DEVELOPMENT OF A GIS BASED GRAPHICAL USER INTERFACE FOR IRRIGATION MANAGEMENT

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

IRRIGATION WATER MANAGEMENT

By

NIKHIL SUNIL MOGAREKAR

DEPARTMENT OF WATER RESOURCES DEVELOPMENT AND MANAGEMENT
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
MAY, 2015

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in this dissertation entitled "DEVELOPMENT OF A GIS BASED GRAPHICAL USER INTERFACE FOR IRRIGATION MANAGEMENT" in partial fulfillment of the requirements for the award of the degree of Master of Technology in Irrigation Water Management under Department of Water Resource Development and Management , Indian Institute of Technology Roorkee is an authentic record of my own work carried out during the period from July, 2014 to May, 2015 under the supervision and valuable guidance of Dr. Ashish Pandey, Associate Professor, Department of Water Resources Development and Management, Indian Institute of Technology Roorkee, India.

I have not submitted the matter embodied in this Dissertation for the award of any other degree.

Dated : 25th May 2015

Place : Roorkee

Nikhil Sunil Mogarekar

Enrollment No. 13547006

....

This is to certify that the above mentioned statement made by the candidate is correct and true to the best of my knowledge.

Dr. Ashish Pandey

Associate professor

WRD & M, IIT Roorkee

Roorkee – 247667

India

ACKNOWLEDGEMENT

I take this opportunity to express all my gratitude and thanks to the God, without whose help and mercy, this work would not have come out in its present form. I would like to thank all associated directly or indirectly in the proper completion and compilation of this report.

I acknowledge my immeasurable gratitude and sincere thanks to my guide Dr. Ashish Pandey, Associate Professor, Department of Water Resources Development and Management, Indian Institute of Technology, Roorkee for invaluable guidance, motivation, patience and feedback during the dissertation work and preparation of this report.

I must acknowledge an enormous debt of gratitude to Dr. Deepak Khare, Professor and Head, WRD&M and other Faculty and Staff members of the Department of Water Resources Development & Management, Indian Institute of Technology, Roorkee for providing facilities and valuable support.

I would like to acknowledge my thanks to Prof. Bhutada S.H., Honorable Principal of Aditya Agricultural Engineering and Technology College, Beed for his motivation and moral support.

I would like to thank Ph.D. Research scholars Dheeraj Kumar, Sushil Himanshu, Amarkant Gautam, Palmate Santosh Subhash and all of my friends of WRD & IWM especially Vijaya Kumar P.G., Gurmeet Singh, Thajuddin Ubaid B M, Alok Kumar and Sandeep Kumar Singh for their cooperation and support during the work. I must gratefully acknowledge to my parents and all of them who have directly or indirectly helped me to provide cooperation, input and encouragement during my research work.

25th May, 2015
IITR, Roorkee

Nikhil Sunil Mogarekar
Enrollment No: 13547006

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CCA	:	Culturable Command Area
CWCR	:	Centre for Research in Water Resources
ET	:	Evapotranspiration
ETc	:	Crop Evapotranspiration
ET_0	:	Reference Evapotranspiration
FAO	:	Food and Agriculture Organization
GIS	:	Geographic information System
IDW	:	Inverse distance Weighted
IIT	:	Indian Institute of Technology
IITR	:	Indian Institute of Technology, Roorkee
Kc	:	Crop coefficient.
RMSPE	:	Root mean square predict.tion error (RMSPE)
SIMIS	:	Scheme Irrigation Management Information System
SWAT	:	Soil and Water Assessment Tool
UI	:	User Interface
VBA	:	Visual basic for Application
WRD	:	Water Resources Development
WRD&M	:	Water Resources Development and Management

ABSTRACT

In this study, Graphical User Interface for Irrigation Management (GUIIM) has been developed employing Geographical Information System (GIS) and customization using Visual Basic for Application (VBA). GUIIM was developed for the purpose of analyses, simulation, handling and integration of a spatial data of irrigation management. The developed interface was used for estimation of spatial irrigation water requirement of Betwa River Basin situated in Madhya Pradesh, India. The output obtained from the GUIIM in the form of spatial maps i.e. rainfall, effective rainfall, daily reference evapotranspiration, daily crop evapotranspiration and net irrigation requirement will be useful in decision making process of irrigation management.

CHAPTER 1
INTRODUCTION

Irrigation plays a vital role for achieving an efficient and sustainable agricultural production (Gundogdu et al., 2002; Ortega et al., 2005). According to the Food and Agriculture Organization (FAO), in upcoming decades assurance of food providence will be a major concern due to uneven rise in world's population. To overcome the problem proper irrigation management will be the best solution. Irrigation Management will not only helps in saving water, improving agriculture production but also helps in reducing environmental and social hazards causes due to either excess or light irrigation. (Hargreaves et al., 1985; Suryavanshi and Reddy, 1986; Bastiaanssen and Makin, 2000). Irrigation Management is multi tasking job which deviates with the goals so it becomes major challenge for experts. Conventional Irrigation Management system may drive dispute due to spatial and temporal variability (Lin et al., 2004).

The ability of Geographic Information Systems (GIS) to analyze and visualize spatial and non spatial data in the form of maps made it an essential tool for agricultural irrigation management systems. GIS is currently converging with several other technologies to provide new levels of accessibility and functionality (Dhakal et al, 2010) hence use of Geographic Information Systems (GIS) customization with Visual Basic for Applications (VBA) can be a better solution for irrigation management. GIS has a wide range of integration of spatial & temporal data required for irrigation management (Pervez and Hoque, 2008; Naidu and Giridhar, 2011). Maps provided by GIS with interpolation techniques can be easily understood by Farmers, Irrigation experts and planners for Irrigation Management. (Martin, 1996; Bastiaanssen and Bos, 1999; Bioggio and Ding, 2001; Kjelds and Storm, 2001; Nixon et al., 2001; Su and Wen, 2001; Ray et al., 2001; Liang and Jianyu, 2012).

ArcGIS is most commonly used GIS application in irrigation and water resource management projects (Acharya Saroj, et.al., 2014;Tsihrintzis et al., 1996). The ArcGIS provides capability to users to analyze and visualize spatial and non spatial data in the form of maps. It also provides an environment for customization to prepare tools/Models. ArcGIS can be customized effectively with VBA for development of an essential tool for estimation of spatial irrigation water requirements which can help decision maker to manage irrigation system quickly and efficiently (Yamashita and

Walker, 1994; Teixeira and Pereira, 1992; Choudhury et al., 1994; Hales and Burton, 2000; Heinemannet al., 2002; Rao et al., 2004; Fortes et al., 2005; Lozano and Mateos, 2008; Raut et al., 2008; Dhakal, 2010).

In this study ArcGIS customization using VBA is carried out to develop an essential tool, A GIS based graphical user interface for irrigation management, which allow modeling irrigation water requirement spatially to support irrigation management systems. The developed interface tool was used to model irrigation water requirement spatially and generates thematic raster maps of rainfall, effective rainfall, daily reference evapotranspiration, daily crop evapotranspiration and net irrigation requirement for the study area.

1.1 Problem Statement

Tools which are available for irrigation management without GIS support has major drawback is that, they cannot deal with spatial and temporal irrigation data and unable to give output in form of maps which are easily visualize and understood by the farmers and irrigation experts. Tools which are available with GIS support, most of them cannot deal with large scale data. These tools are authentic only for small scale data. Due to variability in spatial and temporal components, it is difficult to transmit with traditional analytic techniques since the information is provided in simple graphs and tables regarding irrigation management.

In spite of development of few GIS based irrigation tools, which are not popular among farmers and irrigation experts because of complexity involved in many parameter and temporal data. The main limitation of these models is that they do not simulate in real time basis and are not user friendly. There is still need of work on development of irrigation management tool to fulfill the demand of irrigation experts and farmers which can give answer on spatial water requirements and water surplus/deficit of the command area.

1.2 Objectives

The main objectives of the present study is to develop a GIS based user graphical interface to support irrigation management system, with the aim of analyzing irrigation water requirement and corresponding surplus or deficit spatially. This tool is used to generate raster maps of irrigation requirement parameters.

Specific objectives of the study are as follows:

i) Customization of ArcGIS user interface using new UI controls and Menus for user friendly tool tailored for irrigation management.

ii) ArcGIS customization using VBA programming to compute reference evapotranspiration and crop evapotranspiration for Betwa river basin, India.

iii) Interpolation of point rainfall data to develop spatial rainfall maps using interpolation techniques and VBA programming.

iv) Development of spatial irrigation requirement thematic maps using developed interface based on rainfall and reference evapotranspiration maps to support analysis and decision making process in irrigation management for Betwa river Basin.

In this study main emphasis is to develop a real time simulation, user friendly and graphical user interface so that the decision makers are able to use this system quickly and easily for irrigation management. These maps can be easily understood by farmers, planners and specialists for spatial irrigation management as these maps can provide information most effectively.

1.3 Thesis Structure

The dissertation report is organized in following six chapters,

Chapter 1 (Introduction) provide the general description of the thesis, introduces the problems in irrigation management, defines its objectives and layout of the thesis.

Chapter 2 (Literature Review) describes overview of work done by the previous researchers considering use of GIS in irrigation management.

Chapter 3 (Theoretical Consideration) deals with theoretical background about Geographical Information System, ArcGIS and its customization and Visual Basic for Application development environment.

Chapter 4 (Material and Method) describes study area, data acquisition and software used during the study. This chapter also provides the description of methodology used to develop the tool and its application for development of the output maps.

Chapter 5 (Implementation and Results) gives the general description and feature of the developed software and results obtained from the present study.

Chapter 6 (Summary and Conclusion) presents the conclusion of the study and scope for future work.

CHAPTER 2

LITERATURE REVIEW

This chapter includes study of Irrigation Management tools/systems developed in past to accomplish Irrigation Management problems. A Brief Review of Customization of ArGIS is also included in this chapter. This chapter also includes importance of GIS and integration with Irrigation Management tool/system for improvement of Irrigation Management. This chapter also highlights review on GIS based and Non-GIS based irrigation management tools/systems.

2.1 Irrigation Management Tools/Systems

The management of irrigation system involves large number of data, alternative strategy and tactics resulting introduction of computer and its applications in better irrigation management Although the rate of development of computer based tools in water resource and irrigation management sector is very slow as compared to other sectors, few tools models and decision support systems has been developed in past.

Implementing modern management practice and alternative strategy is not the solution for problem. For this reason decision support systems are developed to help in decision making during the management practice. To get better decisions and proper way for management practice many decision support systems/ tools on water management were published that are: Suryavanshi and Reddy (1986) gives uses of decision system to find out most desirable operational schedule for irrigation distribution system; Yamashita and Walker (1994) presented a system which can predict optimal command water demand and assists in operating irrigation delivery system; Hales and Burton (2000) presented diagnostic analysis and performance enhancement of an irrigation scheme with the use of IRMOS model in Jamaica; Nixon et al. (2001) provided solution for optimization of off-farm irrigation scheduling by applied a genetic algorithm. Food and agriculture organization (FAO) had also come up with a decision support system for managing irrigation scheme (SIMIS) (Mateos et al., 2002; Lozano and Mateos, 2008).

The irrigation management tools developed in past decades has no GIS integration. These tools generally bounded to small command and farm level. The tools without GIS integration are computation of irrigation requirement and irrigation scheduling. The irrigation scheduling simulation model ISAREG tool can simulate irrigation schedule for a given climate, crop and soil (Teixeira and Pereira, 1992) have no integration with GIS.

With the use of procession capability of GIS and its applications, recent irrigation management has its integration with GIS. This integration allows simulating water requirement spatially and with large area for better irrigation management. This type of tools can handle spatially distributed data for input and output data that generally dealt with irrigation management. The GIS based tools have advantages while dealing with remote sense data. GISREG (Fortes et al., 2005), SIMIS (Mateos et al., 2002), Web based GIS (Dhakal, 2010) have GIS integration. Table 2.1 presents some of the tools with their aims, features and integration of GIS.

No	Name	Researcher/ Organization	Area	Goal (Aim)	GIS integration	Key Features
1	ISAREG	Teixeira and Pereira, 1992	Different part of the world	Simulate irrigation schedule for soil-climate-crop combination, evaluation of selected irrigation schedule	No	Decision support in selection of suitable irrigation schedule, applies to field scale and not suitable for large area
2	GISAREG	Fortes et al., 2005	Syr Darya basin, Uzbekistan	Simulate irrigation schedule for region/project , help in project management	Yes	simulate irrigation schedule in different water management scenarios for region/project scale, can be visualize spatial distribution of water demand
3	Scheme irrigation management information system (SIMIS)	FAO,2002	Different part of the world	Facilitate water and day to day management, manage irrigation water delivery schedule, help in integrated management of irrigation project.	Yes	Simulate different crop and irrigation scenario for water delivery schedule, compare existing situation for improvement irrigation, can be visualized input/output.
4	A Web based GIS	Dhakal, 2010	Alentejo region, Portugal	Share weather and evapotranspiration information to farmers through internet to support irrigation management.	Yes	Publishes weather and evapotranspiration map from interpolation of automatic weather data through web.
5	A GIS based interactive tool for irrigation	Pervez and Hoque; 2008	Bangladesh	Help managers in irrigation planning and management at project level, facilitate the	Yes	User friendly graphical user interface, decision support in design stage, real time

	management			operation and management processes of command area.		analysis of irrigation components.
6	GIS based irrigation management system	Todorovic and Steduto, 2003	Apulia region, Italy	To support irrigation authority in evaluation of irrigation scenarios under different soil, climatic and management conditions,	Yes	identifying the areas with water deficit, applied on farm to region level, spatial irrigation requirement computation, ArcGIS data can be used in input/output.
7	GIS-based Decision Support System	Raut et al., 2008	India	conjunctive irrigation management for basin planning	Yes	Handy tool for basin planning, decision support in real time operation of canal system.
8	A GIS-based decision support system for real time water demand estimation in canal irrigation systems	Rao et al., 2004	India	Decision support for real time water demand computation and canal operation in canal irrigation system.	Yes	Computes the field irrigation requirement and water requirement on distributaries on real time, helps in decision making in operation of canal.

Table 2.1: Details of tools/systems with their features, location and application.

2.2 GIS in Irrigation Management

GIS have decisive influence in land and water management for agriculture purpose. A spatial approach such GIS is particularly appropriate for the handling the spatial data in irrigation management. Now a day's GIS software is effectively and efficiently used in water resource management worldwide. Use of GIS in irrigation management especially for computing spatial water and irrigation requirements with their large volumes of spatially and temporally distributed data is widely used. The GIS capability to integrate spatial data, integrating remote sensing data and handling large volume of data has been popular among the irrigation experts. Utilization of geo-referenced data for irrigation management was unacceptable without GIS technology. Over the last decade, rapid advances in computer hardware and software, combined with the development with extensive digital database, have encouraged the application of GIS in irrigation management. Due to its powerful features of integrating and handling spatial database of rainfall, soil, land use, geology, topography, demography transportation and socioeconomic can be implemented for better findings in irrigation planning and management (Lin et al., 2004). For an irrigation project of Bihar, India Rao et al. (2004) developed real time estimation of water demands in delivery system with the aid of GIS based decision support system for soil, crop and weather data. Ray and Dadhwal (2001) used remote sensing and GIS for estimation of the crop evapotranspiration of command area of Mahi Right Bank Canal (MRBC) command in Gujarat, India.

Spatial estimation of regional crop evapotranspiration was described by Hashimi et al. (1994) for the study area of Poudre Basin in Colorado. Muthanna and Amin (2003) illustrated improve irrigation planning in Yemen with the use of A GIS application.

In past, many research studies have been carried out for estimation of irrigation water requirement, mostly focused on finding water requirements on different climate, soil and crop scenario. For the irrigation scheduling and modeling in the field and farm level large number of tools, models and decision support systems are available. Though these tools helped in advancement in irrigation management in specific aspects, many number of models raises doubt about their general validity (Lozano and Mateos, 2008). Most of these available tools are bounded to estimation of crop water requirements, simulate soil water balance, compute the irrigation schedule and evaluate the existing irrigation scheduling on field level. Farmers and irrigation advisory services have been supported by this type of irrigation scheduling tools and models as it is especially useful in farm levels (Ortega et al., 2005). Main focus of these models was to simulate irrigation scheduling with alternate irrigation schedule at different crop stage and to simulate with different water availability. Few tools have been integrated with GIS for estimation of irrigation requirement to extend the work from farm scale to region level along with better irrigation management capabilities. Some of the models are integrated with GIS which expands its analysis from farm scale to region scale, enabling water resource planning and environmental studies.

2.3 GIS for Geo-Spatial Database Creation

In irrigation management, planning and decision making are very difficult without help of GIS software. GIS helps in consolidation of information as well as visually analyze complete picture of situation. The GIS support in irrigation management can be expected when there is geo-spatial database for any command area. Naidu and Giridhar

(2011) prepared Geo-Spatial Database Creation for Wazirabad Canal Command Area with the aid of GIS.

In this study, with the help of GIS techniques from the Digital Elevation Model and topographic map, delineation of chak & block boundaries were carried out. For improving water use efficiency & allocation of water strategies, delineated boundaries were used as base. It will also help in planning and allocating of water resources to make inter canal comparison. The delineated block boundaries were elaborated with existing canal network, drainage and topography in GIS platform. This will help in improving planning management and water resource allocation to its corresponding blocks in command area to irrigation managers.

2.4 ISAREG

ISAREG is a tool made for simulation of irrigation schedule for a present climate, soil & crop (Teixeira and Pereira, 1992). This tool is also used as a model for soil water balance estimation. ISAREG model uses weather, soil & crop data as input and computes crop water requirement for selected study area (Teixeira and Pereira, 1992; Liu et al., 1998).

ISAREG model works with aid of two programs, EVAP56 and KCISA. EVAP56 uses Penman-Monteith method or uses any other method reported by FAO 56 in their guideline for estimation of evapotranspiration (Allen et al., 1998) whereas, KCISA uses method reported by FAO (Allen et al., 1998) and estimates the time averaged crop coefficient for different crop season. ISAREG model accomplish soil water balance for given study area with water content provided by user. This model also gives irrigation scheduling for given study area with aid of crop stress provided by user. ISAREG tool is useful for computing crop irrigation requirement and selecting the irrigation scheduling to support field irrigation. This model is used in many parts of world successfully from long time but it is appropriate for computation at field scale with specific soil, crop and climate condition (Fortes et al., 2005). While applying for large scale due to various numbers of combinations of field and crop characteristics requirement it cannot worked properly. So the key limitation of model is that it cannot handle spatial data which is main input for handling in region scale. To overcome this limitation, there is need of integration of model with GIS to handle spatial data for input and output. With the advance in remote sensing and the availability of remotely sensed data, the interest of this type of integration has increased (Fortes et al., 2005).

2.5 GISAREG

GIS has capability to handle spatial and non spatial data for large area. Under this curriculum to overcome limitations of ISAREG, it had been integrated with GIS and new model named GIAREG had developed for irrigation scheduling over large region or project to improve irrigation management (Fortes et al., 2005). It can compute crop water requirement and irrigation scheduling with spatial climate data of cropped fields and user selected crop scenario. A simulation result map is shown in figure 2.1 (Fortes et al., 2005).

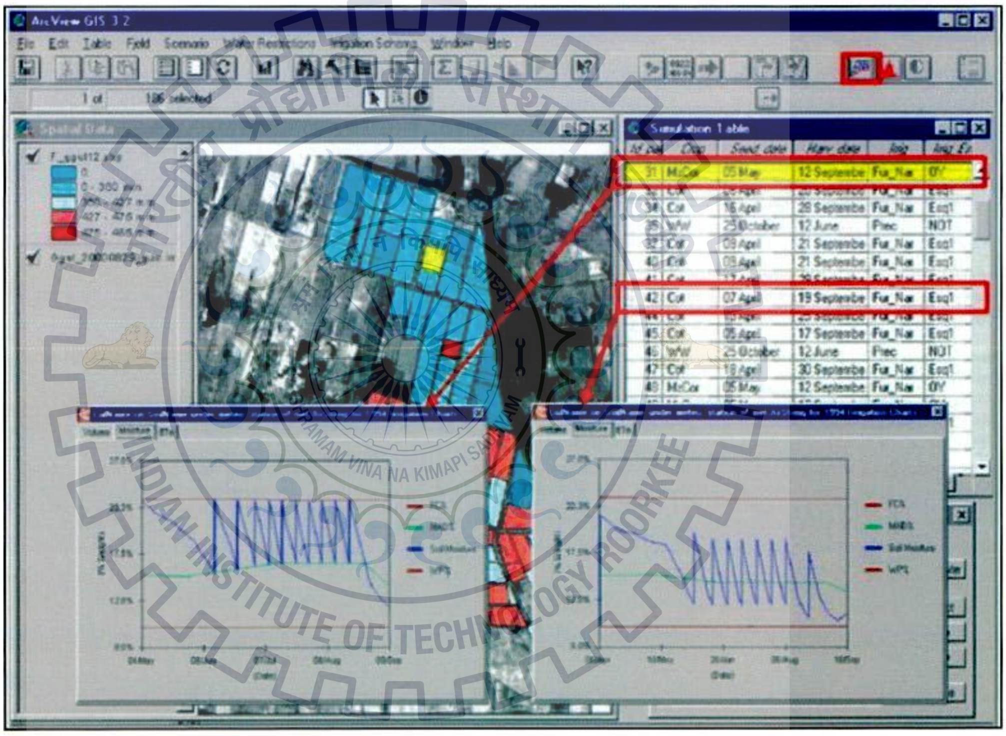

Figure2.1: simulation map showing crop irrigation requirement and water balance chart

The GIS component of the GISAREG allows different simulation scenario regarding spatial distribution of crop, irrigation method, water restriction and irrigation scheduling options. The GIS option in GISARGE allows following processes.

- Input the spatial and non spatial data.
- Creates input files required for the KCISA and ISAREG
- Creation of layer spatial database to identify characteristics of cropped field of soil and climate.

- Calling KCISA and ISAREG for computation of irrigation requirement parameters and irrigation scheduling with user selected crop scenario.
- Mapping the result outputs.

Further improvement on GISREG would be addition of groundwater components allowing its impact on water requirements (Liang and Wu, 2012). Further GISAREG can be updated with newer versions of ArcGIS as it uses Arcview3.2, which is old version of ArcGIS.

2.6 Scheme irrigation management information system (SIMIS)

A decision support system based on simple water balance model for managing irrigation schemes at farm level and canal level developed by FAO which is known as Scheme irrigation management information system (SIMIS) (Mateos et al., 2002). Initially SIMIS started as DOS based information system developed for irrigation managers to help them in their daily tasks of irrigation scheme. SIMIS was developed with the aim of providence of day to day planning, scheduling, maintenance, administration and perform activities on any irrigation scheme. SIMIS has capability to store climate, soil, crop and irrigation data and estimate water requirements, irrigation needs and further water delivery schedule in irrigation scheme.

As SIMIS can store and predict the future need it also gives capability to user to simulate delivery schedule, see the results and test new alternative until the satisfactory condition. In broader sense SIMIS can help in integrated management of irrigation project. The SIMIS is very helpful for the scheme manager for deciding irrigation schedule in different water demand condition. The SIMIS is no longer supported by FAO but it established a framework for developing customized tool to adopt particular feature of irrigation project (Lozano and Mateos, 2008).

2.7 GIS based interactive tool for irrigation management

A GIS based interactive tool for irrigation management was developed (Pervez and Hoque; 2008) for the Meghna-Dhonagoda Irrigation Project (MDIP), Bangladesh. This is an Arcview based GIS user friendly graphical user interface prepared with the Avenue Codes by integrating the GIS and Relational Database Management System (RDMS).This tool helps irrigation managers at project level for planning and management. The information system assists managers or specialists in decision support especially in design stage. It also gives capability to user to assess the impact of design parameters of system and also calculate irrigation efficiency at field level. It can perform real time computation and analysis of irrigation components and capable to generate outputs in form of maps.

Another GIS based tool was developed in Apulia region, Italy for handling spatial and non spatial irrigation data for evaluation of the irrigation scenarios under different soil, climatic and management conditions (Todorovic and Steduto, 2003). The aim of the system was to give support for irrigation authorities on evaluating irrigation scenario and find out the areas with water deficit. GIS was customized with Avenue programming language to design and develop, the irrigation water management system, a new tool for modeling irrigation water requirements. Spatial irrigation requirements can be estimated by taking into account a different scenario of cropping pattern, climatic conditions, and applied irrigation method, volume of water available for irrigation and hydraulic characteristics of the water distribution system. As ArcGIS database is used as an input in this system and integrated for the irrigation computation, outputs can be shown in the form of maps. Figure 2.2 shows seasonal gross irrigation requirement (mm) in Apulia region by GIS Based tool presented in his paper. The system has wide area of application from farm scale to region scale.

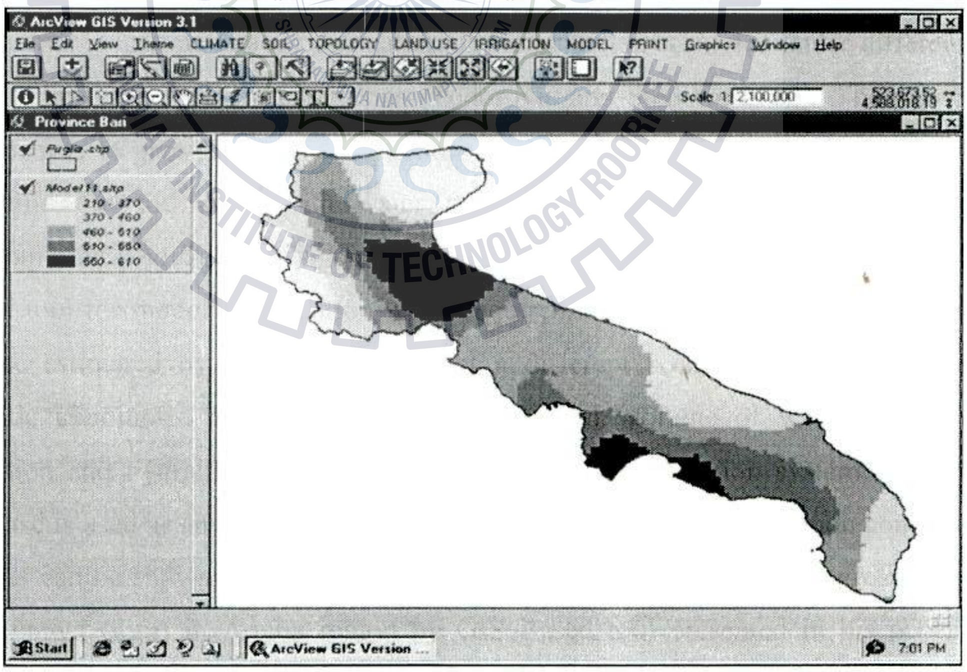

Figure 2.2: Seasonal gross irrigation requirements (mm) in Apulia region by GIS Based tool.

2.8 GIS-based Decision Support Systems

Raut et al. (2008) used a GIS-based Decision Support System developed at the Centre for Research in Water Resources (CRWR) in conjunction with ESRI for conjunctive irrigation management in India for basin planning of Jaunpur Branch Sub-basin format for hydrologic applications including all spatial and time series data. It is handy tool for basin planning and real-time operation of canal systems, particularly run-off river system. Various scenarios can be evaluated related to change in canal operation and management, agriculture practice, land use, climate and so on.

Rao et al.(2004) developed a GIS-based decision support system which estimates water demand for real time in canal irrigation systems in India. This decision support system computes the field irrigation requirement and water requirement on distributaries on real time basis. The system uses current season data, weather forecast, crop and soil data to compute water and irrigation requirement and can compute water delivery required in distributary in canal network. The system allows users to select distributary and real time computation of water demand over the area covered by distributary network and further helps in decision making in operation of canal. The ability of the system to quantify the water delivery required for canal network has strong decision support in canal operation and overall irrigation management.

2.9 Remote sensing with GIS in irrigation management

Remote sensing is the only technology due to which huge advancement takes place in water resources management along with various disciplines. Remote sense data gives valuable information on topography, land use/cover, geological feature with the aid of other traditional data which is useful in irrigation planning and management. It also provides unique handling spatial and temporal irrigation data for better irrigation management. While dealing with large surface/area for irrigation management, remote sensing is the only tool which gives precise outcomes with minimum efforts.

Remote sensing data had been used for estimation of actual evapotranspiration and crop water stress during the eighties (Bastiaanssen and Bos, 1999). Information related to irrigated area, crop type, biomass development, crop yield, crop water requirements, crop evapotranspiration, performance diagnosis, salinity and water logging has been provided by remote sensing with varying degrees of success and accuracy. (Choudhury et al., 1994; Bastiaanssen and Makin., 2000).

In India use of remote sensing technology for irrigation management has been started by previous researches (Ray and Dadhwal, 2000; Ray et al., 2002). It can be used for assessment of water availability in reservoir as well as to find out water logging and salinity problems in irrigated land to make optimal use of water for fulfilling the irrigation demand. Performance and evaluation of irrigation land can be carried out with remote sensing for identification, inventorying and assessment of irrigated crops.

Ray and Dadhwal (2000) used remote sensing and GIS for estimating seasonal crop evapotranspiration. The methodology can be used for estimating weekly evapotranspiration and a real-time irrigation scheduling. The integration of RS data and GIS tools can be used to compute performance indices (Ray et al., 2002). Regular evaluation and monitoring of performance indicator give better outcomes to any irrigation projects. Bastiaanssen and Bos, (1999) as described in his paper quantified irrigation performance indicators based on remotely sense data suggests use of remote sense data for better irrigation management.

Benchmarking (BM), using a set of defined indicators to determine the performance of various components of irrigation system is very important to evaluate the irrigation performance and the applied management for the irrigation project. Suresh et al. (2012) used various components of irrigation systems namely Irrigation Infrastructure (IIS), Agricultural System (AS) and Water Delivery Dynamics (WDD) as performance evaluation indicators for benchmarking study of the Nagarajuna Sagar Leftbank Canal (NSLC) using geospatial approach. Remote sensing technology offers benchmarking data set for cropping, water distribution and other data for comparing among the fields to evaluate the performance of irrigation or other management input. The benchmarking study can be very useful to diagnose how performance is varying along the space and facilitate quick diagnostic analysis of the problem for improved irrigation management.

CHAPTER 3

THEORETICAL CONSIDERATION

This chapter includes the theoretical consideration on geographical information system, customization of ArcGIS and crop water requirement parameters. This chapter also discusses the practice of customization of ArcGIS with the aid of VBA. Theoretical background to estimate crop water requirement and irrigation requirement, further interpolation technique used for spatial analysis is also discussed.

3.1 Geographic information systems (GIS)

GIS is a collection of computer hardware and software, data and skilled personnel for managing and analyzing geographic data (Maguire et al., 1991). In GIS data is represented in the form of points, lines, polygons or pixels, rather than just a dump map. At present, GIS technology is largely used in several fields such as agriculture management, commercial, urban, natural resource management and regional management to narrate complex and multidisciplinary planning and management problems at regional and global scales. Its capability to carry out complex spatial operations and to link spatial and non spatial data affords widespread acceptance as an important versatile tool.

The GIS was developed in Canada in the middle of 1960 by government agencies (Tsihrintzis, et al., 1996). The first known use of the term "Geographic Information System" was accomplished by Roger Tomlinson in the year 1968 in his paper "A Geographic Information System for Regional Planning" (ESRI, 2013). Later, Tomlinson was acknowledged as the "father of GIS". Despite all the technological constraints during the sixties, many of the basic techniques of spatial data handling were invented and applied during that period.The growth of various systems took place rapidly during the end of the 20th century. By this time users began to explore viewing GIS data over the Internet. Now, a growing number of free, open-source GIS packages are available and can be effectively customized to perform specific tasks. Different commercial packages of geospatial data and mapping applications are being largely available in the web.

GIS has the capability to analyzing spatial data managing many layers and integrating it from different sources. GIS also helps in spatial modeling with diverse formats, structures and projections of available data (Goodchild, 1992).GIS is capable to import

the most common data formats both for raster and vector maps. The basic functions of GIS are data collection and capturing; data storing, processing and analysis; store, quarry and analyze data; production of data; display data; produce output from the information in it.

GIS is a collectively broad term that contains number of technologies, processes, and methods. It is attached to many operations and has wider applications related to engineering, planning, management, transport/logistics, insurance, telecommunications, and business.

3.2 VBA development environment

The VBA development environment has two primary tools: 1) The Customize dialog box 2) the Visual Basic Editor. Customize dialog box is used to design interface attractively and the preparation of user forms as well as coding and debugging are carried out with visual basic editor. Because VBA is the industry standard, there are two big advantages to programming in this environment. First, if we are familiar with using VBA in a particular application, we will find it very similar (sometimes nearly identical) in other applications that support it. Second, we will be able to (easily) use object libraries between applications. When developing an application for Power Point, for example, we might choose to embed objects from the Arc Object library (or vice versa).

3.3 Customization with ArcGIS using Visual Basic Applications (VBA)

Visual Basic for Applications (VBA) assists both ArcMap and ArcCatalog which provides facility of an integrated programming environment. VBA is an excellent way to create custom commands in ArcGIS. A combo box or edit box, new button, tools (all referred to as UIControls), can be created and to each command separate code can be attached to the control's events. Once custom control is created, it can be dragged to any toolbar. Macros can also be created which can be executed from the Tools toolbar.

The Visual Basic Editor (VBE) an integrated programming environment, provided by VBA, which allows user to write a Visual Basic (VB) macro, debug and test it with in ArcMap, ArcCatalog, ArcGlobe, and ArcScene (ArcGIS Resource Center). A macro can integrate all of Visual Basic's functionality, such as user forms, input boxes, with the extensive object library that the ArcGIS application contains. ArcGIS can be customized and improve productivity using ArcGIS custom command and tools called

UIControls. Further UIControls can contain macros and can be attached into the application framework so that can be responded to actions that happen on the buttons or commands created.

All the ArcGIS tools and commands called ESRI object library can be initiated and called through the codes inside the buttons and controls. ArcGIS contains library of software components known as ArcObjects (ArcGIS Help, 9.3).The ESRI object libraries are always available for the customization in the VBA environment. These objects can be accessed via menus and buttons on the Graphical User Interface (GUI). The GUI should have some program to provide additional functionality through the objects' interfaces. The easy way to customization with VBA is to create custom commands with a combo box, or edit box, new button, tool, (collectively called UIControls), then attached code to the control's events, such as what happens when click a button. These components are typically created with development environments, such as Visual Basic 6 or .NET. (ESRI Development Network)

Using the VBA development environment, many free code resources are available online, applications can be developed to automate and speed up these tasks (ESRI Developing Network). Due to its relatively easy syntax and lack of visual clutter in the code visual basic is probably one of the easiest programming languages to learn. The Visual Basic Editor (VBE) has many resources built in to help a novice user to understand the code, including auto-complete when typing code for function properties (to avoid accessing inappropriate properties), as well as an object browser that explains object relationships and gives definitions of object properties. VBA can provide use of global variables within the document, application, and operating system so the user can create loop into the application that trigger actions based on what will happen within the user environment. With the help of robust programming environments such as VB .NET and Visual C++, users can create custom user interface forms ,more Also, close relationships of VBA with VB 6 and VB .NET allow relatively easy conversion of VBA code to VB Dynamic Link Library projects in those environments. The Visual Basic Editor interface and run-time debugging tools quickly and easily test and debug (ESRI Developing Network).

3.4 Irrigation requirements

3.4.1 Evaporation

Evaporation is the process of converting liquid water into vapour and removal from the evaporating surface (Allen et al, 1998). In evaporation process loss of liquid water from various surface such as soil, lakes, vegetation, pavements, occurs. The water first converted into vapour and losses into the atmosphere.

The ambient temperature of the air and direct solar radiation provides the energy required to convert the molecules of water from liquid to vapour. After the conversion into the vapour, the vapour pressure difference between the water at the evaporating surface and surrounding atmosphere provides the force to remove water vapour from the evaporating surface. As the evaporation continues, the surrounding air of the evaporating surface becomes gradually saturated and the evaporation process slows down. If the wet air near the surface is not replaced with dry air and transferred to the atmosphere, the evaporation process slows down and even might stop. Thus to continue the evaporation process, replacement of the saturated air with dry air is necessary. The replacement process largely depends on wind speed which helps to replace wet air with dry air. Hence, for assessing the evaporation process, important climatological parameters to be considered are solar radiation, air temperature, air humidity and wind speed.

3.4.2 Transpiration

Transpiration is the vaporization of liquid water from the plant tissues and then removal of vapour into the atmosphere (Allen et al, 1998). The losses of water from crops predominately occur through small opening on the plant leaf called stomata. Water vapour and gas passes through stomata of the plant. The plant takes water up through the roots and transports it into the plant. Most of the water taken up is lost by transpiration and only some fraction is used by the plant. The vaporization mainly occurs within the leaf, from the intercellular spaces, and the vapour exchange with the atmosphere is controlled by the stomatal aperture.

Like direct evaporation transpiration also depends on the energy supply wind speed and vapour pressure gradient. Hence, for assessing the transpiration process, important climatological parameters to be considered are solar radiation, air temperature, air humidity and wind speed. The soil water content and the water transmissibility of the soil also affect the transpiration rate. The transpiration rate is also influenced by water logging, soil water salinity, environmental aspects, crop characteristics and cultivation practices. Different kinds of plants may have different transpiration rates. While assessing transpiration the type of crop, environment and management and crop development also should be considered.

3.4.3 Evapotranspiration

Evapotranspiration is a combination of evaporation and transpiration. Evapotranspiration includes the process of water converted to water vapour through transpiration from the plant and evaporated from the soil surface. Evapotranspiration is an important parameter for the computation of the water requirement and key part of design and operation of agricultural water resource systems (Yoo et al, 2008).

Evaporation and transpiration process cannot be distinguished as they occur simultaneously. Water availability on the top soil and solar radiation reaching to the soil surface governs evaporation and transpiration amount from the surface. When the crop is small most of the evaporation takes place from soil surface depending upon the soil moisture. When the crop grows over the growing period, the crop shades the soil surface and thus evaporation decreases and the transpiration increases. The unit for evapotranspiration is usually expressed in mm/day (depth per unit time).

3.4.4 Reference crop evapotranspiration (ETo)

Reference crop evapotranspiration or reference evapotranspiration is the evapotranspiration rate from a reference surface when there is no shortage of water and is denoted as ETo (Allen et al, 1998). The reference surface is a hypothetical grass reference crop with specified characteristics. The reference evapotranspiration is computed from hypothetical crop. There are several method to compute reference evapotranspiration in existence with high degree of accuracy and applied successfully.

3.4.5 Crop evapotranspiration (ETc)

The crop evapotranspiration is the evapotranspiration taken place from disease-free well-fertilized crops, grown in large fields, under optimum soil water conditions, and achieving full production under the given climatic conditions and denoted as Etc. Crop evapotranspiration can be calculated from climatic data and ratios of ETc/ETo called crop coefficient Kc.

$$ETc = Kc \times ETo$$

Where,

ETo = reference evapotranspiration for reference crop

Kc = crop coefficient

ETc =crop evapotranspiration

3.4.6 Crop coefficient approach

In crop coefficient approach, crop evapotranspiration (ETc) is computed using crop coefficient (Kc) by multiplying it with reference evapotranspiration (ETo) (Allen et al, 1998). The equation is given by:

$$ETc = Kc \times ETo$$

Where,

ETo =reference evapotranspiration for reference crop

Kc = crop coefficient

ETc =crop evapotranspiration

The coefficient Kc adjusts difference in evapotranspiration between the reference grass surface and actual cropped surface. In the equation ETo portion incorporates the effect of the weather condition while Kc portion represents the most of the crop characteristics. The crop coefficient largely influenced by the ground cover, soil evaporation and the stage of the crop. The computation of Kc depends on four main factors: Crop type, Climate, soil evaporation and crop growth stages. All these four factors are considered during computation of the Kc value for standard climatic condition. For non standard adjustment is carried out for ETc.

3.4.7 Duel crop coefficient approach

The evapotranspiration is actually taking place simultaneously from evaporation from soil surface and transpiration from crop. In single crop coefficient approach both evaporation from soil surface and transpiration from crop are combined into single coefficient Kc. In case of dual crop coefficient approach the effects from evaporation from soil surface and transpiration from the crop are determined separately. Two separate coefficients are used in place of single coefficient Kc.

$$Kc = Kcb + Ke$$

Where,

Kcb = basal crop coefficient

Ke = soil water evaporation coefficient

In the above equation basal crop coefficient (Kcb) represents the portion of transpiration taking place from the crop and basal crop coefficient (Ke) represents the portion of evaporation taking place from soil surface. Thus from the duel crop coefficient approach crop evapotranspiration can be computed by the equation:

ETc = (Kcb+Ke) × ETo

3.4.8 Crop evapotranspiration (ETc) on non standard climatic condition.

The standard Kcb values represent Kcb for standard climatic condition of sub-humid climate and with moderate wind speed. The slandered climate condition is defined as a sub-humid climate with average daytime minimum relative humidity (RHmin) ~ 45% and having wind speeds averaging 2 m/s. Adjustment for the Kcb values for mid and end season growth stage values greater 0.45 is adjusted with the following equation (Allen et al, 1998).

$$K_{cb} = K_{cb(Tab)} + [0.04\{U_2 - 2\} - 0.004\{RH_{min} - 45\}] \left(\frac{h}{3}\right)^{0.3}$$

Where,

Kcb = adjusted basal crop coefficient for mid and late season growth stage

$K_{cb(tab)}$ = standard tabulated value for K_{cmid} and K_{cend} (if greater than or equal to 0.45)

U_2 = mean daily wind speed at 2m height during the mid or late season growth stages (m/s) for $1 m/s \leq U_2 \leq 6 m/s$.

RHmin = mean value of wind speed at 2m height during mid or late season growth stages % for 20% ≤ RHmin ≤ 80%

h= mean plant height during mid or late season growth stages for 20% ≤ RHmin ≤ 80%

3.4.9 Soil water availability

Soil water availability is the water availability for the plant retained in the soil. Different soil has different capacity of water retaining. After irrigation or heavy rainfall water is drained by gravitational force. The amount of water which is held by the soil against gravitational force is field capacity. The water held by soil in the root zone is taken by the crop. As water uptake continues the water content at the root zone decreases and remaining water in the soil held with grater force making difficult to extract for the plant. A point where the plant cannot extract remaining water from soil is called wilting point.

Total water availability in the root zone is the different between the water content at field capacity and wilting point. The equation for the total available water is given by:

$TAW = 1000(\theta_{FC} - \theta_{WP}) Z_r$

Where,

TAW = total available soil water in the root zone [mm]

θ_{FC} =the water content at field capacity [$m^3 m^{-3}$]

θ_{WP} = the water content at wilting point [$m^3 m^{-3}$]

Zr = the rooting depth [m].

All the total available water again cannot be extract by plants due to large stress involved for extraction. The fraction of TAW that a crop can extract from the root zone without suffering water stress is the readily available soil water:

RAW = p TAW

Where,

RAW is the readily available soil water in the root zone [mm].

p is average fraction of Total Available Soil Water (TAW) that can be depleted from the root zone before moisture stress (reduction in ET) occurs [0-1].

3.4.10 Crop Water Requirement and Irrigation Requirement

On the cropped field water is required to compensate the evapotranspiration, thus certain amount of water is to be supplied to the crop is called crop water requirement. There is only little difference between crop evapotranspiration and crop water requirement, i.e. crop water requirement refers to the amount of water needs to be supplied and crop evapotranspiration refers to the amount of water that is lost through evapotranspiration but being the values both are identical. The irrigation water requirement is basically the difference between the crop water requirement and effective precipitation. The irrigation water requirement also includes loss of water due to leaching of salts and non-uniformity of water application.

Crop water use is directly depends on evapotranspiration (ET). The crop water use is determined by multiplying the reference ETo with crop coefficient (Kc). The crop coefficient Kc adjusts the calculated reference evapotranspiration ETo to obtain the crop evapotranspiration Etc (yoo and et al, 2008). Different crops have a different crop coefficient and results different water use.

3.5 Computation or reference evapotranspiration (ETo)

The primary requirements of developing irrigation management tool are to compute water requirement spatially. The process requires computation of evapotranspiration and reference evapotranspiration which can be calculated from large number of available method and formula. Some of the important equations widely used for the determination of the reference evapotranspiration and used in the current study have been described subsequent headings.

3.5.1 FAO Penman-Monteith Equation

FAO recommend Penman-Monteith Equation is used for determination of reference evapotranspiration. The equation requires data for large number of weather parameters. The FAO Penman-Monteith Equation for calculating ETo as given FAO 56 can be expressed as (Allen et al, 1998).

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273} u_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

Where

ETo is grass reference evapotranspiration ($mm day^{-1}$)

Rn is net radiation at the crop surface (mm/day)

$$= (1 - \alpha_s)\, H - \sigma\left(\frac{T_{xK}^4 + T_{nK}^4}{2}\right)(0.34 - 0.14\sqrt{e_d})(0.10 + 0.90\, S/S_o)$$

In which,

αs = albedo or radiation reflection coefficient for the surface = 0.23 for grass

H = incoming short-wave (global) radiation ($MJm^{-2}day^{-1}$)

σ = Stephan-Boltzman constant = $4.903\times10{-9}\ MJm^{-1}day^{-1}{}^{\circ}K^{-4}$

T_{xK} = maximum air temperature (°K) = T_x (°C) + 273

where

T_x = maximum air temperature

T_{nK} = minimum air temperatures (°K) = T_n (°C) + 273

where T_n = minimum air temperature

G is Soil heat flux density (MJ/m^2/day)

T is mean daily air temperature at 2 m height (°C)

U2 is wind speed at 2m height (m/s)

$$u_2 = \frac{4.87 u_z}{\ln(67.8 z_m - 5.45)}$$

Uzis wind speed at zm height (m/s)

es is actual vapour pressure (Kpa)

ea actual vapour pressure (kpa)

Δ slope vapour pressure temperature curve (kPa/°C)

$$= \frac{4098 e_s}{(T + 237.3)^2}$$

in which,

e_s = saturation vapour pressure (kPa) at temperature T (°C)

$$= 0.6108 \exp\left(\frac{17.27T}{T + 237.3}\right)$$

γ is psychrometric constant (kPa/°C)

$$= 1.629 \times 10^{-3} \frac{P}{\lambda}$$

in which,

P = atmospheric pressure (kPa) at elevation z (m) above sea level

$$= 101.3\left(\frac{293 - 0.0065z}{293}\right)^{5.255}$$

where

z = station elevation (m)

λ = latent heat of vaporization ($MJkg^{-1}$)

$= 2.501 - 2.361 \times 10^{-3}$ Tm

in which, Tm = mean air temperature (°C)

3.5.2 Irmak

Irmak gave radiation based equation derived using a multilinear regression technique for computing reference evapotranspiration (ETo) (Irmak et al. 2003).This is commonly used radiation based simplified equation and can perform better than other commonly used methods. This equation requires weather data for the computation of (ETo). The equation is given by:

$$ET_o = -0.16 + 0.149 \times R_s + 0.079 \times T_a$$

Where,

ETo = evapotranspiration ($mmday^{-1}$)

Ta is mean daily air temperature at 2 m height (°C)

Rs= incoming solar radiation ~MJ m^{-2} day^{-1}

3.5.3 Tabari H

Tabari developed radiation based equation similar to the Irmak using multiple linear regressions (Tabari et al. 2011). Two equations are described and used in the study for the comparative analysis. One of the equations used in this study is given by:

$$ET_o = -0.478 + 0.156R_s + 0.0112T_{max} + 0.0733T_{min}$$

Where,

ETo = evapotranspiration ($mmday^{-1}$)

Tmax= Maximum daily air temperature(°C)

Tmin= Maximum daily air temperature(°C)

Rs= incoming solar radiation ~MJ m^{-2} day^{-1}

3.5.4 Hargreaves

Another radiation based equation for computing reference evapotranspiration (ETo) was reported by Hargreaves and Samani (1985). The equation is again modified by Hargreaves et al. (1985) which is known as Hargreaves method. The Hargreaves method was most suitablemethod for warm humid and semi-arid climatic conditions. The equation is given by:

$$ET_o = \frac{1}{\lambda} * 0.0023(T_{max} - T_{min})^{0.5}(T_m - 17.8)R_a$$

Where,

ETo = evapotranspiration ($mmday^{-1}$)

λ = Is latent heat of vaporization ($MJ\ kg^{-1}$)

= 2.501-0.002361×Tm

Tm =mean air temperature

Ra = extraterrestrial solar radiation received on earth's surface ($MJ\ m^{-2}$ per day).

Tmax = Maximum daily air temperature (°C)

Tmin = Maximum daily air temperature (°C)

Tm = Mean daily air temperature (°C)

3.5.5 Hargreaves M1

Droogers and Allen (2002) reported three new types of the Hargreaves equation (Hargreaves and Samani 1985) as follows:

Hargreaves Mod-1

$$ET_o = 0.408 \times 0.0030 \times (T_a + 20) \times (T_{max} - T_{min})^{0.4} \times R_a$$

Hargreaves Mod-2

$$ET_o = 0.408 \times 0.0025 \times (T_a + 16.8) \times (T_{max} - T_{min})^{0.5} \times R_a$$

Hargreaves Mod-3

$$ET_o = 0.408 \times 0.0013 \times (T_a + 17) \times (T_{max} - T_{min} - 0.0123P)^{0.76} \times R_a$$

Where,

ETo = evapotranspiration ($mmday^{-1}$)

P = monthly rainfall (mm).

= is the daily extraterrestrial solar Radiation ($mm\ day^{-1}$)

Tmax = Maximum daily air temperature (°C)

Tmin = Maximum daily air temperature (°C)

Ta = average daily air temperature (°C)

The equations are named as Hargreaves M1, Hargreaves M2 and Hargreaves M3 respectively by Tabari et al (2011). Only the Hargreaves M1 is considered in this study.

3.5.6 Turc-Radiation

Turc (1961) introduced a method for computing evapotranspiration using radiation and temperature data. The equation is known as Turc-radiation method. The equation is given by:

$$ET_o = \frac{1}{\lambda} * 0.013 * \frac{T}{T + 15} * (Rs + 50)$$

Where,

ETo = evapotranspiration (mmday^{-1})

T = mean daily temperature for the month (°C)

λ = Is latent heat of vaporization (MJ kg^{-1})

= 2.501-0.002361*Tm

Tm =mean air temperature

Rs= incoming solar radiation Cal cm^{-2}

3.5.7 Blaney Criddle

Blaney criddle is a theoretical method used to estimate reference evapotranspiration as an average for a period of one month. This method gives rough estimate or "order of magnitude" only. Under the extreme climatic conditions it gives inaccurate results. In windy, dry, sunny areas reference evapotranspiration is underestimated where as under clam, humid and clouded areas reference evapotranspiration is overestimated, however due to ease of calculation this method is used in many areas. The equation used for estimation given by,

$$ETo = P\ (0.46 \times T_{mean} + 8)$$

ETo = Reference crop evapotranspiration (mmday^{-1})

T_{mean} = mean daily temperature (°C)

P = Mean daily percentage of annual daytime hours

Mean daily percentage of annual daytime hours depends upon latitudes of the location; respected values of mean daily percentage of annual daytime hours are given in table 3.1

Lat.	N	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov	Dec
	S	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	March	April	May	June
60^0		0.15	0.20	0.26	0.32	0.38	0.41	0.40	0.34	0.28	0.22	0.17	0.13
55^0		0.17	0.21	0.26	0.32	0.36	0.39	0.38	0.28	0.28	0.23	0.18	0.16
50^0		0.19	0.23	0.27	0.31	0.34	0.36	0.35	0.32	0.28	0.24	0.20	0.18
45^0		0.20	0.23	0.27	0.30	0.34	0.35	0.34	0.31	0.28	0.24	0.21	0.20
40^0		0.22	0.24	0.27	0.30	0.32	0.34	0.33	0.30	0.28	0.25	0.22	0.21
35^0		0.23	0.25	0.27	0.29	0.31	0.32	0.32	0.30	0.28	0.25	0.23	0.22
30^0		0.24	0.25	0.27	0.29	0.31	0.32	0.31	0.29	0.28	0.26	0.24	0.23
25^0		0.24	0.26	0.27	0.29	0.30	0.31	0.31	0.29	0.28	0.26	0.25	0.24
20^0		0.25	0.26	0.27	0.28	0.29	0.30	0.30	0.28	0.28	0.26	0.25	0.25
15^0		0.26	0.26	0.27	0.28	0.29	0.29	0.29	0.28	0.28	0.26	0.26	0.25
10^0		0.26	0.27	0.27	0.28	0.28	0.29	0.29	0.28	0.28	0.27	0.26	0.26
5^0		0.27	0.27	0.27	0.28	0.28	0.28	0.28	0.28	0.28	0.27	0.27	0.27
0^0		0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27

(Chapter 3- Crop Water Needs, FAO, Natural Resources Management and Environment Department)

Table3.1: Mean daily percentage of annual daytime hours

3.6 Inverse Distance Weighted (IDW) interpolation

For Interpolation of scatter points most commonly used method is Inverse Distance weighted (IDW). This method is based on assumption of interpolating surface is influenced more by nearer points and less by more distant points (ArcGIS Resource Center). To compute a value of an interpolated point by IDW technique, neighborhood about the interpolated point is identified and weighted average of these neighborhood points is taken. The weighting function of the neighbor and the distance can be controlled by the mathematical function of the weighting function and the size of the

neighbor. The weight for the neighbor and the distance of IDW function is controlled by "The power function" and "The search neighbored".

The weights given to the neighbor points are proportional to the inverse of the distance between the prediction location and known points raised to the power value p. As the distance increases, the weights decrease rapidly. By minimizing the root mean square prediction error (RMSPE), the optimal power (p) value can be determined. How fast the weights decrease is dependent on the value for p. If $p = 0$, there is no decrease with distance, and more the value of p the weight decrease with distance rapidly. If the p value is very high, only the immediate few surrounding points can influence the prediction. Most commonly 2 for the p value is considered.

The shape for the consideration of the point data also influence and restricts how far to look for measured values is called directional influence. If there are no directional influences on the weighting of the data, it is considered points equally in all directions. In this case the shape of neighborhood is considered to be a circle. If there is directional influence, more weight is given for one side then other side, the shape of the neighborhood is not considered to be a circle.

CHAPTER 4

MATERIAL AND METHOD

This chapter describes about the software and data used in this study. Study area and methodology used for preparation of graphical user interface is also discussed here briefly. Further calculation procedures for reference evapotranspiration, crop evapotranspiration and irrigation requirement used in developing the interface is also included in this chapter.

4.1 Software Utilized

The ARCGIS 9.3 and Visual Basic for Applications (VBA) embedded within ArcGIS 9.3 application as a development environment has been used. Customization of ArcGIS is carried out using VBA. To make interface friendly, suitable graphical user controls were used and Visual Basic scripting is used for programming to make it easy. Other Supporting software for development of the interface used were Microsoft office: Excel, Excess, Word, Note pad. Table 4.1 shows the list of software's used and their characteristics.

Table 4.1: Software's used for the work

Software / Environment	Uses
ArcGIS 9.3	Preparation of spatial data, Execution of the interface, printing the output maps.
Visual Basic for Application	Coding, Dubbing, developing user forms, user controls and menus, linking maps and database, running the programs.
MS Notepad	Storing list of commonly used crop with their standard parameters used for computation of crop coefficient.
MS Access	Storing Weather stations with their weather data

Notepad was used to store list of some commonly used crop along with basal crop coefficient their length of development stage, depth of root zone and soil water depletion fraction. These stored crops and their data can be accessed by interface during the execution of the interface. Similarly MS Access was used to store weather data such as station name, minimum and maximum temperature, Relative humidity required for computation of reference evapotranspiration. The stored data can be access through interface and the data can be used during the execution of the interface.

4.2 Study area description

4.2.1 General

Present study is carried out in the Betwa river basin (Figure 4.1). Betwa river is tributary of the Yamuna river located in central part of the India. It is an interstate river between Madhya Pradesh and Uttar Pradesh. Betwa river basin extends from the latitudes of 22° 54' 00" N to 26° 00' 00" N and the longitudes of 77° 10' 00" E to 80° 20' 00" E.

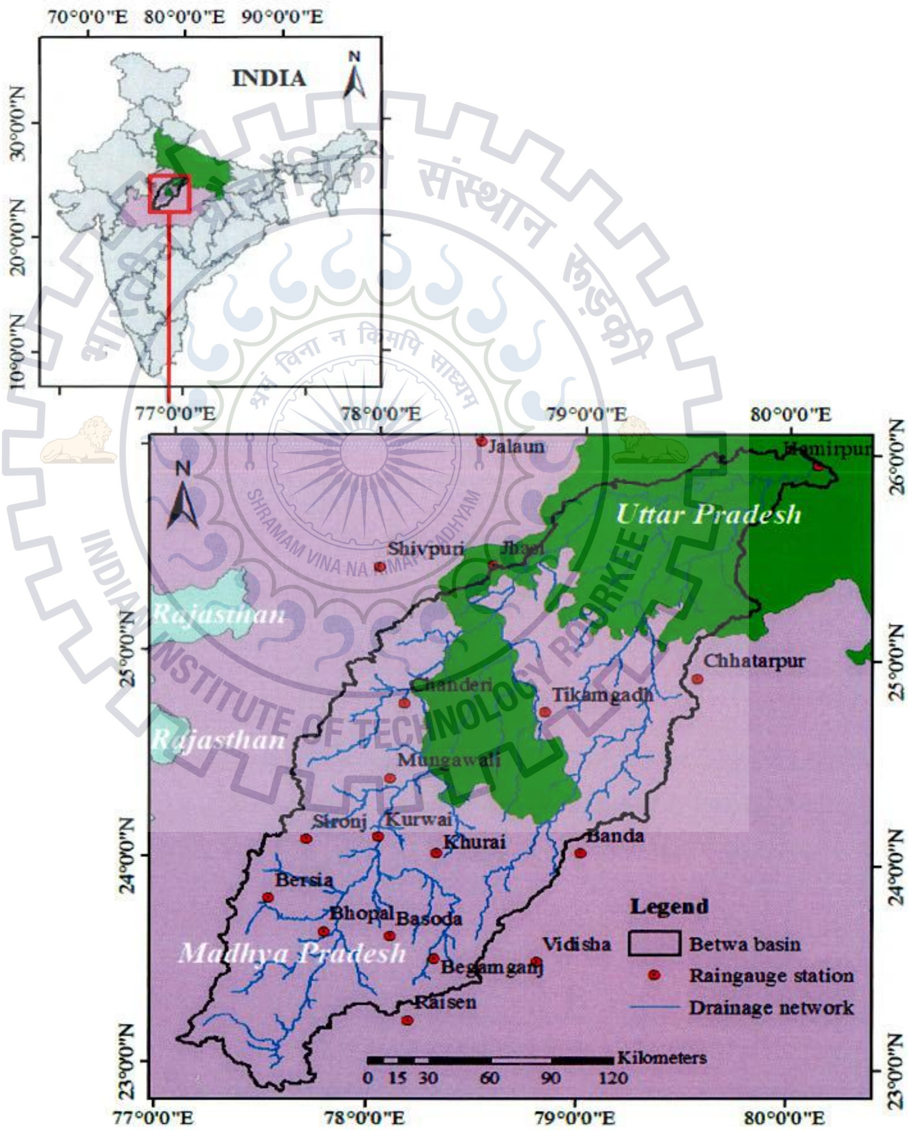

Figure 4.1: location map of study area

From origin to confluence with the Yamuna river, the total length of Betwa river is 590 km, among this 232 km lies in Madhya Pradesh and 358 km lies in Uttar Pradesh. It joins with Yamuna river near Hamirpur in Uttar Pradesh. The range of elevation is about 300 to 700 m above m.s.l. From flat open wheat- growing areas to steep forest-covered hilly areas vary with the vegetation and topography in a fairly complex pattern. Mostly area of the basin is under cultivation of wheat and gram as the main crops in the post- monsoon or winter season and millet also growing in the monsoon season. The climate of basin area is semi- arid to dry sub- humid. The air is being mostly dry in exception of south- west monsoon season. Basin has generally mild winter and hot summer climate. The average minimum and maximum temperatures are 6.7 °C and 44.2 °C respectively. The annual rainfall of the basin varies from 700 to 1200 mm and in the monsoon period (from June to October) receiving annual rainfall is about 90% of the total rainfall.

4.2.2 Data used

Spatially distributed data for the cropped field as polygon is used as input to work with interface. The spatial data for cropped field requires data for weather, soil and crop data. These weather data includes weather data of maximum and minimum temperature, relative humidity, solar radiation, wind speed. Rainfall data of nearby weather stations is also required. For the study standard wheat crop data is included in the interface. Crop days are inserted manually. Other crop data such as root zone depth, crop coefficient for respective development stage is stored inside interface database and selected during the execution of the interface. The soil type selected is sandy loam.

Table 4.1: Data set required to model irrigation requirement

Type of Data	Name of Data	Periods/Sources
Spatial data	Fields area data	Raster DEM, http://gdem.ersdac.jspacesystems.or.jp/
	Polygons	
	Boundaries	
Weather data meteorological data	Maximum/Minimum Temperature	Global Weather Data for SWAT http://globalweather.tamu.edu/
	Relative Humidity	
	Solar radiation	
	Wind speed	
	Precipitation	
Soil, Crop data	Soil type, Crop Type	NBSS & LUP, Nagpur http://mpkrishi.org/
	Crop Stage	

4.3 Development of Interface

A Graphical User Interface will be created within environment provided by ArcGIS for Customization. In this study Visual Basic for Application (VBA) has been used for development of Interface for the purpose of estimation of Irrigation water requirement. The whole interface work is divided into various parts but interlinked with each others. Interface was compiled with scripting & designing of 4 menus, 11 modulus & 4 forms. Each module was assigned with each UIControls button. With the aid forms & UIControls buttons scripting of interface were composed. For the scripting of the interface Visual Basic programming were used. Interface was implemented with weather data & crop data as input to prepare raster maps of rainfall, effective rainfall, daily evapotranspiration, daily crop evapotranspiration & net irrigation requirement. For ease of elaborating as well as for updating, editing & selection of weather & crop data forms within the framework respective modules were provided. Thematic raster maps of selected study area were made with interpolation technique using Visual Basic for Applications (VBA).

For the computation of rainfall & effective rainfall of selected study area rainfall data & rainfall coefficient data has been taken as primary components. For the computation of daily reference evapotranspiration weather data which includes maximum & minimum temperature, relative humidity, daily sunshine hours, wind velocity & latitude were considered as primary components. Seven methods were put in interface for calculation of daily reference evapotranspiration. Among the seven methods for selected study area the standard methods recommended by FAO was used for estimation of daily reference evapotranspiration.(Allen et al., 1998,Mateos et al., 2002; Yoo et al., 2008). Similarly for calculation of daily crop evapotranspiration estimation crop coefficient & daily reference evapotranspiration were took as major components. For the calculation of crop coefficient crop data, crop stage data, crop length data has been used as primary components. Finally with the support of effective rainfall, reference evapotranspiration & reference crop evapotranspiration irrigation requirement of the selected study area were estimated. Flowchart of Development of Interface (GUIIM) is presented in figure 4.2

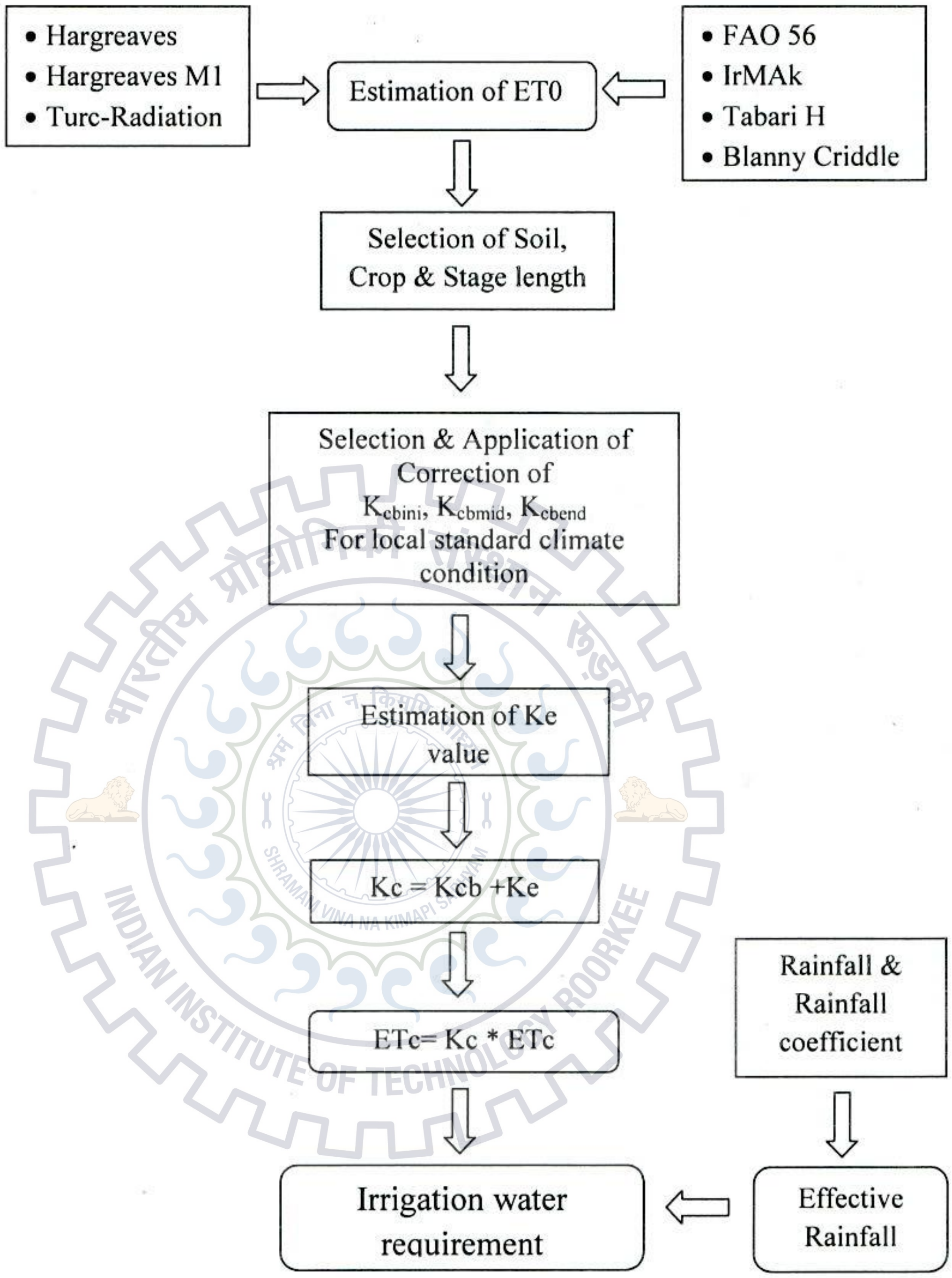

Figure 4.2: Development of Interface (GUIIM)

The point rainfall data is used as rainfall input. For calculation of the effective rainfall (P) the runoff and percolation losses were considered which may be adopted as a fixed fraction of rainfall over the whole area or as a value which varies from cell to cell depending on the amount of rainfall. Effective rainfall is calculated cell-by-cell as Peff = Pcoff * P. where Pcoff is the rainfall coefficient. For the interpolation to raster Inverse Distance Weighted (IDW) interpolation technique was used for the effective rainfall thematic map preparation. The optimal power (p) value is fixed at 2 while search neighborhood value is fixed at 12.

New toolbar named Irrigation Management was created for easy working with GIS for input and processing as interface tool. UIcontrols, buttons, input box and menus are used for better visual performance and easy interaction with users. The controls are attached with the codes which provide the computations and procedures required during the execution of the interface.

The tool contains three menus Rainfall, Reference ET (ET0), Crop ET (Etc) and one module Irrigation requirement. The Visual Basic Codes used to develop the tool has been annexed in Appendix A. The Interface tool within ArcGIS is shown in figure 4.3.

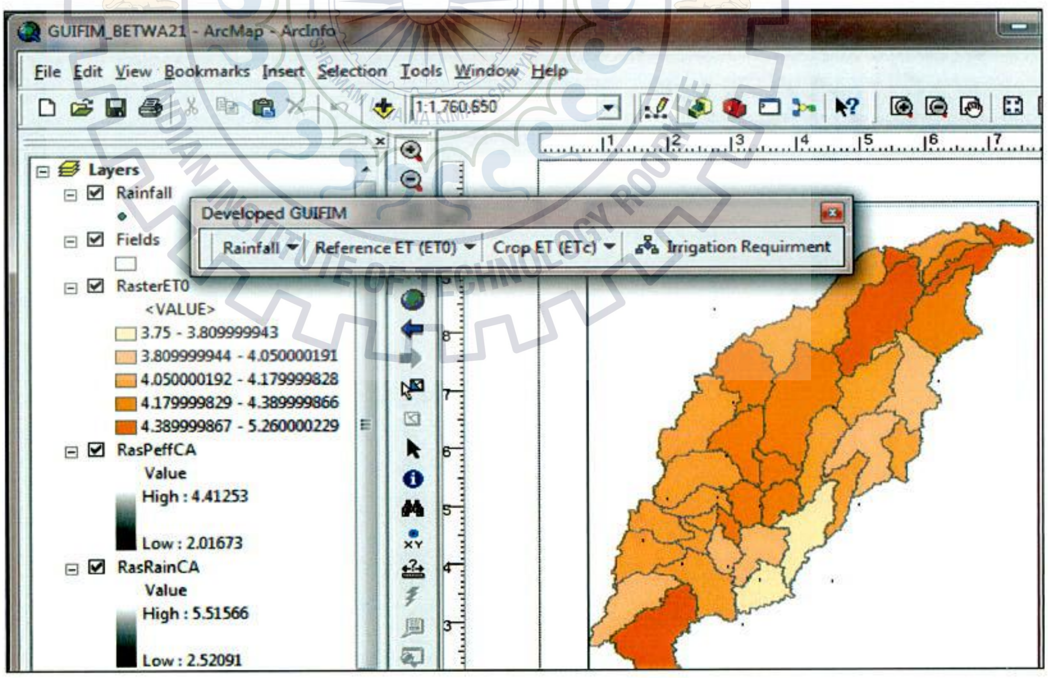

Figure 4.3: Irrigation Management Interface tool.

4.4 Data preparation

Data preparations represent preparation of the spatial distributed GIS data of the selected area with polygon of the cropped field's boundary and the spatial points of the weather stations near by the study area. Separate layers for each rainfall point data covering the field's layer and field's polygon layer is prepared using ArcGIS 9.3. Figure 4.4 shows the field's spatial distributed data for the selected area.

The Rainfall layer contains data for rainfall (mm), runoff coefficient (Pcoff) and effective rainfall (Peff) for each distributed points, so the field of column for each category is prepared and linked with attribute table of the rainfall layer. The rainfall layer contains rainfall field where rainfall data are stored, Pcoff field to store rainfall coefficient to be used for that station and Peff field to store effective rainfall data. Similarly the Field layer contains data of fields Tmax, Tmin, RHmax, RHmin, z, v, n, Lat, h, j, ET_0, ETc for weather parameters input for the computation of reference evapotranspiration. Fields named Kc and ETc is also added to store crop factor data and resulted crop evapotranspiration data.

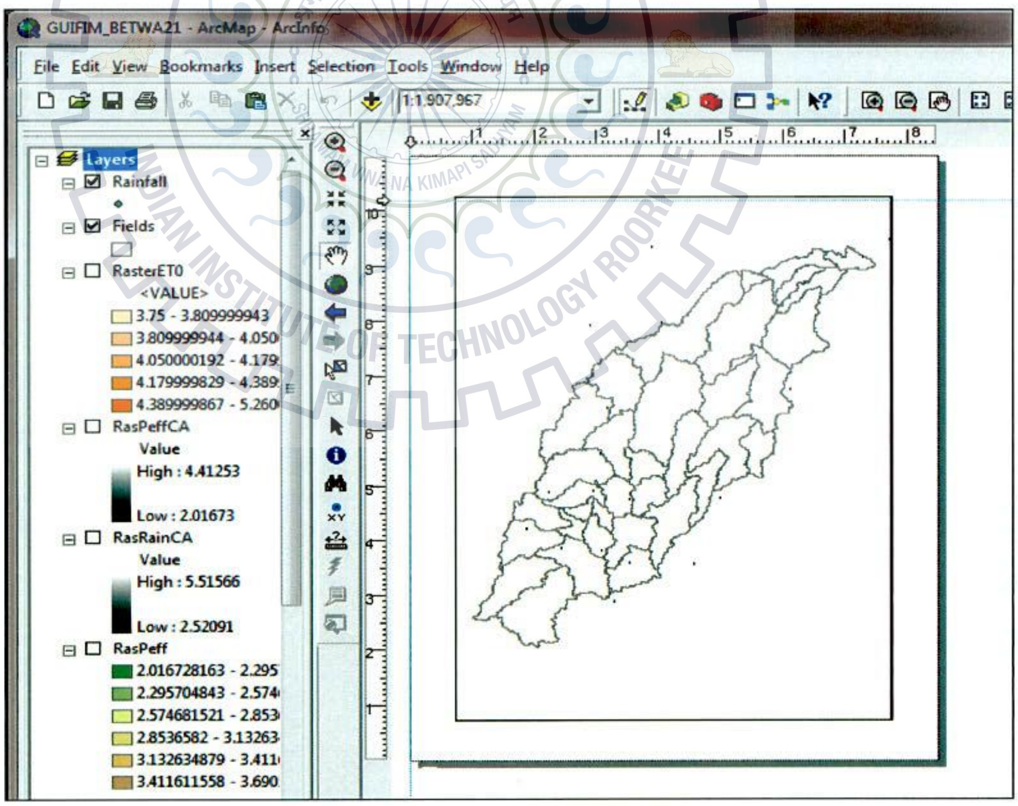

Figure 4.4: Spatially distributed database preparation.

4.5 Interface Setup and Processing

The interactive tool developed inside ArcGIS is to be used to input the data, process the data and produce output thematic maps. Setup of the interface primarily consist of input the required data, selection of the relevant data from list and process for the thematic map. The process also requires selection of appropriated method of computation of reference evapotranspiration from the list provided in the interface, selection of crop data, soil data from list and processing. Whole process is elaborated using flowchart 4.5 and subsequent subheadings.

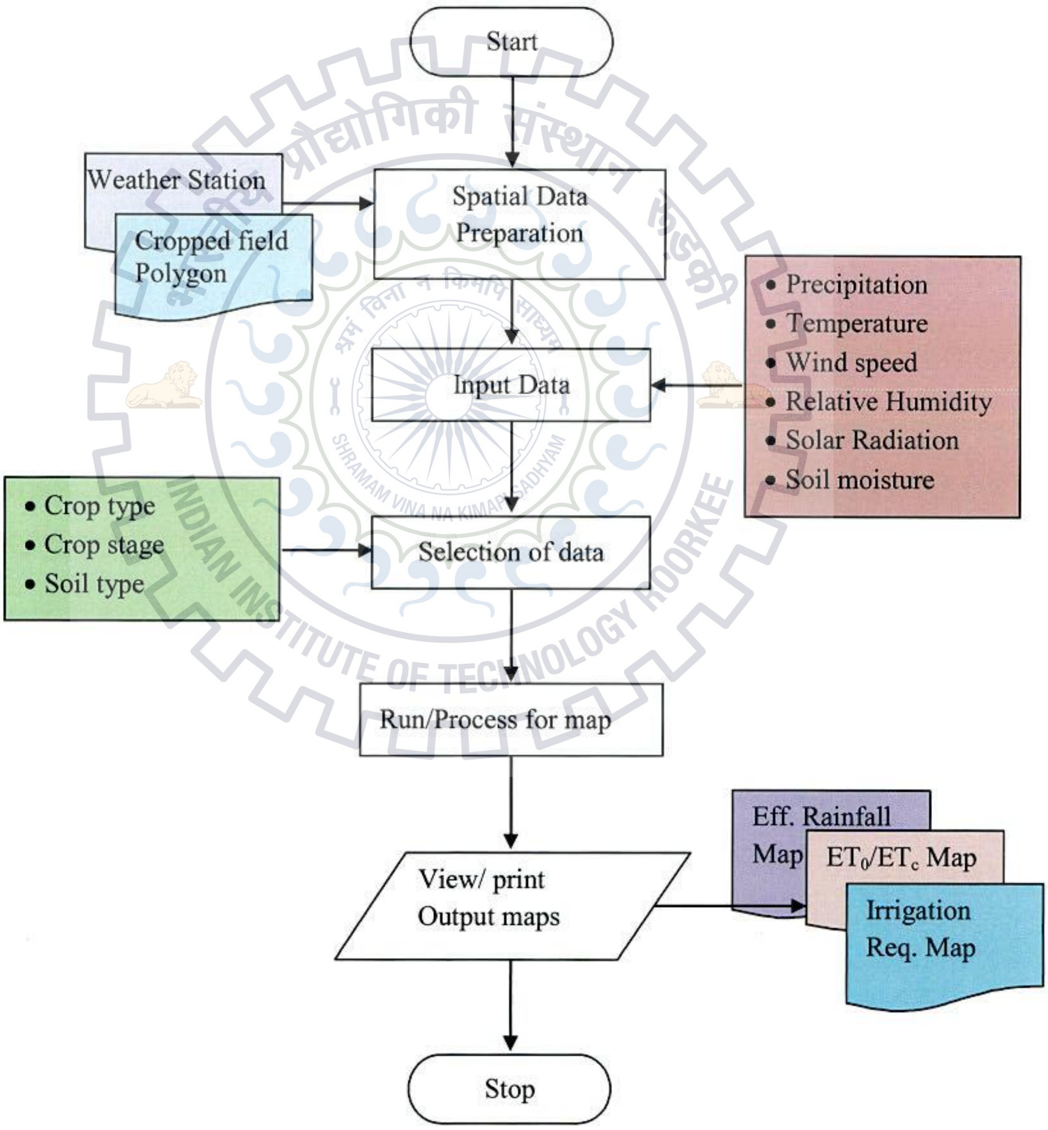

Figure 4.5: Flowchart of interface setup and processing

The climatic data, soil map data and crop database are used as input for the model. The spatial databases of irrigation system are always needed as an input and are modified by the model according to the need. The thematic maps of irrigation requirements, irrigation requirement parameters and effective rainfall maps are visualized and stored as output file for the model. Overview of the input output system for the model is shown in flowchart form in figure 4.6.

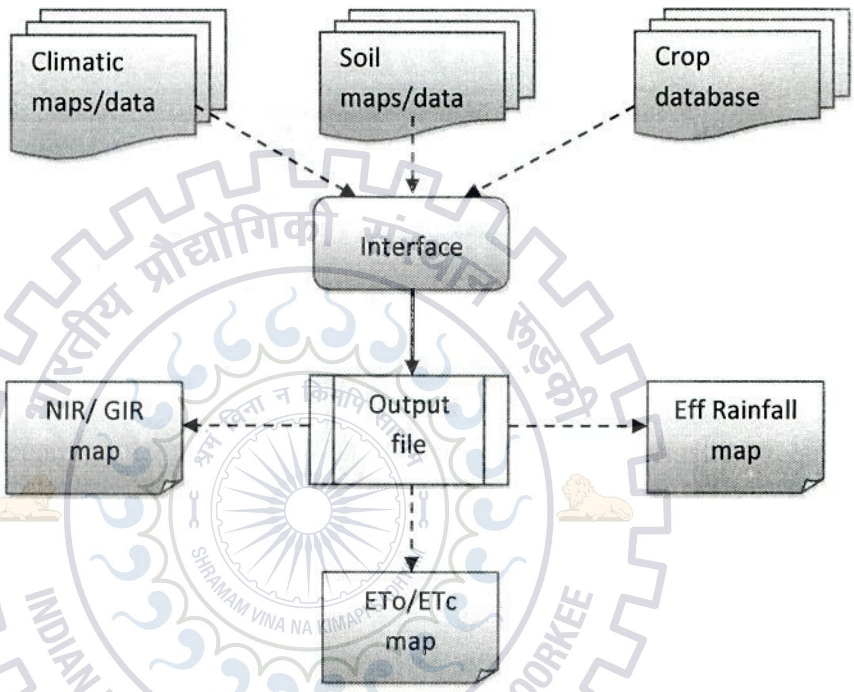

Figure 4.6: Flowchart data exchange between input/output and mechanistic model

4.6 Generation of Thematic map

Visualize and analyze theweather parameters, evapotranspiration and irrigation requirements animated maps will bevery helpful for better irrigation management. Daily weather data for December 2014 was selected to animate maps which should contain basic control functions to display mapsinteractively. To fulfill the requirement the animated mapthus created user caneasily store maps and access easily.

The point data of effective rainfall is used as rainfall input. For the interpolation to raster Inverse Distance Weighted (IDW) interpolation technique was used for the effective rainfall thematic map preparation. The optimal power (p) value is fixed at 2 while search neighborhood value is fixed at 12. The VBA code attached with Peff

Raster module on the interface tool allows generating thematic map of effective rainfall.

The computed value of ETc and ET0 data for each polygon fields is used to generate thematic raster map of reference evapotranspiration and crop evapotranspiration. For the interpolation to raster Inverse Distance Weighted (IDW) interpolation technique was used for raster map of reference evapotranspiration and crop evapotranspiration. The VBA code attached with Reference ET Raster of and Crop ET Raster module of interface tool allows generating thematic map of effective rainfall.

Irrigation requirement thematic raster map is generated using Irrigation Requirement module in the interface tool. The module uses thematic map of effective rainfall raster map and the crop evapotranspiration raster map. Subtraction of cell by cell technique is used to generate irrigation requirement map. The VBA code attached with Irrigation Management module of interface tool allows in generating thematic map of Irrigation requirement. The VBA code is annexed in appendix.

CHAPTER 5

IMPLEMENTATION AND RESULTS

This chapter covers the developed interface tool. This chapter includes detail description of each module and forms along with function of each module, forms and other control buttons within the interface. The features of the interface, the working procedures for the interface are described in this chapter. This chapter also includes the output thematic raster map of rainfall, effective rainfall, reference evapotranspiration, crop evapotranspiration and irrigation requirement generated by the interface.

5.1 The Interface Tool

The Interface tool is a main processing tool to analyze, interpolate, processing and producing output maps developed within ArcGIS using VBA. The tool contains three menus Rainfall, Reference ET (ET_0), Crop ET (Etc) and one module Irrigation requirement. The codes used to develop the tool have been annexed in AppendixA and interface toolbar is shown in figure 5.1.

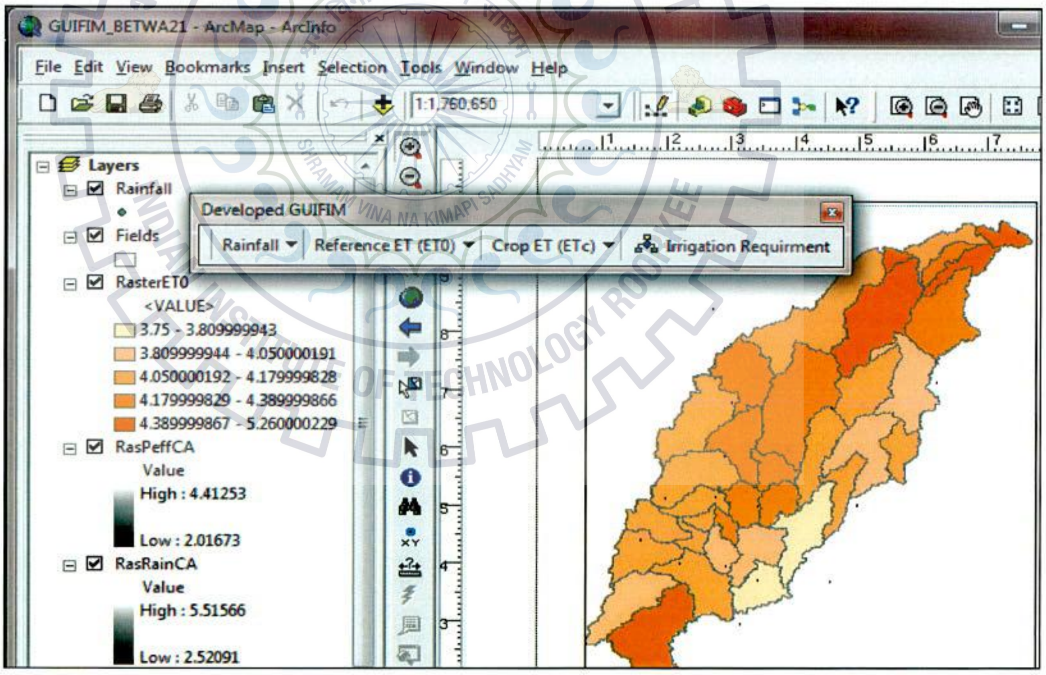

Figure 5.1: Irrigation Management Interface Tool

The Rainfall menu is used to update rainfall and rainfall coefficient data and to produce interpolated thematic raster map of rainfall and effective rainfall. Weather data required for computing reference evapotranspiration can be updated from Reference ET menu. The Reference ET menu allows selection among the seven different methods for computing reference evapotranspiration: FAO Penman-Monteith method, Irmak, Tabari H, Hargreaves, Hargreaves M1 Turc-Radiation and Blaney criddle. Setup and processing of the interface tool is already mentioned in previous chapter.

5.2 Rainfall Menu

The Rainfall menu and attached modules allows updating rainfall data and producing interpolated raster map for rainfall and effective rainfall for the selected area. This module first allows selecting rainfall layer, and allows for updating the rainfall and rainfall coefficient data through rainfall form. Further, it allows generating raster map of the rainfall and effective rainfall. The Rainfall menu contains five attached modules namely Updated Rainfall, Rainfall Raster, Effective Rainfall Raster, Rainfall Raster Fields and Peff Raster Fields.

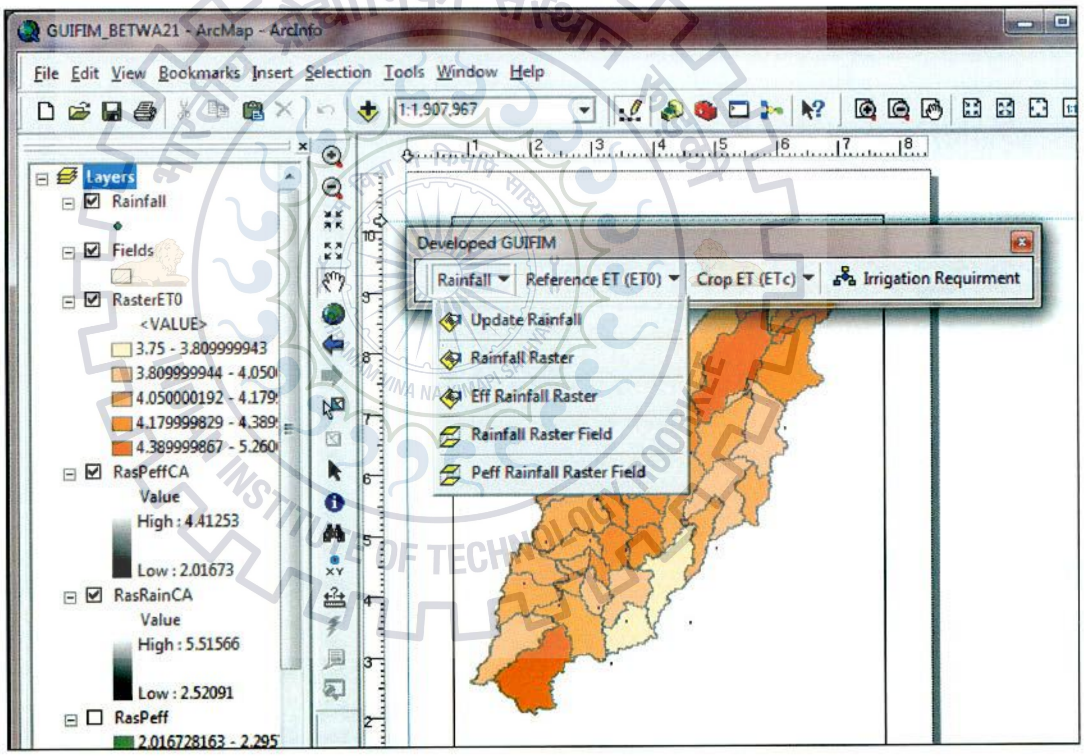

Figure 5.2: Menu Rainfall along with its modules

Update Rainfall module allows first to select layer for rainfall, shows in figure 5.3. After selecting the layer then it allows links to Rainfall form, shows in figure 5.4. The rainfall form allows changing and editing the rainfall and rainfall coefficient values. After editing the values update button allows to update and store the new values with the help of command button. The update command also computes the effective rainfall for each station and updates its value in respective cells.

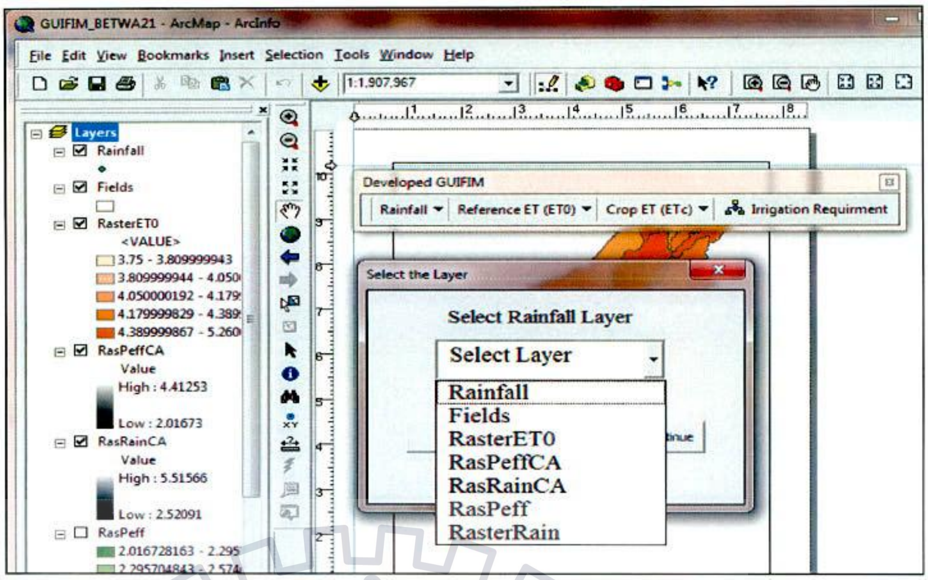

Figure 5.3: Form to select layer for rainfall

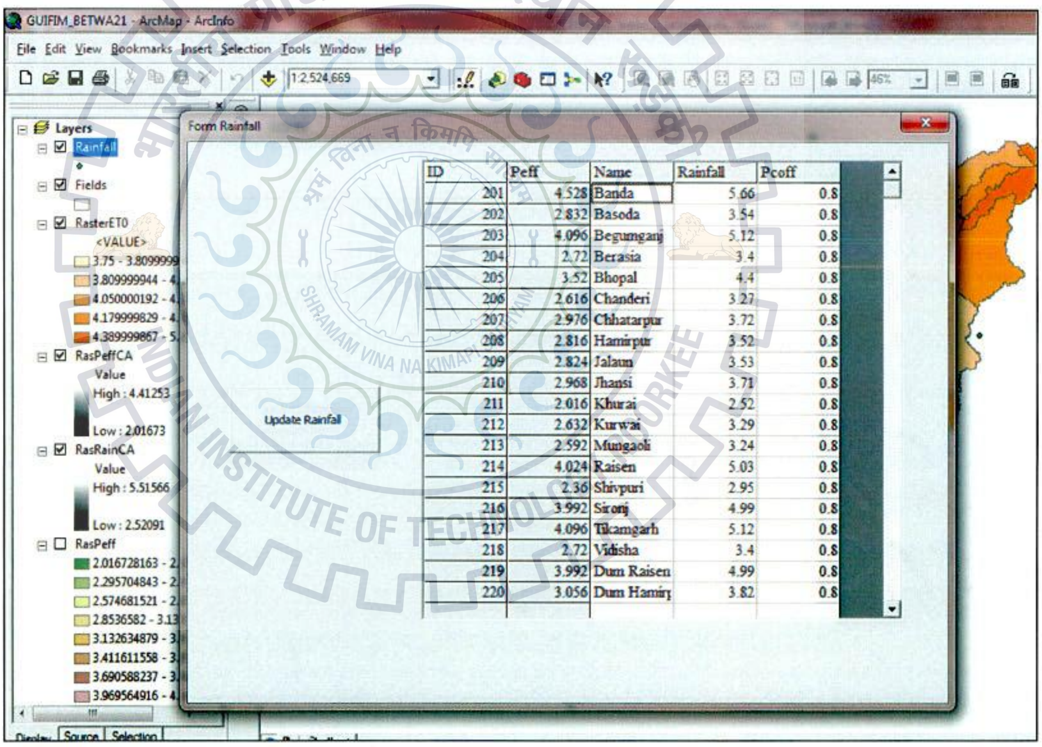

Figure 5.4: Rainfall Form to allows update rainfall parameter

Rainfall Raster module allows access to the rainfall layer and its rainfall field data. This module uses rainfall layer and its rainfall field data as input and interpolates rainfall data using inverse weight techniques to produce rainfall raster map of area included by the points.

Effective Rainfall Raster module allows access to the rainfall layer and its effective rainfall field data. This module uses rainfall layer and its effective rainfall field data as input and interpolates rainfall data using inverse weight techniques to produce effective rainfall raster map of area included by the points.

Rainfall Raster Fields module uses rainfall raster map and fields polygon as mask and extracts the Rainfall Raster map to produce rainfall raster map of the area only included by the field layer. Peff Rainfall Raster Fields module uses rainfall effective raster map and fields polygon as mask and extracts the Rainfall Raster map to produce rainfall raster map of the area only included by the field layer.

5.3 Menu-Reference ET (ET_0)

The Reference ET (ET_0) menu and its attached modules allows updating weather data required for computing reference evapotranspiration and allows producing raster map of reference evapotranspiration raster map for field layer. This module first allows selecting field layer, allows performing all calculation needed and updating the weather and data computation. Further it allows developing raster map of reference evapotranspiration. The Reference ET (ET_0) menu contains two modules namely Update Weather and Raster ET_0.

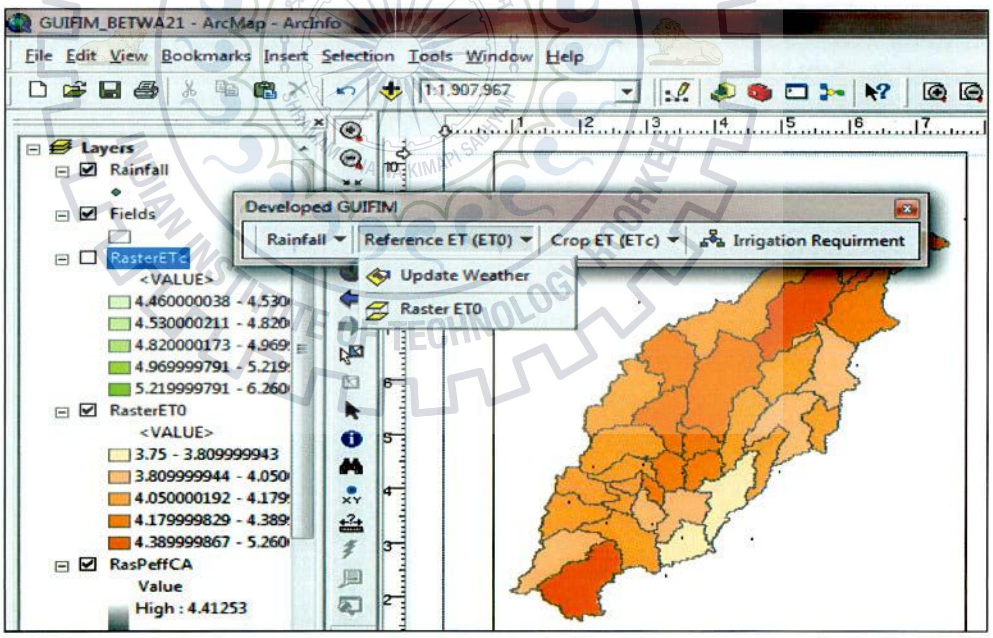

Figure 5.5: Menu Reference ET (ET_0) along with its modules

Update Weather module allows first to select layer for polygon layer of cropped field. After selecting the layer it then allows links to ET_0 form. The ET_0 form allows selecting the weather data from the one of the station already saved in database. The weather data listed in form ET_0 is loaded from the database prepared inside the interface and can be edited and saved for the further uses. The weather data can be used for each field or can be used for all fields at once. After editing the weather data update button allows to update and store the new values.

Computation of reference evapotranspiration is carried out inside this form through respective command buttons. The command button is named according to the method utilized for computation of reference evapotranspiration. Different methods of computing reference evapotranspiration can be triggered by respective command button. The user can choose the method between seven methods for the computation and use reference evapotranspiration (ET_0). The computed reference evapotranspiration values appear in the ET_0 fields in every row of the grid. The update command updates all the weather data and computed reference evapotranspiration value in the respective cells and in its database for further use.

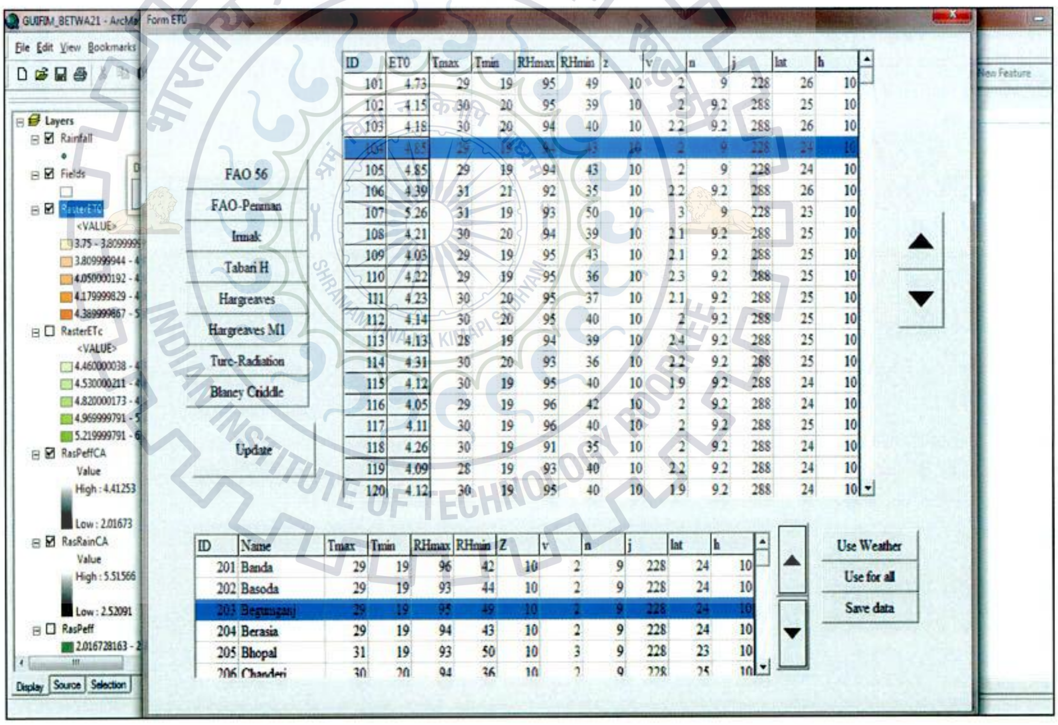

Figure 5.6: Form ET_0 to allow update weather parameter and computation of ET_0

Raster ET_0 module allows access to the field layer and its ET_0 field data. This module uses rainfall layer and its ET_0 field data as input and produce reference evapotranspiration (ET_0) raster map of area included by the field polygon.

5.4 Menu-Crop ET (ETc)

The Crop ET (ETc) menu and its attached modules allows updating and selecting the crop data, soil data and soil moisture data required for computing crop evapotranspiration and irrigation requirement. This menu further allows generating thematic raster map of crop evapotranspiration raster map and water requirement raster map for selected field layer. This module first allows selecting field layer, allows performing all calculation needed and updating the crop data, soil data and soil moisture data. Further, it allows developing thematic raster map of crop evapotranspiration and water requirement for the selected layer.

The Crop ET (ETc) menu contains two modules namely Update Crop ET and Raster ETc. Update Crop ET module allows first to select layer for field. After selecting the layer it then allows links to ETc form.

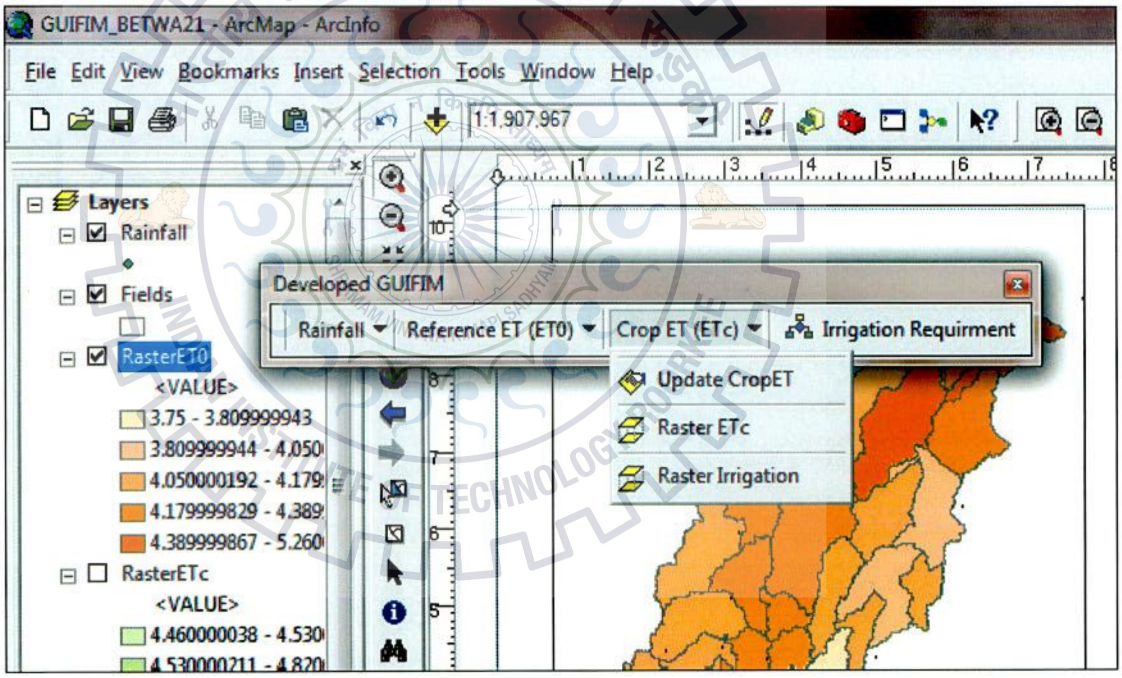

Figure 5.7: Menu Crop ET (ETc) along with its modules

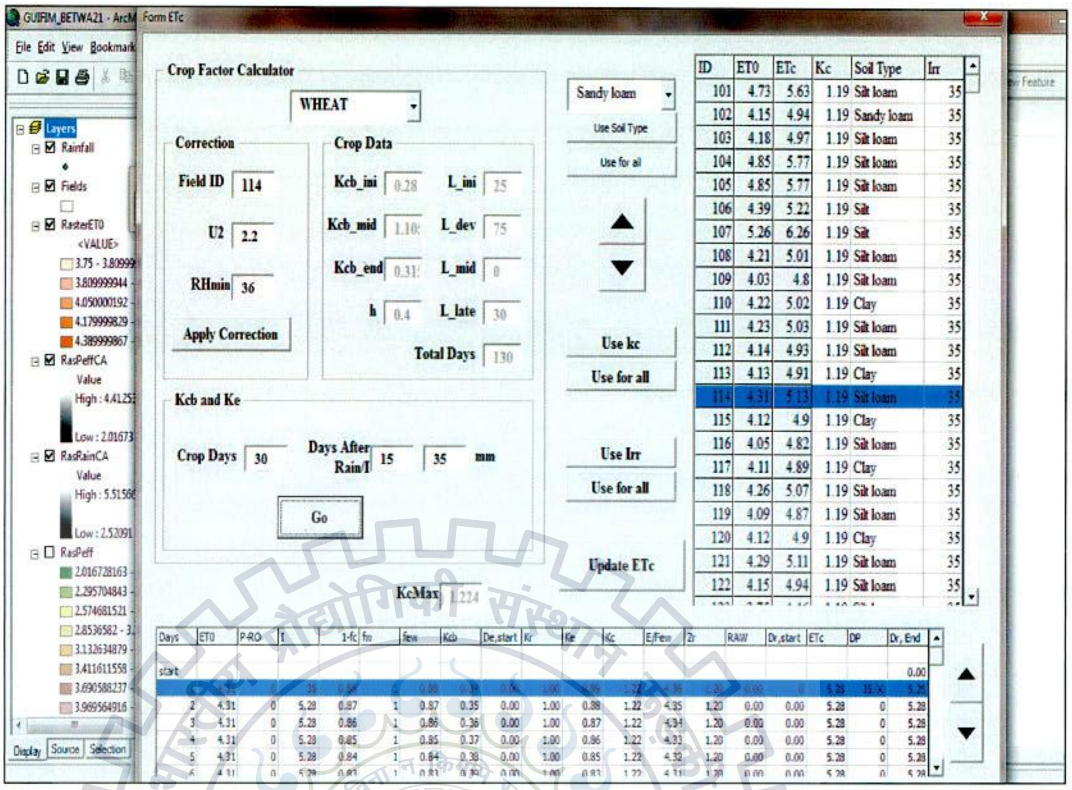

Figure 5.8: Form ETc to allow crop factor and water requirement

The ETc form allows in selecting the crop, applying the correction of crop coefficient for local climatic condition and performs soil water balance on the root zone of the crop to compute crop water requirements. After editing the input, selection and processing data update button allows updating and storing the new computed crop evapotranspiration and irrigation requirement values for each fields.

The crop data listed in form ETc is loaded from the database prepared inside the interface and can be selected for the use in soil water balance process. The computed data for crop evapotranspiration and irrigation requirement values can be used for either each field or for all fields at once and these data can be updated and stored in database using update button.

Raster ETc module allows access to the field layer and its ETc field data in the field's layer attribute table. This module uses field layer and its ETc field data as input and process to produce crop evapotranspiration (ETc) raster map of area included by the field polygon. The form ETc can perform following task regarding computation of crop evapotranspiration and irrigation requirement:

5.4.1 Selection and update soil type

Soil type can be selected for each field from the drop box and then allotted by command button. The soil type can be used for each field or used for all at once through the respective command buttons. Soil type is used for estimation of crop coefficient and water requirement in cropped fields.

5.4.2 Crop coefficient

Dual crop coefficient method is added for getting the specific wetting events of the value Kc. Kc is estimated into two parts i.e. ., the basal crop coefficient (Kcb), and one for soil evaporation (Ke).

$$ET_c = (K_{cb} + K_e)\ ET_0$$

User is allowed to select commonly used crops through the combo box. The crops data which are not in combo box can also be put by the user values defined by the user.

5.4.3 Correction Kc mid and Kc end for local climate

The Kc values (FAO paper-56, table 12) used in interface are typical expected values of average Kc under standard climatic condition. For the local climatic condition the Kc values needs correction. For adjustment of local climatic condition correction factor can be applied to each field through command button.

5.4.4 Soil water balance and allocating irrigation

For prediction of the future irrigation requirement, soil water balance of the root zone on a daily basis is carried out.

5.5 Module - Irrigation Management

The Irrigation management module generates the net irrigation requirement raster map. It uses the water requirement raster map and effective rainfall raster map generated by the interface to produce net irrigation thematic raster map.

5.6 Generation of thematic maps using Interface

The interface was used for the generation of thematic maps for the selected study area with the selected weather, soil and crop data. Daily weather data from Global Weather Data of SWAT (http://globalweather.tamu.edu/) was used for the rainfall and weather data to process interface and simulate irrigation requirement.

5.6.1 Generation of rainfall map and effective rainfall map

The rainfall layer was selected with twenty weather stations. The rainfall data for the period of 20 October 2013 was input and updated in Rainfall form. Selected study area has 20 rainfall stations all over the area and 35 sub watersheds. The interface is processed with these data to produce thematic rainfall and effective rainfall output maps by the interface. The runoff coefficient of 0.8 is applied through the area. The resultant maps are presented in figure 5. 9 a and b.

Generated thematic rainfall map (Fig.5. 9 a) shows the rainfall pattern of Betwa river basin for selected period which is easy to visualize and understand. Generated thematic rainfall map shows that rainfall received at Begumganj, Sironj, and Tikamgarh with its nearby places is higher one where as rainfall received at Khurai with its nearby places is lower one for selected period. Most of the places received moderate amount of rainfall. The rainfall received between 2.52 mm to 5.25 mm throughout the area for selected period.

Similarly generated thematic effective rainfall map (Fig.5. 9 b) shows the amount of effective rainfall received at betwa river basin for selected period is easy to visualize and understand. Generated thematic rainfall map shows that effective rainfall received at root zone of the crop by taking runoff and deep percolation losses as a constant factor (runoff coefficient - 0.8), at Begumganj, Sironj, and Tikamgarh with its nearby places is higher one where as effective rainfall received at Khurai with its nearby places is lower one for selected period. Most of the places received moderate amount of effective rainfall. The effective rainfall received between 2.02 mm to 4.41 mm throughout the area for selected period.

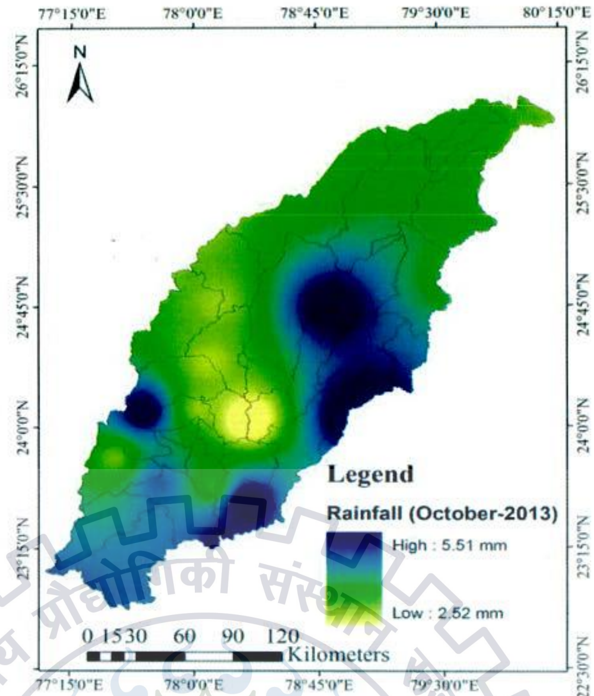

Figure 5.9: a) Rainfall raster map of selected area

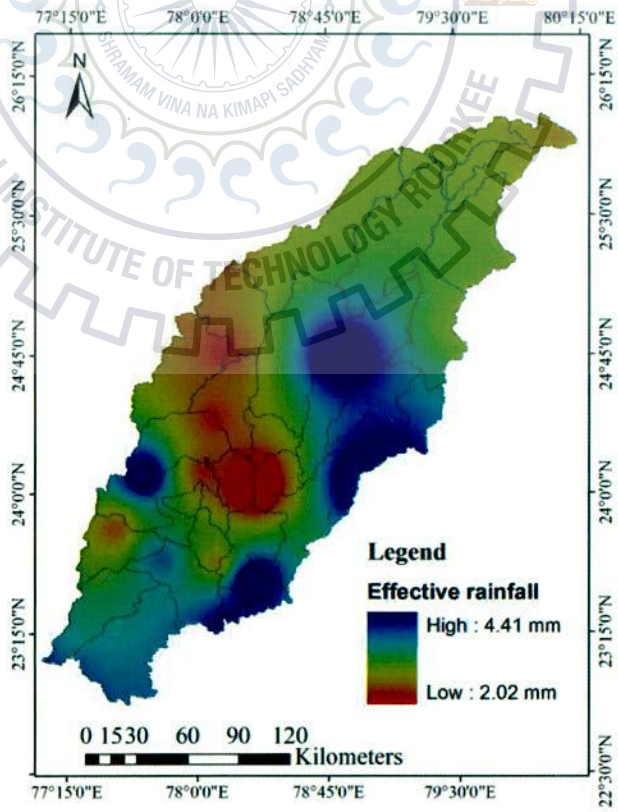

Figure 5.9: b) Effective rainfall raster map of selected area

5.6.2 Generation of reference evapotranspiration map

The interface was input with the weather data for the period of 20 October 2013. The layer of cropped field was used for the processing. Daily reference evapotranspiration map was generated using FAO 56 method one of the seven available methods in interface. FAO 56 method of computing evapotranspiration was used and generated thematic map for the study area. Thematic map is shown in figure 5.10.

Generated thematic daily reference evapotranspiration (Fig. 5.10) shows the reference evapotranspiration amount of Betwa river basin for selected period which can be easily visualize and understand. From the generated thematic reference evapotranspiration map one can easily understand that reference evapotranspiration is higher at some part of Bhopal and little part of Raisen which is attached with each other. Similarly some part of Tikamgarh and some part of Jhansi attached with it have also higher reference evapotranspiration. On the other hand very little portion of Raisen and Sagar attached with each other have lower reference evapotranspiration. Daily reference evapotranspiration lies between 3.75mm to 5.26 mm for selected period.

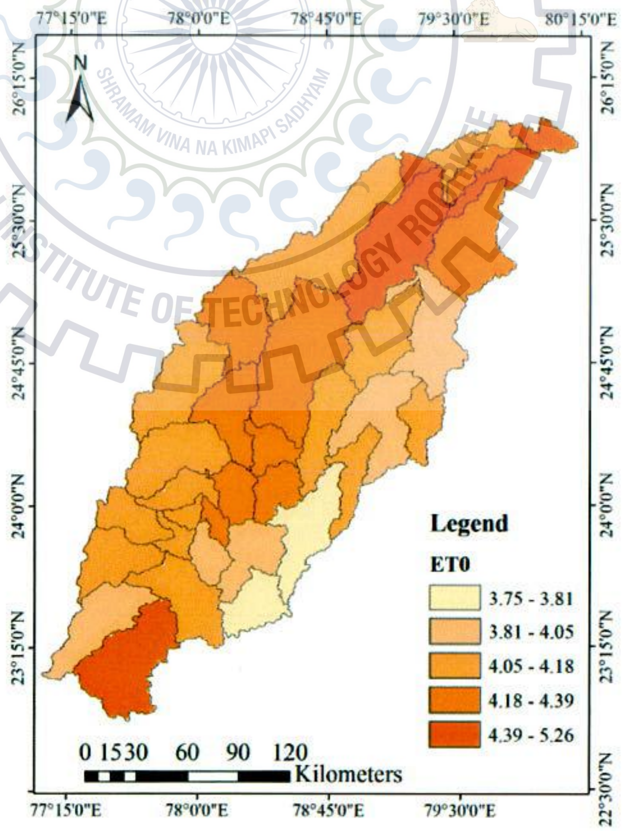

Figure 5.10: Daily Reference evapotranspiration (ET0) raster map of selected area

5.6.3 Generation of crop evapotranspiration map

The crop ET module in interface was implemented with the crop of spring wheat throughout the area. The crop coefficient and crop stage data for standard climate condition was automatically loaded from the interface data. Local climate correction was then applied with the weather data automatically loaded from interface for field layer. The layer of cropped field was used for the processing. Daily crop evapotranspiration map was generated .Figure 5.11 shows daily crop evapotranspiration (ETc) raster map for selected study area.

From the generated thematic reference evapotranspiration map one can easily understand that crop evapotranspiration is higher at some part of Bhopal and little part of Raisen which is attached with each other. Similarly some part of Tikamgarh and some part of Jhansi attached with each other also have higher crop evapotranspiration. On the other hand very little portion of Raisen and Sagar attached with each other have lower crop evapotranspiration. Daily crop evapotranspiration lies between 4.53 mm to 6.36 mm for selected period.

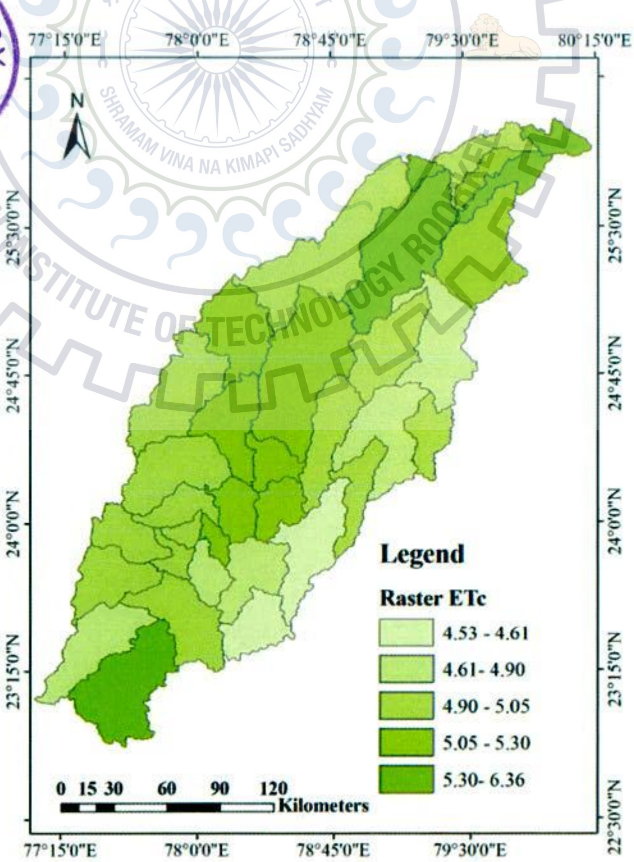

Figure 5.11: Daily crop evapotranspiration (ETc) raster map of selected area

5.6.4 Generation of Irrigation map

The crop data was selected and the correction for local climate was applied in the form crop ET. Initial irrigation depth of 35mm is input and interface was processed for soil water balance with 15 subsequent days. The layer of cropped field was used for the processing. Next water requirement raster map was generated for the study area. Irrigation requirement module was used to generate net irrigation requirement thematic map which uses effective rainfall raster map and water requirement map. Figure 5.12 shows Irrigation Requirement raster map for selected study area.

Generated NIR map shows that, area nearby Khurai rain gauge station has higher irrigation requirement. On the other hand area covered by Begumganj, Sironj, Tikamgarh and Banda attached with it have lower irrigation requirements. Net irrigation requirement lies between 30.59 mm to 32.98 mm for selected data and period.

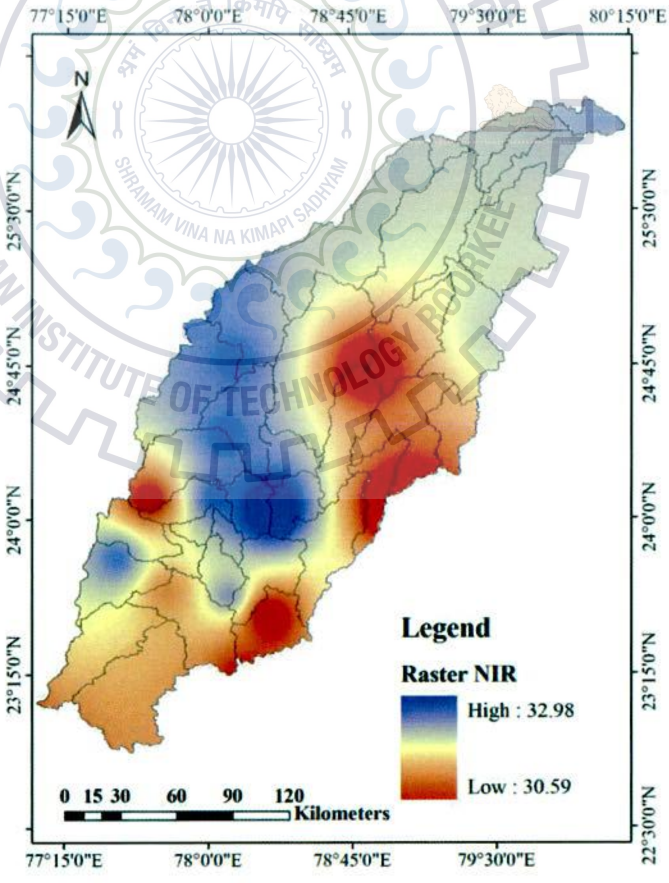

Figure 5.12: Net Irrigation Requirement(NIR) raster map of selected area

CHAPTER 6

SUMMARY AND CONCLUSION

Efficient management of irrigation system results in reducing water demand, water saving for other uses and also helps in improving agriculture productivity. Use of GIS for irrigation management provides solutions for handling spatial and temporal visibilities. Use of GIS also assists irrigation professionals and managers to overcome major challenges of multi tasking with varying goals in irrigation management. To carry out efficient irrigation management, irrigation professionals and managers have major problem due to traditional irrigation management support system and spatial and temporal variability in data hence it gives leads to use of GIS customization with VBA. The present study was carried out with the objective of development of a GIS based graphical user interface for irrigation management.

A GIS based graphical user interface for irrigation management tool has been effectively developed within GIS customizing using VBA which is included and fully supported in ArcGIS. The customization of GIS with the VBA is more powerful and effective when dealing with large area and complex temporal data. GIS based graphical user interface for irrigation management can be a good example of information sharing and visualization for the irrigation experts and farmers regarding irrigation management.

The developed interface tool was successfully applied for a Betwa basin to generate thematic raster map of irrigation requirement including reference evapotranspiration, crop evapotranspiration, rainfall and effective rainfall maps with input of weather, crop and soil data. The interface can be used by the irrigation experts for generating thematic maps and further effective irrigation management regarding irrigation allocating and scheduling.

Following conclusions are drawn from the present study:

1. The Developed interface tool with customization of ArcGIS using new UI controls and Menu will be versatile and user friendly application for irrigation management for the Betwa river basin, India.

2. The developed interface tool is successfully applied in Betwa river basin, India to generate thematic raster map of irrigation requirement including reference evapotranspiration, crop evapotranspiration, rainfall and effective rainfall maps with input of weather, crop and soil data.
3. The interface tool can be effectively employed for large and temporal data with interpolation techniques to generate thematic map of rainfall and effective rainfall, reference evapotranspiration, crop evapotranspiration using weather, crop and soil data.
4. The developed interface tool is capable of generating spatial irrigation requirement thematic maps based on effective rainfall and reference crop evapotranspiration maps to support an analysis and decision making process of irrigation management for Betwa river Basin, India.
5. The interface tool and the thematic maps generated by the interface can be a good example of information sharing and visualization for the irrigation experts and farmers regarding irrigation management.

Scope for future study

Due to the constraints on availability of data, time, several features are not added to the interface and the interface is not applied on full phase command area to test the performance of the interface. Future scopes of studies are listed as below,

1. The canal network can be added for irrigation schedule and water allotment inside the command area.
2. The interface could be fully tested for the performance with the field data with entire command area. The polygon GIS data shall be prepared/available to apply the interface to the whole command area.

REFERENCES

Acharya, S., Pandey, A. and Chaube, U.C. 2014, Use of Geographic Information Systems in Irrigation Management: a review. IWRS, Journal of Indian Water Resource Socity, 34 (2): 32-39.

Allen, R. G., Pereira, L.S., Raes, D. and Smith, M. 1998, Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56, FAO,1998.

Allen R.G and. Droogers P. 2002, Estimating reference evapotranspiration under inaccurate data condition, Irrigation and drainage systems 16: 33-45

ArcGIS Resource Center. http://resources.esri.com/help/9.3/arcgisdesktop/ com/vba_ start.htm

ArcGIS Help,9.3.http://webhelp.esri.com/ arcgisdesktop/9.3/ index.cfm? Topic Na me= welcome

Bastiaanssen, W. G. M. and Bos, M.G., 1999. Irrigation performance indicators based on remotely sensed data: a review of literature. Kuluwel Academic Publishers, Irrigation and Drainage System, 13:291-311.

Bastiaanssen, W. G. M. and Makin, W. I., 2000. Remote sensing for irrigated agriculture: examples from research and possible applications. ELSEVIER, Agricultural Water Management, 46: 137-155.

Bioggio, M. R. and Ding, Y., 2001. An integrated regional watershed model. In: Phelps, D., Sehlke, G. (Eds.), Bridging the Gap Meeting the World's Water and Environmental Resources Challenges, Proceedings of the World Water and Environmental Resources Congress, 20–24 May 2001, Orlando, FL. American Society of Civil Engineers, ASCE, EWRI, p. 10. CD ROM publication.

Choudhury, J., Ahmed, N. U., Idso, S. B., Reginato, R. J. and Daughtry, C.S., 1994. Relations between evaporation coefficients and vegetation indices studied by model simulations. ELSEVIER, Remote Sensing of Environment, 50:1-17.

Dhakal, A., 2010. Web GIS to support irrigation management a prototype for Sagra network, Alentejo Portugal. Master's thesis, Universidade Nova de Lisboa.

ESRI Developing Network, Customizing ArcGIS Desktop 9.3 using VBA. http://resources.esri.com/help/9.3/arcgisdesktop/com/vba_start.htm

Fortes, P. S., Platonov, A. E. and Pereira, L. S., 2005. GISAREG-A GIS based irrigation scheduling simulation model to support improved water use. ELSEVIER, Agricultural Water Management, 77: 159–179.

Goodchild, M. F., 1992. Geographical information science. International Journal of Geographical Information Systems 6(1): 31-45.

Gundogdu, K. S., Degirmenci, H. and Demirtas, C., 2002. Creation of gis supported database in irrigation project management. AGROENVIRON, 26-29 October 2002.

Hales, A.L. and Burton, M.A., 2000. Using the IRMOS model for diagnostic analysis and performance enhancement of the Rio Cobre Irrigation Scheme, Jamaica. Agric. Water Manage. 45, 185–202.

Hargreaves, G.H., Hargreaves, L.A. and Riley, J.P. 1985. Agriculture benefits for Senegal river basin. J. Irrig. Drain. Eng., ASCE, 111(2), 113-124.

Hargreaves, G.H., Samani, Z. A. 1985. Reference crop evavpotranspiration from temperature.Applied Eng. Agric. 1 (2), 96-99

Heinemann, A. B., Hoogenboom, G. and Faria, R. T., 2002. Determination of spatial water requirements at county and regional levels using crop models and GIS and example for the state of parana, Brazil. Agricultural Water Management, 52: 177-196

Irmak, S., Irmak, A., Allen, R., and Jones, J. 2003. Solar and Net Radiation-Based Equations to Estimate Reference Evapotranspiration in Humid Climates. J. Irrig. Drain Eng., 129(5), 336–347.

Kjelds, J. and Storm, B., 2001. Integrated water resources modeling water use and water quality simulation. In: Phelps, D., Sehlke, G. (Eds.),Bridging the Gap Meeting the World_s Water and Environmental Resources Challenges, Proceedings of the World Water and Environmental Resources Congress, 20–24 May 2001, Orlando,FL. American Society of Civil Engineers, ASCE, EWRI, p. 10. CDROM publication.

Liang, Xi. and Jianyu, Wu., 2012. The Application of GIS in the Field of Irrigation Water Management. IEEE, Computer Science and Network Technology, 2nd International conference, Changchun Dec 2012: 1497-1500.

Lin, W. T., Fong, H. Y. and Ming, S, 2004. GIS for Irrigation Management in Irrigation Associations. National Taiwan University, Taipei: D1-71.

Liu, Y., Teixeira, J. L., Zhang, H. J. and Pereira, L. S., 1998. Model validation and crop coefficients for irrigation scheduling in the North China Plain. Agricultural Water Management, 36: 233–246.

Lozano, D. and Mateos, L., 2008. Usefulness and limitations of decision support systems for improving irrigation scheme management. ELSEVIER, Agricultural Water Management, 95: 409–418.

Maguire, D. J., Goodchild, M. F. and Rhind, D. W., 1991. Geographical Information Systems: Principles and Applications, Vol. II. John Wiley & Sons, New Work.

Martin, M., 1996. Application of GIS in Irrigation Districts: How Much and When to Irrigate. Sixth International Conference on Computers in Agriculture, Maxico June 1996: 645-649.

Mateos, L., Lopez, I. and Sagardoy, J. A., 2002. SIMIS: the FAO decision support system for irrigation scheme management. ELSEVIER, Agricultural Water Management, 56(3): 193–206.

Muthanna, G. and Amin, M. S. M., 2003. Irrigation planning using geographic information system: A case study of Sana'a Basin. Yemen. Emerald, Management of Environmental Quality, 16(4): 347-361.

Naidu, C. R. and Giridhar, M. V., 2011. Geo-Spatial Database Creation for Wazirabad Canal Command Area. Geographical Information Systems, 3(4): 290-297.

Nixon, J.B., Dandy, G.C., Simpson, A.R., 2001. A genetic algorithm for optimizing off-farm irrigation scheduling. J. Hydroinformatics 3, 11–22.

Ortega, J. F., Juan, J. A. and Tarjuelo, J. M., 2005. Improving water management: the irrigation advisory service of Castilla la Mancha (Spain). Agricultural Water Management, 77: 37–58.

Pervez, M. S. and Hoque, M. A., 2008. Interactive Information System for Irrigation Management. Pergamon, Physics and Chemistry of the Earth, 28: 927-934.

Rao, N. H., Brownee, S. M. and Sarma, P. B., 2004. GIS-based decision support system for real time water demand estimation in canal irrigation systems. Current Science, 87(5): 628-636.

Raut, A. K., Bansal, A. K., Verma, V. K. and Marr, A. J., 2008. GIS-based decision support system for conjunctive irrigation management in India. 11 th International River Symposium, Brisbane, Australia, September 1-4, 2009.

Ray, S. S. and Dadhwal, V. K., 2001. Estimation of crop evapotranspiration of irrigation command area using remote sensing and GIS. Agricultural Water Management, 49:239-249.

Ray, S. S., Dadhwal, V. K., Chhabra, S. B., Brahmabhatt, V. S., 2000. Application of remote sensing and geographic information systems in irrigation management of Mahi command area in Gujarat, India. NNRMS Bulletin. DOS, Bangalore, 25: 17–21.

Ray, S.S, Dadhwal, V. K. and Navalgund, R. R., 2002. Performance evaluation of an irrigation command area using remote sensing: a case study of Mahi command, Gujarat, India. ELSEVIER, Agricultural Water Management 56: 81-91.

Su, M. D. and Wen, T. H., 2001. Spatial decision system for irrigation demand planning. In: Phelps, D., Sehlke, G. (Eds.), Bridging the GAP meeting the Worlds Water and Environmental Challenges, Proceedings of the World Water and Environmental Resources Congress, 20-24 May 2001, Orlando, FL. ASCE, EWRI, P.11.

Suresh Babu, A. V., Shanker, B. M. and Rao, V. V., 2012. Satellite Derived Geospatial Irrigation Performance Indicators for Benchmarking Studies of Irrigation Systems. Scientific Research, Advance in Remote Sensing, 1:1-13.

Suryavanshi, A.R., Reddy, J.M., 1986. Optimal operation schedule of irrigation distribution systems. Agric. Water Manage. 11, 23–30.

Tabari, H., Trajkovic, S. and Grismer M.E., 2011. Comparative analysis of 31 reference evapotranspiration methods under humid conditions, Irrig Sci (2013) 31:107–117.

Teixeira, J. L. and Pereira, L. S., 1992. ISAREG, an irrigation scheduling model. ICID Bulletin, 41(2): 29-48.

Todorovic, M. and Steduto, P., 2003. A GIS for irrigation management. Pergamon, Physics and Chemistry of the Earth, 28: 163–174.

Tsihrintzis, V. A., Hamid, R. and Hector, R. F. 1996. Use of Geographic information System (GIS) in water Resources: A Review. Kluwer Academic Publishers, Water Resources Management. 10: 251-277.

Yamashita, S., Walker, W.R., 1994. Command area water demands: I. Validation and calibration of UCA model. J. Irrig. Drain. Eng. 120, 1025–1042.

Yoo S.H., Choi J. Y and Jang M.W. 2008. Estimation of design water requirement using FAO Penman–Monteith and optimal probability distribution function in South Korea, Department of Rural Systems Engineering, Seoul National University, Seoul, Republic of Korea pp 1-3.

Appendix

Appenxix A: Codes

(i) Code for General Decleration

```
Public Getlayername As String
Public Rnflayername As String
Public Tinlayername As String
Public cmbupdateyes As Boolean
Public cmbETupdateyes As Boolean
Public cmbETcupdateyes As Boolean
Public ET0layername As String
Public ETclayername As String
```

(ii) Code for UIControls

```
Private Sub UIButtonControl1_Click()
'Give Access for Update Rainfall

If Rnflayername = "" Then
Getlayername = "Rainfall"
FrmLayerSelect.Label1.Caption = " Select Rainfall Layer"
FrmLayerSelect.Show
End If
FrmRain.Show
End Sub

Private Sub UIButtonControl10_Click()
'Updates Irrigation Raster
'Get the Field layer name if not
If ETclayername = "" Then
Getlayername = "ETc"
FrmLayerSelect.Label1.Caption = "Select Field Layer"
FrmLayerSelect.Show
End If

'First Remove existing layer if already exists
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap
Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterIrr" Then
ln = i
Exit For
End If
```

```
Next i

If ln <> -5 Then
Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing
Set pMap = Nothing

If pDS.CanDelete Then
pDS.Delete
End If
End If
' Make new Raster Layer and add to ToC

' Get the layer for Raster.
'Check if layer exists

Set pMap = pMxDoc.FocusMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = ETclayername Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Layer could not found")
Exit Sub
End If

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

Dim in_features As String
```

```
Dim Value_Field As String
Dim out_Raster As String
Dim cell_Assign As String
Dim Priority_Field As String
Dim cell_size As Integer

'//Define the input/output parameter variables
in_features = ETclayername
Value_Field = "Irr"
out_Raster = "C:\GUFIM_BETWA\Maps\RasterIrr"
cell_Assign = "MAXIMUM_AREA"
Priority_Field = "None"
cell_size = 1000

Dim parameters As IVariantArray
Set parameters = New VarArray
'//Populate the variant array with the parameters
parameters.Add (in_features)
parameters.Add (Value_Field)
parameters.Add (out_Raster)
parameters.Add (cell_Assign)
parameters.Add (Priority_Field)
parameters.Add (cell_size)

GP.Execute "PolygonToRaster", parameters, Nothing

'Update content, set Paramerters = Nothing
End Sub

Private Sub UIButtonControl11_Click()
'creates Irrigation Requriment Raster points
'On Error GoTo ErrorHandler
'Delete first if "NIR_Raster" already exisrs

 Dim pMxDoc As IMxDocument
  Set pMxDoc = ThisDocument
Dim pMap As IMap
  Set pMap = pMxDoc.FocusMap

Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterNIR" Then
ln = i
Exit For
End If
Next i

If ln <> -5 Then
```

```
Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing

pDS.Delete

End If

'Get the first raster from the RasterIrr layer in ArcMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterIrr" Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Raster Layer (RasterIrr) could not found")
Exit Sub
End If

Dim pLayer1 As ILayer
Set pLayer1 = pMxDoc.ActiveView.FocusMap.layer(ln)
Dim pRasLayer1 As IRasterLayer
Set pRasLayer1 = pLayer1
Dim pRas1 As IRaster
Set pRas1 = pRasLayer1.Raster

'Get the second raster from the RasterRain layer in ArcMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasPeffCA" Then
ln = i
```

```
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Raster Layer (RasterRain) could not found")
Exit Sub
End If

Dim pLayer2 As ILayer
Set pLayer2 = pMxDoc.ActiveView.FocusMap.layer(ln)
If Not TypeOf pLayer2 Is IRasterLayer Then
Exit Sub
End If
Dim pRasLayer2 As IRasterLayer
Set pRasLayer2 = pLayer2
Dim pRas2 As IRaster
Set pRas2 = pRasLayer2.Raster

 'Create a Spatial operator
Dim pMathOp As IMathOp
Set pMathOp = New RasterMathOps

 'Set output workspace
Dim pEnv As IRasterAnalysisEnvironment
Set pEnv = pMathOp
Dim pWs As IWorkspace
Dim pWSF As IWorkspaceFactory
Set pWSF = New RasterWorkspaceFactory
Set pWs = pWSF.OpenFromFile("C:\GUFIM_BETWA\Maps", 0)
Set pEnv.OutWorkspace = pWs

'Perform the operation
Dim pOutRaster As IRaster
Set pOutRaster = pMathOp.Minus(pRas1, pRas2)
'Create a raster layer and add it into ArcMap
Dim pOutRasLayer As IRasterLayer
Set pOutRasLayer = New RasterLayer
pOutRasLayer.Name = "RasterNIR"
pOutRasLayer.CreateFromRaster pOutRaster
pMap.AddLayer pOutRasLayer
'ErrorHandler:
'MsgBox Err.Description
End Sub

Private Sub UIButtonControl2_Click()
'Create Rainfall Raster from Rainfall points
'Get the rainfall layer name for rainfall raster

If Rnflayername = "" Then
```

```
Getlayername = "Rainfall"
 FrmLayerSelect.Label1.Caption = " Select Rainfall Layer"
FrmLayerSelect.Show
End If

'First Remove existing layer if already exits
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = 0 To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterRain" Then
ln = i
Exit For
End If

Next i

If ln <> -5 Then
Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing
Set pMap = Nothing

If pDS.CanDelete Then
pDS.Delete
End If
End If

'Make New Raster Layer And Add toC
'Get Point Layer For 3D points
```

```
Set pMap = pMxDoc.FocusMap
ln = -5

For i = 0 To pMap.LayerCount - 1
If pMap.layer(i).Name = Rnflayername Then
ln = i
Exit For
End If

Next i

If ln = -5 Then
MsgBox ("Layer Could not found")
Exit Sub
End If

Dim pflayer As IFeatureLayer
Set pflayer = pMxDoc.ActivatedView.FocusMap.layer(ln)

'Add references to esriSystem for licensing and IVariantArray.
'Imports ESRI.ArcGIS.esriSystem
' Add a reference to the geoprocessing namespace.
'Imports ESRI.ArcGIS.Geoprocessing


Dim GP As IGeoProcessor
Set GP = New GeoProcessor
'GP.AddToolbox ("C:\Program Files\ArcGIS\ArcToolBox\Toolboxes\Spatial Analyst
Tools.tbx")

Dim in_features As String
Dim Z_Field As String
Dim out_Raster As String
Dim cell_size, power As Integer
Dim Search_Redius As String
Dim in_barrier_polyline_features As String

' Define Input & output Parameter
in_features = "Rainfall"
Z_Field = "Rainfall"
out_Raster = "C:\GUFIM_BETWA\Maps\RasterRain"
power = 2
cell_size = 1000
Search_Redius = "VARIABLE 12"
in_barrier_polyline_features = ""

Dim parameters As IVariantArray
Set parameters = New VarArray
```

```
'//Populate the varient arrey with parameters
parameters.Add (in_features)
parameters.Add (Z_Field)
parameters.Add (out_Raster)
parameters.Add (cell_size)
parameters.Add (power)
parameters.Add (Search_Redius)
parameters.Add (in_barrier_polyline_features)

'Print parameters(in_features)
' Debug.Print parameters.Element(0)
 'Debug.Print parameters.Element(1)
 'Debug.Print parameters.Count

GP.Execute "IDW_sa", parameters, Nothing


cmbupdateyes = True
'Update content, set Paramentrs = nothing
End Sub

Private Sub UIButtonControl3_Click()
'Create Rainfall Raster from Rainfall points
'Get the rainfall layer name for rainfall raster

If Rnflayername = "" Then
Getlayername = "Rainfall"
FrmLayerSelect.Label1.Caption = "Select Rainfall Layer"
FrmLayerSelect.Show
End If

'First Remove existing layer if already exist
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasPeff" Then
ln = i
Exit For
End If
Next i

If ln <> -5 Then
```

```
Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing
Set pMap = Nothing

If pDS.CanDelete Then
pDS.Delete
End If
End If
' Make new Raster Layer and add to ToC

' Get the Pointlayer for 3D points.

Set pMap = pMxDoc.FocusMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = Rnflayername Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Layer could not found")
Exit Sub
End If

Dim pflayer As IFeatureLayer
Set pflayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

Dim in_features As String
Dim Z_Field As String
```

```
Dim out_Raster As String
Dim cell_size, power As Integer
Dim search_radius As String
Dim in_barrier_polyline_features As String


'//Define the input/output parameter variables
in_features = "Rainfall"
Z_Field = "Peff"
out_Raster = "C:\GUFIM_BETWA\Maps\RasPeff"
power = 2
cell_size = 1000
search_radius = "VARIABLE 12"
in_barrier_polyline_features = ""

Dim parameters As IVariantArray
Set parameters = New VarArray
'//Populate the variant array with the parameters
parameters.Add (in_features)
parameters.Add (Z_Field)
parameters.Add (out_Raster)
parameters.Add (cell_size)
parameters.Add (power)
parameters.Add (search_radius)
parameters.Add (in_barrier_polyline_features)

GP.Execute "IDW_sa", parameters, Nothing

cmbupdateyes = True
'Update content, set Paramerters = Nothing
End Sub
Private Sub UIButtonControl4_Click()
'Extracts the Rainfall DEM for Area of Feild Only
'Define msk layer

If ETclayername = "" Then
Getlayername = "ETc"
FrmLayerSelect.Label1.Caption = "Select Mask Layer"
FrmLayerSelect.Show
End If

'Delete first If "RasRainCA" already exits

Dim pMaxDoc As IMxDocument
Set pMaxDoc = ThisDocument

Dim pMap As IMap
Set pMap = pMaxDoc.FocusMap

Dim pRLayer As IRasterLayer
```

```
Dim i, ln As Integer
ln = -5
For i = 0 To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasRainCA" Then
ln = i
Exit For
End If

Next i

If ln <> -5 Then

Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing

pDS.Delete
End If

'Extracts new Raster layer by extracting the rainfall raster by mask of CA polygon
'Get the first raster from the RasterRain layer in ArcMap

ln = -5
For i = 0 To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterRain" Then
ln = i
Exit For
End If

Next i
If ln = -5 Then
MsgBox ("Raster Layer (RasterRain) could not found")
Exit Sub
End If
```

```
'Get the second raster from the RasterRain layer in ArcMap

ln = -5
 For i = 0 To pMap.LayerCount - 1

If pMap.layer(i).Name = ETclayername Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Layer For Feilds area could not found")
Exit Sub
End If

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

Dim in_Raster As String
Dim in_Mask As String
Dim out_Raster As String

'Check if layer exits

'//Define the input/output parameter variables
in_Raster = "RasterRain"
in_Mask = ETclayername
out_Raster = "C:\GUFIM_BETWA\Maps\RasRainCA"

Dim parameters As IVariantArray
Set parameters = New VarArray

'//Populate the varient arry with the parameters
parameters.Add (in_Raster)
parameters.Add (in_Mask)
parameters.Add (out_Raster)

GP.Execute "ExtractByMask_sa", parameters, Nothing


'Update content, set parmeters = Nothing
End Sub

Private Sub UIButtonControl5_Click()
'Creates peff rainfall raster for Fields only
'Define mask layer
If ETclayername = "" Then
Getlayername = "ETc"
FrmLayerSelect.Label1.Caption = "Select Mask Layer"
```

```
FrmLayerSelect.Show
End If

'Delete first if "RasRainCA" already exists
 Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasPeffCA" Then
ln = i
Exit For
End If
Next i

If ln <> -5 Then

Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing

pDS.Delete

End If

'Extract new Raster layer by extracting the Rainfall Raster by mask of CA polygon

'Get the first raster from the RasterRain layer in ArcMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterRain" Then
```

```
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Raster Layer (RasterRain) could not found")
Exit Sub
End If


'Get the second raster from the RasterRain layer in ArcMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = ETclayername Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Layer for Feilds area could not found")
Exit Sub
End If

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

Dim in_Raster As String
Dim in_Mask As String
Dim out_Raster As String

'Check if layer exits

'//Define the input/output parameter variables

in_Raster = "RasPeff"
in_Mask = ETclayername
out_Raster = "C:\GUFIM_BETWA\Maps\RasPeffCA"

Dim parameters As IVariantArray
Set parameters = New VarArray

'//Populate the varient arry with the parameters
parameters.Add (in_Raster)
parameters.Add (in_Mask)
parameters.Add (out_Raster)

GP.Execute "ExtractByMask_sa", parameters, Nothing
```

```
'Update content, set parameters = Nothing
End Sub

Private Sub UIButtonControl6_Click() 'Update ET0 Values (Weather)
If ET0layername = "" Then
Getlayername = "ET0"
FrmLayerSelect.Label1.Caption = "Select Field Layer"
FrmLayerSelect.Show
End If
FrmET0.Show
End Sub

Private Sub UIButtonControl7_Click()
'Updates ET0 Raster

'Get Fields layer name if not
If ET0layername = "" Then
Getlayername = "ET0"
FrmLayerSelect.Label1.Caption = "Select Field Layer"
FrmLayerSelect.Show
End If

'First Remove existing layer if already exists

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterET0" Then
ln = i
 Exit For
End If
Next i

If ln <> -5 Then

Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
```

```
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing
Set pMap = Nothing

If pDS.CanDelete Then
pDS.Delete
End If
End If
' Make new Raster Layer and add to ToC

' Get the layer for Raster.
'Check if layer exists

Set pMap = pMxDoc.FocusMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = ET0layername Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Layer could not found")
Exit Sub
End If

Dim GP As IGeoProcessor
Set GP = New GeoProcessor

Dim in_features As String
Dim Value_Field As String
Dim out_Raster As String
Dim cell_Assign As String
Dim Priority_Field As String
Dim cell_size As Integer

'//Define the input/output parameter variables
in_features = ET0layername
Value_Field = "ET0"
out_Raster = "C:\GUFIM_BETWA\Maps\RasterET0"
cell_Assign = "MAXIMUM_AREA"
Priority_Field = "None"
```

```
cell_size = 1400

Dim parameters As IVariantArray
Set parameters = New VarArray
'//Populate the variant array with the parameters
parameters.Add (in_features)
parameters.Add (Value_Field)
parameters.Add (out_Raster)
parameters.Add (cell_Assign)
parameters.Add (Priority_Field)
parameters.Add (cell_size)

GP.Execute "PolygonToRaster", parameters, Nothing

'Update content, set Paramerters = Nothing

End Sub


Private Sub UIButtonControl8_Click()
' Updates ETc values (Crop Factors)
If ET0ayername = "" Then
Getlayername = "ETc"
FrmLayerSelect.Label1.Caption = "Select Field Layer"
FrmLayerSelect.Show
End If
FrmETc.Show
End Sub

Private Sub UIButtonControl9_Click()
'Updates ETc Raster
'Get the Field layer name if not
If ETclayername = "" Then
Getlayername = "ETc"
FrmLayerSelect.Label1.Caption = "Select Field Layer"
FrmLayerSelect.Show
End If

'First Remove existing layer if already exists

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim pRLayer As IRasterLayer
Dim i, ln As Integer
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = "RasterETc" Then
```

```
ln = i
Exit For
End If
Next i

If ln <> -5 Then

Set pRLayer = pMxDoc.ActiveView.FocusMap.layer(ln)

Dim pRBCol As IRasterBandCollection
Set pRBCol = pRLayer.Raster

Dim pRB As IRasterBand
Set pRB = pRBCol.Item(0)

Dim pRDS As IRasterDataset
Set pRDS = pRB.RasterDataset

Dim pDS As IDataset
Set pDS = pRDS

pMxDoc.FocusMap.DeleteLayer pRLayer
Set pRLayer = Nothing
Set pRBCol = Nothing
Set pRB = Nothing
Set pMap = Nothing

If pDS.CanDelete Then
pDS.Delete
End If
End If
' Make new Raster Layer and add to ToC

' Get the layer for Raster.
'Check if layer exists

Set pMap = pMxDoc.FocusMap
ln = -5
For i = o To pMap.LayerCount - 1
If pMap.layer(i).Name = ETclayername Then
ln = i
Exit For
End If
Next i

If ln = -5 Then
MsgBox ("Layer could not found")
Exit Sub
End If
```

```
Dim GP As IGeoProcessor
Set GP = New GeoProcessor

Dim in_features As String
Dim Value_Field As String
Dim out_Raster As String
Dim cell_Assign As String
Dim Priority_Field As String
Dim cell_size As Integer

'//Define the input/output parameter variables
in_features = ETclayername
Value_Field = "ETc"
out_Raster = "C:\GUFIM_BETWA\Maps\RasterETc"
cell_Assign = "MAXIMUM_AREA"
Priority_Field = "None"
cell_size = 100

Dim parameters As IVariantArray
Set parameters = New VarArray
'//Populate the variant array with the parameters
parameters.Add (in_features)
parameters.Add (Value_Field)
parameters.Add (out_Raster)
parameters.Add (cell_Assign)
parameters.Add (Priority_Field)
parameters.Add (cell_size)

GP.Execute "PolygonToRaster", parameters, Nothing
'Update content, set Paramerters = Nothing
End Sub
```

(iii) Code for forms

Code for Select Layer forms

```
Private Sub CmdCancel_Click()
FrmLayerSelect.Hide
End Sub

Private Sub CmdSelect_Click()
If Getlayername = "Rainfall" Then
Rnflayername = cmbLayer.Value
ElseIf Getlayername = "ET0" Then
ET0layername = cmbLayer.Value
ElseIf Getlayername = "ETc" Then
ETclayername = cmbLayer.Value
End If
```

```
FrmLayerSelect.Hide
cmbLayer.Clear
End Sub

Private Sub UserForm_Activate()
cmbLayer.Value = "Select Layer"

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim i As Integer
i = pMap.LayerCount

For i = 0 To pMap.LayerCount - 1
cmbLayer.AddItem (pMap.layer(i).Name)
Next i
End Sub
```

Code for Update Rainfall forms

```
Private Sub Cmdupdate_Click()
Dim P, Peff, Pcoff As Single
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeature As IFeature
Dim pFeatureSelection As IFeatureSelection
Dim pSelectionSet As ISelectionSet
Dim pFeatureCursor As IFeatureCursor
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

Dim ln As Integer
For i = 0 To pMap.LayerCount - 1
If pMap.layer(i).Name = Rnflayername Then
ln = i
Exit For
End If

Next i
Set pFeatureLayer = pMap.layer(ln)
Set pFeatureSelection = pFeatureLayer
pFeatureSelection.SelectFeatures Nothing, esriSelectionResultNew, False 'Select All
Features
Set pSelectionSet = pFeatureSelection.SelectionSet
pSelectionSet.Search Nothing, False, pFeatureCursor  'Creates the Frature Cursor
```

```
Set pFeature = pFeatureCursor.NextFeature
i = 0

While Not pFeature Is Nothing
i = i + 1
P = FG2.TextMatrix(i, 3)
Pcoff = FG2.TextMatrix(i, 4)
Peff = Pcoff * P
FG2.TextMatrix(i, 1) = Format(Peff, "0.00")
pFeature.Value(pFeature.Fields.FindField("Rainfall")) = P
pFeature.Value(pFeature.Fields.FindField("Pcoff")) = Pcoff
pFeature.Value(pFeature.Fields.FindField("Peff")) = Peff
pFeature.Store
Set pFeature = pFeatureCursor.NextFeature
Wend
Cleanup:
Set pMxDoc = Nothing
Set pMap = Nothing
Set pFeatureLayer = Nothing
Set pFeatureSelection = Nothing
Set pSelectionSet = Nothing
Set pFeatureCursor = Nothing
   MsgBox ("Values Updated")
End Sub
Private Sub UserForm_Activate()
TxtEdit.Visible = False
TxtEdit = ""

'Put Headings
FG2.TextMatrix(0, 0) = "ID"
FG2.TextMatrix(0, 1) = "Peff"
FG2.TextMatrix(0, 2) = "Name"
FG2.TextMatrix(0, 3) = "Rainfall"
FG2.TextMatrix(0, 4) = "Pcoff"

Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pFeature As IFeature
Dim pFeatureSelection As IFeatureSelection
Dim pSelectionSet As ISelectionSet
Dim pFeatureCursor As IFeatureCursor

Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

Dim i, ln As Integer
ln = -5

For i = 0 To pMap.LayerCount - 1
```

```
If pMap.layer(i).Name = Rnflayername Then
ln = i
Exit For
End If

Next i

Set pFeatureLayer = pMap.layer(ln)
Set pFeatureSelection = pFeatureLayer
pFeatureSelection.SelectFeatures Nothing, esriSelectionResultNew, False 'Select All
Features
Set pSelectionSet = pFeatureSelection.SelectionSet
pSelectionSet.Search Nothing, False, pFeatureCursor 'Creates the feature cursor
Set pFeature = pFeatureCursor.NextFeature

i = 0
While Not pFeature Is Nothing
i = i + 1
FG2.TextMatrix(i, 0) = (pFeature.Value(pFeature.Fields.FindField("ID")))
FG2.TextMatrix(i, 1) = (pFeature.Value(pFeature.Fields.FindField("Peff")))
FG2.TextMatrix(i, 2) = (pFeature.Value(pFeature.Fields.FindField("Name")))
FG2.TextMatrix(i, 3) = (pFeature.Value(pFeature.Fields.FindField("Rainfall")))
 FG2.TextMatrix(i, 4) = (pFeature.Value(pFeature.Fields.FindField("Pcoff")))
Set pFeature = pFeatureCursor.NextFeature
Wend
Set pMxDoc = Nothing
Set pMap = Nothing
Set pFeatureLayer = Nothing
Set pFeature = Nothing
Set pFeatureSelection = Nothing
Set pSelectionSet = Nothing

End Sub

'Sub FG2_KeyPress(KeyAscii As Integer)
'MSHFlexGridEdit FG2, TxtEdit, KeyAscii
'End Sub

'Sub FG2_DoubleClick()
'MSHFlexGridEdit FG2, TxtEdit, 32 'Simulate Space
'End Sub
'Sub MSHFlexGridEdit(MSHFlexGrid As Control, Edt As Control, KeyAscii As
Integer)
'Use the Chareacter That was typed

'Select Case KeyAscii
'Use Space Means Edit Curret Text
'Case 0 To 32
'Edt = MSHFlexGrid
'Edt.SelStart = 1000
```

```
'Anything else means replace the current text
'Case Else
'Edt = Chr(KeyAscii)
Edt.SelStart = 1
'End Select

'Show Edt at the right place
'Edt.Move MSHFlexGrid.Left + MSHFlexGrid.CellLeft, MSHFlexGrid.Top +
MSHFlexGrid.CellTop, MSHFlexGrid.CellWidth - 8, MSHFlexGrid.CellHeight - 8
'Edt.Visible = True

'And Make it Work
'Edt.SetFocus
'End Sub


'Sub FG2_GotFocus()
'If TxtEdit.Visible = False Then
'Exit Sub
'FG2 = TxtEdit
'TxtEdit.Visible = False
'End Sub

'Sub FG2_LeaveCell()
'If TxtEdit.Visible = False Then
'Exit Sub
'FG2 = TxtEdit
'TxtEdit.Visible = False
End If
End Sub


'Sub TxtEdit_KeyDown(KeyCode As Integer, Shift As Integer)
'EditKeyCode FG2, TxtEdit, KeyCode, Shift
'End Sub

'Sub TxtEdit_KeyPress(KeyAscii As Integer)
'Delete return to grid rid of beep
'If KeyAscii = Asc(vbCr) Then KeyAscii = 0
'End Sub

'End Sub

'Private Sub TxtEdit_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal
Shift As Integer)
'Select Case KeyCode
'Case vbKeyEscape
'Leave the text unchanged
'TxtEdit.Visible = False
```

```
'FG2.SetFocus

'Case vbKeyReturn
'Finish Editing
'FG2.SetFocus
'DoEvents
'If FG2.Row < FG2.Rows - 1 Then
FG2.Row = FG2.Row + 1
'End If

'Case vbKeyDown
'Move Down 1 Row
'FG2.SetFocus
 'DoEvents
'If FG2.Row < FG2.Rows - 1 Then
'FG2.Row = FG2.Row + 1
 'End If

 'Case vbKeyUp
 'Move up 1 Row
'FG2.SetFocus
'DoEvents
If FG2.Row > FG2.FixedRows Then
 'FG2.Row = FG2.Row - 1
'End If
End Select
'End Sub
Private Sub GridEdit(KeyAscii As Integer)
 Position the TextBox over the cell.
TxtEdit.Left = FG2.CellLeft + FG2.Left
TxtEdit.Top = FG2.CellTop + FG2.Top
TxtEdit.Width = FG2.CellWidth
TxtEdit.Height = FG2.CellHeight
 TxtEdit.Visible = True
 TxtEdit.SetFocus

 Select Case KeyAscii
 Case 0 To Asc(" ")
TxtEdit.Text = FG2.Text
 TxtEdit.SelStart = Len(TxtEdit.Text)
Case Else
TxtEdit.Text = Chr$(KeyAscii)
 TxtEdit.SelStart = 1
End Select
End Sub

Private Sub Form_Resize()
FG2.Move 0, 0, ScaleWidth, ScaleHeight
End Sub
```

```
Private Sub TxtEdit_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal
Shift As Integer)

'(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyEscape
' Leave the text unchanged.
TxtEdit.Visible = False
FG2.SetFocus

Case vbKeyReturn
'Finish editing.
FG2.SetFocus

Case vbKeyDown
' Move down 1 row.
 FG2.SetFocus
DoEvents
If FG2.Row < FG2.Rows - 1 Then
 FG2.Row = FG2.Row + 1
End If

Case vbKeyUp
' Move up 1 row.
FG2.SetFocus
DoEvents
If FG2.Row > FG2.FixedRows Then
FG2.Row = FG2.Row - 1
End If

End Select
End Sub
' Do not beep on Return or Escape.

 Private Sub Fg2_DblClick()
GridEdit Asc(" ")
End Sub

Private Sub Fg2_KeyPress(KeyAscii As Integer)
GridEdit KeyAscii
End Sub

Private Sub Fg2_LeaveCell()
If TxtEdit.Visible Then
FG2.Text = TxtEdit.Text
TxtEdit.Visible = False
End If
End Sub
Private Sub Fg2_GotFocus()
```

```
If TxtEdit.Visible Then
FG2.Text = TxtEdit.Text
TxtEdit.Visible = False
End If
End Sub
```

Code for ET0 form

```
Dim prow1, prow2 As Integer

Private Sub cmdblaney_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
   i = 1
   While FG2.TextMatrix(i, 11) <> ""
   'pFeature.Value(pFeature.Fields.FindFiled("Rainfall")) = Fg2.TextMatrix(i,2)
   Tmax = FG2.TextMatrix(i, 2)
   Tmin = FG2.TextMatrix(i, 3)
   RHmax = FG2.TextMatrix(i, 4)
   RHmin = FG2.TextMatrix(i, 5)
   z = FG2.TextMatrix(i, 6)
   v = FG2.TextMatrix(i, 7)
   n = FG2.TextMatrix(i, 8)
   j = FG2.TextMatrix(i, 9)
   lat = FG2.TextMatrix(i, 10)
   h = FG2.TextMatrix(i, 11)
   FG2.TextMatrix(i, 1) = GetET0(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
   i = i + 1
   Wend
   End Sub

Private Sub CmdFAO56_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
   i = 1
   While FG2.TextMatrix(i, 11) <> ""
   'pFeature.Value(pFeature.Fields.FindFiled("Rainfall")) = Fg2.TextMatrix(i,2)
   Tmax = FG2.TextMatrix(i, 2)
   Tmin = FG2.TextMatrix(i, 3)
   RHmax = FG2.TextMatrix(i, 4)
   RHmin = FG2.TextMatrix(i, 5)
   z = FG2.TextMatrix(i, 6)
   v = FG2.TextMatrix(i, 7)
   n = FG2.TextMatrix(i, 8)
   j = FG2.TextMatrix(i, 9)
   lat = FG2.TextMatrix(i, 10)
   h = FG2.TextMatrix(i, 11)
   FG2.TextMatrix(i, 1) = GetET0(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
   i = i + 1
   Wend
```

```
    End Sub

Private Sub CmdTabari_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
i = 1
    While FG2.TextMatrix(i, 11) <> ""
    'pfeature.value(pfeature.fields.findFiled("Rainfall")) = Fg2.TextMatrix(i,2)
    Tmax = FG2.TextMatrix(i, 2)
    Tmin = FG2.TextMatrix(i, 3)
    RHmax = FG2.TextMatrix(i, 4)
    RHmin = FG2.TextMatrix(i, 5)
    z = FG2.TextMatrix(i, 6)
    v = FG2.TextMatrix(i, 7)
    n = FG2.TextMatrix(i, 8)
    j = FG2.TextMatrix(i, 9)
    lat = FG2.TextMatrix(i, 10)
    h = FG2.TextMatrix(i, 11)
    FG2.TextMatrix(i, 1) = GetTabari(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
    i = i + 1
    Wend
    End Sub

Private Sub CmdTurc_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
    i = 1
    While FG2.TextMatrix(i, 11) <> ""
    'pFeature.Value(pFeature.Fileds.FindField("Rainfall"))= FG2.TextMatrix (i,2)
    Tmax = FG2.TextMatrix(i, 2)
    Tmin = FG2.TextMatrix(i, 3)
    RHmax = FG2.TextMatrix(i, 4)
    RHmin = FG2.TextMatrix(i, 5)
    z = FG2.TextMatrix(i, 6)
    v = FG2.TextMatrix(i, 7)
    n = FG2.TextMatrix(i, 8)
    j = FG2.TextMatrix(i, 9)
    lat = FG2.TextMatrix(i, 10)
    h = FG2.TextMatrix(i, 11)
    FG2.TextMatrix(i, 1) = GetTurc(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
    i = i + 1
    Wend
End Sub

Private Sub CmdHarg_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
    i = 1
    While FG2.TextMatrix(i, 11) <> ""
    'pFeature.Value(pFeature.Fields.FindField("Rainfall"))= Fg2.TextMatrix (i,2)
```

```
   Tmax = FG2.TextMatrix(i, 2)
   Tmin = FG2.TextMatrix(i, 3)
   RHmax = FG2.TextMatrix(i, 4)
   RHmin = FG2.TextMatrix(i, 5)
   z = FG2.TextMatrix(i, 6)
   v = FG2.TextMatrix(i, 7)
   n = FG2.TextMatrix(i, 8)
   j = FG2.TextMatrix(i, 9)
   lat = FG2.TextMatrix(i, 10)
   h = FG2.TextMatrix(i, 11)
   FG2.TextMatrix(i, 1) = GetHarg(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
   i = i + 1
   Wend
End Sub

Private Sub CmdHargM1_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
   i = 1
   While FG2.TextMatrix(i, 11) <> ""
  'pfeature.value(pfeature.Fileds.FindField("Rainfall")) = Fg2.Textmatrix (i,2)
  Tmax = FG2.TextMatrix(i, 2)
  Tmin = FG2.TextMatrix(i, 3)
  RHmax = FG2.TextMatrix(i, 4)
  RHmin = FG2.TextMatrix(i, 5)
  z = FG2.TextMatrix(i, 6)
  v = FG2.TextMatrix(i, 7)
  n = FG2.TextMatrix(i, 8)
  j = FG2.TextMatrix(i, 9)
  lat = FG2.TextMatrix(i, 10)
  h = FG2.TextMatrix(i, 11)
  FG2.TextMatrix(i, 1) = GetHargM1(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
  i = i + 1
  Wend
End Sub
Private Sub CmdIrmak_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
    i = 1
    While FG2.TextMatrix(i, 11) <> ""
    'pFeature.Value(pFeature.Fields.FindField("Rainfall")) = FG2.TextMatrix(i, 2)
    Tmax = FG2.TextMatrix(i, 2)
    Tmin = FG2.TextMatrix(i, 3)
    RHmax = FG2.TextMatrix(i, 4)
    RHmin = FG2.TextMatrix(i, 5)
    z = FG2.TextMatrix(i, 6)
    v = FG2.TextMatrix(i, 7)
    n = FG2.TextMatrix(i, 8)
    j = FG2.TextMatrix(i, 9)
    lat = FG2.TextMatrix(i, 10)
```

```
      h = FG2.TextMatrix(i, 11)
      FG2.TextMatrix(i, 1) = GetIrmak(Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h)
       i = i + 1
     Wend
End Sub
Private Sub FAOPenman_Click()
Dim i As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
     i = 1
     While FG2.TextMatrix(i, 11) <> ""
      'pFeature.Value(pFeature.Fields.FindField("Rainfall")) = FG2.TextMatrix(i, 2)
      Tmax = FG2.TextMatrix(i, 2)
      Tmin = FG2.TextMatrix(i, 3)
      RHmax = FG2.TextMatrix(i, 4)
      RHmin = FG2.TextMatrix(i, 5)
      z = FG2.TextMatrix(i, 6)
      v = FG2.TextMatrix(i, 7)
      n = FG2.TextMatrix(i, 8)
      j = FG2.TextMatrix(i, 9)
      lat = FG2.TextMatrix(i, 10)
      h = FG2.TextMatrix(i, 11)
      FG2.TextMatrix(i, 1) = GetFAOPenman(Tmax, Tmin, RHmax, RHmin, z, v, n, j,
lat, h)
       i = i + 1
     Wend
End Sub

Private Sub Cmdall_Click()
Dim i, j As Integer
i = 1
j = 3
While FG2.TextMatrix(i, 1) <> ""
For j = 2 To 11
   FG2.TextMatrix(i, j) = FG3.TextMatrix(prow2, j)
Next j
i = i + 1
Wend
End Sub

Private Sub CmdSave_Click()
   Dim i As Integer
   Dim pMxDoc As IMxDocument
   Dim pMap As IMap
    Set pMxDoc = ThisDocument
    Set pMap = pMxDoc.FocusMap

Dim pTableCollection As ITableCollection
Set pTableCollection = pMap

Dim pTable As ITable
```

```
Set pTable = pTableCollection.Table(0)

i = 0
Dim pRow As IRow
While FG3.TextMatrix(i + 1, 11) <> ""
For j = 1 To 12

Set pRow = pTable.GetRow(i)
pRow.Value(j) = FG3.TextMatrix(i + 1, j - 1)
pRow.Store
Next j

i = i + 1
Wend
Cleanup:
   Set pMxDoc = Nothing
   Set pMap = Nothing
   Set pTableCollection = Nothing
   Set pTable = Nothing

End Sub

Private Sub Cmduse_Click()
Dim j As Integer
 For j = 2 To 11
  FG2.TextMatrix(prow1, j) = FG3.TextMatrix(prow2, j)
 Next j
End Sub


Private Sub CmdWupdate_Click()
Dim i, ln As Integer
Dim Tmax, Tmin, RHmax, RHmin, z, v, n, j, lat, h As Single
Dim pMxDoc As IMxDocument
   Dim pMap As IMap
   Dim pFeatureLayer As IFeatureLayer
   Dim pFeature As IFeature
   Dim pFeatureSelection As IFeatureSelection
   Dim pSelectionSet As ISelectionSet
   Dim pFeatureCursor As IFeatureCursor

   Set pMxDoc = ThisDocument
   Set pMap = pMxDoc.FocusMap

   For i = 0 To pMap.LayerCount - 1
   If pMap.layer(i).Name = ET0layername Then
   ln = i
   Exit For
   End If
Next i
```

```
Set pFeatureLayer = pMap.layer(ln)
Set pFeatureSelection = pFeatureLayer
   pFeatureSelection.SelectFeatures Nothing, esriSelectionResultNew, False 'Select All
Feature
Set pSelectionSet = pFeatureSelection.SelectionSet
   pSelectionSet.Search Nothing, False, pFeatureCursor 'Creates the featurcursor
Set pFeature = pFeatureCursor.NextFeature
   i = 0
   While Not pFeature Is Nothing
   i = i + 1
   'pFeature.Value(pFeature.Fields.FindField("Rainfall")) = FG2.TextMatrix(i, 2)
     Tmax = FG2.TextMatrix(i, 2)
     Tmin = FG2.TextMatrix(i, 3)
     RHmax = FG2.TextMatrix(i, 4)
     RHmin = FG2.TextMatrix(i, 5)
     z = FG2.TextMatrix(i, 6)
     v = FG2.TextMatrix(i, 7)
     n = FG2.TextMatrix(i, 8)
     j = FG2.TextMatrix(i, 9)
     lat = FG2.TextMatrix(i, 10)
     h = FG2.TextMatrix(i, 11)
     pFeature.Value(pFeature.Fields.FindField("Tmax")) = Tmax
     pFeature.Value(pFeature.Fields.FindField("Tmin")) = Tmin
     pFeature.Value(pFeature.Fields.FindField("RHmax")) = RHmax
     pFeature.Value(pFeature.Fields.FindField("RHmin")) = RHmin
     pFeature.Value(pFeature.Fields.FindField("z")) = z
     pFeature.Value(pFeature.Fields.FindField("v")) = v
     pFeature.Value(pFeature.Fields.FindField("n")) = n
     pFeature.Value(pFeature.Fields.FindField("j")) = j
     pFeature.Value(pFeature.Fields.FindField("lat")) = lat
     pFeature.Value(pFeature.Fields.FindField("h")) = h
     pFeature.Value(pFeature.Fields.FindField("ET0")) = FG2.TextMatrix(i, 1)
     pFeature.Store
      Set pFeature = pFeatureCursor.NextFeature
    Wend
Cleanup:
   Set pMxDoc = Nothing
   Set pMap = Nothing
   Set pFeatureLayer = Nothing
   Set pFeature = Nothing
   Set pFeatureSelection = Nothing
   Set pSelectionSet = Nothing
   Set pFeatureCursor = Nothing
End Sub


Private Sub Spin1_SpinDown()

If FG2.TextMatrix(prow1 + 1, 0) = "" Then Exit Sub
```

```
FG2.Row = prow1
For i = 0 To 11
FG2.col = i
FG2.CellBackColor = &H80000005
Next i
prow1 = prow1 + 1
FG2.Row = prow1

For i = 0 To 11
FG2.col = i
FG2.CellBackColor = &H8000000D
Next i

End Sub

Private Sub Spin1_SpinUp()
If prow1 < 2 Then Exit Sub
FG2.Row = prow1

For i = 0 To 11
FG2.col = i
FG2.CellBackColor = &H80000005
Next i
prow1 = prow1 - 1
FG2.Row = prow1
For i = 0 To 11
FG2.col = i
FG2.CellBackColor = &H8000000D
Next i

End Sub

Private Sub Spin2_SpinDown()
If FG3.TextMatrix(prow2 + 1, 0) = "" Then Exit Sub
FG3.Row = prow2
For i = 0 To 11
FG3.col = i
FG3.CellBackColor = &H80000005
Next i
prow2 = prow2 + 1
FG3.Row = prow2
For i = 0 To 11
FG3.col = i
FG3.CellBackColor = &H8000000D
Next i

End Sub
Private Sub Spin2_SpinUp()
If prow2 < 2 Then Exit Sub
FG3.Row = prow2
```

```
For i = 0 To 11
FG3.col = i
FG3.CellBackColor = &H80000005
Next i
prow2 = prow2 - 1
FG3.Row = prow2
For i = 0 To 11
FG3.col = i
FG3.CellBackColor = &H8000000D
Next i

End Sub


Private Sub UserForm_Activate()
Dim i, ln As Integer

For i = 0 To 11
FG2.ColWidth(i) = 800
FG3.ColWidth(i) = 800
Next i
FG3.ColWidth(1) = 1600
TxtEdit.Visible = False
TxtEdit = ""

FG2.TextMatrix(0, 0) = "ID"
FG2.TextMatrix(0, 1) = "ET0"
FG2.TextMatrix(0, 2) = "Tmax"
FG2.TextMatrix(0, 3) = "Tmin"
FG2.TextMatrix(0, 4) = "RHmax"
FG2.TextMatrix(0, 5) = "RHmin"
FG2.TextMatrix(0, 6) = "z"
FG2.TextMatrix(0, 7) = "v"
FG2.TextMatrix(0, 8) = "n"
FG2.TextMatrix(0, 9) = "j"
FG2.TextMatrix(0, 10) = "lat"
FG2.TextMatrix(0, 11) = "h"

'Get The name of layer


Dim pMxDoc As IMxDocument
 Dim pMap As IMap
 Dim pFeatureLayer As IFeatureLayer
 Dim pFeature As IFeature
 Dim pFeatureSelection As IFeatureSelection
 Dim pSelectionSet As ISelectionSet
 Dim pFeatureCursor As IFeatureCursor

 Set pMxDoc = ThisDocument
```

```
 Set pMap = pMxDoc.FocusMap

For i = o To pMap.LayerCount - 1
   If pMap.layer(i).Name = ET0layername Then
     ln = i
     Exit For
    End If
Next i

 Set pFeatureLayer = pMap.layer(ln)
 Set pFeatureSelection = pFeatureLayer
    pFeatureSelection.SelectFeatures Nothing, esriSelectionResultNew, False  ' select
all features
 Set pSelectionSet = pFeatureSelection.SelectionSet
    pSelectionSet.Search Nothing, False, pFeatureCursor ' creates the feature cursor
 Set pFeature = pFeatureCursor.NextFeature
    i = 0
    While Not pFeature Is Nothing
     i = i + 1
     FG2.TextMatrix(i, 0) = (pFeature.Value(pFeature.Fields.FindField("ID")))
     FG2.TextMatrix(i, 1) = (pFeature.Value(pFeature.Fields.FindField("ET0")))
     FG2.TextMatrix(i, 2) = (pFeature.Value(pFeature.Fields.FindField("Tmax")))
     FG2.TextMatrix(i, 3) = (pFeature.Value(pFeature.Fields.FindField("Tmin")))
     FG2.TextMatrix(i, 4) = (pFeature.Value(pFeature.Fields.FindField("RHmax")))
     FG2.TextMatrix(i, 5) = (pFeature.Value(pFeature.Fields.FindField("RHmin")))
      FG2.TextMatrix(i, 6) = (pFeature.Value(pFeature.Fields.FindField("z")))
      FG2.TextMatrix(i, 7) = (pFeature.Value(pFeature.Fields.FindField("v")))
      FG2.TextMatrix(i, 8) = (pFeature.Value(pFeature.Fields.FindField("n")))
      FG2.TextMatrix(i, 9) = (pFeature.Value(pFeature.Fields.FindField("j")))
      FG2.TextMatrix(i, 10) = (pFeature.Value(pFeature.Fields.FindField("lat")))
      FG2.TextMatrix(i, 11) = (pFeature.Value(pFeature.Fields.FindField("h")))
      Set pFeature = pFeatureCursor.NextFeature
    Wend
Cleanup:
 Set pMxDoc = Nothing
 Set pMap = Nothing
 Set pFeatureLayer = Nothing
 Set pFeature = Nothing
 Set pFeatureSelection = Nothing
 Set pSelectionSet = Nothing
 Set pFeatureCursor = Nothing


' Load the station data
If Rnflayername = "" Then
Getlayername = "Rainfall"
FrmLayerSelect.Label1.Caption = "Select Rainfall Layer"
FrmLayerSelect.Show
End If
FG3.TextMatrix(0, 0) = "ID"
```

```
FG3.TextMatrix(0, 1) = "Name"
FG3.TextMatrix(0, 2) = "Tmax"
FG3.TextMatrix(0, 3) = "Tmin"
FG3.TextMatrix(0, 4) = "RHmax"
FG3.TextMatrix(0, 5) = "RHmin"
FG3.TextMatrix(0, 6) = "Z"
FG3.TextMatrix(0, 7) = "v"
FG3.TextMatrix(0, 8) = "n"
FG3.TextMatrix(0, 9) = "j"
FG3.TextMatrix(0, 10) = "lat"
FG3.TextMatrix(0, 11) = "h"


prow1 = 1
prow2 = 1
FG2.Row = prow1
FG3.Row = prow2
For i = 0 To 11
FG2.col = i
FG3.col = i
FG2.CellBackColor = &H8000000D
FG3.CellBackColor = &H8000000D
Next i

'Fill Weather data on grid3
 Set pMxDoc = ThisDocument
 Set pMap = pMxDoc.FocusMap
 Dim pTableCollection As ITableCollection
 Set pTableCollection = pMap
 Dim pTable As ITable
  Set pTable = pTableCollection.Table(0)
For i = 0 To 9
Dim pRow As IRow
 Set pRow = pTable.GetRow(i)
 If pRow.Value(3) <> 0 Then
 For j = 1 To 12
 FG3.TextMatrix(i + 1, j - 1) = pRow.Value(j)
 Next j
 End If
 Next i
Cleanup2:
 Set pMxDoc = Nothing
 Set pMap = Nothing
 Set pTableCollection = Nothing
 Set pTable = Nothing
End Sub

Private Sub TxtEdit_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal
Shift As Integer)
Select Case KeyCode
```

```
        Case vbKeyEscape
            ' Leave the text unchanged.
            TxtEdit.Visible = False
            FG3.SetFocus

        Case vbKeyReturn
            ' Finish editing.
            FG3.SetFocus
            DoEvents
            If FG3.Row < FG3.Rows - 1 Then
                FG3.Row = FG3.Row + 1
            End If
        Case vbKeyDown
            ' Move down 1 row.
            FG3.SetFocus
            DoEvents
            If FG3.Row < FG3.Rows - 1 Then
                FG3.Row = FG3.Row + 1
            End If

        Case vbKeyUp
            ' Move up 1 row.
            FG3.SetFocus
            DoEvents
            If FG3.Row > FG3.FixedRows Then
                FG3.Row = FG3.Row - 1
            End If
        End Select
End Sub
Sub FG3_KeyPress(KeyAscii As Integer)
  MSHFlexGridEdit FG3, TxtEdit, KeyAscii
End Sub
Sub FG3_DblClick()
  MSHFlexGridEdit FG3, TxtEdit, 32 ' Simulate a space.
End Sub
Sub MSHFlexGridEdit(MSHFlexGrid As Control, Edt As Control, KeyAscii As
Integer)

  ' Use the character that was typed.
  Select Case KeyAscii

  ' A space means edit the current text.
  Case 0 To 32
    Edt = MSHFlexGrid
    Edt.SelStart = 1000

  ' Anything else means replace the current text.
  Case Else
    Edt = Chr(KeyAscii)
    Edt.SelStart = 1
```

```
  End Select
  ' Show Edt at the right place.
  Edt.Move MSHFlexGrid.Left + MSHFlexGrid.CellLeft, MSHFlexGrid.Top +
MSHFlexGrid.CellTop, MSHFlexGrid.CellWidth - 8, MSHFlexGrid.CellHeight - 8
  Edt.Visible = True
  'And make it work.
  Edt.SetFocus
End Sub
Sub FG3_GotFocus()
  If TxtEdit.Visible = False Then Exit Sub
  FG3 = TxtEdit
  TxtEdit.Visible = False
End Sub
Sub FG3_LeaveCell()
  If TxtEdit.Visible = False Then Exit Sub
  FG3 = TxtEdit
  TxtEdit.Visible = False
End Sub


Private Function GetET0(ByVal Tmax As Single, ByVal Tmin As Single, ByVal
RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single,
ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As
Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso,
Rnl, Rn, del, Et0, X As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) *
Sin(ws))
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
```

```
Rso = (0.75 + 2 * h / 100000) * Ra
Rnl = (sigma * (Tmax + 273.16) ^ 4 + sigma * (Tmin + 273.16) ^ 4) / 2 * (0.34 - 0.14 *
Sqr(Ea)) * (1.35 * Rs / Rso - 0.35)
Rn = 0.77 * Rs - Rnl
Et0 = 0.408 * (Rn - 0) * (delta / (delta + gamma * (1 + 0.34 * u2))) + gamma / (delta +
gamma * (1 + 0.34 * u2)) * 900 / (T + 273) * u2 * (Es - Ea)
GetET0 = Format(Et0, "0.00")
End Function
Private Function GetIrmak(ByVal Tmax As Single, ByVal Tmin As Single, ByVal
RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single,
ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As
Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso,
Rnl, Rn, del, Et0, X As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) *
Sin(ws))
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
Et0 = -0.16 + 0.149 * Rs + 0.079 * T
GetIrmak = Format(Et0, "0.00")
End Function
Private Function GetHargM1(ByVal Tmax As Single, ByVal Tmin As Single, ByVal
RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single,
ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As
Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso,
Rnl, Rn, del, Et0, X As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
```

```
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) *
Sin(ws))
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
Et0 = 0.408 * 0.003 * (T + 20) * (Tmax - Tmin) ^ 0.4 * Ra
GetHargM1 = Format(Et0, "0.00")
End Function
Private Function GetHarg(ByVal Tmax As Single, ByVal Tmin As Single, ByVal
RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single,
ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As
Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso,
Rnl, Rn, del, Et0, X, lamda As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) *
Sin(ws))
```

```
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
lamda = 2.501 - 0.002361 * T
Et0 = 1 / lamda * 0.0023 * (T + 17.8) * (Tmax - Tmin) ^ 0.5 * Ra
GetHarg = Format(Et0, "0.00")
End Function

Private Function GetFAOPenman(ByVal Tmax As Single, ByVal Tmin As Single,
ByVal RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As
Single, ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As
Single) As Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso,
Rnl, Rn, del, Et0, X, c, lamda As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) *
Sin(ws))
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
Rso = (0.75 + 2 * h / 100000) * Ra
Rnl = (sigma * (Tmax + 273.16) ^ 4 + sigma * (Tmin + 273.16) ^ 4) / 2 * (0.34 - 0.14 *
Sqr(Ea)) * (1.35 * Rs / Rso - 0.35)
Rn = 0.77 * Rs - Rnl
lamda = 2.501 - 0.002361 * T
Et0 = 1 / lamda * ((delta / (delta + gamma)) * Rn + (gamma / (gamma + delta)) * 6.43
* (1 + 0.864 * u2) * (Es - Ea))
GetFAOPenman = Format(Et0, "0.00")
End Function
Private Function GetTabari(ByVal Tmax As Single, ByVal Tmin As Single, ByVal
RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single,
ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As
Double
Dim T As Single
```

```
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso, Rnl, Rn, del, Et0, X As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) * Sin(ws))
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
Et0 = -0.478 + 0.156 * Rs + 0.0112 * Tmax + 0.0733 * Tmin
GetTabari = Format(Et0, "0.00")
End Function
Private Function GetTurc(ByVal Tmax As Single, ByVal Tmin As Single, ByVal RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single, ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso, Rnl, Rn, del, Et0, X, lamda, RsCal As Double
pi = 4 * Math.Atn(1)
phi = lat * pi / 180
sigma = 0.0000000049
u2 = v * 4.87 / Log(67.8 * z - 5.42)
T = (Tmax + Tmin) / 2
delta = 4098 * (0.6108 * Exp((17.27 * T) / (T + 237.3))) / (T + 237.3) ^ 2
P = 101.3 * ((293 - 0.0065 * h) / 293) ^ 5.26
gamma = 0.001013 * P / 0.622 / 2.45
E0Tmax = 0.6108 * Exp(17.27 * Tmax / (Tmax + 237.3))
E0Tmin = 0.6108 * Exp(17.27 * Tmin / (Tmin + 237.3))
Es = (E0Tmax + E0Tmin) / 2
Ea = (E0Tmin * RHmax / 100 + E0Tmax * RHmin / 100) / 2
dr = 1 + 0.033 * Cos(2 * pi * j / 365)
del = 0.409 * Sin(2 * pi * j / 365 - 1.39)
X = 1 - (Tan(phi)) ^ 2 * (Tan(del)) ^ 2
If X < 0 Then
```

```
X = 0.00001
End If
ws = pi / 2 - Atn(-Tan(phi) * Tan(del) / X ^ 0.5)
Ra = 24 * 60 / pi * 0.082 * dr * (ws * Sin(phi) * Sin(del) + Cos(phi) * Cos(del) *
Sin(ws))
Rs = (0.25 + 0.5 * n / (24 / pi * ws)) * Ra
RsCal = Rs * 0.23906 * 1000000 / 10000
lamda = 2.501 - 0.002361 * T
Et0 = 1 / lamda * 0.013 * T / (T + 15) * (RsCal + 50)
GetTurc = Format(Et0, "0.00")
End Function

Private Function Getblaney(ByVal Tmax As Single, ByVal Tmin As Single, ByVal
RHmax As Single, ByVal RHmin As Single, ByVal z As Single, ByVal v As Single,
ByVal n As Single, ByVal j As Single, ByVal lat As Single, ByVal h As Single) As
Double
Dim T As Single
Dim pi, phi, u2, delta, gamma, sigma, P, E0Tmax, E0Tmin, Es, Ea, dr, ws, Rs, Ra, Rso,
Rnl, Rn, del, Et0, X, y, lamda, RsCal As Double
T = (Tmax + Tmin) / 2
If lat = 23 Then
y = 0.2616
ElseIf lat = 24 Then
y = 0.2614
ElseIf lat = 25 Then
y = 0.261
ElseIf lat = 26 Then
y = 0.2604
ElseIf y = 27 Then
y = 0.2598
ElseIf lat = 28 Then
y = 0.2592
ElseIf lat = 29 Then
y = 0.2586
End If
Et0 = y(0.46 * T + 8)
Getblaney = Format(Et0, "0.00")
End Function
```

Code for ETc form

```
Dim cropname(100) As String
Dim pRow, trow, kcRow, cmbi As Integer
Dim Kcini(100), Kcmid(100), Kcend(100), lini(100), Ldev(100), Lmid(100),
Llate(100), h(100), Rz(100), Ro(100) As Single

Private Sub CmbCrop_Change()
cmbi = CmbCrop.ListIndex + 1
Txtkini.Text = Kcini(cmbi)
TxtKmid.Text = Kcmid(cmbi)
```

```
TxtKend.Text = Kcend(cmbi)
TxtLini.Text = lini(cmbi)
TxtLdev.Text = Ldev(cmbi)
TxtLmid.Text = Lmid(cmbi)
TxtLlate.Text = Llate(cmbi)
Txth.Text = h(cmbi)
TxtLtotal.Text = lini(cmbi) + Ldev(cmbi) + Lmid(cmbi) + Llate(cmbi)
End Sub


Private Sub CmdCorrection_Click()
Dim CKcbmid, CKcbend As Single
Dim u2, h, Kcmax, Kcmax1, Kcmax2, RHmin As Single
h = Val(Txth.Text)
u2 = Val(TxtU2.Text)
RHmin = Val(TxtRHmin.Text)
i = CmbCrop.ListIndex + 1
CKcbmid = Kcmid(cmbi) + (0.04 * (u2 - 2) - 0.004 * (RHmin - 45)) * (h / 3) ^ 0.3
CKcbend = Kcend(cmbi) + (0.04 * (u2 - 2) - 0.004 * (RHmin - 45)) * (h / 3) ^ 0.3
TxtKmid.Text = Format(CKcbmid, "0.00")
TxtKend.Text = Format(CKcbend, "0.00")
MsgBox ("Correction applied sucessfully")
End Sub

Private Sub Cmdirr_Click()
FG2.TextMatrix(pRow, 7) = FGkc.TextMatrix(kcRow, 3)
End Sub

Private Sub CmdIrrAll_Click()
Dim i As Integer
i = 1
While FG2.TextMatrix(i, 1) <> ""
 FG2.TextMatrix(i, 7) = FGkc.TextMatrix(kcRow, 3)
 i = i + 1
 Wend
End Sub


Private Sub CmdGo_Click()
'Compute KcMax
'compute Kcb first for the days
Dim i, d As Integer
Dim kc, Ld, Lm, Ltot As Single
Dim u2, h, Kcmax, Kcmax1, Kcmax2, RHmin, kcb As Single
i = CmbCrop.ListIndex + 1
d = Val(Txtdays.Text)
Ld = lini(i) + Ldev(i)
Lm = lini(i) + Ldev(i) + Lmid(i)
Ltot = lini(i) + Ldev(i) + Lmid(i) + Llate(i)
If d > Ltot Then
```

```
MsgBox ("The days you typed more then total time of crop season")
Exit Sub
End If
If d <= lini(i) Then
kc = Kcini(i)
ElseIf d > lini(i) And d <= Ld Then
kc = Kcini(i) + (Kcmid(i) - Kcini(i)) * (d - lini(i)) / Ldev(i)
ElseIf d > Ld And d <= Lm Then
kc = Kcmid(i)
Else
kc = Kcmid(i) - (Kcmid(i) - Kcend(i)) * (d - Lm) / Llate(i)
End If

h = Val(Txth.Text)
u2 = Val(TxtU2.Text)
RHmin = Val(TxtRHmin.Text)
kcb = kc
Kcmax1 = 1.2 + (0.04 * (u2 - 2) - 0.004 * (RHmin - 45)) * (h / 3) ^ 0.3
Kcmax2 = kcb + 0.05
If Kcmax1 > Kcmax2 Then
Kcmax = Kcmax1
Else
Kcmax = Kcmax2
End If
TxtKcmax.Text = Format(Kcmax, "0.000")

'Compute the table and fill it
Dim Et0, Ke, ke1, ke2, Kr, few, TEW, REW, RAW, TAW, Irr, Kcmin As Single
Dim fc, fw, DeStart, Efew, Prz, Pro, ETc, DP, DrStart, DrEnd As Single
Dim Kedays, days As Integer
Kedays = Val(TxtKedays.Text)
days = Val(Txtdays.Value)
Irr = Val(TxtIrr.Text)
Kcmin = 0.15
fw = 1

'Fill the headings of table
FGkc.TextMatrix(0, 0) = "Days"
FGkc.TextMatrix(0, 1) = "ET0"
FGkc.TextMatrix(0, 2) = "P-RO"
FGkc.TextMatrix(0, 3) = "I"
FGkc.TextMatrix(0, 4) = "1-fc"
FGkc.TextMatrix(0, 5) = "fw"
FGkc.TextMatrix(0, 6) = "few"
FGkc.TextMatrix(0, 7) = "Kcb"
FGkc.TextMatrix(0, 8) = "De,start"
FGkc.TextMatrix(0, 9) = "Kr"
FGkc.TextMatrix(0, 10) = "Ke"
FGkc.TextMatrix(0, 11) = "Kc"
FGkc.TextMatrix(0, 12) = "E/Few"
```

```
DeStart = 0
End If
FGkc.TextMatrix(i, 8) = Format(DeStart, "0.00")
If DeStart <= REW Then
  Kr = 1
Else
  Kr = (TEW - DeStart) / (TEW - REW)
End If
FGkc.TextMatrix(i, 9) = Format(Kr, "0.00")
ke1 = Kr * (Kcmax - kcb)
ke2 = few * Kcmax
If ke1 < ke2 Then
  Ke = ke1
Else
  Ke = ke2
End If
FGkc.TextMatrix(i, 10) = Format(Ke, "0.00")
kc = Ke + kcb
FGkc.TextMatrix(i, 11) = Format(kc, "0.00")
Efew = Ke * Et0 / few
FGkc.TextMatrix(i, 12) = Format(Efew, "0.00")
FGkc.TextMatrix(i, 13) = Format(pzr, "0.00")
FGkc.TextMatrix(i, 13) = Format(Prz, "0.00")
FGkc.TextMatrix(i, 14) = Format(RAW, "0.00")
ETc = Et0 * kc
FGkc.TextMatrix(i, 16) = Format(ETc, "0.00")
DP = FGkc.TextMatrix(i, 3) - FGkc.TextMatrix(i - 1, 18)
If DP <= 0 Then
  FGkc.TextMatrix(i, 17) = 0
Else
  FGkc.TextMatrix(i, 17) = Format(DP, "0.00")
End If
DrEnd = FGkc.TextMatrix(i - 1, 18) - FGkc.TextMatrix(i, 2) - FGkc.TextMatrix(i, 3) +
ETc + FGkc.TextMatrix(i, 17)
DrStart = FGkc.TextMatrix(i - 1, 18) - FGkc.TextMatrix(i, 2) - FGkc.TextMatrix(i, 3)
If DrStart < 0 Then
 FGkc.TextMatrix(i, 15) = 0
Else
 FGkc.TextMatrix(i, 15) = Format(DrStart, "0.00")
End If
FGkc.TextMatrix(i, 18) = Format(DrEnd, "0.00")
End If
Next i

kcRow = 3
FGkc.Row = kcRow
For i = 0 To 18
FGkc.col = i
FGkc.CellBackColor = &H8000000D
Next i
```

```
End Sub

Private Sub CmdsoilAll_Click()
Dim i As Integer
i = 1
While FG2.TextMatrix(i, 1) <> ""
 FG2.TextMatrix(i, 4) = CmbSoil.Value
 i = i + 1
 Wend
End Sub

Private Sub CmdSoli_Click()
 FG2.TextMatrix(pRow, 4) = CmbSoil.Value
End Sub

Private Sub Cmdupdate_Click()
Dim i, ln As Integer
Dim kc, ETc, Et0 As Single

Dim pMxDoc As IMxDocument
  Dim pMap As IMap
  Dim pFeatureLayer As IFeatureLayer
  Dim pFeature As IFeature
  Dim pFeatureSelection As IFeatureSelection
  Dim pSelectionSet As ISelectionSet
  Dim pFeatureCursor As IFeatureCursor
  Set pMxDoc = ThisDocument
  Set pMap = pMxDoc.FocusMap

   For i = 0 To pMap.LayerCount - 1
     If pMap.layer(i).Name = ETclayername Then
       ln = i
     Exit For
     End If
   Next i

  Set pFeatureLayer = pMap.layer(ln)
  Set pFeatureSelection = pFeatureLayer
     pFeatureSelection.SelectFeatures Nothing, esriSelectionResultNew, False  ' select
all features
  Set pSelectionSet = pFeatureSelection.SelectionSet
     pSelectionSet.Search Nothing, False, pFeatureCursor ' creates the feature cursor
  Set pFeature = pFeatureCursor.NextFeature
     i = 0
     While Not pFeature Is Nothing
      i = i + 1
      kc = FG2.TextMatrix(i, 3)
      Et0 = FG2.TextMatrix(i, 1)
      ETc = kc * Et0
```

```
        pFeature.Value(pFeature.Fields.FindField("Kc")) = kc
        pFeature.Value(pFeature.Fields.FindField("ETc")) = Format(ETc, "0.00")
        pFeature.Value(pFeature.Fields.FindField("Soil_Type")) = FG2.TextMatrix(i, 4)
        pFeature.Value(pFeature.Fields.FindField("Irr")) = FG2.TextMatrix(i, 7)
        pFeature.Store
         Set pFeature = pFeatureCursor.NextFeature
       Wend
Cleanup:
  Set pMxDoc = Nothing
  Set pMap = Nothing
  Set pFeatureLayer = Nothing
  Set pFeature = Nothing
  Set pFeatureSelection = Nothing
  Set pSelectionSet = Nothing
  Set pFeatureCursor = Nothing
End Sub

Private Sub CmdUseKc_Click()
 FG2.TextMatrix(pRow, 3) = FGkc.TextMatrix(kcRow, 11)
End Sub

Private Sub CmdUsekcAll_Click()
Dim i As Integer
i = 1
While FG2.TextMatrix(i, 1) <> ""
 FG2.TextMatrix(i, 3) = FGkc.TextMatrix(kcRow, 11)
 i = i + 1
 Wend
End Sub

Private Sub Spin2_SpinDown()
If pRow = trow Then Exit Sub
FG2.Row = pRow
For i = 0 To 2
FG2.col = i
FG2.CellBackColor = &H8000000F
Next i
For i = 3 To 7
FG2.col = i
FG2.CellBackColor = &H80000005
Next i
pRow = pRow + 1
FG2.Row = pRow
For i = 0 To 7
FG2.col = i
FG2.CellBackColor = &H8000000D
TxtU2.Text = FG2.TextMatrix(pRow, 5)
TxtRHmin.Text = FG2.TextMatrix(pRow, 6)
TxtID.Text = FG2.TextMatrix(pRow, 0)
Next i
```

```
End Sub
Private Sub Spin2_SpinUp()
If pRow < 2 Then Exit Sub
FG2.Row = pRow
For i = 0 To 2
FG2.col = i
FG2.CellBackColor = &H8000000F
Next i
For i = 3 To 7
FG2.col = i
FG2.CellBackColor = &H80000005
Next i
pRow = pRow - 1
FG2.Row = pRow
For i = 0 To 7
FG2.col = i
FG2.CellBackColor = &H8000000D
TxtU2.Text = FG2.TextMatrix(pRow, 5)
TxtRHmin.Text = FG2.TextMatrix(pRow, 6)
TxtID.Text = FG2.TextMatrix(pRow, 0)
Next i
End Sub

Private Sub SpinKc_SpinDown()
If FGkc.TextMatrix(kcRow + 1, 0) = "" Then Exit Sub
FGkc.Row = kcRow
For i = 0 To 18
FGkc.col = i
FGkc.CellBackColor = &H80000005
Next i
kcRow = kcRow + 1
FGkc.Row = kcRow
For i = 0 To 18
FGkc.col = i
FGkc.CellBackColor = &H8000000D
Next i
End Sub

Private Sub SpinKc_SpinUp()
If kcRow < 4 Then Exit Sub
FGkc.Row = kcRow
For i = 0 To 18
FGkc.col = i
FGkc.CellBackColor = &H80000005
Next i
kcRow = kcRow - 1
FGkc.Row = kcRow
For i = 0 To 18
FGkc.col = i
```

```
FGkc.CellBackColor = &H8000000D
Next i
End Sub

Private Sub Txtdays_Change()

End Sub

Private Sub UserForm_Activate()
Dim i, ln As Integer

For i = 0 To 3
FG2.ColWidth(i) = 700
Next i
FG2.ColWidth(4) = 1340
FG2.ColWidth(7) = 700
FG2.ColWidth(5) = 1
FG2.ColWidth(6) = 1
For i = 0 To 18
FGkc.ColWidth(i) = 730
Next i

TxtEdit.Visible = False
TxtEdit = ""

' Put headings
FG2.TextMatrix(0, 0) = "ID"
FG2.TextMatrix(0, 1) = "ET0"
FG2.TextMatrix(0, 2) = "ETc"
FG2.TextMatrix(0, 3) = "Kc"
FG2.TextMatrix(0, 4) = "Soil Type"
FG2.TextMatrix(0, 7) = "Irr"

'Get the name of the layer
Dim pMxDoc As IMxDocument
  Dim pMap As IMap
  Dim pFeatureLayer As IFeatureLayer
  Dim pFeature As IFeature
  Dim pFeatureSelection As IFeatureSelection
  Dim pSelectionSet As ISelectionSet
  Dim pFeatureCursor As IFeatureCursor

  Set pMxDoc = ThisDocument
  Set pMap = pMxDoc.FocusMap

For i = o To pMap.LayerCount - 1
    If pMap.layer(i).Name = ETclayername Then
     ln = i
     Exit For
    End If
```

```
Next i

 Set pFeatureLayer = pMap.layer(ln)
 Set pFeatureSelection = pFeatureLayer
   pFeatureSelection.SelectFeatures Nothing, esriSelectionResultNew, False  ' select
all features
 Set pSelectionSet = pFeatureSelection.SelectionSet
   pSelectionSet.Search Nothing, False, pFeatureCursor ' creates the feature cursor
 Set pFeature = pFeatureCursor.NextFeature
 i = 0
   While Not pFeature Is Nothing
    i = i + 1
    FG2.TextMatrix(i, 0) = (pFeature.Value(pFeature.Fields.FindField("ID")))
    FG2.TextMatrix(i, 1) = (pFeature.Value(pFeature.Fields.FindField("ET0")))
    FG2.TextMatrix(i, 2) = (pFeature.Value(pFeature.Fields.FindField("ETc")))
    FG2.TextMatrix(i, 3) = (pFeature.Value(pFeature.Fields.FindField("Kc")))
    FG2.TextMatrix(i, 4) = (pFeature.Value(pFeature.Fields.FindField("Soil_Type")))
    FG2.TextMatrix(i, 5) = (pFeature.Value(pFeature.Fields.FindField("V")))
    FG2.TextMatrix(i, 6) = (pFeature.Value(pFeature.Fields.FindField("RHmin")))
     FG2.TextMatrix(i, 7) = (pFeature.Value(pFeature.Fields.FindField("Irr")))
     Set pFeature = pFeatureCursor.NextFeature
   Wend
   trow = i
Cleanup:
 Set pMxDoc = Nothing
 Set pMap = Nothing
 Set pFeatureLayer = Nothing
 Set pFeature = Nothing
 Set pFeatureSelection = Nothing
 Set pSelectionSet = Nothing
 Set pFeatureCursor = Nothing
 cmbETcupdateyes = False

' for the use of crop coefficient
Dim sline, getline, sFileName As String
  Dim iFileNum, j As Integer
  ' get the file
  sFileName = "C:\GUFIM_BETWA\Data\Crop Data.txt"

  ' does the file exist?  simpleminded test:
  If Len(Dir$(sFileName)) = 0 Then
    Exit Sub
  End If
j = 0
  iFileNum = FreeFile()
  Open sFileName For Input As iFileNum

  Do While Not EOF(iFileNum)
    j = j + 1
    Line Input #iFileNum, getline
```

```
        sline = Split(getline, vbTab)
        cropname(j) = sline(0)
        If sline(1) = "" Then
          Kcini(j) = Val(sline(2))
          Kcmid(j) = Val(sline(3))
          Kcend(j) = Val(sline(4))
          h(j) = Val(sline(5))
          lini(j) = Val(sline(6))
          Ldev(j) = Val(sline(7))
          Lmid(j) = Val(sline(8))
          Llate(j) = Val(sline(9))
          Rz(j) = Val(sline(10))
          Ro(j) = Val(sline(11))
        Else
          Kcini(j) = Val(sline(1))
          Kcmid(j) = Val(sline(2))
          Kcend(j) = Val(sline(3))
          h(j) = Val(sline(4))
          lini(j) = Val(sline(5))
          Ldev(j) = Val(sline(6))
          Lmid(j) = Val(sline(7))
          Llate(j) = Val(sline(8))
          Rz(j) = Val(sline(9))
          Ro(j) = Val(sline(10))
        End If
    Loop
    ' close the file
    Close iFileNum
    For i = 1 To j
      CmbCrop.AddItem (cropname(i))
    Next i

'Add soil type to the combo box
CmbSoil.AddItem ("Sand")
CmbSoil.AddItem ("Loamy sand")
CmbSoil.AddItem ("Sandy loam")
CmbSoil.AddItem ("Silt loam")
CmbSoil.AddItem ("Silt")
CmbSoil.AddItem ("Silt clay loam")
CmbSoil.AddItem ("Silty clay")
CmbSoil.AddItem ("Clay")

pRow = 1
FG2.Row = pRow
For i = 0 To 7
FG2.col = i
FG2.CellBackColor = &H8000000D
Next i
TxtU2.Text = FG2.TextMatrix(pRow, 5)
TxtRHmin.Text = FG2.TextMatrix(pRow, 6)
```

```
TxtID.Text = FG2.TextMatrix(pRow, 0)
End Sub

Private Sub TxtEdit_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal
Shift As Integer)
Select Case KeyCode
      Case vbKeyEscape
         ' Leave the text unchanged.
         TxtEdit.Visible = False
         FG2.SetFocus

      Case vbKeyReturn
         ' Finish editing.
         FG2.SetFocus
         DoEvents
         If FG2.Row < FG2.Rows - 1 Then
            FG2.Row = FG2.Row + 1
         End If
      Case vbKeyDown
         ' Move down 1 row.
         FG2.SetFocus
         DoEvents
         If FG2.Row < FG2.Rows - 1 Then
            FG2.Row = FG2.Row + 1
         End If

      Case vbKeyUp
         ' Move up 1 row.
         FG2.SetFocus
         DoEvents
         If FG2.Row > FG2.FixedRows Then
            FG2.Row = FG2.Row - 1
         End If
      End Select
End Sub
Sub Fg2_KeyPress(KeyAscii As Integer)
  MSHFlexGridEdit FG2, TxtEdit, KeyAscii
End Sub
Sub Fg2_DblClick()
  MSHFlexGridEdit FG2, TxtEdit, 32 ' Simulate a space.
End Sub
Sub MSHFlexGridEdit(MSHFlexGrid As Control, Edt As Control, KeyAscii As
Integer)

  ' Use the character that was typed.
  Select Case KeyAscii

  ' A space means edit the current text.
  Case 0 To 32
    Edt = MSHFlexGrid
```

```
    Edt.SelStart = 1000

  ' Anything else means replace the current text.
  Case Else
    Edt = Chr(KeyAscii)
    Edt.SelStart = 1
  End Select
  ' Show Edt at the right place.
  Edt.Move MSHFlexGrid.Left + MSHFlexGrid.CellLeft, MSHFlexGrid.Top +
MSHFlexGrid.CellTop, MSHFlexGrid.CellWidth - 8, MSHFlexGrid.CellHeight - 8
  Edt.Visible = True
  'And make it work.
  Edt.SetFocus
End Sub
Sub Fg2_GotFocus()
  If TxtEdit.Visible = False Then Exit Sub
  FG2 = TxtEdit
  TxtEdit.Visible = False
End Sub
Sub Fg2_LeaveCell()
  If TxtEdit.Visible = False Then Exit Sub
  FG2 = TxtEdit
  TxtEdit.Visible = False
End Sub
Private Function getEW(soilclass As String, Ptype As String) As Single
Dim TEW, REW, Tfc, Twp As Single
If soilclass = "Sand" Then
TEW = 10: REW = 5: Tfc = 0.12: Twp = 0.05
ElseIf soilclass = "Loamy sand" Then
TEW = 12: REW = 6: Tfc = 0.15: Twp = 0.07
ElseIf soilclass = "Sandy loam" Then
TEW = 18: REW = 8: Tfc = 0.23: Twp = 0.11
ElseIf soilclass = "Loam" Then
TEW = 20: REW = 9: Tfc = 0.25: Twp = 0.12
ElseIf soilclass = "Silt Loam" Then
TEW = 22: REW = 10: Tfc = 0.29: Twp = 0.15
ElseIf soilclass = "Silt" Then
TEW = 23: REW = 10: Tfc = 0.32: Twp = 0.17
ElseIf soilclass = "Silt clay loam" Then
TEW = 24: REW = 10: Tfc = 0.34: Twp = 0.2
ElseIf soilclass = "Silty clay" Then
TEW = 25: REW = 10: Tfc = 0.36: Twp = 0.22
ElseIf soilclass = "Clay" Then
TEW = 26: REW = 10: Tfc = 0.36: Twp = 0.22
End If
If Ptype = "TEW" Then
getEW = TEW
ElseIf Ptype = "REW" Then
getEW = REW
ElseIf Ptype = "Diff" Then
```

```
getEW = Tfc - Twp
End If
End Function

Private Function getKcb(Gdays As Integer) As Single
Dim kc, liniG, Ld, Lm, Ltot, kiniG, KmidG, kendG, LdevG, LlateG As Single
liniG = Val(TxtLini.Text): LdevG = Val(TxtLdev.Text): LmidG = Val(TxtLmid.Text):
LlateG = Val(TxtLlate.Text)
kiniG = Val(Txtkini.Text): KmidG = Val(TxtKmid.Text): kendG = Val(TxtKend.Text)
Ld = liniG + LdevG
Lm = liniG + LdevG + LmidG
Ltot = Val(TxtLtotal.Text)
If Gdays > Ltot Then
MsgBox ("The days you typed more then total time of crop season")
Exit Function
End If
If Gdays <= liniG Then
kc = kiniG
ElseIf Gdays > liniG And Gdays <= Ld Then
kc = kiniG + (KmidG - kiniG) * (Gdays - liniG) / LdevG
ElseIf Gdays > Ld And Gdays <= Lm Then
kc = KmidG
Else
kc = KmidG - (KmidG - kendG) * (Gdays - Lm) / LlateG
End If
getKcb = Format(kc, "0.000")
End Function
```

Appenxix B:

Crop listed in Interface to be choosen and their crop parameters from FAO 56.

Name	Kini	Kmid	Kend	h	Lini	Ldev	Llate	Rz	ro
PADDY	0.042	1.137	0.434	0.3	30	60	30	0.5	0.45
JOWAR	0.042	1.137	0.434	0.4	30	60	30	0.7	0.45
MAIZE	0.042	1.137	0.434	0.3	30	60	30	0.8	0.35
BAJRA	0.042	1.137	0.434	0.4	30	60	30	0.6	0.45
SMALL MILLETS	0.042	1.137	0.434	0.6	30	60	30	0.4	0.2
KODO-KUTKI	0.042	1.137	0.434	0.3	30	60	30	0.4	0.3
TUR	0.042	1.137	0.434	0.4	30	60	30	0.5	0.3
URAD	0.042	1.137	0.434	0.3	30	60	30	0.5	0.3
MOONG	0.042	1.137	0.434	0.5	30	60	30	0.5	0.3
KULTHI & OTHERS	0.042	1.137	0.434	0.3	30	60	30	0.4	0.2
SOYBEAN	0.042	1.137	0.434	0.3	30	60	30	0.4	0.3
GROUNDNUT	0.042	1.137	0.434	0.8	30	60	30	1	0.45
SESAMUM	0.042	1.137	0.434	0.7	30	60	30	0.7	0.3
NIGER	0.042	1.137	0.434	0.6	30	60	30	1.1	0.4
CASTOR	0.042	1.137	0.434	0.3	30	60	30	1.2	0.4
SUNFLOWER	0.042	1.137	0.434	0.3	30	60	30	1	0.5
COTTON	0.042	1.137	0.434	0.4	30	60	30	1.2	0.35
WHEAT	0.28	1.105	0.315	0.4	25	75	30	1.2	0.4
BARLEY	0.28	1.105	0.315	0.4	25	75	30	0.8	0.4
JOWAR RABI	0.28	1.105	0.315	0.6	25	75	30	0.5	0.35
GRAM	0.28	1.105	0.315	0.4	25	75	30	1.2	0.65
PEAS	0.28	1.105	0.315	0.5	25	75	30	1	0.55
LENTIL	0.28	1.105	0.315	0.4	25	75	30	0.6	0.45
TEORA	0.28	1.105	0.315	0.4	25	75	30	0.8	0.45
URAD RABI	0.28	1.105	0.315	0.4	25	75	30	0.7	0.5
MOONG RABI	0.28	1.105	0.315	0.5	25	75	30	0.7	0.5
KULTHI RABI	0.28	1.105	0.315	0.8	25	75	30	1	0.5
RAPE / MUSTARD	0.28	1.105	0.315	0.7	25	75	30	0.7	0.45
LINSEED	0.28	1.105	0.315	0.5	25	75	30	1.5	0.45
SUNFLOWER	0.28	1.105	0.315	1.4	25	75	30	1.3	0.65
SAFFLOWER	0.28	1.105	0.315	1.2	25	75	30	1.3	0.5
SUGARCANE	0.28	1.105	0.315	0.3	25	75	30	1.5	0.5

Appenxix C:

Soil type listed in Interface to be choosen and their propporties used in this study.

Soil Type	θ_{FC} m^3/m^3	θ_{WP} m^3/m^3	TEW	REW
Sand	0.12	0.05	10	5
Loamy sand	0.15	0.07	12	6
Sandy loam	0.23	0.11	18	8
Loam	0.25	0.12	20	9
Silt loam	0.29	0.15	22	10
Silt	0.32	0.17	23	10
Silt clay loam	0.34	0.20	24	10
Silty clay	0.36	0.22	25	10
Clay	0.36	0.22	26	10