A Framework To Develop An Interactive Web Database For Delivery Of Water Resources Field Data Over The Internet

Swarna Pujari
University of South Florida

Follow this and additional works at: https://scholarcommons.usf.edu/etd

Part of the American Studies Commons

Scholar Commons Citation
Pujari, Swarna, "A Framework To Develop An Interactive Web Database For Delivery Of Water Resources Field Data Over The Internet" (2005). Graduate Theses and Dissertations. https://scholarcommons.usf.edu/etd/824

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
A Framework To Develop An Interactive Web Database For Delivery Of

Water Resources Field Data Over The Internet

by

Swarna Pujari

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
Department of Civil and Environmental Engineering
College of Engineering
University of South Florida

Major Professor: Mark Ross, Ph.D.
Kaushal Chari, Ph.D.
Kenneth Trout, Ph.D.
Paul Zandbergen, Ph.D.

Date of Approval:
July 19, 2005

Keywords: ArcIMS, Access Database, J2EE Technologies,
Linear Correlation, Web GIS, XML

© Copyright 2005, Swarna Pujari
DEDICATION

This work is dedicated to Mrs. and Mr. Chiranjeevi Rao. Thank you for all the support that you have given me in my academic and professional pursuits.
ACKNOWLEDGEMENTS

I would like to thank Dr. Kenneth Trout, Dr. Ahmed Said and Nirjhar Shah for their frequent assistance in this research, and for their advice. I extend special gratitude for Dr. Mark Ross who had given me the opportunity to be part of the elite CMHAS group and for his constant guidance, commitment to quality research and importantly for the way he keeps people at ease when talking to him. Thank you to Dr. Kaushal Chari for his support and his willingness to help. I would like to acknowledge Dr. Paul Zandbergen for meeting with me and providing helpful insights which paved the path for the entire work. Finally I would like to acknowledge some of friends Srinivas, Gazal and Vamsi for their help and encouragement.
TABLE OF CONTENTS

LIST OF TABLES........................................................................................................iv

LIST OF FIGURES....................................................................................................v

ABSTRACT.................................................................................................................vii

CHAPTER 1. INTRODUCTION....................................................................................1
  1.1 Background.........................................................................................................1
  1.2 Definition of Problem..........................................................................................2
  1.3 Objectives and Scope..........................................................................................3
  1.4 Need for an Interactive Environment for Hydrologic Databases over the Internet.................................................................................................................4

CHAPTER 2. REVIEW OF EXISTING LITERATURE.................................................6
  2.1 GIS on the Web....................................................................................................6
  2.2 ArcIMS Solution for Flood Risk Lookup.............................................................9
  2.3 ArcIMS Solution for Transportation.....................................................................10
  2.4 ArcIMS Solution for Crime Mapping System......................................................11
  2.5 ArcIMS Website for Displaying Information about Emergency Facilities.........12

CHAPTER 3. MATERIALS AND METHODOLOGY.................................................13
  3.1 ArcIMS Architecture..........................................................................................13
    3.1.1 Components Needed to Support ArcIMS......................................................14
      3.1.1.1 Web Server..............................................................................................15
      3.1.1.2 JavaVM....................................................................................................15
      3.1.1.3 Servlet Engine.........................................................................................15
    3.1.2 Arc IMS Components in the Business Logic Tier.........................................16
    3.1.3 Managing an ArcIMS Site............................................................................16
REFERENCES.........................................................................................................................53

APPENDICES......................................................................................................................55

Appendix A: Servlet to Query and Download Data from the Database..............56
Appendix B: Visual Basic Code for Linear Correlation Technique.................67
LIST OF TABLES

Table 1. Table Depicting the Eight Different Ways a Dataset Containing Data for Three Variables Can Exist………………………………………………………………32
Table 2. Sample Dataset Used to Explain Methodology of Data Imputation using Simple Ratio Correlation Method………………………………………………………..34
Table 3. Table Showing Artificially Removed Data Points and the Predicted Values for Well USF3…………………………………………………………………………………..47
Table 4. Table Showing the Comparison Between the Removed and the Predicted Values in Case of Two Missing Values (PS43 and USF3) in a Record………………….48
Table 5. Statistics Generated by the F-test……………………………………………………….49
LIST OF FIGURES

Figure 1. Site Location on the Map.................................................................2
Figure 2. Architecture of ArcIMS 9.1...............................................................13
Figure 3. Components Outside of ArcIMS.......................................................14
Figure 4. Components of Business Logic Tier of the ArcIMS............................16
Figure 5. Screenshot of the Database Showing all the Tables............................19
Figure 6. Screenshot of the Data Table in the Database.................................20
Figure 7. Screenshot of the DataCollection Table..........................................21
Figure 8. Screenshot of the DataType Table..................................................23
Figure 9. Screenshot of the Relationships Table.............................................24
Figure 10. The Hierarchy of the ArcXML Elements.........................................25
Figure 11. Figure Illustrating the Frame Structure Chosen for the ArcIMS Website.....27
Figure 12. Plot for the Sensor 6 of the PS40 Well.........................................37
Figure 13. The Toolbar that is seen on the “Thesis” Website..............................39
Figure 14. Screenshot of the Home Page.......................................................42
Figure 15. Screenshot of the ‘Identify’ Tool in Use..........................................43
Figure 16. Screenshot of the ‘Query’ Tool in Use...........................................43
Figure 17. Screenshot of the Results of the ‘Query’ Tool..................................44
Figure 18. Screenshot of the ‘Find’ Tool in Use.............................................44
Figure 19. Screenshot of the Webpage that is Displayed on Clicking Well PS41…..45
A FRAMEWORK TO DEVELOP AN INTERACTIVE WEB DATABASE FOR DELIVERY OF WATER RESOURCES FIELD DATA OVER THE INTERNET

Swarna Pujari

ABSTRACT

The objective of the research is to develop a user friendly framework for an internet GIS (Geographic Information Systems) application. The study aims at providing a map with GIS capabilities without requiring the clients (users of the website) having to install ArcGIS (a product of ESRI) software on their personal computers along with the facility to download hydrological data. It also aims at providing a methodology to synthesize data in case of data gaps. High resolution data was collected in a small watershed in west Central Florida to measure hydrologic storages and fluxes during 2001-2004. Meteorological, surface and groundwater data were collected at 5 to 10 minute intervals. A watershed database was constructed using Microsoft Access and was normalized to 3NF (third Normalization Form) for easy update. The databases available do not involve user-friendly tools to map data collection locations or to facilitate interactive delivery of data. In many fields like hydrology, it is important to provide spatial location of the data points with the data. In addition there is also a need for one platform wherein various user communities (students, planners, hydrologists etc) can be served. Thus there is a need for an online user-friendly environment, which allows for interactive delivery of data along with mapping and spatial analysis. To address this need
an ArcIMS website was constructed. It includes ortho-photography of the site, which can be overlaid with the shapefile of the data locations thus giving the user reference orientation of the data locations. The user is allowed to download data in the form of text files based on the required temporal resolution and date ranges. Furthermore, users are provided with valuable data sets to parameterize or test hydrologic models applied to west-central Florida conditions. The website has a feature to generate synthetic data using linear correlation in cases where data are unavailable for the requested period of time. The website also serves the dual role of providing accessible surface and groundwater information to the public. It can also be used as a desktop geographic work tool for engineers, planners and developers, to help better understand the surface water, vadose zone and groundwater interaction. Hence this website is useful not only for professional hydrologists but also for graduate research.

Key words: ArcIMS environment, Data Synthesis, Linear correlation method
CHAPTER 1. INTRODUCTION

1.1 Background

Two local agencies, Tampa Bay Water, the quasi-governmental water wholesaler for west-central Florida and the Southwest Florida Water Management District, the state agency responsible for the regulation of surface and groundwater resources for west-central Florida, funded the University of South Florida for a high-resolution and a more comprehensive data collection to understand and predict various hydrological processes.

Data was collected for this study at a small watershed in west-central Florida (Figure 1). The study site has a high water table, a shallow surficial aquifer. The site is also characterized by different land types like “Forest” (the land that was covered with trees) and “Grass” (the land that was covered with grass).

Meteorological, surface water and groundwater data was collected at 5 to 10-minute intervals from the summer of 2001 to the summer of 2004. Meteorological data, collected at 5-minute intervals, include rainfall (tipping bucket), temperature, barometric pressure, solar radiation, wind speed and wind direction, humidity and pan evaporation. Surface water data collected were stream flow into and out of the watershed (10-minute intervals), runoff (5-minute intervals), surficial aquifer water levels and soil moisture data was also collected.
1.2 Definition of Problem

There are many existing websites from which hydrologic data from all over the United States can be downloaded and used. The researchers and decision-makers usually come across missing data when trying to download hydrologic data pertaining to a specific location. In such a situation, the website users would definitely appreciate a Geographic Information System (GIS) map that could allow them to synthesize data to replace missing or bad data. ESRI has software named “ArcIMS” for making web based GIS applications. Hence, the main idea of this work is to provide an ArcIMS website capable of doing rudimentary viewing, using the functionality such as zoom, pan, legend

Figure 1. Site Location on the Map of Florida.
controls, identification of features, printing of simple maps, querying of GIS data and also enabling various database interactions. Connecting GIS data with tabular and time-series data has limitless possibilities for water resources management. The idea of a “Virtual” map offers the potential to make the hydrologic database more accessible and current to all kind of users.

1.3 Objectives and Scope

The purpose of this research is to develop an ArcIMS website through which the delivery of both map and data over the Internet is more accessible, current and convenient. ArcIMS allows users to integrate local data sources with Internet data sources for display, query and analysis in a user-friendly web browser.

Creating and maintaining a GIS would be less useful if it could not be used by everyone and from any place. The justification for its existence is the value derived from its use. The decision to concentrate on a web-accessible GIS site was the need for a widely accessible system with a common user interface and no software costs for the end user. The results of this research offers the hydrologic community a better way of looking at and using hydrologic data.

This work is divided into five chapters. Chapter 1 reviews background information for having a web based GIS, coupling hydrological database with a “Virtual” map. Chapter 2 discusses in detail some of the past studies of coupling map with data using in the ArcIMS environment. Chapter 3 describes general methodology used in this research. Chapter 4 describes actual results of the ArcIMS application. Finally, Chapter 5
discusses factors affecting the performance of the method and summarizes the important conclusions and recommendations for future work with this application.

1.4 Need for an Interactive Environment for the Hydrological Databases over the Internet

The World Wide Web (WWW) serves as a means to provide large amounts of information to planners, decision makers and other potential users. In the past, online databases available did not provide user-friendly tools that facilitated, interactive delivery of data along with maps. So the project aims at developing a website with a GIS map of the hydrological parameters and also allowing downloading the data for the hydrological parameters available on the map. This will prove useful not only for professional hydrologists but also for people involved in academia. This will also help the users to synthetically generate data in case of missing/bad data based on a linear correlation method that uses alternate hydrological stations’ data.

ArcIMS allows users to integrate local data sources with Internet data sources for display, query and analysis in a user-friendly web browser. ArcIMS uses java based components that communicate with and access, its services. The platform used to communicate with the ArcIMS application server is Java Server Pages (Sun Developer Network website, http://java.sun.com/products/jsp, last referred on 6/10/2005). JSP is chosen due to the following reasons

- Simple yet robust
- Platform independent
- Multithreaded and gives high performance
• Object oriented, producing reusable object oriented components

• Network oriented

To achieve this objective, a two-tier client server architecture is used. On the server side, the JSP tags programmatically establish communication with an ArcIMS application server via HTTP connection. The application server on receiving the request processes it and returns an appropriate response based on which the application can react accordingly. On the client side the java viewer/browser uses Java 2 Applet for user interface components such as the map display, scale bar and legend. The user is provided with great flexibility in choosing how the data is presented. This is achieved by a variety of complex queries that retrieve the required data from the database. In spite of all the flexibility offered, the user is not allowed to directly manipulate the original database or its contents in any manner.

To conclude, the approach used in the project provides a new perspective for viewing and working with hydrological models.
CHAPTER 2. REVIEW OF EXISTING LITERATURE

A review is provided in this section about the origin of the web-based GIS and ArcIMS solutions for some of the problems in the field of science are discussed.

2.1 GIS on the Web

The Internet has undergone explosive growth during the past decades. The Internet began to be used extensively by the public, business, and society in 1995 (Castells 2001). In 1995, there were about 16 million users worldwide. Internet usage in 2000 was around 400 million people. The usage is currently around 888 million (InternetWorldStats.com, 2005).

Internet was envisioned in the 1960’s as a U.S. military “fail-safe” network. The first network was based on packet switching communications technique and was called ARPANET as it was developed by the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense.

ARPANET was developed and promoted initially by computer scientists working at research institutions and universities. The goal of the development was to generate a new kind of computer-based, digital communications network for non-military and non-commercial reasons. ARPANET became operational in 1975 when it was transferred to
the Defense Communication Agency. The number of Internet host computers reached 10,000 in 1987 (Branscomb 2003).

Internet functions on three basic principles (Castells 2001):

- Decentralized network structure, with no single “headquarters” that control the whole system.
- Distributed computing capabilities throughout many nodes of the network.
- Redundancy of functions and control of the network to minimize risk of disruption in service.

The World Wide Web is one of the major and also rapidly developing applications of the Internet. The web is the collection of all information available anywhere within the Internet system. The technological basis for the Web is Hypertext Transfer Transport Protocol (HTTP), which establishes communication between a browser (client) and a Web server (server) computer. Hypertext Markup Language (HTML) forms the basis for writing documents and creating links on the web. The locations of the files and other resources on remote server computers are identified by a Uniform Resource Locator (URL). Client software opens the window to the web. Web browsers allow the user to move through the Web with text, graphic, video, and audio capabilities.

As Internet became a widely used tool, GIS over the Internet became an important component of Enterprise GIS implementation. Some of the reasons necessitating the Internet GIS is summarized as follows (Goddard et al, 2002)
• The ability to extend the audience of GIS to a large circle of users so that a better return on investment could be possible through the collaborative use of data, services, and applications for multi-user, multi-agency, cooperative work.

• The enhancement in customer satisfaction by providing automated information and related services based on visually referenced/spatially enabled databases.

• Communicating dynamic/interactive spatial information and services to the users (i.e., real-time information dissemination, integration, communication)

• To support daily or emergency operations of the organizations in a timely and cost effective manner.

• Incorporation of complex spatial/statistical analysis functionalities, modeling and decision support (i.e., data-to-models and data-to-interpretation and others).

The Web GIS technology paved path for better ways to share and display spatial information over the internet. The Web offers user interaction, so that a distant user can access, manipulate, and display geographic databases from a GIS server computer.

The Xerox web site, established in 1993 (mapweb.parc.xerox.com/map), is one of the first implementations of GIS over the web. The site had used pre-created image maps for each case scenario for the users to find the particular html page containing the map for display.

GIS capabilities on the World Wide Web, using the middleware program called Internet Map Server (IMS), was first commercially launched by Environmental Systems Research Institute Inc. (ESRI). Internet Map Server was an efficient way to make real-time GIS products available to a large population at minimum cost.
ArcIMS is Internet-based GIS launched by ESRI to allow for authoring and publishing a wide range of GIS maps, data and applications to users over the web. ArcIMS accommodates client and server technologies. It extends the web to serve spatial data and limited functionalities. ArcIMS could also work with a wide range of clients including desktop and mobile devices. The ArcIMS technology thus helps in delivery of GIS maps over the internet, integrating data from multiple sources, scaling of the system to meet the user needs.

2.2 ArcIMS Solution for Flood Risk Lookup.

After the devastation of Tropical storm “Allison”, the Federal emergency Management Agency and the Harris County Flood Control District joined forces to develop new flood insurance rate maps for Harris county, Texas (Jacob Spenn et al, 2004). These maps were made easily available to the public via the Internet. The public had a strong desire to know their specific flood risk and make comparisons to previous versions of these special flood hazard areas. In order to provide the data to the public in a timely and widely accessible manner, the maps were published via the Internet on an easy-to-use web based ESRI ArcIMS platform.

The users could view the new floodplains and their current flood risk by locating properties by street address, zip code, or select by area. Apart from that the users can also set parameters like which layers of the GIS map are they interested in viewing during the request and at what pre-determined scale, for which a map output would be delivered.
2.3 ArcIMS Solution for Transportation

An interactive ArcIMS website was developed for the city of Boulder, Colorado transportation department (Laurence Ferguson et al, 2004). The ArcIMS website displays both the existing and proposed transportation geographic features and attributes (bike, pedestrian, transit, and auto). The website served the dual role of providing accessible transportation information to the public and also functioning as a desktop geographic work tool for engineers, planners and developers, to help them better understand the city’s transportation system, policy plans and infrastructure needs.

A major aspect of the project was the creation of both existing and proposed enterprise transportation GIS data (bike, street, and transit) to make it compatible with the ArcIMS web based application. This was accomplished in ArcGIS by creating a series of Oracle feature data sets and feature classes that were based on a data schema of unique ID numbers which were dependent on travel type (auto, bike, pedestrian and transit). Oracle-based feature data set was used to store GIS transportation data that was digitized and registered to the city’s aerial photography base map. Normalized data tables were created to store attribute information that pertain to proposed and existing transportation features (i.e. type, location, year built, cost). By linking this attribute information table to a transportation GIS feature, Transportation Master Plan (TMP) website visitors were able to query and then parse the responses to generate attractive web pages that displayed maps, PDF documents and photographs of proposed and existing transportation features. The significant advantage to the enterprise GIS user community that came out of the TMP was a single table, or feature class in GIS parlance, containing all transportation routes such as alleys, streets, main roads, trails, bikeways and green ways.
2.4 ArcIMS Solution for Crime Mapping System.

The Fairfax Crime Mapping Website was a joint venture between GIS division and police department at Fairfax County, Virginia (Gao et al, 2004). The purpose of the site was to

- Increase awareness: an interactive, user-driven web application would allow for the public to be more informed about crime in a particular area. It would also allow the public to be more aware about what is being done to address crime and traffic issues.

- Increase participation: an interactive, user-driven site could spark public support and activity to resolve issues or problems.

- Increase efficiency: a majority of the information listed in the goal has currently being provided by the Crime Analysis Unit. While important, this hampers the unit’s overall analytic function. An Internet site is an ideal means to reduce this burden for the Crime Analysis Unit.

The mapping system had three-tier architecture. A Microsoft SQL server database stored the crime incident data tables. The front end (user interface) comprised of a query interface and interactive mapping. The query interface and criteria specified by the user determined the mapping results. Thus the Fairfax crime mapping website was successful in providing general information about police service for an area based on the user entering a specific address.
2.5 ArcIMS Website for Displaying Information about Emergency Facilities.

TNRIS (Texas Natural Resources Information system) has designed a web interface and database design for collecting and displaying information about emergency facilities like evacuation shelters, including location, status, photographs, type of facility, etc. (Brent Porter et al, 2004). This application used ArcIMS, SQL server to obtain information about shelters for visualization and logistics support through a web based tabular interface and a custom ArcIMS application.

Thus, the application helped local, regional and state officials to make better decisions about where and when to open a shelter in the event of an emergency such as a hurricane.
CHAPTER 3. MATERIALS AND METHODOLOGY

3.1 ArcIMS Architecture

The software was designed to ease creation of maps, develop Web pages that communicated with the maps, and administer a Web mapping site. The software was also designed to be distributed across a network and to be scalable as the demand for maps increases. ArcIMS has a multi-tier architecture consisting of presentation, business logic, and data tiers. In addition, ArcIMS has a set of applications for managing a Web mapping site. The illustration below provides an overview of the ArcIMS architecture.

![Architecture of ArcIMS 9.1](image)

Figure 2. Architecture of ArcIMS 9.1
The ArcIMS architecture can be described as follows:

- The presentation tier includes the ArcIMS client viewers for accessing, viewing, and analyzing geographic data.
- The components in the business logic tier are used for handling requests and administering the ArcIMS site.
- The data tier includes all data sources available for use with ArcIMS.
- The ArcIMS site management applications provide access to components in the business logic tier for authoring maps, administering ArcIMS services, and designing Web sites.

3.1.1 Components Needed to Support ArcIMS

ArcIMS is an Internet product that works in a Java™ environment. For ArcIMS to run correctly, supporting components are needed that are not part of ArcIMS. This includes a Web server, JavaVM (Java Virtual Machine), and a servlet engine (see Figure 3). Thus ArcIMS website is a final outcome of all the above discussed components working along with ArcIMS.

Figure 3. Components Outside of ArcIMS (adapted from ESRI website, www.esri.com)
3.1.1.1 Web Server:

A Web server handles requests from a client using Hyper Text Transfer Protocol (HTTP). The Web server forwards a request to the appropriate application and sends a response back to the requesting client. A Web server is not included with ArcIMS.

3.1.1.2 JavaVM:

Many of the ArcIMS components are Java components and require a JavaVM (ArcIMS manual), which provides the basic application programming interface (API) for running these applications. The JavaVM is included in either the Java Runtime Environment (JRE) or Java Developer Kit (JDK). ArcIMS requires a JRE, which is installed if it is not already on the machine. It is possible to have more than one JavaVM on a machine, and ArcIMS can be made to use an existing JavaVM as long as it is compatible.

3.1.1.3 Servlet Engine:

Servlets are Java programs written in a web application such that they would query the database and return results as required to the client viewer. Therefore, ArcIMS requires a servlet engine. A servlet engine is an extension to the JavaVM and provides support for servlets through a servlet API. The servlet engine plugs into a Web server and provides the link between the JavaVM and the Web server. A servlet engine is not included with ArcIMS.
3.1.2 ArcIMS Components in the Business Logic Tier.

An ArcIMS site is made of components in the business logic tier along with data in the data tier. This section discusses the business logic tier, and the data tier. The ArcIMS business logic tier contains the components needed to run services and process requests and responses.

![Figure 4. Components of Business Logic Tier of the ArcIMS](image)

The components include the Application Server Connectors, the ArcIMS Application Server, and the ArcIMS Spatial Server. The framework also requires the Web server, JavaVM, and the servlet engine. A shown in the Figure 4 an ArcIMS request is first handled by the Web server, passed through one of the connectors, and then forwarded to the ArcIMS Application Server. The Application Server, in turn, dispatches the request to an ArcIMS Spatial Server for processing.

3.1.3 Managing an ArcIMS Site

As previously mentioned, the business logic and data tiers make up an ArcIMS site. To access components in the business logic tier, ArcIMS provides a set of management applications. Data in the data tier, although accessed by ArcIMS, is managed using other tools such as ArcGIS. ArcIMS management consists of three tasks:
3.1.3.1 Authoring Map Configuration Files: (Implementation tools used are ArcIMS Author, XML editor.)

The first management task is to generate map configuration files. These files are written in ArcXML and are the input to ArcIMS services. ArcIMS Author is one tool that can be used to create a map configuration file. It can access shape files and some images formats. Once the layers are established, ArcIMS Author is used to define symbology, set scale dependencies, and define other mapping parameters. Map configuration files can also be created and edited using an XML editor. The procedure for creating map configuration files for ArcMap Image Services is the same as for Image or Feature Services. The authoring tool, however, is ArcMap, which is used to define symbology, set scale dependencies, and define other mapping parameters. The output file is in a binary format and has “.axl” extension.

3.1.3.2 Publishing and Administering ArcIMS Services: (Implementation tools used is ArcIMS Administrator)

A second management task is to publish and administer services. Users can add, start, stop, and delete services using ArcIMS Administrator. The input to a service is the map configuration files already described.

3.1.3.3 Designing Web Pages: (Implementation tool used is ArcIMS Designer.)

The final management task is to generate a Web site using ArcIMS Designer. ArcIMS Designer leads the user through a series of panels for selecting which services to use, which page style to use, and which operations and functions will be available in a
client Web browser. Designer has three options: an HTML Viewer, a customizable Java Viewer, and a non customizable Java Viewer. The output from Designer is a group of HTML pages. The Web pages can be modified and enhanced to meet specific requirements of the web application.

3.1.4 The Presentation Tier, ArcIMS Client Viewer

The presentation tier consists of client viewers for users to access, view, and manipulate geographic data. A typical client viewer includes a map and some method for interacting with the map. The client viewer can be generated using ArcIMS Designer. Clients can make a request to a service residing at an ArcIMS site which then processes the request and sends back the results.

3.2 Database Properties

The project involves the collection of rainfall, groundwater elevations, runoff, stream flow, weather and ET data. All this data is uploaded into a relational database. The Database Management Software being used to maintain this relational database is the Microsoft Access 2002-2003 version. The database is regularly updated and modified as required.

3.2.1 Database Description

The database is divided into a number of tables. The advantage in doing this is the ease of expansion. After the original database creation, a new data category can be added without requiring that all existing applications be modified. Relationships and integrity
constraints have been defined between the tables. Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency. A screen shot showing the various tables being used is displayed in the Figure 5.

A screenshot showing the database is displayed in Figure 5.

Figure 5. Screenshot of the Database Showing all the Tables.

A brief description of the different tables used in the database follows below.

3.2.1.1 Data

This is the largest table and contains most of the information of the database (Figure 6). This table contains rainfall, weather, runoff, ET, streamflow and well elevation data. The table has the following columns:

- data collection ID
- date and time
- data value
- corrected data value (the data values for the well elevation have a correction applied so that correspond to NGVD values)

![Figure 6. Screenshot of the Data Table in the Database.](image)

This table has a many-to-one relationship with the “Data Collection” table which is described a little later. This means that every instance in the “Data” table has to be related to one particular instance in the “Data Collection” table.

3.2.1.2 DataCollection

This table contains all the description required for the instruments in the field. The table contains the information relating the data collection ID with the location ID and
type ID. The table also accounts for brief description about the instruments like the depth and elevation. It contains the following columns:

- ID
- Location ID
- Type ID
- Description
- Index
- Depth
- Elevation

Figure 7. Screenshot of the DataCollection Table.
This table has a one-to-many relationship with the “Data”, “Site Maintenance” and “Moisture Data” tables. That is, a particular instance in the “DataCollection” table may have many related instances in any of the above mentioned tables. It also has a many-to-one relationship with the “DataType” table. That is, all instances in the “DataCollection” table are related to one particular entry in the “DataType” table.

3.2.1.3 DataCorrection

This table contains the correction to be applied to the wells so that wells have an elevation that corresponds to the NGVD elevation. This table contains the following columns:

- data collection ID
- starting date
- amount of correction to be applied to the well data collected from the field. (The amount of correction to be applied is calculated with the help of manual readings collected from the field)

3.2.1.4 DataType

This table contains the relationship between the ID and the Data Type. It has only two columns:

- ID
- Type

A screen shot of the “DataType” table with a few of its rows is displayed in Figure 8.
Figure. 8 Screenshot of the DataType Table.

This table has a one-to-many relationship with the “DataCollection” table. That is, one instance in the “DataType” table may have more than one related instance in the “DataCollection” table.

3.2.1.5 Relationships

A number of relationships have been defined between the tables. These relationships are displayed in the screen shot shown in Figure 9. All instances of data collections can be classified into one of the categories specified in the “Data Type” table. Therefore, there exists a one-to-many relationship between the “Data Type” and “Data Collection” table. The location specified in the “Data Collection” table has to be one of the instances specified in the “Location” table. So, there is a one-to-many relationship between the “Location” and “Data Collection” table. Each instance of the “Data Collection” table has with it associated multiple instances of “SiteMaintanance” and
“MoistureData” tables. So there is a one-to-many relationship between the “DataCollection” table and “SiteMantainance” and “MoistureData” table. Integrity Constraints have also been defined. This ensures that changes made to the database by authorized users do not result in a loss of data consistency.

Figure 9. Screenshot of the Relationships Table
3.3 Customizing ArcIMS

3.3.1 ArcXML and its Structure

In ArcIMS, the type of Extensible Markup Language (XML) that is used is called ArcXML. XML allows developers to create their own set of elements that define established functionality. An XML parser is software that can read and interpret XML requests and then perform the correct action. Another way of authoring a map for the ArcIMS website would be by understanding the ArcXML and using it to alter the map configuration file to meet the website requirements. Thus some of the ArcXML elements and their hierarchy is defined below. The client requests are coded in ArcXML and parsed on the server computer. In addition, all server responses are returned in ArcXML format.

In reference to the Figure 10, the following is the ArcXML elements.

Figure 10. The Hierarchy of the ArcXML Elements.
• All ArcXML code begins with the ARCXML element. The sub elements of the ARCXML element are CONFIG, REQUEST, RESPONSE, or MARKUP, and define the type of the ArcXML statement.

• The CONFIG element indicates that it is a map configuration file, defining the map properties and data layers. The CONFIG element has MAP and ENVIRONMENT elements as sub elements.

• The ENVIRONMENT element defines the language being used for this viewer.

• The MAP element indicates that there is information on the definition of the map. The MAP element has three key subelements: PROPERTIES, WORKSPACES, and LAYER.

• The PROPERTIES element describes the general properties of the map. The WORKSPACES element lists the type and location of the data layers for the map. The LAYER elements define how each layer in the map will be drawn.

3.3.2 HTML Frame Structure

HTML frame provides the skeleton for the HTML viewer. Frames are defined within a html file called viewer.htm. (Refer to next page)
Figure 11. Figure Illustrating the Frame Structure Chosen for the ArcIMS Website.

The following HTML file obtains the above frame structure for the ArcIMS website.

```html
<FRAMESET ROWS="100,*,100">
  <FRAME name="top" src="top.htm">
  <FRAMESET COLS="200,*">
    <FRAME name="left" src="left.htm">
    <FRAME name="right" src="right.htm">
  </FRAMESET>
  <FRAME name="bottom" src="bottom.htm">
</FRAMESET>
```
FRAMESET elements control rows and columns. Each frameset contains one or more frames and each FRAME element references one HTML file. In the above HTML code the first FRAMESET element defines three rows. Below that element there are two subservient FRAME elements and one FRAMESET element, for a total of three. Each FRAME element takes a name attribute, which specifies how Web pages may refer to that frame, and an “src” attribute that indicates which HTML page is to fill the available space.

3.3.3 ArcIMSParam.js File

ArcIMSParam.js is a parameter file in the HTML viewer provided by the ArcIMS. The ArcIMSParam.js file has a vast scope for customization. Depending on website requirements the ArcIMSParam.js file can be modified. For the “Thesis” website the file was modified to allow for the addition of a new tool (DownloadData) in the toolbar and also to add hyperlink facility for the “Well Locations” Layer.

A simple description of how the new tool was added into the toolbar.

- Declaration of new variable called “useDownloadData” and initializing the variable to true.
- Creating a button for the tool in the toolbar. To do this the file called “toolbar.htm” was modified.
- Defining the functionality of the tool like what it has to do when the button is clicked. The functionality is defined in a file called “aimsClick.js” file.
• Providing the Actual definition of the function for the Download Data button in the file called “aimsQuery.js” file.

The hyperlink facility was provided to the “Well Location” layer in the following way:

• A new field was added to the attribute table of the “Well Location” layer to contain the urls of the pages of the website.
• In the “ArcIMSParam.js “ file the ActiveLayerIndex is set to the “Well Location “ layer.
• useHyperlink variable is set to true in the “ArcIMSParam.js” file.
• The Hyperlink layers array is set to contain the “Well Location” layer. The Hyperlink Fields array is set to contain the name of the field in the attribute table of the “Well Location” layer which contains the urls of the web pages.

The code added to all these files is provided for reference in the appendix.

3.4 Java Servlets and the HTML Files

Java Servlets are programs that run on a Web server and build Web pages. If the data being shown on the web pages does not change (static) then simple HTML web pages would be sufficient, but normally any web application would demand a dynamic web page.

Building Web pages on the fly is useful (and commonly done) for a number of reasons:
• The Web page is based on data submitted by the user.
• The data changes frequently. (ex. Weather Reports)
• The Web page uses information from corporate databases or other such sources.

To be a servlet, a class should extend HttpServlet and override doGet or doPost (or both), depending on whether the data is being sent by GET or by POST. These methods take two arguments: an HttpServletRequest and an HttpServletResponse. The HttpServletRequest has methods that let users find out about incoming information such as FORM data. The HttpServletResponse has methods that let users specify the HTTP response line (200, 404, etc.), response headers (Content-Type, Set-Cookie, etc.), and, most importantly, lets users obtain a PrintWriter used to send output back to the client. For the “Thesis” website the java servlets required to query the database were built in an IDE called “NetBeans”. The Web Server used to execute these servlets was Apache Tomcat 4.1 version. The java servlets written to query the “cmhas database” along with the html pages is provided in the appendix.

3.5 Synthesizing Data using Multiple Imputation Techniques

Imputation, the practice of ‘filling in’ missing data with plausible values, is an attractive approach to analyzing incomplete data. It apparently solves the missing-data problem at the beginning of the analysis. Most statistical packages do not handle missing data well. Cases with missing values are typically discarded, resulting in substantial loss of information and perhaps biasing the results in unpredictable ways.

One of the important aspects of this thesis is to explore different ways in which the imputation can be done. An important factor to be addressed here is the ease and
availability of the recommended data imputation techniques. Thus the use of commercially available software was avoided. Depending on the data distribution, thus two methods are suggested and described. The first method uses the concept of multivariate normality to impute the missing data while the second method is based on using the correlation of the similar land use sensors which are expected to behave in sync. The following section describe the two methods in detail

3.5.1 Use of Simple Ratio Correlation to Impute the Missing Data

This technique is less rigorous and based more on empiricism than statistics. It is well recognized that the for same precipitation the moisture at same depths for the similar land use will show more strong correlation in the values than the sensors at different depths or in different land covers. For this particular reservoir site the soil moisture sensors located at PS43, USF1 and USF3 all have a pasture grass as land cover and are at similar elevation. Statistical summary from the NORM freeware statistical software (Schafer, 1997) on these wells show a high correlation between their moisture records. Similarly soil moisture sensors located at PS40, PS41, and PS42 have a forest land cover and hence they showed strong correlation in their recorded values. Therefore the wells within the similar land cover were grouped together. This property of correlation was thus used to impute the missing values; the following paragraph describes the procedure used for data imputation.

Data for different sensors for each well was listed and the missing values flagged as -999.99. A VB code was written which, for the available data finds the ratio of the values between each well, hence for each set of available data three ratio (between first
and second, second and third, and first and third) values were found and stored in three arrays. As the resolution of the available data is 5 min, there are 288 data records for each 24 hour period. Assuming that correlation between the values is stronger for past 24 hours than the previous timed records only 288 records were stored and used for calculating the missing values. The assumption that the correlation is stronger for the past 24 hour period is based on the fact that the field data that is being used in this research has temporal variation. For instance consider a situation where the previous week was a total dry period and the present period is wet and trying to use data from the dry period for the correlation would result in wrong prediction and hence the window has been set to 24 hour period. The program scans the data and stores all three ratios using a moving window array where the last 288 ratios are stored and past records are discarded. As soon as a missing value is found the program looks and identifies the pattern (see Table 1) i.e. only one out of three values is missing or pair of values is missing. In an event if all three values are missing the program is incapable of imputing and it leaves the missing values as it is. The next paragraph discusses the way in which the data can be found missing for three variables while the next deal with the way the program handles these forms.

Table1. Table Depicting the Eight Different Ways a Dataset Containing Data for Three Variables Can Exist

<table>
<thead>
<tr>
<th>Form</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Form</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Second Form</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Third Form</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fourth Form</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fifth Form</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sixth Form</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Seventh Form</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Eighth Form</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
With reference to the Table 1, the “0” represents missing data and “1” represents good data. So as seen from the table the data can be missing in all the variables (First Form) or the data can exist for variable 1 and variable 2 and be missing for variable 3 (Fifth Form). Similarly the data can exist for all the three variables of the dataset (Eighth Form).

If only one value is missing, the program from the two arrays of ratios finds two average values of ratio one with each known well. Using the two average values and the two available data two values for the missing well are computed. The average of these two computed values thus gives the missing value which is used to replace the missing value. However if two values are missing (say for example first and third wells are missing) the average of the ratio between the first and the second and second and the third is calculated and then multiplied by the observed value of the second well the get the imputed values for both the well.

In all the cases only the past 288 records are used and if any missing data is observed then that record is completely skipped. Though the code written is for three wells it be easily expanded to included more wells if they are found to be correlated

3.5.1.1 Illustration of Simple Ratio Correlation Method

As an example of how the technique works lets consider a sample data with three variables (Table 2). From the table it can be seen that the data is missing for all the three basic patterns i.e. one value missing, two values missing and all three values missing ( -999.99 represent missing data). Now for all the available data the ratio between the
variable is calculated (as can be seen from the table 2) and stored in an array (NC represents that the ratios are not calculated).

<table>
<thead>
<tr>
<th>Variable 1</th>
<th>Variable 2</th>
<th>Variable 3</th>
<th>Ratio1 (Var1/Var2)</th>
<th>Ratio2 (Var2/Var3)</th>
<th>Ratio3 (Var1/Var3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.73</td>
<td>34.17</td>
<td>33.93</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>30.74</td>
<td>34.17</td>
<td>33.93</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>30.73</td>
<td>34.16</td>
<td>33.93</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>30.73</td>
<td>34.17</td>
<td>33.93</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>Case 1</td>
<td>30.74</td>
<td>34.17</td>
<td>-999.99</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>30.74</td>
<td>34.17</td>
<td>33.94</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>Case 2</td>
<td>30.74</td>
<td>-999.99</td>
<td>-999.99</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>30.74</td>
<td>34.17</td>
<td>33.93</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>30.74</td>
<td>34.17</td>
<td>33.94</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>30.74</td>
<td>34.17</td>
<td>33.94</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
<tr>
<td>Case 3</td>
<td>-999.99</td>
<td>-999.99</td>
<td>-999.99</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>30.73</td>
<td>34.17</td>
<td>33.94</td>
<td>0.90</td>
<td>1.01</td>
<td>0.91</td>
</tr>
</tbody>
</table>

In a record if one variable is missing, say for example Variable 3 is missing (Case 1) than the average of values recorded before the incomplete record under ‘Ratio2’ and ‘Ratio 3’ (see Table 2) are calculated, for the sample data they turn out to be 1.01 and 0.91 respectively. The available values i.e. 30.74 (variable1) and 34.17 (variable 2) are then divided by these ratios to get two values of Variable3. The two values obtained are then averaged to get the final imputed value for variable3 i.e. 33.93.

In case of two values missing in a record for example Variable 2 and Variable 3 (Case 2) the average of values recorded before the incomplete record under Ratio1 and
Ratio3 are calculated. For the sample dataset they turn out to be 0.90 and 0.91 respectively. The observed value i.e. Variable1 is then divided by the two ratios to get imputed values for Variable2 and Variable3 which are 34.17 and 33.93 respectively.

For Case 3 when all the variables are absent in a record the values are not imputed and the record is flagged as completely missing. An important thing to note in the whole process is that the ratios are recorded only for the observed values and the imputed values are never used to find the ratios.

3.5.2 Use of EM Algorithm to do Multiple Imputations for Multivariate Normal Data

An important component of the thesis website is to complete the missing soil moisture data by synthesizing it using statistical methods. These methods device the most expected values of the missing data. Free software called NORM (Schafer, 1999) was used to generate the missing data using a multivariate approach. It can be safely assumed the reading of the soil moisture sensor along a particular probe can be easily taken to be dependent on one another, which imply that the reading of one sensor can be a good indicator of the value of the other sensors. This approach thus allows the soil moisture variation along a particular probe to modeled using a multivariate model. NORM assumes a multivariate normal model and supports a large variety of transformations to convert non-normal data to become almost normal. Even though the software is designed for multivariate normal data, Schafer (1997) has demonstrated that it makes sense to use normal model for multiple imputation (MI) even for non-normal data, as the process is quite forgiving for reasonable departures from normalities.
The soil moisture data for each individual sensor obtained from the field shows a partial log-normal behavior and, using the above rational, can be modeled using the normal MI model. The data for soil moisture is transformed using the logit model (as supported in NORM); this not only allows the data to become almost normal but also constraints it within pre-specified physical bounds (if any). As the soil moisture data can never be lie outside the range of 0-40% in this area these values were defined as the range for the ‘LOGIT model’. Figure 12 shows the data for well PS40 (sensor #6) before and after the transformation. The Figure 12 clearly shows the change in the behavior from non-normal to almost normal, increasing the confidence in the normal model. Similar results were obtained for different wells and sensors. The following paragraphs describe the basic steps followed in using NORM and the statistical tools which the software uses in each step (for more details the software Help manual should be referred).

From the database the soil moisture data for all the eight sensors for each well was obtained as a single text file, with only the eight variables i.e. the eight sensor values. The missing values were flagged by ‘-999.99’ so that the software can distinguish between the observed and the missing values. Each column of the data file is associated with a variable and each row with data values. After the data is imported, an appropriate transformation is applied, for this particular case, as discussed above LOGIT transformation with pre-specified ranges is used. As the data is not truly normal the rounding was reduced to tenth of a decimal place so as to take care of some of the error that may arise due to this non-normality. The data is then analyzed using the EM (Expectation Maximization) algorithm which is a two step process of determining most likely values of parameters of the multivariate normal model; Parameter such as means,
variances, and covariance are calculated for the entire dataset using values from the cases that are partially incomplete. Running the EM algorithm helps in estimating parameters for the multivariate dataset which comes out to be useful in the data augmentation step where the missing values are computed.

![Plot for the Sensor 6 of the PS40 well](image)

Figure 12. Plot for the Sensor 6 of the PS40 well

Following the EM algorithm the parameters are then used for another step called Data Augmentation (DA). DA is an iterative simulation technique, a special kind of Markov chain Monte Carlo (MCMC). In DA there are three types of quantities: observed data, missing data, and parameters. The missing data and parameters are unknown. DA alternately performs the following steps:

- **I-step**: Impute the missing data by drawing them from their conditional distribution given the observed data and assumed values for the parameters

- **P-step**: Simulate new values for the parameters by drawing them from a Bayesian posterior distribution given the observed data and the most recently imputed values for the missing data.
Alternating between these two steps sets up a Markov chain that converges to a stationary distribution, the joint distribution of the missing data and parameters given the observed data. DA bears a strong resemblance to the EM algorithm, and may be regarded as a stochastic version of EM. It is useful for multiple imputation of missing data. By running DA for a large number of cycles, and storing the results of a few I-steps along the way (with enough cycles in between to ensure independence), one obtains proper multiple imputations of the missing data.

The data is imputed from the software and is automatically transformed back and stored in the separate file which is a complete data set including the observed and the missing data (replaced with new imputed values)

The data files obtained are again combined with the date and time fields and are imported back into database. This data can then be accessed using the synthesized data link from the website
CHAPTER 4. RESULTS

4.1 Description of the Toolbar

The toolbar is the leftmost frame of the first page of the “Thesis” website. The toolbar is illustrated in Figure 13.

Figure 13. The Toolbar that is seen on the “Thesis” Website.
The buttons on the toolbar are:

- 1 → Toggle between Legend and Overview map Button: To toggle between the overview map and the Legend.
- 2 → Toggle Button: To toggle the overview map
- 3 → ZoomIn Button: This button zooms in to the map
- 4 → ZoomOut Button: This button zooms out to the map
- 5 → ZoomToFullExtent: This button zooms to full extent of the map
- 6 → ZoomToActiveLayer: This button zooms to full extent of layer that is set active
- 7 → BackToLastExtent: This button zooms in or out depending on what was the previous operation on the map
- 8 → Pan: This button pans the data in the map by dragging the display in any direction with the mouse
- 9 → PanToNorth: This button pans the data in the map in the North direction
- 10 → PanToSouth: This button pans the data in the map in the South direction
- 11 → PanToEast: This button pans the data in the map in the East direction
- 12 → PanToWest: This button pans the data in the map in the West direction
- 13 → Hyperlink: This button gets some information about the data point (here the well selected) and also leads to the user interface where the clients can query the database for information about that data point
- 14 → Identify: This button gets the attribute information about the geographic feature or the data point selected
• 15 → Query: The user can build spatial queries on the data in the map using the Query tool

• 16 → Find: This tool finds a geographical feature or a data point with the given attribute

• 17 → DownloadData: This tool when used provides the user with a few links which would allow them to download some of the useful data of the CMHAS database

• 18 → Measure: This tool measures distances on the map

• 19 → SetUnits: This tool sets the units for the map

• 20 → Buffer: This tool creates buffer polygon to a specified distance around the input feature.

• 21 → SelectByRectangle: This tool selects the features on the map by constructing a rectangle that intersects the features you wanted to select

• 22 → SelectByLine/Polygon: The functionality of this tool is same as the previous tool except that either a line or a polygon should be constructed to intersect the features that needed to be selected

• 23 → ClearSelection: This tool clears the selection that has already been made

• 24 → Print: This tool prints the map

4.2 Home Page

The home page of the “Thesis” website comprises of the actual map of the project site in the center, the legend to the utmost right and the toolbar in the leftmost part of the webpage (Figure 14).
4.2.1 Text Area of the Home Page:

The results of the tools like Identify, Query, Find, DownloadData are displayed in the text area of the home page. The results of each of these tools are depicted in the screenshots which follow. The Figure 15 below shows the results of the “identify” tool when clicked on the data point called “PS41”.

Figure 14. Screenshot of the Home Page.
The Figure 16 below shows how the query is built using the “Query” tool to select the data points on the map which are located in “Grass”. The results are displayed in Figure 17.

Figure 16. Screenshot of the ‘Query’ Tool in Use
Figure 17. Screenshot of the Results of the ‘Query’ Tool.

The Figure 18 below shows the use of the “Find” tool to search for the given attribute among the layers on the map.

Figure 18. Screenshot of the ‘Find’ Tool in Use
The Figure 19 below has a screenshot of the webpage that is displayed when you click on a particular well.

![Welcome to CMHAS Database Online](image)

**Figure 19. Screenshot of the Webpage that is Displayed on Clicking Well PS41**

4.3 Data Synthesis

The soil moisture data for the reservoir was analyzed first for normality and though it generally followed a log normal trend it was not perfectly log normal and for several sensors the trend deviated significantly. As part of this research the data was analyzed using two methods discussed earlier in the Methods and Methodologies section. The data imputed using the statistical software did not reproduce the smooth natural variation of the system because the statistical method generated random numbers to fit
the mean matrix and the variance covariance matrix. Hence the approach of finding the
correlation between the sensors was used to do the imputation of missing data.

The wells in the grassland and forest were segregated as two different groups and
the VB code written (attached in the appendix) was used to synthesize the data. To test
the accuracy of the method some of the known data was artificially removed and marked
as missing to allow the comparison with the predicted and the know values. Table 3
summarizes the results obtained when one of the wells in the data set was removed, it can
be clearly seen that the predicted and the actual values vary by less than 1%. Similarly
data values from two wells were removed and compared against the predicted values
again the correlation seem to give a good result with the average difference of 1%.
Thus, the confidence on the scheme is increased. The same method was used to impute
the data at all the other well records as well as water table elevation records.
Table 3. Table Showing Artificially Removed Data Points and the Predicted Values for Well USF3 (all Data is Percent Soil Water Content for the Second Sensor from Land Surface of Each Well)

<table>
<thead>
<tr>
<th>PS43</th>
<th>USF1</th>
<th>USF3</th>
<th>Data Removed</th>
<th>Data Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.87</td>
<td>34.26</td>
<td>-999.99</td>
<td>34.04</td>
<td>34.00878</td>
</tr>
<tr>
<td>30.98</td>
<td>34.33</td>
<td>-999.99</td>
<td>34.12</td>
<td>34.10642</td>
</tr>
<tr>
<td>31.02</td>
<td>34.34</td>
<td>-999.99</td>
<td>34.15</td>
<td>34.13552</td>
</tr>
<tr>
<td>31.03</td>
<td>34.35</td>
<td>-999.99</td>
<td>34.16</td>
<td>34.14809</td>
</tr>
<tr>
<td>31.04</td>
<td>34.36</td>
<td>-999.99</td>
<td>34.17</td>
<td>34.16063</td>
</tr>
<tr>
<td>31.04</td>
<td>34.37</td>
<td>-999.99</td>
<td>34.16</td>
<td>34.16755</td>
</tr>
<tr>
<td>31.05</td>
<td>34.36</td>
<td>-999.99</td>
<td>34.17</td>
<td>34.16996</td>
</tr>
<tr>
<td>31.05</td>
<td>34.37</td>
<td>-999.99</td>
<td>34.18</td>
<td>34.17734</td>
</tr>
<tr>
<td>31.09</td>
<td>34.41</td>
<td>-999.99</td>
<td>34.24</td>
<td>34.22188</td>
</tr>
<tr>
<td>31.18</td>
<td>34.47</td>
<td>-999.99</td>
<td>34.33</td>
<td>34.30412</td>
</tr>
<tr>
<td>31.21</td>
<td>34.49</td>
<td>-999.99</td>
<td>34.36</td>
<td>34.33404</td>
</tr>
<tr>
<td>31.21</td>
<td>34.50</td>
<td>-999.99</td>
<td>34.39</td>
<td>34.34321</td>
</tr>
<tr>
<td>31.22</td>
<td>34.51</td>
<td>-999.99</td>
<td>34.39</td>
<td>34.35754</td>
</tr>
<tr>
<td>31.23</td>
<td>34.51</td>
<td>-999.99</td>
<td>34.39</td>
<td>34.36562</td>
</tr>
<tr>
<td>31.23</td>
<td>34.52</td>
<td>-999.99</td>
<td>34.39</td>
<td>34.37304</td>
</tr>
<tr>
<td>31.23</td>
<td>34.53</td>
<td>-999.99</td>
<td>34.40</td>
<td>34.38012</td>
</tr>
<tr>
<td>31.23</td>
<td>34.53</td>
<td>-999.99</td>
<td>34.41</td>
<td>34.38204</td>
</tr>
<tr>
<td>31.23</td>
<td>34.53</td>
<td>-999.99</td>
<td>34.42</td>
<td>34.38355</td>
</tr>
<tr>
<td>31.24</td>
<td>34.53</td>
<td>-999.99</td>
<td>34.42</td>
<td>34.39048</td>
</tr>
<tr>
<td>31.23</td>
<td>34.54</td>
<td>-999.99</td>
<td>34.43</td>
<td>34.39142</td>
</tr>
<tr>
<td>31.24</td>
<td>34.54</td>
<td>-999.99</td>
<td>34.43</td>
<td>34.39847</td>
</tr>
<tr>
<td>31.24</td>
<td>34.54</td>
<td>-999.99</td>
<td>34.43</td>
<td>34.39999</td>
</tr>
<tr>
<td>31.25</td>
<td>34.54</td>
<td>-999.99</td>
<td>34.44</td>
<td>34.40668</td>
</tr>
<tr>
<td>31.24</td>
<td>34.54</td>
<td>-999.99</td>
<td>34.45</td>
<td>34.40238</td>
</tr>
</tbody>
</table>

To quantify the results produced by the linear correlation method F-test (MS Excel) has been performed on the original data and the predicted data. The results of the f-test shows that there was no significant difference between the original data and the predicted data (see Table 4). The Table 5 shows the statistics generated by the F-test based on the data in Table 3.
Table 4. Table Showing the Comparison between the Removed and the Predicted Values in Case of Two Missing Values (PS43 and USF3) in a Record (all Data is Percent Soil Water Content for the Second Sensor from Land Surface of Each Well)

<table>
<thead>
<tr>
<th>PS43</th>
<th>USF1</th>
<th>USF3</th>
<th>Values Removed</th>
<th>Predicted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-999.99</td>
<td>35.31</td>
<td>-999.99</td>
<td>32.45 35.62 32.47 35.56</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.31</td>
<td>-999.99</td>
<td>32.45 35.62 32.47 35.56</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.31</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.56</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.31</td>
<td>-999.99</td>
<td>32.45 35.62 32.47 35.56</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.31</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.56</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.62 32.46 35.54</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.61 32.46 35.54</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.62 32.46 35.54</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.61 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.61 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.62 32.46 35.54</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.62 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.60 32.46 35.54</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.30</td>
<td>-999.99</td>
<td>32.44 35.60 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.59 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.59 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.59 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.59 32.47 35.55</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.59 32.46 35.54</td>
<td></td>
</tr>
<tr>
<td>-999.99</td>
<td>35.29</td>
<td>-999.99</td>
<td>32.44 35.59 32.46 35.54</td>
<td></td>
</tr>
</tbody>
</table>

48
Table 5. Statistics Generated by the F-test.

<table>
<thead>
<tr>
<th></th>
<th>Variable 1</th>
<th>Variable 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>34.32405</td>
<td>34.29764038</td>
</tr>
<tr>
<td>Variance</td>
<td>0.017325</td>
<td>0.014716825</td>
</tr>
<tr>
<td>Observations</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>df</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>F</td>
<td>1.177245</td>
<td></td>
</tr>
<tr>
<td>P(F&lt;=f) one-tail</td>
<td>0.34323</td>
<td></td>
</tr>
<tr>
<td>F Critical one-tail</td>
<td>1.955447</td>
<td></td>
</tr>
</tbody>
</table>

The table above shows the result of the F-test done for the data values given in Table 3. The result further proves that though the method is more or less empirical it is really useful for the data imputation especially for the wells located at the reservoir site.
CHAPTER 5. CONCLUSIONS

In this research a framework for presenting the water resources field database online has been developed. The “Thesis” website is a unique way of presenting data as it involves an integration of three different methodologies, namely providing a map with GIS capabilities without requiring the clients (users of the website) having to install ArcGIS software on their personal computers, providing a comprehensive watershed database simultaneously maintaining the security of the database, and developing a methodology to synthesize data in case of data gaps.

The importance of the work can be briefly stated as follows:

- The map having comprehensive information of the project site is provided in order to fill the need of allowing spatial analysis and mapping along with downloading data for a particular data point.

- The users of the website are not required to have ArcMap installed on their PC’s. This means that providing an ArcIMS customized map allows the users to enjoy the GIS functionality while saving them from buying a GIS license. Thus, having an ArcIMS map is quite economical.

- The website makes it easy to access the watershed database by providing an easy-to-use and understand user interface, because many people are not comfortable in
querying the data from the database as such. However useful a database is, it is less useful if users cannot get information out of it easily. A web application is like a “Cherry on the Sundae”, providing access to the knowledge locked inside the database.

- Very few sites provide comprehensive dataset in the field of hydrology. This makes the website a thoroughly useful tool for researchers in multi-disciplinary water resource sciences such as Soil Physics, hydrological modeling. The graduate students worldwide can use data for their research.

- The website, besides providing all the above discussed uses, maintains and preserves the database integrity.

- Providing the users with the capability of filling data gaps allows the water resource scientists to get a complete dataset for the period queried which is again an important requirement for many people in the field of Water Resources to have a complete record for their research.

In the course of creating the “Thesis” web application many issues have been encountered. Some to be considered in future studies include.

- Planning is the key to good application development. Before the web application is developed it is always necessary to talk out the requirements needed form the website which would save lot of “reworking” from efficient user friendly web application.
• A database should be recognized as not just a repository of data but as a mutable business process which constantly needs to be reassessed to determine if it still meets the needs.

• Planning for web application development is equal (if not greater) in importance to coding. While mediocre code can be adjusted to make it run better, web application not developed to meet the needs of the users requires that the whole system be redesigned.

By keeping these thoughts and ideas in mind an ArcIMS web application can be implemented effectively while helping to improve efficiency and productivity at a utility.

For future study I make the following recommendations:

• Upgrading the database to SQL server or Oracle would increase the security of the database and the users of the website can be provided with facility to make their own queries.

• All the sites of CMHAS can be integrated together as a single system.

• Use more sophisticated software like SAS and methods like the propensity score method (which is still in experimental stage) to come up with the missing values to fill data gaps.
REFERENCES


Appendix A: Servlet to Query and Download Data from the Database.

This is a sample servlet to query the database for soil moisture data and this can be taken as a guideline for writing servlets to query data for other hydrological parameters from the database.

```java
import java.sql.*;
import java.util.*;
import java.math.*;
import java.io.*;
import java.net.*;
import java.text.*;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.lang.*;

/**
 * @author nshah5
 * @version
 */
public class NewWellServlet extends HttpServlet {

    /** Initializes the servlet. *
     */

    /** Destroys the servlet. *
     */

    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods. *
     * @param request servlet request *
     * @param response servlet response *
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

    56
```
throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    ResultSet recResults = null;
    Statement STrecResults = null;
    String wellName=request.getParameter("wellID");
    String position=request.getParameter("wellNumber");
    String D1=request.getParameter("Day1");
    String M1=request.getParameter("Month1");
    String Y1=request.getParameter("Year1");
    String D2=request.getParameter("Day2");
    String M2=request.getParameter("Month2");
    String Y2=request.getParameter("Year2");
    String interval=request.getParameter("Interval");
    ResultSetMetaData meta;
    String [] ColumnName =null;
    int DisColumns = 0;
    int unDisCol = 0;
    String temp;
    String [] tempColumnName;
    String sLine = "";
    String sQuery="";
    int d1val=0;
    int d2val=0;
    int m1val1=0;
    int m1val=0;
    int m2val1=0;
    int m2val=0;
    int y1val1=0;
    int y1val=0;
    int y2val1=0;
    int y2val=0;
    int dataID=0;

    if (wellName.equalsIgnoreCase("PS40"))
    {
        if (position.equalsIgnoreCase("first"))
            dataID=36;
        else
            if (position.equalsIgnoreCase("second"))
                dataID=37;
            else

57
Appendix A: (Continued)

```java
if (position.equalsIgnoreCase("third"))
    dataID=38;
else
if (position.equalsIgnoreCase("fourth"))
    dataID=39;
else
if (position.equalsIgnoreCase("fifth"))
    dataID=40;
else
if (position.equalsIgnoreCase("sixth"))
    dataID=41;
else
if (position.equalsIgnoreCase("seventh"))
    dataID=42;
else
if (position.equalsIgnoreCase("eighth"))
    dataID=43;
```

```java
}
else
if (wellName.equalsIgnoreCase("PS41"))
{
    if (position.equalsIgnoreCase("first"))
        dataID=26;
    else
if (position.equalsIgnoreCase("second"))
        dataID=27;
    else
if (position.equalsIgnoreCase("third"))
        dataID=28;
    else
if (position.equalsIgnoreCase("fourth"))
        dataID=29;
    else
if (position.equalsIgnoreCase("fifth"))
        dataID=30;
    else
if (position.equalsIgnoreCase("sixth"))
        dataID=32;
    else
if (position.equalsIgnoreCase("seventh"))
        dataID=33;
    else
```
Appendix A: (Continued)

    if (position.equalsIgnoreCase("eighth"))
        dataID = 34;
}
else
if (wellName.equalsIgnoreCase("PS42"))
{
    if (position.equalsIgnoreCase("first"))
        dataID = 18;
    else
        if (position.equalsIgnoreCase("second"))
            dataID = 19;
    else
        if (position.equalsIgnoreCase("third"))
            dataID = 20;
    else
        if (position.equalsIgnoreCase("fourth"))
            dataID = 21;
    else
        if (position.equalsIgnoreCase("fifth"))
            dataID = 22;
    else
        if (position.equalsIgnoreCase("sixth"))
            dataID = 23;
    else
        if (position.equalsIgnoreCase("seventh"))
            dataID = 24;
    else
        if (position.equalsIgnoreCase("eighth"))
            dataID = 25;
}
else
if (wellName.equalsIgnoreCase("PS43"))
{
    if (position.equalsIgnoreCase("first"))
        dataID = 10;
    else
        if (position.equalsIgnoreCase("second"))
            dataID = 11;
    else
        if (position.equalsIgnoreCase("third"))
            dataID = 12;
else
  if (position.equalsIgnoreCase("fourth"))
    dataID=13;
  else
  if (position.equalsIgnoreCase("fifth"))
    dataID=14;
  else
  if (position.equalsIgnoreCase("sixth"))
    dataID=15;
  else
  if (position.equalsIgnoreCase("seventh"))
    dataID=16;
  else
  if (position.equalsIgnoreCase("eighth"))
    dataID=17;

  }
else
if (wellName.equalsIgnoreCase("USF1"))
  { 
    if (position.equalsIgnoreCase("first"))
      dataID=45;
    else
    if (position.equalsIgnoreCase("second"))
      dataID=46;
    else
    if (position.equalsIgnoreCase("third"))
      dataID=47;
    else
    if (position.equalsIgnoreCase("fourth"))
      dataID=48;
    else
    if (position.equalsIgnoreCase("fifth"))
      dataID=49;
    else
    if (position.equalsIgnoreCase("sixth"))
      dataID=50;
    else
    if (position.equalsIgnoreCase("seventh"))
      dataID=51;
    else
    if (position.equalsIgnoreCase("eighth"))
      dataID=52;
Appendix A: (Continued)

else
    if (wellName.equalsIgnoreCase("USF3"))
    {
        if (position.equalsIgnoreCase("first"))
            dataID = 54;
        else
            if (position.equalsIgnoreCase("second"))
                dataID = 55;
            else
                if (position.equalsIgnoreCase("third"))
                    dataID = 56;
                else
                    if (position.equalsIgnoreCase("fourth"))
                        dataID = 57;
                    else
                        if (position.equalsIgnoreCase("fifth"))
                            dataID = 58;
                        else
                            if (position.equalsIgnoreCase("sixth"))
                                dataID = 59;
                            else
                                if (position.equalsIgnoreCase("seventh"))
                                    dataID = 60;
                                else
                                    if (position.equalsIgnoreCase("eighth"))
                                        dataID = 61;
    
    }

if (interval.equalsIgnoreCase("5 Minute Interval"))
    sQuery = "select * from MoistureData where MoistureData.DataCollectionID=? AND MoistureData.DateTime>? AND MoistureData.DateTime<?";
else
    if (interval.equalsIgnoreCase("1 Hour Interval"))
        sQuery = "select * from HourlyMoistureData where HourlyMoistureData.DataCollectionID=? AND HourlyMoistureData.DateTime>? AND HourlyMoistureData.DateTime<?";
else
    if (interval.equalsIgnoreCase("24 Hour Interval"))
try {
    d1val=Integer.parseInt(D1);
    d2val=Integer.parseInt(D2);
    m1val1=Integer.parseInt(M1);
    m1val=m1val1-1;
    m2val1=Integer.parseInt(M2);
    m2val=m2val1-1;
    y1val1=Integer.parseInt(Y1);
    y1val=y1val1-1900;
    y2val1=Integer.parseInt(Y2);
    y2val=y2val1-1900;
}
    catch(NumberFormatException numberException) {
        numberException.printStackTrace();
        System.exit(1);
    }

Connection con=null;
java.sql.Timestamp startDate;
java.sql.Timestamp stopDate;
PreparedStatement ps1 = null;
//Mark this end
java.util.Date myDate = new java.util.Date();
SimpleDateFormat df = new SimpleDateFormat("MM/dd/yyyy");

String path = "c:/temp";
String fileName = "temp.TXT";
File file = null;
 FileOutputStream fo = null;
 PrintStream ps = null;
 try {
    file = new File(path,fileName);
    if(file.exists()) {

Appendix A: (Continued)

file.delete();
file.createNewFile();
}

fo = new FileOutputStream(file);
ps = new PrintStream(fo);
} catch (IOException exp) {
    System.out.println("IO Exception: "+exp.toString());
}

//Connecting to the Database

try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con = DriverManager.getConnection("jdbc:odbc:ThesisDatabase","","");  
} catch (SQLException sqlException) {
    sqlException.printStackTrace();
    System.exit(1);
} catch (ClassNotFoundException classNotFound) {
    classNotFound.printStackTrace();
    System.exit(1);
}

//Mark this start
try {
    startDate= new java.sql.Timestamp(y1val,m1val,d1val,0,0,0,0);
    stopDate = new java.sql.Timestamp(y2val,m2val,d2val,0,0,0,0);

    ps1 = con.prepareStatement(sQuery);
    ps1.setInt(1,dataID);
    ps1.setTimestamp(2,startDate);
    ps1.setTimestamp(3,stopDate);
}

} catch(IllegalArgumentException illegalArgument){

illegalArgument.printStackTrace();
    System.exit(1);
}
catch (SQLException sqlException) {
    sqlException.printStackTrace();
    System.exit(1);
}

//Mark this end
int columns = 0;
try {
    recResults = ps1.executeQuery();
    meta = recResults.getMetaData();
    columns = meta.getColumnCount();
    tempColumnName = new String[columns];
    if(recResults.next()) { //if_1
        for(int n=0;n<columns;n++) {
            temp = meta.getColumnName(n + 1);
            if (!temp.equals("PROJECTID")&&!temp.equals("BUILDINGID")&&!temp.equals("HAZMATPROFILEID")) {
                sLine = sLine + "'" + temp + "'";
                tempColumnName[DisColumns] = temp;
                DisColumns ++;
                ColumnName = new String[DisColumns];
            } else {
                unDisCol ++;
            }
        } //end for
        for(int i=0;i<(columns-unDisCol);i++) {
            ColumnName = tempColumnName;
        }
        ps.println(sLine);
    }
    do{
        sLine = "";
        for(int n=0;n<(columns-unDisCol);n++) {
            String tempColName = recResults.getString(ColumnName[n]);
            if(tempColName==null) {
                sLine = sLine + "" + " ";
            } else {
            } //end if
        }
        ps.println(sLine);
    } while

} //Mark this end
Appendix A: (Continued)

```java
sLine = sLine + ""+tempColName+"" + " ";
}
}

ps.println(sLine);
} while(recResults.next());
} //end if_1

} catch (SQLException sqlException)
{
    sqlException.printStackTrace();
    System.exit(1);
}
finally{
    try{
        recResults.close();
        recResults = null;
    }
    catch(Exception exception)
    {
        exception.printStackTrace();
        System.exit(1);
    }
}

out.println("<html>");
out.println("<head>");
out.println("<title>Servlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<meta http-equiv="refresh" content="0; URL=/WebApplication1/DownloadServlet?folder="+path+"&fname="+file+""">");
out.println("</body>");
out.println("</html>");

/** Handles the HTTP <code>GET</code> method.
*/
Appendix A: (Continued)

* @param request servlet request
* @param response servlet response
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    System.out.println("Inside do get method");
    processRequest(request, response);
}

/** Handles the HTTP <code>POST</code> method.
* @param request servlet request
* @param response servlet response
*/
}
Appendix B: Visual Basic Code for Linear Correlation Technique

This is the visual basic code written for synthesizing data using the linear correlation technique.

'Code for imputation of data using linear correlation
'Initializing variable
Option Base 1
'Initializing counter for number of ratios recorded
Dim cnt12 As Long, cnt13 As Long, cnt23 As Long
'Dynamic Arrays to store ratio for each pair
Dim r12() As Single, r13() As Single, r23() As Single
Dim t1 As Single, t2 As Single, t3 As Single
Private Sub Command1_Click()

'Opening dialog box to select the file with missing data
cmddlg.ShowOpen
r = cmddlg.FileName
Open r For Input As #1

'Opening dialog box to specify the file to save the imputed data
cmddlg.ShowSave
s = cmddlg.FileName
Open s For Output As #2

top:

'Scanning the whole data file
Do Until EOF(1)

'Reading data from the file
Input #1, t1, t2, t3
'Identifying the missing patterns and sending the control accordingly
If t1 = -999.99 And t2 <> -999.99 And t3 <> -999.99 Then GoTo missing_1 ' Pattern 011
If t2 = -999.99 And t1 <> -999.99 And t3 <> -999.99 Then GoTo missing_2 ' Pattern 101
If t3 = -999.99 And t1 <> -999.99 And t2 <> -999.99 Then GoTo missing_3 ' Pattern 110
If t1 = -999.99 And t2 = -999.99 And t3 <> -999.99 Then GoTo missing_12 ' Pattern 001
If t1 = -999.99 And t3 = -999.99 And t2 <> -999.99 Then GoTo missing_13 ' Pattern 010
If t3 = -999.99 And t2 = -999.99 And t1 <> -999.99 Then GoTo missing_23 ' Pattern 100
If t1 = -999.99 And t3 = -999.99 And t2 = -999.99 Then GoTo missing_123 ' Pattern 000
Appendix B: (Continued)

'Increasing counter for a complete record

cnt12 = cnt12 + 1
cnt13 = cnt13 + 1
cnt23 = cnt23 + 1

'Keeping data for 24 hours
'For all the three dynamic arrays

If cnt12 <= 288 Then
    ReDim Preserve r12(cnt12)
    r12(cnt12) = t1 / t2
Else
    For j = 1 To 287
        r12(j) = r12(j + 1)
    Next j
    r12(288) = t1 / t2
End If

If cnt13 <= 288 Then
    ReDim Preserve r13(cnt12)
    r13(cnt12) = t1 / t3
Else
    For j = 1 To 287
        r13(j) = r13(j + 1)
    Next j
    r13(288) = t1 / t3
End If

If cnt23 <= 288 Then
    ReDim Preserve r23(cnt12)
    r23(cnt12) = t2 / t3
Else
    For j = 1 To 287
        r23(j) = r23(j + 1)
    Next j
    r23(288) = t2 / t3
End If
Write #2, t1, t2, t3
Loop
Close #1, #2
Exit Sub
CALCULATING MISSING VALUES

Calculating values for 1st sensor missing

missing_1:

If cnt23 <= 288 Then
ReDim Preserve r23(cnt12)
r23(cnt12) = t2 / t3
Else
For j = 1 To 287
r23(j) = r23(j + 1)
Next j
r23(288) = t2 / t3
End If

sum1 = 0
If cnt12 < 288 Then
For i = 1 To cnt12
sum1 = sum1 + r12(i)
Next i
avg1 = sum1 / cnt12
Else
For i = 1 To 288
sum1 = sum1 + r12(i)
Next i
avg1 = sum1 / 288
End If

sum1 = 0
If cnt13 < 288 Then
For i = 1 To cnt13
sum1 = sum1 + r13(i)
Next i
avg2 = sum1 / cnt13
Else
For i = 1 To 288
sum1 = sum1 + r13(i)
Next i
avg2 = sum1 / 288
End If

'Finding average for the two values to determine the final value
Appendix B: (Continued)

t1 = (avg1 * t2 + avg2 * t3) / 2

Write #2, t1, t2, t3
GoTo top

'Calculating values for the second sensor
missing_2:

If cnt13 <= 288 Then
    ReDim Preserve r13(cnt12)
    r13(cnt12) = t1 / t3
Else
    For j = 1 To 287
        r13(j) = r13(j + 1)
    Next j
    r13(288) = t1 / t3
End If

sum1 = 0
If cnt12 < 288 Then
    For i = 1 To cnt12
        sum1 = sum1 + r12(i)
    Next i
    avg1 = sum1 / cnt12
Else
    For i = 1 To 288
        sum1 = sum1 + r12(i)
    Next i
    avg1 = sum1 / 288
End If

sum1 = 0
If cnt23 < 288 Then
    For i = 1 To cnt23
        sum1 = sum1 + r23(i)
    Next i
    avg2 = sum1 / cnt23
Else
    For i = 1 To 288
        sum1 = sum1 + r23(i)
    Next i
    avg2 = sum1 / 288
End If
Appendix B: (Continued)

t2 = (t1 / avg1 + avg2 * t3) / 2
Write #2, t1, t2, t3
GoTo top

'Calculating values for the third sensor

missing_3:

If cnt12 <= 288 Then
  ReDim Preserve r12(cnt12)
r12(cnt12) = t1 / t2
Else
  For j = 1 To 287
    r12(j) = r12(j + 1)
  Next j
r12(288) = t1 / t2
End If

sum1 = 0
If cnt13 < 288 Then
  For i = 1 To cnt13
    sum1 = sum1 + r13(i)
  Next i
  avg1 = sum1 / cnt13
Else
  For i = 1 To 288
    sum1 = sum1 + r13(i)
  Next i
  avg1 = sum1 / 288
End If

sum1 = 0
If cnt23 < 288 Then
  For i = 1 To cnt23
    sum1 = sum1 + r23(i)
  Next i
  avg2 = sum1 / cnt13
Else
  For i = 1 To 288
    sum1 = sum1 + r23(i)
  Next i
  avg2 = sum1 / 288
End If
Appendix B: (Continued)

\[
t_3 = \frac{t_1 / \text{avg}1 + t_2 / \text{avg}2}{2}
\]
Write #2, t1, t2, t3
GoTo top

'Calculating missing values for 1st and 2nd sensor

missing_12:
'1st data
sum1 = 0
If cnt13 < 288 Then
    For i = 1 To cnt13
        sum1 = sum1 + r13(i)
    Next i
    avg1 = sum1 / cnt13
Else
    For i = 1 To 288
        sum1 = sum1 + r13(i)
    Next i
    avg1 = sum1 / 288
End If
  t1 = (t3 * avg1)

'2nd data
sum1 = 0
If cnt23 < 288 Then
    For i = 1 To cnt23
        sum1 = sum1 + r23(i)
    Next i
    avg1 = sum1 / cnt23
Else
    For i = 1 To 288
        sum1 = sum1 + r23(i)
    Next i
    avg1 = sum1 / 288
End If
  t2 = (t3 * avg1)
Write #2, t1, t2, t3
GoTo top

'Calculating missing values for 1st and 3rd sensor missing

missing_13:
'1st data
sum1 = 0
If \( \text{cnt12} < 288 \) Then
   For \( i = 1 \) To \( \text{cnt12} \)
      \( \text{sum1} = \text{sum1} + r12(i) \)
   Next \( i \)
   \( \text{avg1} = \text{sum1} / \text{cnt12} \)
Else
   For \( i = 1 \) To 288
      \( \text{sum1} = \text{sum1} + r12(i) \)
   Next \( i \)
   \( \text{avg1} = \text{sum1} / 288 \)
End If
\( t1 = (t2 \times \text{avg1}) \)

'2nd data
\( \text{sum1} = 0 \)
If \( \text{cnt23} < 288 \) Then
   For \( i = 1 \) To \( \text{cnt23} \)
      \( \text{sum1} = \text{sum1} + r23(i) \)
   Next \( i \)
   \( \text{avg1} = \text{sum1} / \text{cnt23} \)
Else
   For \( i = 1 \) To 288
      \( \text{sum1} = \text{sum1} + r23(i) \)
   Next \( i \)
   \( \text{avg1} = \text{sum1} / 288 \)
End If
\( t3 = (t2 / \text{avg1}) \)
Write #2, \( t1 \), \( t2 \), \( t3 \)
GoTo top

'Calculating values for the 2nd and 3rd sensor missing
missing_23:
\( \text{sum1} = 0 \)
'1st data
If \( \text{cnt12} < 288 \) Then
   For \( i = 1 \) To \( \text{cnt12} \)
      \( \text{sum1} = \text{sum1} + r12(i) \)
   Next \( i \)
   \( \text{avg1} = \text{sum1} / \text{cnt12} \)
Else
   For \( i = 1 \) To 288
      \( \text{sum1} = \text{sum1} + r12(i) \)
   Next \( i \)
Appendix B: (Continued)

    avg1 = sum1 / 288
    End If
    t2 = (t1 / avg1)

'2nd data
sum1 = 0
If cnt13 < 288 Then
    For i = 1 To cnt13
        sum1 = sum1 + r13(i)
    Next i
    avg1 = sum1 / cnt13
Else
    For i = 1 To 288
        sum1 = sum1 + r13(i)
    Next i
    avg1 = sum1 / 288
End If
    t3 = (t2 / avg1)
Write #2, t1, t2, t3
GoTo top

'Keeping all the values as missing
missing_123:
Write #2, t1, t2, t3
GoTo top

'Program End
End Sub