

COORDINATE SYSTEMS FOR A NAVAL VIRTUAL ENVIRONMENT

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ASLI KILIÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
COMPUTER ENGINEERING

AUGUST 2005

Approval of the Graduate School of Natural and Applied Sciences.

---

Prof. Dr. Canan ÖZGEN  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Prof. Dr. Ayşe KİPER  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Asst. Prof. Dr. Halit OĞUZTÜZÜN  
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Ahmet COŞAR	(METU,CEng)	_____
Asst. Prof. Dr. Halit OĞUZTÜZÜN	(METU, CEng)	_____
Assoc. Prof. Dr. Veysi İŞLER	(METU,CEng)	_____
Asst. Prof. Dr. Kayhan İMRE	(HACETTEPE,CEng)	_____
Dr. Cevat ŞENER	(METU,CEng)	_____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name : Aslı Kılıç

Signature :

# ABSTRACT

## COORDINATE SYSTEMS FOR A NAVAL VIRTUAL ENVIRONMENT

Kılıç, Aslı

M.S., Department of Computer Engineering

Supervisor : Asst. Prof. Dr. Halit Oğuztüzün

August 2005, 132 pages

The purpose of this thesis is implementing World Geodetic System (WGS) for Naval Surface Tactical Maneuvering Simulation System (NSTMSS), a High Level Architecture (HLA) based naval simulation, and also implementing body coordinate system for the ships of NSTMSS and its combination with WGS so that NSTMSS can be more accurate, and new ship dynamics models can be integrated to the NSTMSS environment more easily.

To improve the coordinate system of NSTMSS these methods were used; World Geodetic System - 1984 (WGS 84) was chosen as the reference system of new coordinate system of NSTMSS and coordinates were transformed to Universal Transverse Mercator (UTM) Coordinate System. Also, the terrains which NSTMSS uses created in OpenFlight (FLT) format with UTM map projection method. In addition to this, Ship Body Coordinate System was implemented as a Cartesian coordinate system.

This thesis has achieved improved coordinate systems for the NSTMSS environment to increase the realism of the simulation.

**Keywords:** WGS, NSTMSS, UTM, Ship Body Coordinate System

# ÖZ

## BİR DENİZ SANAL ORTAMI İÇİN KOORDİNAT SİSTEMLERİ

Kılıç, Aslı

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez yöneticisi : Yar. Doç. Dr. Halit Oğuztüzün

Ağustos 2005, 132 sayfa

Bu tezin amacı, bir Yüksek Seviye Mimari (HLA) tabanlı deniz simülasyonu olan Suüstü Gemileri Taktik Manevra Simülasyon Sistemi (NSTMSS) ni, Dünya Jeodezik Sistemi (WGS) ne uyumlu hale getirmek ve gemi lokal koordinat sistemini gerçekleştirmek ve bunu da WGS ile bütünleştirmektir. Böylece NSTMSS'in daha yanlışsız olması sağlanacak ve yeni gemi dinamik modelleri NSTMSS ortamına daha kolay entegre edilebilecektir.

NSTMSS kordinat sistemini geliştirmek için şu metodlar kullanılmıştır; NSTMSS yeni kordinat sisteminin referans sistemi olarak Dünya Jeodezik Sistemi - 1984 (WGS 84) seçilmiştir ve koordinatlar Universal Transverse Mercator (UTM) koordinat sistemine çevrilmiştir. Ayrıca NSTMSS'in kullandığı araziler, OpenFlight (FLT) formatında UTM harita izdüşüm yöntemi kullanılarak üretilmiştir. Buna ilaveten Gemi Gövde Koordinat Sistemi, Kartezyen Koordinate Sistemi kullanılarak geliştirilmiştir.

Bu tez, simulasyonun gerçekliğini arttırmak amacıyla NSTMSS ortamı için geliştirilmiş koordinat sistemleri elde etmiştir.

**Anahtar Kelimeler:** WGS, NSTMSS, UTM, Gemi Gövde Koordinat Sistemi

To My Family

# **ACKNOWLEDGMENTS**

I am grateful to my advisor Asst. Prof Dr. Halit Oğuztüzün for his unique support and guidance through the research.

Thanks to Captain Ahmet Tanık from General Staff for his guidance about map projections method.

# TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ .....	v
ACKNOWLEDGMENTS.....	vii
TABLE OF CONTENTS.....	viii
LIST OF FIGURES.....	x
LIST OF TABLES .....	xi
LIST OF ABBREVIATIONS.....	xii
CHAPTER	
1.INTRODUCTION.....	1
1.1    PROBLEM DEFINITION .....	2
1.2    BACKGROUND.....	2
1.2.1    COORDINATE SYSTEMS.....	2
1.2.2    MAP PROJECTION .....	4
1.2.3    THE OLD NSTMSS COORDINATE SYSTEM .....	5
1.2.4    THE NEW NSTMSS COORDINATE SYSTEM.....	6
1.3    TOOLS .....	7
1.3.1    DEVELOPMENT TOOLS .....	7
1.3.2    TESTING TOOLS .....	8
1.4    THESIS OUTLINE.....	9
2.COORDINATE SYSTEMS.....	10
2.1    WORLD COORDINATE SYSTEM .....	10
2.1.1    THE SHAPE OF THE EARTH .....	11
2.1.2    TYPES OF COORDINATES .....	16
2.1.3    DATUM .....	23
2.1.4    WORLD GEODETIC SYSTEM 1984 (WGS84).....	28
2.2    LOCAL COORDINATE SYSTEM.....	35
2.3    SHIP BODY COORDINATE SYSTEM.....	36
2.3.1    REPRESENTING ORIENTATIONS.....	36
2.4    RELATED WORK .....	42
3.MAP PROJECTION METHODS AND DIGITAL TERRAINS .....	47
3.1    MAP PROJECTION METHODS.....	47
3.1.1    TYPES OF PROJECTIONS .....	48
3.1.2    CONVERSION FORMULAS .....	55
3.2    DIGITAL TERRAINS .....	60
3.2.1    DIGITAL TERRAIN ELEVATION DATA (DTED).....	60
3.2.2    DIGITAL ELEVATION DATA (DED).....	62
3.2.3    OPENFLIGHT FORMAT (FLT).....	66
3.3    RATIONALE FOR ADOPTED TECHNOLOGIES.....	69



3.3.1	RATIONALE FOR ADOPTED COORDINATE SYSTEMS .....	69
3.3.2	RATIONALE FOR EULER ANGLES .....	70
3.4	RELATED WORK .....	71
4.	IMPLEMENTATION .....	72
4.1	IMPLEMENTATION OF WORLD COORDINATE SYSTEM .....	72
4.2	IMPLEMENTATION OF BODY COORDINATE SYSTEM.....	75
4.3	INTEGRATION OF NEW CLASSES TO NSTMSS .....	77
4.3.1	INTEGRATION OF WORLD COORDINATE SYSTEM CLASS	78
4.3.2	INTEGRATION OF BODY COORDINATE SYSTEM CLASS ...	78
4.4	CHANGES TO NSTMSS .....	79
4.4.1	CHANGES TO ENVIRONMENT FEDERATE.....	79
5.	TESTING .....	82
5.1	UNIT TESTS .....	82
5.1.1	WORLD COORDINATE SYSTEM UNIT TESTS .....	82
5.2	INTEGRATION TESTS.....	84
5.2.1	WORLD COORDINATE SYSTEMS INTEGRATION TESTS ...	84
5.2.2	BODY COORDINATE SYSTEM INTEGRATION TESTS.....	86
6.	CONCLUSIONS AND DISCUSSIONS .....	90
6.1	CONTRIBUTIONS OF THIS WORK .....	90
6.2	DISCUSSION .....	91
6.3	SUGGESTIONS FOR FUTURE WORK.....	92
	REFERENCES.....	93
	APPENDICES	
	APPENDIX- A.....	97
	WGS 84 PARAMETERS OF THE EARTH DEFINITION.....	97
	APPENDIX-B .....	98
	SHIP BODY COORDINATE SYSTEM TEST INPUT DATA.....	98
	APPENDIX-C .....	99
	THE EARTH ACCORDING TO WGS 84.....	99
	APPENDIX D .....	102
	POSITION SHIFTS FROM DATUM DIFFERENCES.....	102
	APPENDIX E.....	103
	DATUMS USED ALL OVER THE WORLD .....	103
	APPENDIX-F .....	107
	NSTMSS INSTALLATION MANUAL (Revised) .....	107

# LIST OF FIGURES

Figure 1.1 Transformation Between Geodetic and Geocentric Coordinate Systems in NSTMSS.....	6
Figure 2.1 Shape of the Earth.....	11
Figure 2.2 Gravity.....	13
Figure 2.3 Geoid and Ellipsoid .....	15
Figure 2.4 An ellipsoid with latitude and longitude and the associated 3-D Cartesian axes.....	17
Figure 2.5 Geodetic Latitude.....	18
Figure 2.6 Geodetic Longitude .....	18
Figure 2.7 Spheroidal Height.....	19
Figure 2.8 Geoid Height.....	21
Figure 2.9 Geoid Heights .....	22
Figure 2.10 Semi major and minor axes of an ellipsoid.....	24
Figure 2.11 Local Geodetic Datum.....	26
Figure 2.12 Geocentric Geodetic Datum.....	27
Figure 2.13 The WGS 84 Coordinate System Definition.....	30
Figure 2.14 Increasing of latitude from equator to poles .....	34
Figure 2.15 Decreasing of longitude from equator to poles .....	34
Figure 2.16 sinop.flt and map origin.....	36
Figure 2.17 zyx notation of Euler Rotation.....	38
Figure 2.18 The ship model of NSTMSS (Meko).....	39
Figure 2.19 Defining a point on the ship in local coordinate system .....	40
Figure 2.20 Defining a point on the ship in world coordinate system.....	41
Figure 2.21 NPSNET World Coordinate Sytem .....	43
Figure 2.22 NPSNET Body Coordinate System .....	44
Figure 3.1 Map projection.....	47
Figure 3.2 Azimuthal Projection .....	49
Figure 3.3 Conic projection .....	49
Figure 3.4 Cylindrical Projection.....	50
Figure 3.5 The zones used in the UTM spatial coordinate system.....	51
Figure 3.6 The central meridian of an UTM zone.....	52
Figure 3.7 UTM zone 13 coordinates for the southern hemisphere. ....	53
Figure 3.8 UTM zone 13 coordinates for the northern hemisphere. ....	54
Figure 3.9 Perspective scenes of Mount Saint Helens. ....	61
Figure 3.10 Conversion from DTED to DED with Multigen Creator.....	64
Figure 3.11 The resultunt DED file.....	65
Figure 3.12 Multigen needs DED (elevation data) to create FLT.....	66
Figure 3.13 The process of creating FLT from DED .....	67
Figure 4.1 Terrain update from Environment module .....	80
Figure 5.1 Unit testing with GEOTRANS .....	83
Figure 5.2 Integration testing .....	85
Figure 5.3 Integration testing command prompt.....	85
Figure 5.4 Boundary test points on the ship.....	86
Figure 5.5 Boundary test points on the ship (cont.) .....	87
Figure 5.6 Arbitrary boundary test points on the ship (front view).....	87
Figure 5.7 Arbitrary boundary test points on the ship cont. (backside) .....	88
Figure 5.8 Arbitrary boundary test points on the ship (cont.) .....	88
Figure D.1 Differences Between Datums .....	102

# LIST OF TABLES

Table 2.1 WGS84 ellipsoid parameters.....	31
Table A.1 WGS84 Parameters .....	97
Table B.1 Ship Body Coordinate System Test Input Data.....	98
Table C.1 The Earth According to WGS84 .....	99
Table E.1 Datums Used All Over The World and Areas and Ellipsoids of Datums.....	103

# LIST OF ABBREVIATIONS

- AGD:** The Australian Geodetic Datum
- BIH:** Bureau International de l'Heure
- CAD:** Computer Aided Design
- CTP:** Conventional Terrestrial Pole
- DED:** Digital Elevation Data
- DEM:** Digital Elevation Model
- DLLC:** Default Lower Left Corner
- DOD:** Department of Defense
- DTD:** Digital Terrain Data
- DTED:** Digital Terrain Elevation Data
- DTM:** Digital Terrain Model
- ECEF:** Earth-Centered Earth-Fixed
- EGM96:** Earth Gravitational Model 1996
- EnviFd:** ENVIronment FeDerate
- FLT:** OpenFlight
- GCS:** Geographic Coordinate Systems
- GIS:** Geographical Information System
- GM:** Earth's Gravitational Constant
- GPS:** Global Positioning System
- GRS80:** Geodetic Reference Spheroid of 1980
- GUI:** Graphical User Interface
- HLA:** High Level Architecture
- IERS:** International Earth Rotation and Reference Systems Service
- IRM:** IERS Reference Meridian
- IRP:** IERS Reference Pole
- IUGG:** International Union of Geodesy and Geophysics
- Lat:** Latitude
- Long:** Longitude

**M&S:** Modeling and Simulation

**MSB:** Most Significant Byte first

**MSL:** Mean Sea Level

**NAD27:** North American Datum of 1927

**NAD83:** North American Datum of 1983

**NASA/GSFC:** National Aeronautics and Space Administration's Goddard Space Flight Center

**NIMA:** The National Imagery and Mapping Agency

**Nm:** Nautical Mile

**NSTMSS:** Naval Surface Tactical Maneuvering Simulation System

**NSWCDD:** Naval Surface Warfare Center Dahlgren Division

**RTI:** RunTime Infrastructure

**USGS:** United States Geological Survey

**UTM:** Universal Transverse Mercator

**WGS:** World Geodetic System

# CHAPTER 1

## INTRODUCTION

Tactical simulations are designed to represent the real world of tactical operations. There are many features to a theory of tactic. However, basic functions of real world must be emulated in almost all simulations of tactic. These fundamental functions includes unit movement, weapon, communications etc. modeling. As a result, it is important to offer properly represented environment and tactical elements spatially for simulations.

The environment and models of simulations are strongly related to the coordinate system and reference model that is implemented.

2000 years ago, Greek scientists trying to understand our universe began to theorize about the shape of the earth and the placement of the stars. They created some of the first Geographic Coordinate Systems (GCS) ways to specifically identify locations on the earth or in the sky. Many different coordinate systems and associated earth reference models are in use in Modeling and Simulation (M&S) applications in the military today. These coordinate systems are based on a variety of geodetic datums, units, projections.

The choice of a coordinate system for simulation applications must be always both well informed and especially rational. Better representations of the geophysical area are needed to support the user, more realism for joint simulation, and M&S directives. Computational performance also effects the selection of reference models and coordinate systems.

This thesis presents a study implementing an appropriate world coordinate system for NSTMSS and also implementing body coordinate system to the ships of NSTMSS and its combination with world coordinate system. This way, it is aimed to improve the simulation behavior so that it can be more accurate, and new models can be integrated to the NSTMSS environment more easily.

## **1.1 PROBLEM DEFINITION**

The main issue of this thesis is to improve the coordinate systems of Naval Surface Tactical Maneuvering Simulation System (NSTMSS). This includes implementing a body coordinate system for ship model of NSTMSS and its combination with the improved coordinate system is another problem of this work.

Naval Surface Tactical Maneuvering Simulation System (NSTMSS) is a study of distributed interactive tactical simulation application [11][35]. NSTMSS is a virtual simulation system that is composed of a 3- dimensional ship handling simulator, a tactical level simulation of operational area, a virtual environment for the ship simulators, and simulation management processes. [11][35]

## **1.2 BACKGROUND**

In this section some technical information about coordinate systems and map projection methods related with this thesis is given.

### **1.2.1 COORDINATE SYSTEMS**

Simulations have become increasingly dependent on the availability of accurate and consistent geographic information to achieve navigation reliably. So, proper coordinate systems are needed to identify locations unambiguously. It also allows distances and areas to be reliably calculated. So, the challenge is to define a coordinate system with which the position of any topographic feature and models as an unambiguous set of numbers can be uniquely and accurately stated.

In this work, it is needed to implement an accurate and reliably coordinate system for more effective navigation in the simulation.

Also, for future uses a ship body coordinate system is needed in this work and its combination with the simulation coordinate system. With this body coordinate system it is possible the ship model to send the coordinate of any point of it's to other models interactively with the help of High Level Architecture (HLA) which is already available.

Briefly, this work consists of two parts; implementing a world coordinate system for the simulation and implementing a coordinate system for the ship models of the simulation.

## **THE EARTH REFERENCE MODEL**

Because the Earth is a very irregular and complex shape, it is needed to be modeled to map the positions of those details on which the coordinate system is based. The science of geodesy, on which all mapping and navigation is based, aims firstly to determine the shape and size of the simplified 'figure of the Earth' and goes on to determine the location of the features of the Earth's land surface from tectonic plates, coastlines and mountain ranges down to the control marks used for surveying and making maps. [2]

There are many reference model such as WGS 1984 (World Geodetic System 1984), WGS 1972 (World Geodetic System 1972), NAD27 (North American Datum of 1927), NAD83 (NAD, 1983).

## **TYPES OF COORDINATES**

- Global Cartesian coordinates (x,y,z): A system for the whole earth (i.e.,  $X= 1,109,928\text{m}$   $Y= -4,860,097\text{m}$   $Z= 3,965,162\text{m}$ )



- Geographic (geodetic) coordinates (latitude, longitude, height)  
(i.e., 38° 41' 08.73" N 077° 08' 08.37" W)
- Projected coordinates (x, y, z) on a local area of the earth's surface  
(i.e., 18 314251mE 4284069mN)

When the reference model is known, it is possible to convert between these representations.

## **1.2.2 MAP PROJECTION**

Map projection is the transformation of a curved earth to a flat map. A map projection cannot be a perfect representation, because it is not possible to show a curved surface on a flat map without creating distortions and discontinuities. Therefore different map projections are used for different applications. In geodesy, map coordinates tend only to be used for visual display purposes. When computations with coordinates latitude and longitude or Cartesian coordinates are needed, the results are converted to map coordinates as a final step if needed. This working procedure is in contrast to the practice in geographical information systems, where map coordinates are used directly for many computational tasks. [2]

### **TYPES OF PROJECTIONS**

- Conic
- Cylindrical (Transverse Mercator)
- Universal Transverse Mercator (UTM). The same type of projection with TM which is used in a worldwide mapping standard.
- Azimuthal (Lambert Azimuthal Equal Area)

### **1.2.3 THE OLD NSTMSS COORDINATE SYSTEM**

This section describes the coordinate system of NSTMSS before this thesis's improvements.

#### **NSTMSS REFERENCE MODEL**

NSTMSS has a flat earth reference model. If the area being mapped is small (for example, a 10 Km square), flat earth is usable, but the area mapped in this work is larger than this. The 'flat earth' approach breaks down rapidly when larger areas are to be measured or mapped. There are two complicating factors, being:

- The curvature of the Earth,
- The composition of the Earth's interior.

So, flat earth reference model is not suitable for this work.

#### **NSTMSS COORDINATE SYSTEM**

It has a coordinate system that is a two dimensional and local. Transformations from this coordinates to geodetic coordinates are simplified, so the precision is low. Figure 1.1 shows this transformation. NSTMSS coordinate system expressed in its document;

*“Degrees of latitude can be translated directly into a linear displacement (1 degree is approximately 60 nautical miles), while the linear displacement associated with the degrees of longitude varies with latitude. This analogy is used for illustrative purposes only. The problem of transforming from geodetic coordinates to geocentric coordinates has received much attention for more accuracy”*

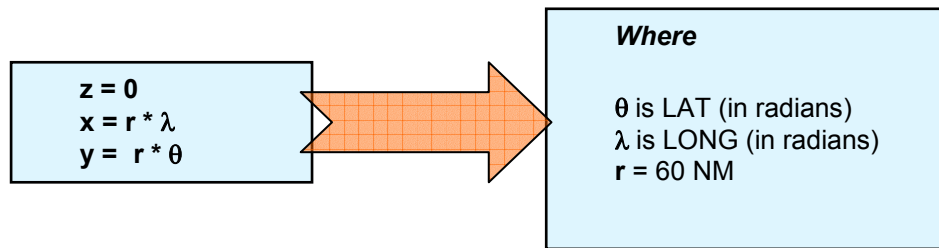


Figure 1.1 Transformation Between Geodetic and Geocentric Coordinate Systems in NSTMSS [11]

So, a more accurate coordinate system is needed.

### **NSTMSS SHIP BODY COORDINATE SYSTEM**

NSTMSS has a static coordinates on specific 7 points on the ship which is hard coded. Coordinate of any other point on the ship is not definable.

### **1.2.4 THE NEW NSTMSS COORDINATE SYSTEM**

To improve NSTMSS world coordinate system these processes are need

- a reference model to identify the Earth, so it can be readily associated with the physical world so that its use is intuitive.
- a proper map projection method
- accurate converting methods between needed coordinate presentations

In this work, World Geodetic System 1984 (WGS 84) was used as the reference model. WGS 84 represents NIMA's best geodetic model of the Earth using data, techniques and technology available through 1996. The WGS 84 represents the best global geodetic reference system for the Earth available at this time for practical applications of mapping, charting, geopositioning and navigation. [3]

UTM was adopted as the map projection method which is good for ocean navigation.

Finally, UTM to latitude, longitude and latitude, longitude to UTM representation conversion methods have been implemented.

To improve NSTMSS body coordinate system it is needed to

- Choosing a appropriate coordinate system for the ship model
- Using a method like Euler or Quaternion for the rotation of the ship body coordinate system to be able to define the coordinate system rotations as the ship is rotating and moving in real time.
- Using the technology and methods explained above it is aimed to implement a reliable coordinate system for navigation system of NSTMSS starting from scratch.

## **1.3 TOOLS**

### **1.3.1 DEVELOPMENT TOOLS**

This section presents a brief explanation about the tools used in the development phase of this work.

#### **MULTIGEN CREATOR (V2.5.1)**

MultiGen Creator is a software for creating highly optimized, high fidelity realtime 3D content for use in visual simulation. Creator Terrain features, software's unique workflow process, geospatial data management infrastructure and user interface allow customers the ability to deliver large-area terrain databases that satisfy varying simulation. The products' approach to building terrain databases revolves around the construction, maintenance and editing of a Common Database. The Common Database is comprised of structured and optimized data in industry

maintained open formats such as the MultiGen-Paradigm OpenFlight®. [21] NSTMSS uses terrain representations in Multigen-Paradigm OpenFlight format. New coordinate system for NSTMSS in this thesis work was tested on new terrains that are created using Multigen Creator tool. With the help of Creator, terrain data from different formats like Digital Elevation Data (.ded) is converted to OpenFlight (.flt) format and loaded to NSTMSS.

### **1.3.2 TESTING TOOLS**

This section presents a brief explanation about the tools used in the testing phase of this work.

#### **GEOTRANS (v2.0)**

GEOTRANS (Geographic Translator) is an application program developed by National Imagery and Mapping Agency (NIMA) which allows converting geographic coordinates among a wide variety of coordinate systems, map projections, and datums. The user interface of GEOTRANS is similar to that of a calculator. To convert a set of coordinates, selecting the coordinate system or map projection, and the datum, in which coordinates are defined, entering the source coordinates, selecting the coordinate system or map projection, and the datum, to which the coordinates to be converted, and clicking on the Convert button is enough. The resulting coordinates are displayed. Currently, twenty-five different coordinate systems, map projections, grids, and coding schemes, and over two hundred different datums, are supported. [20]

To check the program result for the conversion between UTM representation and latitude/longitude presentation this tool was used manually.

## 1.4 THESIS OUTLINE

The former sections of this chapter give a brief definition of problem, objectives, scope and background. The further chapters are arranged as follows:

- Chapter 2 gives general technical information about coordinate systems and the coordinate systems adopted in this thesis.
- Chapter 3 presents general technical information about map projection methods and the adopted map projection method in this work. Then, brief information is given about digital terrains. Also, it describes which methods were used in the implementation for which reason.
- Chapter 4 is dedicated to a detailed explanation of the implementation. This chapter answers the questions how world and ship body coordinate systems were implemented, how the new classes were integrated to NSTMSS and which parts of NSTMSS were modified.
- Chapter 5 presents a brief explanation about how the new features added to NSTMSS were tested.
- Chapter 6 outlines conclusion reached as a result of this research as well as recommendations for future research in this area.
- Appendix-A covers WGS 84 parameters of the Earth definition.
- Appendix-B gives Ship Body Coordinate System test input data.
- Appendix-C includes a table which lists the lengths in feet of 1 minute, 1 second and  $\frac{1}{1000}$  minute of latitude and longitude of the Earth according to WGS 84.
- Appendix-D shows position shifts from datum differences in a figure.
- Appendix-E lists datums used all over the world and areas and ellipsoids of these datums.
- Appendix-F is updated NSTMSS Installation Manual. It is a by-product of the thesis.

## **CHAPTER 2**

### **COORDINATE SYSTEMS**

This chapter gives general technical information about coordinate systems and the coordinate system used in this work. Also, it introduces related works about these issues.

Coordinate systems and many other concepts it includes are presented in three parts; world coordinate system, local coordinate system and body coordinate system.

This work improved two different coordinate systems in NSTMSS, so it is possible to divide the technical information in this chapter into two parts; world coordinate system and coordinate system for the ship model.

Technical information presented in this chapter extracted from the following sources: [2], [3], [4], [6], [7], [8], [11], [12], [15], [18], [22], [26], [27], [28], [29], [30], [31], [32], [36], [37].

#### **2.1 WORLD COORDINATE SYSTEM**

Several important concepts must be given firstly to explain the world coordinate system of this work. This section explains these titles shape of the earth, types of coordinates, datum and World Geodetic System.

### 2.1.1 THE SHAPE OF THE EARTH

When all the topographic and oceanographic details are reviewed, the Earth is a very irregular and complex shape. If the positions of those details are wanted to map, a simpler model of the basic shape of the Earth is needed, sometimes called the ‘figure of the Earth’, on which the coordinate system will be based. The details can then be added by determining their coordinates relative to the simplified shape, to build up the full picture.

The science of geodesy, on which all mapping and navigation is based, aims firstly to determine the shape and size of the simplified ‘figure of the Earth’ and goes on to determine the location of the features of the Earth’s land surface - from tectonic plates, coastlines and mountain ranges down to the control marks used for surveying and making maps. Hence geodesists provide the fundamental ‘points of known coordinates’ which cartographers and navigators take as their starting point. The first question of geodesy, then, is ‘What is the best basic, simplified shape of the Earth?’ Having established this, it can be used as a reference surface, with respect to measured topography.

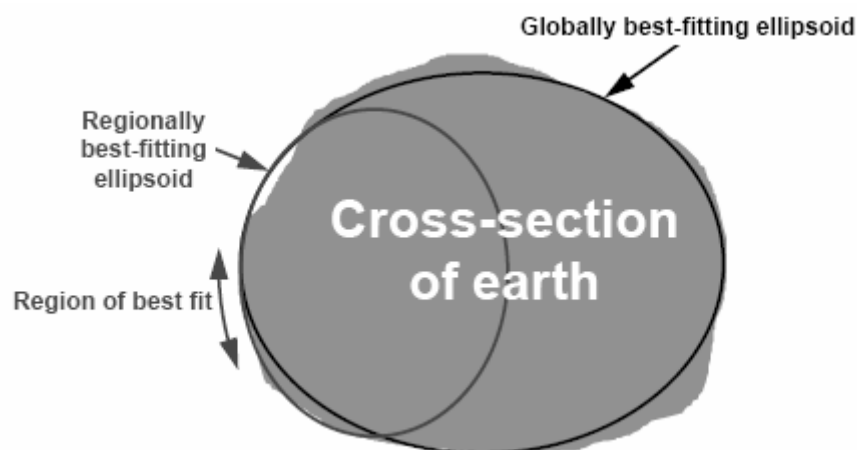


Figure 2.1 Shape of the Earth [2]



Geodesists have two very useful answers to this question: ellipsoids and the Geoid.  
[2]

Figure 2.1 shows shape of the Earth and example ellipsoids.

## **ELLIPSOIDS**

The Earth is very nearly spherical. However, it has a tiny equatorial bulge making the radius at the equator about one third of one percent bigger than the radius at the poles. Therefore the simple geometric shape which most closely approximates the shape of the Earth is a biaxial ellipsoid, which is the three-dimensional figure generated by rotating an ellipse about its shorter axis (less exactly, it is the shape obtained by squashing a sphere slightly along one axis). The shorter axis of the ellipsoid approximately coincides with the rotation axis of the Earth.

Many determinations of the best-fitting ellipsoid have been made. Several hundred ellipsoids have been defined for scientific purposes and about 30 are in use today for mapping. Examples include Airy 1830, International 1924 and the Geodetic Reference Spheroid of 1980 (GRS80). [12]

Because the ellipsoid shape does not fit the Earth perfectly, there are lots of different ellipsoids in use, some of which are designed to best fit the whole Earth, and some to best fit just one region. For instance, the coordinate system used with the Global Positioning System (GPS) uses an ellipsoid called GRS80 (Geodetic Reference System 1980) which is designed to best-fit the whole Earth. The ellipsoid used for mapping in Britain, the Airy 1830 ellipsoid, is designed to best-fit Britain only, which it does better than GRS80, but it is not useful in other parts of the world. So, various ellipsoids used in different regions differ in size and shape, and also in orientation and position relative to each other and to the Earth. The modern trend is to use GRS80 everywhere for reasons of global compatibility. Hence the local best-fitting ellipsoid is now rather an old-fashioned idea, but it is still important because many such ellipsoids are built into national mapping coordinate systems. [2]

## THE GEOID

If heights wanted to be measured, an imaginary surface of ‘zero height’ somewhere underneath is needed which the measurements will be referred. The stated height of any point is the vertical distance above this imaginary surface.

The fact that increasing heights on the map are taken to mean ‘uphill’ and decreasing heights on the map are taken to mean ‘downhill’ implies that the height reference surface must be a level surface that is, everywhere at right angles to the direction of gravity (see Figure 2.2) It is clear that if it is mentioned about the heights of places over the whole world, the reference surface must be a closed shape, and it will be something like the shape of an ellipsoid. Its exact shape will be defined by the requirement to be at right angles to the direction of gravity everywhere on its surface. Level surfaces are not simple geometrical shapes. The direction of gravity, although generally towards the centre of the Earth, varies in a complex way on all scales from global to very local. This means that a level reference surface is not a simple geometric figure like the ellipsoid, but bumpy and complex. Level surfaces can be determined from physical observations such as precise gravity measurements. This is the scientific study known as gravimetry.

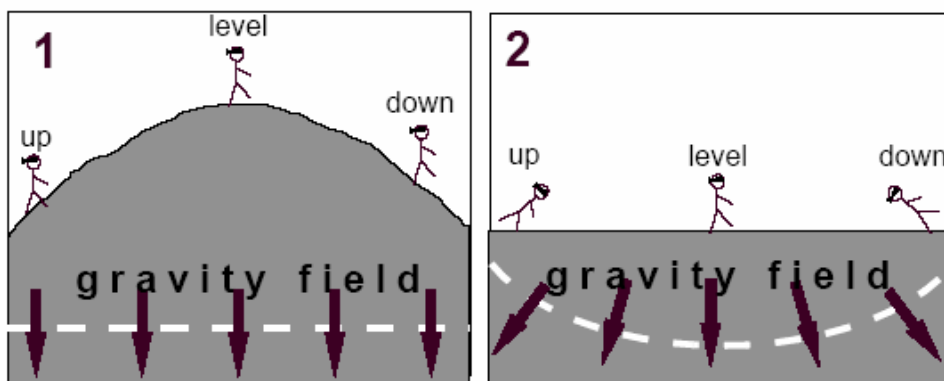


Figure 2.2 Gravity [2]

Depending on what height is chosen as ‘zero height’, there are any numbers of closed level surfaces that can be chosen as the global height reference surface, and the choice is essentially arbitrary. These level surfaces can be thought of like layers of an onion inside and outside the Earth’s topographic surface. Each one corresponds to a different potential energy level of the Earth’s gravitational field, and each one, although an irregular shape, is a surface of constant height. The one is chosen as the height reference surface is that level surface which is closest to the average surface of all the world’s oceans. This irregular three-dimensional shape is called the Geoid. Although it is both imaginary and difficult to measure, it is a single unique surface: it is the only level surface which best-fits the average surface of the oceans over the whole Earth. This is by contrast with ellipsoids, of which there are many fitting different regions of the Earth.

The Geoid is very nearly an ellipsoid shape. A best-fitting ellipsoid matches the Geoid to better than two hundred meters everywhere on its surface. However, that is the best that can be done with an ellipsoid, and usually the height is wanted to known much better than that. The Geoid has the property that every point on it has exactly the same height, throughout the world, and it is never more than a couple of meters from local mean sea level. This makes it the ideal reference surface on which to base a global coordinate system for vertical positioning. The Geoid is in many ways the true ‘figure of the Earth’ because a fundamental level surface is intrinsic to the view of the world, living in a powerful gravity field. [2]

Figure 2.3 shows ellipsoid and geoid representation which is explained above sections.

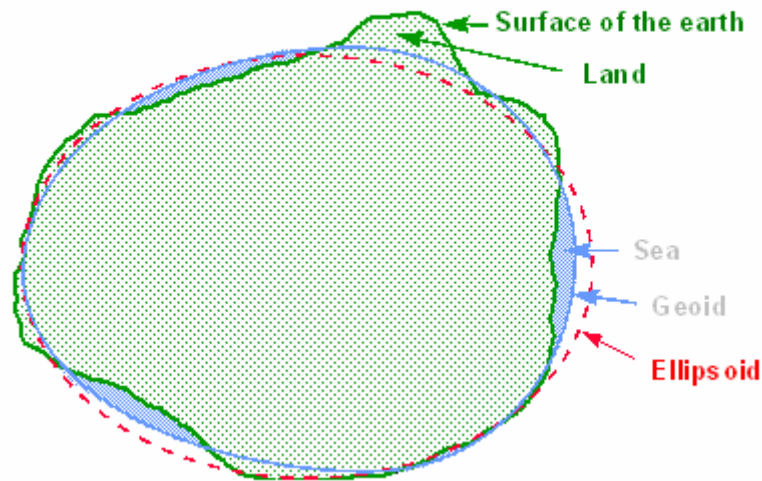


Figure 2.3 Geoid and Ellipsoid [12]

### Local Geoids

Height measurements on maps are usually stated to be height above mean sea level. This means that a different level surface has been used as the ‘zero height’ reference surface, one based on a tide - gauge local to the mapping region rather than the average of global ocean levels. For most purposes, these local reference surfaces can be considered to be parallel to the Geoid but offset from it, sometimes by as much as two meters. For instance, heights in mainland Britain are measured relative to the tide-gauge in Newlyn, Cornwall, giving a reference surface which is about 80 centimeters below the Geoid. What causes this discrepancy between average global ocean levels and local mean sea levels? Pure water left undisturbed does form a level surface, so the two should be in agreement. The problem is that the sea around the coasts is definitely not left undisturbed. The oceanic currents, effects of tides and winds on the coast, and variations in water temperature and purity all cause ‘mean sea level’ to deviate slightly from the truly level Geoid surface. So, the mean sea level surface contains very shallow hills and valleys which are described by the apt term ‘sea surface topography’. The sea around Britain happens to form a ‘valley’ in the sea surface, so the mean sea level is about 80 centimeters below the Geoid. Different countries have adopted different local mean sea levels as their

‘zero height’ definition. Consequently there are many ‘zero height’ reference surfaces used in different parts of the world which are (almost) parallel to, but offset from, the true global Geoid. These reference surfaces are sometimes called ‘local geoids’. [2]

## **2.1.2 TYPES OF COORDINATES**

There are three types of coordinates

- Latitude, longitude and ellipsoid height
- Cartesian coordinates
- Eastings and northings

Here is an explanation of these coordinate types.

### **LATITUDE, LONGITUDE AND ELLIPSOID HEIGHT**

The most common way of stating terrestrial position is with two angles, latitude and longitude. These define a point on the globe. More correctly, they define a point on the surface of an ellipsoid which approximately fits the globe. Therefore, to use latitudes and longitudes with any degree of certainty the ellipsoid that is being dialed with must be known. Figure 2.4 shows an ellipsoid with graticule of latitude and longitude and the associated 3-D Cartesian axes. The relationship between the ellipsoid and latitude and longitude is simple. North-south lines of constant longitude are known as meridians, and east-west lines of constant latitude are parallels. One meridian of the ellipsoid is chosen as the prime meridian and assigned zero longitude. The longitude of a point on the ellipsoid is the angle between the meridian passing through that point and the prime meridian. Usually the scale of longitude is divided into eastern and western hemispheres (hemiellipsoids, actually) from 0 to 180 degrees West and 0 to 180 degrees East. The equator of the ellipsoid is chosen as the circle of zero latitude. The latitude of a point is the angle between the equatorial plane and the line perpendicular to the

ellipsoid at that point. Latitudes are reckoned as 0 to 90 degrees North and 0 to 90 degrees South, where 90 degrees either North or South is a single point the pole of the ellipsoid. So, latitude and longitude give a position on the surface of the stated ellipsoid. Since real points on the ground are actually above (or possibly below) the ellipsoid surface, a third coordinate is needed, the ellipsoid height, which is simply the distance from the point to the ellipsoid surface along a straight line perpendicular to the ellipsoid surface. The term ‘ellipsoid height’ is actually a misnomer, because although this is an approximately vertical measurement, it does not give true height because it is not related to a level surface. It does however unambiguously identify a point in space above or below the ellipsoid surface in a simple geometrical way, which is its purpose.

With the coordinate triplet of latitude, longitude and ellipsoid height, a point with respect to a stated ellipsoid can be unambiguously positioned. To translate this into an unambiguous position on the ground, it is needed to know accurately where the ellipsoid is relative to the piece of ground that is interested in. [2]

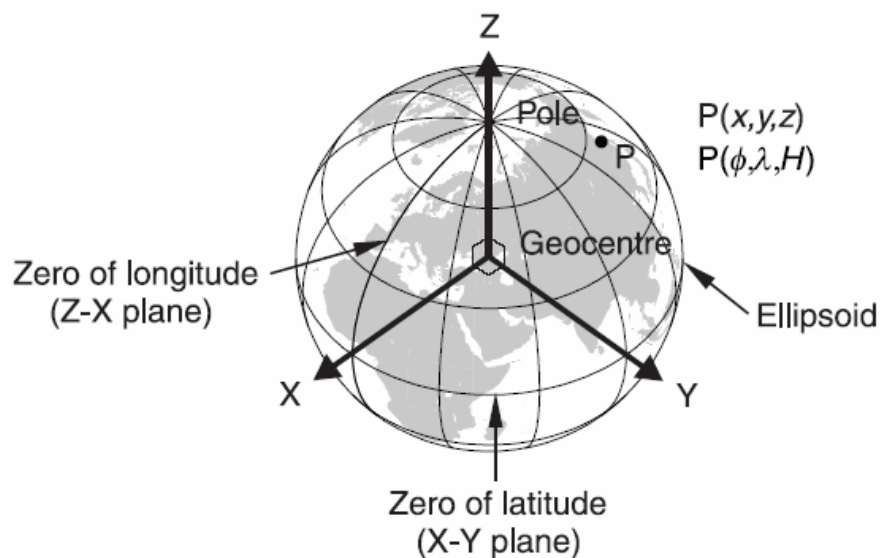


Figure 2.4 An ellipsoid with latitude and longitude and the associated 3-D Cartesian axes.[2]

The example in Figure 2.4 puts the origin at the Geocentre (the centre of mass of the Earth), but this is not always the case. This system allows position of point P to be stated as either latitude, longitude, ellipsoid height H or Cartesian coordinates X, Y and Z – the two types of coordinates give the same information. [2]

Figure 2.5, 2.6 and 2.7 shows geodetic latitude, longitude and height.

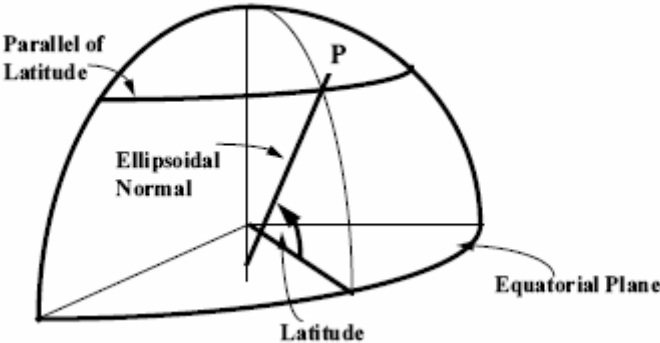


Figure 2.5 Geodetic Latitude [6]

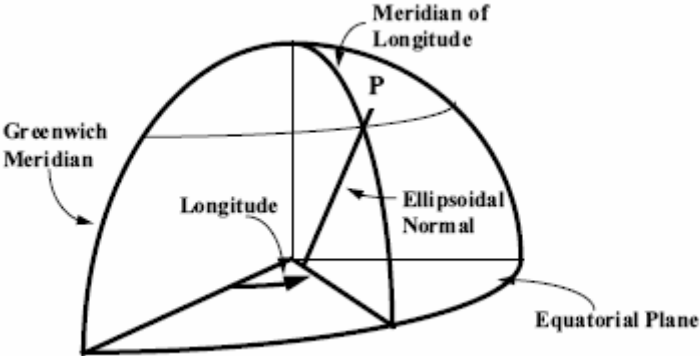


Figure 2.6 Geodetic Longitude [6]

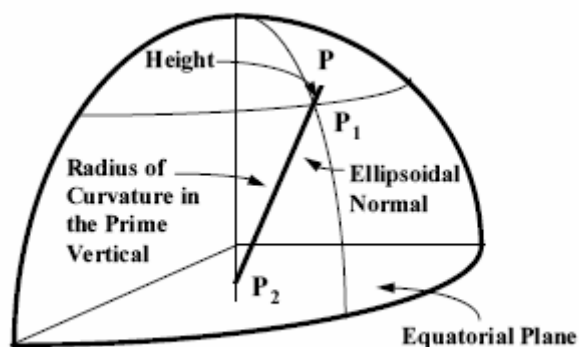


Figure 2.7 Spheroidal Height [6]

## CARTESIAN COORDINATES

Rectangular Cartesian coordinates are a very simple system of describing position in three dimensions, using three perpendicular axes X, Y and Z. Three coordinates unambiguously locate any point in this system. It can be used as a very useful alternative to latitude, longitude and ellipsoid height to convey exactly the same information.

Three Cartesian axes are used aligned with the latitude and longitude system (see Figure 2.4). The origin (centre) of the Cartesian system is at the centre of the ellipsoid. The X axis lies in the equator of the ellipsoid and passes through the prime meridian (0 degrees longitude). The negative side of the X axis passes through 180 degrees longitude. The Y axis also lies in the equator but passes through the meridian of 90 degrees East, and hence is at right angles to the X axis. Obviously, the negative side of the Y axis passes through 90 degrees West. The Z axis coincides with the polar axis of the ellipsoid; the positive side passes through the North Pole and the negative side through the South Pole. Hence it is at right angles to both X and Y axes.

It is clear that any position uniquely described by latitude, longitude and ellipsoid height can also be described by a unique triplet of 3-D Cartesian coordinates, and vice versa.



It is important to remember that having converted latitude and longitude to Cartesian coordinates; the resulting coordinates are relative to a set of Cartesian axes which are unique to the ellipsoid concerned.

They cannot be mixed with Cartesian coordinates associated with any other ellipsoid or coordinate system without first applying a transformation between the two coordinate systems. When considering using coordinates from different sources together, beware that one named coordinate system can have several different realizations which are not necessarily compatible with each other.

Similarly, having converted Cartesian coordinates to latitude, longitude and ellipsoid height, the resulting coordinates are relative to the ellipsoid chosen, and also to the Cartesian reference system of the input coordinates. They cannot be used together with latitudes, longitudes and ellipsoid heights associated with any other ellipsoid or coordinate system, without first applying a suitable transformation.

### **Geoid Height (Orthometric Height)**

The term ellipsoid height is misleading because a distance above a reference ellipsoid does not necessarily indicate height – point A can have a greater ellipsoid height than point B while being downhill of B. This is because the ellipsoid surface is not level – therefore a distance above the ellipsoid is not really a height at all. The reference surface which is everywhere level is the Geoid. To ensure that the relative height of points A and B correctly indicates the gradient between them, height must be measured as the distance between the ground and the Geoid, not the ellipsoid. This measurement is called ‘orthometric height’ or simply ‘Geoid height’. The relationship between ellipsoid height  $H$  and Geoid height (orthometric height)  $h$  is

$$H = h + N$$

where  $N$  is (reasonably enough) the ‘Geoid-ellipsoid separation’. Because the Geoid is a complex surface,  $N$  varies in a complex way depending on latitude and

longitude. A look-up table of  $N$  for any particular latitude and longitude is called a Geoid model. Therefore a Geoid model is needed to convert ellipsoid height to Geoid height and vice-versa. Figure 2.8 shows these quantities for two points A and B. The orthometric height difference between A and B is

$$\Delta h_{AB} = h_B - h_A = \Delta H_{AB} - \Delta N_{AB}$$

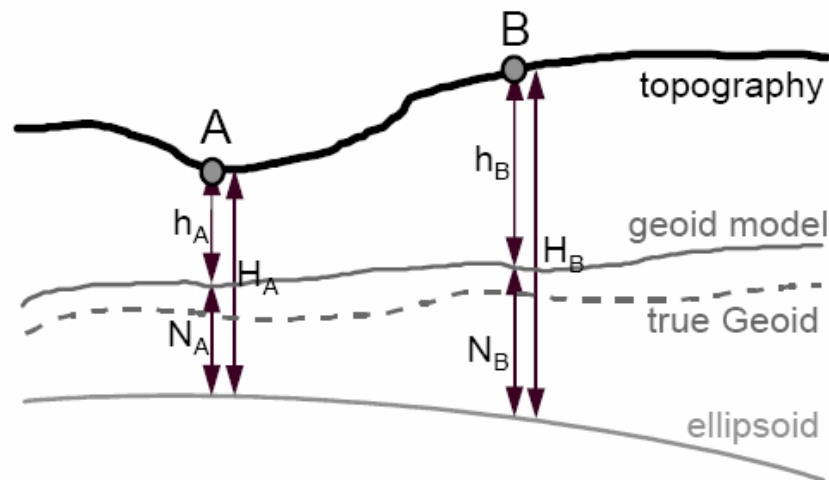


Figure 2.8 Geoid Height [2]

At any point there is a geoid height for every geodetic datum relevant to that point. For a particular geodetic datum, the geoid height value changes continuously with change in horizontal position. An exact value for geoid heights relative to the ellipsoidal surface for a geodetic datum is not usually known. However, for some geodetic datums a model of the geoid heights has been built, and approximate geoid heights can be interpolated across the model. With a locally well-fitting and well-positioned ellipsoid, the values of geoid height will be close to zero. For global geodetic datums such as WGS 84, geoid height values range between +50 to -50 meters. For non-global geodetic datums covering extensive areas, especially if near mountains or if using an old ellipsoid adopted from measurements elsewhere in the world, the absolute values of geoid height can exceed 200 meters in extreme cases. Gravity-related vertical co-ordinate systems are measured relative to a vertical

datum. The vertical datum will be taken to be the geoid, and will usually be mean sea level at a particular location or series of locations over a particular period of time. As with geodetic datums, the parameters required to define a vertical datum can vary and can be complex, but for practical geographical information processing whilst the definition of the vertical datum may be useful it is not essential. However it is critical that the vertical datum is identified. This is because if the datum surface is changed, the values of heights measured from that surface will change. [2]

The difference between geoid and ellipsoid heights are shown in Figure 2.9.

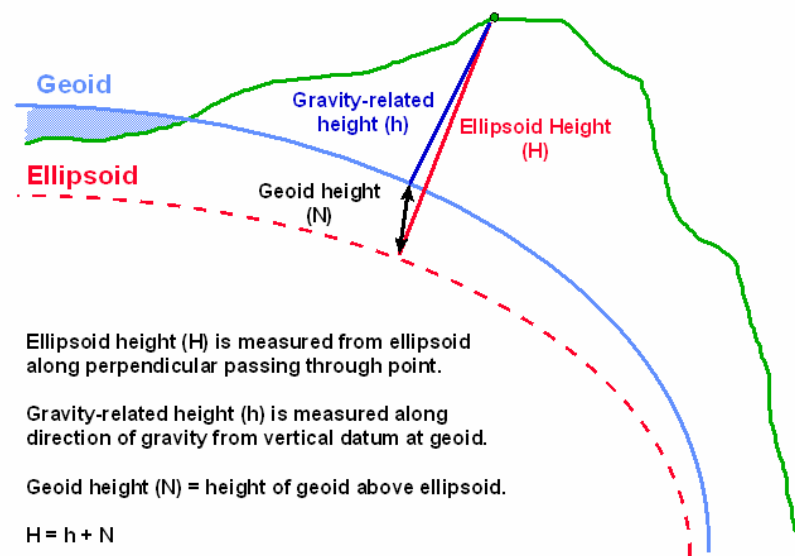


Figure 2.9 Geoid Heights [12]

## EASTINGS AND NORTHINGS

The last type of coordinates that is need to be considered is eastings and northings, also called plane coordinates, grid coordinates or map coordinates. These coordinates are used to locate position with respect to a map, which is a two-dimensional plane surface depicting features on the curved surface of the Earth. These days the ‘map’ might be a computerized Geographical Information System (GIS) but the principle is exactly the same. Map coordinates use a simple 2-D

Cartesian system in which the two axes are known as eastings and northings. Map coordinates of a point are computed from its ellipsoidal latitude and longitude by a standard formula known as a map projection.

A map projection cannot be a perfect representation, because it is not possible to show a curved surface on a flat map without creating distortions and discontinuities. Therefore different map projections are used for different applications. Any coordinates stated as eastings and northings should be accompanied by an exact statement of the map projection used to create them. [6] There is more about map projections in section “Map Projections”.

### **2.1.3 DATUM**

No matter what type of coordinates that are being used, it is required a suitable origin with respect to which the coordinates are stated. For instance, Cartesian coordinates can not be used without defining an origin point of the coordinate axes and defining the directions of the axes in relation to the Earth that are being measured. This is an example of a set of conventions necessary to define the spatial relationship of the coordinate system to the Earth. The general name for this concept is geodetic datum.

To use 3-D Cartesian coordinates, a 3-D datum definition is required, in order to set up the three axes, X, Y and Z. The datum definition must somehow state where the origin point of the three axes lies and in what directions the axes point, all in relation to the surface of the Earth. Each point on the Earth will then have a unique set of Cartesian coordinates in the new coordinate system. The datum definition is the link between the ‘abstract’ coordinates and the real physical world.

To use latitude, longitude and ellipsoid height coordinates, the same type of datum used for 3-D Cartesian coordinates are started with. To this a reference ellipsoid is added centered on the Cartesian origin (as in Figure 2.4), the shape and size of which is added to the datum definition. The size is usually defined by stating the

distance from the origin to the ellipsoid equator, which is called the semi-major axis  $a$ . The shape is defined by any one of several parameters: the semi-minor axis length  $b$  (the distance from the origin to the ellipsoid pole), the squared eccentricity  $e^2$ , or the inverse flattening  $f$ . Each conveys the same information: the shape of the chosen reference ellipsoid.  $a$  and  $b$  parameters of an ellipsoid is shown in figure 2.10

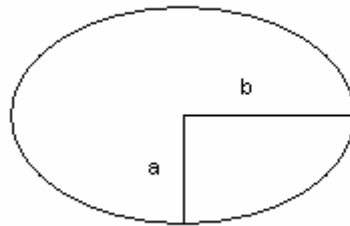


Figure 2.10 Semi major and minor axes of an ellipsoid

Formulas of the squared eccentricity  $e^2$  and the inverse flattening  $f$  are

$$f = a - b/a$$

$$e^2 = 2f - f^2$$

the datum definition consists of eight parameters:

- the 3-D location of the origin (three parameters)
- the 3-D orientation of the axes (three parameters)
- the size of the ellipsoid (one parameter)
- the shape of the ellipsoid (one parameter).[2]

## **TYPES OF DATUMS**

Datums are usually classified into two categories. These are known as horizontal and vertical datums. Horizontal datums are also classified as local geodetic datums and geocentric datums.

### **Horizontal Datums**

The horizontal datum is the model used to measure positions on the earth. A specific point on the earth can have substantially different coordinates, depending on the datum used to make the measurement. There are hundreds of locally-developed horizontal datums around the world, usually referenced to some convenient local reference point. Contemporary datums, based on increasingly accurate measurements of the shape of the earth, are intended to cover larger areas. The WGS84 datum is a common standard datum.

### **Local Datums**

Local Datum is a datum which best approximates the size and shape of a particular part of the earth's sea-level surface. Invariably, the centre of its spheroid will not coincide with the Earth's centre of mass.

Until very recently, most country's spatial information systems were based on local geodetic datums.

The Australian Geodetic Datum (AGD) is an example of a local datum. Its spheroid is a good approximation to the size and shape of the sea-level surface in the region of Australia but a poor approximation in other parts of the world (see Figure 2.11).

[6]

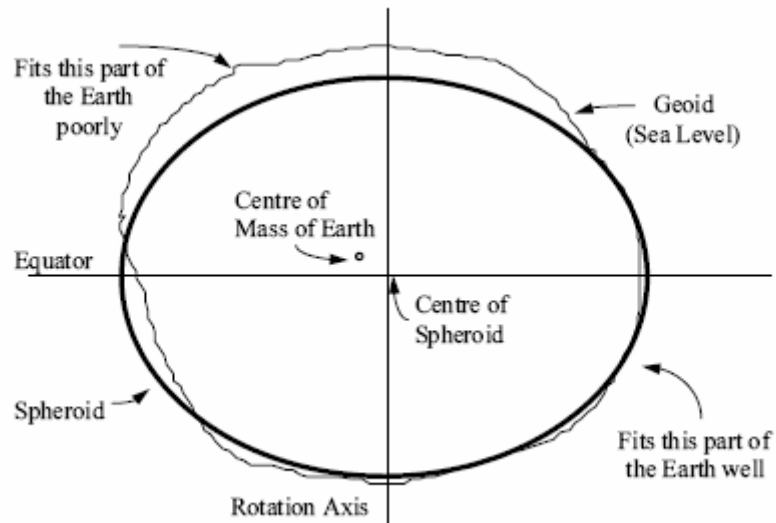


Figure 2.11 Local Geodetic Datum [6]

### **Geocentric Datum**

Geocentric Datum is one which best approximates the size and shape of the Earth as a whole. The centre of its spheroid coincides with the Earth's centre of mass (see Figure 2.12).

Geocentric datums do not seek to be a good approximation to any particular part of the Earth. Rather, their application lies in projects or undertakings which have global application. [6]

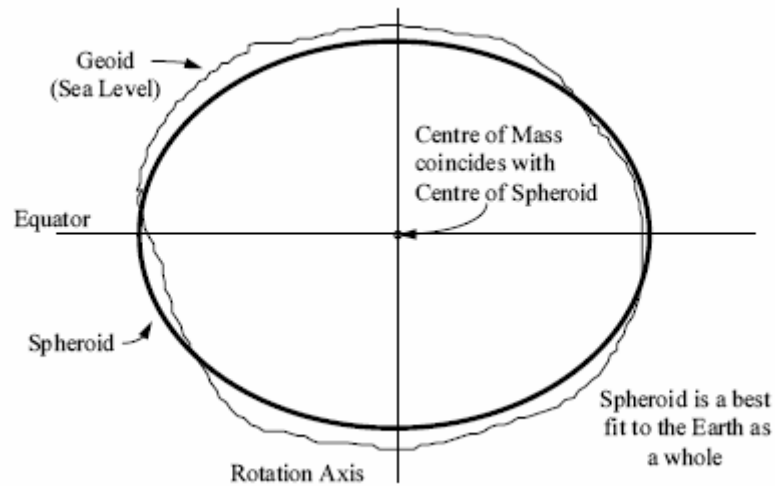


Figure 2.12 Geocentric Geodetic Datum [6]

## Vertical datums

A vertical datum is used for measuring the elevations of points on the earth's surface. Vertical datums are either tidal, based on sea levels, or geodetic, based on the same ellipsoid models of the earth used for computing horizontal datums.

In common usage, elevations are often cited in height above sea level; this is a widely used tidal datum. Because ocean tides cause water levels to change constantly, the sea level is generally taken to be some average of the tide heights. A geodetic vertical datum takes some specific zero point, and computes elevations based on the geodetic model being used, without further reference to sea levels. Usually, the starting reference point is a tide gauge, so at that point the geodetic and tidal datums might match, but due to sea level variations, the two scales may not match elsewhere. [22]

Briefly, the zero surface to which elevations or heights are referred is called a vertical datum. Traditionally, surveyors and mapmakers have tried to simplify the task by using the average (or mean) sea level as the definition of zero elevation, because the sea surface is available worldwide. The mean sea level (MSL) is



determined by continuously measuring the rise and fall of the ocean at "tide gauge stations" on seacoasts for a period of about 19 years. This averages out the highs and lows of the tides caused by the changing effects of the gravitational forces from the sun and moon which produce the tides. The mean sea level (MSL) then is defined as the zero elevation for a local or regional area. [15]

There many datum used all over the world. Appendix E is a list of datums used these datums, their areas and ellipsoids. Appendix D also shows the position shifts from datum differences.

In this work, vertical datum is based on sea-level, and the mean sea level is taken as the reference as zero.

#### **2.1.4 WORLD GEODETIC SYSTEM 1984 (WGS84)**

The National Imagery and Mapping Agency (NIMA) supports a large number and variety of products and users, which makes it imperative that these products all be related to a common worldwide geodetic reference system. This ensures interoperability in relating information from one product to another, supports increasingly stringent accuracy requirements and supports military and humanitarian activities worldwide. The refined World Geodetic System 1984 represents NIMA's best geodetic model of the Earth using data, techniques and technology available through 1996.

The definition of the World Geodetic System has evolved within NIMA and its predecessor agencies from the initial WGS 60 through subsequent improvements embodied in WGS 66, WGS 72 and WGS 84. The improved Earth Gravitational Model 1996 (EGM96), its associated geoid and additional datum transformations have been made possible by the inclusion of these new data. EGM96 was developed jointly by NIMA, the National Aeronautics and Space Administration's Goddard Space Flight Center (NASA/GSFC), The Ohio State University and the Naval Surface Warfare Center Dahlgren Division (NSWCDD). Commensurate with these

modeling enhancements, significant improvements in the realization of the WGS 84 reference frame have been achieved through the use of the NAVSTAR Global Positioning System (GPS). WGS 84 is realized by the coordinates assigned to the GPS tracking stations used in the calculation of precise GPS orbits at NIMA. NIMA currently utilizes the five globally dispersed Air Force operational GPS tracking stations augmented by seven tracking stations operated by NIMA. The coordinates of these tracking stations have been determined to an absolute accuracy of  $\pm 5$  cm.

The WGS 84 represents the best global geodetic reference system for the Earth available at this time for practical applications of mapping, charting, geopositioning and navigation.

This section includes the definition of the WGS 84, definition of WGS 84 ellipsoid and realization of WGS 84 [3].

## **DEFINITION**

It is geocentric, the center of mass being defined for the whole Earth including oceans and atmosphere

- Its scale is that of the local Earth frame, in the meaning of a relativistic theory of gravitation
- Its orientation was initially given by the Bureau International de l'Heure (BIH) orientation of 1984.0
- Its time evolution in orientation will create no residual global rotation with regards to the crust

The WGS 84 Coordinate System is a right-handed, Earth-fixed orthogonal coordinate system and is graphically depicted in Figure 2.13. [3]

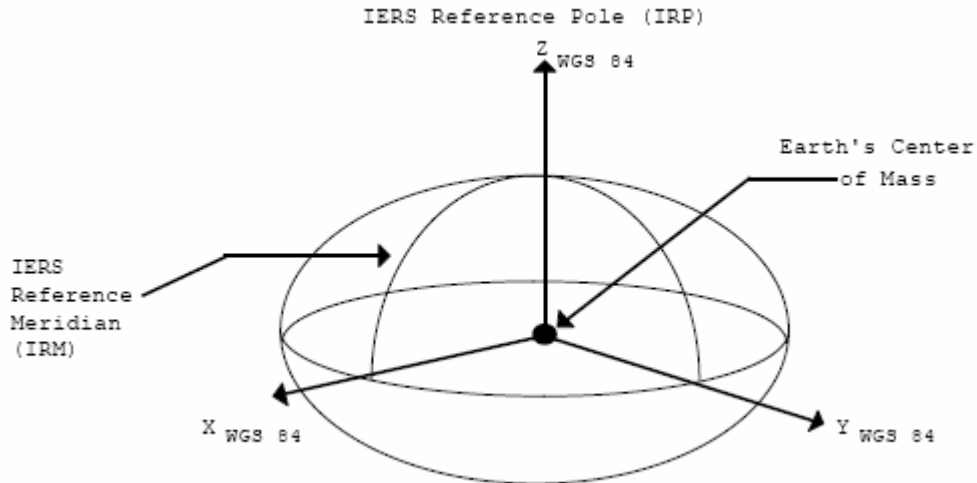


Figure 2.13 The WGS 84 Coordinate System Definition [3]

In Figure 2.13 the origin and axes are defined as follows:

Origin = Earth's center of mass

Z-Axis = The direction of the International Earth Rotation and Reference Systems Service (IERS) Reference Pole (IRP). This direction corresponds to the direction of the BIH Conventional Terrestrial Pole (CTP) (epoch 1984.0) with an uncertainty of 0.0005" [8]

X-Axis = Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis. The IRM is coincident with the BIH Zero Meridian (epoch 1984.0) with an uncertainty of 0.0005" [8]

Y-Axis = Completes a right-handed, Earth-Centered Earth-Fixed (ECEF) orthogonal coordinate system.[3]

## WGS 84 ELLIPSOID

While selecting the WGS 84 Ellipsoid and associated parameters, the original WGS 84 Development Committee decided to closely adhere to the approach used by the International Union of Geodesy and Geophysics (IUGG), when the latter established and adopted Geodetic Reference System 1980 (GRS 80) [3]. Accordingly, a geocentric ellipsoid of revolution was taken as the form for the WGS 84 Ellipsoid. The parameters selected to originally define the WGS 84 Ellipsoid were the semi-major axis ( $a$ ), the Earth's gravitational constant ( $GM$ ), the normalized second degree zonal gravitational coefficient and the angular velocity of the Earth. [3]

### Defining parameters

Table 2.1 WGS 84 four defining parameters

Table 2.1 WGS84 ellipsoid parameters [3]

Parameter	Notation	Value
Semi-major Axis	$a$	6378137.0 meters
Reciprocal of Flattening	$1/f$	298.257223563
Angular Velocity of the Earth	$\omega$	$7292115.0 \times 10^{-11}$ rad/s
Earth's Gravitational Constant (Mass of Earth's Atmosphere Included)	$GM$	$3986004.418 \times 10^8$ m <sup>3</sup> /s <sup>2</sup>

### Semi-major Axis ( $a$ )

The semi-major axis ( $a$ ) is one of the defining parameters for WGS 84. Its adopted value is:

$$a = 6378137.0 \text{ meters}$$

This value is the same as that of the GRS 80 Ellipsoid. As stated in [7], the GRS 80, and thus the WGS 84 semi-major axis is based on estimates from the 1976-1979 time periods determined using laser, Doppler and radar altimeter data and techniques. Although more recent, improved estimates of this parameter have become available, these new estimates differ from the above value by only a few decimeters. More importantly, the vast majority of practical applications such as GPS receivers and mapping processes use the ellipsoid as a convenient reference surface. In these applications, it is not necessary to define the ellipsoid that best fits the geoid. Instead, common sense and the expense of numerous software modifications to GPS receivers and mapping processes guided the decision to retain the original reference ellipsoid. Moreover, this approach obviates the need to transform or re-compute coordinates for the large body of accurate geospatial data which has been collected and referenced to the WGS 84 Ellipsoid in the last decade. Highly specialized applications and experiments which require the ‘best-fitting’ ellipsoid parameters can be handled separately, outside the mainstream of DOD geospatial information generation.

### **Flattening**

The flattening (f) is now one of the defining parameters for WGS 84. Its adopted value is:

$$1/f = 298.257223563$$

The flattening of the WGS 84 Ellipsoid is numerically distinct from the GRS 80 flattening.

### **Earth’s Gravitational Constant (GM)**

The central term in the Earth’s gravitational field (GM) is known with much greater accuracy than either ‘G’, the universal gravitational constant, or ‘M’, the mass of the Earth. Significant improvement in the knowledge of GM has occurred since the

original WGS 84 development effort. The refined value of the WGS 84 GM parameter, along with its 1  $\sigma$  uncertainty is:

$$GM = (3986004.418 \pm 0.008) \times 10^8 \text{ m}^2/\text{s}^2$$

### **Angular Velocity of the Earth ( $\omega$ )**

The value of  $\omega$  used as one of the defining parameters of the WGS 84 (and GRS 80) is:

$$\omega = 7292115 \times 10^{-11} \text{ radians/second}$$

This value represents a standard Earth rotating with a constant angular velocity. Note that the actual angular velocity of the Earth fluctuates with time. [3]

### **THE EARTH ACCORDING TO WGS 84**

Appendix C lists the lengths in feet of 1 minute, 1 second and 1/1000 minute of latitude and longitude according to WGS84. Although, latitude intervals are not much different between poles and the equator, longitude intervals are zero at the poles, and it has the biggest value at equator.

However, NSTMSS old coordinate system takes the average minute of latitude and longitude 1852 and this is not an efficient method.

Figure 2.14 and 2.15 shows how the length of one minute of latitude increases from 1842.905 m at the equator (0°) to 1861.566 m at the poles (90°) (increase = 18.662 m) and how the length of one minute of longitude decreases from 1855.325 m at the equator (0°) to 0 m at the poles (90°)

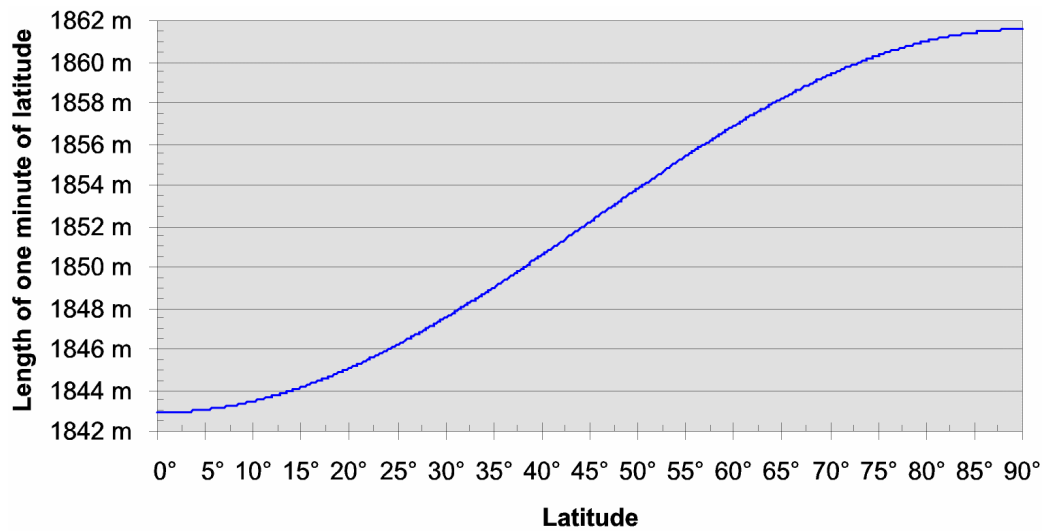


Figure 2.14 Increasing of latitude from equator to poles [18]

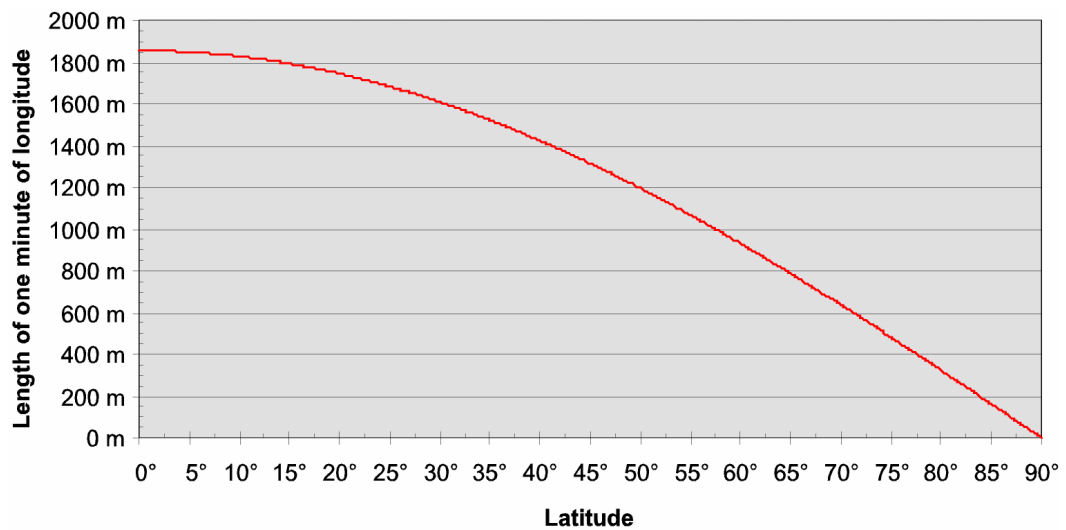


Figure 2.15 Decreasing of longitude from equator to poles [18]

### REALIZATION OF THE WGS 84

The origin and the orientation of coordinate axes in WGS 84 are defined by the X, Y, Z coordinates of the five GPS monitoring stations. [4]

Besides being a map/chart datum WGS 84 also defines the shape and size of the ellipsoid of revolution (an oblate spheroid) that is considered to be the best

representation of the Earth. From these two numbers it is possible to calculate the Earth parameters (see Appendix A)

World Geodetic System 1984, amongst other things, is s the currently accepted 'best fit' for the overall shape of the Earth.

## **2.2 LOCAL COORDINATE SYSTEM**

The second coordinate system in this work is local coordinate system in the other words map coordinate system. As explained in chapter 1, UTM projection method was used and also UTM map coordinates were used for local coordinate system.

The ship moves in UTM coordinates then it is converted to latitude, longitude notation by using the formulas which is given in chapter 3.

Different maps in Multigen OpenFlight (.flt) formats can be loaded to system.

A new map was added named sinop.flt which shows an area of Sinop Harbour (figure 2.16). The origin of the local coordinate system was decided by using MultiGen Creator. The origin was chosen near the land (origin of the red plus sign on the figure, 42 00 00 N, 34.5 00 00E ). The local coordinate system axes are showed at the right-top corner of the figure. The area consists of small polygons represents the sea and the area consists of bigger polygons represent the land.



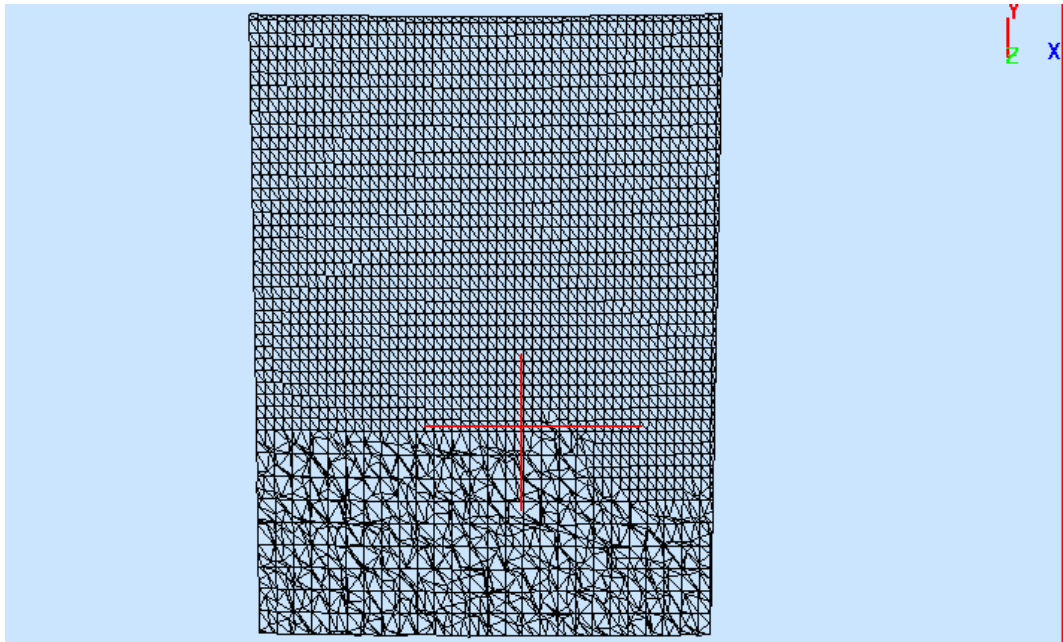


Figure 2.16 sinop.flt and map origin

## 2.3 SHIP BODY COORDINATE SYSTEM

The aim of the ship body coordinate system is to be able to define every point of the ship relative to ship coordinate system, relative to map local coordinate system (refer to section 2.2 in this chapter) and relative to world coordinate system (refer to section 2.1 in this chapter). When a point on the ship is given, it is defined in three different typed of coordinate system;

### 2.3.1 REPRESENTING ORIENTATIONS

There is no a simple means of representing a 3D orientation. There are twol popular options to represent orientations:

- Euler angles
- Quaternions

## EULER'S THEOREM

Leonard Euler (1707-1783)

Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis. This is the Euler's Theorem [36].

### Euler Angles

This means that an orientation can be represented with 3 numbers. A sequence of rotations around principle axes is called an Euler Angle Sequence. Assuming rotations are limited to 3 without successive rotations about the same axis, any of the following 12 sequences can be used:

XYZ XZY XYX XZX  
YXZ YZX YXY YZY  
ZXY ZYX ZXZ ZYZ

This gives 12 alternative ways to store an orientation using Euler angles. Different industries use different conventions for handling Euler angles (or no conventions) [36].

### Euler Angles to Matrix Conversion

To build a matrix from a set of Euler angles, a sequence of rotation matrices are multiplied together [36]:

$$\begin{aligned} R_x \cdot R_y \cdot R_z &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_x & s_x \\ 0 & -s_x & c_x \end{pmatrix} \begin{pmatrix} c_y & 0 & -s_y \\ 0 & 1 & 0 \\ s_y & 0 & c_y \end{pmatrix} \begin{pmatrix} c_z & s_z & 0 \\ -s_z & c_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} c_y c_z & c_y s_z & -s_y \\ s_x s_y c_z - c_x s_z & s_x s_y s_z + c_x c_z & s_x c_y \\ c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & c_x c_y \end{pmatrix} \end{aligned}$$

## Euler Rotation

### Euler Angle Order

As matrix multiplication is not commutative, the order of operations is important.

To use Euler angles, one must choose which of the 12 representations they want. There may be some practical differences between them, and the best sequence may depend on what exactly is accomplished.[36]

In this work z y x notation is used. Since there is only one Euler angle convention corresponds to naval architecture conventions. [29]. Figure 2.17 shows zyx notation.

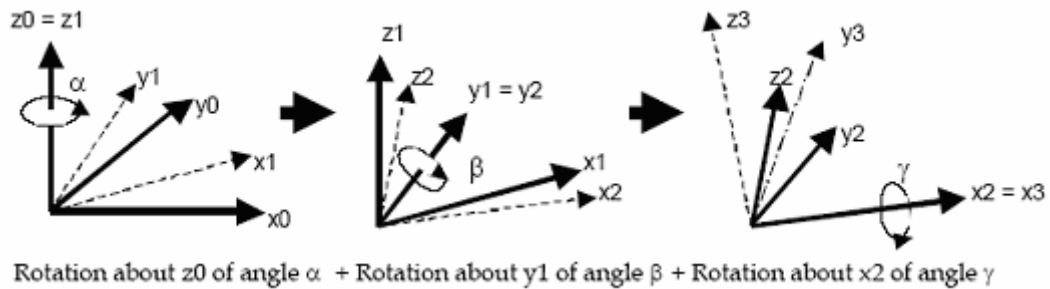


Figure 2.17 zyx notation of Euler Rotation [37]

The matrix representation of this rotation is [37]

$$T_{0,3} = T_{0,1}T_{1,2}T_{2,3} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) \\ \sin(\alpha)\sin(\beta) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) \\ -\sin(\beta) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{pmatrix}$$

## DEFINING A POINT ON THE SHIP IN BODY COORDINATE SYSTEM

To define a point on the ship model, Cartesian coordinate system (expressed in chapter 2) was used. Figure 2.18 shows a view of the ship model in MultiGen tool. Cartesian coordinates (x, y, z) are an easy and natural means of representing a position in 3D space

The axis orientation of the Cartesian coordinate system is showed at the right corner of the Figure 2.18. The x-axis is up-down direction, the y-axis is left-right direction and the z-axis is toward in the screen. The origin is the cross of two red lines.

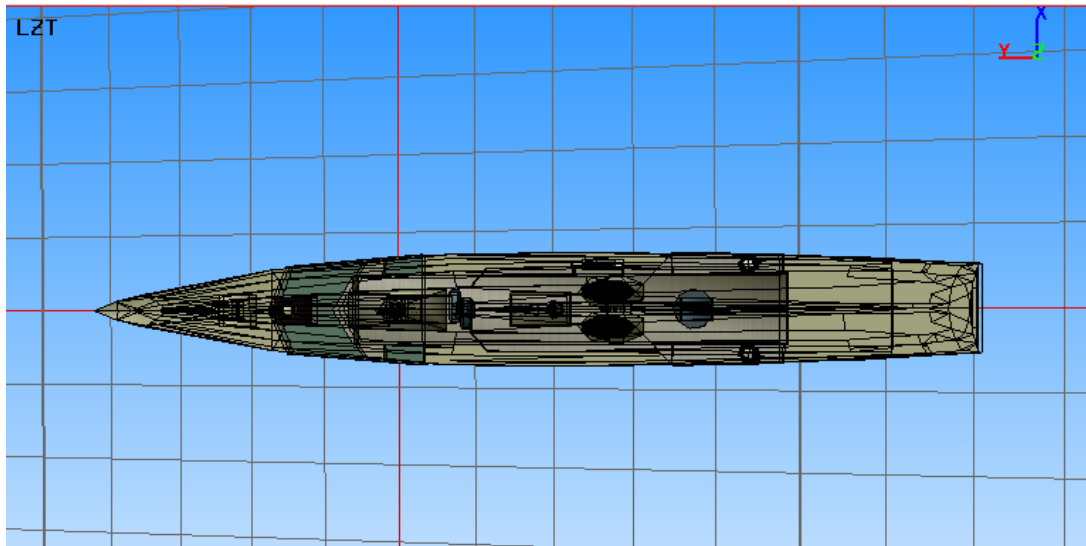


Figure 2.18 The ship model of NSTMSS (Meko) [11]

For orientations, Euler rotation was used as explained in “Representing Orientations“. Yaw, pitch, roll angles are the Euler angles here. For the rotation of body Cartesian axes, rotation matrix multiplied with the x, y, z Cartesian coordinates as the ship moves.

*Rotated point on the ship = point matrix \* rotation matrix*

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Before defining the point on the ship in local and world coordinate system, it is first rotated according to Euler rotation method.

### **DEFINING A POINT ON THE SHIP IN LOCAL COORDINATE SYSTEM**

After defining a point relative to ship body coordinate system as x, y, z, it is also defined in local coordinate system. This x, y, z coordinate components are added to ship origin x, y, z components in the map local coordinate system. They are all metrics, so they are in the same unit, and it is possible to add them. Local coordinates take the origin as the map origin. This is different than UTM. Figure 2.19 shows the origin and a point which of local coordinates are being calculated in the example.

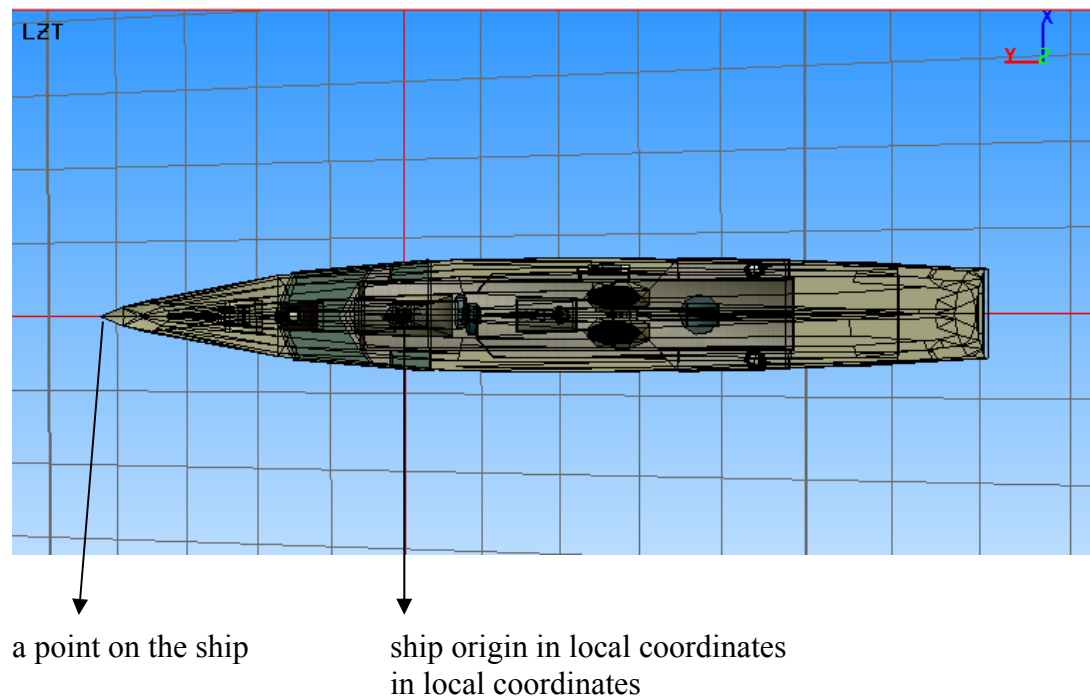


Figure 2.19 Defining a point on the ship in local coordinate system

*a point on the ship in local coordinate system = ship origin in local coord. system + point in body coord. System*

## **DEFINING A POINT ON THE SHIP IN WORLD COORDINATE SYSTEM**

After defining a point relative to local coordinate system as x, y, z, it is also defined in world coordinate system. This x, y, z coordinate components are added to ship origin x, y, z components in the world coordinate system. They are all metrics, so they are in the same unit, and it is possible to add them. Figure 2.20 shows the origin and a point which of world coordinates are being calculated in the example.

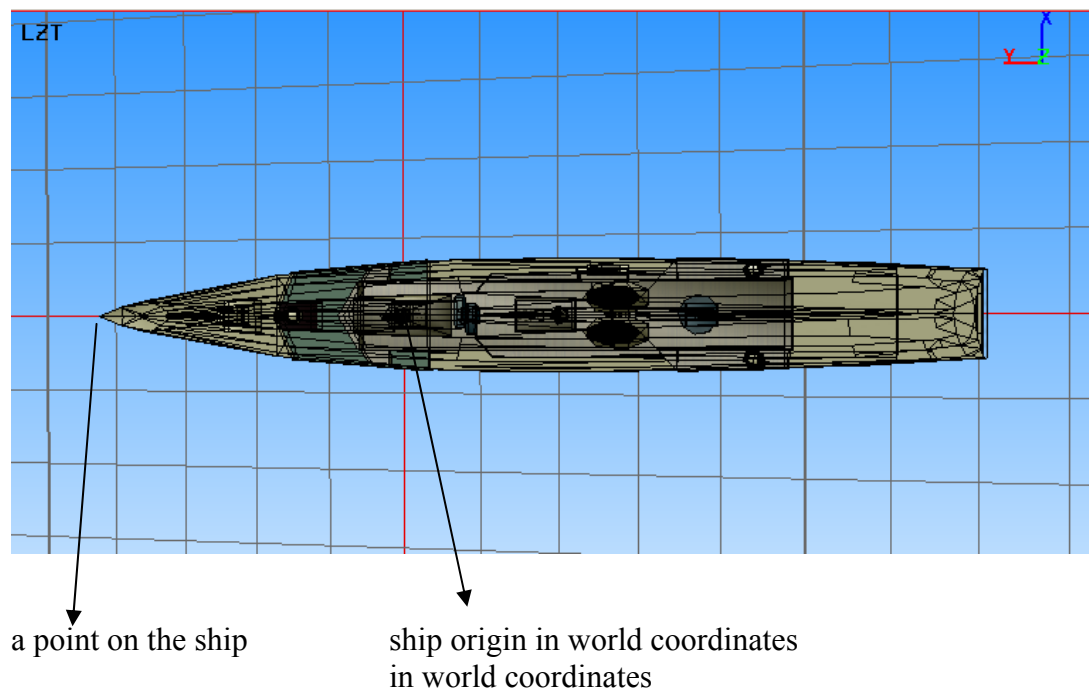


Figure 2.20 Defining a point on the ship in world coordinate system

*a point on the ship in world coordinate system = ship origin in world coord. system + point in body coord. System*

After this calculation, the result is in UTM coordinates, and it is converted to latitude / longitude notation using the formulas expressed in chapter 3.

## **2.4 RELATED WORK**

Up to this section, some technical background information about coordinate systems were introduced. In this section selected related works about world and body coordinate systems and their relation with this thesis are discussed.

There are similar solutions with this thesis solution in academic and commercial applications of coordinate systems for naval virtual environments. Selected ones are explained below.

The first related work is NPSNET. NPSNET is a Java-based component architecture for networked virtual environment applications. It provides a persistent shared virtual world.[26] It has been developed by The MOVES Institute. NPSNET uses two right handed coordinate systems, one for the world and the second for the body coordinates. As shown in the figure 2.21, the world coordinate system has (0,0,0) located at Mean Sea Level (MSL) at the upper left hand, or northwest, corner of the database. The positive X axis is heading east, positive Y is the altitude above sea level, and Z increases to the south. However, in this thesis a right handed coordinate system has been preferred. thesis In order to maintain constancy with most military systems, all distances are measured in meters. One unit of UTM equals one meter. Internally, all angular measurements obey the right hand rule. However, when information is presented to the user, or external interfaces, the intuitive and / or standard units of measurement and reference points are used [27].

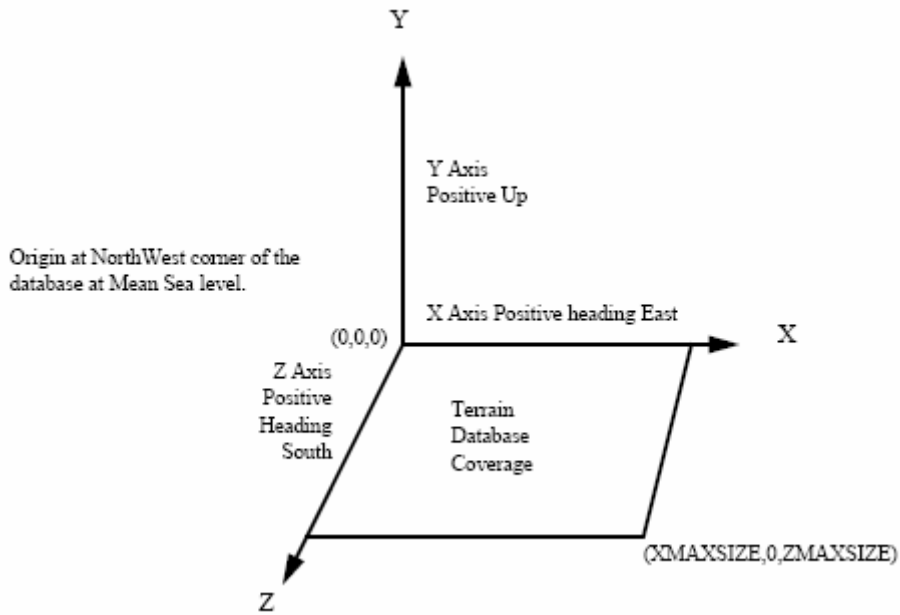


Figure 2.21 NPSNET World Coordinate System [27]

NPSNET presents world coordinates to the user in UTM in this thesis. In order to maintain constancy with most military systems, all distances are measured in meters. One unit of UTM equals one meter. Internally, all angular measurements obey the right hand rule. However, when information is presented to the user, or external interfaces, the intuitive and / or standard units of measurement and reference points are used [27]. Figure 2.22 shows the body coordinate system of NPSNET. The axes are different from this thesis work for project special aims. In this thesis, orientation Y and Z axes is vice versa because z axis present the height in this thesis.



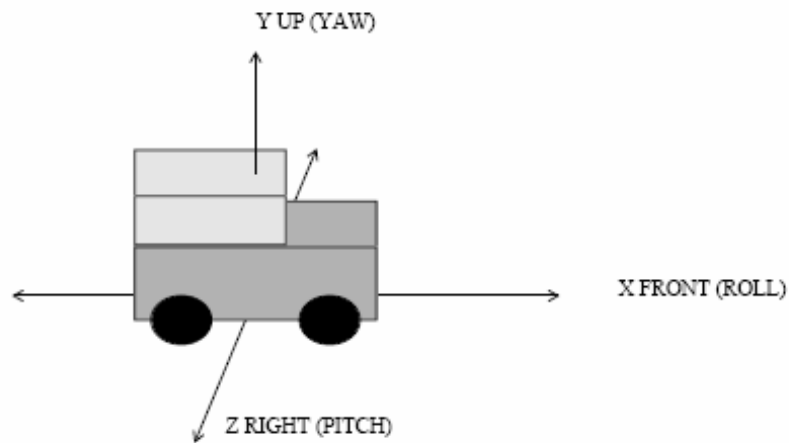


Figure 2.22 NPSNET Body Coordinate System [27]

The second related work is a thesis about rigid body dynamics, “*Rigid Body Dynamics, Inertial Reference Frames, and Graphics Coordinate Systems: A Resolution of Conflicting Conventions and Terminology*” [28]. This thesis examines commonalities and differences between the choice of coordinate systems and methods used for representing rigid body motion. It says that computer simulation of rigid body dynamics must in the end reduce to scalar calculations, since this is the inherent nature of computer arithmetic. To permit this, and yet retain a level of abstraction suitable for human reasoning, it is conventional for computer simulation of rigid body motion to choose a coordinate system attached to an appropriate inertial frame, and then to express all vectors in component form relative to these coordinates. To specify orientation, it is also necessary, for each rigid body, to specify a “body fixed” coordinate system attached to the body. This is also an xyz system [28]. So, in this thesis, cartesian coordinate system has been chosen and this “body fixed” coordinate system has been attached to the rigid body (ship dynamic model) to specify orientation of each point on the ship.

Also it explains how Euler and Quaternions are the solution for representation of rigid body orientation and how Quaternion representation of rigid body orientation eliminates the singularity problem. Finally, it argues that the absolute values of elevation must be greater than 90 degrees when using a set of Euler angles [28].

Realistically, it is not a problem using Euler Angle with the ships of this work because we never have pitches greater than 90 degree.

The third related work is again a thesis about a virtual world for underwater vehicle dynamics model, “Virtual World for An Autonomous UnderWater Vehicle”. Euler angle order in our thesis is the same as Euler angle order in this thesis. This Euler angle order is consistent with naval architecture definitions [30].

It defines body coordinate system of its underwater vehicle dynamics and defines world coordinate system of its virtual world. The world coordinate system of this model is defined by three orthogonal axes originating at an arbitrary local point at the ocean surface. Body coordinates are defined with respect to the body of the vehicle of interest. The three axes of a vehicle are longitudinal pointing in the nominal forward direction of the vehicle, lateral pointing through the right hand side of the level vehicle, and (unlike NPSNET) downward through the nominal bottom of the vehicle. The origin of body coordinates for a submerged vehicle is at the half point along the symmetric longitudinal axis. Typically this point is at or near the center of buoyancy (CB), which is the centroid of volumetric displacement of the submerged vehicle. A related location is the center of gravity (CG), which is the first moment centroid of vehicle mass. Body axes are referred to as longitudinal, lateral and vertical, corresponding to (x, y, z) body coordinates. case. When converting from world to body coordinates using physical order (as might be specified in a three-axis gimbal system), the first rotation is for yaw ( $\psi$ ) about the z-axis, then pitch ( $\theta$ ) about the first intermediate y-axis, then roll ( $\phi$ ) about the second intermediate x-axis. Naturally the orders of rotations are reversed if converting from body to world coordinate frame. These Euler angle definitions are consistent with naval architecture definitions [30]. This is an important property since while twelve different and unique Euler angle coordinate system definitions are possible [31], only one Euler angle convention corresponds to naval architecture conventions. [29].

Another related commercial work is JWARS (Joint Warfare System ). It is a warfare simulation project and supports joint analysis which has been developed

since 1995. Its ellipsoid and coordinate systems are same as selected datum and coordinate systems in this thesis.

## **JWARS**

**represents the entire globe with**

- WGS 84 ellipsoid
- Global Coordinate System

**coordinate system**

- Uses the real-earth shape
- Hides the complexity of a curved earth from most modeling activities
- Presents an intuitive interface based on our “locally apparent flat-earth experience”

**encapsulates several coordinate system views:**

- Latitude/Longitude
- Geocentric (by transform)
- Geodetic
- UTM (by transform) [32]

# CHAPTER 3

## MAP PROJECTION METHODS AND DIGITAL TERRAINS

In this chapter, general technical information about map projection methods and the used map projection methods in this work is presented. Also, this chapter introduces related works about these issues. Then, brief information is given about digital terrains. Finally, it is explained that which methods was used in this work in implementation for which reason.

Technical information presented in this chapter extracted from the following sources: [3], [4], [5], [9], [10], [11], [13], [14], [16], [17], [19], [21], [23], [33].

### 3.1 MAP PROJECTION METHODS

#### DEFINITION

Briefly, map projection is the transformation of a curved earth to a flat map (see Figure 3.1).

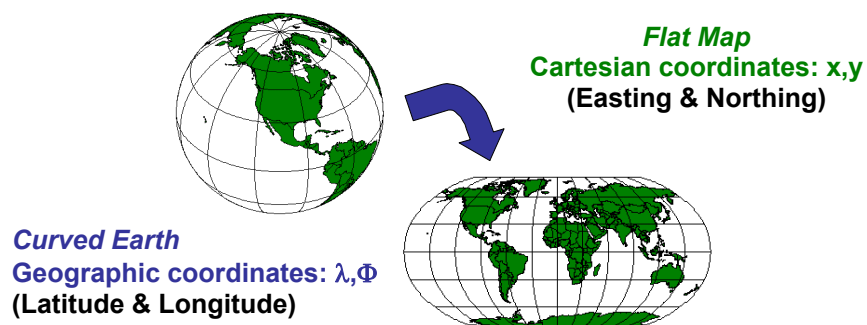


Figure 3.1 Map projection [16]

The advances in information technology during the last two decades have given a boost to automatic cartography and hence, to digital mapping. A hard copy analogue map can now be digitized and transformed into a computer compatible data base, which can then be used for a variety of Computer Aided Design (CAD) applications in planning, civil engineering and Geographical Information Systems (GIS).

The general principle of map projection is as follows:

It is necessary to determine the functions  $f_1$  and  $f_2$  which map the ellipsoidal (or spherical) coordinates  $\phi, \lambda$  onto a plane with rectangular coordinates  $x, y$ .

$$x = f_1(\phi, \lambda)$$

$$y = f_2(\phi, \lambda)$$

$f_1$  and  $f_2$  can be a function of latitude and longitude, or both. Each map projection has unique equations for  $x$  and  $y$ . In other words, there is a one-to-one correspondence between the earth and the map.

### **3.1.1 TYPES OF PROJECTIONS**

Projections can be classified as follows:

Mapping of the earth onto the plane of a

- Azimuthal plane (Lambert Azimuthal Equal Area).  
Used for small scale products with large East-West Expanses (see Figure 3.2).
- Tangent cone (Albers Equal Area, Lambert Conformal Conic).  
Good for East-West land areas (see Figure 3.3).
- Tangent cylinder (Transverse Mercator) (see Figure 3.4).

Plane, cone and cylinder can be in a normal, transverse or oblique position attached to the earth. In addition, the surfaces of the plane, cone and cylinder can intersect the ellipsoid (or sphere), so that there are two lines of contact. These projections are called secant projections. [4]

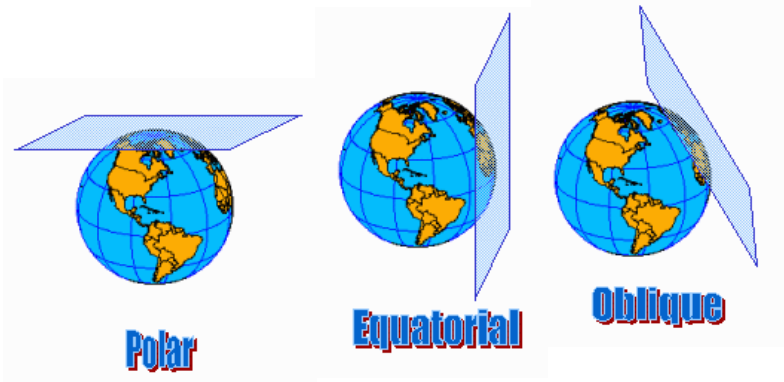


Figure 3.2 Azimuthal Projection [16]

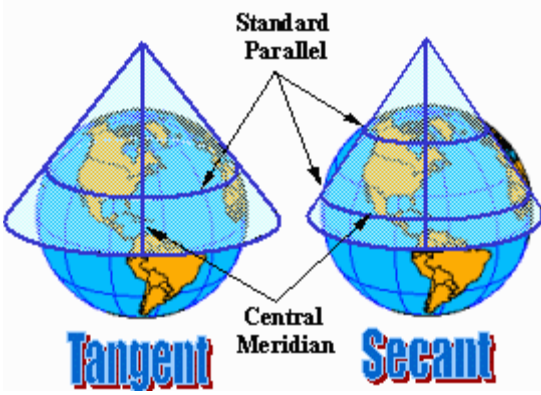


Figure 3.3 Conic projection [16]

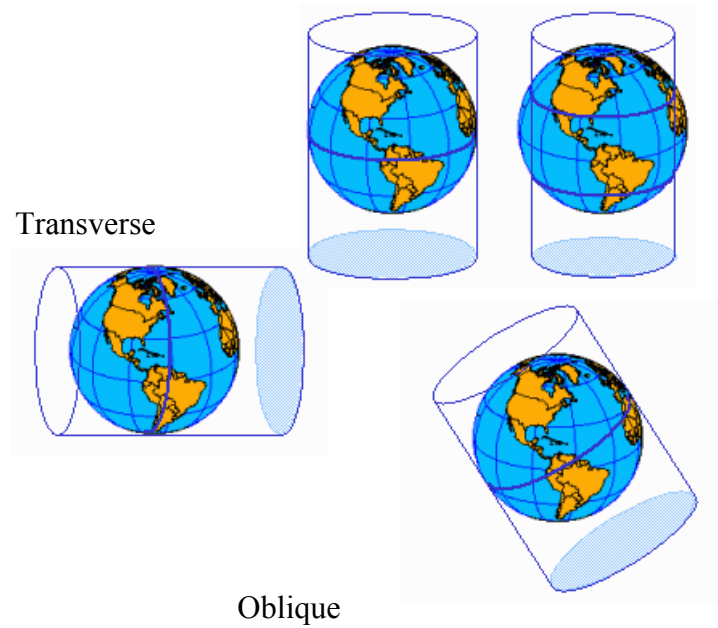


Figure 3.4 Cylindrical Projection [16]

A particular projection is defined by

- a datum
- a projection type
- a set of projection parameters[16]

Among above three map projection methods the idea of the transverse mercator projection has its roots in the 18th century. It has become the most used because it allows precise measurements in meters to within 1 meter. The same type of projection is used in a worldwide mapping standard known as Universal Transverse Mercator (UTM).

The section below is the definition of UTM which is the map projection method adopted for this work.

### **The Universal Transverse Mercator System**

A mercator projection is a ‘pseudocylindrical’ conformal projection (it preserves shape). What poster-size maps of the world is an equatorial mercator projection that

has relatively little distortion along the equator, but quite a bit of distortion toward the poles.

What a transverse mercator projection does, in effect, is orient the ‘equator’ north-south (through the poles), thus providing a north-south oriented swath of little distortion. By changing slightly the orientation of the cylinder onto which the map is projected, successive swaths of relatively undistorted regions can be created.

This is exactly what the UTM system does. Each of these swaths is called a UTM zone and is six degrees of longitude wide (see Figure 3.5). The first zone begins at the International Date Line (180°, using the geographic coordinate system). The zones are numbered from west to east, so zone 2 begins at 174°W and extends to 168°W. The last zone (zone 60) begins at 174°E and extends to the International Date Line. [13]

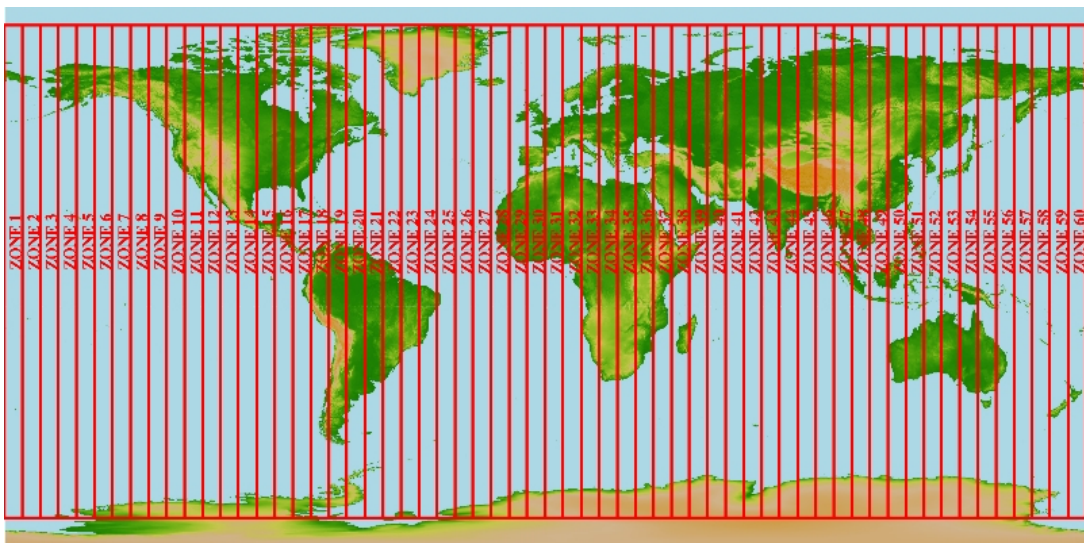


Figure 3.5 The zones used in the UTM spatial coordinate system [14]

The zones are then further subdivided into an eastern and western half by drawing a line, representing a transverse mercator projection, down the middle of the zone. This line is known as the ‘central meridian’ and is the only line within the zone that



can be drawn between the poles and is perpendicular to the equator (in other words, it is the new ‘equator’ for the projection and suffers the least amount of distortion). For this reason, vertical grid lines in the UTM system are oriented parallel to the central meridian. The central meridian is also used in setting up the origin for the grid system (see Figure 3.6). [13]

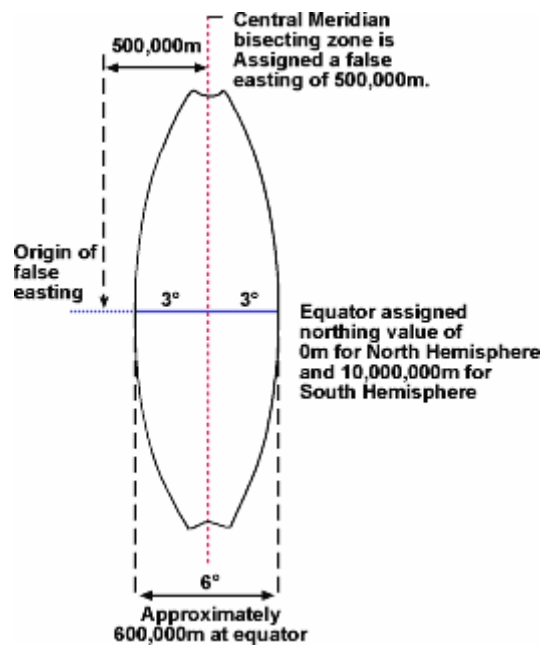


Figure 3.6 The central meridian of an UTM zone [13]

Any point can then be described by its distance east of the origin (its ‘easting’ value). By definition the Central Meridian is assigned a false easting of 500,000 meters. Any easting value greater than 500,000 meters indicates a point east of the central meridian. Any easting value less than 500,000 meters indicates a point west of the central meridian. Distances (and locations) in the UTM system are measured in meters, and each UTM zone has its own origin for east-west measurements.

To eliminate the necessity for using negative numbers to describe a location, the east-west origin is placed 500,000 meters west of the central meridian. This is

referred to as the zone's 'false origin'. The zone does not extend all the way to the false origin.

Figures 3.7 and 3.8 show origins for southern and northern hemisphere. The origin for north-south values depends on whether one is in the northern or southern hemisphere. In the northern hemisphere, the origin is the equator, and all distances north (or 'northings') are measured from the equator. In the southern hemisphere the origin is the south pole, and all northings are measured from there. Once again, having separate origins for the northern and southern hemispheres eliminates the need for any negative values. The average circumference of the earth is 40,030,173 meters, meaning that there are 10,007,543 meters of northing in each hemisphere. [13]

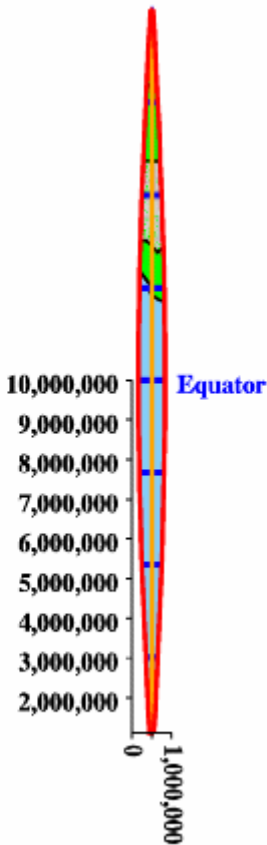


Figure 3.7 UTM zone 13 coordinates for the southern hemisphere. [14]

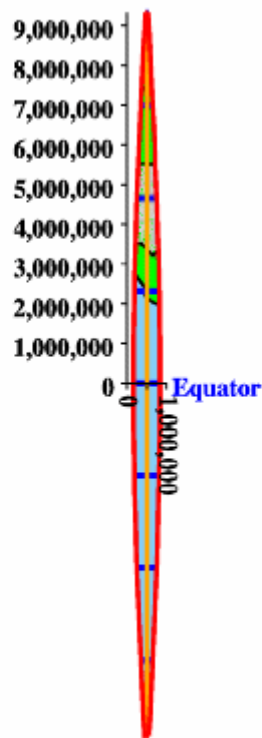


Figure 3.8 UTM zone 13 coordinates for the northern hemisphere. [14]

UTM coordinates are typically given with the zone first, then the easting, then the northing. So, in UTM coordinates, Red Hill is located in zone twelve at 328204 E (easting), 4746040 N (northing). Based on this, the position is west of the central meridian in zone twelve and just under halfway between the equator and the north pole. UTM is an extremely fast and efficient means of finding exact locations and approximating locations on a map. [13]

### Properties

- Shape - UTM is conformal – shapes are preserved in small areas.
- Area - minimal distortion of larger shapes occurs within the same zone.
- Distance – local angles are true
- Direction - Scale is constant along the central meridian, but at a scale factor of 0.9996 to reduce lateral distortion within each zone.

## Limitations

- Designed for a scale error not exceeding 0.1 percent within each zone.
- Error and distortion increase for regions that span more than one UTM zone.[5]

### 3.1.2 CONVERSION FORMULAS

In geodesy, map coordinates tend only to be used for visual display purposes. When it is needed to do computations with coordinates, latitude and longitude or Cartesian coordinates are used, then convert the results to map coordinates as a final step if needed. This working procedure is in contrast to the practice in geographical information systems, where map coordinates are directly used mostly for many computational tasks.

For below two formulas, firstly datum being used must be known. The datum is WGS84 for this work. The parameters for this datum are given in chapter 2. Conversion formulas below were taken from [10].

#### CONVERTING LATITUDE AND LONGITUDE TO UTM

##### Symbols

$lat$  = latitude of point

$long$  = longitude of point

$long_0$  = central meridian of zone

$k_0$  = scale along  $long_0 = 0.9996$

$e = \text{SQRT}(1 - b^2/a^2) = .08$  approximately. This is the eccentricity of the earth's elliptical cross-section.

$e'^2 = (ea/b)^2 = e^2 / (1 - e^2) = .007$  approximately. The quantity  $e'$  only occurs in even powers, so it is need only be calculated as  $e'^2$ .

$$n = (a-b)/(a+b)$$

$$\rho = a(1 - e^2) / (1 - e^2 \sin^2(\text{lat}))^{3/2}$$

This is the radius of curvature of the earth in the meridian plane.

$\nu = a / (1 - e^2 \sin^2(\text{lat}))^{1/2}$  This is the radius of curvature of the earth perpendicular to the meridian plane. It is also the distance from the point in question to the polar axis, measured perpendicular to the earth's surface.

$$p = (\text{long} - \text{long}_0)$$

$$\sin 1'' = \text{sine of one second of arc} = \pi / (180 * 60 * 60) = 4.8481368 \times 10^{-6}.$$

### Calculate the Meridional Arc

S is the meridional arc through the point in question (the distance along the earth's surface from the equator). All angles are in radians.

$S = A' \text{lat} - B' \sin(2\text{lat}) + C' \sin(4\text{lat}) - D' \sin(6\text{lat}) + E' \sin(8\text{lat})$ , where lat is in radians and

$$A' = a[1 - n + (5/4)(n^2 - n^3) + (81/64)(n^4 - n^5) \dots]$$

$$B' = (3an/2)[1 - n + (7/8)(n^2 - n^3) + (55/64)(n^4 - n^5) \dots]$$

$$C' = (15an^2/16)[1 - n + (3/4)(n^2 - n^3) \dots]$$

$$D' = (35an^3/48)[1 - n + (11/16)(n^2 - n^3) \dots]$$

$$E' = (315an^4/51)[1 - n \dots]$$

The United States Geological Survey (USGS) gives this form, which may be more appealing to some. (They use M where the Army uses S)

$$M = a[(1 - e^2/4 - 3e^4/64 - 5e^6/256 \dots)lat - (3e^2/8 + 3e^4/32 + 45e^6/1024 \dots)\sin(2lat) + (15e^4/256 + 45e^6/1024 + \dots)\sin(4lat) - (35e^6/3072 + \dots)\sin(6lat) + \dots]$$

where lat is in radians

### Converting Latitude and Longitude to UTM

All angles are in radians.

$$y = \text{northing} = K1 + K2 p^2 + K3 p^4, \text{ where}$$

$$K1 = S k_0,$$

$$K2 = k_0 \sin^2 1'' v \sin(lat) \cos(lat) / 2$$

$$K3 = [k_0 \sin^4 1'' v \sin(lat) \cos^3(lat) / 24] [(5 - \tan^2(lat) + 9e'^2 \cos^2(lat) + 4e'^4 \cos^4(lat))]$$

$$x = \text{easting} = K4 p + K5 p^3, \text{ where}$$

$$K4 = k_0 \sin 1'' v \cos(lat)$$

$$K5 = (k_0 \sin^3 1'' v \cos^3(lat) / 6) [1 - \tan^2(lat) + e'^2 \cos^2(lat)]$$

Easting  $x$  is relative to the central meridian. For conventional UTM easting add 500,000 meters to  $x$ .

### Converting UTM to Latitude and Longitude

$y = \text{northing}$ ,  $x = \text{easting}$  (relative to central meridian; subtract 500,000 from conventional UTM coordinate).

### Calculate the Meridional Arc

This is easy:  $M = y/k_0$ .

### Calculate Footprint Latitude

$$\mu = M/[a(1 - e^2/4 - 3e^4/64 - 5e^6/256...)]$$

$$e_i = [1 - (1 - e^2)^{1/2}] / [1 + (1 - e^2)^{1/2}]$$

footprint latitude  $fp = \mu + J1\sin(2\mu) + J2\sin(4\mu) + J3\sin(6\mu) + J4\sin(8\mu)$ , where:

$$J1 = (3e_i/2 - 27e_i^3/32 ..)$$

$$J2 = (21e_i^2/16 - 55e_i^4/32 ..)$$

$$J3 = (151e_i^3/96 ..)$$

$$J4 = (1097e_i^4/512 ..)$$

### Calculate Latitude and Longitude

$$e'^2 = (ea/b)^2 = e^2 / (1 - e^2)$$

$$C1 = e'^2 \cos^2(fp)$$

$$T1 = \tan^2(fp)$$

$$R1 = a(1 - e^2) / (1 - e^2 \sin^2(fp))^{3/2}$$

This is the same as rho in the forward conversion formulas above, but calculated for fp instead of lat.

$$N1 = a / (1 - e^2 \sin^2(lat))^{1/2}$$

This is the same as nu in the forward conversion formulas above, but calculated for fp instead of lat.

$$D = x / (N1 k_0)$$

$lat = fp - Q1(Q2 - Q3 + Q4)$ , where:

$$Q1 = N1 \tan(fp) / R1, Q2 = (D^2 / 2)$$

$$Q3 = (5 + 3TI + 10CI - 4CI^2 - 9e'^2) D^4 / 24$$

$$Q4 = (61 + 90TI + 298CI + 45TI^2 - 3CI^2 - 252e'^2) D^6 / 720$$

$long = long_0 + (Q5 - Q6 + Q7) / \cos(fp)$ , where:

$$Q5 = D$$

$$Q6 = (1 + 2TI + CI) D^3 / 6$$

$$Q7 = (5 - 2CI + 28TI - 3CI^2 + 8e'^2 + 24TI^2) D^5 / 120$$

## Precision

Latitude and longitude second is calculated with 3 decimal precision (.000)

## CONVERTING BETWEEN DECIMAL DEGREES, DEGREES, MINUTES AND SECONDS

Another formula that was used in this work is the formula for converting between decimal degrees, degrees, minutes and seconds, and Radians. To show the latitude and longitudes more user friendly, they are converted to “degree minute second” notation in this work.

### **(dd + mm/60 + ss/3600) to Decimal degrees (dd.ff)**

*dd* in whole degrees, *mm* in minutes, *ss* in seconds

$$dd.ff = dd + mm/60 + ss/3600$$

### **Decimal degrees (dd.ff) to (dd + mm/60 + ss/3600)**

For the reverse conversion, dd.ff is wanted to convert to dd mm ss. Here ff = the fractional part of a decimal degree.

$$mm = 60 * ff$$

$$ss = 60 * (\text{fractional part of } mm)$$



## **3.2 DIGITAL TERRAINS**

This section, three digital elevation data format used in this work namely, the Digital Terrain Elevation Data (DTED) and Digital Elevation Data (DED), are explained as well as the OpenFlight (FLT) format. The first is NIMA's and the others are MultiGen database formats. All of these databases can be created from each other as this DTED->DED->FLT to create the project scene.

### **3.2.1 DIGITAL TERRAIN ELEVATION DATA (DTED)**

The DTED format was developed for the distribution of gridded elevation data that was assembled by the US Defense Mapping Agency (DMA), now the National Imagery and Mapping Agency (NIMA). Figure 3.9 presents an elevation grid in DTED format. The one on the left is displayed as shaded-relief and the one on the right is displayed as an elevation grid.

DTED data is available at three different resolutions, which are referred to as Level 0, Level 1 and Level 2. Each DTED file contains elevation data for 1 degree latitude by 1 degree longitude cell. Elevations are stored as signed 2-byte integers with a byte order of MSB (most significant byte first) and units of meters. For latitudes between -50 and 50, pixel dimensions are given by:

DTED Level 0: 30 arcseconds (121 columns x 121 rows)

DTED Level 1: 3 arcseconds (1201 columns x 1201 rows)

DTED Level 2: 1 arcsecond (3601 columns x 3601 rows)

Latitudes closer to the poles are divided into 4 zones, and the x-size of pixels differs for each zone, getting larger toward the poles. [17]

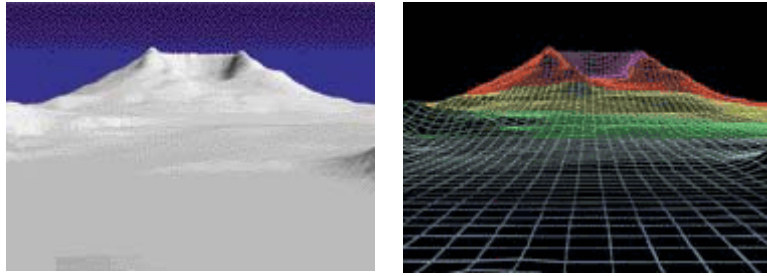


Figure 3.9 Perspective scenes of Mount Saint Helens. [23]

DTED is NIMA's primary format for distributing elevation data. Digital elevation data is also known by the general terms Digital Elevation Model (DEM), Digital Terrain Data (DTD) or Digital Terrain Model (DTM).

DTED represents a matrix or grid of measured elevation posts over a portion of the earth surface. The elevation posts are spaced at regular horizontal intervals, measured in Geographic Lat/Long units, with vertical elevation values in meters. Each row of elevation posts is sometimes termed a profile. DTED elevations are in meters, rounded off to the nearest whole meter and stored as signed 2-byte (16-bit) digital numbers. A signed 16-bit (216) number can store an integer value between -32,767 and +32,767. Since the highest mountains are in the 8,000 meter range, this can handle any possible elevation.

Although DTED is matrix (gridded) data, when DTED is used in an Image processing program or a Raster GIS program, the imported DTED tiles become raster layers and the DTED elevation posts are treated as raster grid cells.

DTED was originally developed to support flight simulators, but it is now used for a wide variety of applications and map products. An operator can create three-dimensional views of an area using DTED, plus derived products such as elevation maps, shaded relief maps, slope maps, aspect maps and contour lines. DTED is also used in photogrammetry to compensate for terrain displacement when orthorectifying imagery from satellites or aerial photography.

Datum and coordinate system used in this format is

- Horizontal datum is WGS-84.
- Vertical datum is Mean Sea Level.

Coordinate Reference System is Geographic Latitude/Longitude. [9]

### **3.2.2 DIGITAL ELEVATION DATA (DED)**

The Digital Elevation Data (DED) format is a uniform elevation map format developed by MultiGen-Paradigm that the terrain generation tools in MultiGen Creator can read. The terrain generation tools in MultiGen Creator do not read source data in its original format. Instead, they require that the original source data first be converted to DED format. It also allows combining any number of individual elevation files into a single DED file.

A DED file contains one or more cells of elevation information. These cells are visually represented in a DED Builder after importing the elevation files. Each cell corresponds to a geographic region and is composed of an elevation map and a cell header. The elevation map contains elevation values (posts) at evenly-spaced intervals within the geographic extents of a cell. The cell header contains a description of the cell, including:

- The extents (latitude and longitude points) of the geographic region covered by the cell.
- The number of elevation sample points in each dimension (longitude and latitude).
- The distance between sample points in each dimension.

The DED Builder plug-in for MultiGen Creator can build DED files which can be read by MultiGen Creator Terrain Generation Tools. Elevation data comes in many different formats; United States Geological Survey (USGS) DEM, NIMA DTED, etc. [21]

In this work, The DED Builder plug-in in MultiGen Creator has been used for conversion from DTED to DED.

To convert elevation data from DED format to DTED format in DED builder of the Multigen firstly all DED files, which are needed to generate a complete DED file, are selected as shown at left of the figure 3.10. After these selection, each DED file is appeared as a square block at the right red side. Then, the output directory is selected which the resultant DED file will be saved. Finally, DED file is created by pressing the “Generate” button. The conversion process and the resulting DED file is showed in Figure 3.10 and 3.11.

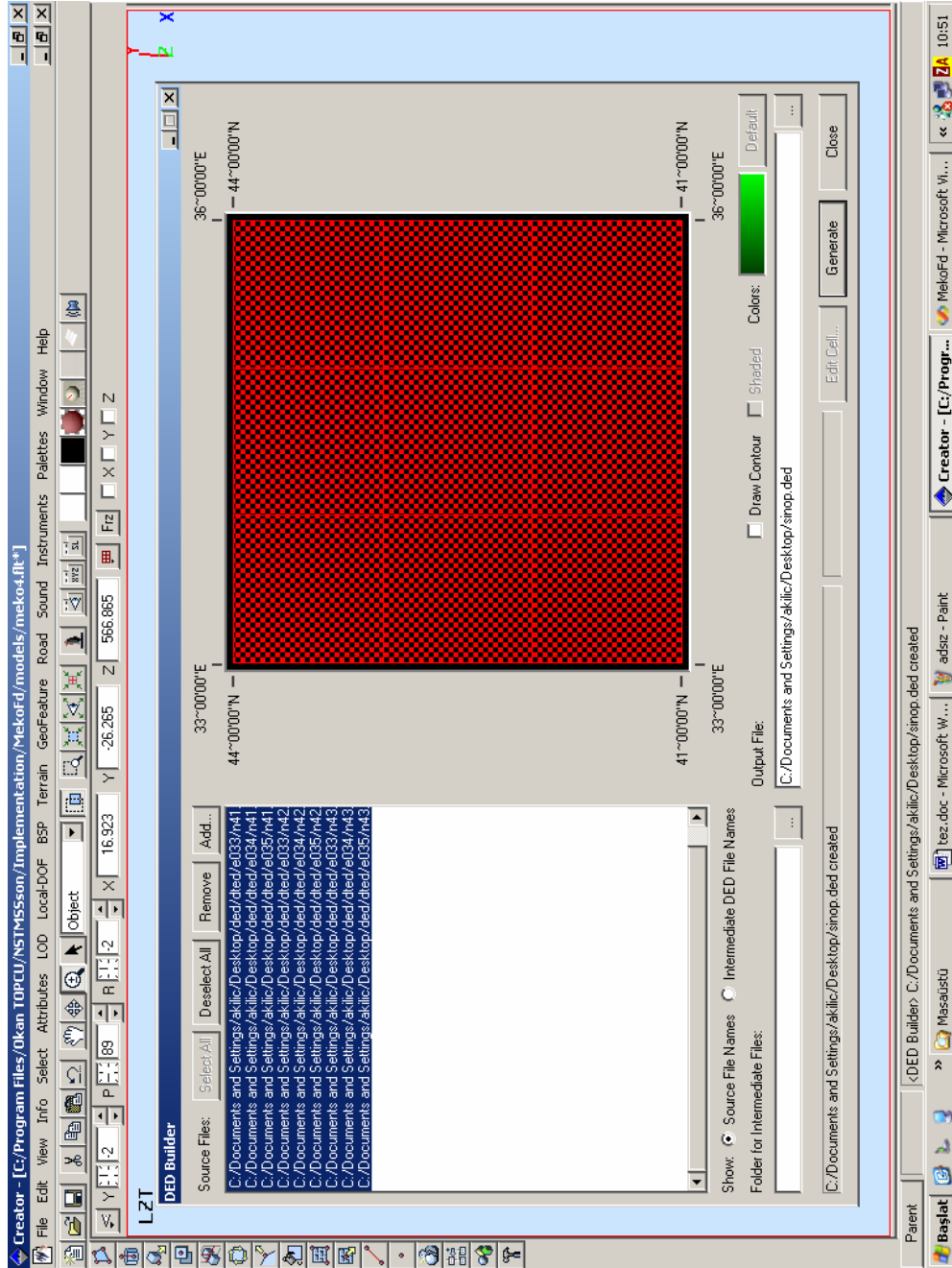


Figure 3.10 Conversion from DTED to DED with Multigen Creator

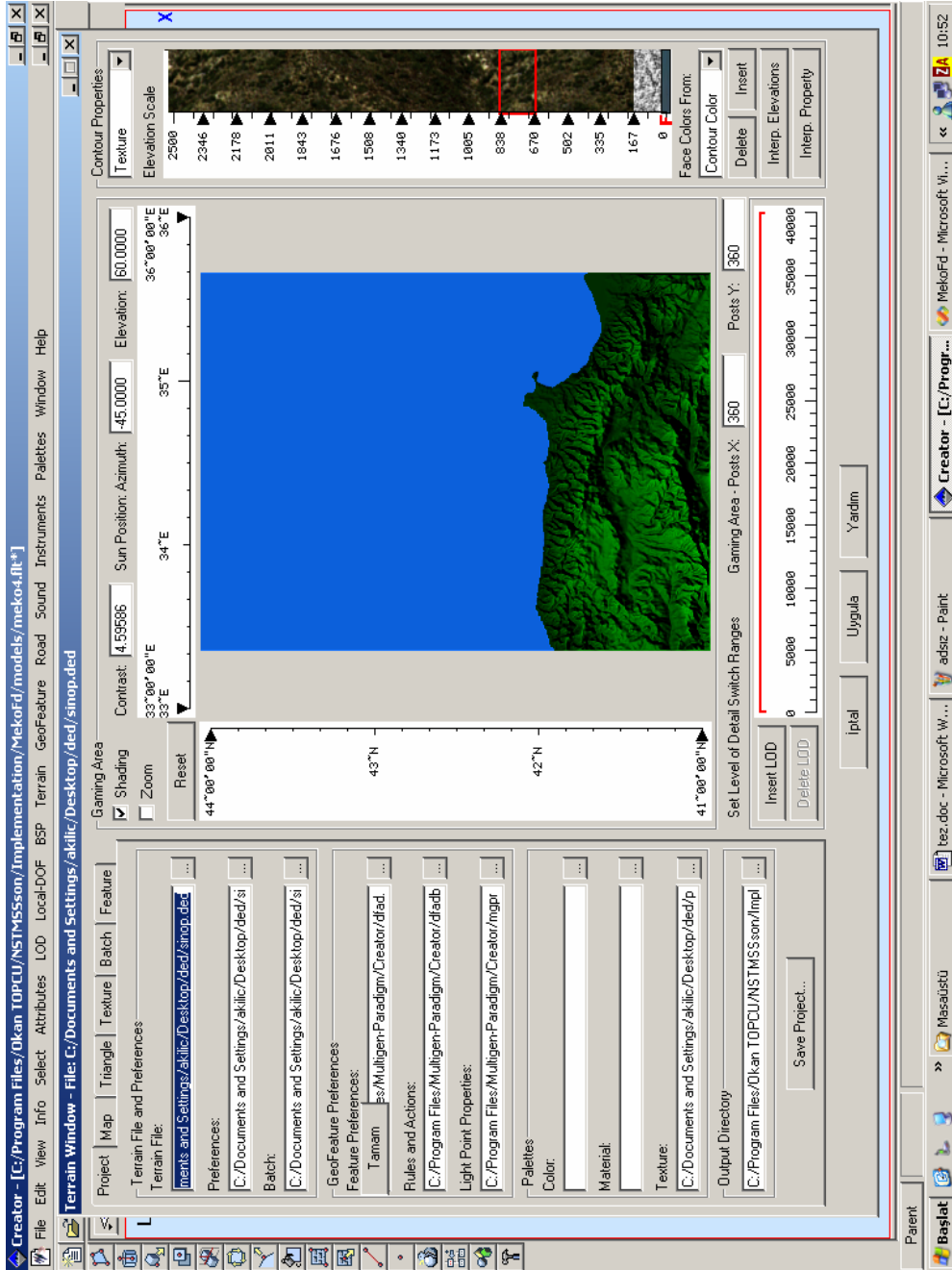


Figure 3.11 The resultumt DED file

## OPENFLIGHT FORMAT (FLT)

The OpenFlight format is a 3D and binary format from Multigen used for input and output by the MultiGen database modeling tool.

In the view of OpenGL Performer [11], a scene graph holds the data that defines a virtual world. The scene graph includes low-level descriptions of object geometry and their appearance, as well as higher-level, spatial information, such as specifying positions. A scene graph might contain thousands of nodes; for that reason, they are retained on disk. Performer has also 3-D database loaders for OpenFlight (FLT) among other database formats.

Files in DED format are needed to get files in FLT format. MultiGen can convert DED format to FLT format. (See figure 3.12)

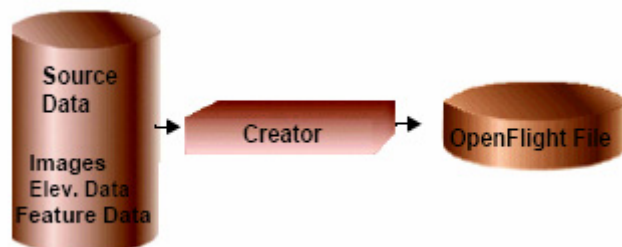


Figure 3.12 Multigen needs DED (elevation data) to create FLT [21]

The process of creating FLT from DED is showed in Figure 3.13.

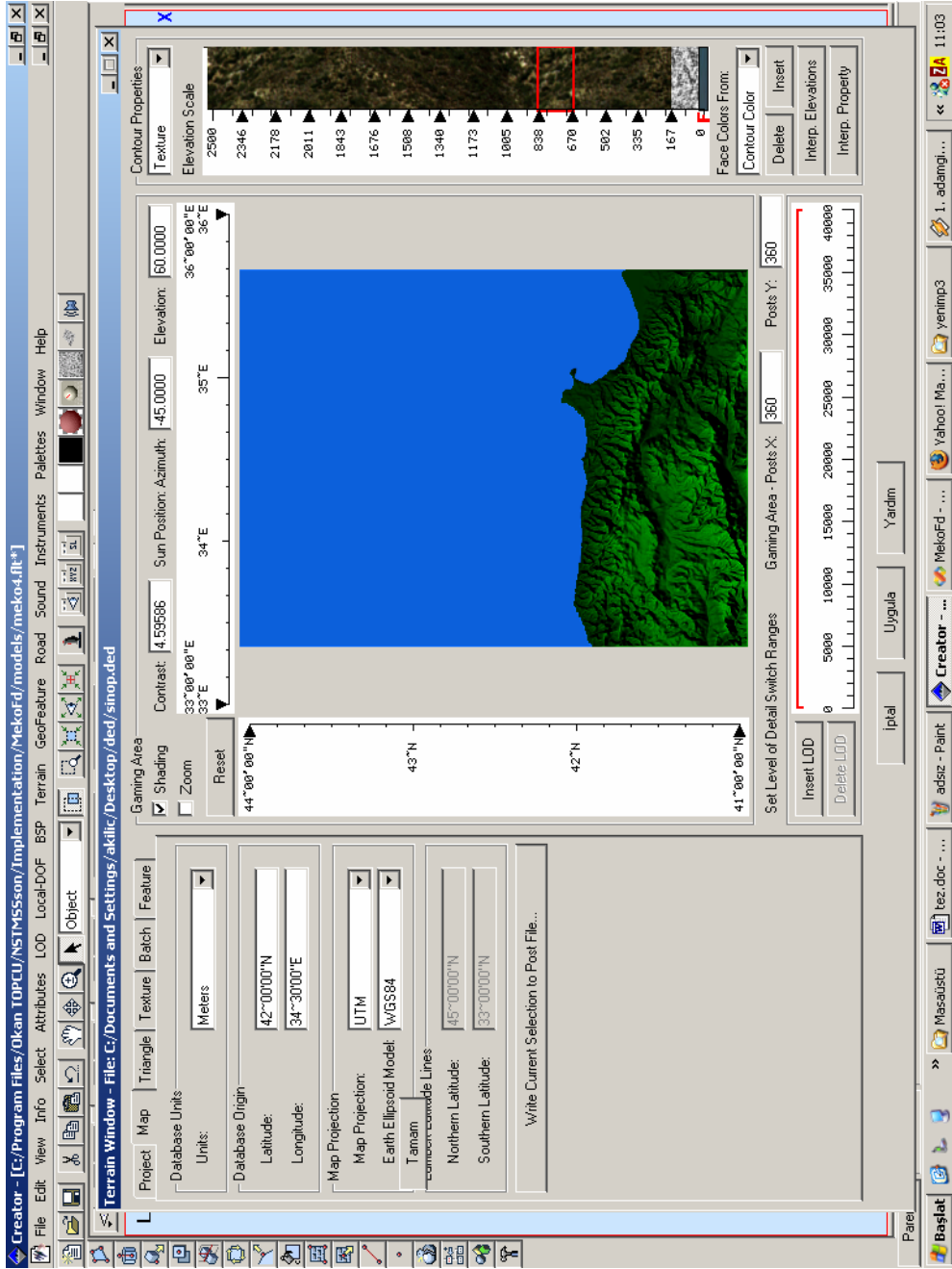


Figure 3.13 The process of creating FLT from DED



To convert elevation data from DED format to FLT format in Multigen firstly a new project is opened. It asks a DED file while opening a new Project. This conversion process is a bit complex than creating a DED file which is explained in previous figure. It is needed to decide some parameters at left side of the tool. Parameters selected during the conversion process are as follows;

#### Map Tab

Database units : meters  
Database origin : 42 00 00 N, 34 30 00 E  
Map projection : UTM  
Earth ellipsoid method: WGS84

#### Triangle Tab

Algorithm : Polymesh  
Post sampling rate: 7  
Polygon Count : 5408

If sampling rate decreases, polygon count increases, related to this the level of detail increases and consequently time of process.

Textures can be added to this creation process also. In this work three different textures are used. One of them is for water, one of them is for coasts and final one is for land. To load textures, “texture” is selected from “Contour Properties” tab at the right side as shown in the figure. Then, textures are assigned to desired elevations in the bar bottom of the Contour Properties tab.

The directory which the resultant FLT file will be saved is selected from “Project” tab.

Finally, the file in FLT format is created by pressing “Apply” button.

### **3.3 RATIONALE FOR ADOPTED TECHNOLOGIES**

This section gives a brief explanation of rationale of the used methods of coordinate systems and map projection.

#### **3.3.1 RATIONALE FOR ADOPTED COORDINATE SYSTEMS**

##### **DATUM**

In this work, World Geodetic System 1984 (WGS 84) is adopted as the reference model. The WGS 84 represents the best global geodetic reference system for the Earth available at this time for practical applications of mapping, charting, geopositioning and navigation. [3] Because full Earth wanted to be covered, WGS84 datum is useful by fitting the full Earth. Also datums and coordinates can be converted between WGS 84 and over 200 local datums.

##### **MAP PROJECTION**

Universal Transverse Mercator (UTM) Coordinate System is selected as the world coordinate system for this work. This uses UTM as the map projection method which is good for ocean navigation. It is a worldwide mapping Standard, and it is an extremely fast and efficient means of finding exact locations and approximating locations on a map. It is most often used on large scale maps and maps in this work is large scale. Also UTM coordinate system is metric and compatible with the ship's metric moves.

Mercator's 1569 map of the world was the first widely available tool for making navigation across the oceans easier. While many contemporary maps showed relatively accurate representations of coastlines, including those of North and South America, few depicted lines of longitude and latitude. Those that did showed the meridians converging toward the poles, which made charting a course difficult, as a line depicting a particular compass bearing needed to be drafted as an arc. Mercator's map allowed a navigator to draw a straight line between two points, representing a constant compass bearing. He accomplished this by simply

straightening out the converging meridians and making them parallel. Then, by tedious mathematical toiling, he stretched the parallels of latitude to match and enlarged the shapes of features to fit. The resulting image of the continents and oceans offers an odd view of the earth, but gave navigators the tool they needed to cross the oceans. These properties make the Mercator projection useful for navigation. [19]

### **3.3.2 RATIONALE FOR EULER ANGLES**

There are several popular options to represent orientations like

Euler angles and Quaternion

In this work Euler method was preferred for ship body coordinate system.

Euler has advantages in two different ways versus Quaternions;

It is easy to understand. It gives an idea about orientation. So, for visualization it is useful to give Euler angles.

Three Euler angle is necessary and enough to define the orientation, but Quaternions, which has four parameters, forms a system with a residual and requires normalization.

#### **Gimbal Lock**

One potential problem about Euler Rotation is ‘gimbal lock’.

Gimbal lock is what happens when an object is first rotated in such a way that one axis of freedom lies directly on another. For example, the rotation  $X=90$ ,  $Y=0$ , and  $Z=0$ , rotates the object that was facing forward to straight up. Now since the Z rotation is performed first, it rotates along the bone axis, and since Y is performed last, it is done after the X rotate, which means it also rotates along the bone axis. This alignment of the Z and the Y axes is referred to gimbal lock: it is not stuck, just limited in its possible movements since one degree of freedom has been lost. The object can be rotated around the vertical axis or the horizontal axis, but not the

Z-axis that points towards the screen. To get to some places, one of the other angles will need to be changed first. Any orientation can be expressed using Euler; however gimbal lock causes certain areas to be trickier than others to reach.

A solution for gimbal lock is using Quaternions. The details of Quaternion method were not explained. In this work, Euler's method was preferred because gimbal lock occurs when pitch is 90 or -90 degree. So, gimbal lock never occurs on ships unless the ship across a huge wave.

### **3.4 RELATED WORK**

Up to this section, some technical background information about digital terrains were given. In this section selected related works about this topic their relation with this thesis are introduced.

A related work is the thesis, "Creating Digital Environments For Multi-agent Simulation". [33]. This thesis explores numerous digital terrain data representations and tools available to create digital environments. The author explores the more general problem of where to find the data, what tools are available, and how to put the pieces together to create a registered digital environment. According to the author, it is key to understand that a computer tracks the digital environment in three separate coordinate systems. These coordinate systems are the world (user) coordinate system, the database coordinate system, and the pixel coordinate system. The world coordinate system is the area in which the user and human interface will interact. This could include tracking entities via Universal Transverse Mercator (UTM) and latitude-longitude coordinate systems. A database coordinate system uses a local x-y-z (0,0,0) scale to track local situated objects.[33] The world and database coordinates system that the author explains are available in our thesis. Database coordinate system that is mentioned is the local coordinate system of our work.

# CHAPTER 4

## IMPLEMENTATION

In this chapter, a detailed explanation of implementation is given. This chapter answers the questions how world and ship body coordinate systems were implemented, how the new classes were integrated to NSTMSS and which parts of NSTMSS were modified.

### 4.1 IMPLEMENTATION OF WORLD COORDINATE SYSTEM

Two new classes are added to the project to implement world coordinate system.

*Cellipsoid*  
*CworldCoordSystemTransformer*

This section firstly explains the solution then gives the related class methods descriptions.

#### SOLUTION

##### Initialization of Coordinate System

To initialize the environment the necessary information are

- Terrain name
- Terrain origin geodetic coordinates (latitude, longitude)
- Initial geodetic coordinates (latitude, longitude)

This information comes from Environment module (EnviFd [11][35]). How Environment module is to be adopted to this process is expressed in chapter 4.

Firstly terrain file (.flt) is loaded then DLLC (origin point of the terrain, default lower left corner), and initial point which the ship is positioned are set to data obtained from Environment module. If the initial position and DLLC are the same, the ship is positioned to origin of the terrain.

If this information does not come from Environment module, the ship is positioned to on a default terrain on a default position. (hawaii2.flt 21.5 N 157.75 W)

During the initialization process, geodetic coordinates are converted to UTM coordinates for later computations.

## **Ellipsoid**

*Cellipsoid* class contains most used ellipsoid parameters. Class *CworldCoordSystemTransformer* creates an ellipsoid object and can use one of these ellipsoids. In this work WGS84 ellipsoid was used. The information of WGS84 datum and the rationale of this datum was explained in chapter 3. Ellipsoid choice is hard coded as WGS84, because the terrain maps created by using this ellipsoid. If a terrain is wanted to load to the system with a different ellipsoid, class *Cellipsoid* allows this without changing it. *Cellipsoid* construction takes the ellipsoid name as its parameter. (In this case, also the terrain must be recreated using that new ellipsoid parameters.)

Ellipsoid parameters are used for conversion geodetic to UTM and UTM to geodetic coordinate system.

## Coordinate Updates

While the ship moves, moves are computed based on UTM metric coordinates then it is converted to geodetic coordinates again to show the user new geodetic position. UTM posture is updated in the ship hydrodynamic model function

```
Chydrodynamics::updatePosture(double dt)
```

by

```
CworldCoordSystemTransformer::updateWorldCoord(double x, double y);
```

## Class descriptions

```
class CworldCoordSystemTransformer
```

The maps of this program are projected using UTM projection method. This class calculates the coordinates of the ship Meko according to UTM projection. It converts geodetic coordinates to UTM coordinates and vice versa.

## Important Methods

```
CworldCoordSystemTransformer()
```

Constructor creates a *Cellipsoid* object. Ellipsoid type can be changed by providing a constructor argument. For available ellipsoid types please look at *Cellipsoid* class header.

```
GeoToUtm( double Lon , char Lon_polar, double Lat, char Lat_polar )
```

Converts a geodetic coordinate to UTM coordinate. A UTM coordinate components are right(east-west), up(north-south) and zone.

Information about conversion formulas is given in chapter 3.

*UtmToGeo(double right\_p , double up\_p , int UtmZone\_p )*

Converts an UTM coordinate to geodetic coordinate. A UTM coordinate components are right(east-west), up(north-south) and zone.

*updateWorldCoord(double x, double y)*

Moves the point on UTM coordinate system by one unit.

*Class Cellipsoid*

This class contains variables and methods of chosen ellipsoid of the map projection. The maps that this program is using are projected using WGS84 ellipsoid. For future use other mostly used ellipsoid information is added to this class. These ellipsoids are WGS-84, ED-50, BESSEL, GRS-1967, GRS-80, CLARKE-1866.

Important Methods

*Cellipsoid(char \*ellipsoid\_name)*

Constructor decides semi-major axis(a) and flattening values(f) according to the chosen ellipsoid.

## **4.2 IMPLEMENTATION OF BODY COORDINATE SYSTEM**

Two new classes and a header file for tests were added to the project to implement world coordinate system.

*CbodyCoordSystemTransformer*  
*CcartesianCoord*  
*body\_test\_coords.h*

This section explains the solution then gives related class methods descriptions.



## SOLUTION

To give the coordinate of a point on/in the ship, Cartesian coordinate system is used. (refer to chapter 2).

Briefly, the solution for body coordinate system is explained as follows;

- Specify a point in/on the ship to calculate body, local and world coordinates of that point.( Some test points coordinates on/in the ship which are based on body coordinates are in `body_test_coords.h`.)
- Apply Euler rotation to that point according to the ship Euler angles to specify the rotation in the Cartesian coordinate system of the ship.
- Calculate the point coordinate in map local coordinate system.
- Calculate the point coordinate in world coordinate system.
- Print results to a output file (`bodyCoordTestOut.txt`)

How to calculate Euler rotation and this coordinates were explained in chapter 2.

### Class descriptions

*Class CbodyCoordSystem*

This class is used for calculating body, local and world coordinates of any point on the ship Meko. Chosen test point results is written to a file.

### Important Methods

*chooseBodyPos(int point)*

Decides x, y, z coordinates of the indicated test point on body coordinate system.

*EulerRotation(double theta, double phi, double psi, int point)*

Calculates Euler Rotation of a point on the body of the ship Meko. Information about Euler rotation is given in chapter 2.

*updateLocalCoord(double localX, double localY)*

Calculates local coordinate of a point on the ship. Information about local coordinate system is given in chapter 2.

*updateWorldCoord(double worldX, double worldY)*

Calculates world coordinate of a point on the ship. Information about world coordinate system was given in chapter 2.

*Class CcartesianCoord*

This class is used to hold any Cartesian coordinates. It is especially used for ship body coordinate system. Information about Cartesian coordinate system is given in chapter 2.

### **Other Classes**

*Class CcoordFormater*

This class is for formatting latitude and longitude. It converts them from floating point notation to degree, min and sec notation.

Information about this conversion formula is given in chapter 3.

## **4.3 INTEGRATION OF NEW CLASSES TO NSTMSS**

This section explains how new classes created by this work are integrated to NSTMSS, which is the test case of this thesis. World coordinate system and body coordinate system integration processes are given separately.

### 4.3.1 INTEGRATION OF WORLD COORDINATE SYSTEM CLASS

All integrated methods mentioned here belongs to class *CworldCoordSystemTransformer*.

*updateWorldCoord* method updates ship world coordinate by moving the ship origin point on UTM coordinate system by one unit. This method was added to *Chydrodynamics::updatePosture(double dt)* method.

*UtmToGeo* method was added to *Cmeko::convert\_geocentric\_to\_geodetic(pfCoord centric)* method and old formulas for conversion were deleted.

### 4.3.2 INTEGRATION OF BODY COORDINATE SYSTEM CLASS

All integrated methods mentioned here belongs to class *CbodyCoordSystemTransformer*.

*EulerRotation* method makes Euler rotation of the ship Cartesian coordinate system. This method was added to *hydrodynamics::updatePosture(double dt)* method.

*updateLocalCoord* method updates body coordinate of the selected point on the ship in the local coordinate system. This method was added to *Chydrodynamics::updatePosture(double dt)* method.

*updateWorldCoord* method updates body coordinate of the selected point on the ship in the world(geodetic) coordinate system. This method was added to *Chydrodynamics::updatePosture(double dt)* method.

*printResultsToFile* method prints the three different coordinate of a ship point to an output file.

## 4.4 CHANGES TO NSTMSS

In addition to adding new features about coordinate systems to NSTMSS also, some changes have been made to environment federate to adopt it to new coordinate systems. Sections below explains these changes.

### 4.4.1 CHANGES TO ENVIRONMENT FEDERATE

Environment federate EnviFd sends some information like fog status, sea state, time of day to other federates via HLA. On the other hand, it does not give any information about which terrain loaded and what is the terrain coordinate specifications from Environment federate. Every federate like MekoFd sets the terrain information itself, this was hard coded.

This work improved Environment federate to send also terrain information via RTI. Terrain information is read from an input file (maps.txt) and sends. The format of maps.txt is as follows;

*Terrain\_file\_name DLLC\_latitude polar DLLC\_longitude polar Initial\_latitude polar Initial\_longitude polar*

Sample file content

*hawaii2.flt 21.5 N 157.75 W 21.5 N 157.8 W  
sinop.flt 42.0 N 34.5 E 42.1 N 34.5 E*

To send this terrain information, RTI Ambassador of EnviFd was updated and eight new handles were added;

*DLLCLatHandle  
DLLCLongHandle  
DLLCLat\_polarHandle  
DLLCLong\_polarHandle*

*InitialLatHandle*  
*InitialLongHandle*  
*InitialLat\_polarHandle*  
*InitialLong\_polarHandle*

And also at the MekoFd side, to handle this information RTI Ambassador of MekoFd was updated.

When the user update federates from Environment Graphical User Interface (GUI) by using Update Federates button (see Figure 4.1), MekoFd updates its environment and loads the terrain. The terrain is chosen in EnviFd constructor like Sinop.flt. It is written in the code manually. In the future, a scenario module can be added to NSTMSS then terrain will be chosen by the scenario. Terrain can be updated at the beginning for only one time. Meko begins with a default environment (it is loaded before the simulation loop) whether the user update federates or not.

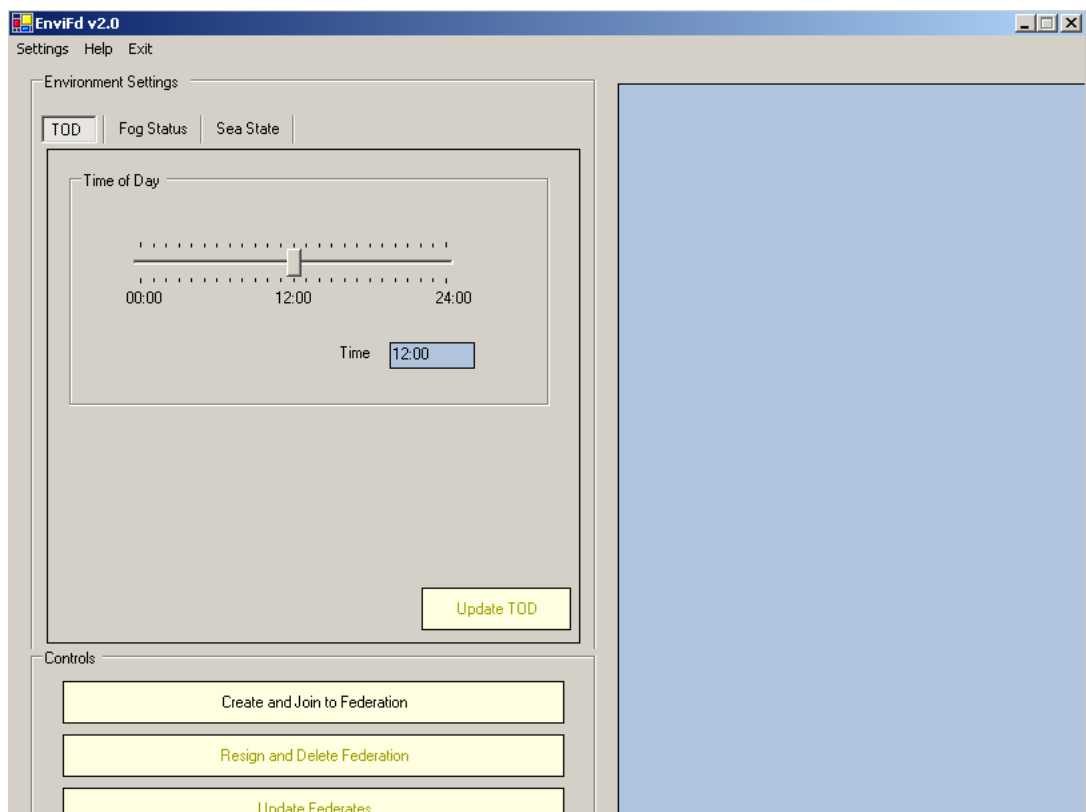


Figure 4.1 Terrain update from Environment module

Methods added to EnviFd module:

```
void setTerrain(char *name, float DLLC_Long, char DLLC_NS, float  
DLLC_Lat, char DLLC_EW, float initial_Long, char initial_NS, float  
inital_Lat, char initial_EW);
```

Reads terrain information from input file maps.txt. This information is sent to federates via RTI.

```
int readMapData(char user_map_name[]);
```

Sets DLLC and initial posture after reading them from input file maps.txt via *readMapData* function. Initial posture is different from DLLC. If the ship is wanted to position to a different location then DLLC, initial posture is used. Otherwise, initial posture must /be set same as DLLC.

Here is the list of changes to EnviFd:

- *readMapData* function added to construction *Cenvi()* to read terrain input file maps.txt. Terrain name is given as the parameter.
- RTI Ambassador of EnviFd module (*CrtiAmb* class) was updated to add new handles mention above. *publishandSubscribe* and *createAHVPS* method was updated so as to contain new terrain handles.
- RTI Ambassador of MekoFd module (*CrtiAmb* class) was updated to add new handles mention above. *publishandSubscribe* and *reflectEnvironmentUpdates* methods were updated so as to get terrain information.

# **CHAPTER 5**

## **TESTING**

This chapter gives a brief explanation about how the new features added to NSTMSS by this work have been tested. As in the Implementation chapter, testing consists of two parts; unit testing of world coordinate system and integration tests of world and body coordinate systems. (For the details of implementation refer to chapter 4).

### **5.1 UNIT TESTS**

This section dedicated to testing phase. Unit and integration tests of world coordinate system and body coordinate system are given separately.

#### **5.1.1 WORLD COORDINATE SYSTEM UNIT TESTS**

The most important part of the world coordinate system implementation is conversions from geodetic coordinates to UTM coordinates and vice versa.

#### **TESTING TOOLS**

For unit testing of these conversion formulas (given in Chapter 4), it is benefited from a tool named GEOTRANS. This tool was used to check conversions results taken from the unit tests. Information about this tool is available in chapter 1.

## TESTS

Two conversion formulas were tested in this unit test;

- conversion form geodetic coordinates to UTM coordinates
- conversion form UTM coordinates to geodetic coordinates

Some arbitrary and some critical limit coordinates (for example the coordinates where the zone changes) were given to unit program and the results were checked from GEOTRANS. A snapshot view of the GEOTRANS tool is depicted in Figure 5.1.

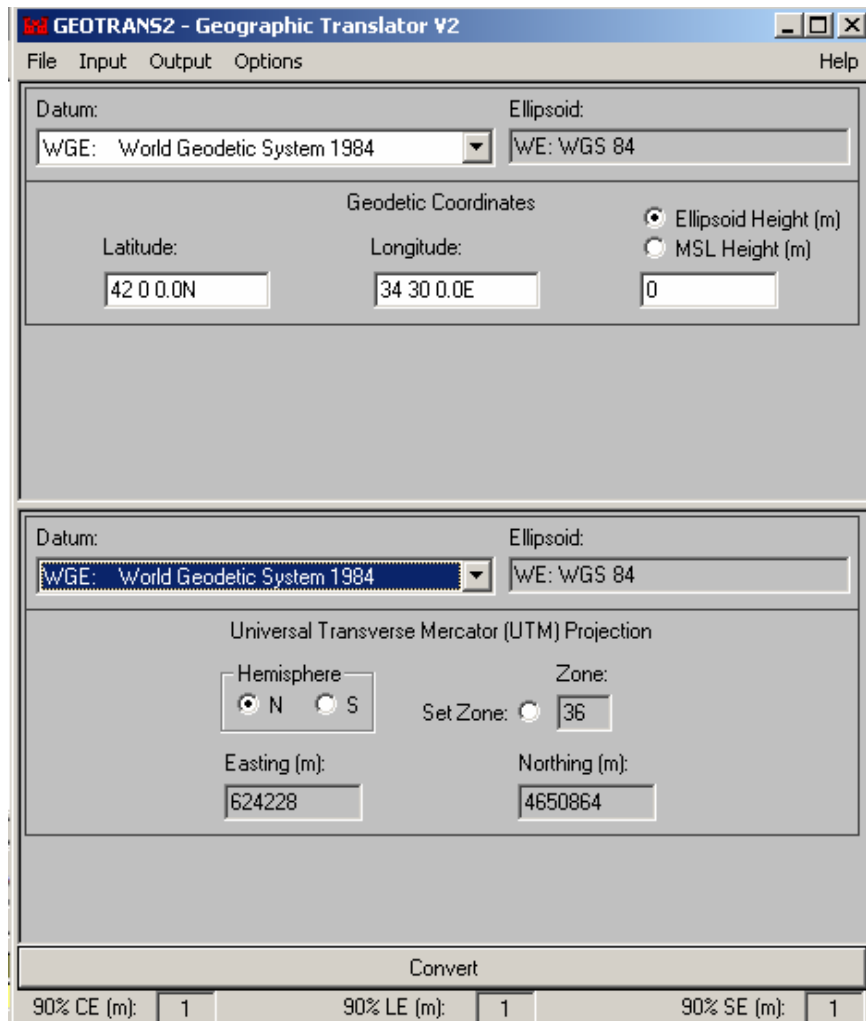


Figure 5.1 Unit testing with GEOTRANS



In the Figure 5.1, origin geodetic coordinates of sinop.flt was given in input part, and the resulting UTM coordinates can be seen in the output part.

## **5.2 INTEGRATION TESTS**

### **5.2.1 WORLD COORDINATE SYSTEMS INTEGRATION TESTS**

Firstly, the class CworldCoordSystemTransformer that contains conversions was integrated to NSTMSS as explained in Chapter 4. Then the simulation was tested with its new coordinate system.

The test data is the simulation data itself. While the ship is moving, the coordinates and polarity was observed. Also, the ship was moved with different headings.

In Figures 5.2 and 5.3 , simulation GUI and DOS prompt can be seen. While the simulation is running the data in the DOS prompt was checked between some time intervals.

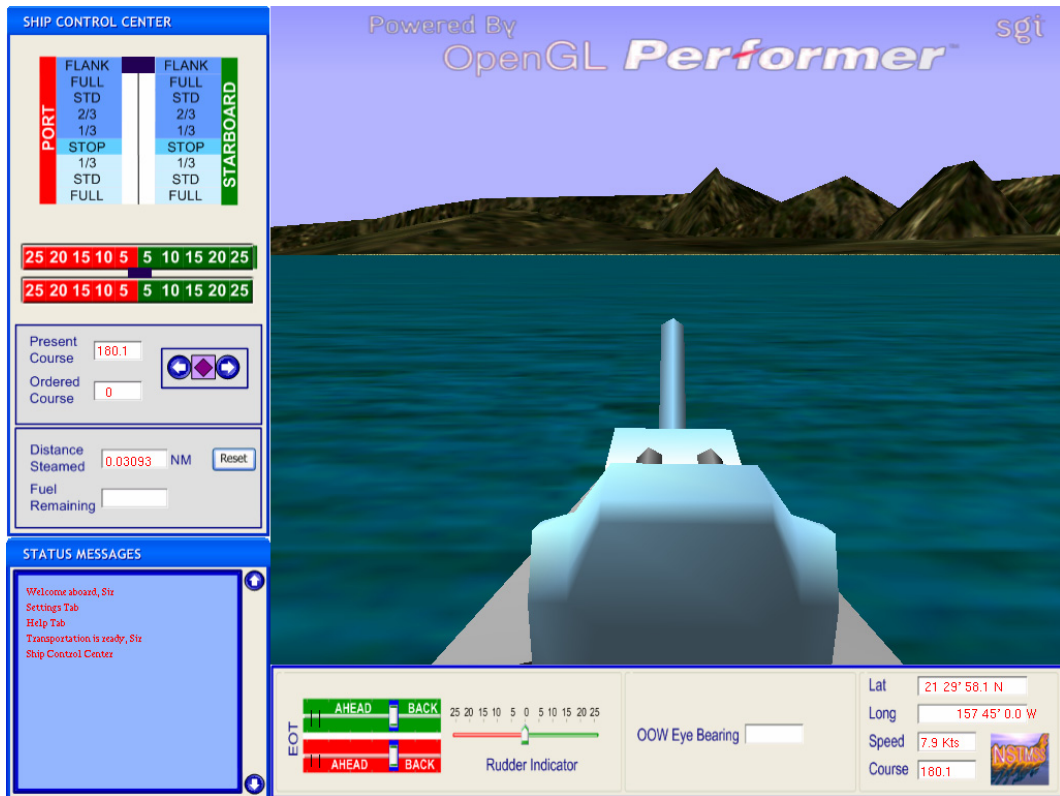


Figure 5.2 Integration testing

```

c:\Program Files\Okan TOPCU\NSTMSSon\Implementation\MekoFd\Release\MekoFd.exe" sinop
UTM x coordinate=629480.901862
UTM y coordinate=2377933.708915
LAT = 21 29' 57.7 N
LONG =157 45' 0.0 W

UTM x coordinate=629480.901107
UTM y coordinate=2377933.445467
LAT = 21 29' 57.7 N
LONG =157 45' 0.0 W

UTM x coordinate=629480.900283
UTM y coordinate=2377933.158488
LAT = 21 29' 57.7 N
LONG =157 45' 0.0 W

UTM x coordinate=629480.899466
UTM y coordinate=2377932.875107
LAT = 21 29' 57.7 N
LONG =157 45' 0.0 W

UTM x coordinate=629480.898687
UTM y coordinate=2377932.605525
LAT = 21 29' 57.7 N
LONG =157 45' 0.0 W

```

Figure 5.3 Integration testing command prompt

### 5.2.2 BODY COORDINATE SYSTEM INTEGRATION TESTS

To test ship body coordinate system, firstly, some test points on the ship were chosen.

Total 25 points on the ship were tested, 13 of them boundary points and the others are arbitrary points. As mentioned in Chapter 4, they are written to a test input file (body\_test\_coords.h).

The test points are showed in Figures 5.4-5.8 above.

#### Boundary Points

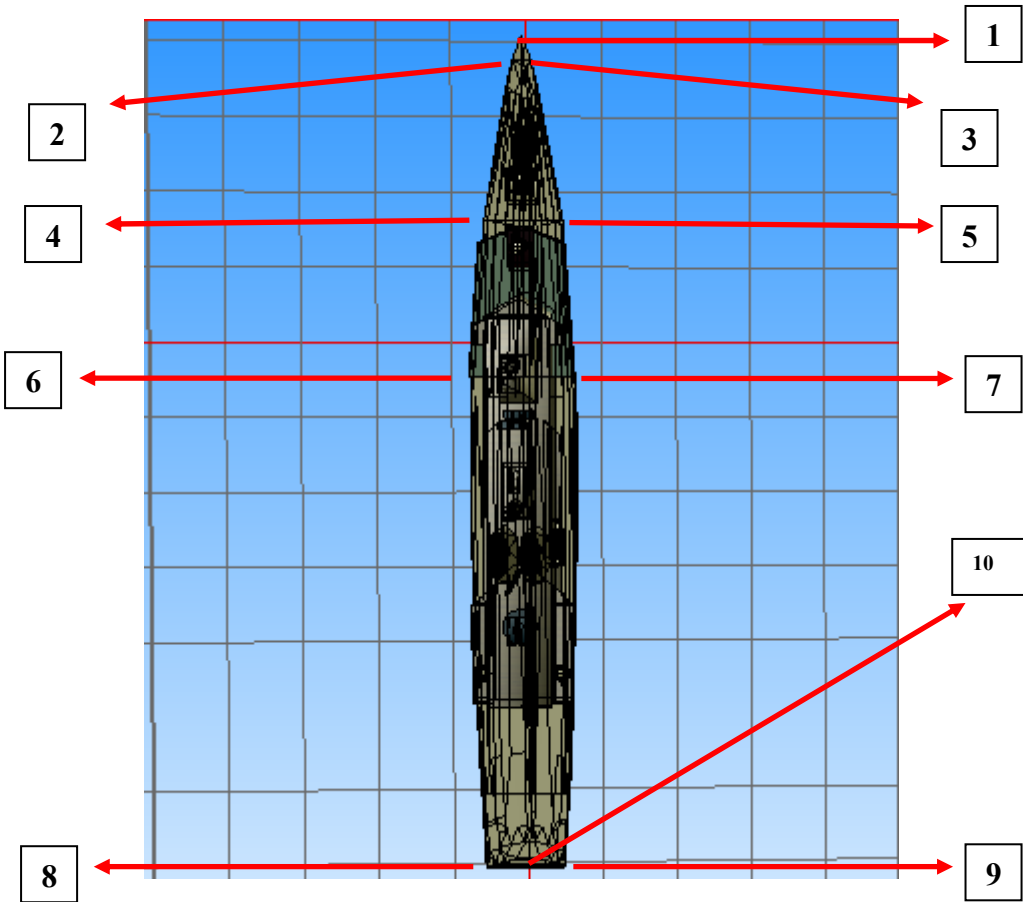


Figure 5.4 Boundary test points on the ship.

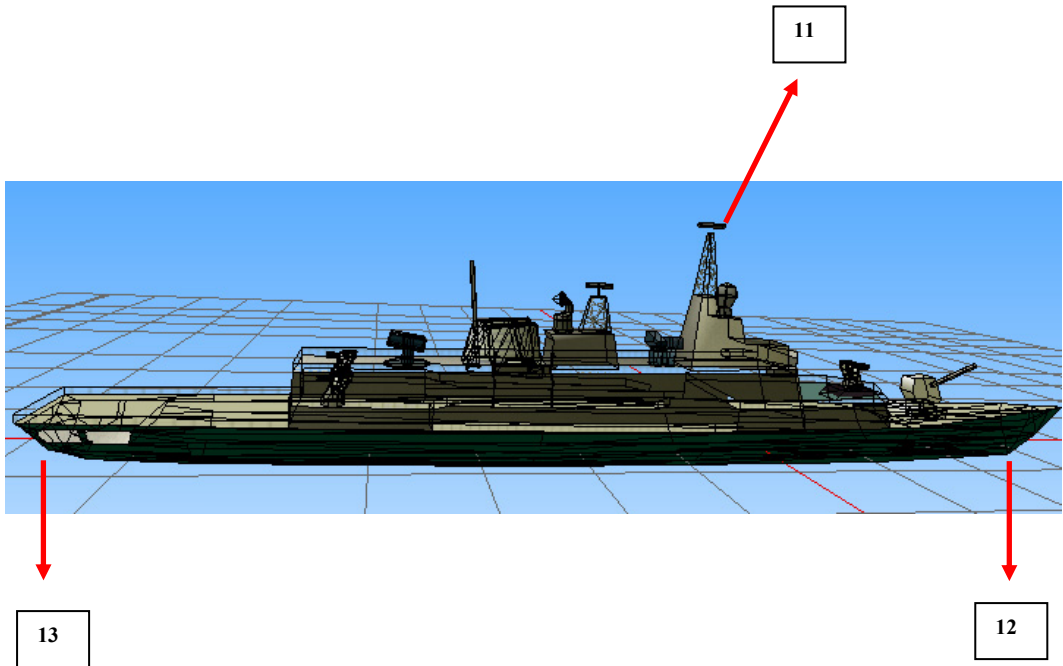


Figure 5.5 Boundary test points on the ship (cont.)

**Other Arbitrary Points**

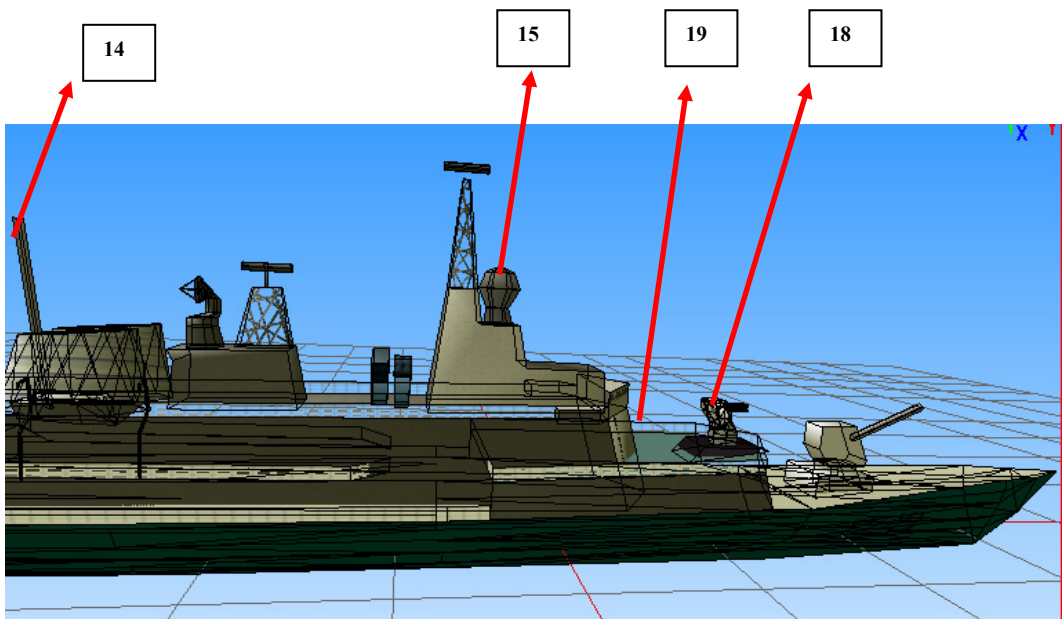


Figure 5.6 Arbitrary boundary test points on the ship (front view)

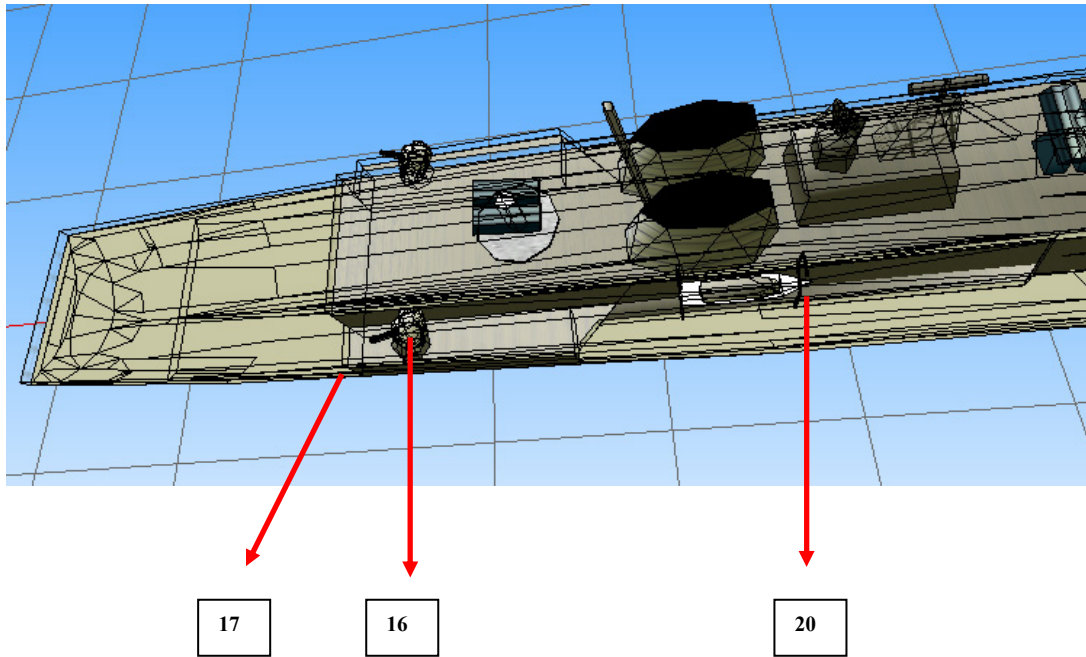


Figure 5.7 Arbitrary boundary test points on the ship cont. (backside)

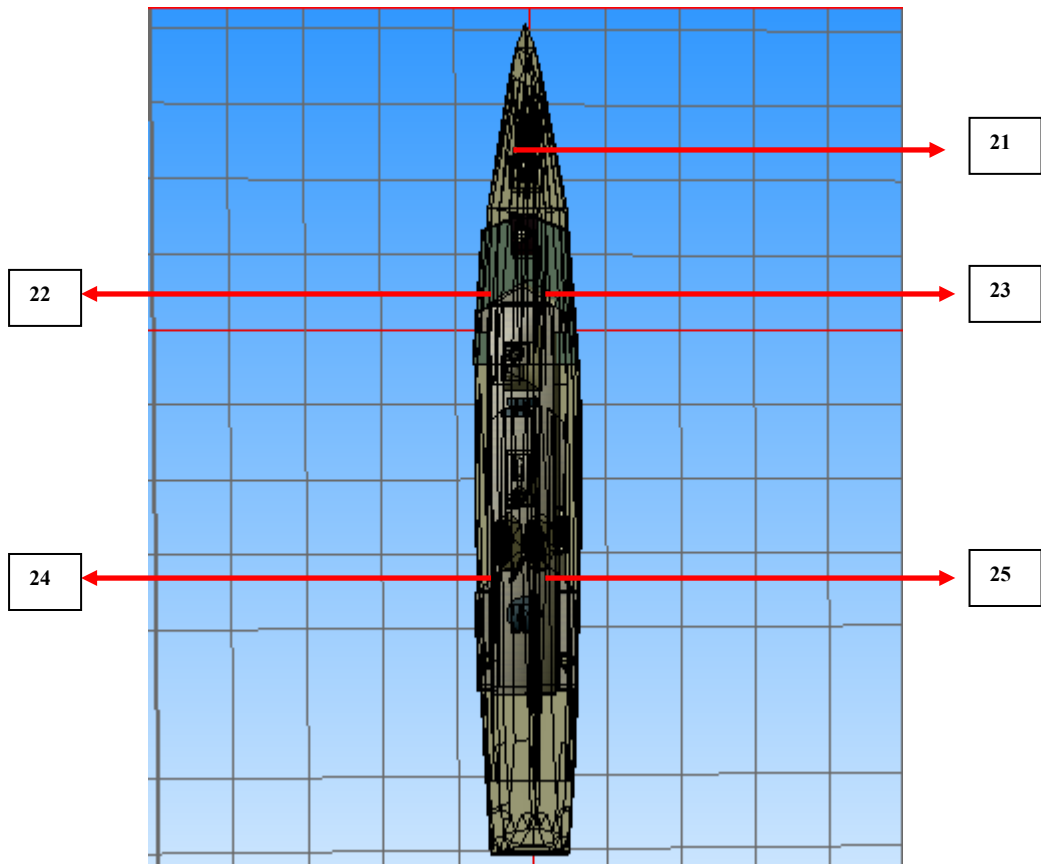


Figure 5.8 Arbitrary boundary test points on the ship (cont.)

Regarding test data for the test points above is in Appendix B.

The results of the tests are written to an output file (bodyCoordTestOut.txt). The format of this file is

*COORDINATES OF TEST POINT*

*BODY COORD:*

*X=0 Y=39.2 Z=4*

*LOCAL COORD:*

*X=-0.102399 Y=-65.5128 Z=4*

*WORLD COORD:*

*LAT:21 29 57.8697N*

*LONG:157 45 0.0217438W*

*HEIGHT:4*

*WORLD COORD. OF THE SHIP:*

*LAT=21 29 59.1443N*

*LONG=157 45 0.00823975W*

The results were checked from the output. The test point can be changed from *body\_coord.EulerRotation(theta, phi, psi, l)* method ( the last parameter ) in *void Chydrodynamics::updatePosture(double dt)* method.

BODY COORD is Cartesian coordinate of the test point according to the ship origin. (ship body coordinate system was explained in chapter 4).

LOCAL COORD is local map coordinate of the test point.(local coordinate system was explained in chapter 4).

WORLD COORD is world coordinate of the test point with.(world coordinate system was explained in chapter 4).

WORLD COORD. OF THE SHIP is world coordinate of the ship (origin of the ship).

## **CHAPTER 6**

### **CONCLUSIONS AND DISCUSSIONS**

Coordinate systems serve as the framework by which geographic data are referenced to the earth's surface. As such, coordinate systems are vital to mapping, surveying, and engineering. Coordinate systems are especially important in navigation systems because they provide a common reference for all information.

“For good reasons many different coordinate systems and associated earth reference models are in use. The same is true for simulation applications. However, it is not clear that the choice of a coordinate system for near-Earth simulation applications has always been either well informed, or especially rational.”[1]

A well formed and rational coordinate system to NSTSMSS has been implemented. A new world coordinate system and also a body coordinate system for ship models were implemented to improve the realism of NSTMSS.

#### **6.1 CONTRIBUTIONS OF THIS WORK**

A world coordinate system was implemented with a reference model to identify the Earth so it can be readily associated with the physical world so that its use is intuitive. For this WGS 84, which is NIMA's best geodetic model of the Earth using data, techniques and technology available through 1996, was chosen. So, the coordinate system is based on a more accurate and efficient system relative to the old coordinate system of NSTMSS.

A proper map projection method was chosen. UTM was used as the map projection method which is particularly good for ocean navigation. Accurate converting methods between needed coordinate presentations were implemented.

A body coordinate system has been implemented for the ship models to able to define every point of a ship models for tactic scenarios.

## **6.2 DISCUSSION**

In [1] it is stated that the environment, geometric, and dynamic models are strongly coupled to the spatial reference model that is employed. model that is employed and many archived environmental databases were developed in augmented, map projection-based systems with respect to spherical earth models. The use of augmented, map projection-based coordinate frameworks distorts reality and may lead to an unlevel playing field. The paper recommends that only real-world coordinate systems and standardized earth reference models be used for future simulation developments.

However, most simulators use a Universal Transverse Mercator (UTM) reference grid because it is locally cartesian, electromagnetic waves travel in straight lines and therefore calculations are easier in such systems

Map projections are very useful, if not essential, for the visual portrayal of the surface of the earth, but they are not adequate as a basis for deriving simulation models for future applications[1]. Thus, our work follows the suggestions of [1].

Another discussion point is the necessity for embedding coordinate conversion functions in every dynamic model of virtual environment in this thesis. As argued in [34], coordinate conversion of entity position data points, can be applied at the Middleware software layer. This also corrects the classic problem of how different coordinate based Federates can interoperate. For this work a middle layer can be added to the federation with these functions;



- Universal Transverse Mercator (UTM) location x,y,z (meters) into Geodetic World Geodetic System (WGS) 84 coordinates
- WGS84 into UTM

## **6.3 SUGGESTIONS FOR FUTURE WORK**

### **TERRAIN BORDER CONTROL**

Military objects are heavily dependent upon the environment in which they operate. The representation of terrain has been of primary concern because of its importance and the difficulty of managing the amount of data required. Terrain border control is beyond the scope of this project. The borders of the terrain which the simulation runs is not determined, and the simulation can overflow beyond the environment borders. So, it is necessary a terrain border control for every terrain used.

### **MULTI-TERRAIN**

This simulation system can run on only one terrain at a simulation time. It is necessary to stop the simulation and load an another terrain to change the environment. To overcome this limitation, a multi-terrain method should be added, so the simulation can run all over the world by changing the terrains in run time.

## REFERENCES

[1] Ralph M. Toms, Paul A. Birkel, “*Choosing a Coordinate Framework for Simulations*”, Synthetic Environment Data Representation and Interchange Specification Organization (SEDRIS), Aug 1999

[2] UK Ordnance Survey, “*A guide to coordinate systems in Great Britain*”, May 2002

[3] National Imagery and Mapping Agency (NIMA), “*Department of Defense World Geodetic System 1984 Third Ed*”, 3 Jan 2000

[4] EUROCONTROL European Organization for the Safety of Air Navigation Brussels, Belgium , IfEN Institute of Geodesy and Navigation (IfEN) University FAF Munich, Germany , “*WGS 84 Implementation Manual*”, 12 Feb 1998

[5] Indiana Geographic Information Council, “*Projections, Datum, Coordinate Systems, and Units of Measure Standard*”, 26 Jun 2001

[6] Jones A., “*Where in the World are We? Version 1.7*”, Resource Information Group Department for Environment, Heritage and Aboriginal Affairs in cooperation with the South Australian Spatial Information Committee, Aug 1999

[7] Moritz H, “*Fundamental Geodetic Constants*”, Report of Special Study Group No. 5.39 of the International Association of Geodesy (IAG), Dec 1979.

[8] D. McCarthy, “*IERS Technical Note 21*”, IERS Conventions (1996), Observatoire de Paris, 1 July 1996.

[9] National Imagery and Mapping Agency (NIMA), “*Performance Specification Digital Terrain Elevation Data (DTED)*”, 19 May 2000

[10] Dutch S., "Converting UTM to Latitude and Longitude (Or Vice Versa)", Natural and Applied Sciences, University of Wisconsin, 19 May 2005

[11] Topçu O., "NSTMSS Technical Report", Oct 2003

[12] The Department of Trade and Industry of UK, "Guidance notes on the use of co-ordinate systems in data management on the UKCS", [http://www.og.dti.gov.uk/regulation/guidance/co\\_systems/index.htm](http://www.og.dti.gov.uk/regulation/guidance/co_systems/index.htm), Aug. 2005

[13] Riesterer J., "UTM - Universal Transverse Mercator Geographic Coordinate System", Geospatial Training and Analysis Cooperative, 18 Sep 2003

[14] University College of Natural Resources, "The Universal Transverse Mercator (UTM) Coordinate System", [http://www.cnr.colostate.edu/class\\_info/nr502/lg3/datums\\_coordinates/utm.html](http://www.cnr.colostate.edu/class_info/nr502/lg3/datums_coordinates/utm.html), Aug 2005

[15] National Imagery and Mapping Agency (NIMA), "Vertical Datums, Elevations, and Heights", [http://www.usna.edu/Users/oceano/pguth/website/so432web/e-text/NIMA\\_datum/VERTICAL%20DATUMS.htm](http://www.usna.edu/Users/oceano/pguth/website/so432web/e-text/NIMA_datum/VERTICAL%20DATUMS.htm), Aug. 2005

[16] Rice University Civil and Environmental Engineering Department, "Geodesy and Map Projections", <http://doctorflood.rice.edu/envi512/>, Aug 2005

[17] European Organization for the Safety of Air Navigation, "Digital Terrain Data and Site Information", <http://www.eurocontrol.int/sass/dted.htm>, Aug. 2005

[18] Sigurd Humerfelt, "The Earth according to WGS 84", [http://home.online.no/~sigurdhu/Grid\\_1deg.htm](http://home.online.no/~sigurdhu/Grid_1deg.htm), Aug. 2005

[19] Cadapult Software Solutions Inc., "History of Cartography Part III – Map Projections", <http://www.cadapult-software.com/tutorials/cartography3>, Aug. 2005

[20] National Imagery and Mapping Agency (NGA), "GEOTRANS Users Guide", <http://earth-info.nga.mil/GandG/geotrans>, Aug 2005

[21] Multigen-Paradigm Inc., "Multigen Creator", <http://www.multigen.com/products/database/creator/index.shtml>, Aug. 2005

[22] Wikipedia.com The Free Encyclopedia, "*Datum*", <http://www.wikipedia.com>, Aug. 2005

[23] National Imagery and Mapping Agency (NGA), "*NGA Raster Roam*", [http://geoengine.nima.mil/geospatial/SW\\_TOOLS/NIMAMUSE/webinter/rast\\_roam.html](http://geoengine.nima.mil/geospatial/SW_TOOLS/NIMAMUSE/webinter/rast_roam.html), Aug. 2005

[24] National Center for Geographic Information & Analysis, "Position shifts from datum differences", <http://www.ncgia.ucsb.edu/education/curricula/giscc/units/u015/figures/figure08.gif>, Aug. 2005

[25] MapInfo Corporation, "Datums", [http://extranet.mapinfo.com/support/downloads/pro\\_datums.cfm](http://extranet.mapinfo.com/support/downloads/pro_datums.cfm), Aug. 2005

[26] The Moves Institute, Naval Postgraduate School, "*NPSNET*", <http://www.npsnet.org/~npsnet/v/index.html>, Aug. 2005

[27] David R. Pratt, "*A Software Architecture for the Construction and Management of Real-Time Virtual Worlds*", Naval Postgraduate School, June 1993

[28] Robert B. McGhee, Eric Robert Bachmann, Michael J. Zyda, "*Rigid Body Dynamics, Inertial Reference Frames, and Graphics Coordinate Systems: A Resolution of Conflicting Conventions and Terminology*", Naval Postgraduate School, Nov 2000

[29] Don Brutzman, "*A Virtual World for An Autonomous UnderWater Vehicle*", Naval Postgraduate School, Dec 1994

[30] Lewis Lidward V., "*Principles of Naval Architecture Volume 3*", Society of Naval Architects Marine Engineers (SNAME), 1988

[31] Fu K.S. Gonzales, "*Robotics: Control Sensing, Vision and Intellegence*", McGraw Hill., 1987

[32] Metron Inc., Metron Simulation Sicence Division, "*JWARS Overview*", [http://www.msiac.dmsso.mil/spug\\_documents/JWARS\\_Overview\\_Brief.ppt](http://www.msiac.dmsso.mil/spug_documents/JWARS_Overview_Brief.ppt), Aug. 2005

[33] Mark B. Taner, "*Creating Digital Environments For Multi-agent Simulation*", Naval Postgraduate School, Dec 2003

[34] Daniel J. Paterson, Erik S. Houglund, Ph.D., Juan J. Sanmiguel, “*A Gateway/Middleware HLA Implementation and The Extra Services That Can Be Provided to The Simulation*”, Research and Engineering Naval Air Warfare Center Training Systems Division (NAWCTSD), 2000

[35] Topçu O. and Oğuztüzün H., “*Developing an HLA-Based Naval Maneuvering Simulation*“, Naval Engineers Journal, vol. 117, no. 1, Winter 2005, pp.23-40.

[36] University of California at San Diego Computer Graphics Laboratory, “*Orientation and Quaternions*”, [http://graphics.ucsd.edu/courses/cse169\\_w05/CSE169\\_04.ppt](http://graphics.ucsd.edu/courses/cse169_w05/CSE169_04.ppt), Aug. 2005

[37] Chocron O., “*Euler ZYX Notation - Course Notes*”, Massachusetts Institute of Technology, Oct 2000

# APPENDIX- A

## WGS 84 PARAMETERS OF THE EARTH DEFINITION

Table A.1 WGS84 Parameters [3]

Semi-minor axis = polar radius = $b = (1-f)a =$	6 356 752.3142 m
Difference between equatorial and polar radius = $a-b =$	<u>21 384.6858 m</u>
Axis ratio = $b/a = 1-f =$	0.996 647 189 335
First eccentricity squared = $e^2 = 1-(b/a)^2 = (2-f)f =$ First eccentricity = $e =$	0.006 694 379 990 14 0.081 819 190 842 622
Surface area of the Earth = $A = 2\pi a^2 + \pi(b^2/e)\ln[(1+e)/(1-e)] =$ Radius of sphere of equal surface area = $\frac{1}{2}\sqrt{(A/\pi)} =$	510 065 621.724 km <sup>2</sup> 6 371 007.1809 m
Volume of the Earth = $V = 4\pi a^2 b/3 =$ Radius of sphere of equal volume = $(\sqrt[3]{V/\pi})^{3/2} = (a^2 b)^{1/3} =$	1 083 207 319 801 km <sup>3</sup> 6 371 000.7900 m
Harmonic mean radius of the Earth = $3ab/(a+2b) =$	6 370 992.8027 m
Maximum circumference of the Earth = circumference of the Earth at the equator = circumference of parallel of latitude at 0° latitude = $2\pi a =$ Radius of sphere of equal circumference = $a =$	40 075.017 km (see above)
Minimum circumference of the Earth = circumference of the Earth through the poles = $4 \times$ (distance from the equator to a pole) = $4 \times 10001.966$ km = Radius of sphere of equal circumference = $40\,007\,863$ m/ $2\pi =$	40 007.863 km 6 367 449.1458 m
Difference between maximum and minimum circumference =	67.154 km
Radius of curvature at the poles = $a/(1-e^2)^{1/2} =$ Radius of curvature in a meridian plane at the equator = $a(1-e^2) =$	6 399 593.6258 m 6 335 439.3273 m
Difference between maximum and minimum radius of curvature =	64 154.2985 m
Latitude halfway between the equator and a pole =	45.14432° 45° 08.659' 45° 08' 39.5" (1 min. of lat. = 1852.243 m) (1 min. of long. = 1310.811 m)

## APPENDIX-B

### SHIP BODY COORDINATE SYSTEM TEST INPUT DATA

Table B.1 Ship Body Coordinate System Test Input Data

<b>Boundary Point Number</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
1	0.0	39.2	4.0
2	-1.2	36.2	3.9
3	1.2	36.2	3.9
4	-5.1	15.7	2.9
5	5.1	15.7	2.9
6	-6.8	-7.3	3.7
7	6.8	-7.3	3.7
8	-5.0	-69.2	3.7
9	5.0	-69.2	3.7
10	0.0	-69.2	3.7
11	0.1	-5.0	22.8
12	0.0	32.8	-1.6
13	0.0	-65.6	-1.9
<b>Arbitrary Points</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
14	0.0	26.5	9.0
15	0.0	-4.3	16.8
16	-5.2	-41.6	7.7
17	6.2	-46.9	5.7
18	0.5	10.2	8.9
19	-6.7	-4.8	5.9
20	5.0	-22.8	8.8
21	0.0	4.76	10.3
22	-4.5	4.3	11.5
23	4.5	4.3	11.5
24	-4.5	-41.0	8.5
25	4.5	-41.0	8.5

## APPENDIX-C

### THE EARTH ACCORDING TO WGS 84

This table lists the lengths in feet of 1 minute, 1 second and 1/1000 minute of latitude and longitude.

Table C.1 The Earth According to WGS84 [18]

Latitude	Distance to nearest pole		1 minute of latitude	1 minute of longitude
	Int. naut. miles	Kilometers	Meters	
90°	0	0	1861.57	0
89°	60.310	111.694	1861.56	32.49
88°	120.619	223.387	1861.54	64.97
87°	180.928	335.079	1861.51	97.43
86°	241.236	446.769	1861.47	129.85
85°	301.542	558.456	1861.42	162.24
84°	361.846	670.139	1861.36	194.58
83°	422.148	781.819	1861.29	226.86
82°	482.448	893.493	1861.20	259.06
81°	542.744	1005.163	1861.11	291.19
80°	603.038	1116.826	1861.00	323.22
79°	663.327	1228.482	1860.88	355.16
78°	723.613	1340.131	1860.75	386.99
77°	783.894	1451.772	1860.61	418.69
76°	844.171	1563.405	1860.47	450.26
75°	904.443	1675.028	1860.31	481.70
74°	964.709	1786.642	1860.14	512.99
73°	1024.970	1898.244	1859.96	544.11
72°	1085.225	2009.836	1859.77	575.07
71°	1145.473	2121.417	1859.57	605.85
70°	1205.715	2232.985	1859.37	636.44
69°	1265.951	2344.541	1859.15	666.84
68°	1326.179	2456.083	1858.93	697.03
67°	1386.400	2567.612	1858.70	727.00
66°	1446.613	2679.127	1858.46	756.75



Table C.1 The Earth According to WGS84 (continued)

65°	1506.818	2790.627	1858.21	786.26
64°	1567.015	2902.112	1857.96	815.53
63°	1627.204	3013.581	1857.69	844.55
62°	1687.384	3125.035	1857.43	873.30
61°	1747.555	3236.472	1857.15	901.79
60°	1807.718	3347.893	1856.87	930.00
59°	1867.871	3459.297	1856.59	957.92
58°	1928.015	3570.683	1856.29	985.55
57°	1988.149	3682.052	1856.00	1012.87
56°	2048.274	3793.403	1855.70	1039.88
55°	2108.388	3904.735	1855.39	1066.57
54°	2168.493	4016.050	1855.08	1092.93
53°	2228.588	4127.345	1854.77	1118.95
52°	2288.673	4238.622	1854.46	1144.63
51°	2348.747	4349.880	1854.14	1169.96
50°	2408.811	4461.119	1853.82	1194.93
49°	2468.865	4572.338	1853.50	1219.53
48°	2528.908	4683.538	1853.17	1243.76
47°	2588.941	4794.719	1852.85	1267.60
46°	2648.963	4905.880	1852.52	1291.05
45°	2708.975	5017.021	1852.20	1314.11
44°	2768.976	5128.143	1851.87	1336.77
43°	2828.966	5239.246	1851.55	1359.02
42°	2888.946	5350.329	1851.22	1380.85
41°	2948.916	5461.392	1850.90	1402.25
40°	3008.875	5572.437	1850.58	1423.23
39°	3068.824	5683.462	1850.26	1443.77
38°	3128.762	5794.468	1849.94	1463.87
37°	3188.690	5905.455	1849.63	1483.53
36°	3248.609	6016.423	1849.32	1502.73
35°	3308.517	6127.373	1849.01	1521.47
34°	3368.415	6238.304	1848.71	1539.75
33°	3428.303	6349.218	1848.41	1557.55
32°	3488.182	6460.113	1848.11	1574.89
31°	3548.052	6570.991	1847.82	1591.74
30°	3607.912	6681.852	1847.54	1608.10
29°	3667.763	6792.696	1847.26	1623.98
28°	3727.605	6903.524	1846.99	1639.36
27°	3787.438	7014.335	1846.73	1654.25
26°	3847.263	7125.131	1846.47	1668.63
25°	3907.080	7235.912	1846.21	1682.50
24°	3966.888	7346.677	1845.97	1695.86
23°	4026.689	7457.428	1845.73	1708.71
22°	4086.482	7568.165	1845.50	1721.04
21°	4146.268	7678.889	1845.28	1732.84
20°	4206.047	7789.599	1845.07	1744.12
19°	4265.819	7900.298	1844.87	1754.87
18°	4325.585	8010.984	1844.67	1765.08

Table C.1 The Earth According to WGS84 (continued)

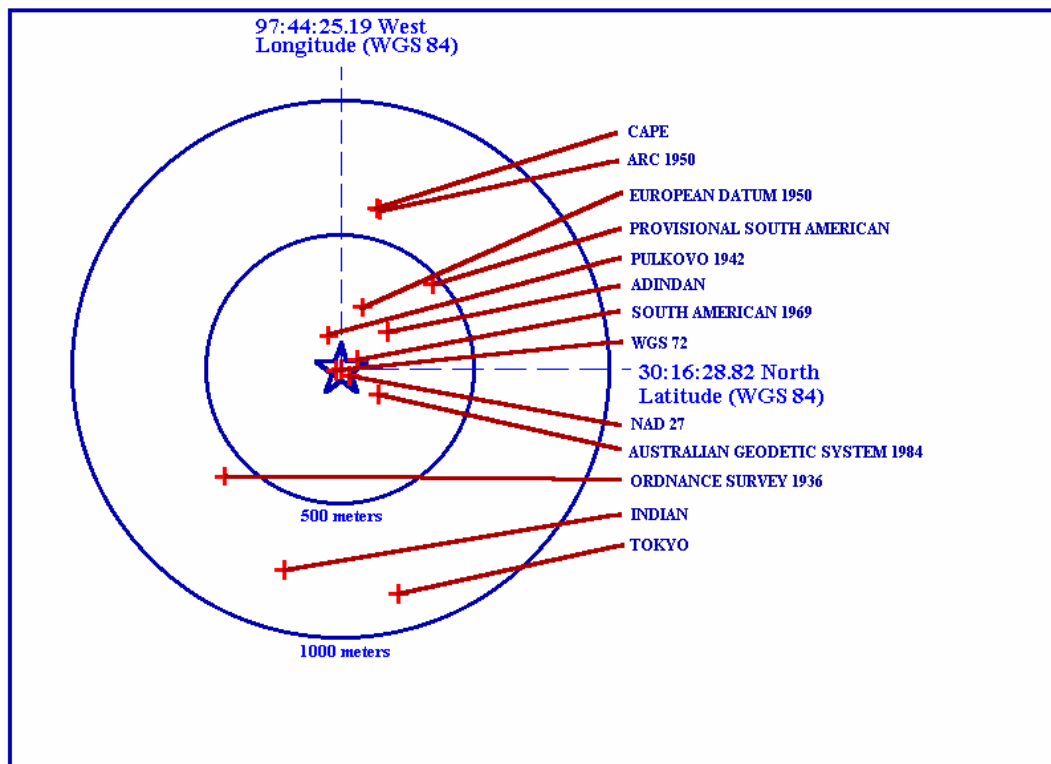
17°	4385.345	8121.659	1844.49	1774.76
16°	4445.099	8232.322	1844.31	1783.91
15°	4504.847	8342.976	1844.14	1792.51
14°	4564.590	8453.620	1843.99	1800.57
13°	4624.328	8564.255	1843.84	1808.08
12°	4684.061	8674.881	1843.70	1815.04
11°	4743.790	8785.500	1843.58	1821.46
10°	4803.516	8896.111	1843.46	1827.32
9°	4863.237	9006.715	1843.36	1832.63
8°	4922.956	9117.314	1843.26	1837.39
7°	4982.671	9227.907	1843.18	1841.59
6°	5042.384	9338.496	1843.11	1845.23
5°	5102.095	9449.080	1843.05	1848.31
4°	5161.804	9559.661	1842.99	1850.84
3°	5221.512	9670.240	1842.96	1852.80
2°	5281.218	9780.816	1842.93	1854.20
1°	5340.924	9891.391	1842.91	1855.04
0°	5400.629	10001.966	1842.90	1855.32
<b>Circumference of Earth through poles =</b>			<b>An average minute of latitude =</b>	
21602.518 int. n. mi. = 40007.863 km			1852.216 m	

An international nautical mile (int. naut. mile) is defined to be 1852 meters exactly (IHC, Monaco, 1929)

# APPENDIX D

## POSITION SHIFTS FROM DATUM DIFFERENCES

Figure D.1 Differences Between Datums [24]



## APPENDIX E

### DATUMS USED ALL OVER THE WORLD

Table E.1 Datums Used All Over The World and Areas and Ellipsoids of Datums. [25]

DATUM	AREA	ELLIPSOID
Deutsches Hauptdreiecksnetz (DHDN)	Germany	Bessel
Adindan	Ethiopia, Mali, Senegal, Sudan	Clarke 1880
Afgooye	Somalia	Krassovsky
Ain el Abd 1970	Bahrain Island	International
Anna I Astro 1965	Cocos Islands Botswana, Lesotho, Malawi, Swaziland, Zaire, Zambia, Zimbabwe	Clarke 1880
Arc 1960	Kenya, Tanzania	Clarke 1880
Ascension Island 1958	Ascension Island	International
Astro Beacon E	Iwo Jima Island	International
Astro B4 Sorol Atol	Tern Island	International
Astro DOS 71/4	St. Helena Island	International
Astronomic Station 1952	Marcus Island	International
Australian Geodetic 1966 (AGD 66)	Australia, Tasmania Island	Australian Nat'l
Australian Geodetic 1984 (AGD 84)	Australia, Tasmania Island	Australian Nat'l
Belgium	Belgium	International
Bellevue (IGN)	Efate and Erromango Islands	International
Bermuda 1957	Bermuda Islands	Clarke 1866
Bogota Observatory	Colombia	International
Campo Inchauspe	Argentina	International
Canton Astro 1966	Phoenix Islands	International
Cape	South Africa	Clarke 1880
Cape Canaveral	Florida and Bahama Islands	Clarke 1866
Carthage	Tunisia	Clarke 1880
Chatham 1971	Chatham Island (New Zealand)	International
Chua Astro	Paraguay	International
Corrego Alegre	Brazil	International
Djakarta (Batavia)	Sumatra Island (Indonesia)	Bessel 1841
DOS 1968	Gizo Island (New Georgia)	International
Easter Island 1967	Easter Island	International

Table E.1 Datums Used All Over The World and Areas and Ellipsoids of Datums  
(continued)

European 1987 (ED 87)	Europe	International
Gandajika Base	Republic of Maldives	International
Geodetic Datum 1949	New Zealand	International
Geodetic Reference System 1967 (GRS 67)	Worldwide	GRS 67
Geodetic Reference System 1980 (GRS 80)	Worldwide	GRS 80
Guam 1963	Guam Island	Clarke 1866
GUX 1 Astro	Guadalcanal Island	International
Hito XVIII 1963	South Chile (near 53°S)	International
Hjorsey 1955	Iceland	International
Hong Kong 1963	Hong Kong	International
Hu-Tzu-Shan	Taiwan	International
Indian	Thailand and Vietnam	Everest
Indian	Bangladesh, India, Nepal	Everest
Ireland 1965	Ireland	Modified Airy
ISTS 073 Astro 1969	Diego Garcia	International
Johnston Island 1961	Johnston Island	International
Kandawala	Sri Lanka	Everest
Kerguelen Island	Kerguelen Island	International
Kertau 1948	West Malaysia and Singapore	Modified Everest
L.C. 5 Astro	Cayman Brac Island	Clarke 1866
Liberia 1964	Liberia	Clarke 1880
Lisboa (DLx)	Portugal	International
Luzon	Philippines (excluding Mindanao Island)	Clarke 1866
Luzon	Mindanao Island	Clarke 1866
Mahe 1971	Mahe Island	Clarke 1880
Marco Astro	Salvage Islands	International
Massawa	Eritrea (Ethiopia)	Bessel 1841
Melrica 1973 (D73)	Portugal	International
Merchich	Morocco	Clarke 1880
Midway Astro 1961	Midway Island	International
Minna	Nigeria	Clarke 1880
Nahrwan	Masirah Island (Oman)	Clarke 1880
Nahrwan	United Arab Emirates	Clarke 1880
Nahrwan	Saudi Arabia	Clarke 1880
Naparima, BWI	Trinidad and Tobago	International
Netherlands	Netherlands	Bessel
North American 1927 (NAD 27)	Continental US	Clarke 1866
North American 1927 (NAD 27)	Alaska	Clarke 1866
North American 1927 (NAD 27)	Bahamas (ex. San Salvador)	Clarke 1866
North American 1927 (NAD 27)	San Salvador Island	Clarke 1866
North American 1927 (NAD 27)	Canada (inc Newfoundland)	Clarke 1866
North American 1927 (NAD 27)	Canal Zone	Clarke 1866
North American 1927 (NAD 27)	Caribbean (Turks, Caicos)	Clarke 1866
North American 1927 (NAD 27)	Mexico	Clarke 1866
North American 1927 (NAD 27)	Cuba	Clarke 1866
North American 1927 (NAD 27)	Cuba	Clarke 1866

Table E.1 Datums Used All Over The World and Areas and Ellipsoids of Datums  
(continued)

North American 1927 (NAD 27)	Greenland (Hayes Peninsula)	Clarke 1866
North American 1927 (NAD 27)	Michigan (used only for State Plane 1927)	Mod. Clarke 1866
North American 1983 (NAD 83)	Alaska, Canada, Central America, Continental US, Mexico	GRS 80
Nouvelle Triangulation Francaise (NTF)	France	Mod. Clarke 1880
NWGL 10	Worldwide	WGS 72
Old Egyptian	Egypt	Helmert 1906
Old Hawaiian	Hawaii	Clarke 1866
Ordnance Survey Great Britain 1936	England, Isle of Man, Scotland, Shetland Islands, Wales	Airy
Pico de las Nieves	Canary Islands	International
Pitcairn Astro 1967	Pitcairn Island	International
Provisional South Chilean 1963	South Chile (near 53°S)	International
Provisional South American 1956	Bolivia, Chile, Colombia, Ecuador, Guyana, Peru, Venezuela	International
Puerto Rico	Puerto Rico & Virgin Islands	Clarke 1866
Qatar National	Qatar	International
Qornoq	South Greenland	International
Reunion	Mascarene Island	International
Rikets Triangulering 1990 (RT 90)	Sweden	Bessel
Santo (DOS)	Espirito Santo Island	International
Santo (DOS)	Espirito Santo Island International São Braz São Miguel, Santa Maria Islands (Azores)	International
Schwarzeck	Namibia	Mod. Bessel 1841
South American 1969	Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Guyana, Paraguay, Peru, Venezuela, Trinidad, and Tobago	So. American 1969
South Asia	Singapore	Modified Fischer 1960
Southeast Base	Porto Santo/Madeira Islands	International
Southwest Base	Faial, Graciosa, Pico, Sao Jorge, Terceira (Azores)	International
Timbalai 1948	Brunei and East Malaysia (Sarawak and Sabah)	Everest
Tokyo	Japan, Korea, Okinawa	Bessel 1841
Viti Levu 1916	Viti Levu Island (Fiji)	Clarke 1880
Wake-Eniwetok 1960	Marshall Islands	Hough
World Geodetic System 1960 (WGS 60)	Worldwide	WGS 60

Table E.1 Datums Used All Over The World and Areas and Ellipsoids of Datums  
(continued)

World Geodetic System 1966 (WGS 66)	Worldwide	WGS 66
World Geodetic System 1972 (WGS 72)	Worldwide	WGS 72
World Geodetic System 1984 (WGS 84)	Worldwide	WGS 84

# APPENDIX-F

## NSTMSS INSTALLATION MANUAL (Revised)

### 1. Installation

#### 1.1. Required Tools

- SGI OpenGL Performer 3.1.1
- Visual Studio .NET 2003 Integrated Development Environment
- GLUT 3.7.6 Library
- DMSO RTI1.3NG-V6

#### 1.2. SGI OpenGL Performer Installation

Double click the OpenGL Performer 3.1.1 executable to run the setup program.

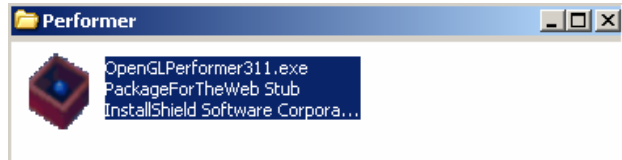


Figure 2. OpenGL Performer 3.1.1 Installation Package

The welcome screen below will appear. Click Next to continue setup.

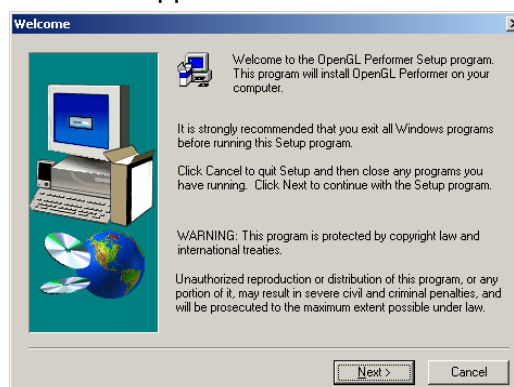


Figure 3. OpenGL Performer 3.1.1 Welcome Screen



After clicking Next button Software Serial Number screen will ask for a serial number. Leave the field blank and click Next button.

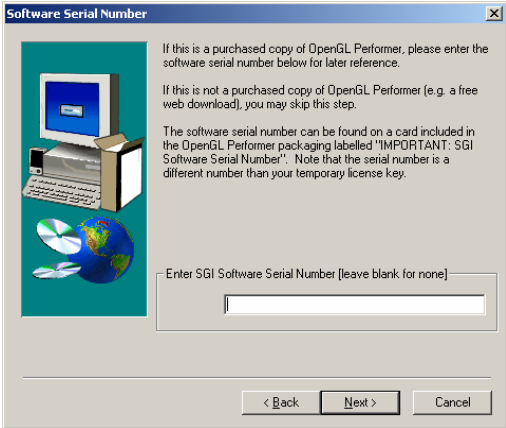


Figure 4. OpenGL Performer Software Serial Number Screen

Default installation folder for the OpenGL Performer is **C:\Program Files\Silicon Graphics\OpenGL Performer**. If you want to change the default installation folder from the Choose Destination Location screen, first click Browse button and then set the custom installation folder. Click Next button.

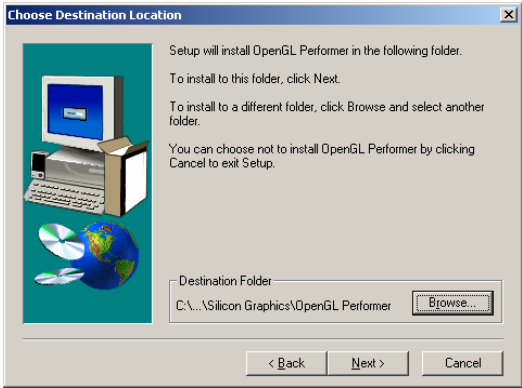


Figure 5. OpenGL Performer Destination Location Screen

Select typical from the Setup Type screen and click Next Button.

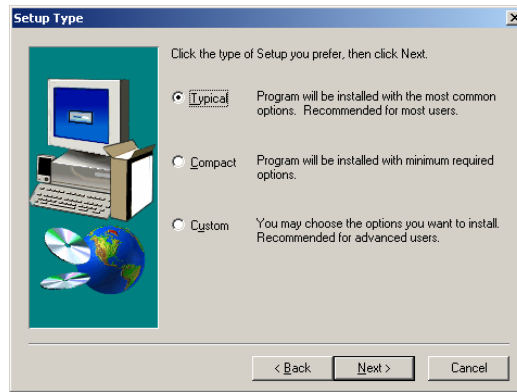


Figure 6. OpenGL Performer Setup Type Screen

Click Next button on the Select Program Folder screen. Setup will create program icons to default folder. (If you want to change program folder just type it to the Program Folders field before clicking Next button)

Setup program will start copying the files to your installation folder. After the file copy finished Setup Complete screen will appear. Click Finish button to finish installation.

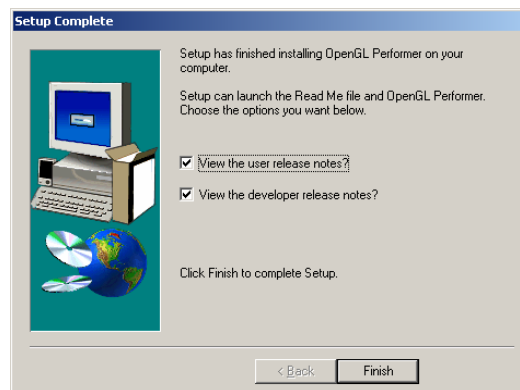


Figure 7. OpenGL Performer Setup Complete Screen

Now setup is completed. You must restart your computer in order to use the program for the first time. Select **“Yes, I want to start my computer now”** option from the Setup Complete screen and click Finish button. Now your computer will restart and then OpenGL Performer will be ready for the first use.

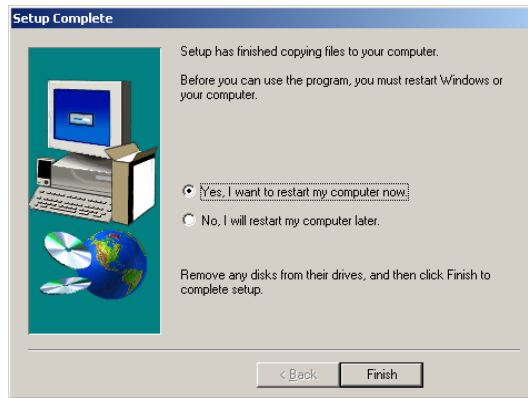


Figure 8. OpenGL Performer Setup Complete Restart Screen

### 1.3.GLUT 3.7.6 Installation

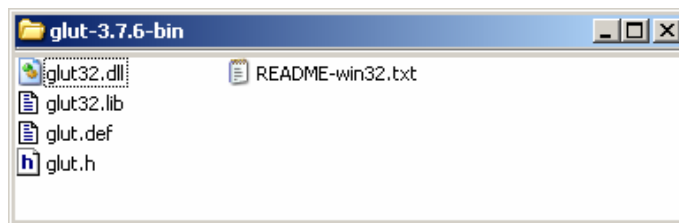


Figure 9. GLUT 3.7.6 Files

- Copy “**glut32.lib**” and “**glut32.dll**” files to “%OpenGL Performer Installation Directory%\Lib” directory.
- Copy “**glut.h**” to “%OpenGL Performer Installation Directory%\Include\Performer” directory.

### 1.4.DMSO RTI1.3NG-V6 Installation

Double click DMSO RTI-1.3NGv6 executable to run the setup program.



Figure 10. DMSO RTI1.3NG-V6 Installation Package

The welcome screen below will appear. Click Next button to continue setup.

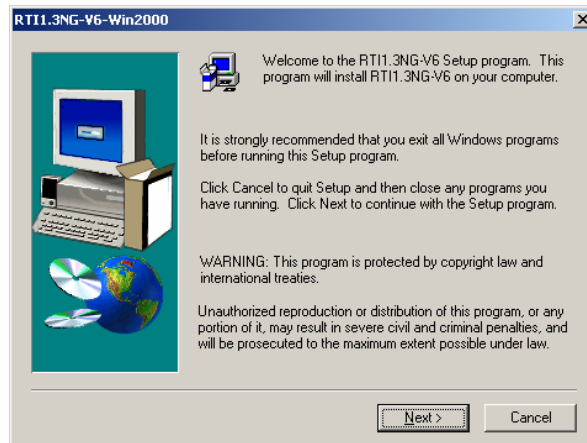


Figure 11. DMSO RTI1.3NG-V6 Welcome Screen

Default installation folder for the DMSO RTI1.3NG-V6 is **C:\Program Files\DMSO\RTI1.3NG-V6**. If you want to change the default installation folder from the Destination Folder screen click Browse button and set the custom installation folder. Then click Next button.

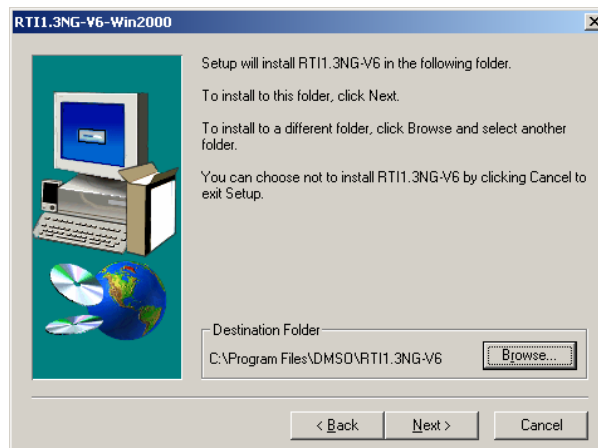


Figure 12. DMSO RTI1.3NG-V6 Destination Folder Screen

Setup program will start copying the files to your installation folder. After the file copy finished Setup Complete screen will appear. Click Finish button to finish installation.

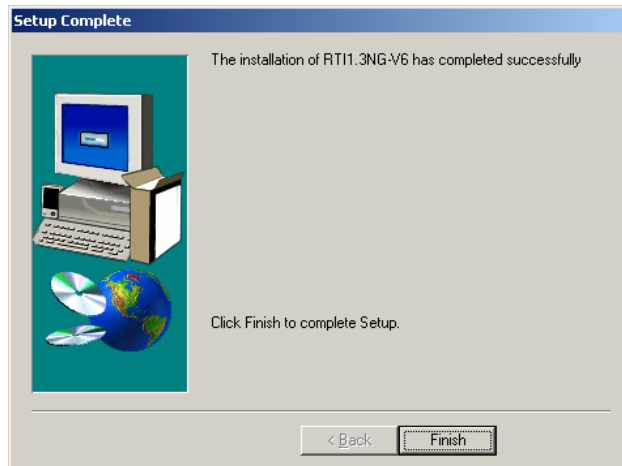


Figure 13. DMSO RTI1.3NG-V6 Setup Complete Screen

Now setup is completed. And DMSO RTI1.3NG-V6 is ready for use.

## 1.5.NSTMSS Installation Steps

Run the setup.exe program. Program installer will check that NSTMSS is already installed or not. If a current version of NSTMSS installation is detected, installer will ask either to remove it or to repair it. If previous version of NSTMSS installation is detected, then installer will uninstall it.

Then NSTMSS Welcome screens will appear. Click Next buttons on both screens.

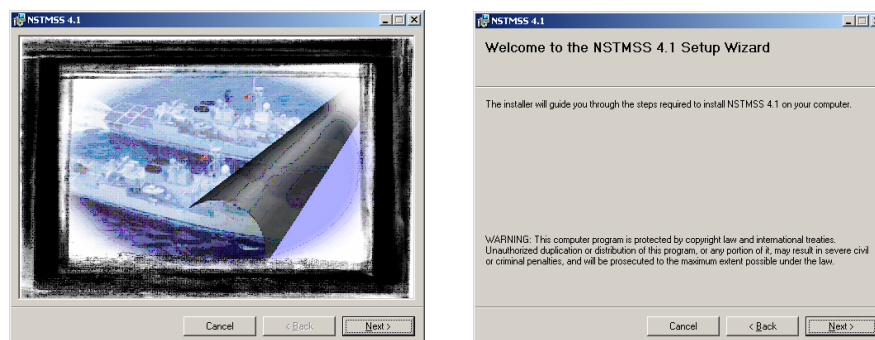


Figure 14. NSTMSS Welcome Screens

After the welcome screens, installer will show the license agreement. If agreed, select I agree radio button and then click next button.

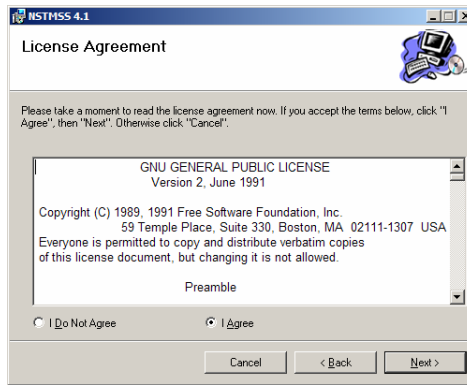


Figure 15. NSTMSS License Agreement Screen

An information screen will remember the user about the software dependencies. If these software runtimes are not installed prior to NSTMSS, NSTMSS federates will not run.

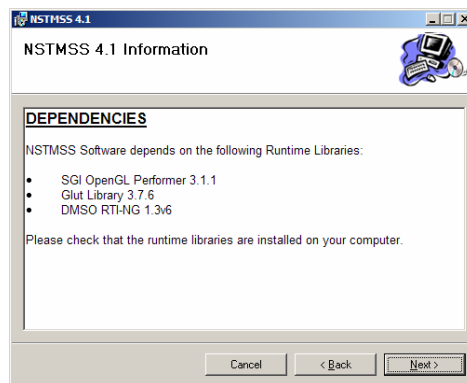


Figure 16. NSTMSS Software Dependencies Screen

After software dependencies screen, select the installation folder and click Next.

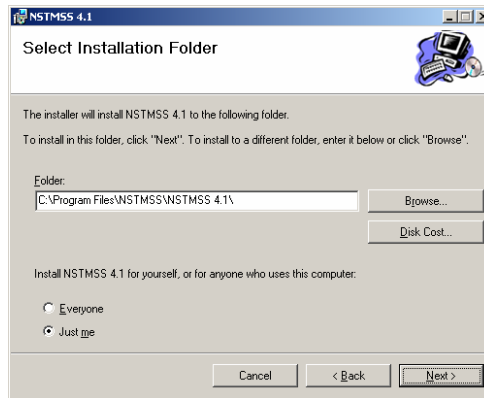


Figure 17. NSTMSS Select Installation Folder Screen

Now the installer is ready to install NSTMSS. Click Next to confirm installation.

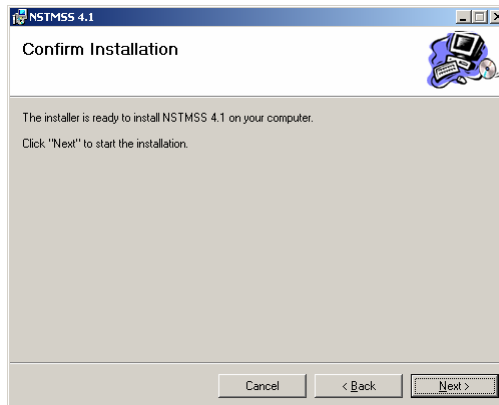


Figure 18. NSTMSS Confirm Installation Screen

Then, the installation will begin. Wait until the installation complete message.

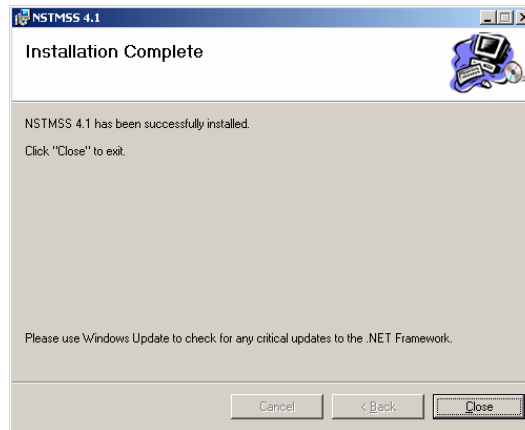


Figure 19. NSTMSS Installation Complete Screen

Click Close button to finish installation.

## 2. Development Environment Configuration

---

### 2.1. SGI OpenGL Performer License Configuration

After installation, copy the license.dat file to Performer installation folder.

### 2.2. DMSO RTI1.3NG-V6 Configuration

#### 2.2.1. Host Configuration

All the NSTMSS federates and “rtiexec” can be run on a network or a single host without a network.

To run on a single host, first Microsoft Loopback Network Adapter must be installed (see Section 2.2).

Be sure that there is link in network adapter. Check the network connectivity and be sure that all the hosts can ping each other if it runs on a network.

Before running the “rtiexec” program, “RTI.rid” configuration file must be configured. Enter the IP address and the port of server to the “RTI.rid” file where the “rtiexec” program you want to execute.

If you want to run it on a single host, after installing Microsoft Loopback Network Adapter you must do the following. From **Start>Run** type “cmd” and click **OK**. **Microsoft Command Prompt** will appear. Type “ipconfig” and press **Enter**. You will see the configured IP address for Microsoft Loopback Network Adapter Local Area Connection. Enter the configured IP address and an unused custom port into “RTI.rid” file.



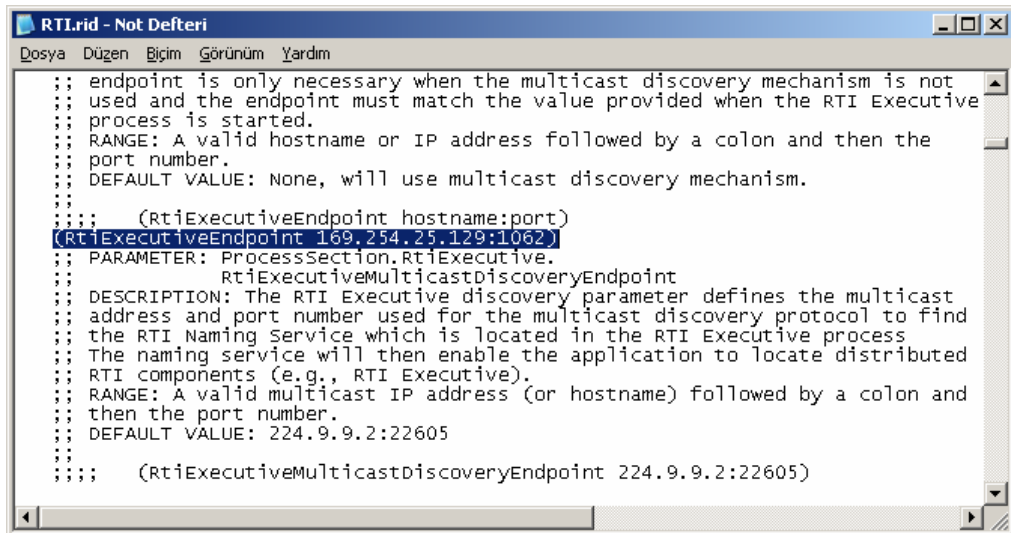


Figure 20. RTI.rid Configuration File

From **Start>Run** type “**cmd**” and click **OK**. Windows Command Prompt will appear. Run the “**rtiexec**” program first by typing “**rtiexec –endpoint hostname:port**” and press **Enter** .Then federates can be run. The order is not important.

## 2.2.2.RTI Environment Configuration

Add new environment variables:

RTI_HOME	RTI Home Directory
RTI_BUILD_TYPE	RTI distribution Type (Win2000-VC6)

Add existing PATH variable:

- ❖ “the folder of RTI DLLs”
- ❖ %RTI\_HOME%\%RTI\_BUILD\_TYPE%\bin

## 2.2.3.Microsoft Loopback Adapter Configuration

From “**Start>Settings>Control Panel**” double click Add New Hardware icon. Add New Hardware Wizard will appear. Click Next button. Wizard will ask you “**Did you connect your device to your computer?**”. Select “**Yes, I connected**” option and click Next button. Then wizard will show you a list of hardware installed in your computer. Scroll at the bottom of the list and select “**Add New Hardware**” from the list. Click Next. From the next screen, wizard will ask you what to do. Select “**I will choose from the List**” option and click Next. Then you will see a list of hardware types. Select “**Network Adapters**” from the list and click Next. You will see a list of network adaptor vendors. Select “**Microsoft**” from the vendor list and select “**Microsoft Loopback Adapter**” from the “**Network Adapters**” list and click Next 2

times. Click Finish button. Microsoft Loopback Adapter is now installed and can be seen on the task bar.

## 2.3. Visual Studio .NET 2003 Configuration

### 2.3.1. General Settings

- Create a C++ Win32 project
- In **Tools/Options/Projects/VC++ Directories**, Add:

Include:

Performer\_Home\include

Performer\_Home\include\performer

%RTI\_HOME%\%RTI\_BUILD\_TYPE%\include

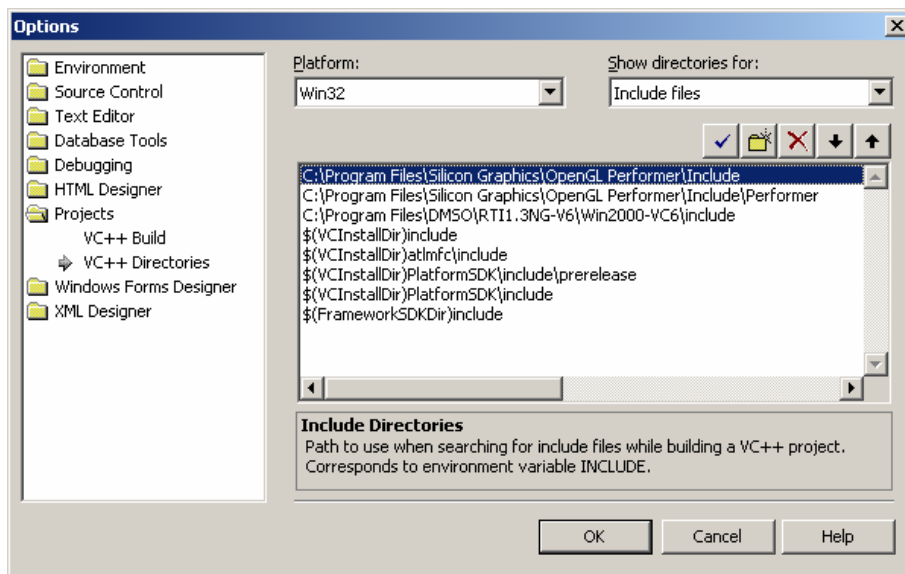


Figure 21. Visual Studio .NET Include Files Screen

Lib:

Performer\_Home\lib

Performer\_Home\lib\debug

%RTI\_HOME%\%RTI\_BUILD\_TYPE%\lib

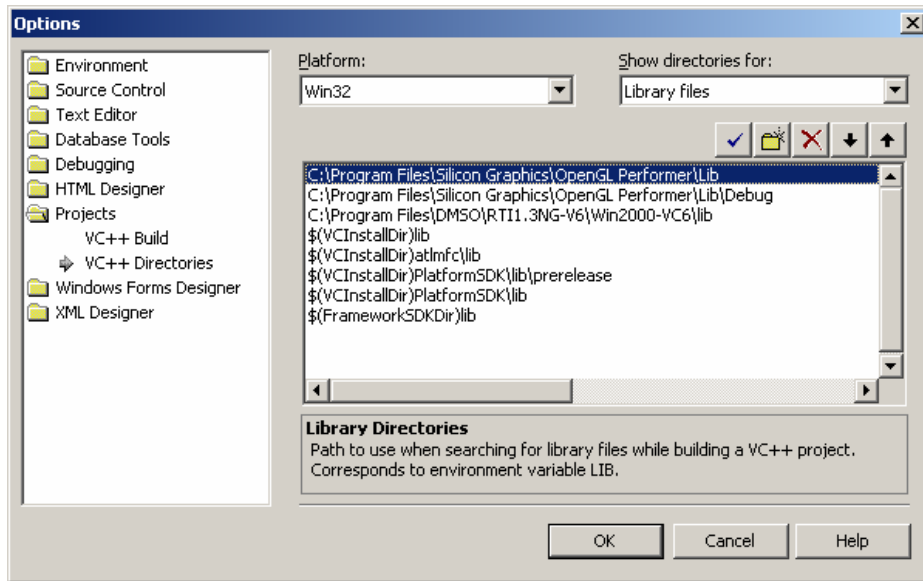


Figure 22. Visual Studio .NET Library Files Screen

- In **Project Properties/Linker/Input/Additional Dependencies** see there are

Libpf.lib, libpfui.lib, libpfdu-util.lib, opengl32.lib, glu32.lib

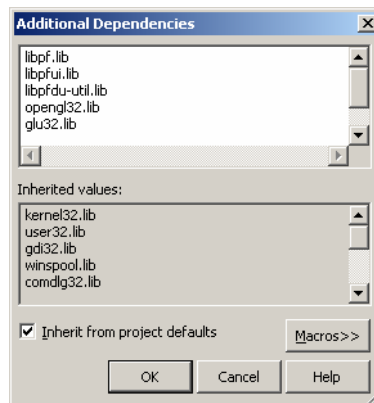


Figure 23. Additional Dependencies Screen

In **Object/Library Modules**, Add “libRTI-NG.lib” and “libFedTime.lib” or Add “libRTI-NGd.lib” and “libFedTimed.lib” for debug

- Change the “%RTI\_HOME%\%RTI\_BUILD\_TYPE%\include\RTI.hh” as shown below

```
// This was modified for the RTI 1.3NG to allow the use the
// Standard C++ fstream
// header file or to use of the legacy fstream.h header file.
// The issue concerns whether ostream is in the global
// namespace or in namespace std.
//
// #ifdef RTI_USES_STD_FSTREAM
```

```

#include <fstream>
#define RTI_STD std
//#else
//#include <fstream.h>
//#define RTI_STD /* nothing */
//#endif

```

- Check RTTI (Runtime Type Info) compile flag is enabled.

## 2.3.2. EnviFd Specific Settings

EnviFd is developed using Managed C++ .NET Windows Forms Classes. Therefore, additional settings are needed. Insert RTI library files to the linker input additional dependencies tab for debug and release configurations.

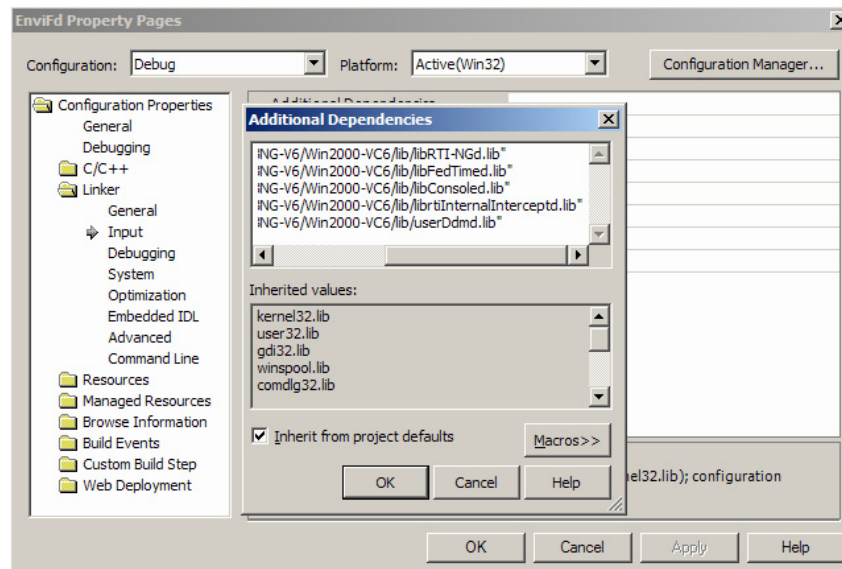


Figure 24. EnviFd Project Settings