.CON.
        DATA LEFT DISTO INFERIOR POINT OF RAMUS,L.DIST.INF.PT.RAM.
        DATA LEFT LATERAL PTERYGOID INSERT,L.LAT.PT.i
        DATA LEFT GONION,L.GON.
        DATA LEFT SUPERFICIAL MASSETER INSERT,L.SUP.MAS.i
        DATA LEFT DEEP MASSETER INSERT,L.DEEP.MAS.i
        DATA RIGHT MEDIAL PTERYGOID INSERT, R. MED. PT. i

SUB CENTER (STATEMENT$, LYNE, WRAPTEXT)

'************************************************************************
'*
'*      CENTER is a routine which takes a statement that is passed to it
'*      and prints it centered on line #LYNE.  If the falg WRAPTEXT is
'*      zero then the cursor advances one line.  If WRAPTEXT is anything
'*      else the cursor remains at the end of the printed STATEMENT$.
'*
'************************************************************************

        LENGTH = LEN(STATEMENT$)
        MID = 40 - (LENGTH / 2)
        LOCATE LYNE, MID
        IF WRAPTEXT = 0 THEN
           PRINT STATEMENT$
        ELSE
           PRINT STATEMENT$;
        END IF
END SUB

SUB CLEARBUFFER

'************************************************************************
'*
'*      CLEARBUFFER is a routine that empties the communications buffer
'*      on COM1.
'*
'************************************************************************

        WHILE NOT EOF(1)
           FOR I = 1 TO 200: NEXT I                  'WAIT FOR BUFFER TO FILL
           A$ = INPUT$(LOC(1), #1)
        WEND
END SUB

SUB COMMENTING (COMMENT$(), COUNT)

'************************************************************************
'*
'*      COMMENTING is a routine that erases a portion of the screen and
'*      then prompts the user for a comment about the curent location.
'*
'************************************************************************

        LOCATE 12, 2
        PRINT SPACE$(78)
        LOCATE 14, 2
        PRINT SPACE$(78)
        LOCATE 13, 25
        INPUT "Comments About This Location"; COMMENT$(COUNT)
        IF COMMENT$(COUNT) = "" THEN COMMENT$(COUNT) = "NO COMMENT"
END SUB

SUB HEADER (TITLE$, ATTRIBUTE)

'************************************************************************
'*
'*      HEADER is the routine that is used to display the graphic border
'*      on the screen.  It CENTERs the TITLE$ on the third line.  The
'*      header is displayed in the color ATTRIBUTE.
'*
'************************************************************************

        CLS
        COLOR ATTRIBUTE
        LOCATE 1, 1
        PRINT CHR$(201) + STRING$(17, 205) + CHR$(209) + STRING$(42, 205) + CHR$(209) + STRING$(17, 205) + CHR$(187)
        PRINT CHR$(186) + SPACE$(17) + CHR$(179) + SPACE$(42) + CHR$(179) + SPACE$(17) + CHR$(186)
        PRINT CHR$(199) + STRING$(17, 196) + CHR$(193) + STRING$(42, 196) + CHR$(193) + STRING$(17, 196) + CHR$(182)
        PRINT CHR$(186); : LOCATE , 80: PRINT CHR$(186)
        PRINT CHR$(199) + STRING$(78, 196) + CHR$(182)
        FOR I = 6 TO 23
```

```
        PRINT CHR$(186); : LOCATE , 80: PRINT CHR$(186)
      NEXT I
      PRINT CHR$(186); : LOCATE , 80: PRINT CHR$(186);
      LOCATE 25, 1
      PRINT CHR$(200) + STRING$(78, 205) + CHR$(188);
      LOCATE 2, 6: PRINT DATE$;
      CENTER "THREE DIMENSIONAL DIGITIZER", 2, 1
      LOCATE , 67: PRINT TIME$
      CENTER TITLE$, 4, 0
END SUB
```

# Appendix #2: Numerical Model Source Code

```
DECLARE SUB ResultantOutputFile (CodeFlag!, NUM!, ANB!, OWTPUT!(), BATCHNUMBER$, DESCRIPTION$)
DECLARE SUB BATCHSELECT (PICKED!)
DECLARE SUB DESCRIBE (LYNE!, LEFT!, RIGHT!, DEFAULT$)
DECLARE SUB SCREENDISPLAY (FILE$, CLASS$, PICKED!)
DECLARE SUB MTRIX ()
DECLARE SUB MOVEORIGIN (X!(), Y!(), Z!(), VECTORS!())
DECLARE SUB TURN (ANGLE!, A!, B!, LOCATION!())
DECLARE SUB OCCLUSALPLANE (X!(), Y!(), Z!(), OFFSET!, PICKED!)
DECLARE SUB MODDATAREAD (CLASS$, OFFSET, ANB!, X!(), Y!(), Z!(), COMMENT$())
DECLARE SUB GETVECTORS (VECTORS!())
DECLARE SUB EMG (PICKED)
DECLARE SUB GETPOINT (START!, LAST!, START$, PICKED!)
DECLARE SUB RESULTANT ()
DECLARE SUB SAVENEW (FILENUM$, X!(), Y!(), Z!(), COMMENT$())
DECLARE SUB UNITVECTOR (NAME$, XI!, YI!, ZI!, XO!, YO!, ZO!)
DECLARE SUB DOSSHELL ()
DECLARE SUB FRANKFURT (X!(), Y!(), Z!(), COMMENT$())
DECLARE SUB CENTER (STATEMENT$, LYNE!)
DECLARE SUB UPDATE ()
DECLARE SUB ADD2CAT (FILE$, CLASS$)
DECLARE SUB DATAREAD (X(), Y(), Z(), COMMENT$(), TYME$, DAYT$, CLASS$)
DECLARE SUB CONVERT ()
DECLARE SUB HEADER (TITLE$)
DECLARE SUB ROTATE (HORZ, VERT, A(), B())

'*****************************************************************
'*
'*    This program is designed to input raw 3-D data from data files
'*    created by the program TMJDATA.EXE used for 3-D digitization of
'*    human skulls for use in research of Temporomandibular joint
'*    loading. This program will translate the raw data into cartesian
'*    coordinates on the "Frankfurt Plane". Once translated it will
'*    proceed and do a 3-D static analysis of the muscle force vectors
'*    in order to find the resultant condylar loads.
'*
'*                                   Version 1.1
'*                                   May 1989
'*                                   by Steven McEvoy
'*
'*           Any reproduction or unauthorized use of these materials
'*        would be unethical, dishonest, and unprofessional. It is also
'*        illegal. Please don't test our friendship.
'*
'*****************************************************************

        KEY OFF
        KEY 15, CHR$(4) + CHR$(70)                        'Trap the <CTRL><BRK>
        ON KEY(15) GOSUB CTRLBRK: KEY(15) ON              'as "Return to main"
        ON ERROR GOTO ERRORHANDLER                        'Turn errorhandling ON
        DIM MENOPT$(5)                                    'Main menu strings
        DIM ERRCODE(11)                                   'Recognized error codes
        DIM UNDERLAY(5555)                                'Graphics storage array
        DIM SHARED VALS(14)                               'EMG values array
        DIM SHARED EMGVALS(14)                            'Used EMG values array
        DIM SHARED BITEPOINT$(10, 2)                      'Bite location strings
        DIM SHARED BYTEPOINT(6)                           'Calculated bite data
        DIM SHARED MATRIX(7, 8)                           'Unknown variables mtrx
        DIM SHARED FORCES(7)                              'Resolved unknowns
        DIM SHARED OWTPUT(8)                              'Resolved output

'****
'*              Menu titles
'****

        MENOPT$(1) = "RAW DATA CONVERSION"
        MENOPT$(2) = "RESULTANT CONDYLAR LOADING"
        MENOPT$(3) = "GRAPHICAL COMPARISONS"
        MENOPT$(4) = "SHELL TO DOS"
        MENOPT$(5) = "EXIT PROGRAM"

'****
'*              Main menu screen
'****

        SCREEN 9                                          'Set the screen to 640 x 350
5       PALETTE 1, 63                                     'Set 1 to high intense white
        PALETTE 2, 54                                     'Set 2 to yellow
        PALETTE 3, 4[illegible]                           'Set 3 to brillant blue
        PALETTE 4, 4                                      'Set 4 to blood red
        PALETTE 5, 28                                     'Set 5 to yuppie purple
```

```
        PALETTE 6, 53                                   'Set 6 to yuppie pink
        PALETTE 7, 56                                   'Set 7 to yuppie grey
        PALETTE 8, 7                                    'Set 8 to light grey
        PALETTE 9, 36                                   'Set 9 to bright orange
        PALETTE 10, 37                                  'Set 10 to bright pink
        PALETTE 11, 26                                  'Set 11 to dayglo green
        PALETTE 12, 20                                  'Set 12 to dark brown
        PALETTE 13, 44                                  'Set 13 to popsicle red
        PALETTE 14, 49                                  'Set 14 to dark blue
        PALETTE 15, 12                                  'Set 15 to yuppie red/purple
        COLOR 2, 1                                      'Set foreground to yellow
        CLS

'****
'              Draw and fill screen header
'****

        HEADER "Main Menu"

'****
'              Setup menu scrolling
'****

        PICK = 1
10      FOR NUM = 1 TO 5
          LOCATE (8 + (2 * NUM)), 25
          COLOR 1
          PRINT MENOPT$(NUM)
        NEXT NUM
        LOCATE (8 + (2 * PICK)), 25
        COLOR 13
        PRINT MENOPT$(PICK)
        COLOR 2
        LOCATE 23, 19
        PRINT "Select option and press <RETURN> to continue..."
20      PICK$ = INKEY$
        IF PICK$ = "" THEN
          UPDATE                                        'Update the clock
          GOTO 20
        END IF
        PICK$ = RIGHT$(PICK$, 1)
        ASCII = ASC(PICK$)
        OLDPICK = PICK
        IF ASCII = 72 OR ASCII = 75 THEN PICK = PICK - 1
        IF PICK = 0 THEN PICK = 5
        IF ASCII = 80 OR ASCII = 77 THEN PICK = PICK + 1
        IF PICK = 6 THEN PICK = 1
        IF ASCII = 13 THEN GOTO 30
        LOCATE (8 + (2 * OLDPICK)), 25
        COLOR 1
        PRINT MENOPT$(OLDPICK)
        LOCATE (8 + (2 * PICK)), 25
        COLOR 13
        PRINT MENOPT$(PICK)
        GOTO 20

'****
'          Call the chosen subroutine
'****

30      IF PICK = 1 THEN CONVERT
        IF PICK = 2 THEN RESULTANT
        IF PICK = 3 THEN DOSSHELL
        IF PICK = 4 THEN DOSSHELL
        IF PICK = 5 THEN END
        GOTO 5
        END

CTRLBRK:

'*****************************************************************
'
'    This subroutine traps the <Control><Break> key sequence and returns
'    to the main menu from any point in the program.
'
'*****************************************************************

        GOTO 5
        KEY(15) ON

RETURN

ERRORHANDLER:
```

```
'*************************************************************************
'*
'*     Errorhandler is a routine which can handle certain defined
'*     execution errors.  A non-handible error causes the program
'*     to stop, provide an explaination message and then die.
'*
'*************************************************************************

'****
'*              Handible Error Codes
'****

        ERRCODE(1) = 7                                  'Out of RAM memory
        ERRCODE(2) = 11                                 'Divide by Zero
        ERRCODE(3) = 27                                 'Printer off line
        ERRCODE(4) = 52                                 'Bad file name
        ERRCODE(5) = 53                                 'File not found
        ERRCODE(6) = 55                                 'File already open
        ERRCODE(7) = 61                                 'Diskette is full
        ERRCODE(8) = 64                                 'Bad file name
        ERRCODE(9) = 71                                 'Disk Drive door open
        ERRCODE(10) = 75                                'Couldn't find file
        ERRCODE(11) = 76                                'Couldn't find path
'****
'*              Scan error codes
'****

        CODE = 1
        FOR I = 1 TO 11
           IF ERR = ERRCODE(I) THEN                     'Error code match
                 GOTO 40                                'Exit loop
           ELSE
                 CODE = CODE + 1                        'No match
           END IF
        NEXT I
        IF CODE = 12 THEN RESUME 0                      'Unhandiable error
'****
'*              Store the soon to be written on
'*              screen area.
'****

40      X1 = 120
        Y1 = 150
        WDTH = 400                                      'Window width
        HEIGHT = 100                                    'Window height
        GET (X1, Y1)-((X1 + WDTH), (Y1 + HEIGHT)), UNDERLAY
        DRAW "C2 BM120,150 R400 D100 L400 U100 BF10 P3,2"
'****
'*              Process each error code
'****

        CENTER "Drive not ready, Press any key to continue...", 14
50      A$ = INKEY$: IF A$ = "" THEN 50
        PUT (X1, Y1), UNDERLAY, PSET
        RESUME
RETURN

SUB ADD2CAT (FILE$, CLASS$)

'*************************************************************************
'*
'*     ADD2CAT is a routine which is called to automatically append
'*     catalog files for class 1, 2 and 3 skulls.
'*
'*************************************************************************

        IF CLASS$ <> "1" AND CLASS$ <> "2" AND CLASS$ <> "3" THEN
           BEEP
           PRINT "Invalid CLASS Passed to Cataloging procedure"
           EXIT SUB
        END IF
        FILENAME$ = "\QB\TMJ\DATA\CLASSDIR." + CLASS$
        OPEN FILENAME$ FOR APPEND AS #2
        WRITE #2, FILE$
        CLOSE #2
END SUB

SUB BATCHSELECT (PICKED) STATIC
```

```
'*********************************************************************
'*
'*      BATCHSELECT is a routine which prompts the user to define a
'*      run # necessary to create output file storage names.  It also
'*      opens these files and writes the relevant data about the runs to
'*      the files.
'*
'*********************************************************************

        CLS
        COLOR 2
        HEADER "Batch Definition"
        BATCHNUMBER = BATCHNUMBER + 1
        SUGGEST$ = LTRIM$(STR$(BATCHNUMBER))
        IF BATCHNUMBER = 1 THEN
           SUGGEST$ = "??"
           BATCHNUMBER = 0
        END IF
        COLOR 2
        LOCATE 11, 16
        PRINT "Please enter the batch # for this run ";
        COLOR 1: PRINT "<"; SUGGEST$; ">";
        COLOR 2: PRINT ":";
        COLOR 1: LOCATE , 60
        INPUT "", BATCHNUMBER$
        IF BATCHNUMBER$ = "" THEN BATCHNUMBER$ = SUGGEST$
        BATCHNUMBER$ = LTRIM$(BATCHNUMBER$)
        BATCHNUMBER = VAL(BATCHNUMBER$)
        IF BATCHNUMBER = 999 THEN
           CodeFlag = 0
        ELSE
           CodeFlag = 1
        END IF
        COLOR 2
        CENTER "Please enter a few descriptive comments about this run:", 14
        COLOR 1
        DESCRIBE 15, 20, 60, DESCRIPTION$
        COLOR 2

'****
'*              Open the storage files
'****

        OWTPUT$(1) = PICKED
        ResultantOutputFile CodeFlag, 0, 0, OWTPUT(), BATCHNUMBER$, DESCRIPTION$

END SUB

SUB CENTER (STATEMENT$, LYNE)

'*********************************************************************
'*
'*      CENTER is a routine which takes a statement that is passed to it
'*      and prints it centered on line #LYNE.
'*
'*********************************************************************

        LENGTH = LEN(STATEMENT$)
        MID = 40 - (LENGTH / 2)
        LOCATE LYNE, MID
        PRINT STATEMENT$
END SUB

SUB CONVERT

'*********************************************************************
'*
'*      CONVERT is a routine which inputs user specified file ranges,
'*      opens the files, reads and catalogs them and finally rotates
'*      the data set into orientation with the Frankfurt Plane.
'*
'*********************************************************************

        DIM X(113), Y(113), Z(113), COMMENT$(113)
2000    CLS
        COLOR 2
        HEADER "Raw Data File Coordinate Conversion"
        CENTER "This routine is for converting raw data files singularily", 7
        CENTER " or by group to data oriented about the Frankfurt Plane.", 8

'****
'*              Input the first file to be converted
'****
```

```
2010    LOCATE 11, 20
        COLOR 1
        INPUT "First Skull Number (i.e. 001): ", START$
        IF START$ = "" THEN GOTO 2010                       'Check for null entry
        START = VAL(START$)
        IF START < 0 OR START > 999 THEN 2010               'Check if out of range
        IF START = 0 THEN
           LAST$ = START$
           LOCATE 13, 20
           PRINT "Last Skull Number <First>:     "; LAST$
           GOTO 2030
        END IF

'****
'*              Input the last file to be converted
'****

2020    LOCATE 13, 20
        INPUT "Last Skull Number <First>:     ", LAST$
        IF LAST$ = "" THEN                                  'Set default to START$
           LAST$ = START$
           LOCATE 13, 51
           PRINT LAST$
        END IF
2030    LAST = VAL(LAST$)
        IF LAST < 0 OR LAST > 999 OR LAST < START THEN 2020

'****                                                        'Check for valid value
'*              Look for non-numeric filenames
'****

        IF LAST$ = START$ THEN
           START = 1000                                     'Set a flag
           LAST = 1000
           COLOR 8
           LOCATE 17, 17
           PRINT "You have chosen to convert file TMJ"; UCASE$(START$); ".DAT"
           GOTO 2040
        END IF

'****
'*              Verify inputs
'****

        COLOR 8
        LOCATE 17, 13
        PRINT "You have chosen to convert files TMJ"; UCASE$(START$); ".DAT to TMJ"; UCASE$(LAST$); ".DAT"
2040    COLOR 13
        LOCATE 24, 7
        PRINT "Press ";
        COLOR 1
        PRINT "<RETURN>";
        COLOR 13
        PRINT " to continue or ";
        COLOR 1
        PRINT "<ESCAPE>";
        COLOR 13
        PRINT " to return to the Main Menu...";
        COLOR 1
2050    A$ = INKEY$: IF A$ = "" THEN 2050
        B$ = RIGHT$(A$, 1)
        IF ASC(B$) = 13 THEN
          GOTO 2060                                          'Pressing RETURN
        ELSE
          IF ASC(B$) = 27 THEN
                  EXIT SUB                                   'Pressing ESCAPE
          END IF
          GOTO 2050
        END IF                                               'Pressing anything else

'****
'*              Loop for the multiple file conversions
'****

2060    CLS
        COLOR 2
        HEADER "Raw Data File Coordinate Conversion"         'Redraw Header
        LOCATE 11, 12
        INPUT "Please enter the full path to the raw data files <A:>:", PATH$
        IF PATH$ = "" THEN PATH$ = "A:"
        FOR NUM = START TO LAST
```

```
'****
'*              Building the filename
'****

        FILENUM$ = STR$(NUM)
        FILENUM$ = LTRIM$(FILENUM$)
        IF NUM = 1000 THEN FILENUM$ = START$            'Check non-numeric flag
        FILE$ = "TMJ" + FILENUM$ + ".DAT"               'Build the new filename
        FILENAME$ = PATH$ + FILE$                       'Build the full-pathname

'****
'*              Call the Conversion Routines
'****

        LOCATE 11, 12
        PRINT SPACE$(65)
        LOCATE 13, 25
        PRINT "CONVERTING DATA FILE: "; FILENAME$
        OPEN FILENAME$ FOR INPUT AS #1

'****
'*              Read Raw-Data file
'****

        DATAREAD X(), Y(), Z(), COMMENTS$(), TYME$, DAYT$, CLASS$

'****
'*              Catalog the skull's filename by skull class
'****

        ADD2CAT FILE$, CLASS$

'****
'*              Align the dataset to the Frankfort plane
'****

        FRANKFURT X(), Y(), Z(), COMMENTS$()

'****
'*              Save the modified dataset to a new file
'****

        SAVENEW FILENUM$, X(), Y(), Z(), COMMENTS$()
      NEXT NUM

'****
'*              Finished file conversions, check for more
'****

      LOCATE 13, 1: PRINT SPACE$(75)
2070  LOCATE 12, 20
      INPUT "Do you wish to convert any more files? (y/n) ", B$
      IF B$ <> "Y" AND B$ <> "y" AND B$ <> "N" AND B$ <> "n" THEN 2070
      IF B$ <> "Y" OR B$ <> "y" THEN EXIT SUB ELSE 2000
END SUB

SUB DATAREAD (X(), Y(), Z(), COMMENTS$(), TYME$, DAYT$, CLASS$)
'******************************************************************************
'*
'*    DATAREAD is a routine which reads the passed file into the
'*    arrays X(), Y() and Z(). It includes several error checks.
'*
'******************************************************************************

      INPUT #1, CLASS$                                  'READ SKULL CLASS
      INPUT #1, TYME$                                   'READ DATE DIGITIZED
      INPUT #1, DAYT$                                   'READ TIME DIGITIZED
      FOR K = 1 TO 117
        INPUT #1, A, B, C, D, E, F, G, H, I, J, A$

'****
'*              Check that file locations match the array locations
'****

        IF K <> C THEN
                BEEP
                COLOR 31
                LOCATE 15, 25
                PRINT "READ LOOP LOCATION MISMATCH, READ TERMINATED!"
                COLOR 15
                FOR DELAY = 1 TO 20000: NEXT DELAY
                LOCATE 15, 1
```

```
                    PRINT SPACE$(75)
                    BEEP
                    EXIT SUB
            END IF

'****
'*              Get the values of X, Y and Z and store them.
'****

            X(K) = E                                'X DIR READINGS ARRAY
            Y(K) = G                                'Y DIR READINGS ARRAY
            Z(K) = I                                'Z DIR READINGS ARRAY
            COMMENTS$(K) = A$                       'COMMENTS ARRAY
        NEXT K
        CLOSE #1
END SUB

SUB DESCRIBE (LYNE, LEFT, RIGHT, DEFAULT$) STATIC

'***********************************************************************
'*
'*      DESCRIBE is a routine which acts as a simple word processor to
'*      allow the neat and tidy input of character strings. You pass it
'*      the LYNE it should start on and the LEFT and RIGHT margins. It
'*      returns the character string in DEFAULT$.
'*
'***********************************************************************

'****
'*              Set the counters
'****

        STP = 0
        NUM = 0
        COUNT = 0
        LOCATE LYNE, LEFT
        LETTERS = LEN(DEFAULT$)                     'Count # of characters
        IF LETTERS = 0 THEN 97                      'If there is no string

'****
'*              Display the default message
'****

        FOR I = 1 TO LETTERS                        'Print the current
           IF POS(0) < (RIGHT + 1) THEN             'DEFAULT$
                    PRINT MID$(DEFAULT$, I, 1);
           ELSE
                    STP = STP + 1
                    LOCATE (LYNE + STP), LEFT
                    PRINT MID$(DEFAULT$, I, 1);
           END IF
        NEXT I

'****
'*              Display the help message
'****

        LOCATE 24, 8
        COLOR 13: PRINT "Press the ";
        COLOR 1: PRINT "<RETURN>";
        COLOR 13: PRINT " key to accept this or the ";
        COLOR 1: PRINT "<DELETE>";
        COLOR 13: PRINT " key to edit.";
        DO                                          'Wait for a key press
           A$ = INKEY$
        LOOP WHILE A$ = ""

'****
'*              Process the key press
'****

        IF ASC(A$) = 13 THEN                        'If <RETURN> pressed
           EXIT SUB                                 'accept DEFAULT$
        ELSE
        'IF ASC(A$) = 83 THEN                       'If <DELETE> pressed
           TEMP = RIGHT - LEFT + 1                  'erase DEFAULT$
           DOWN = INT(LETTERS / TEMP)
           FOR I = 0 TO DOWN
                    LOCATE (LYNE + I), LEFT
                    PRINT SPACE$(TEMP)
           NEXT I
           DEFAULT$ = ""
        END IF
```

```
'****
'*              Display the help message
'****

97      LOCATE 24, 2
        PRINT SPACE$(76);
        LOCATE 24, 8
        COLOR 13: PRINT "Press the ";
        COLOR 1: PRINT "<BACKSPACE>";
        COLOR 13: PRINT " key to delete and the ";
        COLOR 1: PRINT "<END>";
        COLOR 13: PRINT " key to continue...";
        COLOR 1
        LOCATE LYNE, LEFT

'****
'*              Enter the text editor
'****

        DO
          DO WHILE POS(O) > (LEFT - 1) AND POS(O) < (RIGHT + 1)
                DO
                  CHAR$ = INKEY$
                LOOP WHILE CHAR$ = ""
                IF ASC(CHAR$) = 0 THEN                          'If <NUL> precedes the
                  CHAR$ = MID$(CHAR$, 2)                        'ASCII extended code
                END IF
                IF ASC(CHAR$) = 79 THEN EXIT SUB                'Press <END> to end
                IF ASC(CHAR$) = 13 THEN                         'If <RETURN> pressed
                  COUNT = COUNT + 1                             'then carrige return
                  LOCATE (LYNE + COUNT), LEFT
                  GOTO 98
                END IF
                IF ASC(CHAR$) = 8 THEN                          'If <BACKSPACE) pressed
                  COL = POS(O)                                  'Set COL = current clmn
                  COL = COL - 1                                 'Move back one space
                  IF COL < LEFT THEN                            'If text wraps back up
                        LOCATE (LYNE + COUNT), LEFT             'Erase current line
                        PRINT SPACE$(RIGHT - LEFT)
                        COUNT = COUNT - 1                       'Set line counter
                        IF COUNT = -1 THEN
                                COUNT = 0
                                LOCATE LYNE, LEFT
                                GOTO 98
                        END IF
                        COL = RIGHT                             'Set column counter at
                  END IF                                        'far right
                  LOCATE (LYNE + COUNT), COL                    'Move back one space
                  PRINT " ";                                    'Erase that space
                  LOCATE , COL
                  NUM = LEN(DEFAULT$) - 1                       'Shorten DEFAULT$ one
                  DEFAULT$ = MID$(DEFAULT$, 1, NUM)             'character
                  GOTO 98
                END IF
                PRINT CHAR$;                                    'Print the text char
                DEFAULT$ = DEFAULT$ + CHAR$                     'Add it to DEFAULT$
                IF LEN(DEFAULT$) = 256 THEN EXIT SUB            'Limit the length
98        LOOP
          COUNT = COUNT + 1                                     'Increase the line
          LOCATE (LYNE + COUNT), LEFT                           'counter and move
        LOOP
END SUB

SUB DOSSHELL

'*******************************************************************
'*
'*      This routine executes a DOS shell
'*
'*******************************************************************

        COLOR 2
        CLS
        HEADER "DOS Shell"

'****
'*              Create a text box
'****

        DRAW "C1 BM125,130 R390 D115 L390 U115 BF10 P8,1"
        COLOR 1
        LOCATE 13, 21
```

```
        PRINT "Shelling to DOS.  Type 'EXIT' to return."
        LOCATE 15, 26
        PRINT "PRESS ANY KEY TO EXIT TO DOS"
500     A$ = INKEY$: IF A$ = "" THEN 500
        PALETTE                                          'Reset palette colors
        CLS
        SHELL
END SUB

SUB EMG (PICKED) STATIC

'******************************************************************
'*
'*      EMG is a routine which allows the user to alter the % contribution
'*      of each muscle to the bite.
'*
'******************************************************************

        DIM GREYBOX(50), BLUEBOX(50), WHITEBOX(50)
7000    CLS
        COLOR 2
        HEADER "EMG Selection"

'****
'*              Retrieve EMG values
'****

        COLOR 1
        IF FLAG <> 1 THEN                                'Read them only once
          TOTALEMG = 0                                   'Initialize counter
          OPEN "C:\QB\TMJ\DATA\EMG.DAT" FOR INPUT AS #1
          FOR I = 1 TO 14
                  INPUT #1, VALS(I)
                  IF VALS(I) < 0 THEN VALS(I) = 0
                  TOTALEMG = TOTALEMG + VALS(I)          'Increase the counter
          NEXT I
          CLOSE #1
          FLAG = 1
        END IF

'****
'*              Print Out Axis Labels
'****

        LOCATE 13, 2
        PRINT "Muscle"
        LOCATE 14, 2
        PRINT "Force"
        LOCATE 15, 3
        PRINT "(%)"
        LOCATE 21, 8
        PRINT " Post.   Middle   Anterior  Deep    Super.   Medial   Lateral"
        LOCATE 22, 8
        PRINT " Temp.   Temp.    Temp.     Masseter Masseter Pterygoid Pterygoid"
        FOR I = 1 TO 8
          X = (55 + (I - 1) * 80)
          LINE (X, 281)-(X, 308), 2
        NEXT I

'****
'*              Open Graphics Viewport and Window
'****

        VIEW (56, 70)-(614, 280), 0, 2
        WINDOW (0, 0)-(14, 15)

'****
'*              Create the Graphic Building Blocks
'****

        GET (0, 0)-(.97, .1), GREYBOX                    'Background block
        IF VALS(1) < 14 THEN                             'Check if within-
          LINE (0, 0)-(.97, VALS(1)), 3, BF              'Viewport range
        ELSE
          LINE (0, 0)-(.97, 14), 3, BF
        END IF
        GET (0, 0)-(.97, .1), BLUEBOX                    'Blue bar block
        IF VALS(2) < 14 THEN
          LINE (1, 0)-(1.97, VALS(2)), 1, BF
        ELSE
          LINE (1, 0)-(1.97, 14), 1, BF
        END IF
        GET (1, 0)-(1.97, .1), WHITEBOX                  'White bar block
```

```
'****
'*              Draw in intial bars
'****

        FOR I = 1 TO 6
           X = 2 + (I - 1) * 2
           IF VALS(X + 1) < 14 THEN
                       LINE (X, 0)-((X + .97), VALS(X + 1)), 3, BF
           ELSE
                       LINE (X, 0)-((X + .97), 14), 3, BF
           END IF
           IF VALS(X + 2) < 14 THEN
                       LINE ((X + 1), 0)-((X + 1.96), VALS(X + 2)), 1, BF
           ELSE
                       LINE ((X + 1), 0)-((X + 1.96), 14), 1, BF
           END IF
        NEXT I

'****
'*              Draw in Legend
'****

        LINE (10, 12)-(11, 13), 1, BF
        LINE (10, 13)-(11, 14), 3, BF
        LOCATE 7, 64: PRINT "Ipsilateral"
        LOCATE 8, 64: PRINT "Contralateral"

'****
'*              Print bar values to chart
'****

        FOR I = 1 TO 14
           IF VALS(I) < 14 THEN
                       LOCATE (20 - (FIX(VALS(I) + 1))), (8 + (I - 1) * 5)
           ELSE
                       LOCATE 6, (8 + (I - 1) * 5)
           END IF
           PRINT MID$(STR$(VALS(I)), 2, 4)
        NEXT I
        LOCATE 6, 2
        PRINT "Total"
        COLOR 13
        LOCATE 7, 2
        PRINT USING "###.#"; TOTALEMG                    'Display current value
        COLOR 1                                          'of TOTALEMG

'****
'*              Display message
'****

        COLOR 13
        LOCATE 23, 12
        PRINT "Use the ";
        COLOR 1
        PRINT CHR$(27); CHR$(26);
        COLOR 13
        PRINT " keys to select and the ";
        COLOR 1
        PRINT CHR$(18);
        COLOR 13
        PRINT " keys to change the values."
7010    LOCATE 24, 27
        PRINT "Press ";
        COLOR 1
        PRINT "<ENTER>";
        COLOR 13
        PRINT " to continue...";
        COLOR 1

'****
'*              Scan input to manipulate bar chart
'****

        BAR = 1
        COLOR 13
        LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
        PRINT MID$(STR$(VALS(BAR)), 2, 4)
7020    PICK$ = INKEY$
        IF PICK$ = "" THEN
           UPDATE                                        'Update the clock
           GOTO 7020
        END IF
```

```
        PICK$ = RIGHT$(PICK$, 1)
        ASCII = ASC(PICK$)

'****
'*              If 'UP' is pressed
'****

        IF ASCII = 72 THEN
          IF VALS(BAR) > 13.9 THEN
                    LOCATE 6, (8 + (BAR - 1) * 5)
                    PRINT SPACE$(4)
                    VALS(BAR) = (CINT(10 * (VALS(BAR) + .1))) / 10
                    TOTALEMG = TOTALEMG + .1
                    COLOR 13
                    LOCATE 7, 2
                    PRINT USING "###.#"; TOTALEMG
                    LOCATE 6, (8 + (BAR - 1) * 5)
                    PRINT MID$(STR$(VALS(BAR)), 2, 4)
                    GOTO 7020
          ELSE
                    LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                    PRINT SPACE$(4)
                    VALS(BAR) = (CINT(10 * (VALS(BAR) + .1))) / 10
                    TOTALEMG = TOTALEMG + .1
                    COLOR 13
                    LOCATE 7, 2
                    PRINT USING "###.#"; TOTALEMG
                    IF VALS(BAR) < 14 THEN
                      IF INT(BAR / 2) = (BAR / 2) THEN
                              PUT ((BAR - 1), VALS(BAR)), WHITEBOX, PSET
                      ELSE
                              PUT ((BAR - 1), VALS(BAR)), BLUEBOX, PSET
                      END IF
                      LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                      PRINT MID$(STR$(VALS(BAR)), 2, 4)
                      GOTO 7020
                    ELSE
                      IF INT(BAR / 2) = (BAR / 2) THEN
                              PUT ((BAR - 1), 14), WHITEBOX, PSET
                      ELSE
                              PUT ((BAR - 1), 14), BLUEBOX, PSET
                      END IF
                      LOCATE 6, (8 + (BAR - 1) * 5)
                      PRINT MID$(STR$(VALS(BAR)), 2, 4)
                      GOTO 7020
                    END IF
          END IF
        END IF

'****
'*              If 'DOWN' is pressed
'****

        IF ASCII = 80 THEN
          IF VALS(BAR) > 13.9 THEN
                    LOCATE 6, (8 + (BAR - 1) * 5)
                    PRINT SPACE$(4)
                    VALS(BAR) = (CINT(10 * (VALS(BAR) - .1))) / 10
                    TOTALEMG = TOTALEMG - .1
                    COLOR 13
                    LOCATE 7, 2
                    PRINT USING "###.#"; TOTALEMG
                    IF VALS(BAR) < 14 THEN
                      PUT ((BAR - 1), (VALS(BAR) + .1)), GREYBOX, PSET
                      LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                      PRINT MID$(STR$(VALS(BAR)), 2, 4)
                    ELSE
                      LOCATE 6, (8 + (BAR - 1) * 5)
                      PRINT MID$(STR$(VALS(BAR)), 2, 4)
                    END IF
                    GOTO 7020
          ELSE
                    LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                    PRINT SPACE$(4)
                    PUT ((BAR - 1), VALS(BAR)), GREYBOX, PSET
                    VALS(BAR) = (CINT(10 * (VALS(BAR) - .1))) / 10
                    TOTALEMG = TOTALEMG - .1
                    COLOR 13
                    LOCATE 7, 2
                    PRINT USING "###.#"; TOTALEMG
                    LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                    PRINT MID$(STR$(VALS(BAR)), 2, 4)
                    GOTO 7020
```

```
        END IF
    END IF

'****
'*          If 'LEFT' is pressed
'****

    IF ASCII = 75 THEN
        COLOR 1
        IF VALS(BAR) < 14 THEN
                LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
        ELSE
                LOCATE 6, (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
        END IF
        BAR = BAR - 1
        IF BAR = 0 THEN BAR = 14
        COLOR 13
        IF VALS(BAR) < 14 THEN
                LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
                GOTO 7020
        ELSE
                LOCATE 6, (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
                GOTO 7020
        END IF
    END IF

'****
'*          If 'RIGHT' is pressed
'****

    IF ASCII = 77 THEN
        COLOR 1
        IF VALS(BAR) < 14 THEN
                LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
        ELSE
                LOCATE 6, (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
        END IF
        BAR = BAR + 1
        IF BAR = 15 THEN BAR = 1
        COLOR 13
        IF VALS(BAR) < 14 THEN
                LOCATE (20 - (FIX(VALS(BAR) + 1))), (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
                GOTO 7020
        ELSE
                LOCATE 6, (8 + (BAR - 1) * 5)
                PRINT MID$(STR$(VALS(BAR)), 2, 4)
                GOTO 7020
        END IF
    END IF

'****
'*          If 'F1' pressed reset to original EMGVALS
'****

    IF ASCII = 59 THEN
        FLAG = 0
        GOTO 7000
    END IF

'****
'*          If 'RETURN', 'ESCAPE' or anything else is pressed
'****

    IF ASCII = 13 THEN
        IF (CINT(TOTALEMG * 10) / 10) <> 100! THEN                'Ensure TOTALEMG=100%
                BEEP
                LOCATE 24, 25
                COLOR 13
                PRINT "Total EMG value must equal ";
                COLOR 1
                PRINT "100%";
                COLOR 13
                FOR DELAY = 1 TO 10000: NEXT DELAY
                LOCATE 24, 25
                PRINT SPACE$(40);
                GOTO 7010
```

```
            END IF
            IF PICKED > 5 THEN                                          'If bite is on a right
                        FOR I = 1 TO 14                                 'tooth set right-side
                          EMGVALS(I) = VALS(I)                          'Values as Ipsilateral
                        NEXT I
            ELSE                                                        'If bite is on a left
                        FOR J = 1 TO 13 STEP 2                          'tooth set left-side
                          EMGVALS(J) = VALS(J + 1)                      'Values to Ipsilateral
                          EMGVALS(J + 1) = VALS(J)
                        NEXT J
            END IF
            COLOR 1
            GOTO 7030
         ELSE
            IF ASCII = 27 THEN                                          'If 'ESCAPE' is pressed
                        COLOR 1
                        GOTO 7030
            END IF
            GOTO 7020                                                   'Pressing anything else
         END IF

'****
'*               Reset the viewing screen
'****

7030     VIEW
         WINDOW
         CLS

END SUB

SUB FRANKFORT (X(), Y(), Z(), COMMENT$())

'*********************************************************************
'*
'*    FRANKFORT is a routine that processes the imported raw data file
'*    so that it is upright, centered along the sagital plane and with
'*    its horizontal plane corresponding to the Frankfort plane.
'*
'*********************************************************************

'****
'*               Stand the skull 'UPRIGHT'
'****

        FOR I = 1 TO 112
          X(I) = X(I) * -1
          Z(I) = Z(I) * -1
        NEXT I

'****
'*               To locate the Frankfort plane we use 4 points:
'*               the right and left inferior border of orbit and
'*               the right and left superior border of the
'*               auditory meatus.
'*
'*               A plane can be defined using '3' points therefore
'*               we need to define the third point as the midpoint
'*               of a line connecting the right and left superior
'*               border of the auditory meatus.
'****

        X(113) = (X(18) + X(59)) / 2
        Y(113) = (Y(18) + Y(59)) / 2
        Z(113) = (Z(18) + Z(59)) / 2

'****
'*               Proceed by defining the right inferior border
'*               of orbit as the origin for rotations.
'*               Make the R.INF.BORD.ORB. (0,0,0)
'****

        XSTEP = X(5)
        YSTEP = Y(5)
        ZSTEP = Z(5)
        FOR I = 1 TO 113
          X(I) = X(I) - XSTEP
          Y(I) = Y(I) - YSTEP
          Z(I) = Z(I) - ZSTEP
        NEXT I

'****
'*               First rotate L.INF.BORD.ORB. in X,Z plane for Z=0
```

```
'****
        HORZ = X(47)
        VERT = Z(47)
        ROTATE HORZ, VERT, X(), Z()

'****
'               Rotate the L.INF.BORD.ORB in X,Y plane for Y=0
'****

        HORZ = X(47)
        VERT = Y(47)
        ROTATE HORZ, VERT, X(), Y()

'****
'               Rotate point 113 in Y,Z plane for X=0
'****

        HORZ = Y(113)
        VERT = Z(113)
        ROTATE HORZ, VERT, Y(), Z()

'****
'               Move the origin to the point on the Frankfurt
'               Plane directly beneath the ANTERIOR NASAL SPINE
'****

        XSTEP = X(3)
        YSTEP = Y(3)
        FOR I = 1 TO 113
          X(I) = X(I) - XSTEP
          Y(I) = Y(I) - YSTEP
        NEXT I

'****
'               Rotate the data set so that the POSTERIOR
'               NASAL SPINE lies in the X,Y plane with X=0
'****

        HORZ = Y(13)
        VERT = X(13)
        ROTATE HORZ, VERT, Y(), X()
END SUB

SUB GETPOINT (START, LAST, START$, PICKED) STATIC

'*******************************************************************
'
'       GETPOINT is a Routine that prompts the user to define the filerange
'       and biteforce location for calculating the resultant condylar loads.
'
'*******************************************************************

'****
'               Setup arrays
'****

        DIM EMPTYBOX(77), FULLBOX(77)
        BITEPOINT$(1, 1) = "LEFT FIRST PREMOLAR": BITEPOINT$(1, 2) = "LPM1"
        BITEPOINT$(2, 1) = "LEFT SECOND PREMOLAR": BITEPOINT$(2, 2) = "LPM2"
        BITEPOINT$(3, 1) = "LEFT FIRST MOLAR": BITEPOINT$(3, 2) = "LM1"
        BITEPOINT$(4, 1) = "LEFT SECOND MOLAR": BITEPOINT$(4, 2) = "LM2"
        BITEPOINT$(5, 1) = "LEFT THIRD MOLAR": BITEPOINT$(5, 2) = "LM3"
        BITEPOINT$(6, 1) = "RIGHT FIRST PREMOLAR": BITEPOINT$(6, 2) = "RPM1"
        BITEPOINT$(7, 1) = "RIGHT SECOND PREMOLAR": BITEPOINT$(7, 2) = "RPM2"
        BITEPOINT$(8, 1) = "RIGHT FIRST MOLAR": BITEPOINT$(8, 2) = "RM1"
        BITEPOINT$(9, 1) = "RIGHT SECOND MOLAR": BITEPOINT$(9, 2) = "RM2"
        BITEPOINT$(10, 1) = "RIGHT THIRD MOLAR": BITEPOINT$(10, 2) = "RM3"

'****
'               Setup screen
'****

6000    CLS
        COLOR 2
        HEADER "Resultant Condylar Load Calculation"
        LOCATE 6, 2
        PRINT "Filerange"
        COLOR 1
        LOCATE 7, 20
        PRINT "First Skull Number (i.e. 001):  "; START$
        LOCATE 8, 20
        PRINT "Last Skull Number <First>:       "; LAST$
```

```
        COLOR 2
        LOCATE 12, 2
        PRINT "Bite Force Location"
        COLOR 1
        DRAW "C2 BM225,168 R12 D12 L12 U11 R11 D10 L10 U10"
        GET (225, 162)-(251, 180), EMPTYBOX
        FOR I = 1 TO 5
           LOCATE (13 + ((I - 1) * 2)), 7
           PRINT BITEPOINT$(I, 1); TAB(33); BITEPOINT$(I, 2); TAB(46);
           PRINT BITEPOINT$(I + 5), 1); TAB(72); BITEPOINT$((I + 5), 2)
           PUT (225, (162 + (I - 1) * 28)), EMPTYBOX, PSET
           PUT (537, (162 + (I - 1) * 28)), EMPTYBOX, PSET
        NEXT I

'****
'*           Input the first file to be converted
'****

6010    COLOR 2
        LOCATE 7, 52
        INPUT "", START$
        IF START$ = "" THEN GOTO 6010                    'Check for null entry
        START = VAL(START$)
        IF START < 0 OR START > 999 THEN 6010            'Check if out of range
        IF START = 0 THEN
           LAST$ = START$
           LOCATE 8, 52
           PRINT LAST$
           GOTO 6030
        END IF

'****
'*           Input the last file to be converted
'****

6020    LOCATE 8, 52
        INPUT "", LAST$
        IF LAST$ = "" THEN
           LAST$ = START$                                'Set default to START$
           LOCATE 8, 52
           PRINT LAST$
        END IF
6030    LAST = VAL(LAST$)
        IF LAST < 0 OR LAST > 999 OR LAST < START THEN 6020
                                                         'Check for valid value
'****
'*           Look for non-numeric filenames
'****

        IF LAST$ = START$ THEN
           START = 1000
           LAST = 1000
           COLOR 8
           LOCATE 10, 16
           PRINT "You have chosen to work with file TMJ"; UCASE$(START$); ".MOD "
           GOTO 6040
        END IF                                           'Set a flag

'****
'*           Verify inputs
'****

        COLOR 8
        LOCATE 10, 16
        PRINT "You have chosen files TMJ"; UCASE$(START$); ".MOD to TMJ"; UCASE$(LAST$); ".MOD"

'****
'*           Display Message
'****

6040    COLOR 13
        LOCATE 23, 8
        PRINT "Press ";
        COLOR 1
        PRINT CHR$(27); CHR$(18); CHR$(26);
        COLOR 13
        PRINT " to move, the ";
        COLOR 1
        PRINT "SPACEBAR";
        COLOR 13
        PRINT " to select or ";
        COLOR 1
        PRINT "RETURN";
```

```
        COLOR 13
        PRINT " to continue..."
        COLOR 1

'****
'*              Draw checkmark in FULLBOX
'****

        DRAW "C1 BM229,172 M+5,+2 M+10,-12 M+6,0 M-2,+2 M-3,0 M-9,+13 M-7,-2 M229,172 BF2 P1,1"
        GET (225, 162)-(251, 180), FULLBOX

'****
'*              Scan inputs for menu scrolling
'****

        IF BITEPOINT = 0 THEN BITEPOINT = 1
        IF OLDPOINT = 0 THEN OLDPOINT = 1
        PICKED = 1
6050    IF OLDPOINT <= 5 THEN                          'If the old point is in
          LOCATE (13 + (OLDPOINT - 1) * 2), 7          'the left-hand column
          COLOR 1
          PRINT BITEPOINT$(OLDPOINT, 1);
          LOCATE , 33
          PRINT BITEPOINT$(OLDPOINT, 2)
        ELSE                                           'The right-hand column
          LOCATE (13 + (OLDPOINT - 6) * 2), 46
          COLOR 1
          PRINT BITEPOINT$(OLDPOINT, 1);
          LOCATE , 72
          PRINT BITEPOINT$(OLDPOINT, 2)
        END IF
        IF BITEPOINT <= 5 THEN                         'If the new point is in
          LOCATE (13 + (BITEPOINT - 1) * 2), 7         'the left-hand column
          COLOR 13
          PRINT BITEPOINT$(BITEPOINT, 1);
          LOCATE , 33
          PRINT BITEPOINT$(BITEPOINT, 2)
        ELSE                                           'The right-hand column
          LOCATE (13 + (BITEPOINT - 6) * 2), 46
          COLOR 13
          PRINT BITEPOINT$(BITEPOINT, 1);
          LOCATE , 72
          PRINT BITEPOINT$(BITEPOINT, 2)
        END IF
6060    PICK$ = INKEY$
        IF PICK$ = "" THEN
          UPDATE                                       'Update the clock
          GOTO 6060
        END IF
        PICK$ = RIGHT$(PICK$, 1)
        ASCII = ASC(PICK$)
        IF ASCII = 72 THEN                             'If 'UP' is pressed
          OLDPOINT = BITEPOINT
          BITEPOINT = BITEPOINT - 1
          IF BITEPOINT = 5 THEN BITEPOINT = 10
          IF BITEPOINT = 0 THEN BITEPOINT = 5
          GOTO 6050
        END IF
        IF ASCII = 80 THEN                             'If 'DOWN' is pressed
          OLDPOINT = BITEPOINT
          BITEPOINT = BITEPOINT + 1
          IF BITEPOINT = 6 THEN BITEPOINT = 1
          IF BITEPOINT = 11 THEN BITEPOINT = 6
          GOTO 6050
        END IF
        IF ASCII = 75 THEN                             'If 'LEFT' is pressed
          OLDPOINT = BITEPOINT
          BITEPOINT = BITEPOINT - 5
          IF BITEPOINT < 1 THEN BITEPOINT = BITEPOINT + 10
          GOTO 6050
        END IF
        IF ASCII = 77 THEN                             'If 'RIGHT' is pressed
          OLDPOINT = BITEPOINT
          BITEPOINT = BITEPOINT + 5
          IF BITEPOINT > 10 THEN BITEPOINT = BITEPOINT - 10
          GOTO 6050
        END IF
        IF ASCII = 32 THEN                             'If 'SPACEBAR' pressed
          IF PICKED < 6 THEN
                  PUT (225, (162 + (PICKED - 1) * 28)), EMPTYBOX, PSET
          ELSE
                  PUT (537, (162 + (PICKED - 6) * 28)), EMPTYBOX, PSET
          END IF
```

```
            PICKED = BITEPOINT
            IF PICKED < 6 THEN
                      PUT (225, (162 + (PICKED - 1) * 28)), FULLBOX, PSET
            ELSE
                      PUT (537, (162 + (PICKED - 6) * 28)), FULLBOX, PSET
            END IF
            GOTO 6050
         END IF
         IF ASCII = 13 THEN                                          'If 'RETURN' is pressed
            EXIT SUB
         ELSE
            IF ASCII = 27 THEN                                       'If 'ESCAPE' is pressed
                      EXIT SUB
            END IF
            GOTO 6050                                                'Pressing anything else
         END IF
END SUB

SUB GETVECTORS (VECTORS())

'**********************************************************************
'*
'*       GETVECTORS is a routine to retrieve the 14 muscle unitvectors and
'*       attachment points from the modified datafiles.
'*
'**********************************************************************

         FOR J = 1 TO 14
            INPUT #1, A$, B, C, D, E, F, G
            VECTORS(J, 1) = B
            VECTORS(J, 2) = C
            VECTORS(J, 3) = D
            VECTORS(J, 4) = E
            VECTORS(J, 5) = F
            VECTORS(J, 6) = G
         NEXT J
END SUB

SUB HEADER (TITLE$)

'**********************************************************************
'*
'*       Header is a routine which draws the nice looking graphics at
'*       the top of the screen. It displays the date, time and the section
'*       of the program currently running, passed to it. It uses screen
'*       attribute 2.
'*
'**********************************************************************

         LINE (0, 0)-(638, 0), 2
         LINE -STEP(0, 348), 2
         LINE -STEP(-638, 0), 2
         LINE -STEP(0, -348), 2
         LINE (0, 35)-(638, 35), 2
         LINE (0, 62)-(638, 62), 2
         LINE (130, 0)-(130, 35), 2
         LINE (508, 0)-(508, 35), 2
         LOCATE 2, 4: PRINT DATE$
         LOCATE 2, 24: PRINT "TEMPOROMANDIBULAR FORCE ANALYSIS"
         LOCATE 2, 68: PRINT TIME$
         CENTER TITLE$, 4
END SUB

SUB MODDATAREAD (CLASS$, OFFSET, ANB, X(), Y(), Z(), COMMENT$())

'**********************************************************************
'*
'*       MODDATAREAD is a routine which reads the modified datafile into the
'*       arrays X(), Y(), Z() and COMMENT$(). It includes several error
'*       checks.
'*
'**********************************************************************

         INPUT #1, CLASS, ANB
         CLASS$ = LTRIM$(STR$(CLASS))
         INPUT #1, OFFSET
         FOR K = 1 TO 113
            INPUT #1, A, B, C, D, E$

'****
'*                Check that file locations match the array locations
'****
```

```
          IF K <> A THEN
                    BEEP
                    COLOR 13
                    LOCATE 15, 25
                    PRINT "READ LOOP LOCATION MISMATCH, READ TERMINATED!"
                    COLOR 2
                    FOR DELAY = 1 TO 20000: NEXT DELAY
                    LOCATE 15, 1
                    PRINT SPACE$(75)
                    BEEP
                    EXIT SUB
          END IF

'****
'*              Get the values of X, Y and Z and store them.
'****

          X(K) = B                                              'X DIR READINGS ARRAY
          Y(K) = C                                              'Y DIR READINGS ARRAY
          Z(K) = D                                              'Z DIR READINGS ARRAY
          COMMENTS$(K) = E$                                     'COMMENTS ARRAY
        NEXT K
END SUB

SUB MOVEORIGIN (X(), Y(), Z(), VECTORS())

'*********************************************************************
'*
'*      MOVEORIGIN is a routine which moves the (0,0,0) origin to the
'*      force application point on the right condyle. It first has to
'*      calculate the left and right force application points based on
'*      several logical assumptions. Moving the origin to the point
'*      where three of the unknowns pass through (Rx, Ry and Rz) we make
'*      life just a little easier by eliminating them from the moment
'*      equations.
'*         The condylar force application points are found by bisecting
'*      a vector between digitized fossa point #2 and #4.
'*
'*********************************************************************

        X(114) = (X(15) + X(17)) / 2                            'Right cond. X appl. pt
        Y(114) = (Y(15) + Y(17)) / 2                            'Right cond. Y appl. pt
        Z(114) = (Z(15) + Z(17)) / 2                            'Right cond. Z appl. pt
        X(115) = (X(56) + X(58)) / 2                            'Left cond. X appl. pt
        Y(115) = (Y(56) + Y(58)) / 2                            'Left cond. Y appl. pt
        Z(115) = (Z(56) + Z(58)) / 2                            'Left cond. Z appl. pt

'****
'*              Proceed by defining the right condylar force
'*              application point as the origin.
'*              Make the R.C.F.A.P. (0,0,0)
'****

        XSTEP = X(114)
        YSTEP = Y(114)
        ZSTEP = Z(114)
        FOR I = 1 TO 115
          X(I) = X(I) - XSTEP
          Y(I) = Y(I) - YSTEP
          Z(I) = Z(I) - ZSTEP
        NEXT I

'****
'*              Translate the muscle vector origins
'****

        FOR J = 1 TO 14
          VECTORS(J, 1) = VECTORS(J, 1) - XSTEP
          VECTORS(J, 2) = VECTORS(J, 2) - YSTEP
          VECTORS(J, 3) = VECTORS(J, 3) - ZSTEP
        NEXT J
END SUB

SUB MTRIX

'*********************************************************************
'*
'*      MTRIX is a matrix solver routine that returns the values of the
'*      7 unknowns in the 7x8 augmented matrix passed to it in the MATRIX
'*      array. The values of the unknowns are returned in the FORCES array.
'*
'*********************************************************************
```

```
        ROW = 7                                                  The # of matrix rows
        COL = 8                                                  The # of matrix column
        LAST = ROW - 1

'****
'           Start overall loop for (ROW-1) pivots
'****

        FOR I = 1 TO LAST

'****
'           Find the largest remaining term in the I-th column
'           for pivot
'****

          BIG = 0
          FOR K = I TO ROW
                TERM = ABS(MATRIX(K, I))
                IF (TERM - BIG) < 0 OR (TERM - BIG) = 0 THEN 6510 ELSE 6500
6500            BIG = TERM
                L = K
6510      NEXT K

'****
'           Check whether a non-zero term has been found
'****

          IF BIG < 0 OR BIG > 0 THEN 6530 ELSE 6520
6520      GOTO 6580

'****
'           L-th row has the biggest term -- is I = L
'****

6530      IF (I - L) < 0 OR (I - L) > 0 THEN 6540 ELSE 6550

'****
'           I is not equal to L so switch rows I and L
'****

6540      FOR J = 1 TO COL
                TEMP = MATRIX(I, J)
                MATRIX(I, J) = MATRIX(L, J)
                MATRIX(L, J) = TEMP
          NEXT J

'****
'           Now start pivotal reduction
'****

6550      PIVOT = MATRIX(I, I)
          NEXTR = I + 1

'****
'           For each of the rows after the I-th
'****

          FOR J = NEXTR TO ROW

'****
'           Multiplying constant for the J-th row is...
'****

                CONNST = MATRIX(J, I) / PIVOT

'****
'           Now reduce each term of the J=th row
'****

                FOR K = I TO COL
                   MATRIX(J, K) = MATRIX(J, K) - CONNST * MATRIX(I, K)
                NEXT K
          NEXT J
        NEXT I

'****
'
'           End of pivotal reduction,
'           now perform back substitution
'****

        FOR I = 1 TO ROW

'****
```

```
'
''''        IREV is the backward index, going from ROWback to 1

        IREV = ROW + 1 - I

''''
'           Get Y(IREV) in preparation
''''

        Y = MATRIX(IREV, COL)
        IF (IREV - ROW) < 0 OR (IREV - ROW) > 0 THEN 6560 ELSE 6570

''''
'           Not working on last row, I is 2 or greater
''''

6560    FOR J = 2 TO I

''''
'           Work backwards for FORCES(N), FORCES(N-1)---,
'           substituting previously found values
''''

              K = COL + 1 - J
              Y = Y - MATRIX(IREV, K) * FORCES(K)
        NEXT J

''''
'           Finally compute FORCES(IREV)
''''

6570      FORCES(IREV) = Y / MATRIX(IREV, IREV)
        NEXT
6580    EXIT SUB
END SUB

SUB OCCLUSALPLANE (X(), Y(), Z(), OFFSET, PICKED)

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'       OCCLUSALPLANE is a very important routine.  Its primary function
'       is to return the location of the Bite Force's origin and its
'       corresponding unit vector.
'          To do this we start by defining the occlusal plane from
'       the available digitized locations on the lingual cusps of the
'       left and right first molars and the left and right first
'       premolars (bicuspids).  The bicuspid locations are averaged to
'       contribute the third nessecary point to define the plane.
'          The bite force origin is located as a single point lying
'       on the occlusal plane directly above/beneath the chosen bite
'       location.  The bite location on the chosen tooth is presently
'       taken as the tooths central fossa, point 5 in the digitized
'       surface.  The single point on the occlusal plane and the
'       coordinate of point 5 are then used to calculate the unit vector.
'          Since we digitized only the teeth in the Maxilla (Upper Jaw)
'       but the bite force was to be applied through the teeth on the
'       Mandible (Lower Jaw) it was nessecary to develop a method to
'       transpose the maxillary teeth to reflect the bite locations
'       of the mandibular teeth.
'          This is accomplished in a two step shift.  After finding the
'       occlusal plane and bite force origin, the origin is
'       shifted anteriorly/posteriorly an amount determined by OFFSET.
'       OFFSET was calculated as the ACTUAL front/rear displacement of
'       the lower first molar measial buccal cusp tip with that of the
'       same location on the upper first molar.  This distance was
'       measured off Lateral SEF's (X-ray's) for skulls 1-40 and by hand
'       from the actual skulls for 41-57.  This value was included at
'       the top of the skull's .MOD datafile.
'          The second shift is lingually (towards the tongue) to
'       account for the fact that the mandible is always smaller than
'       the maxilla.  This amount is determined from the lingual shift
'       from the upper first molars measial buccal cusp tip to it's
'       central fossa.  The value of lingual shift used was taken as
'       the average of this value from both sides.
'         Once this was completed the origin was rotated back to
'       original (X,Y,Z) axis and the unit vector calulated and the
'       origin and vector values stored in the BYTEPOINT() array.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        DIM LOCATION(5, 3), INSERT(3)

''''
```

```
'*
'*         Assign the proper location number to the chosen
'*         bite location.  This is the central fossa of the
'*         selected tooth, point #5.
'****

      IF PICKED = 1 THEN PNT = 87: GOTO 9000          'Left first premolar
      IF PICKED = 2 THEN PNT = 82: GOTO 9000          'Left second premolar
      IF PICKED = 3 THEN PNT = 76: GOTO 9000          'Left first molar
      IF PICKED = 4 THEN PNT = 71: GOTO 9000          'Left second molar
      IF PICKED = 5 THEN PNT = 66: GOTO 9000          'left third molar
      IF PICKED = 6 THEN PNT = 46: GOTO 9000          'Right first premolar
      IF PICKED = 7 THEN PNT = 41: GOTO 9000          'Right second premolar
      IF PICKED = 8 THEN PNT = 35: GOTO 9000          'Right first molar
      IF PICKED = 9 THEN PNT = 30: GOTO 9000          'Right second molar
      IF PICKED = 10 THEN PNT = 25: GOTO 9000         'Right third molar

'****
'*
'*         Fill array LOCATION(1,?) with the X,Y,Z location
'*         of the tooth's central fossa.  In slang I'll
'*         call this location the insert.
'****

9000  LOCATION(1, 1) = X(PNT)
      LOCATION(1, 2) = Y(PNT)
      LOCATION(1, 3) = Z(PNT)

'****
'*
'*         Calculate the three locations used to define the
'*         Occlusal plane.
'*         LOCATION(2,?) = Location on right first molar
'*         LOCATION(3,?) = Location on left first molar
'*         LOCATION(4,?) = Location between left and right
'*                         first premolars
'****

      LOCATION(2, 1) = (X(33) + X(34)) / 2
      LOCATION(2, 2) = (Y(33) + Y(34)) / 2
      LOCATION(2, 3) = (Z(33) + Z(34)) / 2
      LOCATION(3, 1) = (X(74) + X(75)) / 2
      LOCATION(3, 2) = (Y(74) + Y(75)) / 2
      LOCATION(3, 3) = (Z(74) + Z(75)) / 2
      LOCATION(4, 1) = (X(44) + X(45) + X(85) + X(86)) / 4
      LOCATION(4, 2) = (Y(44) + Y(45) + Y(85) + Y(86)) / 4
      LOCATION(4, 3) = (Z(44) + Z(45) + Z(85) + Z(86)) / 4

'****
'*         Calculate the LINGUAL shift for the appropriate side
'****

      LINGUAL = (ABS(X(73) - X(76)) + ABS(X(32) - X(35))) / 2
      IF PICKED < 6 THEN LINGUAL = LINGUAL * -1      'For bite on the left

'****
'*
'*         Move the origin to the right first molars point
'*         on the occlusal plane. Make it 0,0,0.
'****

      XSTEP = LOCATION(2, 1)
      YSTEP = LOCATION(2, 2)
      ZSTEP = LOCATION(2, 3)
      FOR I = 1 TO 4
        LOCATION(I, 1) = LOCATION(I, 1) - XSTEP
        LOCATION(I, 2) = LOCATION(I, 2) - YSTEP
        LOCATION(I, 3) = LOCATION(I, 3) - ZSTEP
      NEXT I

'****
'*
'*         Rotate the LOCATION() dataset through DELTATHETA1
'*         degrees in the X-Z plane so that the left first
'*         molar location is lying on the X-Y plane.
'****

      HORZ = LOCATION(3, 1)
      IF HORZ = 0 THEN HORZ = .000001
      VERT = LOCATION(3, 3)
      IF VERT = 0 THEN 9010
      DELTATHETA1 = ATN(VERT / HORZ)
      TURN DELTATHETA1, 1, 3, LOCATION()

'****
'*
'*         Rotate the LOCATION() dataset through DELTATHETA2
'*         degrees in the X-Y plane so that the left first
'*         molar location is on the X axis.
```

```
'****
9010    HORZ = LOCATION(3, 1)
        IF HORZ = 0 THEN HORZ = .000001
        VERT = LOCATION(3, 2)
        IF VERT = 0 THEN 9020
        DELTATHETA2 = ATN(VERT / HORZ)
        TURN DELTATHETA2, 1, 2, LOCATION()
'****
'           Rotate the LOCATION() dataset through DELTATHETA3
'           degrees in the Y-Z plane so that the left and right
'           first premolar location is lying on the X-Y plane.
'****
9020    HORZ = LOCATION(4, 2)
        IF HORZ = 0 THEN HORZ = .000001
        VERT = LOCATION(4, 3)
        IF VERT = 0 THEN 9030
        DELTATHETA3 = ATN(VERT / HORZ)
        TURN DELTATHETA3, 2, 3, LOCATION()
'****
'           Rotate the LOCATION() dataset through -DELTATHETA2
'           degrees in the X-Y plane so that the left first
'           molar location is moved back to its original location
'****
9030    TURN -DELTATHETA2, 1, 2, LOCATION()
'****
'           Shift the Insert location by OFFSET and LINGUAL
'****
        LOCATION(1, 1) = LOCATION(1, 1) + LINGUAL
        LOCATION(1, 2) = LOCATION(1, 2) + OFFSET
'****
'           Locate the biteforce vectors origin as directly
'           beneath the Insert ON the occlusal plane.
'****
        LOCATION(5, 1) = LOCATION(1, 1)
        LOCATION(5, 2) = LOCATION(1, 2)
        LOCATION(5, 3) = 0
'****
'           Begin the move back to the original coordinate
'           system by rotating the LOCATION() dataset through
'           DELTATHETA2 degrees so the left first molar location
'           is on the X axis.
'****
        TURN DELTATHETA2, 1, 2, LOCATION()
'****
'           Rotate the LOCATION() dataset through -DELTATHETA3
'           degrees so the left and right first premolars
'           location is restored to its original Y-Z plane
'           position.
'****
        TURN -DELTATHETA3, 2, 3, LOCATION()
'****
'           Rotate the LOCATION() dataset through -DELTATHETA2
'           degrees so the left first molar location is restored
'           to its original X-Y plane position.
'****
        TURN -DELTATHETA2, 1, 2, LOCATION()
'****
'           Rotate the LOCATION() dataset through -DELTATHETA1
'           degrees so the left first molar location is restored
'           to its original X-Z plane position.
'****
        TURN -DELTATHETA1, 1, 3, LOCATION()
'****
'           Step the LOCATION() dataset back so that the right
'           condylar force application point is the origin again.
```

```
'****
        FOR I = 1 TO 5
          LOCATION(I, 1) = LOCATION(I, 1) + XSTEP
          LOCATION(I, 2) = LOCATION(I, 2) + YSTEP
          LOCATION(I, 3) = LOCATION(I, 3) + ZSTEP
        NEXT I
'****
'               Store the bitepoint origin in the array
'               BYTEPOINT(1-3), then calculate the unitvector i, j, k
'               between the origin and insert and store them in
'               BYTEPOINT(4-6).
'****

        BYTEPOINT(1) = LOCATION(5, 1)
        BYTEPOINT(2) = LOCATION(5, 2)
        BYTEPOINT(3) = LOCATION(5, 3)
        DELTAX = BYTEPOINT(1) - LOCATION(1, 1)
        DELTAY = BYTEPOINT(2) - LOCATION(1, 2)
        DELTAZ = BYTEPOINT(3) - LOCATION(1, 3)
        DENOMINATOR = SQR(DELTAX ^ 2 + DELTAY ^ 2 + DELTAZ ^ 2)
        BYTEPOINT(4) = DELTAX / DENOMINATOR
        BYTEPOINT(5) = DELTAY / DENOMINATOR
        BYTEPOINT(6) = DELTAZ / DENOMINATOR
END SUB

SUB RESULTANT

'*********************************************************************
'*
'*    This routine calculates the resultant condylar and bite forces
'*    caused during mastication for the active data set.
'*
'*********************************************************************

'****
'*              Nomenclature
'****

' Rx = RIGHT CONDYLAR REACTION FORCE IN THE X-DIRECTION
' Ry = RIGHT CONDYLAR REACTION FORCE IN THE Y-DIRECTION
' Rz = RIGHT CONDYLAR REACTION FORCE IN THE Z-DIRECTION
' Lx = LEFT CONDYLAR REACTION FORCE IN THE X-DIRECTION
' Ly = LEFT CONDYLAR REACTION FORCE IN THE Y-DIRECTION
' Lz = LEFT CONDYLAR REACTION FORCE IN THE Z-DIRECTION
' BFX = BITE FORCE X-DIRECTION COMPONENT
' BFY = BITE FORCE Y-DIRECTION COMPONENT
' BFZ = BITE FORCE Z-DIRECTION COMPONENT
' RSMX = RIGHT SUPERFICIAL MASSETER X-DIRECTION FORCE
' RSMY = RIGHT SUPERFICIAL MASSETER Y-DIRECTION FORCE
' RSMZ = RIGHT SUPERFICIAL MASSETER Z-DIRECTION FORCE
' LSMX = LEFT SUPERFICIAL MASSETER X-DIRECTION FORCE
' LSMY = LEFT SUPERFICIAL MASSETER Y-DIRECTION FORCE
' LSMZ = LEFT SUPERFICIAL MASSETER Z-DIRECTION FORCE
' RDMX = RIGHT DEEP MASSETER X-DIRECTION FORCE
' RDMY = RIGHT DEEP MASSETER Y-DIRECTION FORCE
' RDMZ = RIGHT DEEP MASSETER Z-DIRECTION FORCE
' LDMX = LEFT DEEP MASSETER X-DIRECTION FORCE
' LDMY = LEFT DEEP MASSETER Y-DIRECTION FORCE
' LDMZ = LEFT DEEP MASSETER Z-DIRECTION FORCE
' RLPX = RIGHT LATERAL PTERYGOID X-DIRECTION FORCE
' RLPY = RIGHT LATERAL PTERYGOID Y-DIRECTION FORCE
' RLPZ = RIGHT LATERAL PTERYGOID Z-DIRECTION FORCE
' LLPX = LEFT LATERAL PTERYGOID X-DIRECTION FORCE
' LLPY = LEFT LATERAL PTERYGOID Y-DIRECTION FORCE
' LLPZ = LEFT LATERAL PTERYGOID Z-DIRECTION FORCE
' RMPX = RIGHT MEDIAL PTERYGOID X-DIRECTION FORCE
' RMPY = RIGHT MEDIAL PTERYGOID Y-IRECTION FORCE
' RMPZ = RIGHT MEDIAL PTERYGOID Z-DIRECTION FORCE
' LMPX = LEFT MEDIAL PTERYGOID X-DIRECTION FORCE
' LMPY = LEFT MEDIAL PTERYGOID Y-DIRECTION FORCE
' LMPZ = LEFT MEDIAL PTERYGOID Z-DIRECTION FORCE
' RPTX = RIGHT POSTERIOR TEMPORALIS X-DIRECTION FORCE
' RPTY = RIGHT POSTERIOR TEMPORALIS Y-DIRECTION FORCE
' RPTZ = RIGHT POSTERIOR TEMPORALIS Z-DIRECTION FORCE
' LPTX = LEFT POSTERIOR TEMPORALIS X-DIRECTION FORCE
' LPTY = LEFT POSTERIOR TEMPORALIS Y-DIRECTION FORCE
' LPTZ = LEFT POSTERIOR TEMPORALIS Z-DIRECTION FORCE
' RMTX = RIGHT MIDDLE TEMPORALIS X-DIRECTION FORCE
' RMTY = RIGHT MIDDLE TEMPORALIS Y-DIRECTION FORCE
' RMTZ = RIGHT MIDDLE TEMPORALIS Z-DIRECTION FORCE
' LMTX = LEFT MIDDLE TEMPORALIS X-DIRECTION FORCE
```

```
' LMTY = LEFT MIDDLE TEMPORALIS Y-DIRECTION FORCE
' LMTZ = LEFT MIDDLE TEMPORALIS Z-DIRECTION FORCE
' RATX = RIGHT ANTERIOR TEMPORALIS X-DIRECTION FORCE
' RATY = RIGHT ANTERIOR TEMPORALIS Y-DIRECTION FORCE
' RATZ = RIGHT ANTERIOR TEMPORALIS Z-DIRECTION FORCE
' LATX = LEFT ANTERIOR TEMPORALIS X-DIRECTION FORCE
' LATY = LEFT ANTERIOR TEMPORALIS Y-DIRECTION FORCE
' LATZ = LEFT ANTERIOR TEMPORALIS Z-DIRECTION FORCE

'****
'*          Setup Storage Arrays
'****

     DIM X(115), Y(115), Z(115), COMMENTS$(115), VECTORS(14, 6)

'****
'*          Call the routine to determine bite location
'****

     GETPOINT START, LAST, START$, PICKED

'****
'*          Call the routine to select EMG data
'****

     EMG PICKED

'****
'*          Call the routine for batch description
'****

     BATCHSELECT PICKED

'****
'*          Setup the loop for multiple files
'****

     FOR NUM = START TO LAST

'****
'*          Build the filename
'****

       FILENUM$ = LTRIM$(STR$(NUM))
       IF NUM = 1000 THEN FILENUM$ = START$
       FILE$ = "TMJ" + FILENUM$ + ".MOD"
       FILE$ = UCASE$(FILE$)
       PATH$ = "C:\QB\TMJ\DATA\"
       FILENAME$ = PATH$ + FILE$
       OPEN FILENAME$ FOR INPUT AS #1

'****
'*          Setup the new screen
'****

       CLS
       COLOR 2
       HEADER "Resultant Force Calculation for " + FILE$
       COLOR 1
       CENTER "Yo, I'm busy thinking so please be patient and wait...", 12
       COLOR 2

'****
'*          Read the 113 locations from the modified datafile
'****

       MODDATAREAD CLASS$, OFFSET, ANB, X(), Y(), Z(), COMMENTS$()

'****
'*          Read the 14 unitvectors and origins
'****

       GETVECTORS VECTORS()
       CLOSE #1

'****
'*          Move origin to RIGHT CONDYLE
'****

       MOVEORIGIN X(), Y(), Z(), VECTORS()

'****
'*          Calculate the unitvector and origin for the
```

```
'*
'****                   selected bitepoint

              OCCLUSALPLANE X(), Y(), Z(), OFFSET, PICKED

'****
'*
'*                      Calculate the remaining matrix components.
'*                      Start with the individual muscle components.
'*                      The odd values of the EMGVALS() array are assumed
'*                      to contain the right side values based on
'****                   ipsa/contra lateral positioning

            RSMX = EMGVALS(9) * VEC[illegible]S(1, [illegible]
            RSMY = EMGVALS(9) * VECT[illegible]
            RSMZ = EMGVALS(9) * VECTO[illegible] 6[illegible]
            RDMX = EMGVALS(7) * VECTO[illegible]
            RDMY = EMGVALS(7) * VECTORS(2, 5)
            RDMZ = EMGVALS(7) * VECTORS(2, 6)
            RLPX = EMGVALS(13) * VECTORS(3, 4)
            RLPY = EMGVALS(13) * VECTORS(3, 5)
            RLPZ = EMGVALS(13) * VECTORS(3, 6)
            RMPX = EMGVALS(11) * VECTORS(4, 4)
            RMPY = EMGVALS(11) * VECTORS(4, 5)
            RMPZ = EMGVALS(11) * VECTORS(4, 6)
            RPTX = EMGVALS(1) * VECTORS(5, 4)
            RPTY = EMGVALS(1) * VECTORS(5, 5)
            RPTZ = EMGVALS(1) * VECTORS(5, 6)
            RMTX = EMGVALS(3) * VECTORS(6, 4)
            RMTY = EMGVALS(3) * VECTORS(6, 5)
            RMTZ = EMGVALS(3) * VECTORS(6, 6)
            RATX = EMGVALS(5) * VECTORS(7, 4)
            RATY = EMGVALS(5) * VECTORS(7, 5)
            RATZ = EMGVALS(5) * VECTORS(7, 6)
            LSMX = EMGVALS(10) * VECTORS(8, 4)
            LSMY = EMGVALS(10) * VECTORS(8, 5)
            LSMZ = EMGVALS(10) * VECTORS(8, 6)
            LDMX = EMGVALS(8) * VECTORS(9, 4)
            LDMY = EMGVALS(8) * VECTORS(9, 5)
            LDMZ = EMGVALS(8) * VECTORS(9, 6)
            LLPX = EMGVALS(14) * VECTORS(10, 4)
            LLPY = EMGVALS(14) * VECTORS(10, 5)
            LLPZ = EMGVALS(14) * VECTORS(10, 6)
            LMPX = EMGVALS(12) * VECTORS(11, 4)
            LMPY = EMGVALS(12) * VECTORS(11, 5)
            LMPZ = EMGVALS(12) * VECTORS(11, 6)
            LPTX = EMGVALS(2) * VECTORS(12, 4)
            LPTY = EMGVALS(2) * VECTORS(12, 5)
            LPTZ = EMGVALS(2) * VECTORS(12, 6)
            LMTX = EMGVALS(4) * VECTORS(13, 4)
            LMTY = EMGVALS(4) * VECTORS(13, 5)
            LMTZ = EMGVALS(4) * VECTORS(13, 6)
            LATX = EMGVALS(6) * VECTORS(14, 4)
            LATY = EMGVALS(6) * VECTORS(14, 5)
            LATZ = EMGVALS(6) * VECTORS(14, 6)

'****
'*       Column 1:   Coefficients of unknown Rx
'*       Column 2:   Coefficients of unknown Ry
'*       Column 3:   Coefficients of unknown Rz
'*       Column 4:   Coefficients of unknown Lx
'*       Column 5:   Coefficients of unknown Ly
'*       Column 6:   Coefficients of unknown Lz
'*       Column 7:   Coefficients of unknown BF
'*       Column 8:   Coefficients of total known Muscle Values
'*       Row 1:   Summation of forces in the X-dir = 0
'*       Row 2:   Summation of forces in the Y-dir = 0
'*       Row 3:   Summation of forces in the Z-dir = 0
'*       Row 4:   Summation of moments about the X-axis = 0
'*       Row 5:   Summation of moments about the Y-axis = 0
'*       Row 6:   Summation of moments about the Z-axis = 0
'****    Row 7:   Seventh Equation Assumption,  Rx = Lx

'****
'*
'*                   Fill in known array values.
'****                Start with: Summation of forces in X-dir = 0

          MATRIX(1, 1) = 1
          MATRIX(1, 2) = 0
          MATRIX(1, 3) = 0
          MATRIX(1, 4) = 1
```

```
        MATRIX(1, 5) = 0
        MATRIX(1, 6) = 0

'****
'*              Summation of forces in the Y-dir = 0
'****

        MATRIX(2, 1) = 0
        MATRIX(2, 2) = 1
        MATRIX(2, 3) = 0
        MATRIX(2, 4) = 0
        MATRIX(2, 5) = 1
        MATRIX(2, 6) = 0

'****
'*              Summation of forces in the Z-dir = 0
'****

        MATRIX(3, 1) = 0
        MATRIX(3, 2) = 0
        MATRIX(3, 3) = 1
        MATRIX(3, 4) = 0
        MATRIX(3, 5) = 0
        MATRIX(3, 6) = 1

'****
'*              Assumption that Rx = Lx
'****

        MATRIX(7, 1) = 1
        MATRIX(7, 2) = 0
        MATRIX(7, 3) = 0
        MATRIX(7, 4) = -1
        MATRIX(7, 5) = 0
        MATRIX(7, 6) = 0
        MATRIX(7, 7) = 0
        MATRIX(7, 8) = 0

'****
'*              Workout remaining summation of forces matrix coeffs.
'*              Starting with the remaining coefficient for
'*              summation of the forces in the X direction.
'****

        MATRIX(1, 7) = BYTEPOINT(4)
        MATRIX(1, 8) = -(RSMX + RDMX + RLPX + RMPX + RPTX + RMTX + RATX + LSMX + LDMX + LLPX + LMPX + LPTX +
        LMTX + LATX)

'****
'*              Remaining coefficients for the summation of the
'*              forces in the Y direction.
'****

        MATRIX(2, 7) = BYTEPOINT(5)
        MATRIX(2, 8) = -(RSMY + RDMY + RLPY + RMPY + RPTY + RMTY + RATY + LSMY + LDMY + LLPY + LMPY + LPTY +
        LMTY + LATY)

'****
'*              Remaining coefficients for the summation of the
'*              forces in the Z direction.
'****

        MATRIX(3, 7) = BYTEPOINT(6)
        MATRIX(3, 8) = -(RSMZ + RDMZ + RLPZ + RMPZ + RPTZ + RMTZ + RATZ + LSMZ + LDMZ + LLPZ + LMPZ + LPTZ +
LMTZ + LATZ)

'****
'*              Remaining coefficients for the summation of the
'*              moments about the X axis.
'****

        MATRIX(4, 1) = 0
        MATRIX(4, 2) = 0
        MATRIX(4, 3) = 0
        MATRIX(4, 4) = 0
        MATRIX(4, 5) = -Z(115)
        MATRIX(4, 6) = Y(115)
        MATRIX(4, 7) = (BYTEPOINT(6) * BYTEPOINT(2)) - (BYTEPOINT(5) * BYTEPOINT(3))
        MATRIX(4, 8) = RSMY * VECTORS(1, 3) + RDMY * VECTORS(2, 3) + RLPY * VECTORS(3, 3) + RMPY * VECTORS(4, 3) + RPTY
        * VECTORS(5, 3) + RMTY * VECTORS(6, 3) + RATY * VECTORS(7, 3)
        MATRIX(4, 8) = MATRIX(4, 8) + LSMY * VECTORS(8, 3) + LDMY * VECTORS(9, 3) + LLPY * VECTORS(10, 3) + LMPY *
        VECTORS(11, 3) + LPTY * VECTORS(12, 3) + LMTY * VECTORS(13, 3) + LATY * VECTORS(14, 3)
        MATRIX(4, 8) = MATRIX(4, 8) - (RSMZ * VECTORS(1, 2) + RDMZ * VECTORS(2, 2) + RLPZ * VECTORS(3, 2) + RMPZ * VECTORS(4,
```

```
2) + RPTZ * VECTORS(5, 2) + RMTZ * VECTORS(6, 2) + RATZ * VECTORS(7, 2))
MATRIX(4, 8) = MATRIX(4, 8) - (LSMZ * VECTORS(8, 2) + LDMZ * VECTORS(9, 2) + LLPZ * VECTORS(10, 2) + LMPZ *
VECTORS(11, 2) + LPTZ * VECTORS(12, 2) + LMTZ * VECTORS(13, 2) + LATZ * VECTORS(14, 2))

'****
'*
'*        Remaining coefficients for the summation of the
'*        moments about the Y axis.
'****

MATRIX(5, 1) = 0
MATRIX(5, 2) = 0
MATRIX(5, 3) = 0
MATRIX(5, 4) = Z(115)
MATRIX(5, 5) = 0
MATRIX(5, 6) = -X(115)
MATRIX(5, 7) = (BYTEPOINT(4) * BYTEPOINT(3)) - (BYTEPOINT(6) * BYTEPOINT(1))
MATRIX(5, 8) = RSMZ * VECTORS(1, 1) + RDMZ * VECTORS(2, 1) + RLPZ * VECTORS(3, 1) + RMPZ * VECTORS(4, 1) + RPTZ
* VECTORS(5, 1) + RMTZ * VECTORS(6, 1) + RATZ * VECTORS(7, 1)
MATRIX(5, 8) = MATRIX(5, 8) + LSMZ * VECTORS(8, 1) + LDMZ * VECTORS(9, 1) + LLPZ * VECTORS(10, 1) + LMPZ *
VECTORS(11, 1) + LPTZ * VECTORS(12, 1) + LMTZ * VECTORS(13, 1) + LATZ * VECTORS(14, 1)
MATRIX(5, 8) = MATRIX(5, 8) - (RSMX * VECTORS(1, 3) + RDMX * VECTORS(2, 3) + RLPX * VECTORS(3, 3) + RMPX *
VECTORS(4, 3) + RPTX * VECTORS(5, 3) + RMTX * VECTORS(6, 3) + RATX * VECTORS(7, 3))
MATRIX(5, 8) = MATRIX(5, 8) - (LSMX * VECTORS(8, 3) + LDMX * VECTORS(9, 3) + LLPX * VECTORS(10, 3) + LMPX *
VECTORS(11, 3) + LPTX * VECTORS(12, 3) + LMTX * VECTORS(13, 3) + LATX * VECTORS(14, 3))

'****
'*
'*        Remaining coefficients for the summation of the
'*        moments about the Z axis.
'****

MATRIX(6, 1) = 0
MATRIX(6, 2) = 0
MATRIX(6, 3) = 0
MATRIX(6, 4) = -Y(115)
MATRIX(6, 5) = X(115)
MATRIX(6, 6) = 0
MATRIX(6, 7) = (BYTEPOINT(5) * BYTEPOINT(1)) - (BYTEPOINT(4) * BYTEPOINT(2))
MATRIX(6, 8) = RSMX * VECTORS(1, 2) + RDMX * VECTORS(2, 2) + RLPX * VECTORS(3, 2) + RMPX * VECTORS(4, 2) + RPTX
* VECTORS(5, 2) + RMTX * VECTORS(6, 2) + RATX * VECTORS(7, 2)
MATRIX(6, 8) = MATRIX(6, 8) + LSMX * VECTORS(8, 2) + LDMX * VECTORS(9, 2) + LLPX * VECTORS(10, 2) + LMPX *
VECTORS(11, 2) + LPTX * VECTORS(12, 2) + LMTX * VECTORS(13, 2) + LATX * VECTORS(14, 2)
MATRIX(6, 8) = MATRIX(6, 8) - (RSMY * VECTORS(1, 1) + RDMY * VECTORS(2, 1) + RLPY * VECTORS(3, 1) + RMPY *
VECTORS(4, 1) + RPTY * VECTORS(5, 1) + RMTY * VECTORS(6, 1) + RATY * VECTORS(7, 1))
MATRIX(6, 8) = MATRIX(6, 8) - (LSMY * VECTORS(8, 1) + LDMY * VECTORS(9, 1) + LLPY * VECTORS(10, 1) + LMPY *
VECTORS(11, 1) + LPTY * VECTORS(12, 1) + LMTY * VECTORS(13, 1) + LATY * VECTORS(14, 1))

'****
'*
'*        Call the matrix solver to return unknown values
'****

MTRIX

'****
'*
'*        Call the screen output routine for the results
'****

OWTPUT(1) = PICKED
OWTPUT(2) = VAL(CLASS$)
SCREENDISPLAY FILE$, CLASS$, PICKED

'****
'*
'*        Save the results to the output file
'****

    ResultantOutputFile 2, NUM, ANB, OWTPUT(), "", ""
  NEXT NUM
  ResultantOutputFile 3, NUM, ANB, OWTPUT(), "", ""
END SUB

SUB ResultantOutputFile (CodeFlag, NUM, ANB, OWTPUT(), BATCHNUMBER$, DESCRIPTION$) STATIC

'**********************************************************************
'*
'*    Resultant Output File is a routine that handle the setup, sorting
'*    and saving of the resultant loading information calculated in the
'*    RESULTANT subroutine.
'*
'*    Resultant Output File process the CodeFlag variable passed to it
'*    to decide on the current function to perform.  The other passed
'*    variables are used as:
'*
'*        CodeFlag        =    Function Select Flag
'*        NUM             =    Numeric value of current file being processed
```

```
'       ANB             =   The Value of the current files ANB angle
'       OWTPUT(1)       =   Chosen Bite Location
'       OWTPUT(2)       =   The current files Angles Class
'       OWTPUT(3)       =   Right Condyle's Resultant Load Magnitude
'       OWTPUT(4)       =   Left Condyle's Resultant Load Magnitude
'       OWTPUT(5)       =   Right Condyle's Resultant Parasagital Incline
'       OWTPUT(6)       =   Left Condyle's Resultant Parasagital Incline
'       OWTPUT(7)       =   Right Condyle's Resultant Lateral Incline
'       OWTPUT(8)       =   Left Condyle's Resultant Lateral Incline
'       BATCHNUMBER$    =   The user defined batchnumber for the output file
'       DESCRIPTION$    =   The user defined batch description
'
'*********************************************************************

        SELECT CASE CodeFlag

'****
'       If the CodeFlag = 0 then make no output file,
'       do nothing and return.
'****

        CASE IS <= 0
                InternalFlag = 1
                EXIT SUB

'****
'       If the CodeFlag = 1 then open the output file,
'       write the file header information and intialize
'       the information storage arrays.
'****

        CASE 1
                DIM ClassI(24, 10), ClassIIDivI(15, 10), ClassIIDivII(4, 10), ClassIII(14, 10)
                ERASE ClassI, ClassIIDivI, ClassIIDivII, ClassIII
                COUNTER1 = 0
                COUNTER21 = 0
                COUNTER22 = 0
                COUNTER3 = 0
                OPEN "C:\QB\TMJ\DATA\RESULTS\" + BATCHNUMBER$ + ".RES" FOR OUTPUT AS #2
                LOCATE 24, 2
                PRINT SPACE$(76);
                CENTER "Opening file C:\QB\TMJ\DATA\RESULTS\" + BATCHNUMBER$ + ".RES for raw data storage", 20

'****
'               Write file headers to the storage files
'****

                WRITE #2, "This is output file " + BATCHNUMBER$ + ".RES"
                WRITE #2, "This file was generated on " + DATE$
                WRITE #2, "This file was generated at " + TIME$
                WRITE #2, "Occlusal load occuring at the " + BITEPOINT$(OWTPUT(1), 1)
                WRITE #2, DESCRIPTION$
                WRITE #2, " "
                WRITE #2, "EMG values used in this run are:"
                WRITE #2, "-----------------------"
                WRITE #2, "RPT", "LPT", "RMT", "LMT", "RAT", "LAT", "RDM", "LDM", "RSM", "LSM", "RMP", "LMP", "RLP", "LLP"
                WRITE #2, EMGVALS(1), EMGVALS(2), EMGVALS(3), EMGVALS(4), EMGVALS(5), EMGVALS(6), EMGVALS(7),
                EMGVALS(8), EMGVALS(9), EMGVALS(10), EMGVALS(11), EMGVALS(12), EMGVALS(13), EMGVALS(14)
                WRITE #2, " "
                WRITE #2, " File ", " ANB  ", " Class ", " Ipsi ", " Contra ", " Bite ", " Ipsi ", " Contra ", " Ipsi ", " Contra "
                WRITE #2, "  #   ", " Angle ", "  #   ", " Mag ", "  Mag ", " Mag  ", " Para ", " Para ", " Lat ", " Lat  "
                WRITE #2, "------------------------------------------------------"

'****
'       If the CodeFlag = 2 then sort the new incoming data
'       and store it for a later save. The storage arrays are
'       defined as:
'
'           Class?(?,1) =   The filenumber
'           Class?(?,2) =   The Angle's ANB angle
'           Class?(?,3) =   OWTPUT(2) for the active file [class]
'           Class?(?,4) =   OWTPUT(3) for the active file [Rmag]
'           Class?(?,5) =   OWTPUT(4) for the active file [Lmag]
'           Class?(?,6) =   OWTPUT(5) for the active file [Rpara]
'           Class?(?,7) =   OWTPUT(6) for the active file [Lpara]
'           Class?(?,8) =   OWTPUT(7) for the active file [Rlat]
'           Class?(?,9) =   OWTPUT(8) for the active file [Llat]
'           Class?(?,10)=   FORCES(7) [Bite Magnitude]
'
'****

        CASE 2
                IF InternalFlag = 1 THEN EXIT SUB
```

```
                SELECT CASE OWTPUT(2)
                    CASE 1
                        COUNTER1 = COUNTER1 + 1
                        ClassI(COUNTER1, 1) = NUM
                        ClassI(COUNTER1, 2) = ANB
                        ClassI(COUNTER1, 3) = OWTPUT(2)
                        ClassI(COUNTER1, 4) = OWTPUT(3)
                        ClassI(COUNTER1, 5) = OWTPUT(4)
                        ClassI(COUNTER1, 6) = OWTPUT(5)
                        ClassI(COUNTER1, 7) = OWTPUT(6)
                        ClassI(COUNTER1, 8) = OWTPUT(7)
                        ClassI(COUNTER1, 9) = OWTPUT(8)
                        ClassI(COUNTER1, 10) = FORCES(7)

                    CASE 2
                        SELECT CASE NUM
                            CASE 4, 16, 14
                                COUNTER22 = COUNTER22 + 1
                                ClassIIDivII(COUNTER22, 1) = NUM
                                ClassIIDivII(COUNTER22, 2) = ANB
                                ClassIIDivII(COUNTER22, 3) = OWTPUT(2)
                                ClassIIDivII(COUNTER22, 4) = OWTPUT(3)
                                ClassIIDivII(COUNTER22, 5) = OWTPUT(4)
                                ClassIIDivII(COUNTER22, 6) = OWTPUT(5)
                                ClassIIDivII(COUNTER22, 7) = OWTPUT(6)
                                ClassIIDivII(COUNTER22, 8) = OWTPUT(7)
                                ClassIIDivII(COUNTER22, 9) = OWTPUT(8)
                                ClassIIDivII(COUNTER22, 10) = FORCES(7)
                            CASE ELSE
                                COUNTER21 = COUNTER21 + 1
                                ClassIIDivI(COUNTER21, 1) = NUM
                                ClassIIDivI(COUNTER21, 2) = ANB
                                ClassIIDivI(COUNTER21, 3) = OWTPUT(2)
                                ClassIIDivI(COUNTER21, 4) = OWTPUT(3)
                                ClassIIDivI(COUNTER21, 5) = OWTPUT(4)
                                ClassIIDivI(COUNTER21, 6) = OWTPUT(5)
                                ClassIIDivI(COUNTER21, 7) = OWTPUT(6)
                                ClassIIDivI(COUNTER21, 8) = OWTPUT(7)
                                ClassIIDivI(COUNTER21, 9) = OWTPUT(8)
                                ClassIIDivI(COUNTER21, 10) = FORCES(7)
                        END SELECT
                    CASE 3
                        COUNTER3 = COUNTER3 + 1
                        ClassIII(COUNTER3, 1) = NUM
                        ClassIII(COUNTER3, 2) = ANB
                        ClassIII(COUNTER3, 3) = OWTPUT(2)
                        ClassIII(COUNTER3, 4) = OWTPUT(3)
                        ClassIII(COUNTER3, 5) = OWTPUT(4)
                        ClassIII(COUNTER3, 6) = OWTPUT(5)
                        ClassIII(COUNTER3, 7) = OWTPUT(6)
                        ClassIII(COUNTER3, 8) = OWTPUT(7)
                        ClassIII(COUNTER3, 9) = OWTPUT(8)
                        ClassIII(COUNTER3, 10) = FORCES(7)
                END SELECT

'****
'*          If the CodeFlag = 3 then write all the stored results
'*          to disk.
'****

        CASE 3
                IF InternalFlag = 1 THEN
                        InternalFlag = 0
                        EXIT SUB
                END IF
                SELECT CASE OWTPUT(1)

'****
'*          If bite is on left side (Chosen Bite Location
'*          between 1 and 5), make left the ipsilateral side
'****

                    CASE 1 TO 5
                        I = 1
                        WHILE I <= COUNTER1
                                WRITE #2, ClassI(I, 1), ClassI(I, 2), ClassI(I, 3), ClassI(I, 5), ClassI(I, 4), ClassI(I, 10), ClassI(I, 7), ClassI(I,
                                6), ClassI(I, 9), ClassI(I, 8)
                                I = I + 1
                        WEND
                        I = 1
                        WHILE I <= COUNTER21
                                WRITE #2, ClassIIDivI(I, 1), ClassIIDivI(I, 2), ClassIIDivI(I, 3), ClassIIDivI(I, 5), ClassIIDivI(I, 4), ClassIIDivI(I,
                                10), ClassIIDivI(I, 7), ClassIIDivI(I, 6), ClassIIDivI(I, 9), ClassIIDivI(I, 8)
```

```
                        I = I + 1
                WEND
                I = 1
                WHILE I <= COUNTER22
                        WRITE #2, ClassIIDivII(I, 1), ClassIIDivII(I, 2), ClassIIDivII(I, 3), ClassIIDivII(I, 5), ClassIIDivII(I, 4),
                        ClassIIDivII(I, 10), ClassIIDivII(I, 7), ClassIIDivII(I, 6), ClassIIDivII(I, 9), ClassIIDivII(I, 8)
                        I = I + 1
                WEND
                I = 1
                WHILE I <= COUNTER3
                        WRITE #2, ClassIII(I, 1), ClassIII(I, 2), ClassIII(I, 3), ClassIII(I, 5), ClassIII(I, 4), ClassIII(I, 10), ClassIII(I,
                        7), ClassIII(I, 6), ClassIII(I, 9), ClassIII(I, 8)
                        I = I + 1
                WEND
                CLOSE #2

'****
'*          If bite is on right side (Chosen Bite Location
'*          between 6 and 10), make right the ipsilateral side
'****

                CASE 6 TO 10
                  I = 1
                  WHILE I <= COUNTER1
                        WRITE #2, ClassI(I, 1), ClassI(I, 2), ClassI(I, 3), ClassI(I, 4), ClassI(I, 5), ClassI(I, 10), ClassI(I, 6), ClassI(I,
                        7), ClassI(I, 8), ClassI(I, 9)
                        I = I + 1
                  WEND
                  I = 1
                  WHILE I <= COUNTER21
                        WRITE #2, ClassIIDivI(I, 1), ClassIIDivI(I, 2), ClassIIDivI(I, 3), ClassIIDivI(I, 4), ClassIIDivI(I, 5), ClassIIDivI(I,
                        10), ClassIIDivI(I, 6), ClassIIDivI(I, 7), ClassIIDivI(I, 8), ClassIIDivI(I, 9)
                        I = I + 1
                  WEND
                  I = 1
                  WHILE I <= COUNTER22
                        WRITE #2, ClassIIDivII(I, 1), ClassIIDivII(I, 2), ClassIIDivII(I, 3), ClassIIDivII(I, 4), ClassIIDivII(I, 5),
                        ClassIIDivII(I, 10), ClassIIDivII(I, 6), ClassIIDivII(I, 7), ClassIIDivII(I, 8), ClassIIDivII(I, 9)
                        I = I + 1
                  WEND
                  I = 1
                  WHILE I <= COUNTER3
                        WRITE #2, ClassIII(I, 1), ClassIII(I, 2), ClassIII(I, 3), ClassIII(I, 4), ClassIII(I, 5), ClassIII(I, 10), ClassIII(I,
                        6), ClassIII(I, 7), ClassIII(I, 8), ClassIII(I, 9)
                        I = I + 1
                  WEND
                  CLOSE #2
            END SELECT
      END SELECT
END SUB

SUB ROTATE (HORZ, VERT, A(), B())

'************************************************************************
'*
'*      ROTATE is a routine which reads in a horizontal (adjacent)
'*      and vertical (opposite) dimensions to calculate the angle
'*      change of the dataset. The A() array should be the
'*      dimensional array corresponding to the horizontal direction,
'*      and the B() array to the vertical direction.
'*
'************************************************************************

      CONST PI# = 3.141592653589793#

'****
'*              Calculate the angle to rotate in the plane,
'*              DELTATHETA
'****

      IF HORZ = 0 THEN HORZ = .000001
      IF VERT = 0 THEN EXIT SUB
      DELTATHETA = ATN(VERT / HORZ)

'****
'*              Loop through the 113 locations, rotating each one
'*              DELTATHETA degrees
'****

      FOR I = 1 TO 113

'****
'*              Check for a horizontal component = 0
```

```
'*
'****            to avoid zero division

        IF A(I) = 0 THEN A(I) = .000001
        IF A(I) > 0 AND B(I) >= 0 THEN GOTO 1000
        IF A(I) > 0 AND B(I) < 0 THEN GOTO 1010
        IF A(I) < 0 AND B(I) >= 0 THEN GOTO 1020
        IF A(I) < 0 AND B(I) < 0 THEN GOTO 1020
1000    THETA = ATN(B(I) / A(I)): GOTO 1030
1010    THETA = (2 * PI) + ATN(B(I) / A(I)): GOTO 1030
1020    THETA = PI + ATN(B(I) / A(I))

'****
'*               Calculate the new angle for point I
'****

1030    THETANEW = THETA - DELTATHETA
        R = SQR(A(I) ^ 2 + B(I) ^ 2)
        A(I) = R * COS(THETANEW)
        B(I) = R * SIN(THETANEW)
        IF A(I) < .000001 AND A(I) > -.000001 THEN A(I) = 0
        IF B(I) < .000001 AND B(I) > -.000001 THEN B(I) = 0
       NEXT I
END SUB

SUB SAVENEW (FILENUM$, X(), Y(), Z(), COMMENTS$())

'**********************************************************************
'*
'*      This routine calculates the muscle unit vectors and
'*      their origins and then saves the new data set followed by
'*      the vectors.
'*
'**********************************************************************

      FILE$ = "TMJ" + FILENUM$ + ".MOD"
      FILENAME$ = "C:\QB\TMJ\DATA\" + FILE$
      OPEN FILENAME$ FOR OUTPUT AS #1

'****
'*               Write the new dataset
'****

      FOR NUM = 1 TO 113
         WRITE #1, NUM, X(NUM), Y(NUM), Z(NUM), COMMENTS$(NUM)
      NEXT NUM

'****
'*               Calculate the unit vectors
'****

      UNITVECTOR "RSM", X(93), Y(93), Z(93), X(9), Y(9), Z(9)
      UNITVECTOR "RDM", X(94), Y(94), Z(94), X(10), Y(10), Z(10)
      UNITVECTOR "RLP", X(91), Y(91), Z(91), X(11), Y(11), Z(11)
      UNITVECTOR "RMP", X(112), Y(112), Z(112), X(54), Y(54), Z(54)
      UNITVECTOR "RPT", X(88), Y(88), Z(88), X(8), Y(8), Z(8)
      UNITVECTOR "RMT", X(88), Y(88), Z(88), X(7), Y(7), Z(7)
      UNITVECTOR "RAT", X(88), Y(88), Z(88), X(6), Y(6), Z(6)
      UNITVECTOR "LSM", X(110), Y(110), Z(110), X(51), Y(51), Z(51)
      UNITVECTOR "LDM", X(111), Y(111), Z(111), X(52), Y(52), Z(52)
      UNITVECTOR "LLP", X(108), Y(108), Z(108), X(53), Y(53), Z(53)
      UNITVECTOR "LMP", X(95), Y(95), Z(95), X(12), Y(12), Z(12)
      UNITVECTOR "LPT", X(105), Y(105), Z(105), X(50), Y(50), Z(50)
      UNITVECTOR "LMT", X(105), Y(105), Z(105), X(49), Y(49), Z(49)
      UNITVECTOR "LAT", X(105), Y(105), Z(105), X(48), Y(48), Z(48)
      CLOSE #1
END SUB

SUB SCREENDISPLAY (FILE$, CLASS$, PICKED)

'**********************************************************************
'*
'*      SCREENDISPLAY is a routine which, suprise suprise, presents the
'*      "Final Answers" to the user, at last.
'*
'**********************************************************************

'****
'*               List the current attributes.
'****

      COLOR 2
      HEADER "Results for the skull of dataset " + FILE$
```

```
LOCATE 7, 3: PRINT "Selected Datafile: ";
COLOR 1: PRINT FILE$
COLOR 2: LOCATE 7, 52: PRINT "Skull Class: ";
COLOR 1: PRINT "#"; CLASS$
COLOR 2: LOCATE 9, 3: PRINT "Selected Bite Location: ";
COLOR 1: PRINT BITEPOINT$(PICKED, 1)

'****
'*              Display the selected EMG values.
'****

COLOR 2: LOCATE 11, 3: PRINT "Selected EMG values:"
COLOR 8: LOCATE 12, 5
PRINT " RPT LPT RMT LMT RAT LAT RDM LDM RSM LSM RMP LMP RLP LLP"
COLOR 1
FOR I = 1 TO 14
   LOCATE 13, (5 + (I - 1) * 5)
   PRINT USING "###.#"; EMGVALS(I);
NEXT I

'****
'*              Box in the EMG values.
'****

COLOR 2: LINE (36, 154)-(600, 182), , B: LINE (36, 168)-(600, 168)
FOR I = 1 TO 13
   LINE ((79 + (I - 1) * 40), 154)-((79 + (I - 1) * 40), 182)
NEXT I

'****
'*              Display the first line of the resultant loads.
'****

LOCATE 15, 3: PRINT "Resultant Loads:"
LOCATE 16, 5: PRINT "Rx = ";
COLOR 1: PRINT USING "+###.#"; FORCES(1);
PRINT "%";
COLOR 8: LOCATE , 18: PRINT CHR$(187);
LOCATE , 50: PRINT CHR$(201);
COLOR 2: LOCATE , 52: PRINT "Magnitude = ";

'****
'*              Calculate and display the magnitude of the
'*              right condylar load.
'****

OWTPUT(3) = SQR(FORCES(1) ^ 2 + FORCES(2) ^ 2 + FORCES(3) ^ 2)
COLOR 1: PRINT USING "###.#"; OWTPUT(3);
PRINT "%"

'****
'*              Display the second line.
'****

COLOR 2: LOCATE 17, 5: PRINT "Ry = ";
COLOR 1: PRINT USING "+###.#"; FORCES(2);
PRINT "%";
COLOR 8: LOCATE , 18: PRINT CHR$(204) + CHR$(205) + CHR$(205);
PRINT "    Right Condylar Load";
LOCATE , 48: PRINT CHR$(205) + CHR$(205) + CHR$(185);
COLOR 2: LOCATE , 52: PRINT "Parasagital Incline = ";

'****
'*              Calculate and display the right condylar loads
'*              angle parallel to the parasagital plane.
'****

OWTPUT(5) = (180 / 3.1415926#) * ATN(ABS(FORCES(2)) / ABS(FORCES(3)))
COLOR 1: PRINT USING "##.#"; OWTPUT(5);
PRINT CHR$(248)

'****
'*              Display the third line.
'****

COLOR 2: LOCATE 18, 5: PRINT "Rz = ";
COLOR 1: PRINT USING "+###.#"; FORCES(3);
PRINT "%";
COLOR 8: LOCATE , 18: PRINT CHR$(188);
COLOR 13
IF PICKED < 6 THEN                                      'If bite is on the left
   LOCATE , 25: PRINT "Contralateral Side";
ELSE                                                    'If bite is on the right
```

```
   LOCATE , 25: PRINT "Ipsilateral Side";
END IF
COLOR 8: LOCATE , 50: PRINT CHR$(200);
COLOR 2: LOCATE , 52: PRINT "Lateral Incline = ";

'****
'*          Calculate and display the right condylar loads
'*          lateral incline perpendicular to the parasagital
'*          plane.
'****

OWTPUT(7) = ABS((180 [illegible] 415 [illegible] * ATN(ABS(FORCES(1)) / ABS(FORCES(3))))
IF FORCES(1) < 0 THEN [illegible] ) = -OWTPUT(7)                 'Sign the angle
COLOR 1: PRINT USING [illegible] OWTPUT(7);
PRINT [illegible]R$(248)

'****
'*          Display the fourth line.
'****

COLOR 2: LOCATE 19, 5: PRINT "Lx = ";
COLOR 1: PRINT USING "+###.#"; FORCES(4);
PRINT "%";
COLOR 8: LOCATE , 18: PRINT CHR$(187);
LOCATE , 50: PRINT CHR$(201);
COLOR 2: LOCATE , 52: PRINT "Magnitude = ";

'****
'*          Calculate and display the magnitude of the
'*          left condylar load.
'****

OWTPUT(4) = SQR(FORCES(4) ^ 2 + FORCES(5) ^ 2 + FORCES(6) ^ 2)
COLOR 1: PRINT USING "###.#"; OWTPUT(4);
PRINT "%"

'****
'*          Display the fifth line
'****

COLOR 2: LOCATE 20, 5: PRINT "Ly = ";
COLOR 1: PRINT USING "+###.#"; FORCES(5);
PRINT "%";
COLOR 8: LOCATE , 18: PRINT CHR$(204) + CHR$(205) + CHR$(205);
PRINT "    Left Condylar Load";
LOCATE , 48: PRINT CHR$(205) + CHR$(205) + CHR$(185);
COLOR 2: LOCATE , 52: PRINT "Parasagital Incline = ";

'****
'*          Calculate and display the left condylar loads
'*          angle parallel to the parasagital plane.
'****

OWTPUT(6) = (180 / 3.1415926#) * ATN(ABS(FORCES(5)) / ABS(FORCES(6)))
COLOR 1: PRINT USING "##.#"; OWTPUT(6);
PRINT CHR$(248)

'****
'*          Display the sixth line.
'****

COLOR 2: LOCATE 21, 5: PRINT "Lz = ";
COLOR 1: PRINT USING "+###.#"; FORCES(6);
PRINT "%";
COLOR 8: LOCATE , 18: PRINT CHR$(188);
COLOR 13
IF PICKED > 6 THEN                                   'If bite is on the right
   LOCATE , 25: PRINT "Contralateral Side";
ELSE                                                 'If bite is on the left
   LOCATE , 25: PRINT "Ipsilateral Side";
END IF
COLOR 8: LOCATE , 50: PRINT CHR$(200);
COLOR 2: LOCATE , 52: PRINT "Lateral Incline = ";

'****
'*          Calculate and display the left condylar loads
'*          lateral incline perpendicular to the parasagital
'*          plane.
'****

OWTPUT(8) = ABS((180 / 3.1415926#) * ATN(ABS(FORCES(4)) / ABS(FORCES(6))))
IF FORCES(4) < 0 THEN OWTPUT(8) = -OWTPUT(8)                 'Sign the angle
COLOR 1: PRINT USING "+##.#"; OWTPUT(8);
```

```
        PRINT CHR$(248)

'****
'*                  Display the resultant bite force.
'****

        COLOR 2: LOCATE 23, 3: PRINT "Resultant Biteforce =";
        COLOR 1: PRINT USING "+###.#"; FORCES(7);
        PRINT "%";

'****
'*                  Display the bite force unit vector.
'****

        COLOR 2
        LOCATE , 38: PRINT "Unit vector = ";
        COLOR 1
        PRINT USING "+#.###"; (INT(BYTEPOINT(4) * 1000)) / 1000;
        PRINT "i + ";
        PRINT USING "+#.###"; (INT(BYTEPOINT(5) * 1000)) / 1000;
        PRINT "j + ";
        PRINT USING "+#.###"; (INT(BYTEPOINT(6) * 1000)) / 1000;
        PRINT "k"

'****
'*                  Display this screen for approximately 10 seconds
'*                  and then proceed in the loop.  Pause the loop upon a
'*                  key press, then resume after another key press.
'****

        COLOR 2
        FOR DELAY = 1 TO 1000
          UPDATE
          A$ = INKEY$
          IF A$ <> "" THEN
                  COLOR 11
                  LOCATE 24, 37
                  PRINT "PAUSED";
                  COLOR 2
3750              B$ = INKEY$
                  IF B$ = "" THEN
                    UPDATE
                    GOTO 3750
                  END IF
                  LOCATE 24, 37
                  PRINT "      ";
          END IF
        NEXT DELAY
END SUB

SUB TURN (ANGLE, A, B, LOCATION())

'*************************************************************************
'*
'*      TURN is a routine very similar to rotate.  Its purpose is to rotate
'*      the LOCATION() dataset through the specified ANGLE in the A-B plane
'*      where A is the horizontal component and B is the vertical component.
'*        For the LOCATION() dataset a value of 1, 2 or 3 for A or B
'*      corresponds to a direction of X, Y or Z respectively.
'*
'*************************************************************************

        CONST PI# = 3.14159265358978#

'****
'*                  Loop through the 5 locations, rotating each one
'*                  ANGLE degrees.
'****

        FOR I = 1 TO 5
          IF LOCATION(I, A) = 0 THEN LOCATION(I, A) = .000001
          IF LOCATION(I, A) > 0 AND LOCATION(I, B) >= 0 THEN GOTO 9500
          IF LOCATION(I, A) > 0 AND LOCATION(I, B) < 0 THEN GOTO 9510
          IF LOCATION(I, A) < 0 AND LOCATION(I, B) >= 0 THEN GOTO 9520
          IF LOCATION(I, A) < 0 AND LOCATION(I, B) < 0 THEN GOTO 9520
9500      THETA = ATN(LOCATION(I, B) / LOCATION(I, A)): GOTO 9530
9510      THETA = (2 * PI) + ATN(LOCATION(I, B) / LOCATION(I, A)): GOTO 9530
9520      THETA = PI + ATN(LOCATION(I, B) / LOCATION(I, A))

'****
'*                  Calculate the new angle THETANEW for LOCATION(I,?)
'****
```

```
9530      THETANEW = THETA - ANGLE
          R = SQR(LOCATION(I, A) ^ 2 + LOCATION(I, B) ^ 2)
          LOCATION(I, A) = R * COS(THETANEW)
          LOCATION(I, B) = R * SIN(THETANEW)
          IF LOCATION(I, A) < .000001 AND LOCATION(I, A) > -.000001 THEN LOCATION(I, A) = 0
          IF LOCATION(I, B) < .000001 AND LOCATION(I, B) > -.000001 THEN LOCATION(I, B) = 0
        NEXT I
END SUB

SUB UNITVECTOR (NAMES$, XI, YI, ZI, XO, YO, ZO)

'************************************************************************
'*
'*      This routine takes two muscle coordinates, the insert (located on
'*      the mandible) and the origin (located on the skull) and saves
'*      their unit vector and origin to the already opened file #1.
'*
'************************************************************************

        XSTEP = (XO - XI)
        YSTEP = (YO - YI)
        ZSTEP = (ZO - ZI)
        DENOMINATOR = SQR(XSTEP ^ 2 + YSTEP ^ 2 + ZSTEP ^ 2)
        I = XSTEP / DENOMINATOR
        J = YSTEP / DENOMINATOR
        K = ZSTEP / DENOMINATOR
        WRITE #1, NAMES$, XI, YI, ZI, I, J, K
END SUB

SUB UPDATE STATIC

'************************************************************************
'*
'*      Update is a routine to update the header clock every second while
'*      in an input wait loop.
'*
'************************************************************************

        IF TIME$ <> OLDTIME$ THEN
           LOCATE 2, 68
           COLOR 2
           PRINT TIME$
           OLDTIME$ = TIME$
        ELSE
           EXIT SUB
        END IF
END SUB
```