

**A GIS-BASED MODELING APPROACH FOR
IMPLEMENTATION OF SUSTAINABLE
FARM MANAGEMENT PRACTICES:
APPLICATION TO FIELD LEVEL
NITROGEN MODELING**

By

MAHESH N. RAO

**Bachelor of Science
Andhra Pradesh Agricultural University
Hyderabad, India
1986**

**Master of Science
Andhra Pradesh Agricultural University
Hyderabad, India
1988**

**Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 1996**

Thesis
1996D
R215g

A GIS-BASED MODELING APPROACH FOR
IMPLEMENTATION OF SUSTAINABLE
FARM MANAGEMENT PRACTICES:
APPLICATION TO FIELD LEVEL
NITROGEN MODELING

Thesis Approved:

David A. Waits

Thesis Advisor

Grant R. Schuler

Jan G. Gibe

Howard Ellett

Thomas C. Collins

Dean of Graduate College

ACKNOWLEDGMENTS

I wish to thank my major advisor, Dr. David Waits, for the support he provided throughout my degree, and help me learn to look at the ‘big’ picture. I would also like to thank Drs. Tom Wikle, Dan Storm, Frank Schiebe, and Norm Elliott for serving as members of my graduate committee, and for their valuable comments and suggestions.

I am thankful for the University Center for Water Research for providing me the financial support enabling me to pursue this degree. I am also appreciative of the OSU Department of Geography for all the additional support throughout my degree.

I would like to thank Dr. Mitch Nielsen, Assistant Professor, Computer Science Department for his help with this project. Special thanks go to Dewain Davis for his agricultural insights and John Anderson for making the datasets available for the project.

The support and encouragement of my family back home, “The RAOs,” has been the foundation for all my achievements. Sincere thanks are also due for each and everyone in the Department of Geography for their support and friendship during my study period.

To my wife, Tally. Words fail to express her patience, support, and sacrifices. Thanks for putting up with me, I find it hard to imagine how things would have turned out without you.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
Statement of Problem.....	3
Goals and Specific Objectives	4
Justification of the study.....	5
II. REVIEW OF LITERATURE.....	6
Sustainable Agriculture	6
Simulation Modeling.....	9
Geographic Information Systems (GIS).....	13
Integrated Approach: GIS-Based Modeling.....	17
Summary	20
III. MODEL SELECTION, DESCRIPTION, AND INTEGRATION.....	21
1. Model Selection.....	21
2. Model Description	24
Model Operation.....	27
Model Structure.....	28
Basic User-Supplied Data File	28
Crop Parameter File (USDACROP.DAT).....	30
Tillage Parameter File (CLASSTILL.DAT).....	31
The Pesticide Parameter File (PEST5300.DAT)	31
The Fertilizer Parameter File (FERT5300.DAT).....	31
Miscellaneous Parameter File (PARM5300.DAT).....	32
The Graphics Control File (GRAF5300.DAT).....	32
The Multi-Run File (MLRN5300.DAT).....	32
The Output Variable File (PRNT5300.DAT).....	32
The Daily Weather Data File.....	33
EPIC's Universal Text Integration Language (UTIL)	33
3. Integrated System - Design and Development	35
Integrated System Components	35
ArcView 2.1.....	35
Graphical User Interface (GUI)	40
EPIC-View - Requirements Specifications.....	41

System Interfacing Tools.....	44
ArcView Scripting Language - Avenue.....	44
Visual Basic 4.0.....	45
Definitions:	46
Management Unit.....	46
Curve Number (CN).....	47
UTIL Files.....	48
FORM#.DAT.....	49
SHELL Function.....	49
FORM.PRS.....	50
System Interface Development	51
Data Input Interface	51
Weather Data	51
Soil Data	51
Constant Data.....	54
Management Data.....	56
Spatial Data.....	58
Modeling Interface	58
Output Options.....	60
Model Run	60
Display Interface	63
IV. APPLICATION OF THE INTEGRATED SYSTEM TO NITROGEN	
MODELING	66
Introduction.....	66
EPIC Applications : An Overview.....	67
Crop Productivity.....	67
Soil Degradation	67
Response to Climate and Soil.....	68
Climate Change.....	69
Input Levels and Management Practices	69
Nitrogen Modeling.....	71
Nitrogen Cycling.....	71
Study Area.....	75
Methods	76
Development of GIS Database	79
Development of Hydrologic Response Units (HRUs)	79
Development of EPIC Base Data.....	81
Development of Management Practices Database	82
Simulation of Farm Level Nitrogen.....	83
Creation and Display of N Modeling Results	86
Results and Discussion.....	86
Integrated System Framework and Design.....	87
Data Coverages and Hydrologic Response Units	88

V. CONCLUSIONS.....	132
LITERATURE CITED.....	135
APPENDICES.....	145
APPENDIX I.a - Sample EPIC Input Dataset.....	146
APPENDIX I.b - Sample UTIL Batch File.....	148
APPENDIX II.a - EPIC-View: Program Code for Weather Data Tool.....	150
APPENDIX II.b - EPIC-View: Program Code for Soil Data Tool.....	153
APPENDIX II.c - EPIC-View: Program Code for Constant Data Tool.....	159
APPENDIX II.d - EPIC-View: Program Code for Management Data Tool.....	169
APPENDIX II.e - EPIC-View: Program Code for Spatial Data Tool.....	192
APPENDIX II.f - EPIC-View: Program Code for Output Options Tool.....	195
APPENDIX II.g - EPIC-View: Program Code for Model Run Tool.....	199
APPENDIX II.h - EPIC-View: Parser Program Code.....	204
APPENDIX II.i - EPIC-View: Program Code for Display Tool.....	209
APPENDIX II.j - EPIC-View: Miscellaneous Application Program Codes.....	212
APPENDIX III.a - EPIC-View: Base Data Input Values for the Various Parameters ...	219
APPENDIX III.b - EPIC-View: Management Data Values for the Various Parameters.....	221
APPENDIX III.c - EPIC-View: Botchlet Farm Specific Management Data - Corn Soybean Rotation under Conventional Tillage System.....	223
APPENDIX III.d - EPIC-View: Botchlet Farm Specific Management Data - Corn Soybean Rotation under Minimum Tillage System.....	224

LIST OF TABLES

Table		Page
1.	Sources of spatial data used in the development of GIS database	80
2.	Output variables pertaining to the nitrogen nutrient cycling component of EPIC.....	84
3.	Summary statistics for the different nitrogen variables modeled using EPIC-View or conventional tillage conditions.....	109
4.	Balance sheet of nitrogen estimated using mean values of the different N variables for conventional tillage conditions.....	110
5.	Summary statistics for the different nitrogen variables modeled using EPIC-View for minimum tillage conditions.....	124
6.	Balance sheet of nitrogen estimated using EPIC-View for minimum tillage conditions.....	125
7.	Comparison of coefficient of variation (CV) for the different nitrogen variables under conventional and minimum tillage conditions.....	130

LIST OF FIGURES

Figure		Page
1.	EPIC System File Structure (Dumesnil 1993).....	29
2.	Schematic representation of the integrated system components.....	36
3.	EPIC-View interface design framework.....	42
4.	Steps involved in storing latitude and longitude coordinates for accessing weather data file.....	52
5.	Steps involved in storing soil information, and soil code for accessing EPIC soil data file.....	53
6.	Processes involved in creating base data files (FORM#.DAT).....	55
7.	Steps involved in creating management data UTIL files.....	57
8.	Processes involved in creating spatial data UTIL file (FORM#.UTL).....	59
9.	Program flow outline for specifying EPIC output options.....	61
10.	Program flow during a model run.....	62
11.	Steps involved in deriving thematic maps of the model results.....	65
12.	Schematic representation of N transformations in a general agricultural N cycle (Legg and Meisinger 1982).....	73
13.	Study area of the project situated in Caddo County, OK.....	77
14.	Schematic representation of N modeling project.....	78
15.	Map showing major land cover categories existing in the study area.....	89

16.	Map of the major soil series existing in the study area.....	90
17.	Map showing the elevation (m) categories existing in the study area.....	91
18.	Map showing the slope categories (%) existing in the study area.....	92
19.	Hydrologic response units (HRUs) delineated by overlaying crop, soil, elevation, and slope coverages.....	94
20.	Map of average N uptake (kg ha^{-1}) by crop (UNO3) estimated using EPIC-View under conventional tillage conditions.....	96
21.	EPIC-View estimated map showing average annual mineral N loss (kg ha^{-1}) in percolate (PRKN) under conventional tillage conditions.....	97
22.	EPIC-View estimated map of average annual rate (kg ha^{-1}) of N volatilization (AVOL) under conventional tillage conditions.....	98
23.	Map showing average annual nitrate (kg ha^{-1}) loss in surface runoff (YNO3) estimated using EPIC-View under conventional tillage conditions.....	99
24.	EPIC-View estimated map of average annual mineral N loss (kg ha^{-1}) in subsurface flow (SSFN) under conventional tillage conditions.....	100
25.	Map showing EPIC-View estimated average annual organic N (kg ha^{-1}) loss with sediment (YON) under conventional tillage conditions.....	101
26.	EPIC-View estimated map of average annual rate (kg ha^{-1}) of nitrification (NITR) under conventional tillage conditions.....	102
27.	EPIC-View estimated map of average annual rate (kg ha^{-1}) of N fixation (NFIX) under conventional tillage conditions.....	103
28.	EPIC-View estimated map of average annual rate (kg ha^{-1}) of N mineralization (MNN) under conventional tillage conditions.....	104
29.	EPIC-View estimated map of average annual rate (kg ha^{-1}) of N mineralized from stable organic matter (HMN) under conventional tillage conditions.....	105
30.	Map showing EPIC-View estimated map of average annual $\text{NH}_3\text{-N}$ (kg ha^{-1}) present in the soil profile (TNH3) under conventional tillage conditions.....	106
31.	Map showing EPIC-View estimated average annual $\text{NO}_3\text{-N}$ (kg ha^{-1}) present in the soil profile (TNO3) under conventional tillage conditions.....	107

32.	EPIC-View estimated map of average annual N uptake (kg ha ⁻¹) by crop (UNO3) under minimum tillage conditions.	111
33.	EPIC-View estimated map of average annual mineral N loss (kg ha ⁻¹) in percolate (PRKN) under minimum tillage conditions.	112
34.	EPIC-View estimated map of average annual rate (kg ha ⁻¹) of N volatilization (AVOL) under minimum tillage conditions.	113
35.	EPIC-View map showing average annual nitrate loss (kg ha ⁻¹) in surface runoff (YNO3) under minimum tillage conditions.	114
36.	EPIC-View estimated map showing average annual mineral N loss (kg ha ⁻¹) in subsurface flow (SSFN) under minimum tillage conditions.	115
37.	EPIC-View estimated map of average annual organic N loss (kg ha ⁻¹) with sediment (YON) under minimum tillage conditions.	116
38.	EPIC-View estimated map of average annual rate of nitrification (kg ha ⁻¹) under (NITR) minimum tillage conditions.	117
39.	EPIC-View estimated map of average annual rate (kg ha ⁻¹) of N fixation (NFIX) under minimum tillage conditions.	118
40.	EPIC-View estimated map of average annual rate (kg ha ⁻¹) of N mineralization (MNN) under minimum tillage conditions.	119
41.	EPIC-View estimated map of average annual rate (kg ha ⁻¹) of N mineralization from stable organic matter (HMN) under minimum tillage conditions.	120
42.	EPIC-View estimated map of average annual NH ₃ -N (kg ha ⁻¹) present in the soil profile (TNH3) under minimum tillage conditions.	121
43.	EPIC-View estimated map of average annual NO ₃ -N (kg ha ⁻¹) present in the soil profile (TNO3) under minimum tillage conditions.	122
44.	Map showing average annual corn yield (Mg ha ⁻¹) estimated using EPIC-View for conventional tillage conditions.	126
45.	Map showing average annual soybean yield (Mg ha ⁻¹) estimated using EPIC-View for conventional tillage conditions.	127
46.	Map showing average annual corn yield (Mg ha ⁻¹) estimated using EPIC-View for minimum tillage conditions.	128
47.	Map showing average annual soybean yield (Mg ha ⁻¹) estimated using EPIC-View for minimum tillage conditions.	129

CHAPTER I

INTRODUCTION

The agriculture of the twenty-first century will be more industrialized than it is today, with less than 300,000 large farms accounting for four-fifths of farm output (Tweeten 1994). The technologies underlying this phenomenal success place great demands on rural environments and natural resources. The long-term sustainability of such a technology-intensive agriculture system depends upon a recognition of economic, socioeconomic, and natural resources conservation issues.

Concerns over environmental impacts, resource depletion, safety of conventionally produced food, the vigor of rural society, and the expense of government farm programs all underlie calls for new approaches to agriculture (U.S. Government Accounting Office 1990). A new umbrella term --“Sustainable Agriculture”-- emerged to encompass the kinds of new approaches to farming which include ways to protect the environment, conserve natural resources, reduce the use of potentially toxic chemicals, increase farm productivity and efficiency, and increase financial independence. The report on Alternative Agriculture (National Research Council 1989) was a pivotal contribution which has helped bring the issue of sustainable agriculture to the attention of general and scientific publics to a degree not previously experienced. There are numerous definitions and interpretations of sustainable agriculture including the following -- sustainable agriculture requires

management of the land so that production and productivity are enhanced while sustaining a healthy ecological balance within the agricultural ecosystems (Ruttan 1990).

Sustainable agriculture requires the farmer to be well-informed, possessing management and analytical skills. In order to manage farm inputs effectively, the farmer must be able to comprehend the complex interrelationships among crops, weeds, pests, and soil fertility in order to overcome risks of crop damage and encourage the factors that maintain competitive yields (Crosson 1989). An important underlying concept of sustainable agriculture is to bring about changes in farm practices which improve efficiency in the use of all resources and increase the profitability of the farming enterprise. Some of the various interrelated components of a sustainable agriculture system are soil management including fertility, irrigation, crop management, economics of farming significantly affecting crop production, and the environment. An effective sustainable agriculture system strives to develop a farming strategy involving the various components such that crop productivity is maintained while reducing adverse environmental impacts. Field evaluations of these management practices are cumbersome, error prone, impractical, and expensive. Computer models can simulate the various management scenarios, and thus can serve as an efficient and cost effective alternative. Computer models aid in the evaluation of alternative chemicals, rates, and timing of application as well as evaluation of various field management practices (Kalita et al. 1992).

An important characteristic of environmental modeling is that all the basic units (water, soil and chemicals) have a spatial distribution, and since this distribution does affect the processes and dynamics of their interaction considerably, geographic

information systems (GIS) are emerging as an important tool in modeling. GIS softwares are developed to capture, manipulate, process and display spatial or georeferenced data.

The advantages of using recent advances in computer technology like GIS, and systems integration have not been fully studied from a farmer's perspective. This research effort addresses the design and development of a GIS-based, farm level modeling approach.

Statement of Problem

GIS-based modeling is restricted mostly to the research communities and agencies of the federal and state governments for purposes of model testing, model refinement, and other activities including targeting nonpoint-source pollution areas and prioritizing funds for conservation. Irrespective of the scale at which various nonpoint source models operate, it is known that management practices geared towards conservation and productivity are initiated at the field level. At present, however, few agricultural producers are utilizing the true analytical power of GIS and computer simulation models, partly because the loose linkages developed to-date between GIS and mostly public-domain modeling software are extremely cumbersome to use or are esoteric. Therefore, to fully unleash the power of the GIS integrated nonpoint source modeling system, it should come out of the realm of research and development labs and become available to the farming community. This "lab to land" transformation is feasible through a user friendly GUI (graphical user interface) which walks the user through the various steps of GIS-based modeling. Thus, a need exists for a fully integrated, user-friendly, GIS modeling system to allow agricultural producers to simulate soil erosion, plant growth and related processes,

and economic components for assessing the cost of erosion and the benefits of using different management strategies.

Goals and Specific Objectives

The goal of this study is to create and implement a single, seamless interactive system that fully integrates a hydrologic-crop management model with ArcView (Version 2.1), a desktop GIS (Environmental Systems Research Institute 1994a) to function as a planning tool aimed at implementing sustainable farm management practices. The basic tenet associated with this goal is to facilitate the data flow and consistency between the GIS and the simulation model. The specific objectives are:

1. Identify and select an appropriate model that can provide relevant information about the effects of cropping and management practices on crop productivity and water quality at the sub-field scale.
2. Integrate the model with ArcView using an object-oriented scripting language called Avenue (Environmental Systems Research Institute 1994b) in order to facilitate the development of a user friendly framework for model use in terms of model parameter input, and display of model output.
3. Apply the integrated system to farm level nutrient nitrogen dynamics and evaluate the results in terms of a nitrogen balance.

Justification of the Study

Agricultural producers must often deal with the variability of soil characteristics and fertility within fields. Excessive application of fertilizers and chemicals is costly for the

producer and increases the potential for negative environmental impacts. On the other hand, insufficient application leads to diminished economic return as there is a reduction in crop yields. Furthermore, insufficient nutrients retard plant development and the ability of roots to hold soil in place, increasing the potential for erosion (Richter and Tank 1991).

This research highlights the importance of a GIS-based modeling approach as an important tool that can be used by producers and farm consultants to maximize crop productivity and minimize the adverse effects of nonpoint source (NPS) pollution. Also, an important tool will be available to producers through which they can demonstrate to the environmental regulatory agencies the sustainability of the farm and justify the farming practices. Resource managers will be able to identify land areas with critical NPS pollution problems and make priorities toward implementation of conservation programs and best management practices. Added benefits will accrue to the producer from the ability to deal with the inherent variability at the field scale by adopting the concept of variable rate technology (VRT) -- customizing the rates of farm inputs such as fertilizers at every location in an agricultural field. In addition to agricultural producers, environmental modelers will benefit as they engage in research pertaining to model validation, sensitivity analysis, and model calibration to further fine tune the model and enhance its performance. Model datasets will be assembled and ready for runs in a shorter period of time, thus increasing the efficiency of research activities.

CHAPTER II

REVIEW OF LITERATURE

The first chapter introduced terms such as sustainable agriculture, simulation modeling, GIS, and integrated GIS modeling system. With an intent to help understand the respective role played by these factors in attaining a sustainable system, this chapter reviews some important concepts. The review is organized as follows:

1. Sustainable Agriculture
2. Simulation Modeling
3. Geographic Information Systems (GIS)
4. Integrated Approach: GIS-Based Simulation Modeling
5. Summary

1. Sustainable Agriculture

Conventional agriculture in the United States includes: capital intensive monocultures; continuous cropping; substantial reliance on manufactured inputs such as fertilizers, pesticides, and machinery; and heavy dependence on credit and government subsidies (Neher 1992, Pesek 1994). However, conventional agriculture is increasingly perceived as unsustainable (Parr et al. 1990, Senanayake 1991).

Agriculture is identified as the largest nonpoint source of surface water pollution, and pesticides and nitrate from fertilizers have been detected in the groundwater in many agricultural regions. Contamination of groundwater from routine use of agricultural pesticides or nitrates has been documented in 36 states (O'Hare et al. 1985; Cohen et al. 1984). Productivity is reduced as agricultural soils are eroded, and their fertility and physical and biological conditions degraded (National Research Council 1989).

The National policy on agriculture also had a major influence. About 70% of the nation's cropland is in crops covered by federal commodity programs and some 88% of the eligible production is enrolled (National Research Council 1989). The size of a farmer's subsidy is determined by the base acreage and the base yield of land in program crops. The farmer, therefore, is encouraged to produce as much as possible on each acre and to keep the highest possible acreage of land in program crops. This may have led to overapplication of fertilizers and other chemicals and cultivation of fragile land (National Research Council 1989). Eventually, growing evidence and awareness of the imprudent use of natural resources, causing environmental deterioration and reduced productivity, led to a distrust by non-agricultural groups interested in food quality and safety, natural resources, environmental quality, and human resources (Davidson 1989). This led to one of the major challenges to agriculture - to develop farming systems that can produce quantity and quality of food and fiber without adversely affecting our natural resources. Subsequently, many farming systems have been proposed that are assumed to be sustainable. The American Society of Agronomy has adopted the following definition of sustainable agriculture (Schaller 1990):

“Sustainable agriculture is one that over the long-term, enhances environmental quality and the resource base on which agriculture depends, provides for basic human food and fiber needs, is economically viable, and enhances the quality of life of farmers and society as a whole.”

These sustainable farming systems are categorized using various labels such as *organic farming*, *alternative agriculture*, and *low-input/sustainable agriculture*.

However, the alternative to conventional farming is not a single set of practices but an array of farming systems tailored to particular conditions (Stockle et al. 1994). Sumner and Miller (1992) present various strategies such as reduced tillage methods, mulching, rotational pastures and cover crops to reduce soil degradation. The authors suggest that these practices have the greatest potential both to protect the soil surface physically, and to gradually increase soil humus levels. Stinner and Blair (1990) state that multiple cropping, minimum tillage, agroforestry, and integration with animal husbandry all have the potential to sustain productivity using lower chemical and energy inputs. Sadler and Turner (1994) hypothesize that for agriculture to be sustainable and productive, crop water use must balance water available from rainfall, irrigation, and soil storage. The authors go on to add that both crop and soil management strategies are available to enable this balance to be achieved.

Sustainable farming practices involve a variety of approaches. Specific strategies must take into account topography, soil characteristics, climate, pests, cost and local availability of inputs, and the individual grower's goals. With so many factors to consider, it becomes difficult to determine the emphasis that should be given to a particular factor.

To address these concerns, there is a continuing search for tools which can provide better understanding of the interactions among the various components of sustainable farming. Stockle et al. (1994) propose a framework for evaluating the relative sustainability of a farming system using nine attributes: profitability, productivity, soil quality, water quality, air quality, energy efficiency, fish and wildlife habitat, quality of life, and social acceptance. These attributes are constrained by different factors. For example, profitability is constrained by low net income, high input costs, no markets, etc. Similarly, water quality is constrained by chemical runoff, chemical percolation and sedimentation, etc. Some of these constraints can be quantified by direct measurements. However, in many situations, measurements are not feasible because of limits in time or resources. The authors state that simulation modeling opens a new option; although, it may not be as reliable as direct measurements, when it is used with judgment it can produce satisfactory results. The authors add that because the objective is to estimate the relative sustainability of farming systems, great accuracy may not be required. Furthermore, simulation modeling helps in projecting the long-term rate of change of individual constraints.

2. Simulation Modeling

Rapid advances in computer technology allowed the development of sophisticated models to help in management and decision-making. Webster's dictionary defines a model as "a generalized, hypothetical description, often based on an analogy, used in analyzing or explaining something." From a natural resources conservation perspective, models are developed as a result of the need to better understand natural phenomena and to find the most

appropriate methods of production and conservation. Models improve the understanding of hydrologic processes in operation, their variability, and future research needs (Turton et al. 1988). Modeling, therefore, aims to present a crude approximation of reality. The degree of simulation success depends on the quality of the model and the accuracy of the database (Hodge et al. 1988). Moreover, applications in research, management, and education regarding certain conditions in the environment determine the purpose of the model.

A number of models representing natural processes have been developed for varied reasons and applications (Ball 1994, Varis 1994, U.S. Department of Agriculture 1989). The natural processes that need to be modeled and the purpose for which it will be applied determine the range of complexity of the model in terms of structure, design, data requirements, and facility use.

Computer simulation models are increasingly being used to study the complex processes involving soil, climate and plant variables (Hanks and Ritchie 1991, Haan et al. 1982). More comprehensive models have been developed, such as crop environment resource synthesis (CERES), which not only simulate growth and yield of several crops but also predict the effects of varying N fertilizer rate, timing, placement depth, and source on fertilizer use efficiency within a growing season (Godwin and Jones 1991, Jones and Ritchie 1991). SOYGRO (Wilkerson et al. 1983) is a physiologically based growth model which simulates dry-matter, vegetative and reproductive development, leaf area index, and yield as a function of both management and environmental variables specified by the user. 'Century' simulates the dynamics of C, N, P, and S cycling in soil-plant systems using a monthly time step and incorporates the effects of moisture, temperature, soil properties, plant phenology, and organic

matter decomposition on nutrient flows (Patron et al. 1988). Hill (1991) proposed the PCET and SOIL WATER BUDGET programs for crop irrigation modeling. PCET uses ET_r (ET of reference crop) calculated from weather data usually for a 7-d update period while the SOIL WATER BUDGET program determines when to irrigate an individual field based on output from the PCET program and site-specific soil, rain and irrigation data input by the irrigator. Carcamo et al. (1994), using linear programming and MOTAD, studied on-site costs and benefits of alternative soil conservation practices for subsistence farmers in Honduras and discussed policy implications.

The early seventies saw a general awareness and concern about environmental pollution. Under section 208 of the Federal Water Pollution Control Act of 1972 (Nonpoint Source Task Force 1984), planning agencies throughout the United States were given the responsibility of identifying areas having substantial water quality problems and regulatory programs. An important element of this water quality protection legislature is the control of NPS pollution of surface waters by sediments and nutrients (Tim et al. 1992). This provided a major thrust to hydrologic/nonpoint source water quality modeling both at the watershed and field scales. Calvin et al. (1988) provide a good treatise on this topic.

Few models exist which can be termed as management oriented holistic models. The erosion-productivity impact calculator (EPIC) is one such model (Sharpley and Williams 1990). It consists of physically based components for simulating erosion, plant growth and related processes and economic components for such assessments as the cost of erosion and determining optimum management strategies. Extensive work has been carried out focusing on

some of the important components of EPIC: weather simulation (Williams et al. 1992); hydrology, erosion, and sedimentation (Benson 1989); nutrient cycling (Sharpley et al. 1990); crop growth (Martin et al. 1993); tillage (Phillips et al. 1993); economics (Lee and Lacewell 1989); and plant environment control (Robertson et al. 1987). Chemicals Runoff and Erosion from Agricultural Management Systems (CREAMS) is another such model developed to describe infiltration, tillage, crop growth, erosion, sediment, plant nutrients, and pesticide movements (Knisel 1980). Simulator for Water Resources in Rural Basins (SWRRB) was developed from CREAMS (Arnold et al. 1990). SWAT (Soil and Water Assessment Tool) provides several extensions to the SWRRB model for assessing water supplies and nonpoint source pollution on watersheds and large basins (Arnold et al. 1993). In addition to simulating runoff, sediment, nutrient, and pesticide movement through a watershed, SWAT provides considerable flexibility in watershed configuration and discretization allowing watersheds to be subdivided into cells and/or subwatersheds. The Nitrogen, Tillage, Residue Management Model (NTRM) is useful in nitrogen, tillage and crop residue management. Also, it is capable of assessing seasonal and long-term effects of soil fertility on crop yields (Shaffer et al. 1983). Areal Nonpoint Source Watershed Environment Response Simulation (ANSWERS) was developed to supply information concerning the effects that land use, management and conservation practices or structures might have on the quality and quantity of water from both agricultural and non-agricultural watersheds (Beasley et al. 1980). Agricultural Nonpoint Source Pollution (AGNPS) is an event based model which simulates runoff, sediment and nutrient transport from agricultural watersheds (Young et al. 1989). WEPP (Water Erosion

Prediction Project) can be used to predict soil erosion, plant growth, and soil water balance on a daily basis (Lafren et al. 1991).

Thus there are various environmental models; however, the major impediments in using these models include intensive data requirements and time involved in compiling the model input file. Geographic Information Systems (GIS) help overcome these problems to a great extent (Jankowski and Haddock 1993). Furthermore, GIS can be used to extract or derive spatial parameters such as topographic parameters for hydrologic models.

3. Geographic Information Systems (GIS)

Geographic Information Systems (GIS) present a logical computing environment to manipulate, analyze, and model diverse data sources for the purpose of research, planning, and decision-making (Keller 1991). A GIS can also be thought of as a specialized form of database management system (Bracken and Webster 1990). It differs from other database management systems by its ability to handle spatially-referenced geographical data as graphical maps (Valenzuela 1991). Some reasons for using GIS are enumerated by Dangermond (1991):

1. Data are maintained in physically compact format (i.e., the magnetic file).
2. Data can be maintained and extracted at a lower cost per unit of data handled.
3. Data can be retrieved with much greater speed.

4. Various computerized tools allow a variety of types of manipulation including map measurement, map overlay, transformation, graphic design, and database manipulation.

5. Graphic and non-graphic (i.e., attribute information) data can be merged and manipulated simultaneously.

6. Rapid and repeated analytical testing of conceptual models about geography can be performed (e.g., land suitability/capability). This facilitates the evaluation of both scientific and policy criteria over large areas in short periods of time.

7. Change analysis can be efficiently performed for two or more different time periods.

8. Interactive graphic design and automated drafting tools can be applied to cartographic design and production.

9. Certain forms of analysis can be performed cost effectively that simply could not be done efficiently if performed manually (e.g., digital terrain analysis, calculation of slope, aspect, solar intensity, watershed, and overlay analysis of multiple sets of complex polygon maps.).

10. There is a resultant tendency to integrate data collections, spatial analysis, and decision-making processes into a common information flow context. This has great advantage in terms of efficiency and accountability.

GIS technology has evolved as a means of assembling and analyzing diverse spatial data. The origins of geographic information systems go back to the mid-eighteenth century when the first accurate base maps were produced. This led to the development of high quality base maps for accurate geographic depiction of spatial attributes and was subsequently followed by thematic mapping (Parent 1988). Some of the first thematic maps were produced by meteorologists for forecasting and portraying crude land contours (Coppock 1988). Perhaps the most significant development paving the way for adoption of GIS in agricultural research and management was creation of the Canada Geographic Information System (CGIS) by the Canadian government in the late 1960s. It had the ability to carry out continent-wide geographical analysis from high-resolution map data concerning many aspects of Canada's natural resources and socio-economic conditions (Tomlinson 1988).

A GIS database can be divided into graphic and non-graphic types of data (Antenucci et al. 1991). Graphic data are digital descriptions of map features which may include coordinates, rules, and symbols defining specific cartographic elements of a map. These data are translated to generate a map or cartographic picture on a display media. Non-graphic data, often called textual data or attributes, represent the characteristics, qualities or relationships of map features, and geographic locations.

Graphic data are represented either as a vector model or raster model. In the vector model, objects or conditions in the real world are represented by points and lines that define their boundaries as if drawn on a map (Crane and Herrington 1992). The location of each object is defined by its position in a coordinate reference system. In a

raster model, the space is regularly subdivided into cells. The location of each geographic object is defined by the row and column positions of the cells they occupy (Crane and Herrington 1992). In general, both the data structures can be used to represent any type of geographical data, depending on the scale desired for analysis. However, raster models fit best for modeling type of analysis involving natural resources data such as landuse, soils, and vegetation as there is spatial control for continuous variables and also uniform sampling of the surface being modeled. When the different thematic layers exist as grids with a common resolution, there is computational efficiency in overlaying those grids. Spatial analytical functionality is easy to implement and fast and efficient especially for operations such as spatial averaging and intersections which pose problems (slivers) in vector systems. For analysis involving linear data where precision and accuracy of the spatial feature is of utmost importance, such as detailed parcel boundaries, transportation network, gas pipelines, the raster data model is not suitable. Such types of applications are best suited for vector data models, owing to the accuracy and precision in storing the locational information in x and y coordinates. In addition, the raster data model presents a blocky appearance of the map feature and so applications where the quality of graphics display is of paramount importance, raster models are not recommended. Again, applications where graphic display is of importance such as transportation networks, parcel boundaries, the vector data model is best. The advantages and disadvantages of each data model are thoroughly discussed by Crane and Herrington (1992).

GIS applications software consisting of programs used to perform mapping and geographic operations have evolved to such an extent that a typical GIS user now expects

to be able to define requirements and interact with the system through a “user-friendly”, intuitive interface that makes use of such contemporary concepts as graphic icons and desktop metaphors (Mark and Gould 1991). Desktop GIS, an immensely powerful computer mapping system, is gaining in popularity. Desktop GIS combines all the capabilities of display-only, thematic, and street-based mapping systems along with the ability to analyze geographic locations and the information linked to those locations (Environmental Systems Research Institute 1995). Desktop GIS is dynamic, because a simple update of the linked attribute information of a map will automatically reflect those changes in the display. Furthermore, it is possible to access information from the maps or access the map from information in a simple point and click environment.

4. Integrated Approach: GIS-Based Modeling

A great deal of modeling is now done using GIS technology and other tools such as engineering models, site analysis models and surface flow (Dangermond 1991). The following are common levels of GIS integration in environmental modeling (Fedra 1991):

1. GIS and models exchange files. The model may read some of its input data from GIS files and produce some of its output in a format that allows processing and display in GIS.
2. Use a common interface and transparent file. This involves the sharing and transfer of information between the respective components.
3. Dedicated system. This is most common for a specific application wherein only a subset of GIS functions may be required.

4. Deep level. The model becomes one of the analytical functions of a GIS or the GIS becomes another option to generate additional state and output variables in the model.

GISs describe the spatial characteristics of the environment; environmental modeling simulates the functioning of environmental processes and such simulations require data describing the environment within which the processes occur. Furthermore, the simulation output provides additional data to better understand the environment. Thus, GIS and environmental modeling are synergistic, and GIS can serve as a common data and analysis framework for environmental models (Maidment 1993, Ragan 1991).

Furthermore, GISs are designed to store, organize and manipulate large amounts of spatial data such as elevation, soil mapping units, landuse, and hydrography (Shanholtz et al. 1990). GISs have been widely used in water-related studies to assess cumulative impacts on wetlands (Johnston et al. 1988), delineate soil salinity areas (Corwin and Rhoades 1988), represent factors in USLE (Universal Soil Loss Equation) to determine soil erosion (Chrismer et al. 1990), and estimate potential sediment loading to streams in agricultural areas.

GIS have been used to describe the areal extent of hydrologic response units (Wolfe and Neal 1988), estimate runoff volume using the SCS curve number method (Stuebe and Johnston 1990), and develop a model to predict landslide and gully erosion hazards (Jayawardana and Hill 1990).

The Oklahoma Geographic Information Retrieval System (OGIRS), a grid-oriented/cell-based system, was developed to expedite data integration problems and the time-consuming process of synthesizing tremendous amounts of information for the spatial examination of non-point pollution (Walsh 1985). Mellerowicz et al. (1994) used CARIS (Canadian Resource Information Systems) GIS to predict and display annual soil losses in a New Brunswick watershed based on the USLE, and to evaluate alternative management scenarios.

GRASS and ARC/INFO are two of the most widely used GISs in the raster and vector domains respectively. GRASS was used to determine the susceptibility of surface water to nonpoint pollution from dairy operations in Erath County, Texas (Deliman and Wolfe 1990). Chen et al. (1994) developed an application which integrated a phosphorus transport model directly in the GRASS GIS. The model predicts runoff, phosphorus and sediment yields, and then prioritizes phosphorus loading sources in a watershed, displays them as maps, and evaluates alternate management practices. A similar system was developed by Srinivasan and Engel (1994); they interfaced the distributed-parameter AGNPS pollutant runoff model with GRASS to provide event-based decision support options for a wider range of nutrient and sediment variables. Also, Srinivasan and Arnold (1994) linked the SWAT model to GRASS and found average monthly predicted streamflow in agreement with measured values in a Texas watershed.

On the other hand, Tim and Jolly (1994) linked AGNPS with ARC/INFO and simulated the influence of vegetated filter strips, contour buffer strips or a combination of the two in reducing sediment yield in a southern Iowa watershed. Similarly, Lo (1994)

used an integrated AGNPS and ARC/INFO system to compare simulated and measured sediment yields for a watershed in Taiwan. Warwick and Hanes (1994) used ARC/INFO for providing spatial input to the HEC-1 hydrologic model.

5. Summary

One of the major challenges facing agriculture in this decade will be to develop sustainable farming systems which minimize the threat to the environment without sacrificing crop productivity. Sustainable farming is multi-faceted involving specific strategies that take into account soil characteristics, topography, climate, pests, costs and local availability of farm inputs, and the grower's goals. A GIS based modeling approach provides the capability to describe and better understand the interactions among the various components of sustainable farming.

This chapter reviews some of the important concepts associated with simulation modeling, GIS, and an integrated approach to help understand their respective role in attaining a sustainable system. There is ample literature pertaining to GIS-based modeling from a research perspective. However, a paucity of literature exists from a farmer's perspective. It is important to recognize the need of the agricultural community to adopt sustainable farming practices. This research effort is a step in that direction.

CHAPTER III

MODEL SELECTION, DESCRIPTION, AND INTEGRATION

This chapter deals with the selection of a model for developing the integrated system, description of the model, and integration of the model with GIS. This chapter is structured into the following sections:

1. Model Selection
2. Model Description
3. Integrated System - Design and Development

1. Model Selection

Proper model selection is the most important step in any modeling exercise. The earlier chapter described a few environmental models. As is evident, a plethora of environmental models exist, each with unique characteristics and capabilities. In addition to this, optimistic claims by model developers, and conflicting modeling objectives, all create a dilemma for the model user in selecting a model for a particular objective. The choice of the best model depends on (Haan et al. 1994): 1. problem to be solved; 2. computer facilities available; 3. the likelihood of other applications for the model; 4. the availability of documentation and other forms of assistance; and 5. a user's modeling

experience. Haan et al. (1994) state that the model selected should match the problem to be solved; and, not the other way around. For example, if the problem requires continuous simulation of stream flow, selecting a model which simulates single events would reflect a bad choice. Also, the model selected should be compatible with the computer system in terms of input and output of data and results, operating systems, internal memory, disk storage, and graphics. The functionality of the model in terms of its applicability and use is also an important consideration. A well documented model not only helps operate the model and explain the algorithms used in the model, but guides the user in parameter estimation and error checking. At times, a user's interest and inclination play an important role in the selection process. It is important to realize that model complexity is not tantamount to model accuracy. Model complexity and the profuse parameters requirement of some detailed models are overwhelming. A simple model sensitive to the purpose of the modeling effort is the ideal choice.

In the selection process of the model for this project, it was necessary that the model provide agricultural managers with a powerful tool to assess simultaneously the affect of farm practices on crop production as well as on soil and water resources. Other model selection criteria included minimum data requirements to run the model, wide usage, and a reasonable accuracy in predictions. The following agricultural water quality models were examined: (1) CREAMS and GLEAMS; (2) AGNPS; (3) ANSWERS; (4) SWRRB; and (5) EPIC. Other models such as CERES, SOYGRO, NTRM, etc. were not considered because they had specific applications and hence were restrictive.

Although CREAMS and GLEAMS are field scale continuous models, they do not possess a robust crop growth model (Ramanarayanan 1994). ANSWERS and AGNPS were eliminated as they are watershed scale models simulating the effect of single rainfall events and are not suited for simulating the continuous nature of plant growth (Binger 1990). Also, the data requirements for these models are extensive because of the distributed nature of modeling. Furthermore, distributed models provide tools to advance one's understanding of physical processes, but for management purposes, their usefulness is limited (Grayson 1992). SWRRB and EPIC are almost synonymous, except for the fact that SWRRB is a basin scale simulation model. EPIC has improved residue handling capabilities over SWRRB, and better nutrient cycling. Also, EPIC has the added advantage of accounting for wind erosion (Binger 1990).

Based on the literature reviewed and expert opinion gathered from model developers, EPIC was selected for this project. Some additional model features that favored the selection of EPIC are (Dumesnil 1993):

1. EPIC is a continuous, field scale agricultural management/water quality model.
2. The EPIC model is broad-based in terms of its components to model major biophysical processes which include weather, hydrology, erosion, nutrients (nitrogen and phosphorus) cycling, pesticide fate, soil temperature, crop growth, tillage, plant environmental controls, and economics.
3. The data required by EPIC are minimal and readily available. The model provides parameter data files for major crops, soils, and tillage practices. EPIC is also equipped with a stochastic weather generator.

4. EPIC is well documented, relatively simple and easy to use. The model includes example datasets for new users.

5. EPIC is capable of simulating the long term effects of cropping systems on soil erosion and productivity in specific environments.

6. EPIC is accompanied by a utility package, UTIL, Universal Text Integration Language which is helpful for dataset assembly.

7. EPIC is an IBM-PC compatible, public domain model that requires only 640 KB of random access memory and 4.5 MB of disk space.

2. Model Description

In 1981, the National Soil-Erosion/Soil-Productivity Research Planning Committee, in a effort to meet the requirement of the Soil and Water Resources Conservation Act (RCA) which demanded a status report of soil and water resources in the United States, documented the effect of erosion on long-term soil productivity (Williams 1981). Furthermore, the committee identified additional knowledge that was needed, and outlined a research approach for solving the problem. One of the most urgent needs outlined was the development of a mathematical model for simulating erosion, crop production, and related processes. In the early 1980s, scientists from the USDA Agricultural Research Service (ARS), Soil Conservation Service (SCS), and Economic Research Service (ERS) set about developing EPIC. Four requirements of the model were: (a) it must be physically based and capable of simultaneously and realistically simulating the processes involved in erosion by using readily available inputs; (b) it must

be capable of simulating the processes as they would occur over hundreds of years, if necessary, because erosion can occur relatively slowly; (c) it must be applicable to a wide range of soils, climates, and crops encountered in the United States; and (d) it must be efficient, convenient to use, and capable of assessing the effects of management changes on erosion and soil productivity (Dumesnil 1993). Thus, EPIC consists of (a) physically based components for simulating erosion, plant growth, and related processes and (b) economic components for assessing the cost of erosion and determining optimal management strategies. Beyond the analysis for the RCA-mandated report, the potential use of EPIC was realized -- as a decision making tool for determining optimal management strategies from the farm to the national level, and also as a research tool (Sharpley and Williams 1990). As a decision making tool, EPIC can help identify optimal management strategies concerning drainage, irrigation, water yield, erosion control (wind and water), weather, fertilizer and lime applications, pest control, planting dates, tillage, and crop residue management. As a research tool, EPIC can be used in developing, testing, and refining model components for simulating various physical and chemical processes; in sensitivity analyses to determine the importance of experimental variables and their interactions; and in designing field experiments to obtain maximum information for minimum cost.

EPIC is composed of physically based components for simulating erosion, plant growth, and related processes and economic components for assessing the cost of erosion, and determining optimal management strategies. In EPIC, the major biophysical processes simulated, also called components, include (Dumesnil 1993):

1. **Weather:** Daily rain, snow, maximum and minimum temperature, solar radiation, wind and relative humidity can be based on measured data and/or generated stochastically.

2. **Hydrology:** Runoff, percolation, lateral subsurface flow, and snow melt are simulated.

3. **Erosion:** EPIC simulates soil erosion caused by wind and water. Sheet and rill erosion/sedimentation result from runoff from rainfall, snow melt, and irrigation.

4. **Nutrient Cycling:** The model simulates nitrogen and phosphorus fertilization, transformations, crop uptake and nutrient movement. Nutrients can be applied as mineral fertilizers, in irrigation water, or as animal manures.

5. **Pesticide Fate:** The model simulates pesticide movement with water and sediment as well as degradation on foliage and in soil.

6. **Soil Temperature:** Soil temperature responds to weather, soil water content, and bulk density. It is computed daily in each soil layer.

7. **Tillage:** Tillage equipment affects soil hydrology and nutrient cycling. The user can change the characteristics of simulated tillage equipment, if needed.

8. **Crop Growth:** A single crop model capable of simulating major agronomic crops, horticultural crops, pastures, and trees is used. Crop-specific parameters are available for most crops. The user may adjust or create new sets of parameters as needed. The model can also simulate crops grown in complete rotations.

9. **Crop and Soil Management:** EPIC is capable of simulating a variety of cropping variables, management practices, and other naturally occurring processes. These include

different crop characteristics, plant populations, dates of planting and harvest, fertilization, irrigation, artificial drainage systems, tillage, runoff control with furrow dikes and other methods, liming, and pest control. The model can also gauge the effects of varied management practices, such as whether the crop is harvested for grain or fodder or if it is grazed or burned.

10. Economics: A simple accounting package is included to calculate cost of inputs and compute returns.

For this study the latest version of EPIC (EPIC5300) was used. The version number refers to the year and the day of release. The first number codes the year -- 1995, while the subsequent three digits refer to the day -- 300th day of that year.

Model Operation

EPIC is developed specifically to estimate the long-term relationship between erosion and productivity. Erosion/productivity relationships (E/P) are estimated using two approaches; and both approaches involve plotting values of a term called erosion/productivity index (EPI) on the y axis against the corresponding values of erosion of the x axis (Williams et al. 1990). The model runs on a daily time step and uses the simplest and most efficient components available that will give adequate results. The model considers small areas (≈ 1 ha) due to the spatial homogeneity of soil and management effects. However, the model handles variability in the vertical direction of the soil profile; and divides the soil into a maximum of 10 layers whose thickness can be varied. The EPIC model is implemented as a set of 83 subprograms written in Microsoft FORTRAN. The EPIC main program reads data, initializes variables, and calls

subprograms to the daily simulation and to summarize and output data. The subprograms are aggregated into eight groups in alphabetical order. The group names are adjunct, crop, erosion, hydrology, nutrients, soil, tillage, and weather. A description of the operation of each subprogram is given by Williams et al. (1990).

Model Structure

As this study involves integration of EPIC with GIS, a brief description of the EPIC system file structure is warranted. EPIC system file structure (Figure 1) is comprised of text files which contain estimate of parameters of different physical processes modeled by EPIC. These files include (Dumesnil 1993): Basic User-Supplied Data File, Crop Parameter File, Tillage Parameter File, Pesticide Parameter File, Fertilizer Parameter File, Miscellaneous Parameter File, Graphics Control File, Multi-Run File, Output Variables File, and Daily Weather Data File.

Basic User-Supplied Data File

The Basic User-Supplied File, as the name suggests, is developed by the user. There are 8 categories of data that the user must specify: title; program control codes; general data; water erosion data; weather data; wind erosion data; soil data; and management information. There are about 617 values that must be estimated to develop a Basic User-Supplied Data File (Appendix I.a). The first segment of parameter values represent the program control codes, general data about the site to be modeled, and also parameters values that provide the water erosion data required by the erosion component. The next block of parameter values relate to weather data and wind erosion. Monthly weather data files are available within EPIC for over 137 sites across the United States.

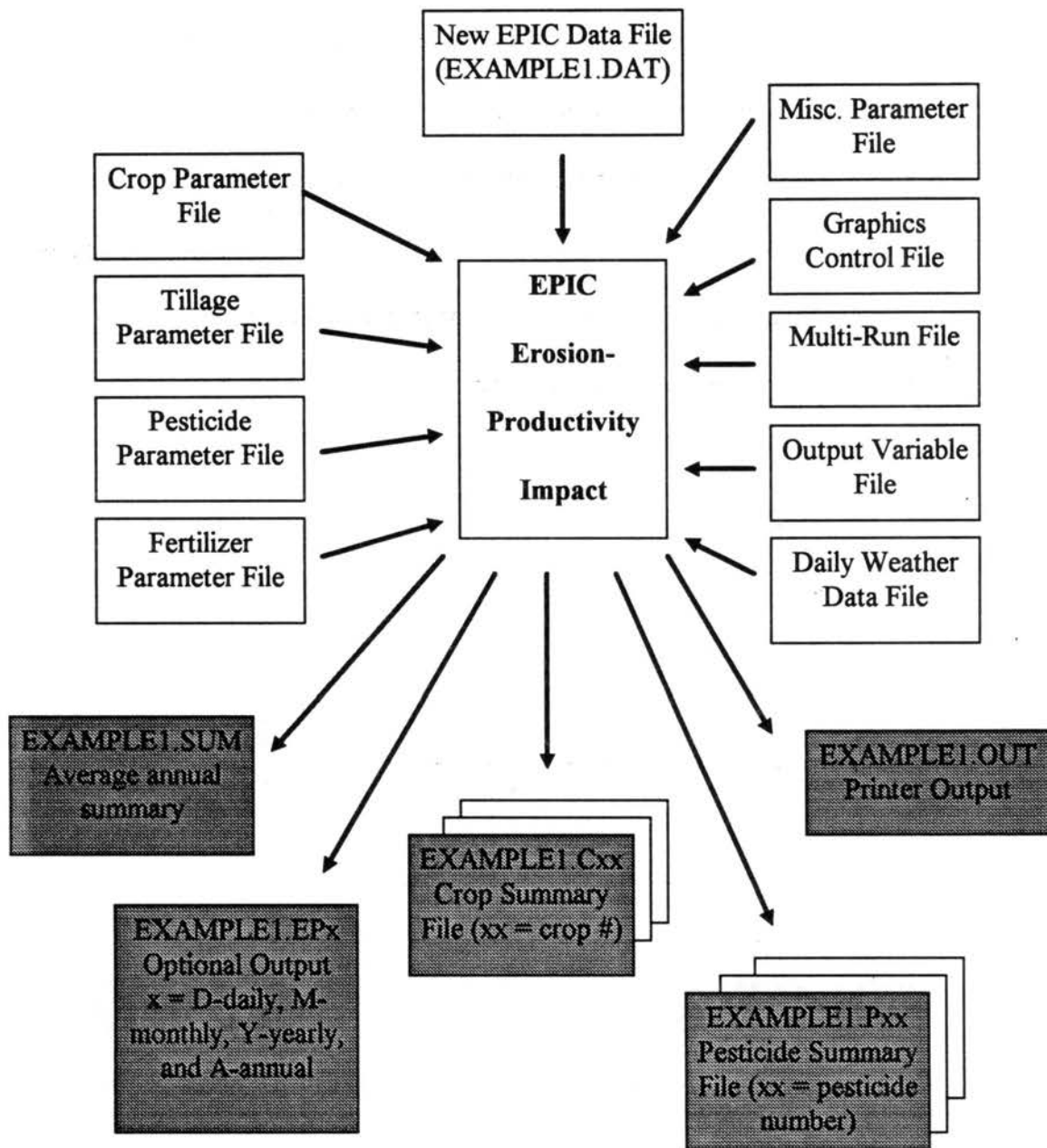


Figure 1. EPIC System File Structure (Dumesnil 1993).

These parameters supply the basic information required by EPIC to simulate daily weather and evaporation. The third segment of parameter values represent soil data. Soil data files containing estimates for various parameters are available within EPIC for 737 soil series found in the United States. The block following soil data represents management data. These parameters define the management information scenario associated with the production of the crop(s) of interest. Further, as an option, daily weather data for one or more of the six driving weather variables can be incorporated by inserting the name and location of the Daily Weather Data File at the end of the Basic User-Supplied Data File.

Crop Parameter File (USDACROP.DAT)

The EPIC crop model accesses the Crop Parameter File which consists of parameter values for a wide range of crops. The Crop Parameter File is the result of research carried out in the United States utilizing most major crops. The latest version of EPIC (EPIC5300) uses the crop data file produced by the U.S. Department of Agriculture. This data file provides crop parameter values for major crops including some horticultural crops. The feature supports about 82 crops for which crop parameter data are included with the EPIC software. There are about 47 parameters related to crop growth, its environment, and crop growth in a stressed environment. Also included are parameters for per unit cost of seeds and per unit price of yield which are useful for estimating seed cost and yield returns. Other parameters include parameters for estimating fraction of nutrient in crop biomass at emergence, mid-season, and maturity. The reader is referred to Dumesnil (1993) for a detailed description of the different crop parameter

variables. The model developers recommend that users refrain from altering crop parameter values unless there is substantial data from research which warrants the change.

Tillage Parameter File (CLASSTILL.DAT)

The Tillage Parameter File provides parameter estimates for different tillage practices. The parameters refer to tillage operations and include the type of implement used for tillage, per unit operation cost, mixing efficiency of the operation, tillage depth, height and interval for ridges and furrow dike, and harvest efficiency. There are about 51 tillage operations that are supported by EPIC.

The Pesticide Parameter File (PEST5300.DAT)

The Pesticide Parameter File contains chemical information about major insecticides and herbicides. The information refers to the pesticides solubility, foliage half life, wash off fraction, organic carbon partition coefficient, and cost of the pesticide. For simulations dealing with cost analysis, the cost parameter should be input by the user as it varies from place to place. There are 76 pesticides for which information is available in EPIC.

The Fertilizer Parameter File (FERT5300.DAT)

EPIC enables simulations involving the use of different types of manures and fertilizer products. For the different manures and fertilizers, the Fertilizer Parameter File provides estimates of mineral N fraction (Nitrate - N and Ammonium - N), organic nitrogen form, mineral phosphorus, organic phosphorus, and also fraction of mineral nitrogen in ammonium form.

Miscellaneous Parameter File (PARM5300.DAT)

The Miscellaneous Parameter File consists of experimental variables which undergo testing to develop estimates for alternate crop-soil-climate situations or to develop more precise relationships. Subsequently, after thorough testing these parameters may be added to future versions of EPIC. The developers of the model suggest extreme care in adjusting these parameters if the application is research oriented. These experimental parameters are specific to the version of EPIC; the miscellaneous parameters in the previous version of EPIC usually becomes part of the model for input. There are about 20 miscellaneous parameters in EPIC5300 which undergo research and tests.

The Graphics Control File (GRAF5300.DAT)

The Graphics Control File provides a means of controlling the graphical output of the model. The file consists of variables that define the number of graphs to display, the scale for the Y-axis, the variables that are to be plotted, and the types of graphs available.

The Multi-Run File (MLRN5300.DAT)

The Multi-Run File can be utilized to get a quick estimate of a long-term simulation. The water erosion control factor and the wind erosion control factors can be set to 1.0 which allows for the normal simulation of water and wind erosion, while a factor value of 10.0 accelerates erosion at the rate of 10 times the normal.

The Output Variable File (PRNT5300.DAT)

The parameters of the Output Variable File help define the model output. The user can customize the output by selecting the variables to be output on a daily, monthly, and

yearly basis. The variables are divided based on state, concentration, and accumulated values for the different time periods. In addition, a summary file is generated that outputs a summary of the selected variables values at the end of the simulation period.

The Daily Weather Data File

In addition to using the EPIC supplied weather generator to generate all daily weather data, the user can input daily weather data from an external file, or combine input and generated data. The Weather Input Code can be altered to allow EPIC to read the user supplied Daily Weather Data File. The options available are as follows: 1. generating all weather data; 2. reading only rain data from the user specified file; 3. reading Max-Min temperature (and rain) from the user specified file; 4. reading solar radiation (and rain) only; 5. reading wind speed (and rain only); 6. reading relative humidity (and rain) only; 7. reading rain, maximum and minimum temperature, and solar radiation; 8. reading rain, maximum and minimum temperature, solar radiation, and wind speed; and 9. to read all weather variables from the user specified file. The importance of rain is evident from the above options, because it is used to generate the other weather data values.

EPIC's Universal Text Integration Language (UTIL)

UTIL, a companion package to EPIC, is a data file editor that was originally developed to help users build datasets for large computer models and other data intensive programs (Dumesnil 1993). It is designed to edit any file with a fixed number of variables, cells or fields and is easy to use since it combines command-line and full-screen editing. It provides an on-line description of the variable, range limits for the variable, and the variable's usage. Furthermore, extra commands are available to load blocks of data from

database files. This greatly facilitates data entry. The different commands used in UTIL are designed to be entered interactively or to be stored in batch files to allow for executing groups of commands.

EPIC Basic User-Supplied Data File (hereafter, EPIC Basic Data File) has a specific format; each data variable has a particular position and takes a value from a bounded range. This range is based on the research data, and is optimized for a general run of the model. UTIL facilitates data entry by automatically placing each variable value in its assigned slot. The interactive mode of UTIL involves moving the cursor over the variables which simultaneously displays the variable name and a description of the variable. Also, command line data entry is possible, once the user becomes familiar with the variable names. The batch mode of data entry using UTIL is through the creation of a UTIL batch file. This batch file contains UTIL specific commands or the variable name and its assigned value separated by at least a single space. This is a powerful feature of UTIL as it places the user assigned value of each variable to its unique position in the EPIC Basic Data File. Thus, the possibility of preparing the data file with a wrong format is eliminated as the user is no longer concerned about the placement of the variables' values. Appendix I.b provides a sample of a UTIL file.

In addition to providing the functionality of editing the EPIC Basic Data File, UTIL provides for editing the different parameter files. The user can edit these parameter files by first loading their respective drivers. This is particularly helpful for users in customizing the Output Variable File and the Graphics Control File to generate output of variables desired by the user.

3. Integrated System - Design and Development

In this study, a system framework was designed and developed integrating ArcView 2.1, EPIC, and a graphical user interface (Figure 2). The integrated system was developed on a PC having Windows95 as the operating environment. The integration involved identifying the different files (EPIC parameter files, text files, and UTIL batch files) that are needed for assembling the EPIC Base Data File. A great amount of time was spent comprehending the EPIC file structure. Also, databases (management and spatial) were created to provide site-specific information. Subsequent to this, Avenue and Visual Basic programs were written to create files required for dataset assembly as well as for model execution. During program compilation, a significant amount of help was acquired from the Computer Science Department, Oklahoma State University. The following paragraphs describe in detail the design and development of the integrated system.

Integrated System Components

The system components pertaining to ArcView 2.1 and the graphical user interface are briefly described in the following paragraphs; the description of the EPIC component of the system is omitted in this section as it was described under the model description section of this chapter.

ArcView 2.1

ArcView is a powerful, easy-to-use tool that brings geographic information to the desktop. ArcView allows users to visualize, explore, query, and analyze data spatially. Data in ARC/INFO format are compatible with ArcView, and a user can easily use

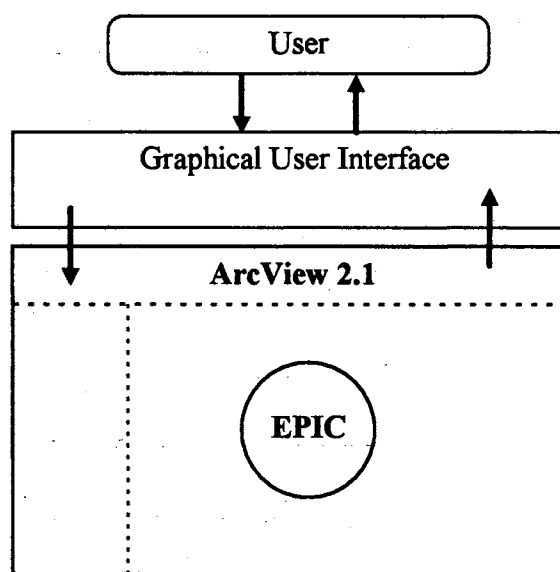


Figure 2. Schematic representation of the integrated system components.

ArcView to access all of this data, including vector coverages, map libraries, grids, images, and event data.

ArcView, a desktop GIS, represents the real world on a computer similar to the way maps represent the world on paper; however, ArcView, owing to its spatial and analytical capabilities, far surpasses paper maps. ArcView stores all the information about map features in a GIS database and links features on the map to information (attributes) about them. Thus, a user can access all information about a feature by simply clicking on it. For example, a spatial map feature such as soil might have a host of attributes attached to it such as texture, pH, bulk density, cation exchange capacity, slope class, fertility level, etc. The map feature and the attribute are dynamically linked, change in one is reflected in the other; it is this linkage that makes ArcView a truly dynamic system (Environmental Systems Research Institute 1995).

ArcView links sets of map features and their attributes and manages them together in units called themes. All the themes for a geographic area of interest comprise a GIS database. The user can add new themes, create more themes, delete old ones, or combine themes with common characteristics. These themes can be utilized to analyze multiple situations, get a better perspective, and solve problems. Thus, the design of a GIS database is strong because it is flexible (Environmental Systems Research Institute 1995).

With ArcView, a user works with geographic data utilizing different components - Views, Tables, Charts, Layouts, and Scripts. All these components are stored in one file called a project. ArcView's project window shows the contents of the project and makes it easy to manage.

In ArcView, geographic data in the form of interactive maps are housed in windows known as views. Every view includes a 'Table of Contents' compartment, which helps the user to identify and control the theme displayed. As already mentioned, a view is a collection of themes which represents a distinct set of map features in a particular geographic data source. A view defines the geographic data that will be used and how it will be displayed, but it does not contain the geographic data files themselves. Instead, a view references these source data files. Thus, a view is dynamic, because it reflects the current status of the data. If the source data changes, a view that uses this data will reflect the change the next time the view redraws. Also, the same data can be displayed on more than one view depending on the application.

The attributes of the map features are presented in ArcView as tables. The table formats the attributes in rows and columns. Each column stores a different attribute and each row relates to a single feature. If a user clicks on a map feature, its record or a row is selected in the table, thus identifying its attributes and vice versa. Additionally tables also have a full range of features for obtaining summary statistics, sorting, querying, and joining to other tables. The ArcView table references the tabular data source it represents, but does not contain the tabular data itself; hence tables are dynamic, because they reflect the current status of the source data that they are based upon. Some tables allow editing, depending on the data source; all edits are written back to the source data file.

Spatial data sources such as ARC/INFO coverages have attribute tables containing descriptive information about the geographic features they contain. A user can use a view containing themes that represent these spatial data sources and have access to their

attribute tables. ArcView manages the relationship between themes and their attribute tables. These tables do not need to be loaded into ArcView separately. Users can add dBase tables, INFO tables (specific to ARC/INFO), and tab or comma delimited text files into ArcView as tables. Also, from within ArcView, connection to a database server such as Oracle or Sybase is possible. Thus, a Structured Query language (SQL) query can be performed to retrieve records in the form of a table.

Charts offer a powerful business graphics and data visualization capability that is fully integrated into the geographic environment. ArcView allows a user to work simultaneously with geographic, tabular, and chart representations of data. Charts are created from tables to visualize trends, patterns, and distributions. Since charts reference tabular data in an existing ArcView table, they are dynamic and reflect the current status of the data in the table. So if the table is edited, the chart mirrors the change. Also, any change in the source data on which the table is based will automatically get reflected in both the table and the chart. A chart can represent all or a subset of records in a table. By querying and selecting records, the message provided by the data also changes.

A layout is a map that lets a user display views, charts, tables, imported graphics, and graphic primitives. The layout is used to prepare these graphics for output from ArcView. A layout defines the data to be used for output and how they will be displayed. A layout is dynamic because specific graphics reflect the current status of the data; if the data in a view changes, the layout also changes. Different layouts can be created based on the same data; each layout provides a different way of presenting the data. Customizing functions, user interfaces and cartographic templates can be made use of in creating the

output. Layout provides standard graphics and operations which can be drawn using the Draw tool and includes points, lines, polygons, polylines, rectangles, and circles. Layout also contains objects specific to the ArcView environment such as legends and scale bars.

A script is the component of an ArcView project that contains the ArcView programming language called Avenue. Avenue represents the customization and application development environment for ArcView. Scripts written in Avenue are geared toward achieving three general objectives: 1) the automation of tasks, 2) the additions new capabilities to ArcView, or 3) the development of complete applications. In fact, all of ArcView can be considered a collection of scripts. Every control that the user uses by clicking or dragging in ArcView, has an associated internal or system script that either gets activated or deactivated. The script editor is used for writing a script using Avenue code. Alternatively, other text editors can be used to create a text file containing the code; code can be loaded into the projects script later. Once the script is written, it can be compiled, debugged, and executed. After satisfactory performance a script, can be associated with a control or with an event, such as starting up or shutting down a project.

Graphical User Interface (GUI)

The main purpose of the GUI is to serve as the system-user link through which the user can communicate with the different components of the integrated system. It mainly involves communicating with the ArcView GIS database to input and retrieve data needed for EPIC simulations and displaying the output results from EPIC in ArcView. The GUI, henceforth, will be referred to as "EPIC-View." EPIC-View is a pull down menu embedded in the menu bar (Figure 3). The menu consists of different menu items

representing the operations that are performed by the integrated system. The detailed functions of EPIC-View are outlined in the following section.

EPIC-View: Requirements Specifications

The project EPIC-View consists of developing different tools available in a menu form that are used to run the EPIC model and display the model results. The different menus are: data input; modeling; and display.

(a) Data Input. The data input menu is an interactive tool with which the user can create input data files to run EPIC. Initially a common input data file is created having values of variables that remain constant for the whole field. This common data file is then replicated for all cells in the field. The data input tool consists of five sub-tools:

1. Weather Data. When this menu item is selected, the user is prompted for a weather file. The file used can be a user defined file or the default EPIC weather generator file. When the default option is selected, the user is prompted for latitude and longitude coordinates of the farm and the file from the weather station nearest to the farm is loaded.

2. Soil Data. When this menu item is selected, the user is prompted to select a soil type which is common for the whole field or extent selected. The user is also prompted for information, such as land use, cover treatment (or practice), and hydrologic conditions. Based on the information provided the run-off curve number is automatically determined.

3. Constant Data. When this tool is selected, the user is prompted to enter the basic EPIC data which remains constant for the entire field. This option is enabled only if these characteristics were not entered previously. As a result, a common input data file is

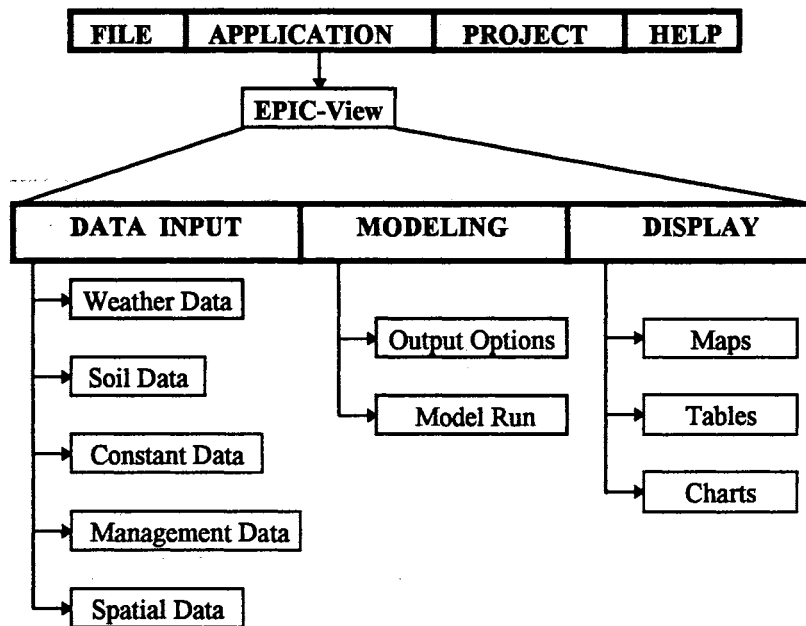


Figure 3. EPIC-View interface design framework.

created and is replicated for all cells of the field.

4. **Management Data.** On selecting this menu item, the user is prompted to draw a polygon or several polygons to specify management zones. There might be one or more management zones within the spatial extent. The user is prompted to enter the management practices, in chronological order, that were carried out on the management zones. The user can also specify one set of management practices (generic) for all management zones. The data are extracted so that they apply to all cells present within a management zone. The user can also save the set of management data for future use. EPIC supported management practices are presented as a scrolling window to facilitate selection and input.

5. **Spatial data.** When this menu item is selected, the spatial attributes (soils, slope, elevation, etc.) for the selected cells are extracted from the GIS database and stored in cell specific files. These files are loaded when EPIC is executed on individual cells.

(b) **Modeling.** The modeling menu of EPIC-View is also an interactive tool. When the tool is selected the user can specify output options and run the model. It consists of two menu items:

1. **Output options.** On selecting this menu item, the user is prompted to specify the variables for which output is desired. The user also can opt for EPIC output files in terms of every day, month, or year. The output file generated by the model will contain only those variable(s) selected by the user.

2. **Model Run.** On selecting this tool, the process executes a Visual Basic waitshell to run the EPIC model on each individual cell input data file. The corresponding output

files are parsed and retrieved as a table in ArcView. The table is linked with the existing GIS database to form new themes based on the output options selected.

(c) Display. The display menu is also an interactive tool which allows the user to select the type of model output display. When this tool is activated, the user can select Table, Chart or Map menu items to display the model run results as tables, charts or as maps respectively.

System Interfacing Tools

Interfacing is the means by which the system components communicate and interact. The interfacing processes were undertaken in ArcView with the development of Avenue scripts and Visual Basic programs to automate execution of functions outlined in the preceding section. The selection of ArcView and Visual Basic should not be misinterpreted as an endorsement of ESRI and Microsoft products. The selection was based solely upon the functionality and growing popularity within both the public and private sectors. A brief description of the programming environment is given in the following paragraphs.

ArcView Scripting Language - Avenue.

Avenue is the programming language and development environment that is part of ArcView. Avenue is fully integrated with ArcView. Avenue can be used to: (1) customize the working environment of ArcView, (2) direct ArcView to perform a specific task and, (3) develop a complete application that enhances ArcView's existing functionality.

ArcView provides the necessary customization and language environment for a user to

work with Avenue. A user can create the graphical user interface to establish initial properties for graphical controls a user will interact with, fine tune the behavior and appearance of the controls, and write Avenue code that responds to what goes on in the interface created. In addition, scripts written in Avenue can be linked to events such as starting up or shutting down a project.

Visual Basic 4.0

Visual Basic 4.0 allows a user to create applications for Windows operating systems. Visual Basic (VB) provides a user with appropriate tools for the different aspects of GUI development. A user can create a graphical user interface for different applications by drawing objects in a graphical way and setting properties on these objects to define their appearance and behavior. Program codes attached to these objects perform specific functions in response to events initiated by the user. Some of the features of Visual Basic include:

1. Data access features that allow a user to create databases and front-end applications for popular database formats.
2. OLE features that allow a user to use the functionality provided by other applications, such as the Microsoft Word for Windows word processor, the Microsoft Excel spreadsheet or the Microsoft Project business project planning system.
3. An executable file is created that uses a run-time dynamic-link library (DLL).

In addition to the above features, Visual Basic was found to provide more capabilities than Avenue in terms of creating graphical user objects. Importantly, it had

enhanced execution functionality in terms of delaying or waiting for a called program to finish. Thus, Visual Basic was used to build added functionality to the ArcView environment.

The EPIC-View menu and its different items are easily created in ArcView's customization window. The menu items are associated with different scripts which are executed when the item is selected. The attachment of the scripts to the menu items is easily done using the properties dialog box of ArcView's customization window.

The EPIC-View integration procedure will be explained with respect to the requirement specifications mentioned in the preceding section. However, certain terms and file characteristics require definition and description for a better understanding of the integration process.

Definitions:

Management Unit

A management unit is an arbitrary land area of a landscape. An example would be a field in an agricultural landscape. Farm management practices are considered uniform across a management unit. A management unit is considered uniform in its hydrologic response, and can also be termed a Hydrologic Response Unit (HRU). For this study, a management unit is the basic unit on which the model runs. Since, a raster data model was selected for this study, a management unit is synonymous to the cell of the raster coverage. When two or more management units have the same set of management practices, the management units are collectively referred to as management zone. In

ArcView, the user has the option of delineating the management zone by drawing a polygon in the view window using the selection tool. Alternatively, the management zone can be defined by querying the attribute table on certain criteria. Either way, the selected cells constitute the management zone.

Thus, the GIS data model defines the management unit as it is defined based on the raster cell resolution. On the other hand, the user's perspective defines the management zone as it is based on the management practices undertaken. Nevertheless, the management zone is potentially limited to the raster cell resolution, i.e., the management unit. For example, if a corn field consists of 50 cells or management units, and if the farmer manages 25 management units under no-till practices and the other 25 management units under conventional tillage, then there are two management zones in the corn field.

Curve Number (CN)

Hydrologic models use the curve number method to estimate runoff. To use this method, the user must assign a curve number that represents the innate hydrologic characteristics (hydrologic group) of the soil, the condition of its surface (good or poor) relative to infiltration, the type of crop, and the use of soil conservation practices that affect infiltration. The runoff curve number used in EPIC is the SCS antecedent moisture condition II (moist) curve number. Perhaps the most important factor in determining the curve number is the hydrologic soil-soil cover complex. EPIC provides a table to help estimate the CN.

There are four hydrologic soil groups: Group A, Group B, Group C, and Group D. Hydrologic group A has low runoff potential. The soils belonging to this group have high

infiltration rates, consisting chiefly of deep, well to excessively drained sands or gravel. Group B is characterized by soils having moderate infiltration rates when thoroughly wet. Group C is characterized by soils having slow infiltration rates when thoroughly wet, and consists chiefly of soils with a layer that impedes downward movement of water, or soils with moderately fine to fine texture. Group D soils have high runoff potential. Soils in this group are characterized by slow infiltration rates when thoroughly wet, and consist chiefly of clay soils with a high swelling potential, soils with a permanent high water table, soils with a claypan or clay layer at or near the surface, and also shallow soils over nearly impervious material. These soils have a very slow rate of water transmission. Hydrologic soil group in combination with soil cover are used to estimate the curve number. For example, a particular hydrologic soil group with row crop as cover will have a higher curve number than the same hydrologic soil group with pasture as soil cover. Thus, the higher the curve number, the lower the infiltration into the soil, and consequently the greater the runoff.

UTIL Files

UTIL files are special batch files specific to the UTIL program. These files are basically text files with an extension of ".UTL." UTIL files enable automatic loading of variable values into an EPIC Basic Data File, either from the DOS prompt or from the command line. Alternatively, UTIL files can be included in DOS batch files and sequentially executed. These files are invoked by UTIL by prefixing the character "@" to the UTIL filename. Furthermore, an UTIL file can be embedded in another UTIL file and be sequentially loaded.

UTIL files have a predefined format; each line of the file represents an EPIC specified variable name and its user assigned value separated by at least a single space. The variables in the file need not be in the same order as they occur in the EPIC Basic Data File. In order to speed up the creation of the data file it is better to follow the sequence specified by EPIC. Nevertheless, UTIL recognizes the variable name as EPIC defined and assigns its user assigned value into its specified slot in the EPIC Basic Data File.

FORM#.DAT

FORM#.DAT is the EPIC Basic Data File that is needed for each management unit in order to run the model. The '#' part of the filename represents the management unit ID number. The different components of this file such as weather, soil, management data, spatial data, and other model options are loaded through different UTIL files by invoking UTIL.

SHELL Function

Because ArcView does not wait for the completion of the batch files, the shell functionality of Visual Basic was utilized to temporarily enable ArcView to wait for the process to finish. The Shell function basically starts another program synchronously and returns control to the visual basic application. The shelled program continues to run independent of the application until the user closes it. VB application can be made to wait until the shelled program finishes by using the window API (GetModuleUsage()) to poll the status of the application. When the shelled program is finished, GetModuleUsage

returns a zero as the value for the instance handle of the shell. Thus, by using a conditional loop to check for the returned value the process can be delayed and control manipulated.

FORM.PRS

This file is a comma delimited text file that is created by the parser program. It holds the output summary file of an EPIC model run. This file is formed by parsing the output summary file (FORM#.SUM) of the model run. The first line of the file represents the output variables names specified by the user. Subsequent to the first line, the output values for the different variable are listed. The first column identifies the management units on which the model was run. This column is the key field which will be used later on to join this file to the attributes table in the GIS database for subsequent display in thematic maps.

Thus, the interfacing of the system components generally involves the creation of the above mentioned data files and UTIL files and subsequent execution of batch files during EPIC runs. The text files hold the different user-specified data as well as the data extracted from the spatial database, while the batch files hold the different commands required to invoke UTIL and EPIC. Through sequential execution of the batch file, the datasets are built and fed to EPIC for simulation runs, following which the model output is parsed into a comma delimited format recognizable by ArcView for subsequent display as thematic maps. The following paragraphs will describe in greater detail the procedures involved in the interfacing process.

System Interface Development

System interface development is explained in terms of its constituent EPIC-View components; namely, Data Input Interface, Modeling Interface, and Display Interface.

Data Input Interface

The Data Input Interface, as the name suggests, helps the user input data needed by the model. The interface is in the form of a menu comprising different items; namely, weather data, soil data, management data, spatial data, and constant data.

Weather Data

The weather data menu item facilitates input of weather data required by EPIC. Figure 4 illustrates the steps involved in this routine. On selecting the weather data tool of the data input menu, an Avenue program is executed which in turn initiates a VB executable program. The program codes are given in Appendix II.a. Basically, the executable program submits a form prompting the user to input the latitude and longitude of the area modeled. The latitude and longitude are used to locate the name of the nearest weather station to access EPIC supplied weather data. The latitude and longitude are written as a UTIL command in a UTIL file (CONST.UTL). Also, the latitude entered is stored separately as a variable required by EPIC.

Soil Data

The soil data menu item enables the user to input soil data. Avenue and Visual Basic program codes are given in Appendix II.b. The different steps comprising this routine are outlined in Figure 5. On selecting the menu item, the Avenue script executes a

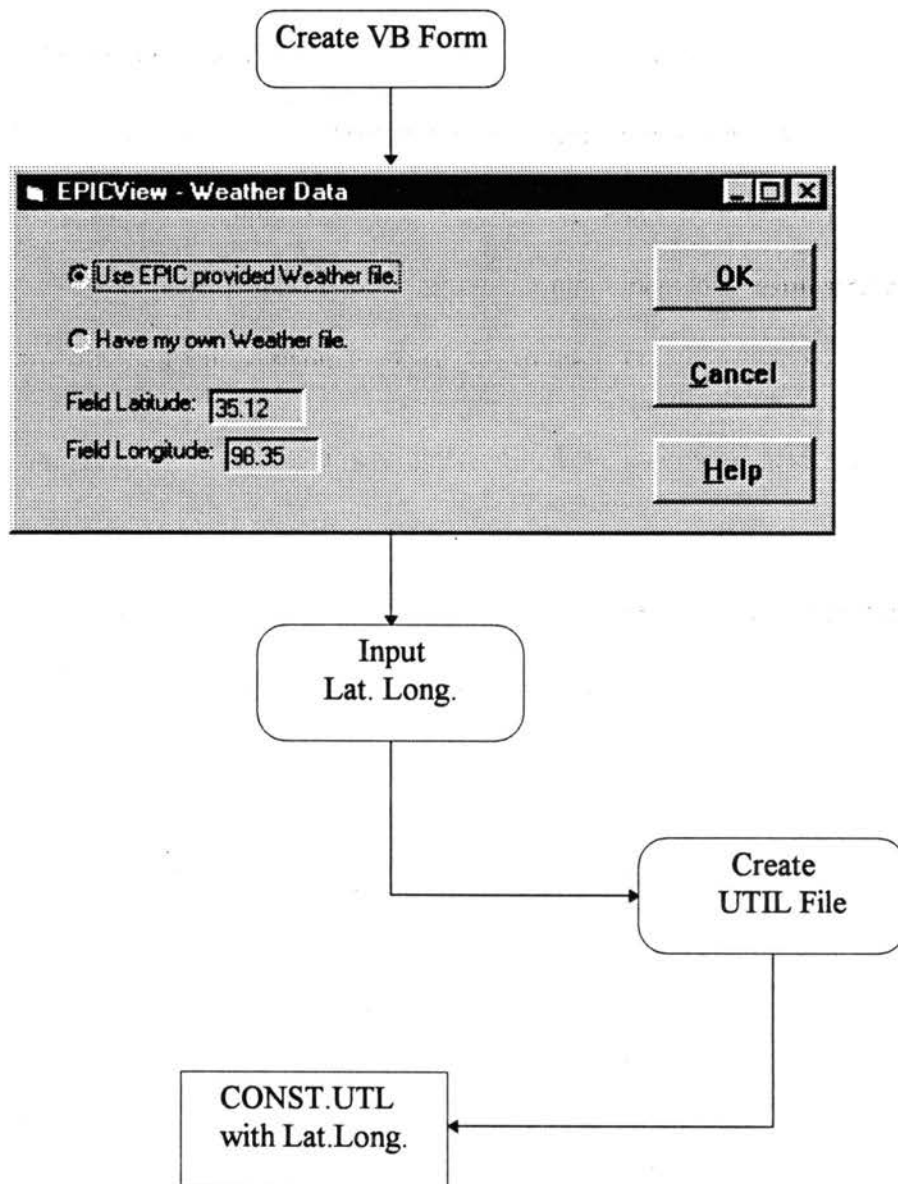


Figure 4. Steps involved in storing latitude and longitude coordinates for accessing weather data file.

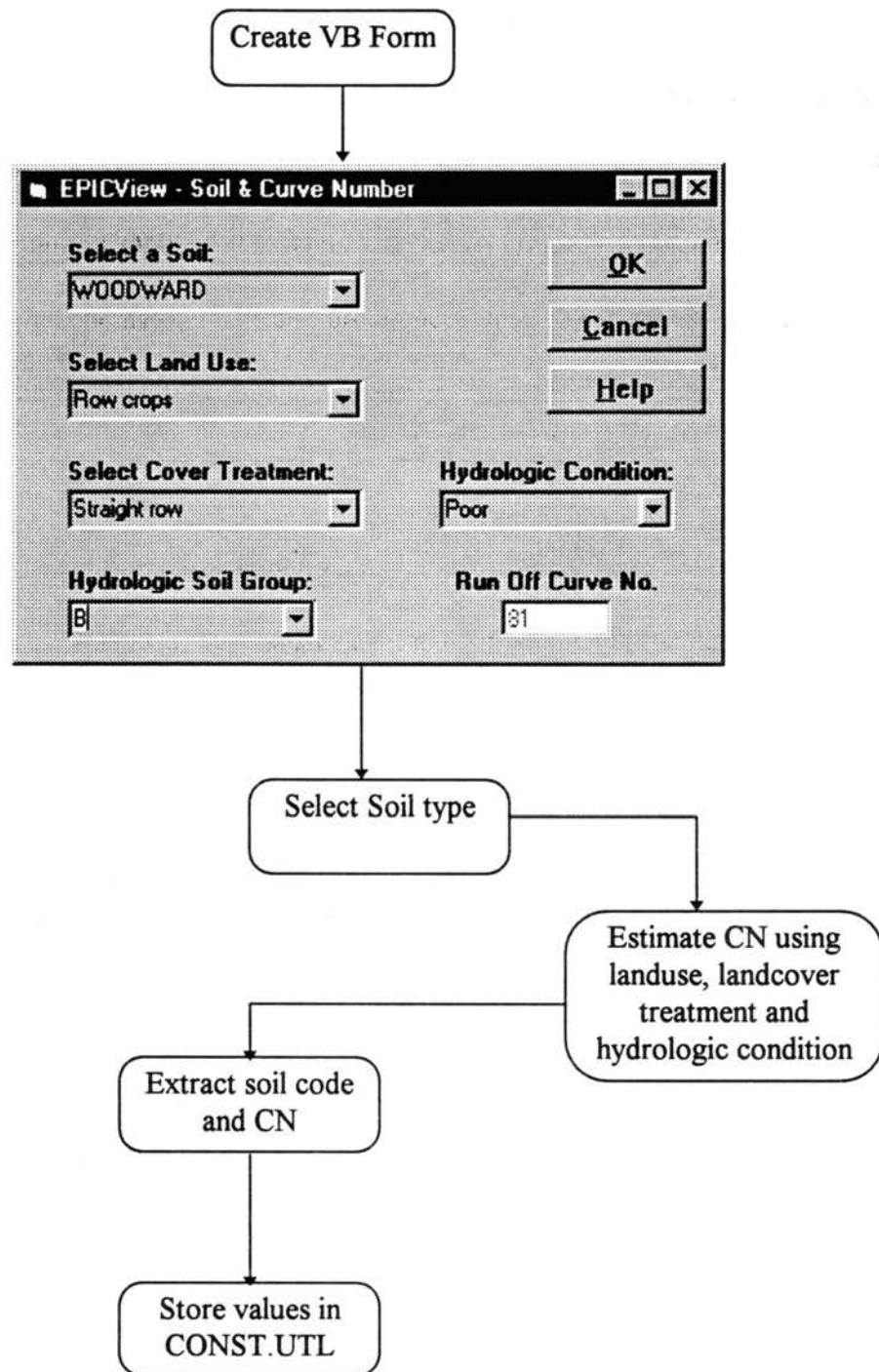


Figure 5. Steps involved in storing soil information, and soil code for accessing EPIC soil data file.

VB program. The VB program presents an input screen prompting the user to select the soil type. The code of the selected soil type is written as a UTIL command to a UTIL file (CONST.UTL). The other options in the form allow the user to estimate the SCS curve number based on the land use, landuse treatment, hydrologic condition and soil hydrologic group (U.S. Department of Agriculture 1972). The curve number is automatically calculated using a four-dimensional array to hold the curve numbers. The estimated curve number is written with the EPIC variable name (CN2) to the UTIL file (CONST.UTL).

The menu item is made inactive if it is detected that the user has a field defined in the GIS database to hold the soil series name. This is achieved by the update property of the script after checking for the soil field in the startup script. The soil series information, under such a situation, is extracted during the spatial data extraction process.

Constant Data

Some of the data required by EPIC is constant for a specific area and lacks any spatial characteristics. These data include values for variables such as EPIC program control codes specifying the equations used for estimating water and wind erosion, potential evapotranspiration, etc. EPIC suggested values can be used for most of these variables. The Avenue program and VB program codes are provided in Appendix II.c. The steps involved in this procedure are illustrated in Figure 6.

Basically, the program counts and stores the total number of cells in the study area for which the EPIC Base Data Files are required. The VB program submits data entry screens to facilitate the input of constant data. The user has an option to change the

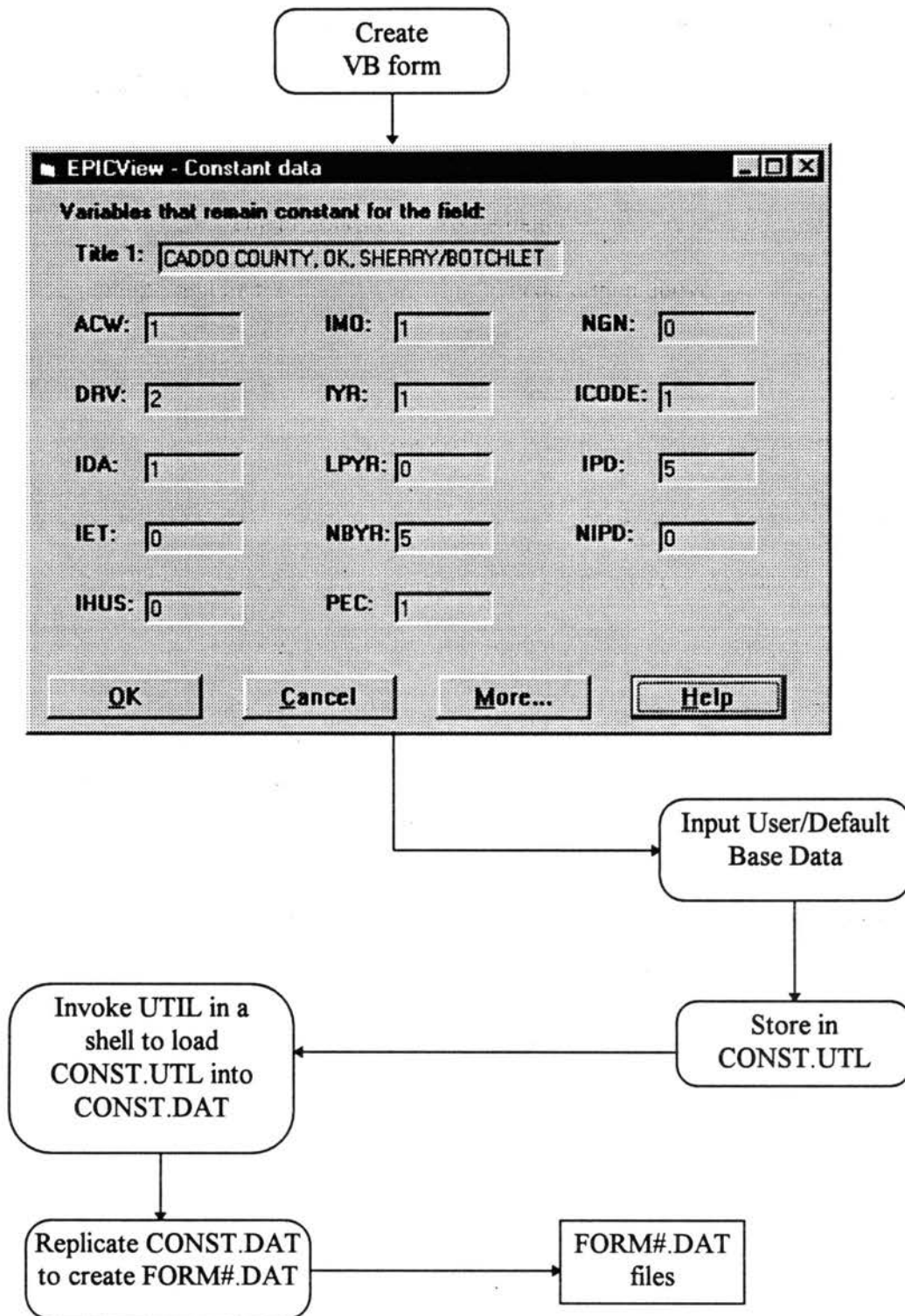


Figure 6. Processes involved in creating base data files (FORM#.DAT)

default values of the variables, for example the year of start of simulation or the number of years to simulate. The values are stored along with the variable name to a UTIL file (CONST.UTL). The UTIL file is loaded into CONST.DAT by invoking UTIL in a wait shell. It may be recalled that weather, latitude, and soil information (if user opts for EPIC soil) are also written to CONST.UTL. Thus, CONST.UTL loads up weather, latitude, and possibly soil data depending on the user's preference. Once CONST.DAT is created, it is replicated depending on the total number of cells.

Management Data

The management practices undertaken on a management unit are used by EPIC in processing the base data. Using the management data menu item of the data input menu, the different management practices performed on a management zone are entered and stored in a UTIL file (MGMT#.UTL). Figure 7 depicts the different steps involved in this process. Appendix II.d gives the Avenue program and the Visual Basic program codes for this routine.

Upon selecting the management data menu item, the user delineates the management zone using the select tool of ArcView, or alternatively selects management units using the query tool of ArcView. Once the management units of a particular zone are selected the user inputs the management operations particulars such as month of operation, day of operation, type of operation, etc. in chronological order. The user is prompted by different VB screens depending on the type of operation. For example, if the operation performed is fertilizer application a fertilizer screen pops up prompting the user

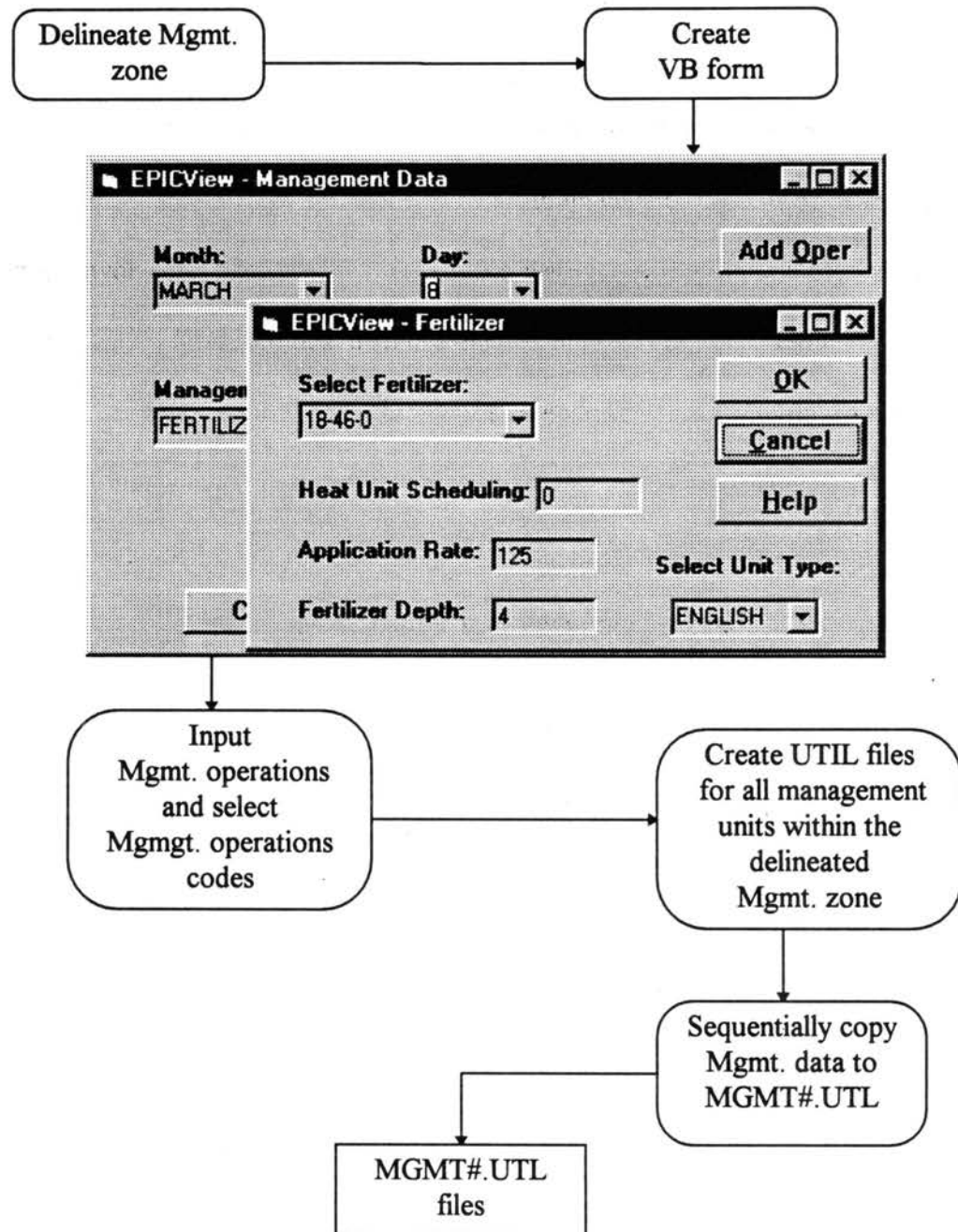


Figure 7. Steps involved in creating management data UTIL files

to enter the fertilizer application details such as application rate and depth of application. Based on the crop's maturity the user can also specify heat unit scheduling for the different operations. Application rates can be specified in metric or English units. Additionally, in case of a crop rotation, specific management practices can be entered for the next crop. New management zones can also be delineated for which management practices are entered in a method similar to what was described above.

Spatial Data

The spatial data in the GIS database are extracted by selecting the spatial data menu item. Each management unit has a unique set of spatial data such as elevation, slope, soil etc. Further details of the spatial database is given in the succeeding chapter. The spatial data needed for the EPIC simulation are extracted based on the selected cells of the management zone delineated. The Avenue program code is given in Appendix II.e. Basically, the program steps through the different processes outlined in Figure 8. The procedure seeks the spatial variables in the attribute table, extracts the values, and writes the values along with the variable name to a UTIL file (FORM#.UTL). The path name of UTIL file with the management data (MGMT#.UTL) is appended to the end of FORM#.UTL so that both the management data and the spatial data can be loaded together.

Modeling Interface

The Modeling Interface consists of the modeling menu which is comprised of two menu items: 1) output options and 2) model run. The modeling interface of EPIC-View

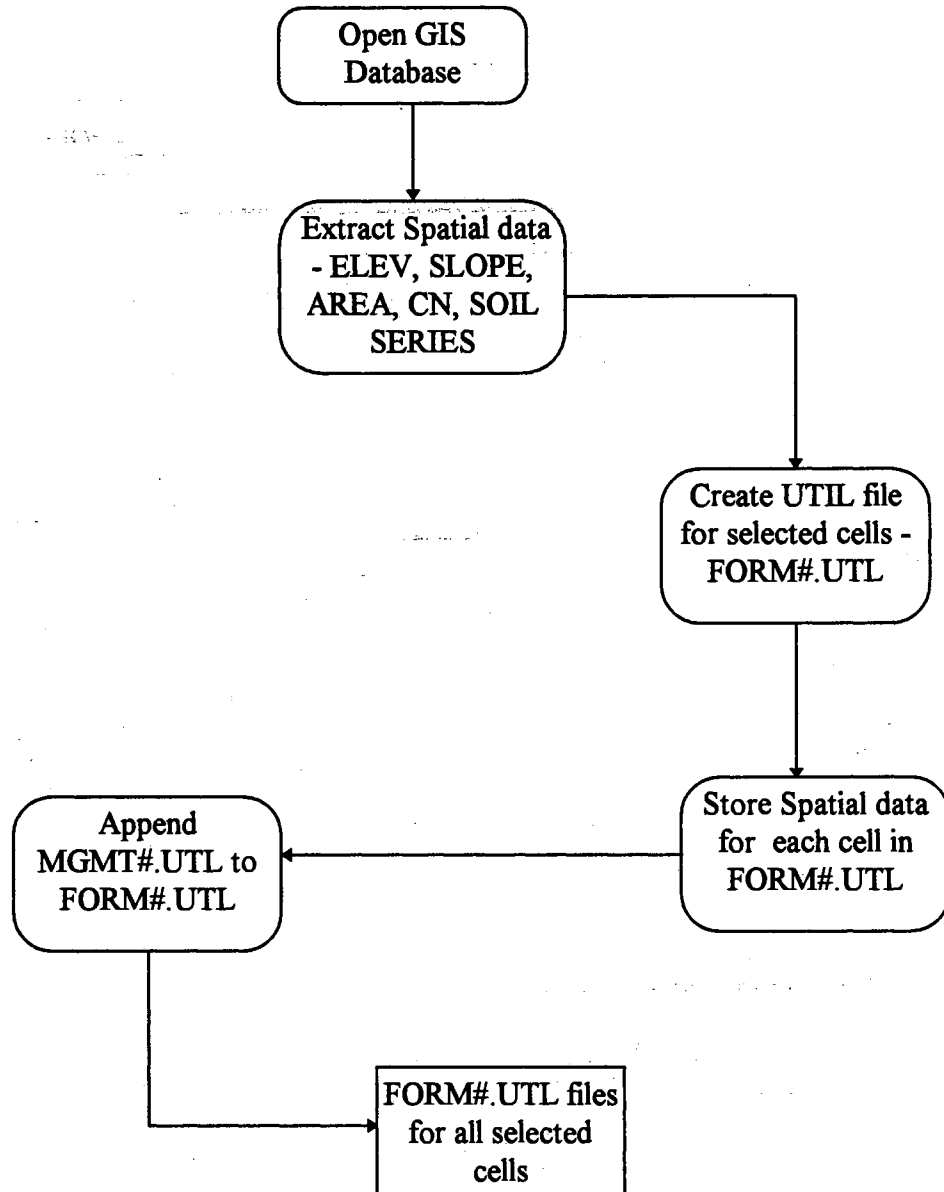


Figure 8. Processes involved in creating spatial data UTIL file (FORM#.UTL).

helps specify the model output in terms of the different variables desired and also executes the model.

Output Options

The output options menu item enables the user to specify the variables of interest for which output is generated by the model. The program code is listed in Appendix II.f and the steps involved are depicted in Figure 9. Basically, the program provides an option to create either new output settings or to use the existing setting. A maximum of 30 output variables can be selected. The option codes are written along with the variable name to a UTIL file which is automatically loaded into the EPIC print file (PRNT5300.DAT). PRNT5300.DAT is an EPIC generated file which lists the codes of various output variables. Thus, the output file is customized to user specified variables.

Model Run

The essence of interfacing the components of EPIC-View is accomplished with the model run menu item. This option integrates the UTIL files created by the other menu items such as management, spatial, etc. The integration involves invoking UTIL to load specific UTIL files into the EPIC data input file (FORM#.DAT). Integration is also done by calling one UTIL file from within another UTIL file, thereby aggregating UTIL files. After integration is completed, EPIC is executed using the integrated data files as input. Figure 10 elaborates the integration process of the different system components. Appendix II.g provides the Avenue program involved in executing the model on assembled data files.

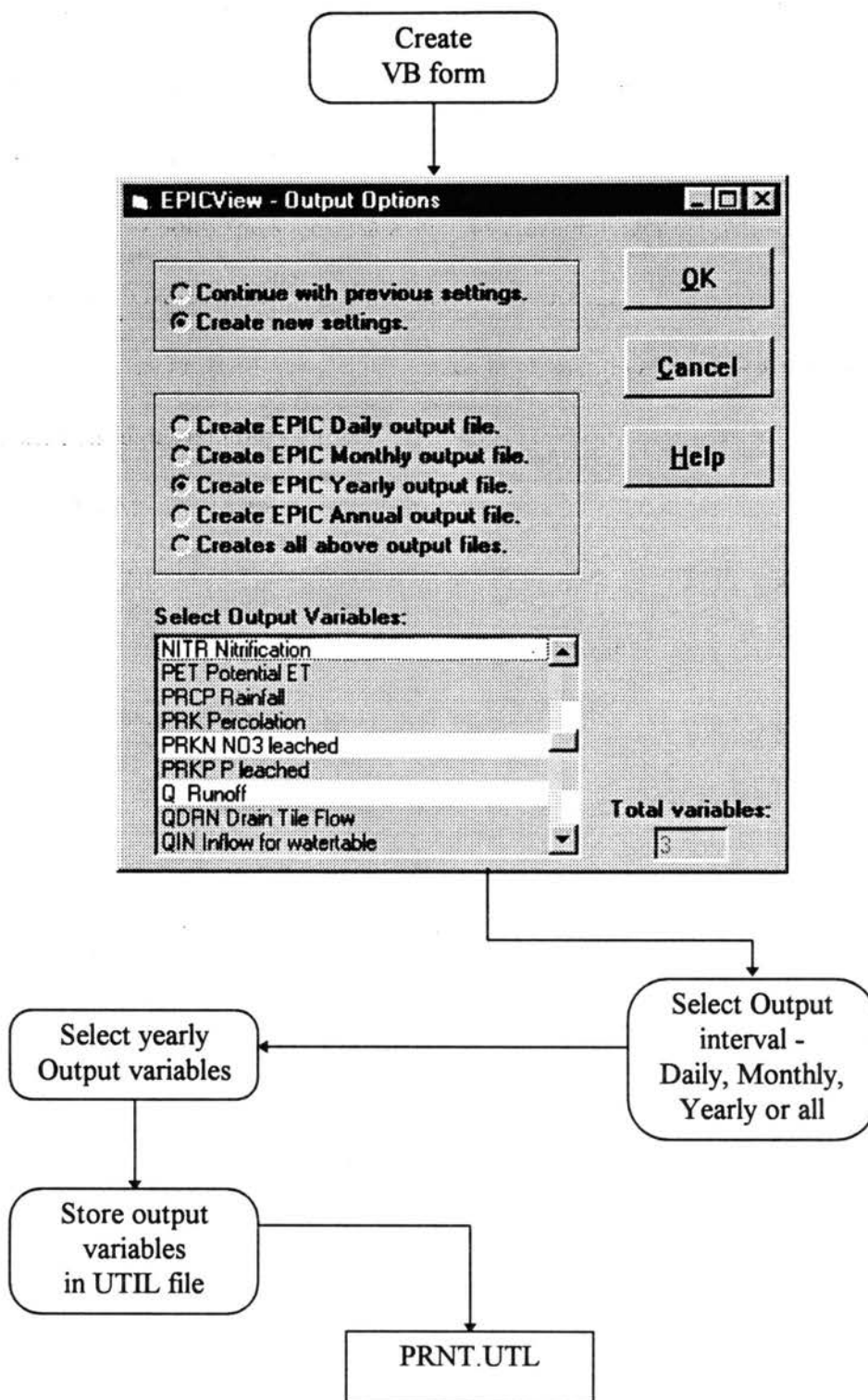


Figure 9. Program flow outline for specifying EPIC output options.

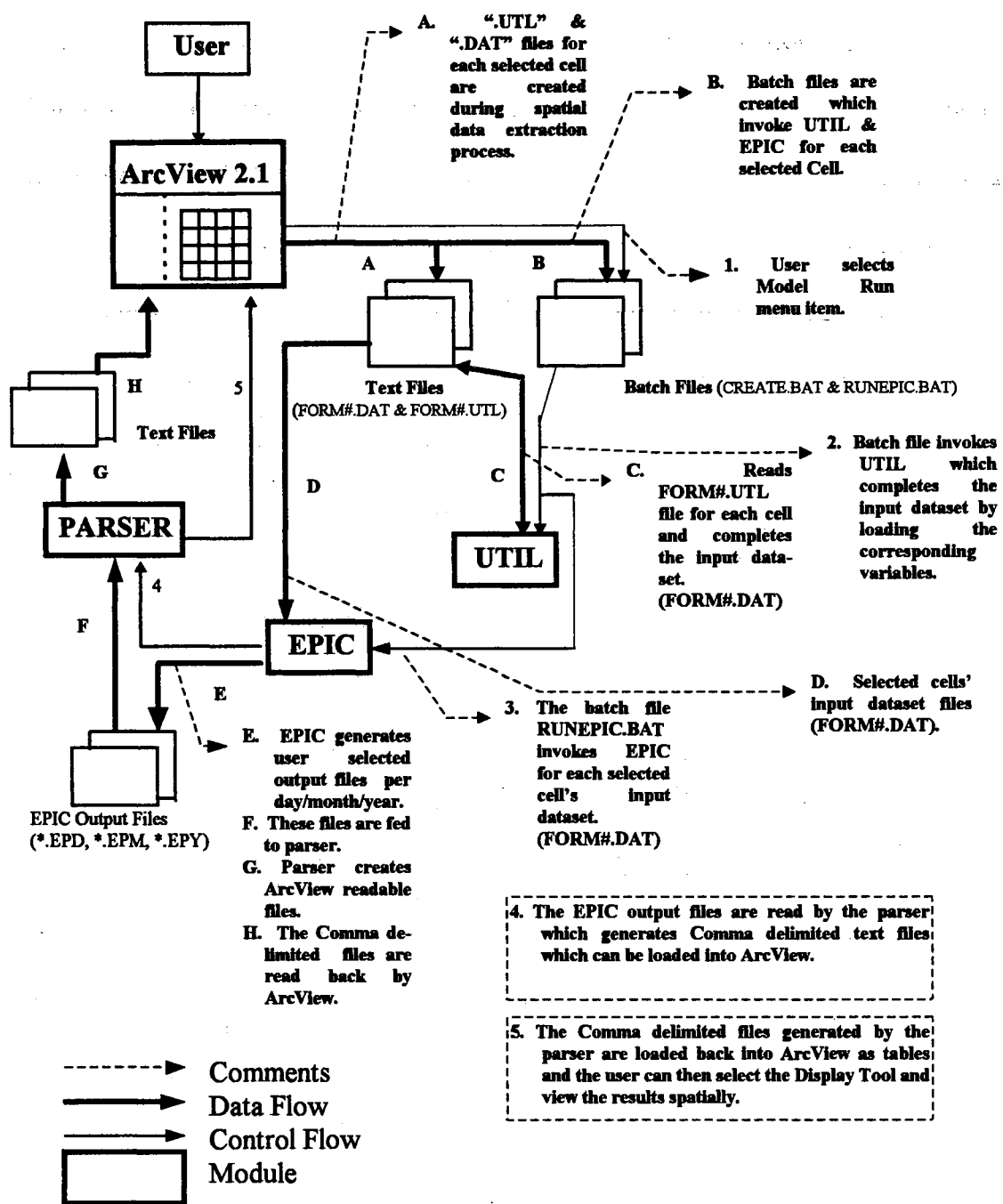


Figure 10. Program flow during a model run.

Basically, the program involves creating and executing batch files which hold UTIL and EPIC commands. The main files created are CREATE.BAT and RUNEPIC.BAT. CREATE.BAT invokes UTIL to load the file FORM#.UTL. FORM#.UTL also holds the spatial data such as elevation, slope, area, soil, curve number, etc. Moreover, FORM#.UTL calls the management practices file (MGMT#.UTL). Thus, execution of CREATE.BAT sequentially loads in to each FORM#.DAT the spatial as well as management data required by EPIC. Following the loading of FORM#.UTL into the respective FORM#.DAT, the output options file (PRNT.UTL) is loaded into the EPIC print file (PRNT5300.DAT) to specify model output. Once the UTIL files are loaded into the respective FORM#.DAT, data file assembly is complete. The last line of CREATE.BAT calls RUNEPIC.BAT.

The batch file RUNEPIC.BAT consists of command statements to invoke EPIC to run the completed FORM#.DAT files. The EPIC simulation runs generate summary files for selected output variables. The last command line of RUNEPIC.BAT invokes the parser program (Appendix II.h) to execute on the output summary files. Thus, the execution of different files in chain like fashion occurs with the execution of CREATE.BAT. The batch file is executed in a wait shell, to make ArcView wait during program execution.

Display Interface

The Display Interface consists of the menu items for displaying model results. Model results can be viewed as thematic maps, tables, or charts using the Maps, Tables, and Charts tool respectively.

Maps

The Maps tool helps in visualizing results of simulations as thematic maps. On selecting this tool, the parsed file (FORM.PRS) is pulled into the ArcView project as a table. This table is joined to the existing GIS database table (Polygon Attribute Table, PAT) based on the common field, i.e., the HRU ID number. Once the table is joined, any field in the merged table can be displayed as a thematic map. The fields existing in the imported table (FORM.PRS) represent different model output variables that need to be displayed as maps. This display is achieved by copying the original theme and pasting it a number of times in the view, depending on the number of fields present in FORM.PRS. The field names of the themes are sequentially changed to display different model output variables. Each field's values are classified into 5 levels based on the quantile method of classification. The classes are ramped on a color scheme to bring out the contrast in the display. Figure 11 depicts the steps involved in deriving thematic maps of model results. Avenue code is given in Appendix II.i.

Additional display options available through the display tool involve Charts and Tables tools. On selecting these tools, results of simulations can be viewed as simple charts or tables that ArcView supports. Avenue code that enables this is presented in Appendix II.i.

Miscellaneous program code used in the project is listed in Appendix II.j. Codes was written to accomplish tasks such as initializing global variables and updating other scripts.

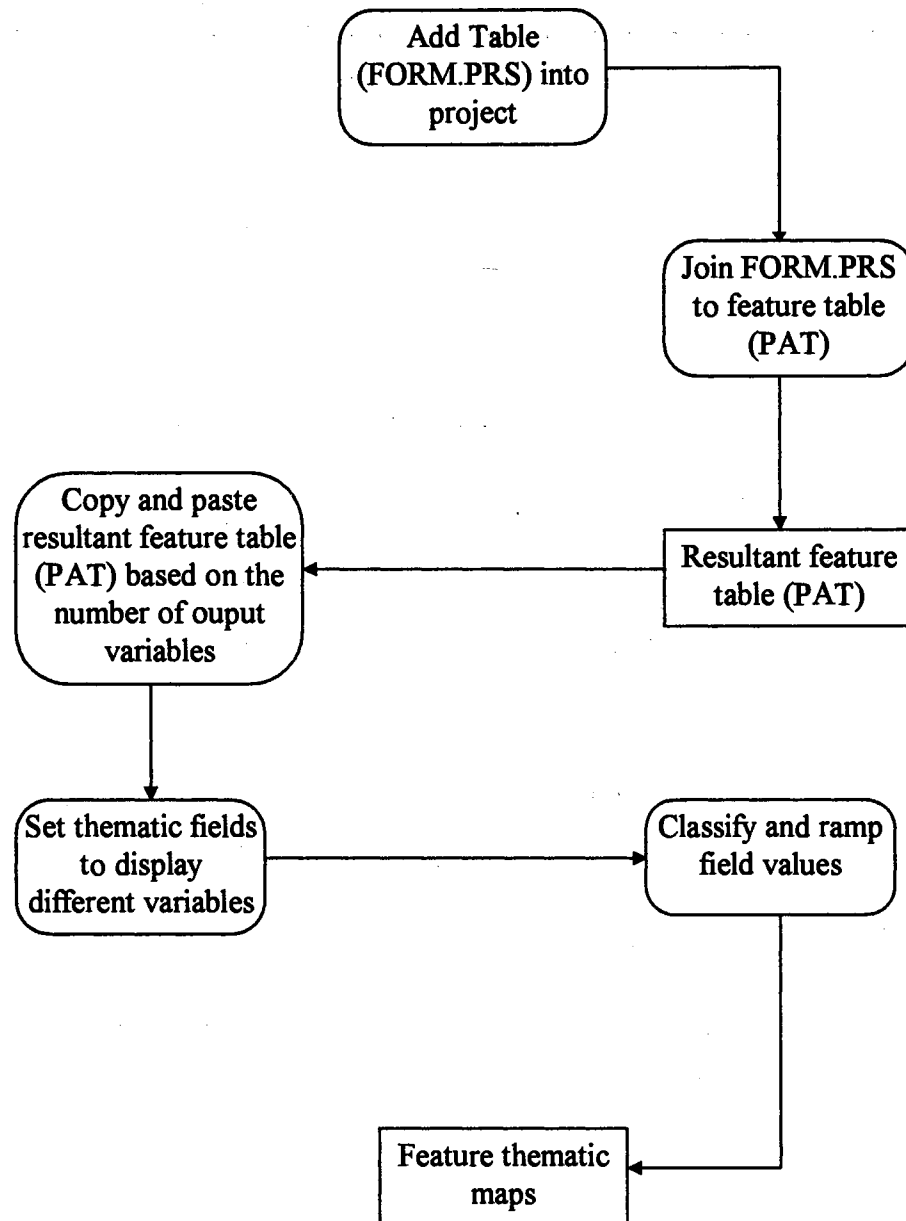


Figure 11. Steps involved in deriving thematic maps of the model results.

CHAPTER IV

APPLICATION OF THE INTEGRATED SYSTEM TO NITROGEN MODELING

Introduction

Agricultural systems typically evolve over long periods of time in response to climate, soils, agricultural technology, socioeconomic conditions and other factors. Comprehensive models capable of simulating crop growth, development, and yield became widely available on microcomputers during the 1980s. EPIC is designed to help decision makers analyze alternative cropping systems and determine if they are socioeconomically and environmentally sustainable. Although originally designed to assess the impacts of soil erosion on crop productivity, EPIC has been vastly improved to simulate a variety of management alternatives. These improvements give EPIC the capability to simulate the effects of weather, soil properties, crop rotations, and management practices on complex cropping systems. The holistic approach taken by EPIC permits users to assess the sustainability of cropping systems from the standpoint of water quality and socioeconomic considerations.

First this chapter describes several applications of the model, including evaluation of crop productivity, crop failure, soil resource degradation, water quality impacts, different input levels and management practices, response to spatial variation in climate

and soils, and sensitivity to long-term changes in climate. Second, the integrated modeling system developed in the preceding chapter will be used for farm level nitrogen modeling. Transformation of nitrogen will be forecasted and spatially displayed using the integrated modeling system.

EPIC Applications : An Overview

Crop Productivity

Accurate simulation of crop yield is necessary for most applications of models like EPIC. Model developers continually monitor the effects of model improvements on the simulation of yields. Williams et al. (1989) evaluated EPIC's ability to simulate yields of maize, wheat, rice, sunflower, barley, and soybeans using a total of 227 measured yields reported by independent research groups throughout the world. For these crops, mean simulated yields were always within 7% of mean measured yields. For 118 comparisons of measured and simulated maize yields mean measured yield and their standard deviation were 6.5 Mg ha^{-1} and 3.1 Mg ha^{-1} , respectively. The means were not significantly different at the 95% confidence level. Williams et al. (1989) also demonstrated that EPIC can accurately simulate maize responses to irrigation at locations in the western USA and to fertilizer nitrogen in Hawaii.

Soil Degradation

As part of the Resources Conservation Act (RCA) analysis, EPIC simulation runs of 100 years were made of over 13,000 combinations of crops, soils, climates, tillage, and soil conservation practices. Simulation results were used by a large linear programming

model to assess the impact of soil conservation practices and erosion on agricultural production of the USA (Putman and Dyke 1987). Benson et al. (1989) used EPIC to demonstrate that, even though the effects of soil erosion on crop productivity may be small for a long period, high rates of erosion can drastically shorten the productive life of soil.

Response to Climate and Soil

EPIC has been widely used by agricultural economists and others to simulate the effects of weather, climate, and crop management practices on crop productivity. Lacewell et al. (1990) used EPIC to evaluate crop yield, erosion and net returns for 12 alternative dryland crop rotations in the Texas Southern High Plains, with and without farmer participation in federal farm programs. They concluded that, with participation, cotton is associated with high rates of soil erosion and thus, requires rotation with wheat to reduce erosion and to comply with the farm program. Continuous cotton planted after a winter wheat cover crop terminated with herbicide late in winter appears to be a viable cropping system.

Arnold and Jones (1987) evaluated EPIC's sensitivity to soil, climate, and rotation effects on crop (maize, soybean, wheat, barley, peanut, and hay) productivity and fertilizer nitrogen requirements. It was concluded that EPIC can be used to evaluate previously untested combinations of soil, climate, and crop management, thereby reducing the amount of site-specific research needed to assess agricultural technology.

Climate Change

In addition to regional and farm-scale economic analysis, EPIC has been used to assess the effects of short- and long-term climatic changes. The U.S. Department of Agriculture used EPIC during the summer of 1988 to predict the effects of that year's severe drought on US crop production (Benson 1989).

Robertson et al. (1987, 1990) used EPIC to predict the impacts of CO₂ and climate change scenarios on crop yields, soil erosion, and farm management for the Great Plains, the Corn Belt, and the southeastern United States. Recent model improvements permit more accurate simulation and the effects of CO₂ and climatic change on hydrology and crop growth. A study in progress estimates the effects of change in atmospheric CO₂ and a future climate similar to that of the 1930s on crop yields and farm profitability in the midwestern United States.

Input Levels and Management Practices

Cabelguenne et al. (1990) used EPIC to simulate growth and yield of corn, grain sorghum, sunflower, soybean, and wheat grown in rotations over a 5-year period at three levels each of fertilizer, irrigation, and tillage in southeastern France. The root mean square of simulated grain yields ranged from 0.4 Mg ha⁻¹ for sunflower to 1.6 Mg ha⁻¹ for corn. Mean simulated yields were not significantly different from mean measured yields for summer crops, and for individual plots, simulated and measured yields were within 20% of each other for 81% of the comparisons.

Dyke et al. (1990) compared simulated and measured yields for a total of 204 treatment years for the Southern Coastal Plains and Southern High Plains of Texas. Crops

included maize, grain sorghum, and cotton. Tillage systems, irrigation, and crop rotations also varied. Simulated yields were within 20% of mean measured yields for 70% and 90% of treatment-years for the Coastal Plains and High Plains, respectively. Simulated yields were within the 95% confidence interval of measured yields for 69% and 88% of the treatment-years for the two sites. Wegener (1989) used AUSCANE, a version of EPIC adapted to Australian sugarcane (Jones et al. 1989), to demonstrate the importance of irrigation in reducing the risk in sugarcane production near Mackay, in Queensland. Segarra (1989) used EPIC to evaluate optimum nitrogen fertilizer rates for cotton in the Southern Plains of Texas. Carryover nitrogen and cotton prices were found to affect optimum fertilizer rates, and use of decision rules for nitrogen fertilization could improve the cash flow of producers.

As EPIC can simulate a variety of important agricultural practices, the model has also been used successfully to estimate crop fertilizer requirements (Putman and Dyke 1986), nutrient transport in runoff (Smith et al. 1986), soil and fertilizer phosphorus dynamics (Jones et al. 1984), the effects of furrow diking on crop yields (Krishna et al. 1987), and low-input legume based crop rotations (Arnold and Jones 1987, Benson et al. 1990).

Recent addition of salinity and pesticide components will permit the model to simulate movement of pesticides, nutrients, and salts toward ground and surface waters, both in solution, and as appropriate, attached to sediments (Sabbagh et al. 1991).

Nitrogen Modeling

The GIS-based integrated modeling system developed in the preceding chapter will be used for farm level nitrogen modeling. However, in order to better understand the different aspects of the nitrogen modeling study, a brief description is warranted concerning the role of nitrogen and its various transformations in the soil-crop-atmosphere continuum.

Nitrogen Cycling

Nitrogen (N) is one of the most important primary nutrients required by plants for normal growth and development. The efficiency with which nitrogen is utilized in crop production is a subject of intense interest to agriculturists. However, emphasis has been on the fertilizer N efficiency in crop production with little or no emphasis on unutilized N in the soil (Gilliam et al. 1985). It was not until the late 1960s when Commoner (1970) provided awareness of the effect of N fertilization on N movement into waters. Since that time, extensive work has been undertaken to study the dynamics of N in agricultural croplands with the main intention being nutrient management.

Plant uptake is the primary objective of fertilizer application; however, the efficiency of fertilizer N uptake is seldom much above 50%, even with good management (Gilliam et al. 1985). When fertilizer N is added to soil, the N may be harvested with the crop, incorporated into soil organic matter, denitrified, volatilized, or lost to drainage waters. The factors controlling transformations are complex and vary greatly with soils and geographical locations. The different processes that occur in a general nitrogen cycle

are briefly explained to understand the various sources and sinks for different forms of nitrogen.

Figure 12 shows the sources of nitrogen that gain entrance into the soil, the transformations that occur, and the fate of the end products. The principal sources of crop nitrogen are soil organic matter of plant and animal origin, precipitation, adsorption of atmospheric N, biological N fixation, and commercial fertilizers. In many soils, soil organic matter is often a primary source of N and other nutrients. Most of this N is immobilized as proteins and other nitrogenous compounds, often as components of dead microbial cells or absorbed on clay surfaces. However, soil organic matter gets slowly converted by microbial degradation and made available to growing plants through the process called mineralization (Paul and Juma 1981). The rate and efficiency of mineralization depends many factors -- soil temperature, water regimes, C/N ratio of the crop residues, residue placement, size, and other factors (Prasad and Power 1991). The plant available forms of nitrogen are nitrate and ammonium. Since mineralization is slow, much of the residual inorganic N often originates from excess applications of fertilizers and manures (Schepers and Fox 1989).

Atmospheric sources of N consist primarily of wet and dry deposition and gaseous ammonia adsorption from the atmosphere (Harper et al. 1987). Wet and dry deposition includes various atmospheric oxides of N that are oxidized to nitrate and washed out of the air by precipitation, plus ammonia absorbed on particulate matter.

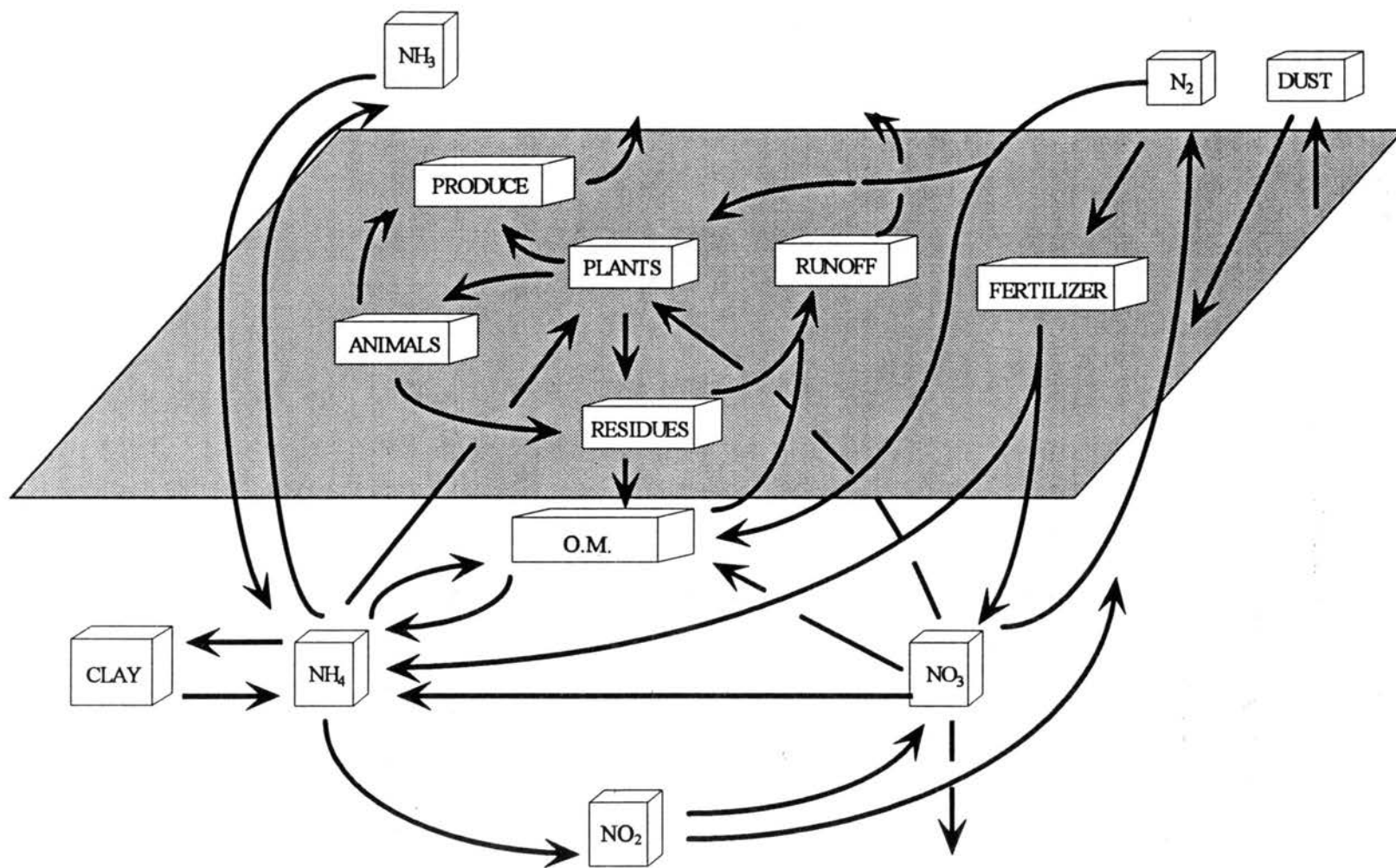


Figure 12. Schematic representation of a general agricultural N cycle (Legg and Meisinger 1982)

In biological nitrogen fixation the active agents are microorganisms that either live in symbiosis with certain species of higher plants (mostly legumes) or nonsymbiotically in soils, water, or on vegetation. Usually fixation varies inversely with the level of available nitrogen in the soil, and hence the nitrogen content of the crop may bear little relation to fixation.

Mineralization of N from organic forms to soluble nitrate (NO_3^-) is a two step process. First, heterogeneous groups of soil organisms hydrolyze proteins and amino acids in the organic fraction of soil to produce ammonium N (NH_4^+). The ammonium form is then oxidized by select groups of bacteria (*Nitrosomonas*, *Nitrococcus*, *Nitrobacter*) to nitrite and then to nitrate forms (Boswell et al. 1985). Ammonium N is formed as a result of ammonification, while nitrate N is formed as a result of nitrification. Bacteria, responsible for nitrification, are generally more sensitive to water deficits than fungi, and may essentially cease to operate in dry soil. On the other hand, ammonification accomplished by more drought tolerant fungi may still proceed. Thus, appreciable amounts of ammonium N usually accumulate in soils during prolonged dry spells. This ammonium N rapidly nitrifies when the soil is again exposed to an environment conducive to activity of nitrifying organisms (Power et al. 1974). However, ammonium N can be converted rapidly to gaseous N (volatilization losses) under alkalinity and high temperature conditions. Thus, alternate wetting and drying of soil, especially in summer or following improper irrigation practices, affects the nitrogen flux. Linn and Doran (1984) have shown that mineralization increases as the percentage of soil pores filled with water increases to about 60%. At higher values, rates of these aerobic processes decrease while rates of anaerobic processes (i.e., denitrification) increase. The 60% threshold approximates the

water content at field capacity. However, as the soil water content increases beyond field capacity, the tendency for nitrates to leach to the ground water increases. Amounts of nitrogen lost depend on a number of factors including, (1) form and amount of soluble and unabsorbed nitrogen present or added, (2) amount and time of rainfall, (3) infiltration and percolation rates, (4) water holding capacity of the soil, (5) presence or absence of a crop, and its growth characteristics; (6) evapotranspiration, (7) rate of removal of nitrogen by the crop, (8) extent to which there is an upward movement of nitrogen in the soil, and (9) whether the nitrogen is leached below the root zone, especially to the ground water (Allison 1965). Also, nitrogen can be lost from the system through runoff. Much of the N transported in surface runoff from cultivated and fertilized fields is organic N which is sediment bound (Langdale et al. 1979).

Thus, the nutrient nitrogen cycle involves the transformations and availability of nutrients from many sources. Through proper soil and crop management, the nutrient cycle can be manipulated in such a manner that nutrient requirements of the growing crop are met but not exceeded at each stage of crop growth. Research to date has identified many of the principles involved in the cycling and integrating of nutrient sources for crop production. Simulation modeling offers an approach to acquiring a comprehensive understanding and predictive capability to the complex interactions involving soils, crops, weather patterns, and management practices.

Study Area

The study area is a quarter section located within the Fort Cobb Reservoir Watershed in Caddo County, west central Oklahoma. These areas are subject to intensive

farming of high-value crops such as peanuts, beans and corn. This is evidenced by the extensive use of inputs such as center-pivot irrigation, fertilizers, and pesticides. These characteristics call for intensive field-level management.

The study area is the southwest quarter of section 31, township 9 north, range 12 west, designated as Farm Service Agency Tract 702 located approximately eight miles south of Eakley, Oklahoma on the west side of Fort Cobb Reservoir (Figure 13). Corn and soybeans, and sometimes watermelons, are the primary crops grown on this farm which is under a center-pivot irrigation system.

Methods

Figure 14 shows a schematic representation of the nitrogen modeling component of this research. Creation of the spatial database and the management database for the study area is described in the following paragraphs. Subsequent to creation of the databases, trial simulations are conducted and explained using the integrated system developed in the previous chapter. Based on the HRUs selected, the integrated system was run for a period of five years to estimate the different losses of nitrogen through leaching, volatilization, runoff, and denitrification. In addition to this, other fluxes of nitrogen were also estimated such as nitrogen in crop uptake, nitrogen in yield, residual nitrogen, mineralized nitrogen, and nitrogen that has been biologically fixed.

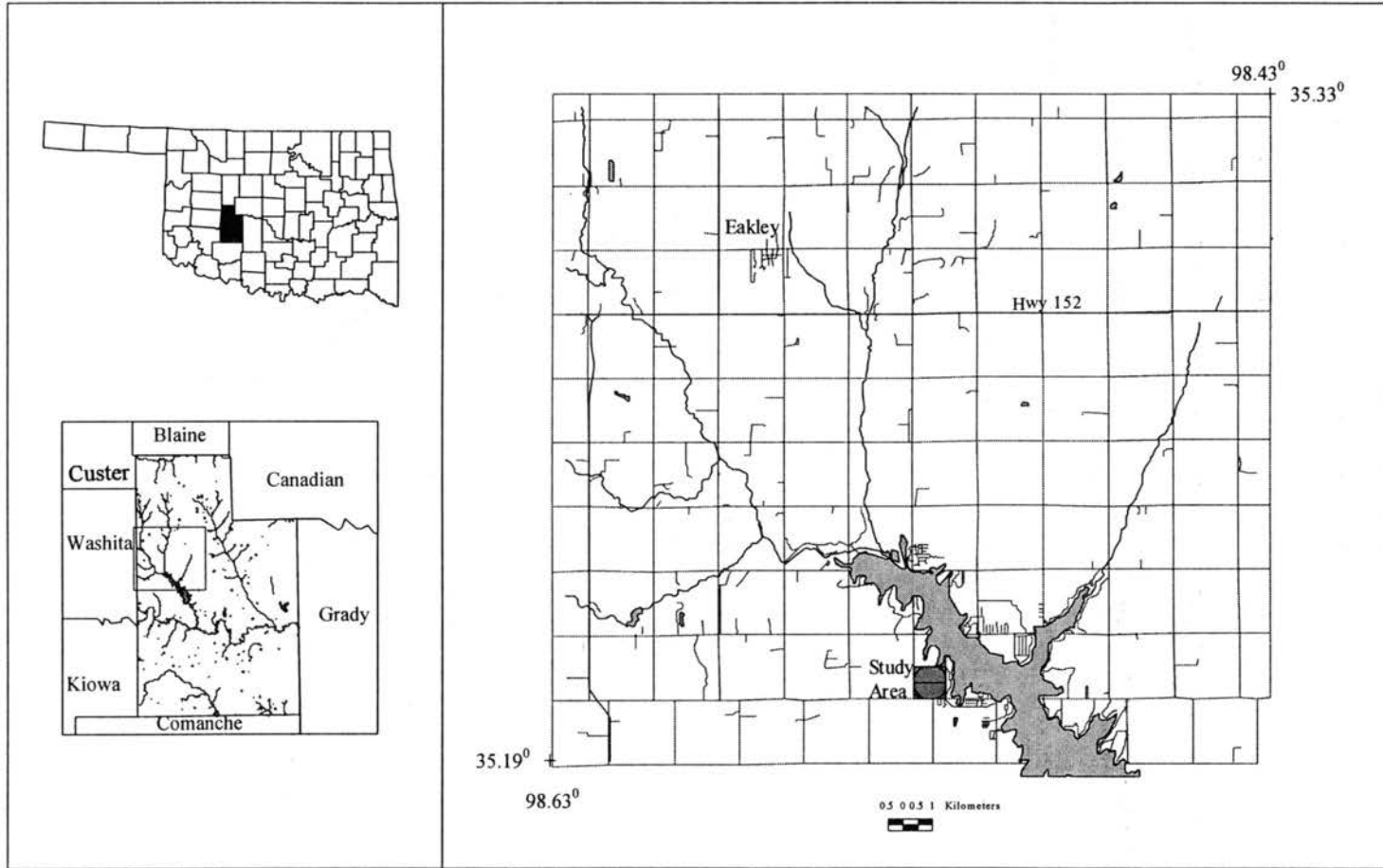


Figure 13. Study area of the project situated in Caddo County, OK.

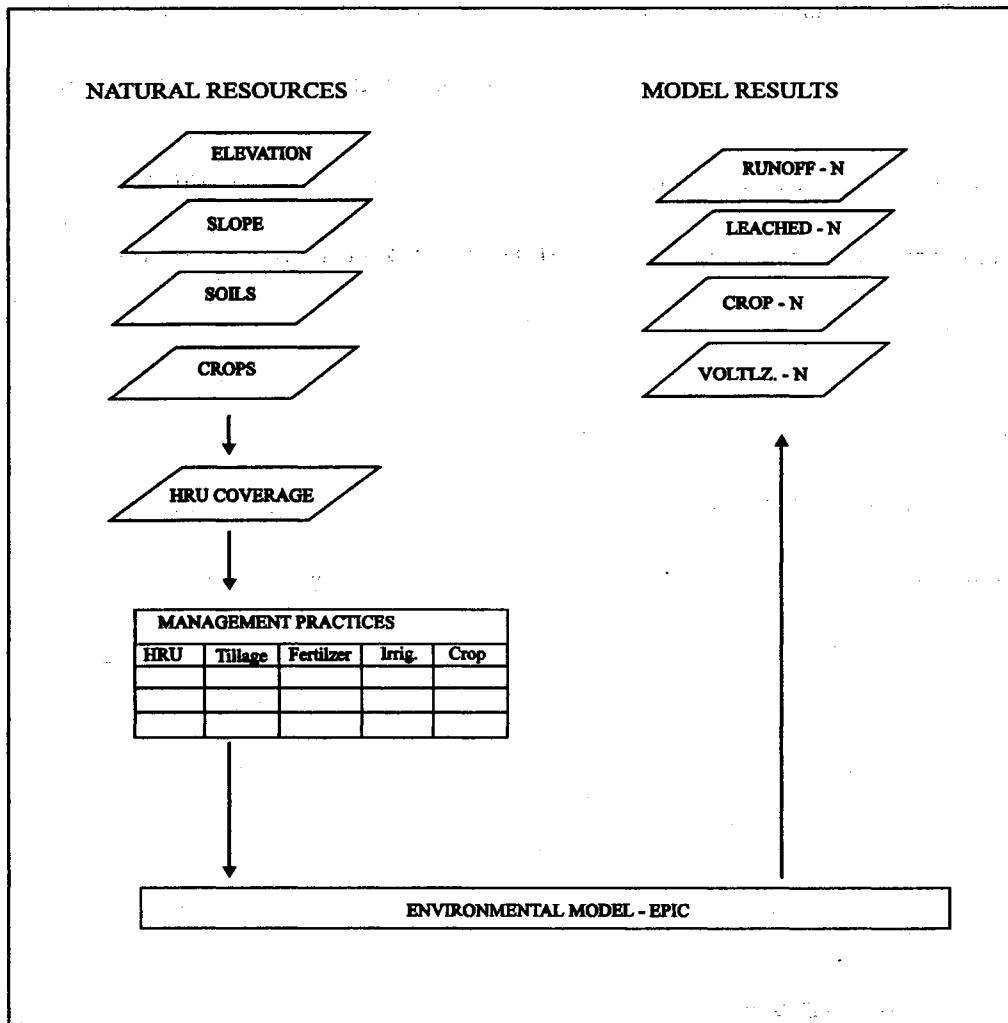


Figure 14. Schematic representation of the N modeling project.

Development of GIS Database

Spatial data pertaining to the area of study were acquired from an ongoing project for the same area (Anderson 1994). Table 1 lists details of the spatial data. The spatial data in vector format were comprised of soil types, land use, and elevation. A raster-based geographic information system capable of spatial analysis was necessary. As such, ARC/INFO Version 7.0 was used to create the spatial database because of its superior hydrologic modeling functions and ease in converting spatial data from one format to another, i.e., vector to grid. This functionality was found lacking in ArcView 2.1, although at the time of writing, ArcView 3.0 (beta) was released which featured a Spatial Analyst module capable of some hydrologic modeling functions.

Development of Hydrologic Response Units (HRUs)

Using the GRID module of ARC/INFO, the georeferenced (UTM) spatial data pertaining to soil types, elevation, and land use were converted to grid layers with a cell resolution of 58 m. The pixel resolution for the grid was set at 58 m, to exclude the possibility of creating 'nodata' value cells during the rasterization process. These 'nodata' value cells are created when either no or insufficient information is available about the particular characteristics of the location. The slope grid was derived from the elevation grid with the help of the 'slope' function. This function is a neighborhood operation which identifies the maximum rate of change in Z value from each cell to its eight neighbors. These grid coverages were then vectorized using the 'gridpoly' function of ARC/INFO. The polygon coverages were combined using the overlay function to create hydrologic response units (HRUs). The HRU layer constituted the base map from which areas were

Table 1. Sources of spatial data used in the development of GIS database.

Data	Source	Details
Soil	Soil Survey of Caddo County, Oklahoma USDA-SCS (1973)	Digitized soil survey maps (1:20000)
Elevation	Field Data	Primary field data collection by John A. Anderson and A. Dewain Davis. Grid sampling at 50 m.
Field boundaries	USGS Topographic 7.5' Quadrangle map (Alfalfa)	Digitized by John A. Anderson.

selected for simulation. Each HRU has unique attributes attached involving soil type, land cover, and elevation or slope. Thus, areas could be selected based on queries to any of the layers that fused to form the HRU layer. For example, the simulation could be done based on areas selected by querying particular soil types, or crops, etc.

Attribute data pertaining to soils prevalent in the study area were acquired from the Soil Survey Investigation Reports (SOILS V data base). The data were saved as UTIL files for easy entry into EPIC Basic Data Files.

In the polygon attribute table of the HRU layer, an item was added to hold data pertaining to curve number (CN). The curve number value for a particular HRU served as an index for the combined hydrologic effects of antecedent moisture, land cover, hydrologic soil group and hydrologic condition such as 'good' or 'poor' in terms of infiltration. A look-up table was created containing the CN's (U.S. Department of Agriculture 1972) and later joined to the attribute table of the HRU layer.

Development of EPIC Base Data

The development of the EPIC base data involved the entry of data which are uniform for the particular site such as weather, and several parameters. Other data entered included the input and output options of the model. The resultant data are common for all HRUs delineated on the field.

Base data required by EPIC were entered using the Constant Data menu item of EPIC-View (Appendix III.a). These data remain constant for all model runs. EPIC is capable of estimating the values of most of these parameters, and default parameter

estimates were entered. Furthermore, the intention of this project was not to test the model, and hence default estimates for parameters were selected. The base data were loaded in the text file - CONST.DAT using UTIL. The weather data were entered using the Weather Data tool of EPIC-View. The latitude (35.21) and longitude (98.51) of the study area were entered which enabled EPIC to look for the nearest weather station (Weatherford). Using UTIL, weather statistics for each month of the year were automatically loaded into CONST.DAT. This also included monthly wind data.

Once the file CONST.DAT was created, it was replicated for all HRUs delineated. This resulted in the creation of FORM#.DAT which holds the base data for all HRUs delineated. Thus, an EPIC Basic Data File was assembled for each HRU. The data file was ready for simulation runs except for the spatial data and the management data part of the data file which were loaded during the model run.

Development of the Management Practices Database

Data pertaining to the management practices were obtained from the operator of the farm. The management data consisted of particulars of farm operations such as tillage, planting, agrochemical usage, etc., for corn-soybeans rotation under a conventional tillage system during the year 1991. The data were input using the Management Data tool of EPIC-View. Other data pertaining to management options required by EPIC were also entered (Appendix III.b and Appendix III.c). The management information was stored in UTIL files for ready access by UTIL during model runs.

Simulation of Farm Level Nitrogen

The GIS-based integrated modeling system developed was used to estimate average annual rates for nitrogen transformations at the sub-field level. The simulation was run for 5 years to provide an estimate of various sources and sinks of nitrogen in a conventional tillage cropping system of corn-soybeans rotation. The model was not calibrated for any parameters, owing to lack of data for calibration purposes.

The different steps in the simulation to estimate average annual rates for different forms of nitrogen involved selecting the HRUs for model run, accessing the spatial database and extracting the spatial data, specifying the output options, and running the model.

1. Selection of HRUs: the selection of HRUs for the model run was done by querying the HRU layer for corn. The selected HRUs were saved to a file which listed their ID number. A circular zone of 160 HRUs falling under the center-pivot irrigation system was selected for the simulation exercise.

2. Accessing the spatial database and loading the data: the spatial database of the HRU layer was accessed by selecting the Spatial Data menu item of EPIC-View. The spatial data (soil, elevation, slope, area, and curve number) pertaining to each selected HRU were accessed sequentially and written to a UTIL file (FORM#.UTL).

3. Specifying the output options: before running the model, the Output Options item was selected to specify the output of the model. Output variables pertaining to the nutrient nitrogen component of EPIC were selected (Table 2).

Table 2. Output variables pertaining to the nitrogen nutrient cycling component of EPIC.

Output Variable	Variable Description
UNO3	N uptake by crop (kg ha ⁻¹)
YNO3	Nitrate loss in runoff (kg ha ⁻¹)
YON	N loss with sediment (kg ha ⁻¹)
PRKN	Nitrate leached (kg ha ⁻¹)
SSFN	Nitrate loss in subsurface flow (kg ha ⁻¹)
AVOL	N volatilized (kg ha ⁻¹)
MNN	N mineralized (kg ha ⁻¹)
HMN	N mineralized from stable organic matter (kg ha ⁻¹)
NFIX	Nitrogen fixation (Biological) (kg ha ⁻¹)
NITR	Nitrification (NH ₃ -N conversion to NO ₃ -N) (kg ha ⁻¹)
DN	N loss by denitrification (kg ha ⁻¹)
TNO3	Total NO ₃ -N present in the soil profile (kg ha ⁻¹)
TNH3	Total NH ₃ -N present in the soil profile (kg ha ⁻¹)
FNO3	Fertilizer nitrate (kg ha ⁻¹)
FNH3	Fertilizer ammonia (kg ha ⁻¹)

4. Model run: estimating the different losses of nitrogen for each HRU was done by selecting the Model Run menu item of EPIC-View. The UTIL files (FORM#.UTL and MGMT#.UTL) were invoked by UTIL to load the spatial and management data into the respective FORM#.DAT. This resulted in a complete user supplied data file acceptable by EPIC. Consequently, EPIC was run sequentially on the FORM#.DAT of the selected HRUs. EPIC model outputs for each run, by default, an *.OUT and a *.SUM text files. The '*' represents FORM# of the EPIC Basic Data File. The output file FORM#.OUT is a detailed listing of the model output and is very comprehensive, while the output file FORM#.SUM is an average summary of the selected output variables. Thus, the summary output file represents the mean output of the variables for the specified number of years simulated.

The summary files output by EPIC are space delimited, and for ArcView to convert the output to a table, the file needs to be in a comma delimited format. This was achieved by parsing the FORM#.SUM files to create a composite comma delimited output file (FORMS.PRS). Basically, the parser program refers to an index file which lists the identity number (ID) of selected HRUs. Based on the index, the FORM#.SUM files were sequentially parsed to a comma delimited format and consolidated into a common file (FORMS.PRS). Thus, FORMS.PRS consisted of sequential rows of selected ID numbers and corresponding model output values.

Creation and Display of Nitrogen Modeling Results

The model output of the simulation run can be viewed as tables, charts, and maps. For this study, model results were displayed as maps. Maps showing rates of nitrogen fluxes for each selected HRU were created by selecting the Maps menu item of EPIC-View. The model results in the form of thematic maps were drawn to the same view as the HRU coverage. Thematic maps were later imported into ArcView's layout documents for hardcopy printouts.

Results and Discussion

Nutrient management for agricultural production is a highly complex process. The effect of fertilizer use in the agricultural sector has been a major concern since the 1960s. When this concern first received public attention, there was little scientific evidence to support those who contend that fertilizers were a great threat to the environment or who believed there was little or no environmental danger associated with fertilizer application. Fortunately, there is now a significant amount of data from which definitive conclusions can be made. Research to date has developed many of the principles involved in the cycling, managing, and integrating of nutrient sources for crop production. Much has been learned about the various sources of crop-available nutrients including nutrient transformations by soil microbial activity. Thus, as we continue to accumulate knowledge of this type, a better management plan is required to manage N and other nutrients in agricultural production.

Computer simulation models provide a cost-effective tool to integrate the various factors involved in the crop production system, and eventually assist in the decision

making process. One of the main problems of physically-based models is the time required for data preparation and computation of hydrologic parameters may be many times greater than the actual time needed for simulation and result interpretation. The interface of GIS with hydrologic/water quality models result in a heightened efficiency because the user can define the boundaries of the modeling unit or field. The computer can then use these boundaries to retrieve the spatial data from the GIS database and compute parameters required for the hydrologic model (Muzik 1988). The integration of data and the process of synthesizing tremendous amounts of information for the examination of spatial and temporal characteristics of fields or watersheds are easily expedited. The relative magnitude of many hydrologic processes operating in natural landscapes are sensitive to topographic positions (Moore et al. 1988). Towards this end, involvement of GIS in hydrologic analyses of watersheds or fields is valuable. Furthermore, from a sustainable farming point of view, it is desirable to manage the various resources in a manner such that productivity is enhanced with simultaneous environmental quality benefits.

Integrated System Framework and Design

In this study, an integrated system framework was designed interfacing ArcView and EPIC. The graphical user interface (EPIC-View) served as the system-user link thorough which the different Avenue and Visual basic programs were executed. The execution of the programs, display of the GUI, and access to the various databases were accomplished in the user-friendly environment of ArcView. The EPIC model was embedded in ArcView, and using the integrated system, nitrogen losses as well as other transformations of nitrogen were estimated. This facilitated the storage of simulation

results as attributes of the GIS coverage for subsequent retrieval and mapping. GIS representation of model results allows improved process visualization (Vieux 1991).

Data Coverages and Hydrologic Response Units

The spatial analysis capabilities of ARC/INFO resulted in the creation of the spatial data coverages. The original vector data for the study area were rasterized using the GRID module of ARC/INFO to perform spatial analysis, mainly deriving the slope coverage. Furthermore, discretization of the coverages with an intention to capture the spatial variability of the coverage was achieved by this process. The coverages were then vectorized to extract area edges of the discrete units represented by one type of attribute such as soil type, cover, etc. Also, since the basic spatial data model in ARC/INFO is vector-based, and in order to perform the overlay analysis using the coverages for HRU delineation, this vectorization process was necessary. Another important benefit of this process was elimination of spurious polygons (“slivers”) during the overlay of coverages to create HRUs. The different categories in each coverage are shown in Figures 15 to 18. Figure 15 shows the major land cover for the coverage, i.e., corn, grassland, and wheat pasture. Figure 16 shows the different soil series, i.e., Konawa loamy fine sand (KoC2), Eufaula fine sands (Efd), Dougherty loamy fine sands (DoB), Dougherty and Eufaula loamy fine sands (Dud). The elevation and slope categories are shown in Figure 17 and Figure 18, respectively.

The grid data structure was used in creating the coverages, consisted of a square grid network used to facilitate computer implementation and attain computational efficiency (Moore and Grayson 1991). Grids are used extensively in hydrologic modeling

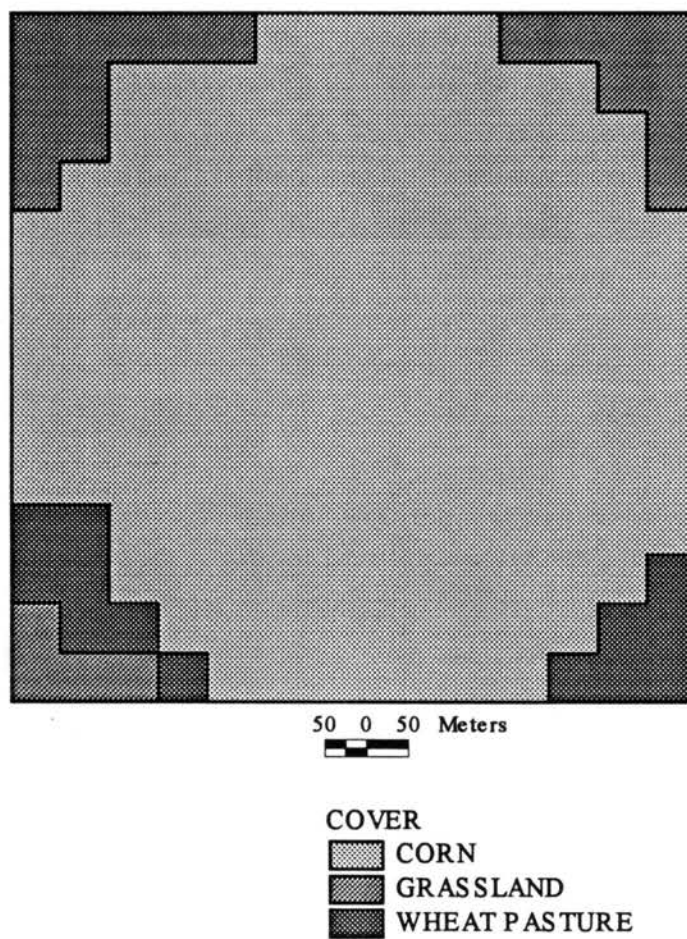
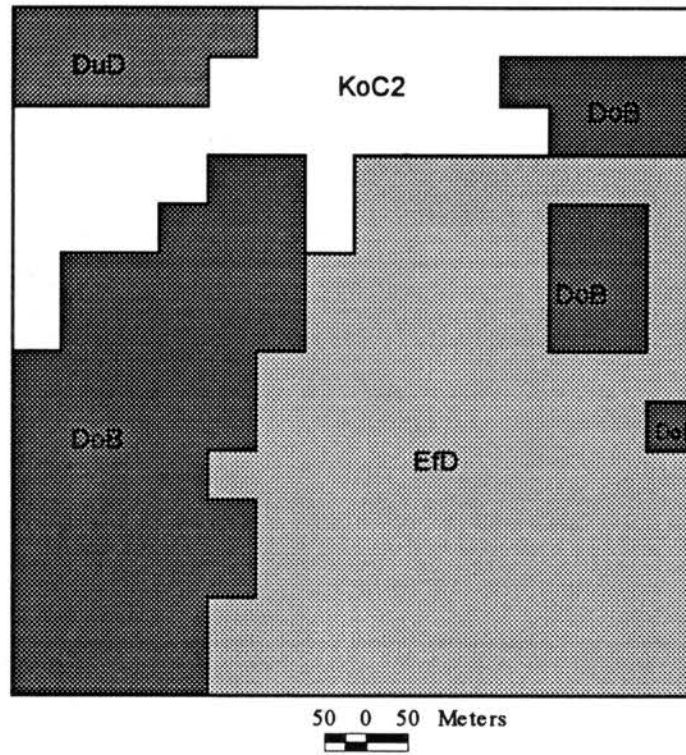


Figure 15. Map showing major land cover categories existing in the study area.

**SERIES**





-  DoB (Dougherty loamy fine sand)
-  DuD (Dougherty and Eufaula loamy fine sand)
-  EfD (Eufaula fine sand)
-  KoC2 (Konawa loamy fine sand)

Figure 16. Map of the major soil series existing in the study area.

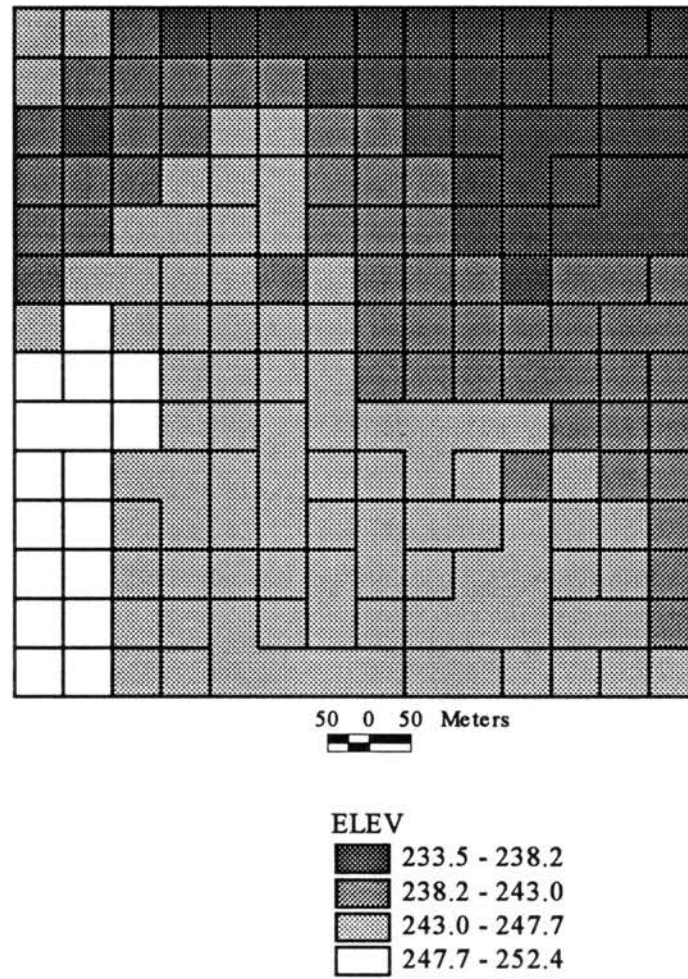


Figure 17. Map showing the elevation (m) categories existing in the study area.

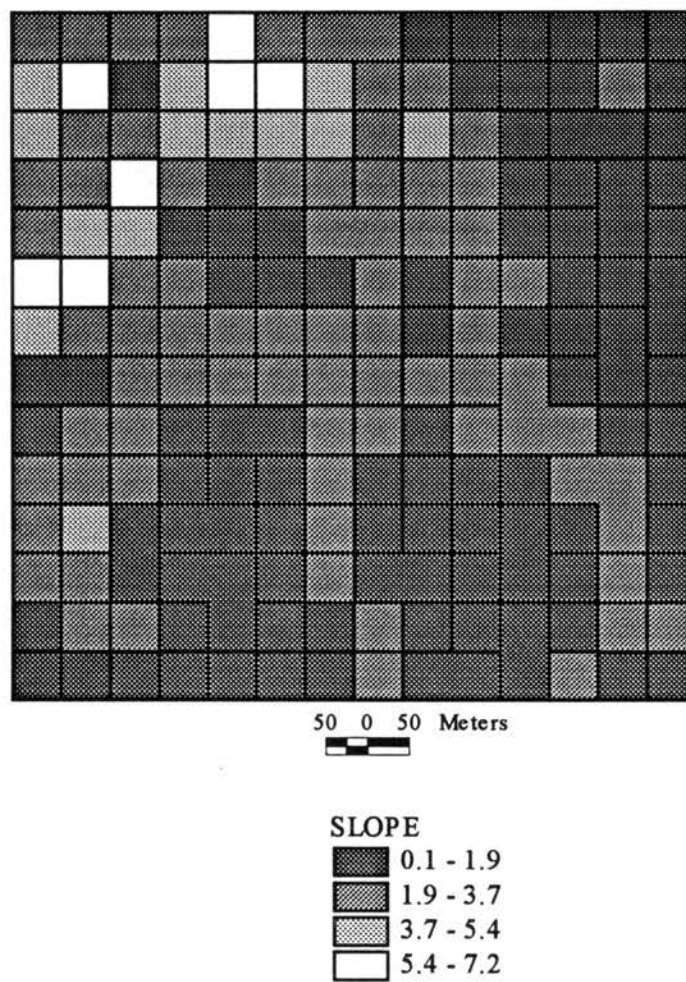


Figure 18. Map showing the slope categories (%) existing in the study area.

as a finite difference representation of planar flow processes, that is, flows that are essentially horizontal. However, the disadvantages of using grids include difficulty in handling abrupt changes in elevation, resolution effects on results, and zigzag pattern of flow paths used in hydrologic analyses (Moore and Grayson 1991). In other hydrologic modeling studies, triangulated irregular networks (Maidment et al. 1989, Silfex et al. 1987) and contour data (Moore et al. 1988, Moore and Grayson 1991) were used.

A total of 195 HRUs were delineated from the overlay of coverages, creating the HRU coverage (Figure 19). Each HRU was characterized by unique attributes specified by a single soil type, elevation, slope, and land cover. This overlay capability of GIS applies to intervariable relationships where several variables are evaluated simultaneously while holding location constant (Avery and Berlin 1985). The resulting coverage containing the HRUs had an entirely new topology and descriptive database representation. Since each HRU is homogeneous it is expected to respond hydrologically in a uniform manner (Gan 1990). A similar approach for decomposing a watershed into homogenous polygons was applied in another study (Curtis 1992). The attribute table of the resultant HRU coverage was modified to include CN values needed by the hydrology component of EPIC for simulating runoff. The values estimated ranged from 67 to 78 for the existing land cover and soil hydrologic group.

The use of GIS expedited data integration and synthesis of information for the study area. It allowed the integration of data into a comprehensive system capable of combining as well as cross-referencing diverse data elements. The flexibility of a GIS permits the development of a common database to be used by GIS as well as rainfall

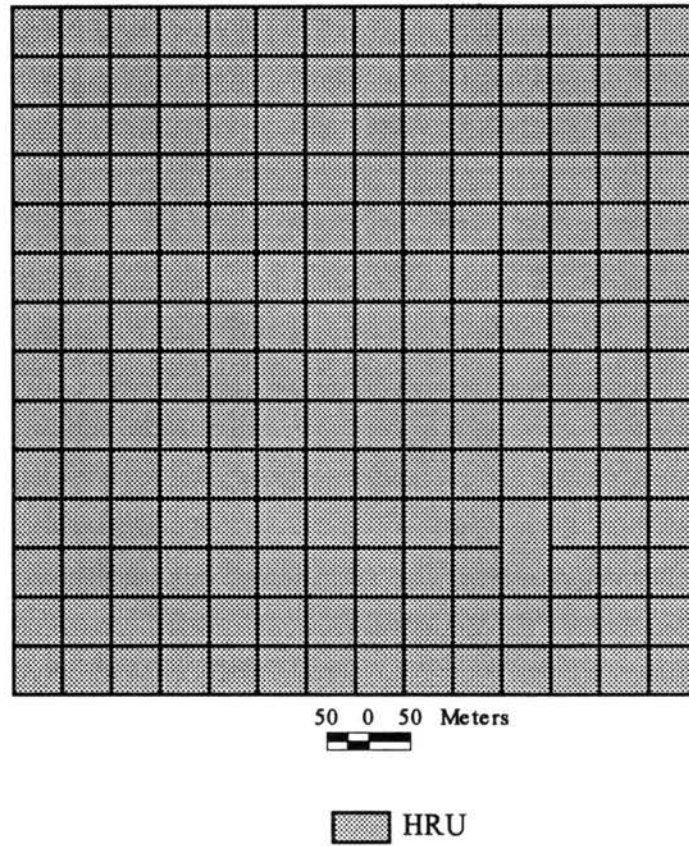


Figure 19. Hydrologic response units (HRUs) delineated by overlaying crop, soil, elevation, and slope coverages.

runoff predictive models (Smith et al. 1990). Thus, better access to data and visualization of disparate data elements were achieved.

Estimation of N Transformation Rates

Trial operations of the integrated system were undertaken to simulate different N transformation rates under conventional tillage for the 160 HRUs covered by the center pivot irrigation system. However, validation of system results was not conducted because of unavailable observed data for the study area as a basis for comparison.

The estimated N transformation rates for the study areas were shown as thematic maps (Figures 20 to 31). The existence of spatial variations in the sources and sinks for N are clearly evident from the thematic maps. For the selected set of HRUs under the center-pivot irrigation system, the mean annual N uptake by the crop ranged from 476.3 to 527.9 kg ha⁻¹, while the leaching losses averaged 79 to 104 kg ha⁻¹ annually. Mean annual loss of N through volatilization ranged between 1.7 to 2.4 kg ha⁻¹. Between 1 to 4 kg ha⁻¹ and 0 to 3 kg ha⁻¹ was lost as nitrates in surface runoff and subsurface flow respectively, while sediment bound organic N loss in runoff ranged from 116.7 to 239.2 kg ha⁻¹ annually. On the other hand, nitrification of ammonia to nitrate averaged 110.9 to 111.6 kg ha⁻¹ annually. Biological N fixation averaged from 107.9 to 126.4 kg ha⁻¹. The average annual rate of N mineralization ranged from 69.7 kg ha⁻¹ to 97.9 kg ha⁻¹ while N mineralized from stable organic matter ranged from 13.5 to 35.9 kg ha⁻¹. Residual nitrogen present in the soil profile ranged between 43.5 to 45.5 kg ha⁻¹ annually for total NH₃-N, and 594.5 to 710.4 kg ha⁻¹ annually for total NO₃-N.

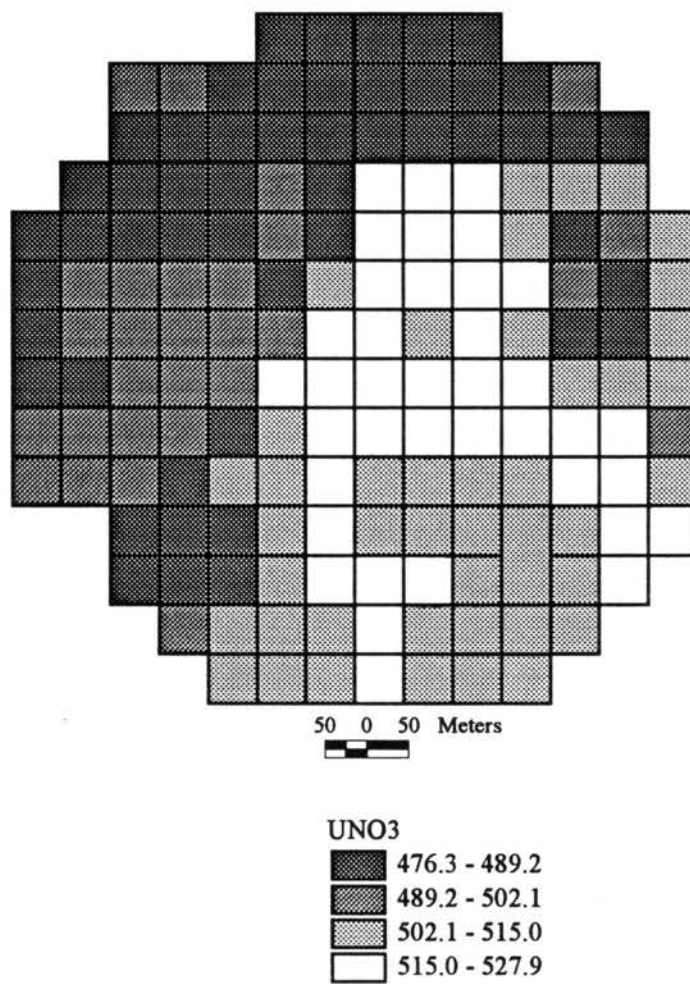


Figure 20. Map of average N (kg ha^{-1}) uptake by crop (UNO3) estimated using EPIC-View under conventional tillage conditions.

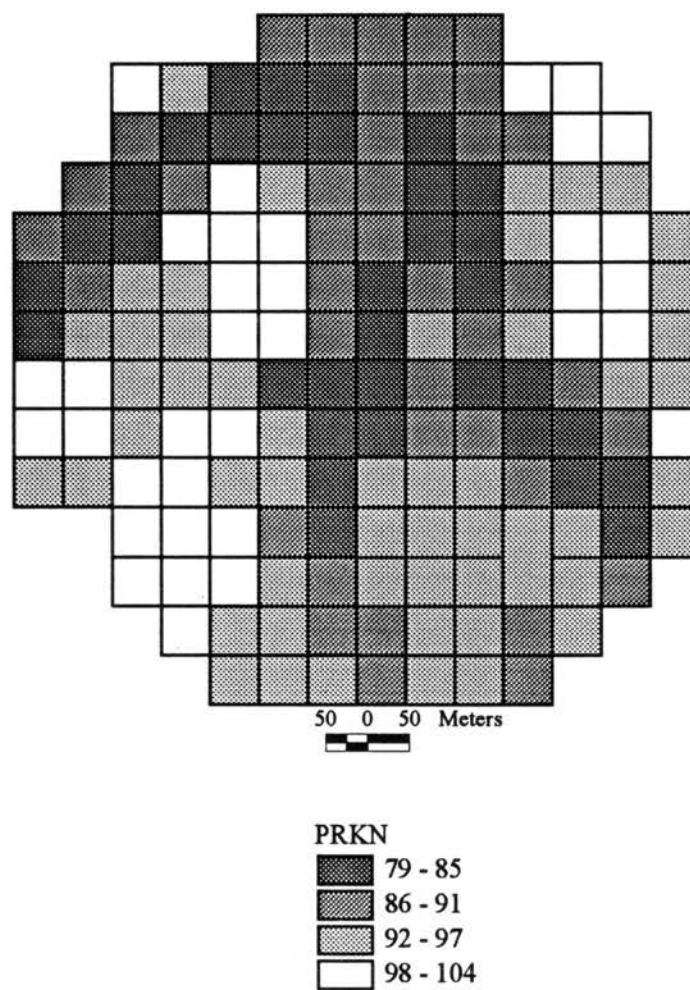


Figure 21. EPIC-View estimated map showing average annual mineral N loss (kg ha^{-1}) in percolate (PRKN) under conventional tillage conditions.

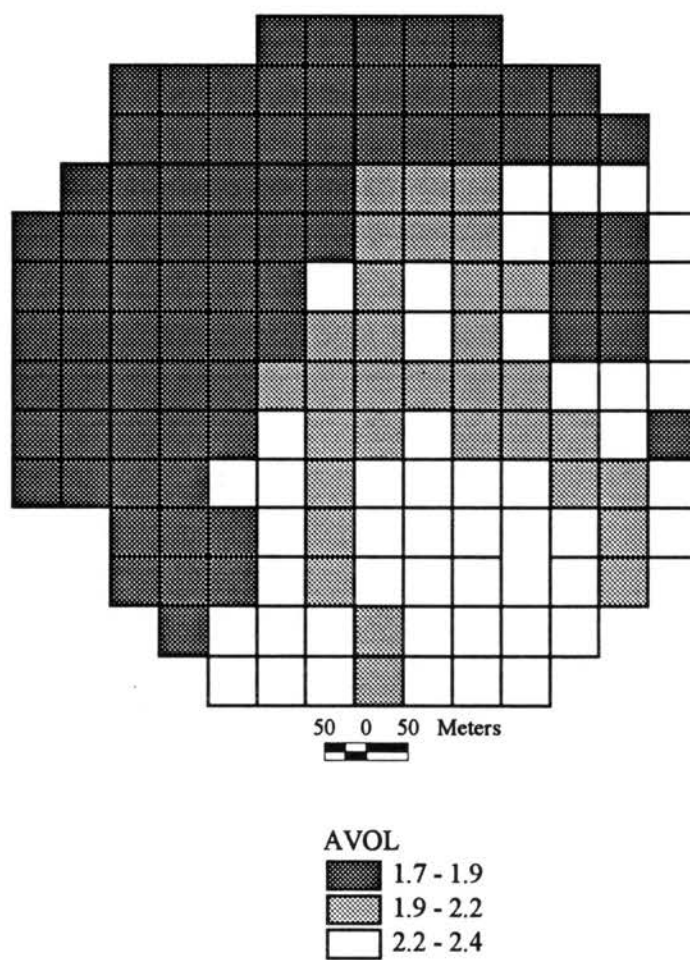


Figure 22. EPIC-View estimated map of average annual rate (kg ha^{-1}) of N volatilization (AVOL) under conventional tillage conditions.

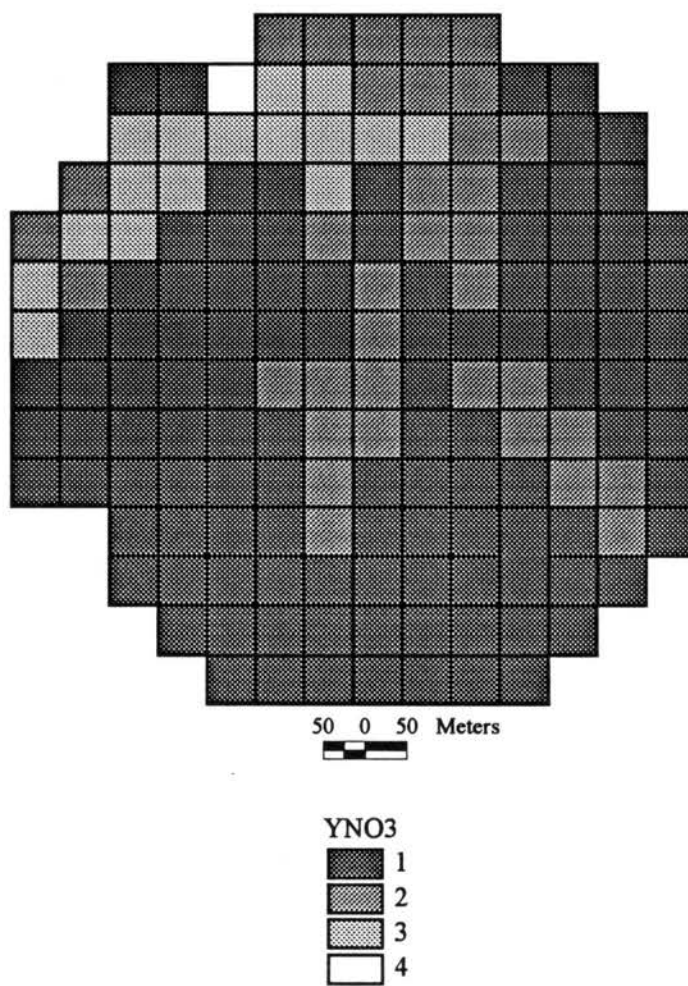


Figure 23. Map showing average annual nitrate (kg ha^{-1}) loss in surface runoff (YNO3) estimated using EPIC-View under conventional tillage conditions.

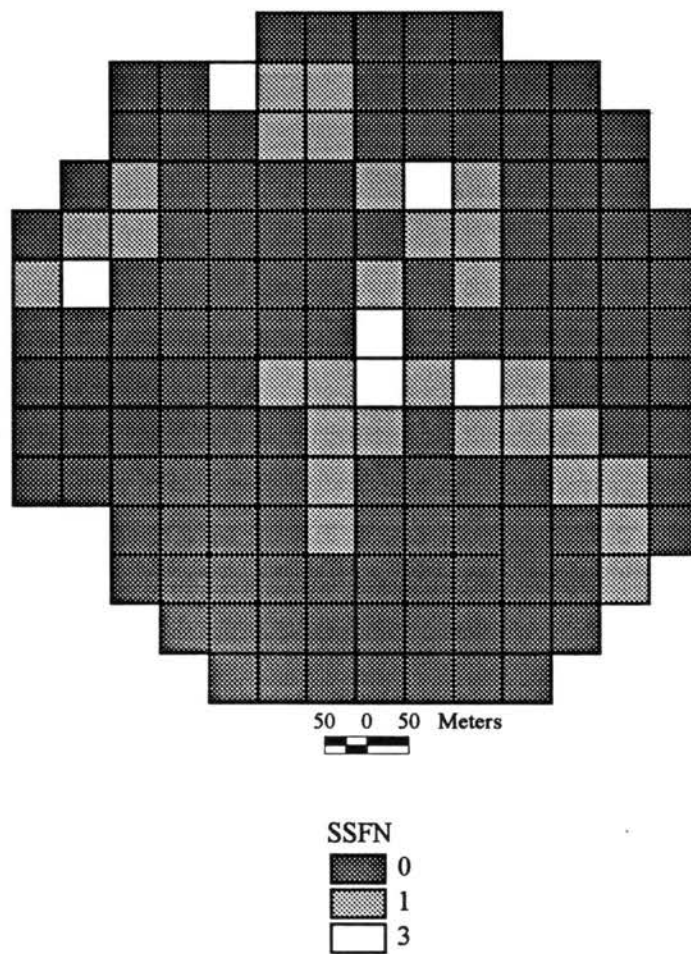


Figure 24. EPIC-View estimated map of average annual mineral N loss (kg ha⁻¹) in subsurface flow (SSFN) under conventional tillage conditions.

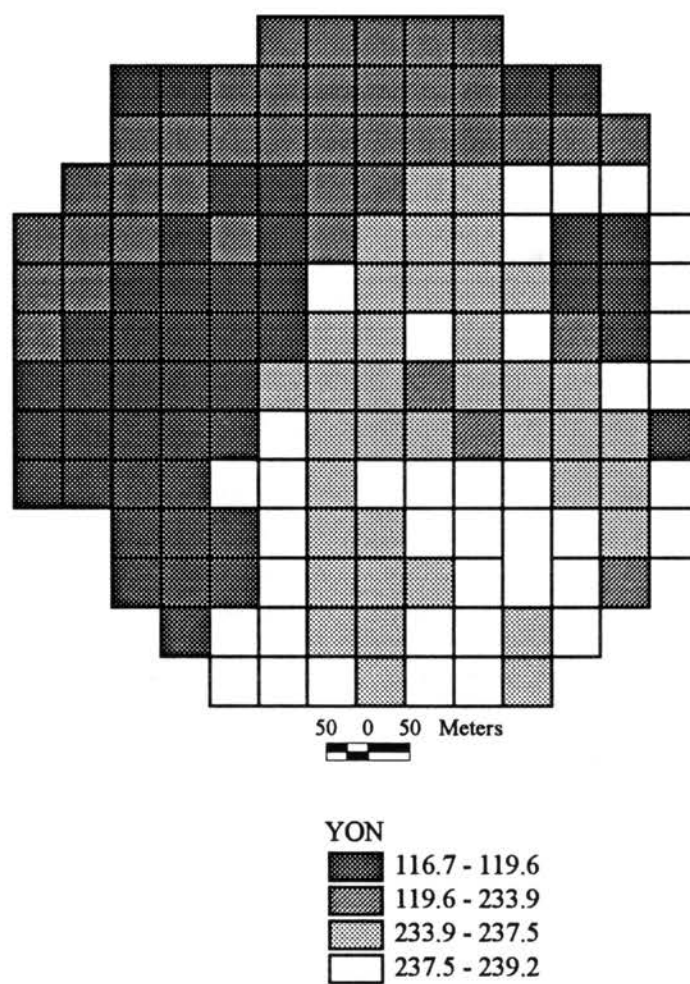


Figure 25. Map showing EPIC-View estimated average annual organic N (kg ha^{-1}) loss with sediment (YON) under conventional tillage conditions.

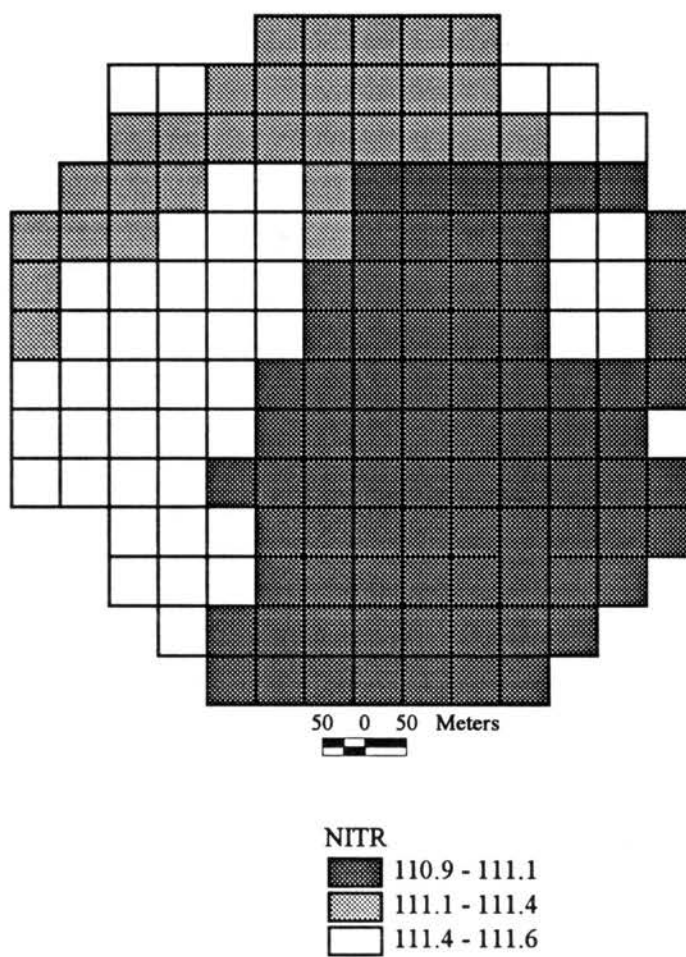


Figure 26. EPIC-View estimated map of average annual rate (kg ha^{-1}) of nitrification (NITR) under conventional tillage conditions.

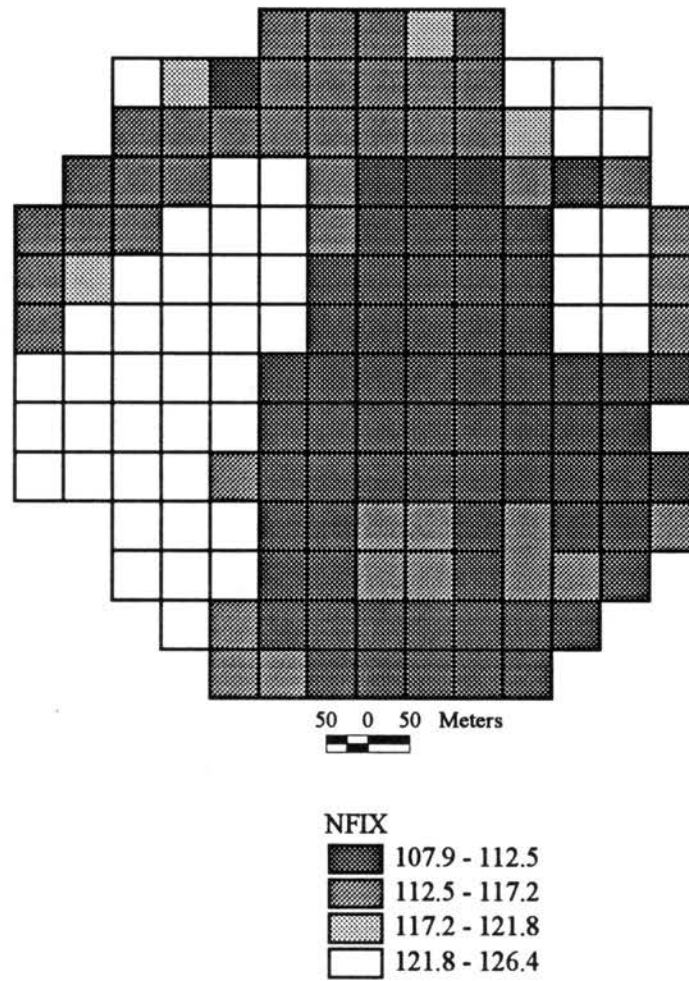


Figure 27. EPIC-View estimated map of average annual rate (kg ha^{-1}) of N fixation (NFIX) under conventional tillage conditions.

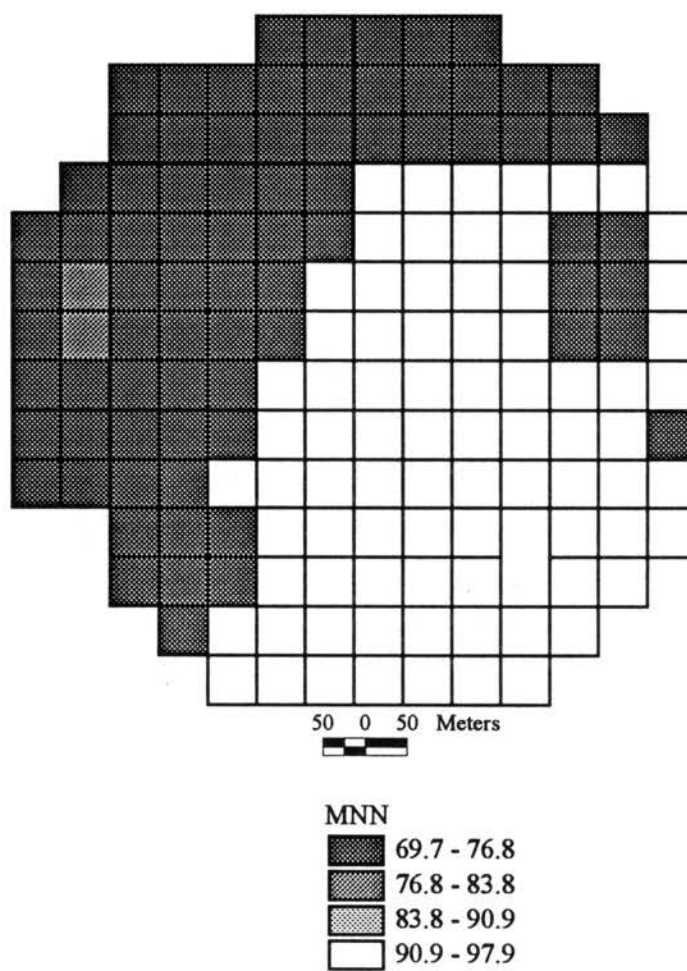


Figure 28. EPIC-View estimated map of average annual rate (kg ha⁻¹) of N mineralization (MNN) under conventional tillage conditions.

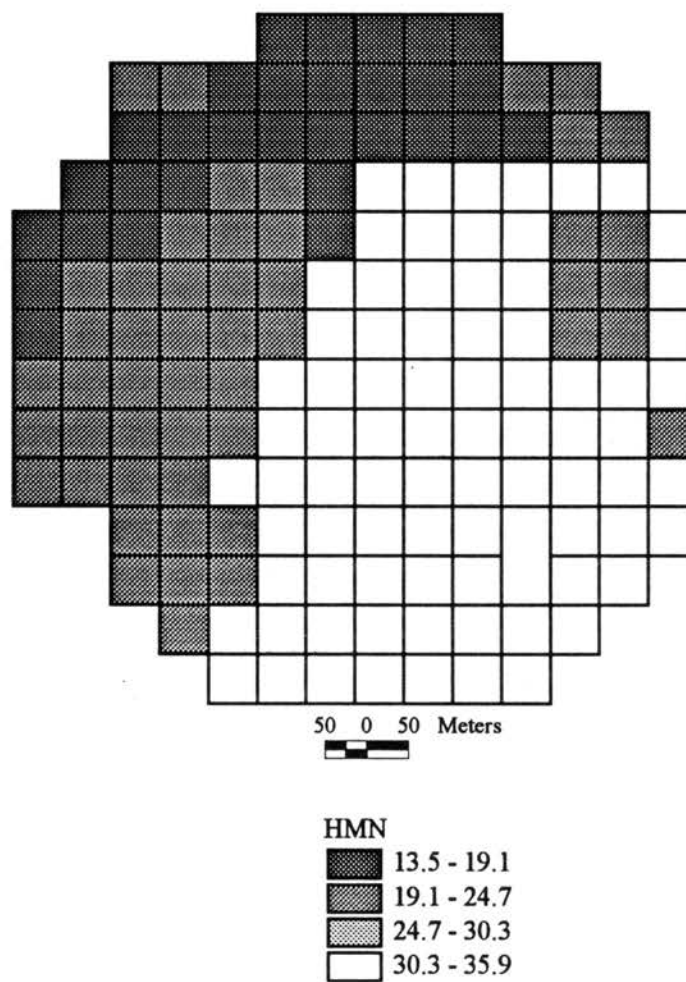


Figure 29. EPIC-View estimated map of average annual rate (kg ha) of N mineralized from stable organic matter (HMN) under conventional tillage conditions.

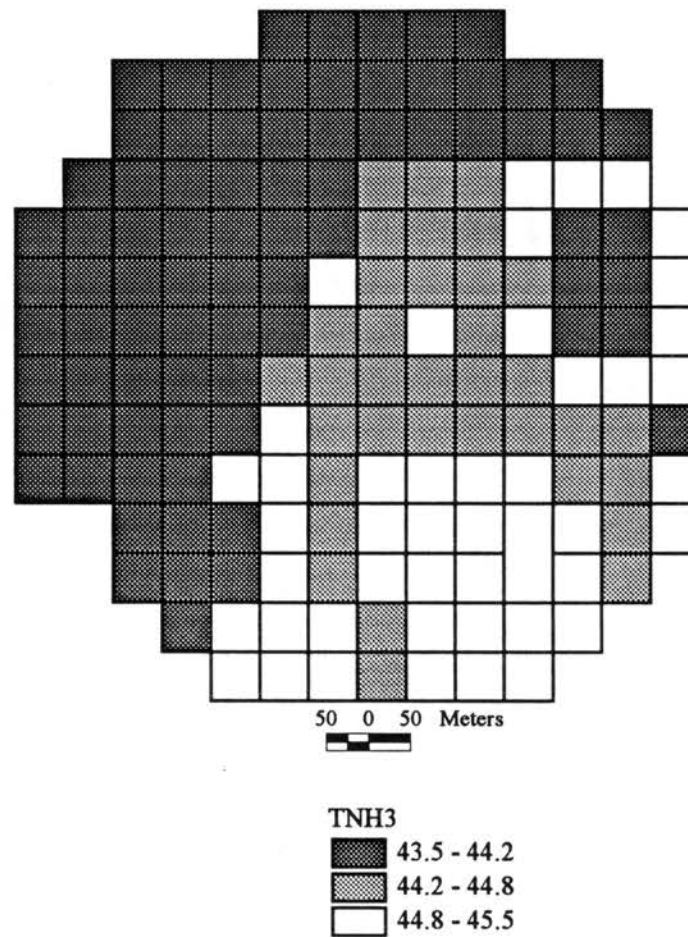


Figure 30. Map showing EPIC-View estimated map of average annual $\text{NH}_3\text{-N}$ (kg ha^{-1}) present in the soil profile (TNH3) under conventional tillage conditions.

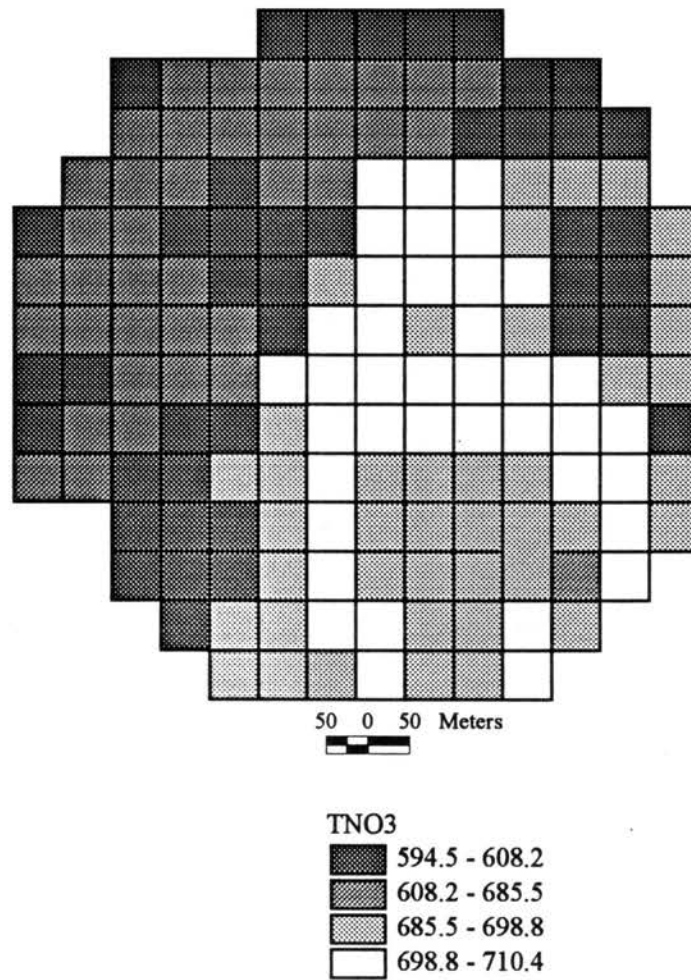


Figure 31. Map showing EPIC-View estimated average annual $\text{NO}_3\text{-N}$ (kg ha^{-1}) present in the soil profile (TNO3) under conventional tillage conditions.

Summary statistics for the different output variables pertaining to nitrogen are shown in Table 3. Large variances are evident for N variables. Notably HMN, MNN, TNO₃, YON, UNO₃ exhibit high absolute variance. However, the relative variability as expressed by the coefficient of variation (CV) shows that SSFN is highly variable, followed by YNO₃, HMM, YON, and to a lesser extent AVOL and MNN. There was negligible loss of nitrogen through denitrification, and hence it is not reported. The well drained conditions existing in the study area owing to the sandy texture of soils, thus inhibiting anaerobic conditions might explain the low denitrification losses. Also, sprinkler irrigation has the potential to minimize denitrification losses due to its positive effect on soil aeration. An average annual balance sheet for nitrogen was calculated based on the mean values of N variables for the five-year simulation (Table 4).

An alternate farm management scenario representing a minimum tillage cropping system was simulated to observe the change in nitrogen dynamics. Input data were held constant except for management practices which were altered to reflect minimum tillage (Appendix III.d). The N outputs for the alternate scenario are illustrated as thematic maps (Figures 32 to 43). The spatial distribution of N transformations are clearly evident in the maps. For the same HRUs, the mean annual N uptake by the crop ranged from 473.9 to 519.7 kg ha⁻¹ while leaching losses averaged 82 to 109 kg ha⁻¹ annually under minimum tillage conditions. Mean annual loss of N through volatilization ranged from 1.7 to 1.9 kg ha⁻¹. Between 1 to 4 kg ha⁻¹ and 0 to 3 kg ha⁻¹ were lost as nitrates in surface runoff and in subsurface flow respectively, while sediment bound organic N loss in runoff ranged between 109.0 to 224.6 kg ha⁻¹ annually. Gains of nitrogen by nitrification of ammonia

Table 3. Summary statistics for the different nitrogen variables modeled using EPIC-View for conventional tillage conditions.

N variable (kg ha ⁻¹)	Mean	SD	VAR	CV
AVOL	2.0	0.3	0.1	0.1
HMN	27.3	8.8	77.2	0.3
MNN	85.0	11.0	120.1	0.1
NFIX	115.7	6.1	37.2	0.1
NITR	111.2	0.3	0.1	0.00
PRKN	91.8	6.0	36.2	0.1
SSFN	0.3	0.7	0.4	2.2
TNH3	44.4	0.7	0.5	0.0
TNO3	652.8	45.2	2046.3	0.1
YNO3	1.4	0.7	0.5	0.5
YON	179.8	57.9	3352.6	0.3
UNO3	501.8	15.1	227.1	0.0

Table 4. Balance sheet of nitrogen estimated using mean values of the different N variables for conventional tillage conditions.

Nitrogen variable (kg ha ⁻¹)	GAIN	LOSS
AVOL		2.1
HMN	27.3	
MNN	85.0	
NFIX	115.7	
NITR	111.2	
PRKN		20.7
SSFN		0.1
TNH3	36.6	
TNO3	147.5	
YNO3		0.3
YON		179.8
UNO3		501.8
FNO3	6.7	
FNH3	113.3	
<i>SUM</i>	643.3	704.8

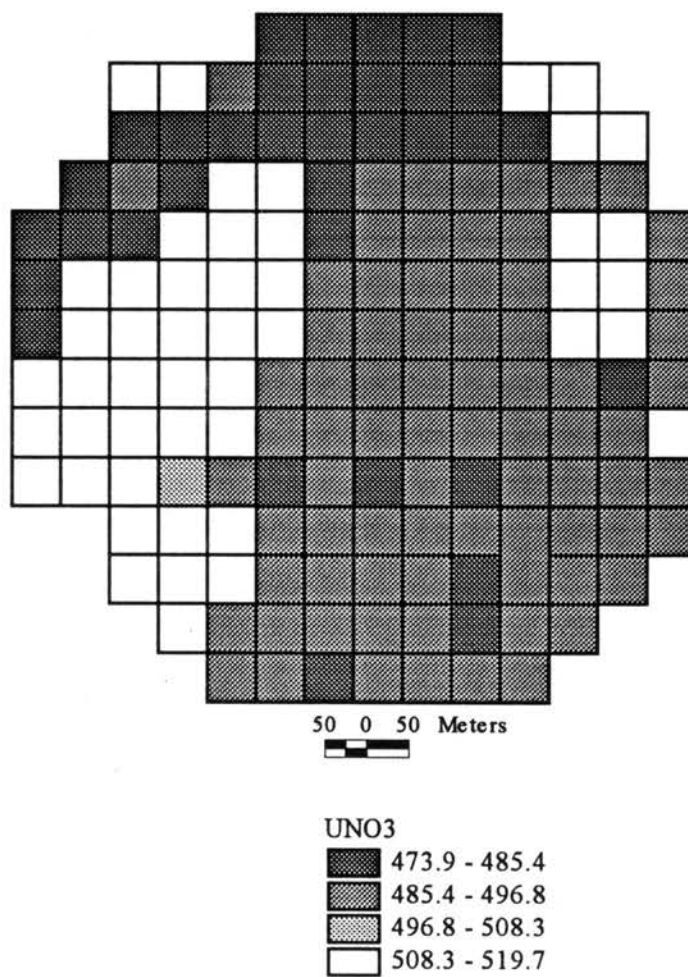


Figure 32. EPIC-View estimated map of average annual N (kg ha⁻¹) uptake by crop (UNO3) under minimum tillage conditions.

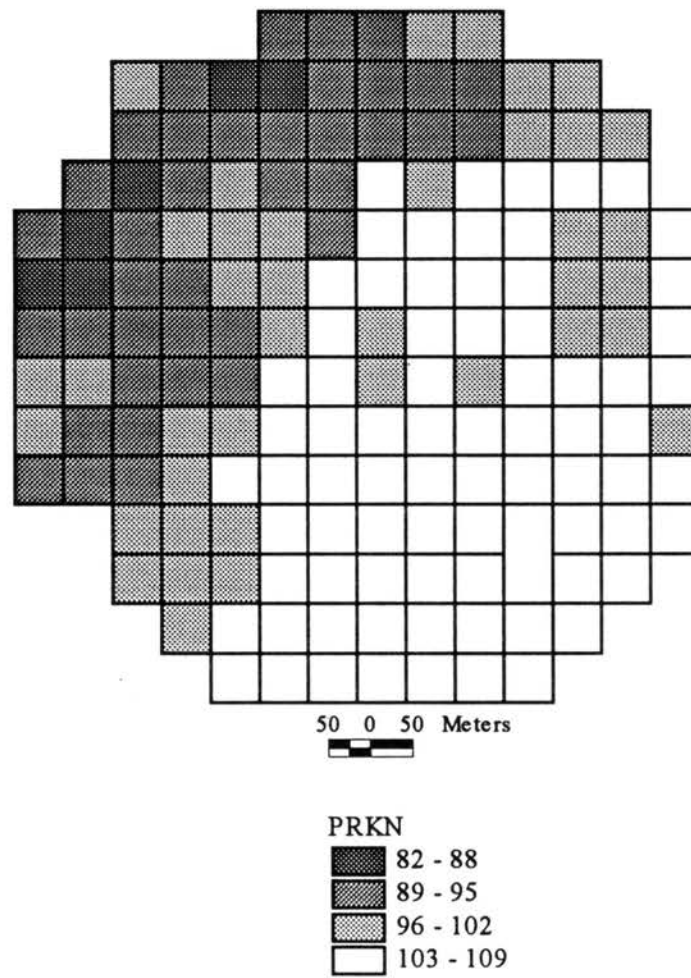


Figure 33. EPIC-View estimated map of average annual mineral N loss (kg ha^{-1}) in percolate (PRKN) under minimum tillage conditions.

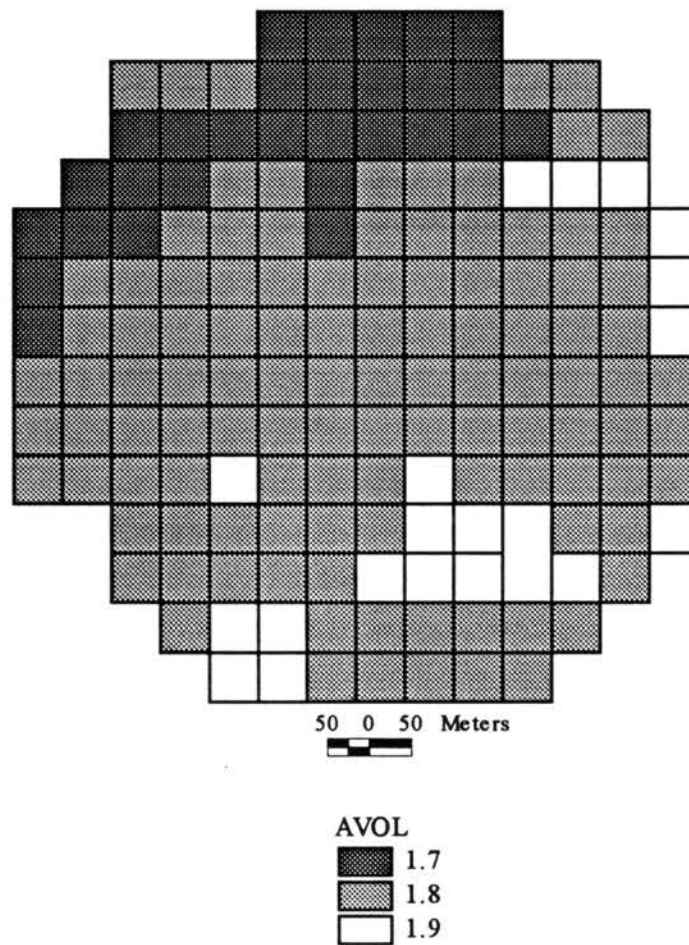


Figure 34. EPIC-View estimated map of average annual rate (kg ha^{-1}) of N volatilization (AVOL) under minimum tillage conditions.

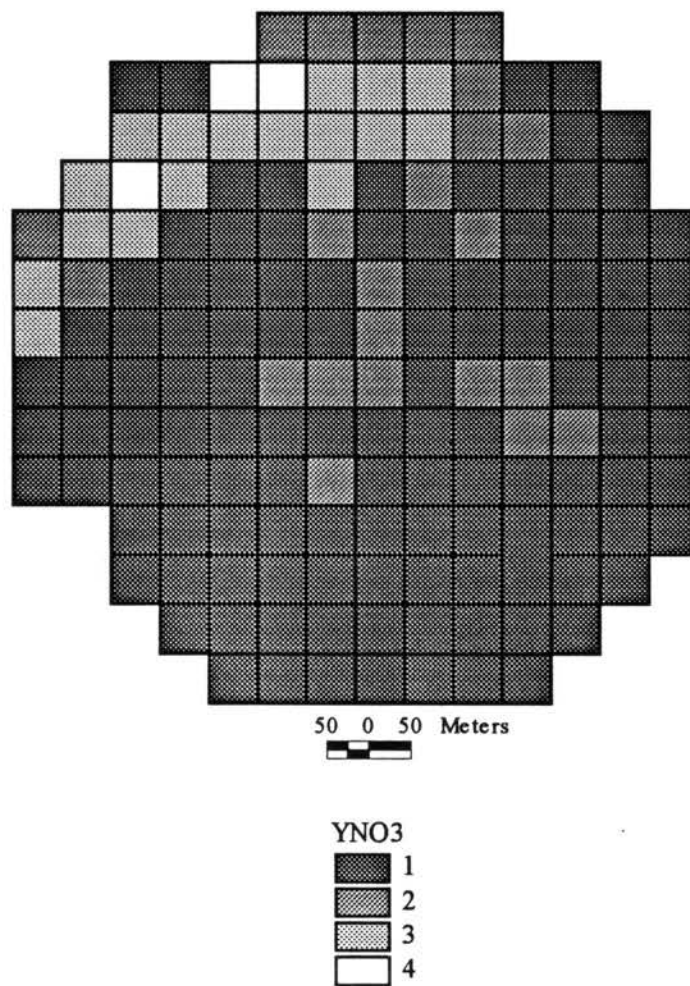


Figure 35. EPIC-View map showing average annual nitrate loss (kg ha^{-1}) in surface runoff (YNO3) under minimum tillage conditions.

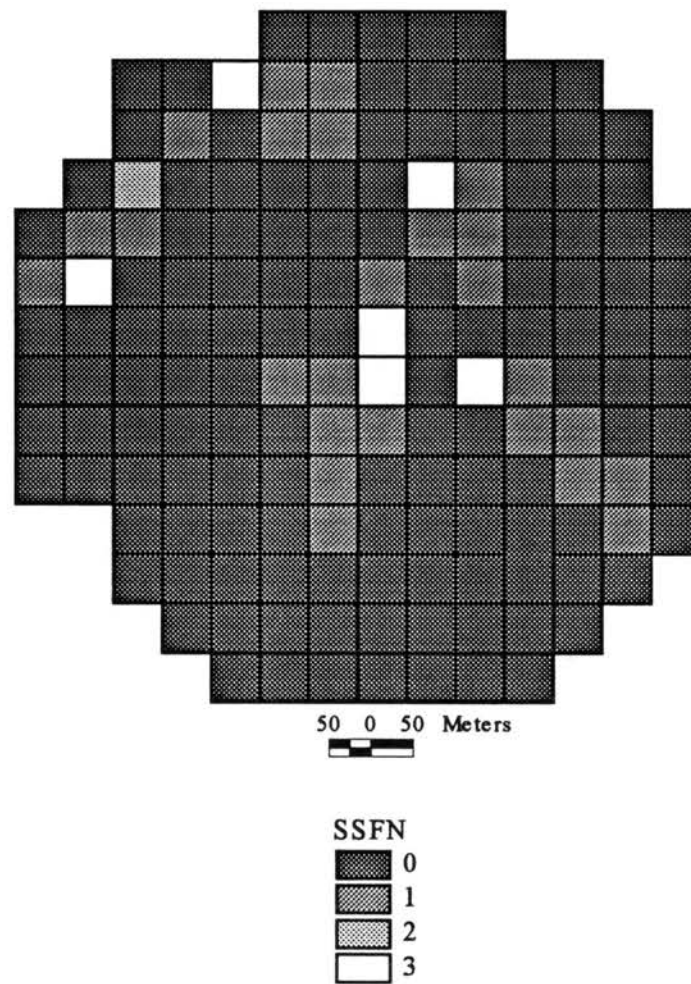


Figure 36. EPIC-View estimated map showing average annual mineral N loss (kg ha^{-1}) in subsurface flow (SSFN) under minimum tillage conditions.

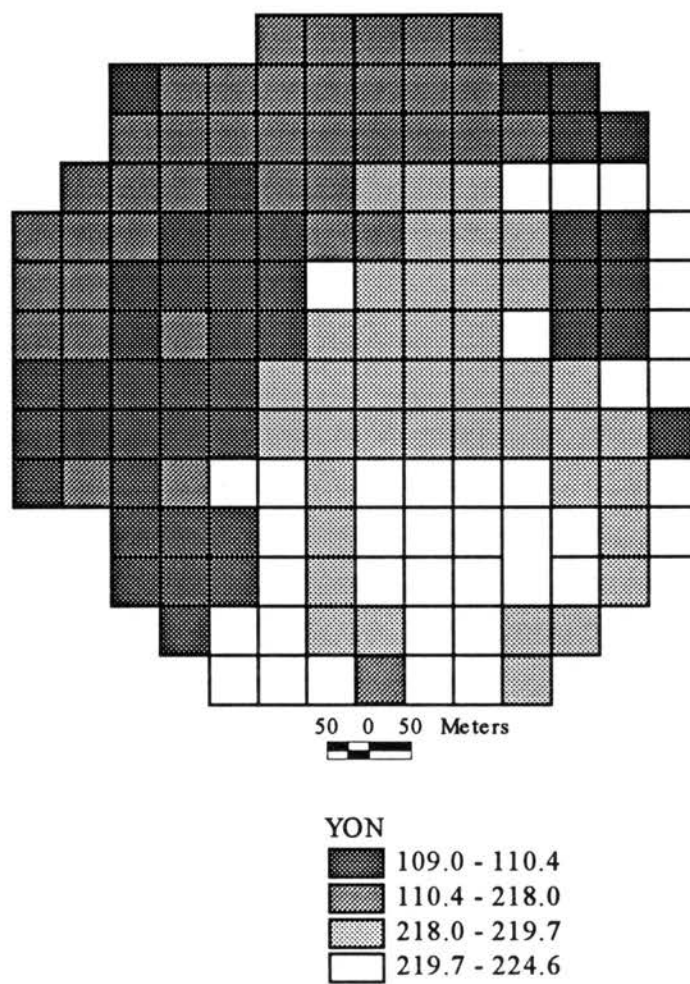


Figure 37. EPIC-View estimated map of average annual organic N loss (kg ha^{-1}) with sediment (YON) under minimum tillage conditions.

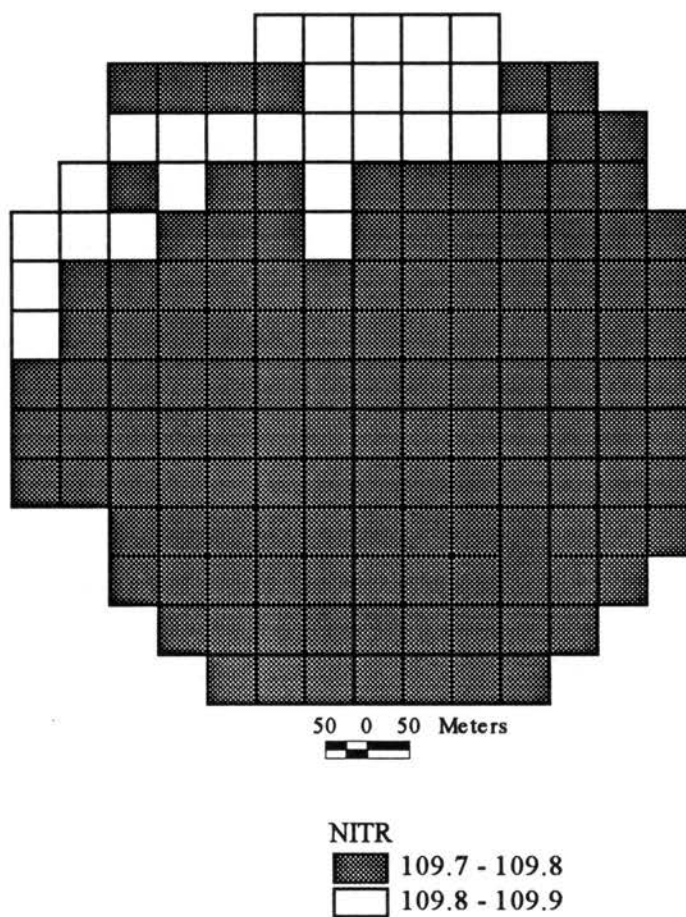


Figure 38. EPIC-View estimated map of average annual rate (kg ha^{-1}) of nitrification (NITR) under minimum tillage conditions.

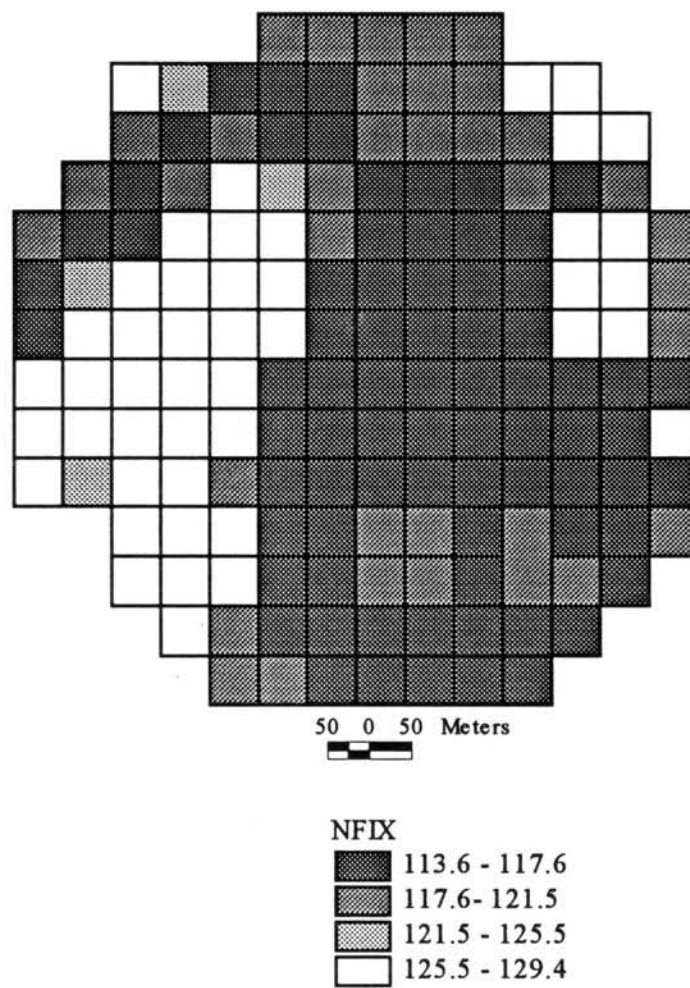


Figure 39. EPIC-View estimated map of average annual rate (kg ha^{-1}) of N fixation (NFIX) under minimum tillage conditions.

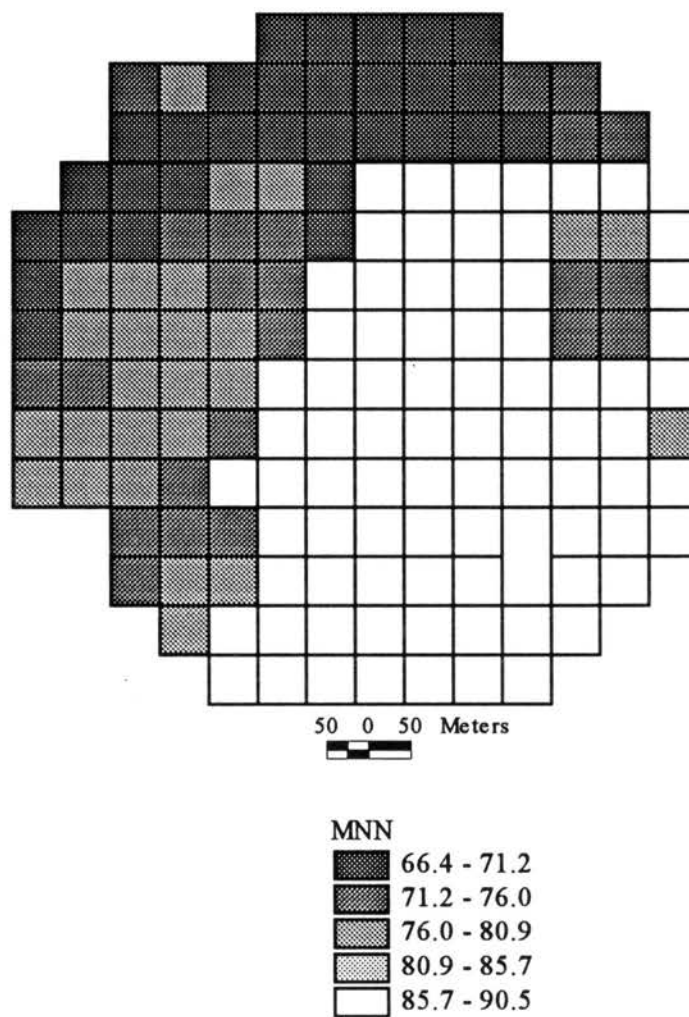


Figure 40. EPIC-View estimated map of average annual rate (kg ha^{-1}) of N mineralization (MNN) under minimum tillage conditions.

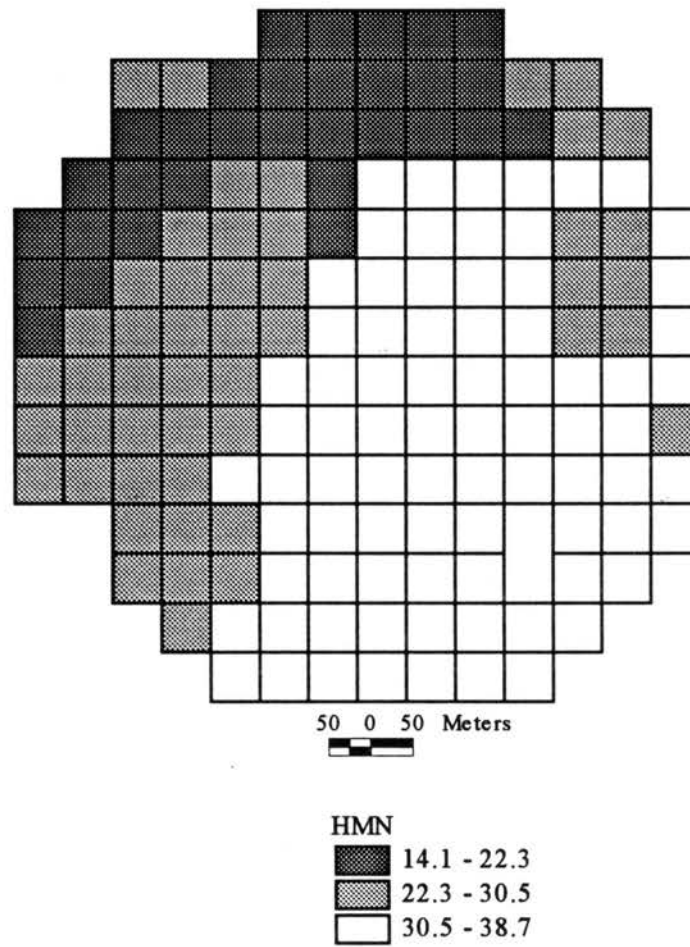


Figure 41. EPIC-View estimated map of average annual rate (kg ha^{-1}) of N mineralization from stable organic matter (HMN) under minimum tillage conditions.

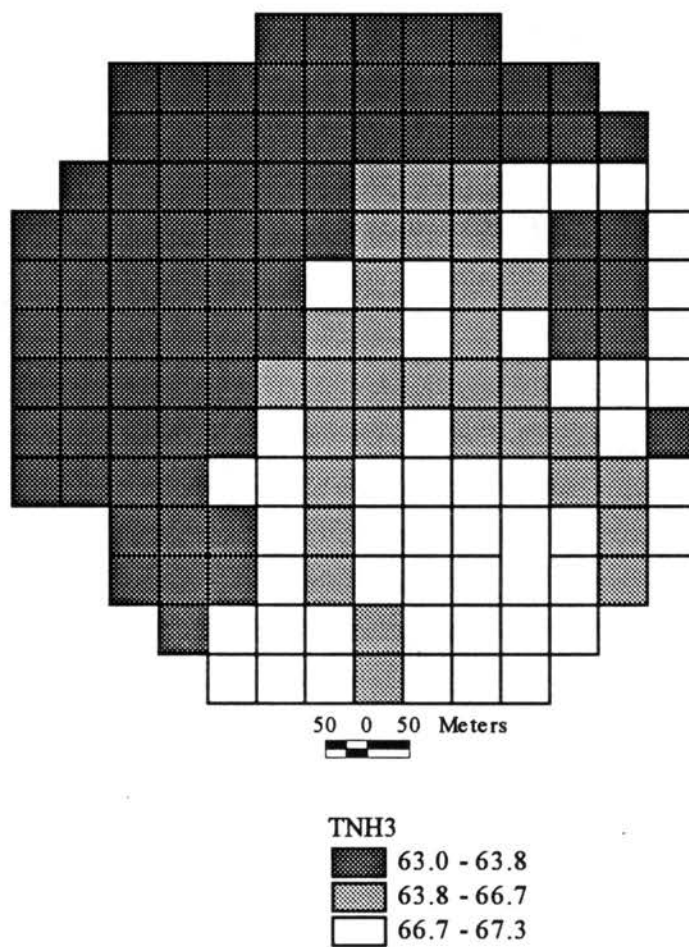


Figure 42. EPIC-View estimated map of average annual $\text{NH}_3\text{-N}$ (kg ha^{-1}) present in the soil profile (TNH3) under minimum tillage conditions.

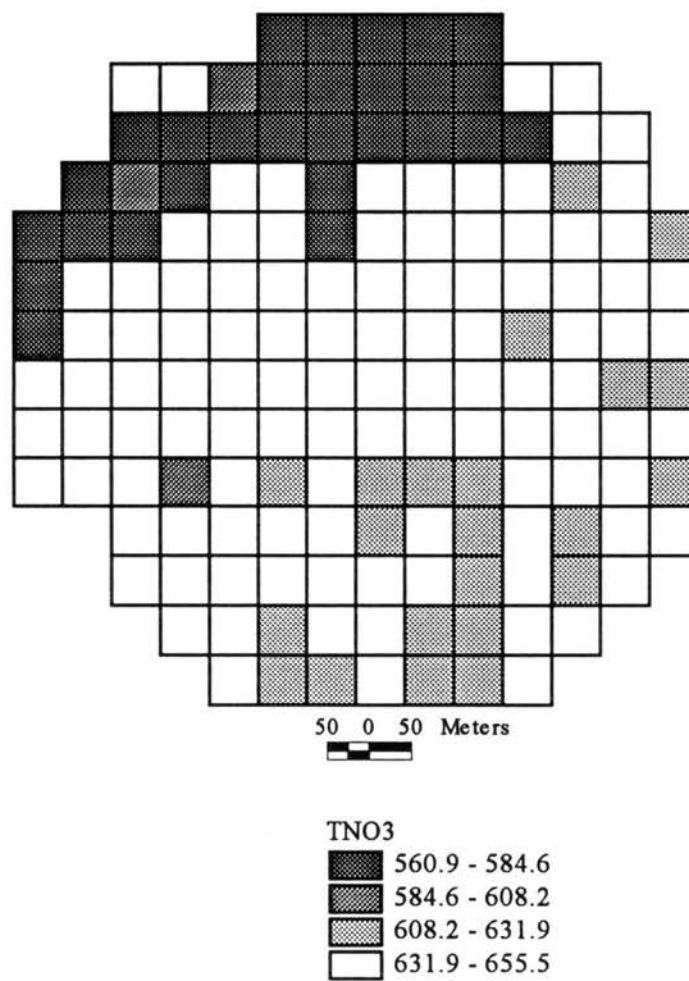


Figure 43. EPIC-View estimated map of average annual $\text{NO}_3\text{-N}$ (kg ha^{-1}) present in the soil profile (TNO3) under minimum tillage conditions.

to nitrate averaged 109.8 kg ha^{-1} annually. Biological N fixation averaged 113.6 to 129.4 kg ha^{-1} . The average annual rate of N mineralization ranged from 66.4 kg ha^{-1} to 90.5 kg ha^{-1} while N mineralized from stable organic matter ranged from 14.1 to 38.7 kg ha^{-1} . Residual nitrogen present in the soil profile ranged from 63.0 to 67.3 kg ha^{-1} annually for total $\text{NH}_3\text{-N}$, and from 560.9 to 655.5 kg ha^{-1} annually for total $\text{NO}_3\text{-N}$.

The basic statistics for the simulation run are depicted in Table 5. Large variances are evident for the different nitrogen variables. Notably HMN, MNN, TNO3, YON, UNO3 exhibit high absolute variance. However, the coefficient of variation shows SSFN to be highly variable, followed by YNO3, HMM, YON, and to a lesser extent AVOL and MNN. Although the trend observed was similar to the trend observed for conventional tillage, the magnitude of the variances were considerably reduced under minimum tillage conditions. There was a negligible loss of nitrogen through denitrification. An average annual balance sheet for nitrogen was calculated based on the mean values for N variables of the 5-year simulation (Table 6).

The thematic maps and tables produced as a result of the simulations for the two management scenarios help in interpreting some results. For example, the pattern of the spatial distribution of yield under the two management scenarios (Figures 44 to 47) follow the pattern of the N uptake map as well that of residual N in the soil profile. The coefficient of variation provides a useful index to summarize variability existing in the distribution of each variable relative to others. Table 7 provides a comparative summary of the relative variability existing in the maps under conventional tillage and minimum tillage

Table 5. Summary statistics for the different nitrogen variables modeled using EPIC-View for minimum tillage conditions.

N variable (kg ha ⁻¹)	Mean	SD	VAR	CV
AVOL	1.8	0.0	0.00	0.03
HMN	29.2	9.9	97.7	0.34
MNN	81.5	9.1	82.0	0.11
NFIX	120.0	5.1	26.0	0.04
NITR	109.8	0.1	0.0	0.00
PRKN	100.0	6.4	41.1	0.06
SSFN	0.3	0.7	0.4	2.28
TNH3	65.1	1.8	3.2	0.03
TNO3	627.7	26.5	701.2	0.04
YNO3	1.4	0.8	0.6	0.53
YON	167.8	53.7	2881.6	0.32
UNO3	495.1	13.5	180.8	0.03

Table 6. Balance sheet of nitrogen estimated using EPIC-View for minimum tillage conditions.

Nitrogen variable (kg ha ⁻¹)	GAIN	LOSS
AVOL		1.8
HMN	29.2	
MNN	81.5	
NFIX	120.0	
NITR	109.8	
PRKN		22.6
SSFN		0.1
TNH3	53.7	
TNO3	141.9	
YNO3		0.3
YON		167.8
UNO3		495.1
FNO3	12.4	
FNH3	111.6	
<i>SUM</i>	660.1	687.7

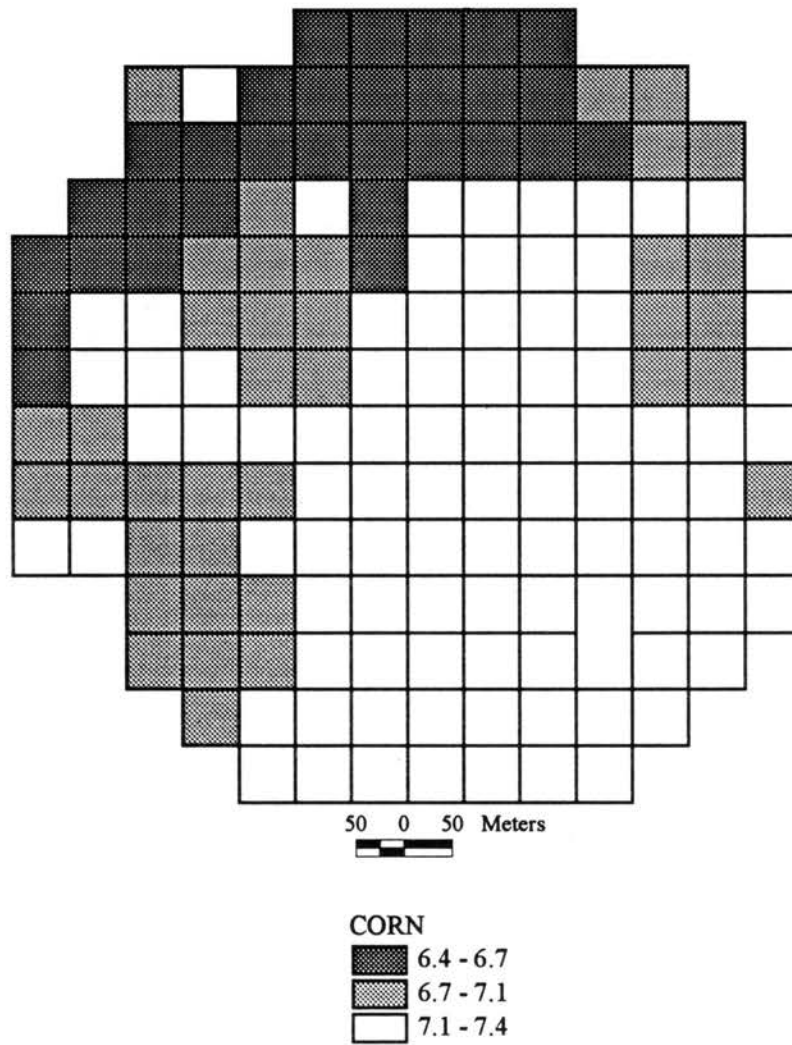


Figure 44. Map showing average annual corn yield (Mg ha^{-1}) estimated using EPIC-View for conventional tillage conditions.

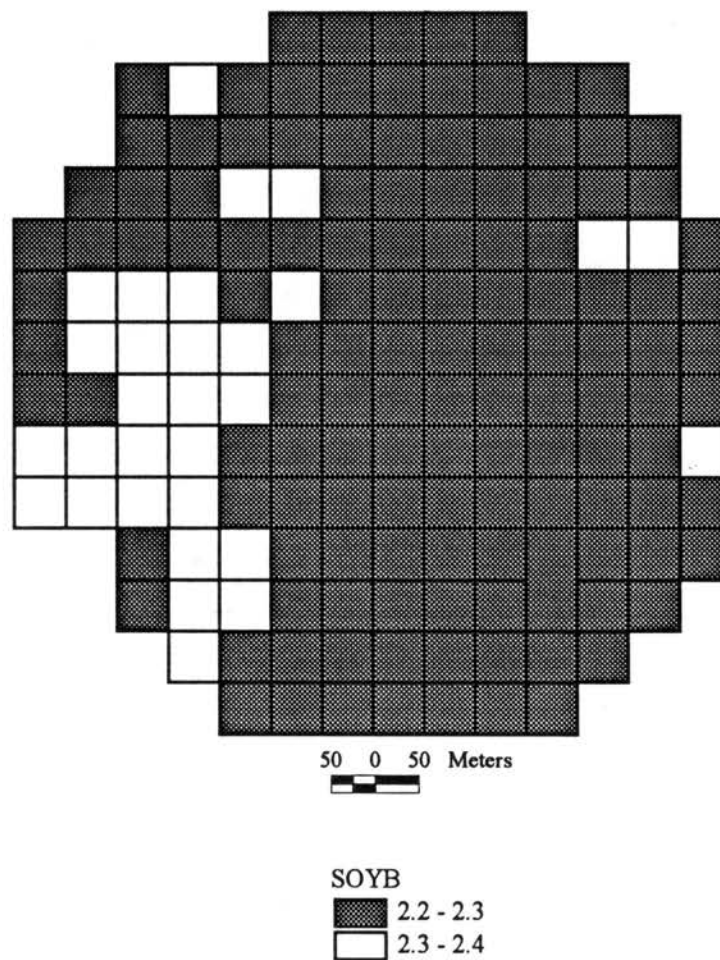


Figure 45. Map showing average annual soybean yield (Mg ha^{-1}) estimated using EPIC-View for conventional tillage conditions.

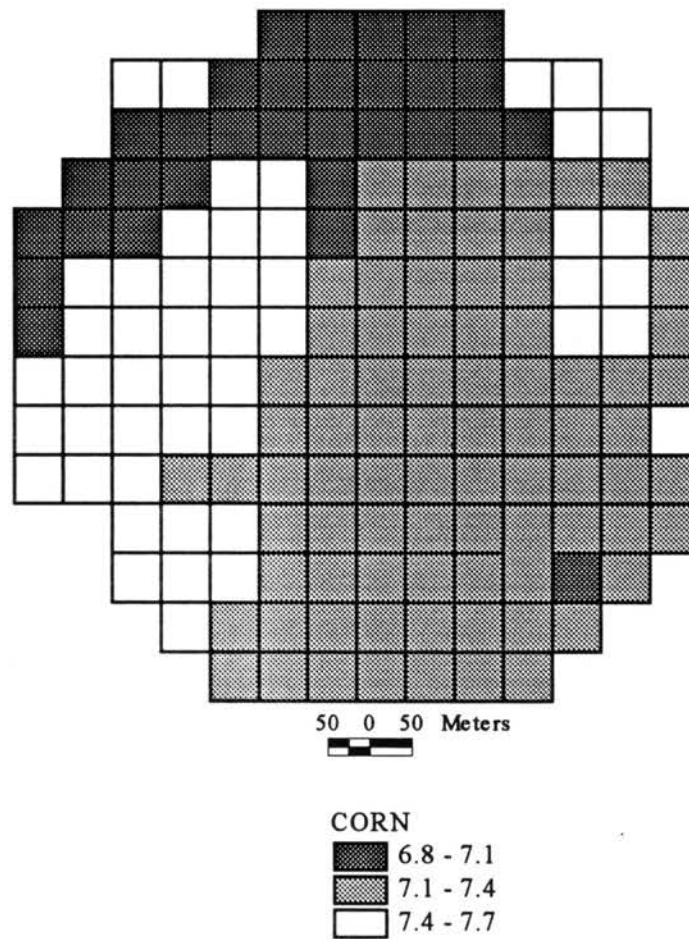


Figure 46. Map showing average annual corn yield (Mg ha⁻¹) estimated using EPIC-View for minimum tillage conditions.

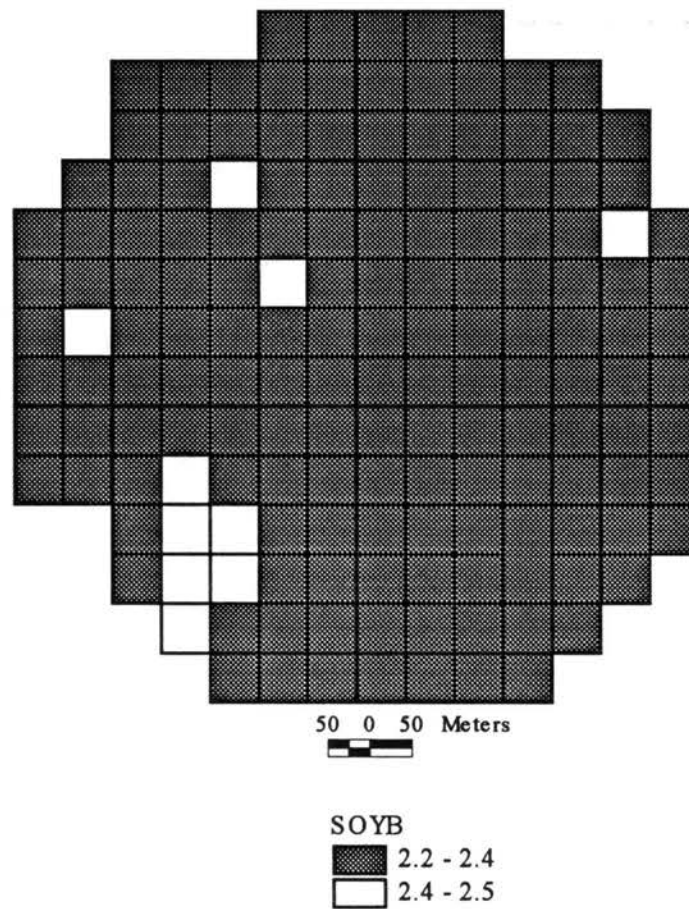


Figure 47. Map showing average annual soybean yield (Mg ha^{-1}) estimated using EPIC-View for minimum tillage conditions.

Table 7. Comparison of coefficient of variation (CV) for the different nitrogen variables under conventional and minimum tillage conditions.

Nitrogen variable (kg ha ⁻¹)	coefficient of variation	
	Conventional tillage	Minimum tillage
AVOL	0.14	0.03
HMN	0.32	0.34
MNN	0.13	0.11
NFIX	0.05	0.04
NITR	0.00	0.00
PRKN	0.07	0.06
SSFN	2.18	2.28
TNH3	0.02	0.03
TNO3	0.07	0.04
YNO3	0.48	0.53
YON	0.32	0.32
UNO3	0.03	0.03

conditions. A marked decrease in relative variability is observed with respect to volatilization losses (AVOL) under minimum tillage, suggesting that crop residue might be an important factor. However, a similar pattern was not observed for other losses, particularly nitrogen losses in surface and subsurface flows (PRKN and SSFN). Although the balance sheets for nitrogen calculated for the two scenarios show small differences for various nitrogen variables, considerable differences are seen for residual nitrogen (TNO₃ and TNH₃), sediment bound N (YON), and crop uptake of N. There is a 17.11% increase in residual ammonia in the soil profile under minimum tillage when compared to conventional tillage. This might be attributed to the decreased volatilization losses under minimum tillage. In the case of sediment bound loss of N in runoff, there is a 7.4% decrease under minimum tillage when compared to conventional tillage. This again might be explained by crop residue on the soil surface. Nevertheless, the overall N balance obtained under minimum tillage is better than the balance obtained under conventional tillage. This can be inferred from unexplained losses in the balance sheets computed by taking the difference between gains and losses of nitrogen. Unexplained losses of N averaged 9.55% and 4.2 % of the gain in N under conventional and minimum tillage respectively.

CHAPTER V

CONCLUSIONS

The system developed in the study allowed the integration of a comprehensive hydrologic-crop management model that can aid in selecting sustainable farm management practices. The use of GIS made possible the integration of diverse spatial data into a comprehensive database. This organized database allowed easy access and input to the model. Also, categorizing areas into HRUs facilitated differential simulation of dynamics across the field. This was based on the spatial characteristics of each HRU instead of selecting average values for the entire field. Visual output in the form of maps enabled the study of farm level response to inputs which could aid in better farm management.

The inclusion of a pull-down menu, fused with that of ArcView's, served as the link between the user and the system. This facilitated the execution of different Avenue and Visual Basic programs in an intuitive and user-friendly environment; thus enabling users to operate the system without an in-depth understanding of its individual components. It is important to realize that model adoption and user leads to subsequent understanding and familiarity with the model. The framework could also serve as a building block for inclusion of additional tools geared toward better data input and output visualization, such as 3-D.

Integrating ArcView with EPIC, a hydrologic-crop management model, enhanced understanding of the spatial variability of N transformation occurring within a crop field. As such, decision-making regarding spatially dependent activities such as nitrogen nutrient management would be facilitated by access to the type of maps, tables, and charts generated by this study. Similarly, other components of a sustainable agricultural system including irrigation management, crop management, soil management, and pest management, would be efficiently managed using similar models making farms more economically viable and ecologically sound. Furthermore, alternate management scenarios can be studied and evaluated by altering model input parameter values thereby functioning as a planning tool. With adequate model calibration and validation, the recommendations for best management practices should be based on long-term model runs in order to overcome variability in model input data. Also, this integrated system has potential for substantiating current management practices; thereby challenge blanket regulatory policies of federal agencies, or ward off frivolous litigation.

Additionally, the integrated system developed in this study could be a convenient tool for researchers interested in testing model performance, as datasets could be assembled in a short period of time. Thus, efficiency is improved as greater time could be spent evaluating the effect of estimated parameters on model output. The integrated system could also be an excellent teaching tool for explaining and illustrating different concepts of farm management, soil and water conservation, and systems applications in natural resources management and conservation.

However, the integrated system has its own limitations and drawbacks. The system does not take into account temporal variability in the processes simulated. System output is an annual average for each HRU. Although the model can output results at daily or monthly intervals, the system, owing to the simple database structure of ArcView, could not be designed to accommodate the display of the vast amount of data that would be generated. The capabilities of a relational database coupled with a higher level of programming language could accomplish this task.

Recommendations for Future Research

The integrated system possesses immense potential as a farm management tool. However, the system requires improvements. Run-time could be vastly improved if the attribute table of the HRU coverage could be extended to encompass the management practices. Again, this requires the use of a relational database to hold the different management operations performed on a zone of different HRUs. An expert system could also be designed to help a user navigate through the maze of different processes specific to the farm. Future research studies should address the effect of different sized modeling unit (HRU) on the model output. An offshoot of such a research venture could involve the effects of input data having different scales. Furthermore, research should be directed at acquiring site-specific data to validate and calibrate the model. The use of GPS technology in tandem with harvesters and soil samplers could be used to acquire such data, and elevate site-specific simulation to higher levels of accuracy.

LITERATURE CITED

- Allison, F.E. 1965. Evaluation of incoming and outgoing processes that affect soil nitrogen. *Agronomy* 10:573-606.
- Anderson, J.A. 1994. Integrating site-specific and public domain data in a geographic information system to facilitate agricultural producers' decision making. Masters' thesis, Oklahoma State University, Stillwater. OK.
- Antenucci, J.C., K. Brown, P.L. Croswell, M.J. Kevany, and H. Archer, eds. 1991. Geographic information system software. *In: Geographic Information Systems. A Guide to the Technology.* Chapman & Hall. p 301.
- Arnold, J.G., B.A. Engel, and R. Srinivasan. 1993. Continuous time, grid cell watershed model. *Application of Advanced Information Technologies: Effective Management of Natural Resources.* ASAE Publication 04-93. pp. 267-278.
- Arnold, J.G., J.R. Williams, A.D. Nicks, and N.B. Sammons. 1990. Basin scale simulation model for soil and water resources management. College Station, TX: Texas A&M Univ. Press
- Arnold, R.W. and C.A. Jones. 1987. Soil and climate effects upon crop production and nutrient use. *In Soil Fertility and Organic Matter as Critical Components of Production Systems.* Soil Sci. Soc. Am. Spec. Pub. no. 19. Soil Sci. Soc. Am., Madison, WI. pp 9-17
- Avery, T.E. and G.L. Berlin. 1985. Interpretation of aerial photographs. Burgess Pub. Co., Minneapolis, MN.
- Ball, G.L. 1994. Ecosystem modeling with GIS. *Environ. Manage.* 18:345-349.
- Beasley, D.B., L.F. Huggins, and E.J. Monke. 1980. ANSWERS: A model for watershed planning. *Trans ASAE.* 23(4):938-944.
- Benson, V.W. 1989. EPIC: A planning tool for soil and water conservation programs. *In: Proc. 1989 Summer Computer Simulation Conference, Society for Computer Simulation, San Diego, CA.* pp 18-721.
- Benson, V.W., O.W. Rice, P.T. Dyke, J.R. Williams, and C.A. Jones. 1989. Conservation impacts on crop productivity for life of a soil. *J. Soil and Water Conserv.* 44:600-604.
- Benson, V.W., W.A. Goldstein, D.L. Young, J.R. Williams, C.A. Jones, and J.R. Kiniry. 1990. Impacts of integrated cropping practices on nitrogen use and movement. *In: P. W. Unger, W.R. Jordan, T.V. Sneed, and R.W. Jensen, eds. Challenges in Dryland Agriculture - A Global Perspective.* Texas Agric. Exp., Sta., College Station, TX. pp 426-428.

- Binger, R.L. 1990. Comparison of the components used in several sediment yield models. *Trans. ASAE*. 33(4):1229-1238.
- Boswell, F.C., J.J. Meisinger, and N.L. Case. 1985. Production, marketing and use of nitrogen fertilizers. *In: O.P. Engelstad, ed. Fertilizer Technology and Use*. Soil Sci. Soc. of Am., Madison, WI. pp 229-292.
- Bracken, I. and C. Webster. 1990. Information technology in geography and planning including principles of GIS. Routledge, London.
- Cabelguenne, M., C.A. Jones, J.R. Marty, P.T. Dyke, and J.R. Williams. 1990. Calibration and validation of EPIC for crop rotations in southern France. *Agric. Syst.* 33:153-171.
- Calvin W.R., W.T. Dickinson, H. Ghadiri, and S.E. Jorgensen. 1988. Agricultural nonpoint source runoff and sediment yield water quality (NPSWQ) models: modeler's perspective. *In: Proc. International Symposium on Water Quality Modeling of Agricultural Nonpoint Sources: June 19-23, 1988, Utah State University - Logan, UT.*
- Carcamo, J.A., J. Alwang, and G.N. Norton. 1994. On-site economic evaluation of soil conservation practices in Honduras. *Agric. Econ.* 11:257-269.
- Chen, Z., D.E. Storm, M.D. Smolen, C.T. Haan, M.S. Gregory, and G.S. Sabbagh. 1994. Prioritizing nonpoint source phosphorus loading using a GRASS modeling system. *Water Resource Bull.* 30:589-603.
- Chrismer, N.R., D.F. Mezera, D.D. Moyer, B.J. Niemann, Jr., J.G. Sullivan, and A.P. Vonderohe. 1990. Soil erosion planning in Wisconsin: An application and evaluation of a multipurpose land information system, *In: H.J. Onsurd and D.W. Cook, eds. Geographic and Land Information Systems for Practicing Surveyors*. American Congress on Surveying and Mapping, Bethesda, MD. pp185-194.
- Cohen, S.Z., C. Eiden, and M.N. Lorber. 1984. Monitoring ground water for pesticides. *In: W. Garner, R.C. Honeycutt, and H.N. Nigg, eds. Evaluation of Pesticides in Ground Water*. ACS Symposium Series 315. American Chemical Society, Washington, D.C.
- Commoner, B. 1970. Threats to the integrity of the nitrogen cycle: Nitrogen compounds in soil, water, atmosphere and precipitation. *In: S.F. Singer, ed. The Changing Global Environment*. D. Reidel Publishers, Hingham, MA. pp 341-366.
- Coppock, J.T. 1988. The risks of software innovation: A case study of the Harvard Lab. *The Am. Cartographer* 15:291.
- Corwin, D.L. and J.D. Rhoades. 1988. The use of computer-assisted mapping techniques to delineate potential areas of salinity development in soils. *Hilgardia J. of Agric. Sci.* 56:18-32.
- Crane, P.J. and L.P. Herrington. 1992. GIS applications. A wide spectrum not without problems. *Photogrammetric Eng. and Remote Sens.* 8:1092-1094.
- Crosson, P. 1989. What is alternative agriculture? *Am. J. Altern. Agri.* 4:28-31.

- Curtis, T.G. 1992. The SWMM hydrologic model from CARD images. *In: Proc. 12th Annual ESRI User Conference, Redlands, CA.*
- Dangermond, J. 1991. Where is the technology leading us? *In: M. Heit and A. Shortreid, eds. GIS Applications in Natural Resources. GIS World, Inc., Ft. Collins, CO.*
- Davidson, J.M. 1989. Anticipating future research needs in response to an expanding clientele. 27th Annual Agronomic Administrators Round table. Personal communication.
- Deliman, P.N. and M.L. Wolfe. 1990. Assessing nonpoint pollution potential of surface waters using GIS. *In: R.E. Riggins, E.B. Jones, R. Singh, and P.A. Richard, eds. Watershed Planning and Analysis in Action. American Society of Civil Engineers, New York, N.Y. pp 191-200.*
- Dumesnil, D., ed. 1993. EPIC user's guide-draft. USDA-ARS, Grassland, Soil and Water Research Laboratory, Temple, TX.
- Dyke, P.T., C.A. Jones, J.W. Keeling, J.E. Matocha, and J.R. Williams. 1990. Calibration of a farming systems model for the Southern Coastal Plains and High Plains of Texas. Misc. Pub. 1696. Texas Agric. Exp. Sta., College Station, TX. pp 19.
- Environmental Systems Research Institute. 1994a. ArcView- The Geographic System for Everyone. Environmental Systems Research Institute, Inc., Redlands, CA.
- Environmental Systems Research Institute. 1994b. Avenue - Customization and Application Development for ArcView. Environmental Systems Research Institute, Inc., Redlands, CA.
- Environmental Systems Research Institute. 1995. Getting to Know Desktop GIS. Featuring ArcView: The Geographic Information Systems for Everyone. Environmental Systems Research Institute, Inc., Redlands, CA.
- Fedra, K. 1991. GIS and environmental modeling. *In: Proc. First International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling, Boulder, CO.*
- Gan, T.Y. 1990. Distributed hydrologic models based on remotely sensed data. *In: Proc. 23rd International Symposium on Remote Sensing of the Environment, Lansing, MI.*
- Gilliam, J.W., T.J. Logan, and F.E. Broadbent, 1985. Fertilizer Use in Relation to the Environment. *In: O.P. Engelstad, ed. Fertilizer Technology and Use. Soil Sci. Soc. of Am., Madison, WI. pp 561-589.*
- Godwin, D. C. and C. A. Jones. 1991. Nitrogen dynamics in soil-plant systems. *In: R.J. Hanks and J. T. Ritchie, eds. Fertilizer Technology and Use, 3rd ed. Soil Sci. Soc. of Am., Madison, WI.*
- Grayson, R.B. 1992. Physically based hydrologic modeling. 2. Is the concept realistic? *Water Resources 26(10):2659-2666.*

- Haan, C.T., B.J. Barfield, and J.C. Hayes. 1994. Hydrologic modeling. *In: Design Hydrology and Sedimentology for Small Catchments*. Academic Press, San Diego, CA. 588 p.
- Haan, C.T., H.P. Johnson, and D.L. Brakensiek, eds. 1982. Hydrologic modeling of small watersheds. American Society of Agricultural Engineers. St. Joseph, MI. 533 p.
- Hanks, J., and J.T. Ritchie, eds. 1991. Modeling Plant and Soil Systems. *Agronomy* 31. American Soc. of Agronomy, Madison, WI. p 545.
- Harper, L.A., R.R., Sharpe, G.W. Langdale, and J.E. Giddens. 1987. Nitrogen cycling in a wheat crop: soil, plant, and aerial nitrogen transport. *Agron J.* 79:965-973.
- Hill, R.W. 1991. Irrigation Scheduling. *In: R.J. Hanks and J. T. Ritchie, eds. Modeling Plant and Soil Systems. Agronomy* 31. American Society of Agronomy, Inc. Madison, WI. pp 491-509.
- Hodge, W., M. Larson, and W. Goran. 1988. Linking the ARMSED watershed process model with GRASS geographic information systems. *Proc. Modeling Agricultural, Forest and Rangeland Hydrology Conference*. ASAE, St. Joseph, MI. pp. 501-510.
- Jankowski, P. and G. Haddock. 1993. Integrated nonpoint-source pollution modeling system. *In: Second International Conference/Workshop on Integrating GIS and Environmental Modeling*. Breckenridge, CO. 26-30 Sept. 1993.
- Jayawardana, P.M. and G.J.E. Hill. 1990. The development of remote sensing and geographical information systems based models for predicting landslide and gully erosion hazard areas in Australia. *In: Proc. 23rd Int. Symposium on Remote Sensing of Environment, Vol. 1*. Environmental Research Institute of Michigan, Ann Arbor, MI. pp 325-334.
- Johnston, C.A., N.E. Detenbeck, J.P. Bonde, and G.J. Niemi. 1988. Geographic information systems for cumulative impact assessment. *Photogrammetric Eng. and Remote Sens.* 11:1609-1615.
- Jones, C.A., A.N. Sharpley, and J.R. Williams. 1984. A simplified soil and plant phosphorus model. III. Testing. *Soil Sci. Soc. Am. J.* 48:810-813.
- Jones, C.A., A.N. Sharpley, and J.R. Williams. 1989. Modeling phosphorus dynamics in soil-plant systems. *In: R.J. Hanks and J.T. Ritchie, eds. Modeling Soil and Plant Systems*. No. 31. Am. Soc. Agron. Monograph., Madison, WI. pp 323-337.
- Jones, J.W. and J.T. Ritchie. 1991. Crop growth models. *In: G.J. Hoffman, T.A. Howell, and K.H. Solomon, eds. Management of Farm Irrigation Systems*. Amer. Soc. Agricultural Engineering, St. Joseph, MI. pp.63-89.
- Kalita, P.K., R.S. Kanwar, and J.H. Bischoff. 1992. Using the ADAPT model to simulate water table management effects in groundwater quality. *ASAE Paper # 92-2124*. St. Joseph, MI.

- Keller, C.P. 1991. Time-space analysis and GIS. *In*: M. Heit and A. Shortreid, eds. GIS Applications in Natural Resources. GIS World, Inc., Fort Collins, CO. pp 141-144.
- Knisel, W.G., Jr. 1980. A Field Scale Model for Chemicals, Runoff and Erosion from Agricultural Management Systems. Cons. Res. Rpt. No. 26. USDA. 640.
- Krishna, J.H., G.F. Arkin, J.R. Williams, and J.R. Mulkey. 1987. Simulating furrow-dike impacts on runoff and sorghum yields. *Trans. ASAE* 39:143-147.
- Lacewell, R.D., J.G. Lee, C.W. Wendt, R.J. Lascano, and J.W. Keeling. 1990. Implications of alternative dryland crop rotation: Texas High Plains. *In*: P.W. Unger, W.R. Jordan, T.V. Sneed, and R.W. Jensen, eds. Challenges in Dryland Agriculture - A Global Perspective. Texas Agric. Exp., Sta., College Station, TX.
- Laflen, J. M., L.J. Lane, and G.R. Foster. 1991. WEPP: A new generation of erosion prediction technology. *J. Soil and Water Conserv.* 46(1):34-37.
- Langdale, G.W., R.A. Leonard, W.G. Flemming, and W.A. Jackson. 1979. Nitrogen and chloride movement in small upland piedmont watersheds: II. Nitrogen and chloride transport in runoff. *J. Environ. Qual.* 8:57-63.
- Legg, J.O. and J.J. Meisinger. 1982. Soil nitrogen budgets. *In*: F.J. Stevenson, ed. Nitrogen in Agricultural Soils. *Agronomy* 22:503-565.
- Lee, J. G. and R. D. Lacewell. 1989. Bio-economic modeling for conservation planning. *In*: 1989 Summer Computer Simulation Conference. Society for Computer Simulation, San Diego, CA. pp. 727-731.
- Lo, K.F.A. 1994. Quantifying soil erosion for the Shihmen reservoir watershed, Taiwan. *Agric. Syst.*, 45:105-116.
- Maidment D.R. 1993. Environmental modeling within GIS. *In*: Proc. Second International Conference/Workshop on Integrating GIS and Environmental modeling. Breckridge, CO. Sept. 2, 1993.
- Maidment, D.R., D. Djokic, and K.G. Lawrence. 1989. Hydrologic modeling on a triangulated irregular network. *Trans. Am. Geophys. Union* 70:1091.
- Mark, D.M. and M.D. Gould. 1991. Interacting with geographic information: a commentary. *Photogrammetric Eng. and Remote Sens.* 57(11):1427-1430.
- Martin, S.M., M.A. Nearing, and R.R. Bruce. 1993. An evaluation of the EPIC model for soybeans grown in Southern Piedmont soils. *Trans ASAE.* 36(5):1327-1331.
- Mellerowicz, K.T., H.W. Rees, T.L. Chow, and I. Ghanem. 1994. Soil Conservation planning at the watershed level using the Universal Soil Loss Equation with GIS and microcomputer technologies: A case study. *J. Soil and Water Conserv.* 49:194-200.
- Moore, I.D. and R.B. Grayson. 1991. Terrain-based catchment partitioning and runoff prediction using vector elevation data. *Water Resour. Res.* 6:1177-1191.

- Moore, I.D., E.M. O'Loughlin, and G.S. Burch. 1988. A contour-based topographic model for hydrological and ecological applications. *Earth Surf. Processes Landform* 13:305-320.
- Muzik, I. 1988. Application of GIS to SCS procedure for design flood hydrographs. *In: Modeling Agricultural, Forest and Rangeland Hydrology. Proc. ASAE International Symposium. American Society of Agricultural Engineers, St. Joseph, MI. pp 494-500.*
- National Research Council. 1989. Report on Alternative Agriculture. National Academy Press. Washington, D.C.
- Neher, D. 1992. Ecological sustainability in agricultural systems: Definitions and measurement. *J. Sustainable Agri.* 2:51-61.
- Nonpoint Source Task Force. 1984. Nonpoint source policy. U.S. Environmental Protection Agency, Washington, D.C.
- O'Hare, M.D. Curry, S. Atkinson, S. Lee, and L. Center. 1985. Contamination of groundwater in the contiguous United States from usage of agricultural chemicals. Environmental and Groundwater Institute, Norman, OK.
- Parent, P.J. 1988. Geographic information systems: Evolution, academic involvement and issues arising from the proliferation of information. Masters' thesis, University of California, Santa Barbara.
- Parr, J.F., B.A. Stewart, S.B. Hornick, and R.P. Singh. 1990. Improving the sustainability of dryland farming systems: A global perspective. *In: R.P. Singh, J.F. Parr, and B.A. Stewart, eds. Advances in Soil Science, Vol. 13. Dryland Agriculture: Strategies for Sustainability. Speinger Verlag, New York, N.Y.*
- Patron, W.J., J.W.B. Stewart, and C.V. Cole. 1988. Dynamics of C, N, P and S in grassland soils: A model. *Biogeochemistry* 5:109-131.
- Paul, E.A., and N.G. Juma. 1981. Mineralization and immobilization of nitrogen by microorganisms. *In: F.E. Clark and T. Rosswall, eds. Terrestrial Nitrogen Cycles. Ecol. Bull. (Stockholm) 33. Swedish Acad. of Sci., Stockholm. pp 179-195.*
- Pesek, J. 1994. Historical perspective. *In: J.L. Hatfield and D.L. Karlen, eds. Sustainable Agriculture Systems. Lewis Publishers. Boca Raton, FL. pp 1-19.*
- Phillips, D.L., P.D. Hardin, V.W. Benson, and J.V. Baglio. 1993. Nonpoint source pollution impacts of alternate agricultural management practices in Illinois: A simulation study. *J. Soil and Water Conserv.* 48(5):449-457.
- Power, J.F., J.J. Bond, F.M. Sandoval, and W.O. Willis. 1974. Nitrification in Paleocene shales. *Science* 1984:1077-1079.
- Prasad, R. and J.F. Power. 1991. Crop Residue Management - Literature Review. *Adv. in Soil Sci.* 15:205-251.

- Putman, J.W. and P.T. Dyke. 1986. An analysis of the impact of Erosion on Productivity and Fertilizer. Appraisal Task 5A. 1985 Resource Conservation Act. US Department of Agriculture, Economic Research Service, Washington, D.C. September 1986.
- Putman, J.W. and P.T. Dyke. 1987. The erosion -productivity impact calculator as formulated for the Resource Conservation Act Appraisal. US Department of Agriculture, Economic Research Service, Staff Report No. AGES861204.
- Ragan, R.M. 1991, ed. A geographic information system to support state-wide hydrologic and nonpoint pollution modeling, Dept. of Civil Engineering, University of Maryland, 284 p.
- Ramanarayanan, T.S. 1994. Performance of transport models in predicting nitrate in runoff flow in high water table areas. Presented at the 1994 International Summer Meeting, Paper No. 94-2152, ASAE, St. Joseph, MI.
- Richter, S. and L. Tank. 1991. Best management practices. Cooperative Partners 4(3):6-11.
- Robertson, T., C. Rosenzweig, V.W. Benson, and J.R. Williams. 1990. Projected impacts of carbon dioxide and climate change in the Great Plains. *In*: P. W. Unger, W.R. Jordan, T.V. Sneed, and R.W. Jensen, eds. Challenges in Dryland Agriculture - A Global Perspective. Texas Agric. Exp., Sta., College Station, TX. pp 675-677.
- Robertson, T., V.W. Benson, J.R. Williams, J.R. Kiniry, and C.A. Jones. 1987. Impacts of climate change on yields and erosion for selected crops in the southern United States. *In*: M. Meo, ed. Proc. Symposium on Climate Change in the Southern United States: Future Impacts and Present Policy. Science and Public Policy Program, University of Oklahoma. pp 73-78.
- Ruttan, V.W. 1990. Alternative agriculture sustainability is not enough. *In* : Alternative Agriculture Scientists Review. Spec. Pub. No. 16. Council for Agricultural Science and Technology, Ames, IA. pp130-134.
- Sabbagh, G.J., S. Geleta, R.L. Elliott, J.R. Williams, and R.H. Griggs. 1991. Modification of EPIC to simulate pesticide activities: EPIC-PST. Trans. ASAE 34:1683-1692.
- Sadler, E.J. and N.C. Turner. 1994. Water relationship in a sustainable agriculture system. *In*: J.L. Hatfield and D.L. Karlen, eds. Sustainable Agricultural Systems. Lewis Publisher, Boca Raton, FL. pp 21-39.
- Schaller, N. 1990. Mainstreaming low input agriculture. J. Soil and Water Conserv. 45:9-12.
- Schepers, J.S. and R.H. Fox. 1989. Estimation of N budgets for crops. Chp. VIII. *In*: R.F. Follett and R.A. Olson, eds. Nitrogen Management for Groundwater Protection. Elsevier Sci. Pub., Amsterdam. pp 221-246.
- Segarra, E. 1989. Optimizing nitrogen use in cotton production. *In* Proc. 1989 Summer Computer Simulation Conference, Society for Computer Simulation, San Diego, CA. pp 722-726.

- Senanayake, R. 1991. Sustainable agriculture. Definitions and parameters for measurement. *J. Sust. Agric.* 1:7-28.
- Shaffer, N.J., S.C. Gupta, D.R. Linden, J.A.E. Molina, C.E. Clapp, and W.E. Larson. 1983. Nitrogen, Tillage, residue management model (NTRM). *In: W.K. Lauenroth, G.V. Skogerboe, and M. Flug, eds. Analysis of Ecological Systems: State of the Art in Ecological Models.* Elsevier Publ., New York, N.Y. pp. 525-544.
- Shanholtz, V.O., C.J. Desai, N. Zhang, and J.W. Kleene, C.D. Metz, and J.M. Flagg. 1990. Hydrologic/water quality modeling in a GIS environment. ASAE Paper No. 90-3033. ASAE, St. Joseph, MI.
- Sharpley, A.N. and J.R. Williams, eds. 1990. EPIC-Erosion/Productivity Impact Calculator. I. Model Documentation. U.S. Department of Agriculture Tech. Bull. 1768. 127 p.
- Sharpley, A.N., C.A. Jones, and J.R. Williams. 1990. The nutrient component of EPIC. *In: EPIC-Erosion/Productivity Impact Calculator. I. Model Documentation.* U.S. Department of Agriculture Tech. Bull. 1768.
- Silfex. A.T., G.J. Kinn, and J.M. Hassett. 1987. A geographic information system utilizing the triangulated irregular network as a basis for hydrologic modeling. *In: Proc. Auto-Carto 8, Baltimore, MD.*
- Smith, J.W., Z.I. Muhammad, and E.L. Lim. 1990. Utility of a GIS in watershed management: A case study. *In: R.E. Riggins, E.B. Jones, R. Singh, and P.A. Richard, eds. Watershed Planning and Analysis in Action.* American Society of Civil Engineers, N.Y. pp 201-210.
- Smith, S.J., A.N. Sharpley, J.R. Williams, and C.A. Jones. 1986. Prediction of nutrient transport in agricultural runoff using computed runoff values. *Trans. XIII Cong. Intl. Soc. Sci., 13-20 August 1986. Hamburg Int. Soc. Soil Sci., Hamburg.* pp 1604-1605.
- Srinivasan, R. and J.G. Arnold. 1994. Integration of a basin-scale water quality model with GIS. *Water Resources Bull.* 30:453-462.
- Srinivasan, R. and B.A. Engel. 1994. A spatial decision support system for a assessing agricultural nonpoint source pollution. *Water Resources Bull.* 30:441-452.
- Steube, M.M. and D.M. Johnston. 1990. Runoff volume estimation using GIS techniques. *Water Resources Bull.* 26:611-620.
- Stinner, B.R. and J.M. Blair. 1990. Ecological and agronomic characteristics of innovative cropping systems. *In: C.A. Edwards, R. Lal, P. Madden, R.H. Miller, and G. House, eds. Sustainable Agricultural Systems.* Soil and Water Conserv. Soc., IA. pp 123-140.
- Stockle, C.O., R.I. Papendick, K.E. Saxton, G.S. Campbell, and F.K. van Evert. 1994. Framework for evaluating the sustainability of agricultural production systems. *Am. J. Alternative Agri.* 9(1&2):45-50.

- Sumner, M.E. and W.P. Miller. 1992. Soil crusting in relation to global soil degradation. *Am. J. Alternative Agri.* 7(1&2):56-62.
- Tim, U.S. and R. Jolly. 1994. Evaluating agricultural nonpoint source pollution using integrated geographic information systems and hydrologic/water quality model. *J. Environ. Qual.* 23:25-35.
- Tim, U.S., S. Mostaghimi, and V.O. Shanholtz. 1992. Identification of critical nonpoint pollution source areas using geographic information systems and water quality modeling. *Water Resources Bulletin.* 28(5):877-887.
- Tomlinson, R.F. 1988. The impact of transition from digital cartographic representation. *The Am. Cartographer* 15:252-257.
- Turton, D.J., C.T. Haan, and E.L. Miller. 1988. Modeling water quality of a small watershed in southeastern Oklahoma: The hydrologic component. *In: Modeling Agricultural, Forest and Rangeland Hydrology. Proc. ASAE International Symposium.* American Society of Agricultural Engineers, St. Joseph, MI. pp 95-107.
- Tweeten, L. 1994. Twenty-first Century Agriculture: Besieged, Productive and Profitable. *Public Policy Digest.* Vol. 1 no. 2. AFBF Public Policy Division.
- U.S. Department of Agriculture-SCS. 1972. National engineering handbook. Hydrology section 4. Govt. Print Office, Washington, D.C.
- USDA-SCS. 1973. Soil Survey of Caddo County, Oklahoma. U.S. Government Printing Office, Washington, D.C. 20402.
- U.S. Department of Agriculture. 1989. Use computers for farm management. January 1986-March 1989. Quick Bibliography Series. NAL.BIBL. QB. 89-102.
- U.S. Government Accounting Office. 1990. Alternative Agriculture: Incentives and Opinions, GAO/PEMD-90-12, U.S. Gov. Printing Office, Washington, D.C.
- Valenzuela, C.R. 1991. Spatial databases. *In: A.S. Belward and C.R. Valenzuela, eds. Remote Sensing and Geographical Information Systems for Resource Management in Developing Countries.* Kluwer Academic Publishers, Netherlands. pp 311-333.
- Varis, O. 1994. Water quality models: Tools for the analysis of data, knowledge, and decisions. *Water Sci Technology.* 30(2):13-19.
- Vieux, B.E. 1991. Geographic information systems and nonpoint source water quality and quantity modeling. *Hydrological Proc.* 5:101-113.
- Walsch, S. 1985. Geographic information system for natural resource management. *J. Soil and Water Conserv.* 40:203-208.
- Warwick, J.J. and S.J. Haness. 1994. Efficacy of ARC/INFO GIS application to hydrologic modeling. *J. Water Resour. Plann. Manage.* 120:366-381.

- Wilkerson, G.G., J.W. Jones, K.J. Boote, K.T. Ingram, and J.W. Mishoe. 1983. Modeling soybean growth for management. *Trans. ASAE*. 26(1):63-73.
- Williams, J.R. 1981. National Soil-Erosion-Soil Productivity Research Planning Committee, USDA, ARS. 1981. Soil erosion effects on soil productivity: A research perspective. *J. Soil and Water Conserv.* 36:82-90.
- Williams, J.R., C.A. Jones, J.R. Kiniry, and D.A. Spanel. 1989. The EPIC crop growth model. *Trans ASAE* 32:497-511.
- Williams, J.R., C.W. Richardson, R.H. Griggs. 1992. The weather factor. incorporating weather variance into computer simulation. *Weed Technology* 6(3):731-735.
- Williams, J.R. P.T. Dyke, W.W. Fuchs, V.W. Benson, O.W. Rice, and E.D. Taylor. 1990. EPIC - Erosion/Productivity Impact Calculator 2. User Manual. USDA Agricultural Research Service. Tech Bull No. 1768.
- Wolfe, M.L. and C.M.U. Neale. 1988. Input development for a distributed parameter hydrologic model (FESHM). *In: Modeling Agricultural, Forest and Rangeland Hydrology. Proc. ASAE International Symposium. American Society of Agricultural Engineers, St. Joseph, MI.* pp 462-469.
- Young, R.A., M.A. Otterby, and A. Roos. 1982. A technique for evaluating feedlot pollution potential. *J. Soil and Water Conservation* 37:21-23.

APPENDICES

APPENDIX I.a

Sample EPIC Input Dataset

! Comment starts with '!'. Comments are not allowed in the input dataset. They are put here for clarity.

! Title

CADDO_COUNTY,_OK,_SHERRY/BOTCHLET

1:23 31mar96

BOTCH.DAT PREPARED BY MAHESH.RAO DT. 2/21/96

Wea: 22 OK WEATHERFORD wi: 22 OK WEATHERFORD

! Parameters defining program control codes, general data, and water erosion data.

5 91 1 1 51 0 0 1 1 0 1 0 0 0 0 0
 .34 67.0 1.0 30.35 413.4 .0
 .8 100.0 350.0 .0 .100
 81.2 .0200 1.00 2.

! Weather data and wind erosion data begins.

8.0 .0 .0
 8.97 12.12 17.08 23.13 27.43 32.42 35.26 34.67 30.12 24.27 16.01 10.73
 -3.99 -1.34 2.62 8.83 13.73 18.75 21.13 20.28 16.05 9.86 2.87 -1.97
 7.53 7.34 7.14 5.66 4.73 4.09 3.58 3.63 5.02 5.48 6.23 6.68
 5.82 5.46 5.54 5.12 4.16 3.32 2.44 2.70 4.23 4.86 5.29 5.21
 18.9 24.9 40.2 51.3 126.2 102.8 64.2 71.6 72.0 70.7 33.5 19.2
 7.1 7.6 9.4 10.7 21.3 20.3 14.5 15.2 19.3 22.1 10.2 8.4
 .89 .81 .66 .97 2.81 3.24 1.39 1.57 2.60 2.01 1.25 1.77
 .080 .100 .130 .150 .220 .200 .150 .180 .140 .110 .100 .090
 .280 .360 .290 .370 .410 .370 .340 .250 .310 .350 .350 .240
 3.10 3.92 4.80 5.77 8.42 7.23 5.74 6.00 5.06 4.49 4.00 3.28
 7.6 7.1 19.6 28.2 36.3 49.3 41.4 30.2 27.9 18.8 21.3 13.0
 257. 327. 420. 514. 558. 636. 629. 586. 498. 377. 296. 244.
 .63 .63 .50 .53 .64 .60 .55 .54 .56 .56 .56 .61
 .00 .00 .00 .00
 .50 .00 .00
 5.72 6.00 6.76 6.60 6.35 6.09 5.14 5.01 5.23 5.32 5.42 5.62
 16.0 14.0 11.0 10.0 7.0 4.0 3.0 3.0 6.0 9.0 12.0 14.0
 8.0 9.0 7.0 7.0 6.0 3.0 3.0 4.0 6.0 7.0 8.0 8.0
 5.0 6.0 5.0 5.0 4.0 3.0 4.0 5.0 6.0 5.0 4.0 5.0
 3.0 3.0 4.0 5.0 4.0 3.0 4.0 4.0 5.0 3.0 3.0 3.0
 3.0 4.0 4.0 5.0 6.0 5.0 7.0 6.0 5.0 3.0 3.0 2.0
 1.0 3.0 3.0 3.0 4.0 5.0 5.0 4.0 3.0 2.0 2.0 1.0
 2.0 4.0 4.0 5.0 7.0 9.0 9.0 8.0 7.0 3.0 3.0 2.0
 4.0 7.0 8.0 9.0 13.0 18.0 14.0 14.0 13.0 10.0 7.0 6.0
 18.0 16.0 19.0 20.0 27.0 31.0 27.0 28.0 28.0 27.0 20.0 18.0
 12.0 9.0 9.0 9.0 8.0 10.0 13.0 12.0 9.0 13.0 13.0 12.0
 7.0 5.0 4.0 4.0 3.0 3.0 5.0 5.0 4.0 5.0 7.0 6.0
 4.0 3.0 3.0 2.0 2.0 1.0 2.0 2.0 2.0 2.0 3.0 3.0
 3.0 3.0 3.0 3.0 2.0 1.0 1.0 1.0 1.0 2.0 3.0 4.0
 2.0 3.0 3.0 2.0 1.0 1.0 1.0 1.0 1.0 2.0 3.0 4.0
 4.0 5.0 5.0 4.0 3.0 1.0 1.0 1.0 2.0 2.0 4.0 6.0 8.0 8.0 8.0 5.0 3.0 2.0 1.0 1.0
 2.0 5.0 6.0 7.0 .44 0. .00 .00 .00 .00 .00 0. 0.

! Soil data begins.

.010	1.016	2.032
1.693	1.693	1.688
.036	.036	.054
.101	.101	.120
93.5	93.5	84.2
3.7	3.7	8.8
6.2	6.2	6.2
.44	.44	.15
3.9	3.9	5.0
.5	.5	.6

1.70	1.70	1.71
------	------	------

115.18	115.18	34.65
--------	--------	-------

! Management data options.

1	11	3	0	0	0	0	0	0	0	0
.85	.00	600.00	.00	20.00	.00	.00	.0	.00	.00	.00

! Management practices begin.

3	25	33								
4	8	71	51	448.36	50.80					
4	10	2	2	1500.00	.85					
5	12	19					.25			
6	1	71	62	112.09	50.80			.50		
8	15	9		9999.00				.95		
9	15	51						1.00		
9	15	41								
9	16	28								

APPENDIX I.b

Sample UTIL Batch File

```
! Comment starts with '!'. ! Comments are allowed in this file.
!
! This is a util batch file.
! UTIL command to load a soil file.
@c:\mahesh\epic5300\util\KoC2.utl
! ----- Spatial parameters -----
!
! Different variables and their values that need to be loaded.
!
ELEV 18.89
S 0.048
WSA 0.3364
!
! ----- Management Practices -----
! Each Management Operation has a set of variables associated with it.
!
MON(1) 3
DAY(1) 25
COD(1) 33
!
MON(2) 4
DAY(2) 8
COD(2) 71
FN(2) 51
FAP(2) 448.36
FDP(2) 50.80
!
MON(3) 4
DAY(3) 10
COD(3) 2
CRP(3) 2
GRZ(3) 2
PHU(3) 1500
WSF(3) 0.85
!
MON(4) 5
DAY(4) 12
COD(4) 19
HUSC(4) .25
!
MON(5) 6
DAY(5) 1
COD(5) 71
FN(5) 62
FAP(5) 112.09
FDP(5) 50.80
```

HUSC(5) .50
!
MON(6) 8
DAY(6) 15
COD(6) 9
WSF(6) 9999.00
HUSC(6) 0.95
!
MON(7) 9
DAY(7) 15
COD(7) 51
HUSC(7) 1.00
!
MON(8) 9
DAY(8) 15
COD(8) 41
!
MON(9) 9
DAY(9) 16
COD(9) 28
!

APPENDIX II.a

EPIC-View: Program Code for Weather Data Tool

```
' Epic.WeatherData
' This Avenue script loads weather file into constant datafile by providing
' a weather data entry user interface.
' Mahesh Rao. Dated 5/16/96
'
command = _exeDir+"weather.exe"
system.execute(command)
' Enable Constant Data and Soil Data menu options.
_constEnableFlag = 1
_soilEnableFlag = True

'Weather.frm
' This VB fom allows a user to enter the latitudes and longitudes of the field or specify the
' path of the weather file. This information is stored in "const.utl" file.
```

Option Explicit

```
Dim cellDatasetDir As String
Dim epicDir As String
Dim soilDir As String
Dim epicOutputDir As String
Dim exeDir As String
Dim fileNum As Integer
Dim fileName As String
Dim Latitude As Single
Dim Longitude As Single
Dim validateText As String 'To use as a buffer for validation.

Private Sub cmdCancel_Click()
    Dim resp As Integer
    resp = MsgBox("Do you wish to close?", vbYesNo + vbCritical + vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        End
    End If
End Sub

Private Sub cmdOk_Click()
    Dim resp As Integer
    'Store values if a user chooses to.
    resp = MsgBox("Do you wish to store?", vbYesNo + vbCritical + vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        If optEPICFile(0).Value = True Then
            If ((txtLat.Text <> "") And (txtLong.Text <> "")) Then
                fileNum = FreeFile
                Open cellDatasetDir + "const.utl" For Output As fileNum
                Print #fileNum, "LOCWEAT " + txtLat.Text + " " + txtLong.Text
                Print #fileNum, "YLT " + txtLat.Text
                Close fileNum
            End If
        End If
    End If
End Sub
```

```

        End
    End If
    ElseIf optMyFile(1).Value = True Then
        If (txtPath.Text <> "") Then
            fileNum = FreeFile
            Open cellDatasetDir + "const.utl" For Output As fileNum
            Print #fileNum, "@" + txtPath.Text
            Close fileNum
        End
    End If
End If
End Sub

Private Sub Form_Load()
    'Get directory paths.
    fileNum = FreeFile
    Open "c:\EVPPaths.txt" For Input As fileNum
    Input #fileNum, cellDatasetDir, epicOutputDir, soilDir, epicDir, exeDir
    Close fileNum
End Sub

Private Sub optEPICFile_Click(Index As Integer)
    txtLat.Visible = True
    txtLong.Visible = True
    txtPath.Visible = False
    lblLat.Visible = True
    lblLong.Visible = True
    lblPath.Visible = False
End Sub

Private Sub optMyFile_Click(Index As Integer)
    txtLat.Visible = False
    txtLong.Visible = False
    txtPath.Visible = True
    lblLat.Visible = False
    lblLong.Visible = False
    lblPath.Visible = True
End Sub

' ValidateData
' Date: June 12, 1996
'-----
' validateData Subroutine:
' - Validates the value entered by a user so
' - that it lies in the range of real numbers.
'-----

Public Sub validateData()
    Dim length As Integer
    Dim resp As Integer
    Dim start As Integer
    Dim alreadyDecimal As Integer

```

```

Dim chr As String
chr = 0
length = 1
alreadyDecimal = 0

Do
  chr = Mid(validateText, length, 1)
  ' If character other than a number or a decimal.
  If (chr <> "." And chr <> "0" And chr <> "1" And chr <> "2" And chr <> "3" And chr <> "4" _
    And chr <> "5" And chr <> "6" And chr <> "7" And chr <> "8" And chr <> "9") Then
    Beep
    If (length = 1) Then
      start = 2
    Else
      start = 1
    End If
    If (Len(validateText) > 0) Then
      validateText = Mid(validateText, start, Len(validateText) - 1)
    End If
    Exit Do
  ' If character is a decimal.
  ElseIf (chr = "." And alreadyDecimal = 0) Then
    alreadyDecimal = 1
  ' If character is a second decimal point.
  ElseIf (chr = "." And alreadyDecimal = 1) Then
    Beep
    If (length = 1) Then
      start = 2
    Else
      start = 1
    End If
    If (Len(validateText) > 0) Then
      validateText = Mid(validateText, start, Len(validateText) - 1)
    End If
    Exit Do
  End If
  length = length + 1
Loop While (length <= Len(validateText))
End Sub

Private Sub txtLat_Change()
  validateText = txtLat.Text
  validateData
  txtLat.Text = validateText
End Sub

Private Sub txtLong_Change()
  validateText = txtLong.Text
  validateData
  txtLong.Text = validateText
End Sub

```

APPENDIX II.b

EPIC-View: Program Code for Soil Data Tool

' **Epic.SoilData**

' This Avenue script loads soil and curve number into constant datafile by invoking
 ' a soil data entry user interface. This menu option is enabled only
 ' if a user does not has his own soil files.
 ' Mahesh Rao. Dated 5/18/96
 '

```
command = _exeDir+"soil.exe"
system.execute(command)
_constEnableFlag = 2
_soilEnableFlag = False
```

' **Soil.frm**

' This form allows a user to select a soil which becomes generic for the whole field and
 ' also select the run off curve number. This information is stored in the "const.utl" file.

Option Explicit

```
Dim soilNames(1000) As String
Dim soilCodes(1000) As Integer
Dim landUse(20) As String
Dim coverTrtmnt(5) As String
Dim hydCondition(5) As String
Dim hydSoilGrp(5) As String
Dim curveNumber(0 To 11, 0 To 3, 0 To 3, 0 To 4) As Integer
Dim fileNum As Integer
Dim cellDatasetDir As String
Dim epicOutputDir As String
Dim soilDir As String
Dim epicDir As String
Dim ctr1 As Integer
Dim ctr2 As Integer
Dim ctr3 As Integer
Dim ctr4 As Integer
```

Private Sub cboCoverTrtmnt_Change()

```
Dim Counter As Integer
Counter = 0
'Get the index for selected cover treatment.
Do
  If (coverTrtmnt(Counter) = cboCoverTrtmnt.Text) Then
    Exit Do
  End If
  Counter = Counter + 1
Loop While (Counter < 5)
If Counter < 5 Then
  ctr2 = Counter
  'Set curve number.
```

```
    txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
End If
End Sub
```

```
Private Sub cboCoverTrtmnt_Click()
    'Get the index for selected cover treatment.
    ctr2 = cboCoverTrtmnt.ListIndex
    'Set curve number.
    txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
End Sub
```

```
Private Sub cboHydCondition_Change()
    Dim Counter As Integer
    Counter = 0
    'Get the index for selected hydrologic condition.
    Do
        If (hydCondition(Counter) = cboHydCondition.Text) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While (Counter < 5)
    If Counter < 5 Then
        ctr3 = Counter
        'Set the curve number.
        txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
    End If
End Sub
```

```
Private Sub cboHydCondition_Click()
    'Get the index for selected hydrologic condition.
    ctr3 = cboHydCondition.ListIndex
    'Set the curve number.
    txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
End Sub
```

```
Private Sub cboHydSoilGrp_Change()
    Dim Counter As Integer
    Counter = 0
    'Get the index for selected hydrologic soil group.
    Do
        If (hydSoilGrp(Counter) = cboHydSoilGrp.Text) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While (Counter < 5)
    If Counter < 5 Then
        ctr4 = Counter
        'Set the curve number.
        txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
    End If
End Sub
```

```

Private Sub cboHydSoilGrp_Click()
    'Get the index for selected hydrologic soil group.
    ctr4 = cboHydSoilGrp.ListIndex
    'Set the curve number.
    txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
End Sub

Private Sub cboLandUse_Change()
    Dim Counter As Integer
    Counter = 0
    'Get the index for selected land use.
    Do
        If (landUse(Counter) = cboLandUse.Text) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While (Counter < 5)
    If Counter < 5 Then
        ctr1 = Counter
        'Set the curve number.
        txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
    End If
End Sub

Private Sub cboLandUse_Click()
    'Get the index for selected land use.
    ctr1 = cboLandUse.ListIndex
    'Set the curve number.
    txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
End Sub

Private Sub cmdCancel_Click()
    Dim resp As Integer

    resp = MsgBox("Do you wish to close?", vbYesNo + vbCritical + vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        End
    End If
End Sub

Private Sub cmdOK_Click()
    Dim Counter As Integer
    Dim resp As Integer

    'Get the soil code.
    Counter = 0
    Do
        If (UCCase(cboSoilNames.Text) = soilNames(Counter)) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While Counter < 1001

```

```

'Store the soil code and curve number if a user chooses to.
If (Counter < 1001) Then
    resp = MsgBox("Do you wish to store?", vbYesNo + vbCritical + vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        fileNum = FreeFile
        Open cellDatasetDir + "const.utl" For Append As fileNum
        Print #fileNum, "GETSOIL  " + Str(soilCodes(Counter))
        Print #fileNum, "CN2      " + txtCurveNumber.Text
        Close fileNum
    End
End If
End Sub

Private Sub Form_Load()
    Dim Counter As Integer
    Dim fileName As String
    Dim soilCode As Integer
    Dim soilName As String
    Dim c1 As Integer
    Dim c2 As Integer
    Dim c3 As Integer
    Dim c4 As Integer
    Dim ct1 As Integer
    Dim ct2 As Integer
    Dim ct3 As Integer
    Dim ct4 As Integer
    Dim value As Integer

    'Get directory paths.
    fileNum = FreeFile
    Open "c:\EVPaths.txt" For Input As fileNum
    Input #fileNum, cellDatasetDir, epicOutputDir, soilDir, epicDir
    Close fileNum

    'Open the soil file and create a soil names list.
    fileNum = FreeFile
    fileName = epicDir + "soil.lis"
    Open fileName For Input As fileNum
    Counter = 0
    Do
        Input #fileNum, soilCodes(Counter), soilNames(Counter)
        cboSoilNames.AddItem _
            soilNames(Counter)
        Counter = Counter + 1
    Loop While Not (EOF(fileNum) Or (Counter > 1000))
    cboSoilNames.ListIndex = 641
    Close fileNum

    'Open the curve number file and create various supporting lists.
    fileNum = FreeFile

```

```

fileName = epicDir + "curvenum.dat"
Open fileName For Input As fileNum
ctr1 = 0
'Create land use list.
Do
  Input #fileNum, landUse(ctr1)
  If (landUse(ctr1) <> "") Then
    cboLandUse.AddItem _
      landUse(ctr1)
  Else: Exit Do
  End If
  ctr1 = ctr1 + 1
Loop While Not (EOF(fileNum) And (ctr1 > 20))
ct1 = ctr1
cboLandUse.ListIndex = 0

'Create cover treatment list.
ctr2 = 0
Do
  Input #fileNum, coverTrtmnt(ctr2)
  If (coverTrtmnt(ctr2) <> "") Then
    cboCoverTrtmnt.AddItem _
      coverTrtmnt(ctr2)
  Else: Exit Do
  End If
  ctr2 = ctr2 + 1
Loop While Not (EOF(fileNum) And (ctr2 > 5))
ct2 = ctr2
cboCoverTrtmnt.ListIndex = 0

'Create hydrologic condition list.
ctr3 = 0
Do
  Input #fileNum, hydCondition(ctr3)
  If (hydCondition(ctr3) <> "") Then
    cboHydCondition.AddItem _
      hydCondition(ctr3)
  Else: Exit Do
  End If
  ctr3 = ctr3 + 1
Loop While Not (EOF(fileNum) And (ctr3 > 5))
ct3 = ctr3
cboHydCondition.ListIndex = 0

'Create hydrologic soil group list.
ctr4 = 0
Do
  Input #fileNum, hydSoilGrp(ctr4)
  If (hydSoilGrp(ctr4) <> "") Then
    cboHydSoilGrp.AddItem _
      hydSoilGrp(ctr4)
  Else: Exit Do

```



```

    End If
    ctr4 = ctr4 + 1
Loop While Not (EOF(fileNum) And (ctr4 > 5))
ct4 = ctr4
cboHydSoilGrp.ListIndex = 0
c1 = 0
c2 = 0
c3 = 0
c4 = 0

'Load the values in the curve number array.
Do
    If (c1 = ct1) Then
        Exit Do
    End If
    c2 = 0
    Do
        If (c2 = ct2) Then
            Exit Do
        End If
        c3 = 0
        Do
            If (c3 = ct3) Then
                Exit Do
            End If
            c4 = 0
            Do
                If (c4 = ct4) Then
                    Exit Do
                End If
                Input #fileNum, value
                curveNumber(c1, c2, c3, c4) = value
                c4 = c4 + 1
            Loop While Not (EOF(fileNum) And (c4 < ct4))
            c3 = c3 + 1
        Loop While Not (EOF(fileNum) And (c3 < ct3))
        c2 = c2 + 1
    Loop While Not (EOF(fileNum) And (c2 < ct2))
    c1 = c1 + 1
Loop While Not (EOF(fileNum) And (c1 < ct1))
Close fileNum
txtCurveNumber.Text = curveNumber(ctr1, ctr2, ctr3, ctr4)
End Sub

```

APPENDIX II.c

EPIC-View: Program Code for Constant Data Tool

' **Epic.constantData**

' This Script writes total number of cells, available in the gridded
' coverage, to file "selected.cll" and then invokes the constant data
' entry user interface.

' Mahesh Rao. Dated 5/19/96

```

selectedCell=( _cellDatasetDir.AsString+"selected.cll").AsFileName
selectedFile = TextFile.Make(selectedCell, #FILE_PERM_WRITE)
selectedFile.Write(_totalCells.AsString, _totalCells.AsString.Count)
selectedFile.WriteElt(_newLineChar)
selectedFile.Close
_epicDir.AsFileName.setCWD
command = _exeDir+"constdat.exe"
system.execute(command)
' Disable the Constant Data menu option.
_constEnableFlag = 0

```

' **constdat.bas**

' This form allows a user to modify the values of variables which remain constant for the
' whole field. They are stored in the file "const.utl" and later a constant EPIC input
' datafile "const.dat" is created and replicated for all the cells present in the gridded
' coverage.

Option Explicit

' directory paths.

Global cellDatasetDir As String

Global epicOutputDir As String

Global soilDir As String

Global epicDir As String

Global totalCells As Integer

Global validateText As String 'To use as a buffer for validation.

Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModule%)

'-----
' WaitShell

' Date: May 21, 1996
'-----

' WaitShell Subroutine:

' - Makes a synchronous call.
'-----

Public Sub WaitShell(ByVal AppName As String)

Dim hMod As Integer

hMod = Shell(AppName, 1)

If (Abs(hMod) > 32) Then

While (GetModuleUsage(hMod))

```

    DoEvents
    Wend
Else
    Debug.Print "Unable to start " & AppName
End If

End Sub

'-----
' ValidateData
' Date: June 12, 1996
'-----
' validateData Subroutine:
' - Validates the value entered by a user so
' - that it lies in the range of real numbers.
'-----

Public Sub validateData()
    Dim length As Integer
    Dim resp As Integer
    Dim start As Integer
    Dim alreadyDecimal As Integer
    Dim chr As String
    chr = 0
    length = 1
    alreadyDecimal = 0

    Do
        chr = Mid(validateText, length, 1)
        ' If character other than a number or a decimal.
        If (chr <> "." And chr <> "0" And chr <> "1" And chr <> "2" And chr <> "3" And chr <> "4" _
            And chr <> "5" And chr <> "6" And chr <> "7" And chr <> "8" And chr <> "9") Then
            Beep
            If (length = 1) Then
                start = 2
            Else
                start = 1
            End If
            If (Len(validateText) > 0) Then
                validateText = Mid(validateText, start, Len(validateText) - 1)
            End If
            Exit Do
        ' If character is a decimal.
        ElseIf (chr = "." And alreadyDecimal = 0) Then
            alreadyDecimal = 1
        ' If character is a second decimal point.
        ElseIf (chr = "." And alreadyDecimal = 1) Then
            Beep
            If (length = 1) Then
                start = 2
            Else
                start = 1
            End If
        End If
    End Do

```

```

    If (Len(validateText) > 0) Then
        validateText = Mid(validateText, start, Len(validateText) - 1)
    End If
    Exit Do
End If
length = length + 1
Loop While (length <= Len(validateText))
End Sub

```

'Const.dat.frm

' This VB form allows a user to modify the values of variables which remain constant for the
' whole field. They are stored in the file "const.utl" and later a constant EPIC input
' datafile, "const.dat" is created and replicated for all the cells present in the gridded
' coverage.

Option Explicit

Dim fileNum As Integer

Private Sub cmdCancel_Click()

Dim resp As Integer

resp = MsgBox("Do you wish to close?", vbYesNo + vbCritical + vbDefaultButton2, "EPIC-View")

If resp = vbYes Then

End

End If

End Sub

Private Sub cmdHelp_Click()

frmConstDat1Hlp.Show

End Sub

Private Sub cmdMore_Click()

frmConstDatMore.Show

End Sub

Private Sub cmdOk_Click()

Dim resp As Integer

Dim runCommand As String

Dim fileName As String

Dim Counter As Integer

' Write data to file if a user chooses to.

resp = MsgBox("Do you wish to load constant variables in the datasets?", vbYesNo + vbCritical +
vbDefaultButton2, "EPIC-View")

If resp = vbYes Then

Hide

fileNum = FreeFile

Open cellDatasetDir + "const.utl" For Append As fileNum

Print #fileNum, "TITLE(1) " + txtTitle1.Text

Print #fileNum, "TITLE(2) Field"

```

Print #fileNum, "NBYR"    " + txtNBYR.Text
Print #fileNum, "IYR"    " + txtIYR.Text
Print #fileNum, "IMO"    " + txtIMO.Text
Print #fileNum, "IDA"    " + txtIDA.Text
Print #fileNum, "NIPD"   " + txtNIPD.Text
Print #fileNum, "IPD"    " + txtIPD.Text
Print #fileNum, "NGN"    " + txtNGN.Text
Print #fileNum, "IGN"    " + frmConstDatMore.txtIGN.Text
Print #fileNum, "IGSD"   " + frmConstDatMore.txtIGSD.Text
Print #fileNum, "LPYR"   " + txtLPYR.Text
Print #fileNum, "IET"    " + txtIET.Text
Print #fileNum, "ISCN"   " + frmConstDatMore.txtISCN.Text
Print #fileNum, "IGRAF"  " + frmConstDatMore.txtIGRAF.Text
Print #fileNum, "ICODE"  " + txtICODE.Text
Print #fileNum, "ITYP"   " + frmConstDatMore.txtITYP.Text
Print #fileNum, "ISTA"   " + frmConstDatMore.txtISTA.Text
Print #fileNum, "IHUS"   " + txtIHUS.Text
Print #fileNum, "CHL"    " + frmConstDatMore.txtCHL.Text
Print #fileNum, "CHS"    " + frmConstDatMore.txtCHS.Text
Print #fileNum, "CHN"    " + frmConstDatMore.txtCHN.Text
Print #fileNum, "SN"     " + frmConstDatMore.txtSN.Text
Print #fileNum, "APM"    " + frmConstDatMore.txtAPM.Text
Print #fileNum, "SNO"    " + frmConstDatMore.txtSNO.Text
Print #fileNum, "RCN"    " + frmConstDatMore.txtRCN.Text
Print #fileNum, "RTN"    " + frmConstDatMore.txtRTN.Text
Print #fileNum, "CO2"    " + frmConstDatMore.txtCO2.Text
Print #fileNum, "CSALT"  " + frmConstDatMore.txtCSALT.Text
Print #fileNum, "CHD"    " + frmConstDatMore.txtCHD.Text
Print #fileNum, "PEC"    " + txtPEC.Text
Print #fileNum, "DRV"    " + txtDRV.Text
Print #fileNum, "BUS(1)" " + frmConstDatMore.txtBUS1.Text
Print #fileNum, "BUS(2)" " + frmConstDatMore.txtBUS2.Text
Print #fileNum, "BUS(3)" " + frmConstDatMore.txtBUS3.Text
Print #fileNum, "BUS(4)" " + frmConstDatMore.txtBUS4.Text
Print #fileNum, "FL"     " + frmConstDatMore.txtFL.Text
Print #fileNum, "FW"     " + frmConstDatMore.txtFW.Text
Print #fileNum, "ANG"    " + frmConstDatMore.txtANG.Text
Print #fileNum, "STD"    " + frmConstDatMore.txtSTD.Text
Print #fileNum, "ACW"    " + txtACW.Text

Close fileNum
fileNum = FreeFile
fileName = cellDatasetDir + "const.dat"
Open fileName For Output As fileNum
Close fileNum
' Load the constant variables' values to the const.dat.
runCommand = epicDir + "util epic " + cellDatasetDir + "const.dat @" + cellDatasetDir +
"const.utl"
WaitShell (runCommand)
' Replicate the const.dat file for all the cells' dataset files.
resp = MsgBox("Creating datasets for all the cells.", vbCritical, "EPIC-View")
Counter = 1

```

```

Do
    fileName = cellDatasetDir + "form" + Trim(Str(Counter)) + ".dat"
    FileCopy cellDatasetDir + "const.dat", fileName
    Counter = Counter + 1
Loop While Counter <= totalCells
resp = MsgBox("Datasets have been created.", vbInformation, "EPIC-View")
End
End If
End Sub

```

```

Private Sub Form_Load()
' Get all directory paths.
fileNum = FreeFile
Open "c:\EVPaths.txt" For Input As fileNum
Input #fileNum, cellDatasetDir, epicOutputDir, soilDir, epicDir
Close fileNum
fileNum = FreeFile
' Get total number of cells in the gridded coverage.
Open cellDatasetDir + "selected.cll" For Input As fileNum
Input #fileNum, totalCells
Close fileNum
End Sub

```

```

Private Sub txtACW_Change()
    validateText = txtACW.Text
    validateData
    txtACW.Text = validateText
End Sub

```

```

Private Sub txtDRV_Change()
    validateText = txtDRV.Text
    validateData
    txtDRV.Text = validateText
End Sub

```

```

Private Sub txtICODE_Change()
    validateText = txtICODE.Text
    validateData
    txtICODE.Text = validateText
End Sub

```

```

Private Sub txtIDA_Change()
    validateText = txtIDA.Text
    validateData
    txtIDA.Text = validateText
End Sub

```

```

Private Sub txtIET_Change()
    validateText = txtIET.Text
    validateData
    txtIET.Text = validateText
End Sub

```

```
Private Sub txtIHUS_Change()  
    validateText = txtIHUS.Text  
    validateData  
    txtIHUS.Text = validateText  
End Sub
```

```
Private Sub txtIMO_Change()  
    validateText = txtIMO.Text  
    validateData  
    txtIMO.Text = validateText  
End Sub
```

```
Private Sub txtIPD_Change()  
    validateText = txtIPD.Text  
    validateData  
    txtIPD.Text = validateText  
End Sub
```

```
Private Sub txtIYR_Change()  
    validateText = txtIYR.Text  
    validateData  
    txtIYR.Text = validateText  
End Sub
```

```
Private Sub txtLPYR_Change()  
    validateText = txtLPYR.Text  
    validateData  
    txtLPYR.Text = validateText  
End Sub
```

```
Private Sub txtNBYR_Change()  
    validateText = txtNBYR.Text  
    validateData  
    txtNBYR.Text = validateText  
End Sub
```

```
Private Sub txtNGN_Change()  
    validateText = txtNGN.Text  
    validateData  
    txtNGN.Text = validateText  
End Sub
```

```
Private Sub txtNIPD_Change()  
    validateText = txtNIPD.Text  
    validateData  
    txtNIPD.Text = validateText  
End Sub
```

```
Private Sub txtPEC_Change()  
    validateText = txtPEC.Text  
    validateData
```

```
txtPEC.Text = validateText  
End Sub
```

'Constmore.frm

' This form allows a user to modify the values of variables which remain constant for the
' whole field. They are stored in the file "const.utl" and later a constant EPIC input
' datafile "const.dat" is created and replicated for all the cells present in the gridded
' coverage.

```
Private Sub cmdHelp_Click()  
frmConstDat2Hlp.Show  
End Sub
```

```
Private Sub cmdOk_Click()  
Hide  
End Sub
```

```
Private Sub txtANG_Change()  
validateText = txtANG.Text  
validateData  
txtANG.Text = validateText  
End Sub
```

```
Private Sub txtAPM_Change()  
validateText = txtAPM.Text  
validateData  
txtAPM.Text = validateText  
End Sub
```

```
Private Sub txtBUS1_Change()  
validateText = txtBUS1.Text  
validateData  
txtBUS1.Text = validateText  
End Sub
```

```
Private Sub txtBUS2_Change()  
validateText = txtBUS2.Text  
validateData  
txtBUS2.Text = validateText  
End Sub
```

```
Private Sub txtBUS3_Change()  
validateText = txtBUS3.Text  
validateData  
txtBUS3.Text = validateText  
End Sub
```

```
Private Sub txtBUS4_Change()  
validateText = txtBUS4.Text  
validateData  
txtBUS4.Text = validateText
```


End Sub

```
Private Sub txtCF_Change()  
    validateText = txtCF.Text  
    validateData  
    txtCF.Text = validateText  
End Sub
```

```
Private Sub txtCHD_Change()  
    validateText = txtCHD.Text  
    validateData  
    txtCHD.Text = validateText  
End Sub
```

```
Private Sub txtCHL_Change()  
    validateText = txtCHL.Text  
    validateData  
    txtCHL.Text = validateText  
End Sub
```

```
Private Sub txtCHN_Change()  
    validateText = txtCHN.Text  
    validateData  
    txtCHN.Text = validateText  
End Sub
```

```
Private Sub txtCHS_Change()  
    validateText = txtCHS.Text  
    validateData  
    txtCHS.Text = validateText  
End Sub
```

```
Private Sub txtCO2_Change()  
    validateText = txtCO2.Text  
    validateData  
    txtCO2.Text = validateText  
End Sub
```

```
Private Sub txtCSALT_Change()  
    validateText = txtCSALT.Text  
    validateData  
    txtCSALT.Text = validateText  
End Sub
```

```
Private Sub txtFL_Change()  
    validateText = txtFL.Text  
    validateData  
    txtFL.Text = validateText  
End Sub
```

```
Private Sub txtFW_Change()  
    validateText = txtFW.Text
```

```
    validateData
    txtFW.Text = validateText
End Sub

Private Sub txtIGN_Change()
    validateText = txtIGN.Text
    validateData
    txtIGN.Text = validateText
End Sub

Private Sub txtIGRAF_Change()
    validateText = txtIGRAF.Text
    validateData
    txtIGRAF.Text = validateText
End Sub

Private Sub txtIGSD_Change()
    validateText = txtIGSD.Text
    validateData
    txtIGSD.Text = validateText
End Sub

Private Sub txtISCN_Change()
    validateText = txtISCN.Text
    validateData
    txtISCN.Text = validateText
End Sub

Private Sub txtISTA_Change()
    validateText = txtISTA.Text
    validateData
    txtISTA.Text = validateText
End Sub

Private Sub txtITYP_Change()
    validateText = txtITYP.Text
    validateData
    txtITYP.Text = validateText
End Sub

Private Sub txtRCN_Change()
    validateText = txtRCN.Text
    validateData
    txtRCN.Text = validateText
End Sub

Private Sub txtRTN_Change()
    validateText = txtRTN.Text
    validateData
    txtRTN.Text = validateText
End Sub
```

```
Private Sub txtSN_Change()  
    validateText = txtSN.Text  
    validateData  
    txtSN.Text = validateText  
End Sub
```

```
Private Sub txtSNO_Change()  
    validateText = txtSNO.Text  
    validateData  
    txtSNO.Text = validateText  
End Sub
```

```
Private Sub txtSTD_Change()  
    validateText = txtSTD.Text  
    validateData  
    txtSTD.Text = validateText  
End Sub
```

```
Private Sub txtSWV_Change()  
    validateText = txtSWV.Text  
    validateData  
    txtSWV.Text = validateText  
End Sub
```

APPENDIX II.d

EPIC-View: Program Code for Management Data Tool

```

' Epic.mgmtData
' This Avenue script provides a user with choices to make a set of management practices
' generic or select specific cells and enter management practices for
' those cells by invoking Management Practices data entry user interface.
' Mahesh Rao. Dated 5/23/96
'
' Reset the selectedCells list.
totFiles=_selectCellsList.Count
index = totFiles - 1
while (index >= 0)
  _selectCellsList.Remove(index)
  index = index - 1
end
theTable = av.GetProject.FindDoc(_mainTable)
if(nil=theTable)then
  MsgBox.Error("The table: "+_mainTable+", not found.", "Epic")
  exit
end
theTableWin=theTable.GetWin
if (theTableWin.IsOpen.Not)then
  theTableWin.Open
else
  theTableWin.Activate
end
theTableWin.Minimize
'
genericFlag = false
theVTab = theTable.GetVTab
myVTab = theVTab.GetSelection
'
' If no cells are selected, give user a choice to make management practices generic.
if (0=theVTab.GetSelection.Count) then
  genericFlag = MsgBox.YesNo("Do you wish to make this set of management practices generic for the
whole field?", "EPIC-View", False )
  if(genericFlag.Not)then
    MsgBox.Error("In that case, please select the cells for entering management practices.", "EPIC-View")
    exit
  end
end
'
sortField = theVTab.FindField(_cellIdFld)
theTable.Sort(sortField, False)
'
' If cells are selected, build the selectedCells list.

for each rec in myVTab
  cellIdField = theVTab.FindField(_cellIdFld)

```

```

    cellId = theVTab.ReturnValueString(cellIdField, rec)
    _selectCellsList.Add(cellId)
end
'
' Write selected cells ids to file selected.cll.
'
selectedCell=(_cellDatasetDir.AsString+"selected.cll").AsFileName
selectedFile = TextFile.Make(selectedCell, #FILE_PERM_WRITE)
path=_epicOutputDir
selectedFile.Write(path, path.Count)
selectedFile.WriteElt(_newLineChar)
selectedFile.Write(_totalCells.AsString, _totalCells.AsString.Count)
selectedFile.WriteElt(_newLineChar)
selectedFile.Write(genericFlag.AsString, genericFlag.AsString.Count)
selectedFile.WriteElt(_newLineChar)

if (genericFlag.Not) then
  for each cellId in _selectCellsList
    selectedFile.Write(cellId, cellId.Count)
    selectedFile.WriteElt(_newLineChar)
  end
end
selectedFile.Close
'Run Visual Basic management data entry screen.
command = _exeDir+"mgmt.exe"
system.execute(command)

```

'Mgmt.bas

' This form allows a user to enter management practices for selected cells or for the whole
 ' field as chosen by a user. Different operations can be selected and the operations are
 ' stored in the cell specific "mgmt.utl" file.

Option Explicit

```

Global monthSel As Integer 'For selected month.
Global daySel As Integer 'For selected day.
Global unitType(0 To 2) As String 'For selected unitd type.
Global operationCode As Integer 'For selected management operation.
Global cellFiles() As Integer 'To maintain different cell files' pointers.
Global cellIndex() As Integer 'To maintain different cell indexes.
Global cellExist() As Integer 'To maintain different flags if cell file exists.
Global selectedCells(1 To 1000) As String 'To store number of selected cells.
Global nro() As Integer 'To maintain different cells' NRO values.
Global totCellsSel As Integer
Global didCropRotation As Integer
Global currentCell As Integer
Global justDidCropRotation As Integer
Global validateText As String 'To use as a buffer for validation.

```

' Directory paths.

```
Global cellDatasetDir As String
Global epicOutputDir As String
Global soilDir As String
Global epicDir As String
```

```
' Management related variables.
```

```
Global armn As Single
Global armx As Single
Global bft As Single
Global bir As Single
Global drt As Single
Global efi As Single
Global fdsf As Single
Global fmx As Single
Global fnp As Single
Global idft As Single
Global idr As Single
Global ifa As Single
Global ifd As Single
Global iffr As Single
Global iri As Single
Global irr As Single
Global lm As Single
Global nirr As Single
Global vimx As Single
```

```
' ValidateData
```

```
' Date: June 12, 1996
```

```
'-----
' validateData Subroutine:
```

```
' - Validates the value entered by a user so
' - that it lies in the range of real numbers.
```

```
'-----
Public Sub validateData()
```

```
Dim length As Integer
Dim resp As Integer
Dim start As Integer
Dim alreadyDecimal As Integer
Dim chr As String
chr = 0
length = 1
alreadyDecimal = 0
```

```
Do
```

```
chr = Mid(validateText, length, 1)
```

```
' If character other than a number or a decimal.
```

```
If (chr <> "." And chr <> "0" And chr <> "1" And chr <> "2" And chr <> "3" And chr <> "4" _
And chr <> "5" And chr <> "6" And chr <> "7" And chr <> "8" And chr <> "9") Then
```

```
Beep
```

```
If (length = 1) Then
```

```
start = 2
```

```

Else
    start = 1
End If
If (Len(validateText) > 0) Then
    validateText = Mid(validateText, start, Len(validateText) - 1)
End If
Exit Do
' If character is a decimal.
ElseIf (chr = "." And alreadyDecimal = 0) Then
    alreadyDecimal = 1
' If character is a second decimal point.
ElseIf (chr = "." And alreadyDecimal = 1) Then
    Beep
    If (length = 1) Then
        start = 2
    Else
        start = 1
    End If
    If (Len(validateText) > 0) Then
        validateText = Mid(validateText, start, Len(validateText) - 1)
    End If
    Exit Do
End If
length = length + 1
Loop While (length <= Len(validateText))
End Sub

```

'Mgmt.frm

' This form allows a user to enter management practices for selected cells or for the whole
' field as chosen by a user. Different operations can be selected and the operations are
' stored in the cell specific "mgmt.utl" file.

Option Explicit

```

Dim month(1 To 12) As String
Dim day(1 To 31) As Integer
Dim operation(1 To 100) As String
Dim operCode(1 To 100) As Integer
Dim fileNum As Integer
Dim firstTime As Integer

```

Private Sub cboDay_Change()

```

    Dim Counter As Integer
    Counter = 1
    Do
        If (day(Counter) = cboDay.Text) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While Counter < 32
    If Counter < 32 Then

```

```
    daySel = Counter ' Store to global variable.
End If
End Sub

Private Sub cboDay_Click()
    Dim Counter As Integer
    Counter = 1
    Do
        If (day(Counter) = cboDay.Text) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While Counter < 32
    daySel = Counter ' Store to global variable.
End Sub

Private Sub cboMonth_Change()
    Dim Counter As Integer
    Counter = 1
    Do
        If (UCase(month(Counter)) = UCCase(cboMonth.Text)) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While Counter < 13
    If Counter < 13 Then
        monthSel = Counter ' Store to global variable.
    End If
End Sub

Private Sub cboMonth_Click()
    Dim Counter As Integer
    Counter = 1
    Do
        If (UCase(month(Counter)) = UCCase(cboMonth.Text)) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While Counter < 13
    monthSel = Counter ' Store to global variable.
End Sub

Private Sub cboOper_Change()
    addOperation
End Sub

Private Sub cboOper_Click()
    addOperation
End Sub

Private Sub cmdAddOper_Click()
    addOperation
End Sub
```



```

Private Sub cmdCancel_Click()
    Dim Counter As Integer
    Dim resp As Integer

    resp = MsgBox("Do you wish to close?", vbYesNo + vbCritical + vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        End
    End If
End Sub

Private Sub cmdClose_Click()
    Dim Counter As Integer
    Dim index As String
    Dim nroVal As String
    Dim resp As Integer
    Dim fileName As String

    ' Store the values if a user chooses to.
    resp = MsgBox("Do you wish load this set of management practices and close?", vbYesNo + vbCritical
+ vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        Counter = 1
        Do
            cellFiles(Counter) = FreeFile
            fileName = "mgmt" + selectedCells(Counter) + ".utl"
            Open cellDatasetDir + fileName For Append As cellFiles(Counter)
            'Store all management practices related variables (one time only).
            If cellExist(Counter) = 0 Then
                Print #cellFiles(Counter), "NRO " + Str(nro(Counter))
                Print #cellFiles(Counter), "NIRR " + Str(nirr)
                Print #cellFiles(Counter), "IRR " + Str(irr)
                Print #cellFiles(Counter), "IRI " + Str(iri)
                Print #cellFiles(Counter), "IFA " + Str(ifa)
                Print #cellFiles(Counter), "LM " + Str(lm)
                Print #cellFiles(Counter), "IFD " + Str(ifd)
                Print #cellFiles(Counter), "IDR " + Str(idr)
                Print #cellFiles(Counter), "IFFR " + Str(iffrr)
                Print #cellFiles(Counter), "IDFT " + Str(idft)
                Print #cellFiles(Counter), "BIR " + Str(bir)
                Print #cellFiles(Counter), "EFI " + Str(efi)
                Print #cellFiles(Counter), "VIMX " + Str(vimx)
                Print #cellFiles(Counter), "ARMN " + Str(armn)
                Print #cellFiles(Counter), "ARMX " + Str(armx)
                Print #cellFiles(Counter), "BFT " + Str(bft)
                Print #cellFiles(Counter), "FNP " + Str(fnp)
                Print #cellFiles(Counter), "FMX " + Str(fmx)
                Print #cellFiles(Counter), "DRT " + Str(drt)
                Print #cellFiles(Counter), "FDSF " + Str(fdsf)
            ElseIf (didCropRotation = 1) Then
                Print #cellFiles(Counter), "NRO " + Str(nro(Counter))
            End If
        Loop
    End If

```

```

    Close cellFiles(Counter)
    cellFiles(Counter) = FreeFile
    ' Rewrite the current number of operations and current NRO value.
    Open cellDatasetDir + fileName For Binary As cellFiles(Counter)
    index = Space(4 - Len(Str(cellIndex(Counter))))
    nroVal = Space(4 - Len(Str(nro(Counter))))
    index = index + Str(cellIndex(Counter))
    nroVal = nroVal + Str(nro(Counter))
    Put #cellFiles(Counter), 4, index
    Put #cellFiles(Counter), 10, nroVal
    Close cellFiles(Counter)
    Counter = Counter + 1
  Loop While Counter <= totCellsSel ' Loop for all selected cells.
End
End If
End Sub

Private Sub cmdHelp_Click()
  frmMgmtHelp.Show
End Sub

Private Sub cmdMore_Click()
  frmMgmtDefa.Show
End Sub

Private Sub cmdNewCrop_Click()
  Dim Counter As Integer
  Dim resp As Integer
  ' Provide for crop rotation if a user chooses to.
  If justDidCropRotation = 0 Then ' To avoid consecutive crop rotations.
    resp = MsgBox("Do you wish to add new crop rotation?", vbYesNo + vbCritical + vbDefaultButton2,
"EPIC-View")
    If resp = vbYes Then
      justDidCropRotation = 1
      Counter = 1
      Do
        If (cellIndex(Counter) > 0) Then
          cellIndex(Counter) = cellIndex(Counter) + 1 'Increment no. of operations.
          nro(Counter) = nro(Counter) + 1 'Increment NRO value.
        End If
        Counter = Counter + 1
      Loop While Counter <= totCellsSel 'Loop for all selected cells.
      didCropRotation = 1
    End If
  Else 'If a user attempted consecutive crop rotations.
    resp = MsgBox("Please enter an operation before another crop rotation.", vbInformation, "EPIC-View")
  End If
End Sub

Private Sub Form_Load()
  Dim Counter As Integer

```

```

Dim fileName As String
Dim oper As String
Dim opCode As Integer
Dim path As String
Dim str1 As String
Dim resp As Integer
Dim totCells As Integer
Dim genericFlag As String

```

```

'Store months.
month(1) = "JANUARY"
month(2) = "FEBRUARY"
month(3) = "MARCH"
month(4) = "APRIL"
month(5) = "MAY"
month(6) = "JUNE"
month(7) = "JULY"
month(8) = "AUGUST"
month(9) = "SEPTEMBER"
month(10) = "OCTOBER"
month(11) = "NOVEMBER"
month(12) = "DECEMBER"

```

```

'Store units type.
unitType(0) = "ENGLISH"
unitType(1) = "METRIC"

```

```

'Create months list.
Counter = 1
Do
    cboMonth.AddItem month(Counter)
    Counter = Counter + 1
Loop While Counter < 13
cboMonth.ListIndex = 0
'Create days list.
Counter = 1
Do
    day(Counter) = Str(Counter)
    cboDay.AddItem day(Counter)
    Counter = Counter + 1
Loop While Counter < 32
cboDay.ListIndex = 0

```

```

'Get directory paths.
fileNum = FreeFile
Open "c:\EVPPaths.txt" For Input As fileNum
Input #fileNum, cellDatasetDir, epicOutputDir, soilDir, epicDir
Close fileNum

```

```

'Check if this set is generic for whole field.
fileNum = FreeFile
fileName = cellDatasetDir + "selected.cll"

```

```

Open fileName For Input As fileNum
Input #fileNum, path, totCells, genericFlag
Counter = 1
totCellsSel = 0
If UCase(genericFlag) = "FALSE" Then 'For selected cells only.
  Do
    If (EOF(fileNum)) Then
      Exit Do
    End If
    Input #fileNum, selectedCells(Counter)
    Counter = Counter + 1
    totCellsSel = totCellsSel + 1
  Loop While Not EOF(fileNum)
  resp = 0
Else 'Generic for the field.
  Do
    selectedCells(Counter) = Counter
    Counter = Counter + 1
    totCellsSel = totCellsSel + 1
  Loop While Counter <= totCells
  resp = vbYes
End If
Close fileNum

'Resize all arrays.
ReDim cellFiles(1 To totCellsSel)
ReDim cellIndex(1 To totCellsSel)
ReDim cellExist(1 To totCellsSel)
ReDim openMode(1 To totCellsSel)
ReDim nro(1 To totCellsSel)

'Initialize NRO array with 1.
Counter = 1
Do
  nro(Counter) = 1
  Counter = Counter + 1
Loop While Counter <= totCellsSel
Counter = 1
' Open all selected cells' mgmt#.utl files for writing/appendng.
Do
  fileName = "mgmt" + selectedCells(Counter) + ".utl"
  path = Dir(cellDatasetDir + fileName)
  If path = UCase(fileName) Then
    currentCell = selectedCells(Counter)
    If (resp = 0) Then
      resp = MsgBox("Overwrite all previously existing management files?", vbYesNo + vbQuestion +
vbDefaultButton2, "EPIC-View")
    End If
    If (resp = vbNo) Then
      cellFiles(Counter) = FreeFile
      ' Read the previous index and NRO values.
      Open cellDatasetDir + fileName For Input As cellFiles(Counter)

```

```

    str1 = Input(2, cellFiles(Counter))
    Input #cellFiles(Counter), cellIndex(Counter), nro(Counter)
    Close cellFiles(Counter)
    If (cellIndex(Counter) > 0) Then
        cellExist(Counter) = 1
    Else
        cellExist(Counter) = 0
    End If
    ElseIf (resp = vbYes) Then
        cellFiles(Counter) = FreeFile
        Open cellDatasetDir + fileName For Output As cellFiles(Counter)
        Print #cellFiles(Counter), "! 0 1" 'Write initial index on this line.'
        Close cellFiles(Counter)
    End If
Else
    cellFiles(Counter) = FreeFile
    Open cellDatasetDir + fileName For Output As cellFiles(Counter)
    Print #cellFiles(Counter), "! 0 1" 'Write initial index on this line.'
    Close cellFiles(Counter)
End If
Counter = Counter + 1
Loop While Counter <= totCellsSel

'Open the management operation file and create a list of
'management operations.
fileNum = FreeFile
fileName = epicDir + "mgmtope.dat"
Open fileName For Input As fileNum
Counter = 1
Do
    If (EOF(fileNum)) Then
        Exit Do
    End If
    Input #fileNum, opCode
    operCode(Counter) = opCode
    Input #fileNum, operation(Counter)
    cboOper.AddItem operation(Counter)
    Counter = Counter + 1
Loop While Not (EOF(fileNum) Or (Counter > 100))
firstTime = 0
cboOper.ListIndex = 0 'Set highlight to first item in list.
Close fileNum
initMgmtVars 'Initialize management related variables.
firstTime = 1
End Sub

' InitMgmtVars
' Date: May 19, 1996
'-----
' initMgmtVars Subroutine:
' - Initializes all global variables.
'-----

```

```

Public Sub initMgmtVars()
    armn = Val(frmMgmtDefa.txtARMN.Text)
    armx = Val(frmMgmtDefa.txtARMX.Text)
    bft = Val(frmMgmtDefa.txtBFT.Text)
    bir = Val(frmMgmtDefa.txtBIR.Text)
    drt = Val(frmMgmtDefa.txtDRT.Text)
    efi = Val(frmMgmtDefa.txtEFI.Text)
    fdsf = Val(frmMgmtDefa.txtFDSF.Text)
    fmx = Val(frmMgmtDefa.txtFMX.Text)
    fnp = Val(frmMgmtDefa.txtFNP.Text)
    idft = Val(frmMgmtDefa.txtIDFT.Text)
    idr = Val(frmMgmtDefa.txtIDR.Text)
    ifa = Val(frmMgmtDefa.txtIFA.Text)
    ifd = Val(frmMgmtDefa.txtIFD.Text)
    iffr = Val(frmMgmtDefa.txtIFFR.Text)
    iri = Val(frmMgmtDefa.txtIRI.Text)
    irr = Val(frmMgmtDefa.txtIRR.Text)
    lm = Val(frmMgmtDefa.txtLM.Text)
    nirr = Val(frmMgmtDefa.txtNIRR.Text)
    didCropRotation = 0
    justDidCropRotation = 0
    vimx = Val(frmMgmtDefa.txtVIMX.Text)
End Sub

```

```

' AddOperation
' Date: May 19, 1996
'-----
' addOperation Subroutine:
' - Checks for the operation code selected by
' - a user and calls appropriate form to enter
' - remaining data.
'-----

```

```

Public Sub addOperation()
    Dim Counter As Integer
    Dim resp As Integer
    Dim choice As Integer
    Dim fileName As String

    Counter = 1
    Do
        If (operation(Counter) = UCase(cboOper.Text)) Then
            Exit Do
        End If
        Counter = Counter + 1
    Loop While Counter < 100
    If Counter < 100 Then
        operationCode = operCode(Counter)
        If firstTime <> 0 Then
            choice = operationCode
            If choice = 71 Then ' Fertilize
                justDidCropRotation = 0
            End If
        End If
    End If

```

```

    frmFert.Show
ElseIf choice = 11 Then ' Sprayer.
    justDidCropRotation = 0
    frmPest.Show
ElseIf choice = 2 Then ' Row Planter.
    justDidCropRotation = 0
    frmRowPlntr.Show
ElseIf choice = 72 Then ' Irrigation.
    justDidCropRotation = 0
    frmIrrig.Show
ElseIf (choice = 19 Or choice = 21 Or choice = 23 Or choice = 29 _
    Or choice = 30 Or choice = 51) Then
    justDidCropRotation = 0
    frmCultivate.Show
ElseIf (choice = 41 Or choice = 28 Or choice = 33) Then
    resp = MsgBox("Do you wish to store this operation?", vbYesNo + vbCritical +
vbDefaultButton2, "EPIC-View")
    If resp = vbYes Then
        justDidCropRotation = 0
        Counter = 1
        Do
            cellFiles(Counter) = FreeFile
            fileName = "mgmt" + selectedCells(Counter) + ".utl"
            Open cellDatasetDir + fileName For Append As cellFiles(Counter)
            cellIndex(Counter) = cellIndex(Counter) + 1
            Print #cellFiles(Counter), "MON(" + Trim(Str(cellIndex(Counter))) + ") " +
Str(monthSel)
            Print #cellFiles(Counter), "DAY(" + Trim(Str(cellIndex(Counter))) + ") " + Str(daySel)
            Print #cellFiles(Counter), "COD(" + Trim(Str(cellIndex(Counter))) + ") " +
Str(operationCode)
            Close cellFiles(Counter)
            Counter = Counter + 1
        Loop While Counter <= totCellsSel
    End If
End If
End If
End If
End Sub

```

'Mgmtdefa.frm

' This form allows a user to enter management related variables for selected cells or for
' the whole field as chosen by a user. These are stored in the cell specific "mgmt.utl" file.

```

Private Sub cmdHelp_Click()
    frmMgmDefHlp.Show
End Sub

```

```

Private Sub cmdOk_Click()
    'Set global variables.
    armn = Val(txtARMN.Text)
    armx = Val(txtARMX.Text)

```

```
bft = Val(txtBFT.Text)
bir = Val(txtBIR.Text)
drt = Val(txtDRT.Text)
efi = Val(txtEFI.Text)
fdsf = Val(txtFDSF.Text)
fmx = Val(txtFMX.Text)
fnp = Val(txtFNP.Text)
idft = Val(txtIDFT.Text)
idr = Val(txtIDR.Text)
ifa = Val(txtIFA.Text)
ifd = Val(txtIFD.Text)
iffr = Val(txtIFFR.Text)
iri = Val(txtIRI.Text)
irr = Val(txtIRR.Text)
lm = Val(txtLM.Text)
nirr = Val(txtNIRR.Text)
vimx = Val(txtVIMX.Text)

Hide
End Sub

Private Sub txtARMN_Change()
    validateText = txtARMN.Text
    validateData
    txtARMN.Text = validateText
End Sub

Private Sub txtARMX_Change()
    validateText = txtARMX.Text
    validateData
    txtARMX.Text = validateText
End Sub

Private Sub txtBFT_Change()
    validateText = txtBFT.Text
    validateData
    txtBFT.Text = validateText
End Sub

Private Sub txtBIR_Change()
    validateText = txtBIR.Text
    validateData
    txtBIR.Text = validateText
End Sub

Private Sub txtDRT_Change()
    validateText = txtDRT.Text
    validateData
    txtDRT.Text = validateText
End Sub

Private Sub txtEFI_Change()
    validateText = txtEFI.Text
```



```
    validateData
    txtEFI.Text = validateText
End Sub

Private Sub txtFDSF_Change()
    validateText = txtFDSF.Text
    validateData
    txtFDSF.Text = validateText
End Sub

Private Sub txtFMX_Change()
    validateText = txtFMX.Text
    validateData
    txtFMX.Text = validateText
End Sub

Private Sub txtFNP_Change()
    validateText = txtFNP.Text
    validateData
    txtFNP.Text = validateText
End Sub

Private Sub txtIDFT_Change()
    validateText = txtIDFT.Text
    validateData
    txtIDFT.Text = validateText
End Sub

Private Sub txtIDR_Change()
    validateText = txtIDR.Text
    validateData
    txtIDR.Text = validateText
End Sub

Private Sub txtIFA_Change()
    validateText = txtIFA.Text
    validateData
    txtIFA.Text = validateText
End Sub

Private Sub txtIFD_Change()
    validateText = txtIFD.Text
    validateData
    txtIFD.Text = validateText
End Sub

Private Sub txtIFFR_Change()
    validateText = txtIFFR.Text
    validateData
    txtIFFR.Text = validateText
End Sub

Private Sub txtIRI_Change()
```

```

    validateText = txtIRI.Text
    validateData
    txtIRI.Text = validateText
End Sub

```

```

Private Sub txtIRR_Change()
    validateText = txtIRR.Text
    validateData
    txtIRR.Text = validateText
End Sub

```

```

Private Sub txtLM_Change()
    validateText = txtLM.Text
    validateData
    txtLM.Text = validateText
End Sub

```

```

Private Sub txtNIRR_Change()
    validateText = txtNIRR.Text
    validateData
    txtNIRR.Text = validateText
End Sub

```

```

Private Sub txtVIMX_Change()
    validateText = txtVIMX.Text
    validateData
    txtVIMX.Text = validateText
End Sub

```

'Fert.frm

' This form allows a user to select a fertilizer from a list of fertilizers and other variable
' values related to fertilize operation and store them in the cell specific "mgmt.utl" files.

Option Explicit

```

Dim fertilizer(1 To 100) As String
Dim fileNum As Integer

```

```

Private Sub cmdCancel_Click()
    Hide
End Sub

```

```

Private Sub cmdOk_Click()
    Dim Counter As Integer
    Dim ind As Integer
    Dim resp As Integer
    Dim addStr As String
    Dim fileName As String
    'Get the fertilizer code for selected fertilizer.
    ind = 1
    Do

```

```

    If fertilizer(ind) = cboFert.Text Then
        Exit Do
    End If
    ind = ind + 1
Loop While ind < 100

'Store the management operation if a user chooses to.
Counter = 1
If (ind < 100) Then
    resp = MsgBox("Do you wish to store this operation?", vbYesNo + vbCritical + vbDefaultButton2,
"EPIC-View")
    If resp = vbYes Then
        If (UCase(cboUnit.Text) = "ENGLISH") Then
            addStr = "E"
        Else
            addStr = ""
        End If
        Do
            cellFiles(Counter) = FreeFile
            fileName = "mgmt" + selectedCells(Counter) + ".utl"
            Open cellDatasetDir + fileName For Append As cellFiles(Counter)
            cellIndex(Counter) = cellIndex(Counter) + 1
            Print #cellFiles(Counter), "MON(" + Trim(Str(cellIndex(Counter))) + ") " + Str(monthSel)
            Print #cellFiles(Counter), "DAY(" + Trim(Str(cellIndex(Counter))) + ") " + Str(daySel)
            Print #cellFiles(Counter), "COD(" + Trim(Str(cellIndex(Counter))) + ") " + Str(operationCode)
            Print #cellFiles(Counter), "FN(" + Trim(Str(cellIndex(Counter))) + ") " + Str(ind)
            Print #cellFiles(Counter), "FAP(" + Trim(Str(cellIndex(Counter))) + ") " + txtAppRate.Text +
addStr
            Print #cellFiles(Counter), "FDP(" + Trim(Str(cellIndex(Counter))) + ") " + txtFertDepth.Text
+ addStr
            Print #cellFiles(Counter), "HUSC(" + Trim(Str(cellIndex(Counter))) + ") " + txtHUSched.Text
            Close cellFiles(Counter)
            Counter = Counter + 1
        Loop While Counter <= totCellsSel 'Loop for all selected cells.
        Hide
    End If
End If
End Sub

Private Sub Form_Load()
    Dim Counter As Integer
    Dim fileName As String
    Dim code As Integer

    'Open fertilizer file and create fertilizer list.
    fileNum = FreeFile
    fileName = epicDir + "ferldata.dat"
    Open fileName For Input As fileNum
    Counter = 1
    Do
        If EOF(fileNum) Then
            Exit Do
        End If
    Loop

```

```

End If
Input #fileNum, code
'These codes do not have any operations.
If ((code < 6 Or code > 10) And (code < 16 Or code > 20) _
    And code <> 25 And (code < 27 Or code > 30) And code <> 35 _
    And code <> 37 And code <> 39 And code <> 40 And (code < 42 Or code > 49)) Then
    Input #fileNum, fertilizer(Counter)
    cboFert.AddItem fertilizer(Counter)
End If
Counter = Counter + 1
Loop While Not (EOF(fileNum) And Counter > 100)
cboFert.ListIndex = 0
Close fileNum
cboUnit.AddItem unitType(0)
cboUnit.AddItem unitType(1)
cboUnit.ListIndex = 0
End Sub

Private Sub txtAppRate_Change()
    validateText = txtAppRate.Text
    validateData
    txtAppRate.Text = validateText
End Sub

Private Sub txtFertDepth_Change()
    validateText = txtFertDepth.Text
    validateData
    txtFertDepth.Text = validateText
End Sub

Private Sub txtHUSched_Change()
    validateText = txtHUSched.Text
    validateData
    txtHUSched.Text = validateText
End Sub

'frmculvt.frm
' This form allows a user to select enter variable values related to cultivation operation
' and store them in the cell specific "mgmt.utl" files.

Private Sub cmdCancel_Click()
    Hide
End Sub

Private Sub cmdOk_Click()
    Dim Counter As Integer
    Dim resp As Integer
    Dim fileName As String

    'Store the management operation of a user chooses to.
    Counter = 1

```

```

    resp = MsgBox("Do you wish to store this operation?", vbYesNo + vbCritical + vbDefaultButton2,
"EPIC-View")
    If resp = vbYes Then
        Do
            cellFiles(Counter) = FreeFile
            fileName = "mgmt" + selectedCells(Counter) + ".utl"
            Open cellDatasetDir + fileName For Append As cellFiles(Counter)
            cellIndex(Counter) = cellIndex(Counter) + 1
            Print #cellFiles(Counter), "MON(" + Trim(Str(cellIndex(Counter))) + ") " + Str(monthSel)
            Print #cellFiles(Counter), "DAY(" + Trim(Str(cellIndex(Counter))) + ") " + Str(daySel)
            Print #cellFiles(Counter), "COD(" + Trim(Str(cellIndex(Counter))) + ") " + Str(operationCode)
            Print #cellFiles(Counter), "HUSC(" + Trim(Str(cellIndex(Counter))) + ") " + txtHUSC.Text
            Close cellFiles(Counter)
            Counter = Counter + 1
        Loop While Counter <= totCellsSel 'Loop for all selected cells.
    Hide
    End If
End Sub

Private Sub Form_Activate()
    activateForm
End Sub

Private Sub Form_Load()
    activateForm
End Sub

' ActivateForm
' Date: May 22, 1996
'-----
' activateForm Subroutine:
' - Activates the form with a particular title
' - so that same form can be used for more than
' - one management operation.
'-----

Public Sub activateForm()
    If operationCode = 19 Then
        frmCultivate.Caption = "EPIC-View - Row Cultivator"
    ElseIf operationCode = 21 Then
        frmCultivate.Caption = "EPIC-View - Hoe"
    ElseIf operationCode = 23 Then
        frmCultivate.Caption = "EPIC-View - Sweep"
    ElseIf operationCode = 29 Then
        frmCultivate.Caption = "EPIC-View - Disk"
    ElseIf operationCode = 30 Then
        frmCultivate.Caption = "EPIC-View - Chisel"
    ElseIf operationCode = 51 Then
        frmCultivate.Caption = "EPIC-View - Harvest"
    End If
End Sub

```

```
Private Sub txtHUSC_Change()
    validateText = txtHUSC.Text
    validateData
    txtHUSC.Text = validateText
End Sub
```

'Irrig.frm

' This form allows a user to enter values for variables related to irrigate operation and
' store the values in the cell specific "mgmt.utl" files.

Option Explicit

```
Private Sub cmdCancel_Click()
    Hide
End Sub
```

```
Private Sub cmdOk_Click()
    Dim Counter As Integer
    Dim resp As Integer
    Dim addStr As String
    Dim fileName As String
```

'Store the operation if a user chooses to.

```
Counter = 1
resp = MsgBox("Do you wish to store this operation?", vbYesNo + vbCritical + vbDefaultButton2,
"EPIC-View")
```

If resp = vbYes Then

```
If (UCase(cboUnit.Text) = "ENGLISH") Then
    addStr = "E"
```

Else

```
addStr = ""
```

End If

Do

```
cellFiles(Counter) = FreeFile
```

```
fileName = "mgmt" + selectedCells(Counter) + ".utl"
```

```
Open cellDatasetDir + fileName For Append As cellFiles(Counter)
```

```
cellIndex(Counter) = cellIndex(Counter) + 1
```

```
Print #cellFiles(Counter), "MON(" + Trim(Str(cellIndex(Counter))) + ") " + Str(monthSel)
```

```
Print #cellFiles(Counter), "DAY(" + Trim(Str(cellIndex(Counter))) + ") " + Str(daySel)
```

```
Print #cellFiles(Counter), "COD(" + Trim(Str(cellIndex(Counter))) + ") " + Str(operationCode)
```

```
Print #cellFiles(Counter), "IA(" + Trim(Str(cellIndex(Counter))) + ") " + txtIA.Text + addStr
```

```
Print #cellFiles(Counter), "QVOL(" + Trim(Str(cellIndex(Counter))) + ") " + txtQVol.Text
```

```
Close cellFiles(Counter)
```

```
Counter = Counter + 1
```

Loop While Counter <= totCellsSel 'Loop for all selected cells.

Hide

End If

```
End Sub
```

```
Private Sub Form_Load()
    cboUnit.AddItem unitType(0)
```

```

    cboUnit.AddItem unitType(1)
    cboUnit.ListIndex = 0
End Sub

```

```

Private Sub txtIA_Change()
    validateText = txtIA.Text
    validateData
    txtIA.Text = validateText
End Sub

```

```

Private Sub txtQVol_Change()
    validateText = txtQVol.Text
    validateData
    txtQVol.Text = validateText
End Sub

```

'Pest.frm

' This form allows a user to select a pesticide from a list of pesticides and other variable
' values related to sprayer operation and store them in the cell specific "mgmt.utl" files.

Option Explicit

```

Dim pesticide(1 To 300) As String
Dim fileNum As Integer

```

```

Private Sub cmdCancel_Click()
    Hide
End Sub

```

```

Private Sub cmdOk_Click()
    Dim Counter As Integer
    Dim ind As Integer
    Dim resp As Integer
    Dim addStr As String
    Dim fileName As String

```

'Get the pesticide code.

ind = 1

Do

 If pesticide(ind) = cboPest.Text Then

 Exit Do

 End If

 ind = ind + 1

Loop While ind < 100

Counter = 1

If (ind < 100) Then

 'Store the operation if a user chooses to.

 resp = MsgBox("Do you wish to store this operation?", vbYesNo + vbCritical + vbDefaultButton2,
"EPIC-View")

 If resp = vbYes Then

 If (UCase(cboUnit.Text) = "ENGLISH") Then

```

    addStr = "E"
Else
    addStr = ""
End If
Do
    cellFiles(Counter) = FreeFile
    fileName = "mgmt" + selectedCells(Counter) + ".utl"
    Open cellDatasetDir + fileName For Append As cellFiles(Counter)
    cellIndex(Counter) = cellIndex(Counter) + 1
    Print #cellFiles(Counter), "MON(" + Trim(str(cellIndex(Counter))) + ") " + str(monthSel)
    Print #cellFiles(Counter), "DAY(" + Trim(str(cellIndex(Counter))) + ") " + str(daySel)
    Print #cellFiles(Counter), "COD(" + Trim(str(cellIndex(Counter))) + ") " + str(operationCode)
    Print #cellFiles(Counter), "PST(" + Trim(str(cellIndex(Counter))) + ") " + str(ind)
    Print #cellFiles(Counter), "PCF(" + Trim(str(cellIndex(Counter))) + ") " + txtPCF.Text
    Print #cellFiles(Counter), "PAR(" + Trim(str(cellIndex(Counter))) + ") " + txtAppRate.Text +
addStr
    Close cellFiles(Counter)
    Counter = Counter + 1
    Loop While Counter <= totCellsSel 'Loop for all selected cells.
Hide
End If
End If
End Sub

Private Sub Form_Load()
    Dim Counter As Integer
    Dim fileName As String
    Dim pest As String
    Dim str As String

    'Open pesticide file and create a list of pesticides.
    fileNum = FreeFile
    fileName = epicDir + "usdapest.dat"
    Open fileName For Input As fileNum
    Counter = 1
    Do
        If EOF(fileNum) Then
            Exit Do
        End If
        pest = Input(16, fileNum)
        pesticide(Counter) = pest
        cboPest.AddItem pesticide(Counter)
        Input #fileNum, str
        Counter = Counter + 1
    Loop While Not (EOF(fileNum) And Counter > 300)
    cboPest.ListIndex = 0
    Close fileNum
    cboUnit.AddItem unitType(0)
    cboUnit.AddItem unitType(1)
    cboUnit.ListIndex = 0
End Sub

```



```
Private Sub txtAppRate_Change()
    validateText = txtAppRate.Text
    validateData
    txtAppRate.Text = validateText
End Sub
```

```
Private Sub txtPCF_Change()
    validateText = txtPCF.Text
    validateData
    txtPCF.Text = validateText
End Sub
```

'Rowplntr.frm

' This form allows a user to select a crop from a list of crops and other variable
' values related to rowplanter operation and store them in the cell specific "mgmt.utl" file.

Option Explicit

```
Dim Crop(1 To 100) As String
Dim fileNum As Integer
```

```
Private Sub cmdCancel_Click()
    Hide
End Sub
```

```
Private Sub cmdOk_Click()
    Dim Counter As Integer
    Dim ind As Integer
    Dim resp As Integer
    Dim fileName As String
```

```
'Get the crop code.
ind = 1
Do
    If Crop(ind) = cboCrop.Text Then
        Exit Do
    End If
    ind = ind + 1
Loop While ind < 100
```

```
'Store the operation if a user chooses to.
Counter = 1
If (ind < 100) Then
    resp = MsgBox("Do you wish to store this operation?", vbYesNo + vbCritical + vbDefaultButton2,
"EPIC-View")
    If resp = vbYes Then
        Do
            cellFiles(Counter) = FreeFile
            fileName = "mgmt" + selectedCells(Counter) + ".utl"
            Open cellDatasetDir + fileName For Append As cellFiles(Counter)
            cellIndex(Counter) = cellIndex(Counter) + 1
```

```

Print #cellFiles(Counter), "MON(" + Trim(Str(cellIndex(Counter))) + ") " + Str(monthSel)
Print #cellFiles(Counter), "DAY(" + Trim(Str(cellIndex(Counter))) + ") " + Str(daySel)
Print #cellFiles(Counter), "COD(" + Trim(Str(cellIndex(Counter))) + ") " + Str(operationCode)
Print #cellFiles(Counter), "CRP(" + Trim(Str(cellIndex(Counter))) + ") " + Str(ind)
Print #cellFiles(Counter), "GRZ(" + Trim(Str(cellIndex(Counter))) + ") " + Str(ind)
Print #cellFiles(Counter), "PHU(" + Trim(Str(cellIndex(Counter))) + ") " + txtPHU.Text
Close cellFiles(Counter)
Counter = Counter + 1
Loop While Counter <= totCellsSel 'Loop for all selected cells.
Hide
End If
End If
End Sub

```

```

Private Sub Form_Load()
Dim Counter As Integer
Dim fileName As String
Dim code As Integer
Dim crpCode As String
Dim crp As String

'Open the crop file and create the crops list.
fileNum = FreeFile
fileName = epicDir + "usdacrop.txt"
Open fileName For Input As fileNum
Counter = 1
Do
If EOF(fileNum) Then
Exit Do
End If
Input #fileNum, code
If (code <> 9 And code <> 29) Then
crpCode = Input(4, fileNum)
Input #fileNum, crp
Crop(Counter) = crp
cboCrop.AddItem Crop(Counter)
Else
Input #fileNum, crp
End If
Counter = Counter + 1
Loop While Not (EOF(fileNum) And Counter > 100)
cboCrop.ListIndex = 0
Close fileNum

End Sub

```

```

Private Sub txtPHU_Change()
validateText = txtPHU.Text
validateData
txtPHU.Text = validateText
End Sub

```

APPENDIX II.e

EPIC-View: Program Code for Spatial Data Tool

```

' Epic.getGISData
' This script retrieves elevation, slope, crop, soil series
' field values of the selected records from the theme's table.
' Mahesh Rao. Dated 2/27/96
'
' Reset any previously existing selectCells list.
totFiles=_selectCellsList.Count
index = totFiles - 1
while (index >= 0)
  _selectCellsList.Remove(index)
  index = index - 1
end
theTable = av.GetProject.FindDoc(_mainTable)
if (nil=theTable)then
  MsgBox.Error("The table: "+_mainTable+", not found.", "Epic")
  exit
end
theTableWin=theTable.GetWin
if (theTableWin.IsOpen.Not)then
  theTableWin.Open
else
  theTableWin.Activate
end
theTableWin.Minimize
'
theVTab = theTable.GetVTab
myVTab = theVTab.GetSelection
if (0=theVTab.GetSelection.Count) then
  MsgBox.Error("There are no cells selected to extract spatial attributes.", "EPIC-View")
  exit
end
'
soilField = theVTab.FindField("Series")
sortField = theVTab.FindField(_cellIdFld)
theTable.Sort(sortField, False)
'
'Show status bar
av.ShowMsg("creating files...")
canceled = False
'av.ShowStopButton
statusIndex = 0
'av.SetStatus (statusIndex)
selRecords=theVTab.GetNumSelRecords
statusIncrement = 100 / selRecords
'
for each rec in myVTab
  cellIdField = theVTab.FindField(_cellIdFld)

```

```

cellId = theVTab.ReturnValueString(cellIdField, rec)
aFileName=( _cellDatasetDir+"form"+cellId+".utl").AsFileName
aTextFile = TextFile.Make(aFileName, #FILE_PERM_WRITE)
_selectCellsList.Add(cellId)

```

```

'Get soil series
if(_userSoilsAbsent = False)then
  soilSeries = theVTab.ReturnValueString(soilField, rec)
  soilSeries = "@"+_soilDir+soilSeries+".utl"
  aTextFile.Write(soilSeries, soilSeries.Count)
  aTextFile.WriteElt(_newlineChar)
end

```

```

'Get elevation value
elevField = theVTab.FindField("Elev")
elev = theVTab.ReturnValueNumber(elevField, rec)
elev = "ELEV    "+elev.AsString
'Write to the text file
aTextFile.Write(elev.AsString, elev.AsString.Count)
aTextFile.WriteElt(_newlineChar)

```

```

'Get slope value
slopeField = theVTab.FindField("Slope")
slope = theVTab.ReturnValueNumber(slopeField, rec)
slope = slope / 100
slope="S      "+slope.AsString
'Write to the text file
aTextFile.Write(slope, slope.AsString.Count)
aTextFile.WriteElt(_newlineChar)

```

```

'Get area value
areaField = theVTab.FindField("Area")
area = theVTab.ReturnValueNumber(areaField, rec)
area = area / 10000
areaStr="WSA    "+area.AsString
'Write to the text file
aTextFile.Write(areaStr, areaStr.AsString.Count)
aTextFile.WriteElt(_newlineChar)

```

```

'Get Runoff Curve number value(if user specified
'soil is absent).
if(_userSoilsAbsent = False)then
  cNumField = theVTab.FindField("Cn2")
  cNum = theVTab.ReturnValueNumber(cNumField, rec)
  cNum="CN2    "+cNum.AsString
  'Write to the text file
  aTextFile.Write(cNum, cNum.AsString.Count)
  aTextFile.WriteElt(_newlineChar)
end

```

```

'Store slope length after calculating it.
area = area * 10000

```

```
side = area.Sqrt
SL = side * 2.Sqrt
SLStr = "SL      " + SL.AsString
aTextFile.Write(SLStr, SLStr.Count)
aTextFile.WriteElt(_newlineChar)

' Store pointer to cell's corresponding mgmt file.
mgmtFile = "@"+_cellDatasetDir+"mgmt"+cellId+".utl"
aTextFile.Write(mgmtFile, mgmtFile.Count)
aTextFile.WriteElt(_newlineChar)
aTextFile.Close
end

if(canceled) then
    av.ShowMsg("Process interrupted.")
else
    MsgBox.Info("Extracted Spatial data from selected cells.", "Epic")
end

' Enable the Output Options and Run Simulator menu options.
_opOptionEnableFlag = 1
_runEpicEnableFlag = True
```

APPENDIX II.f

EPIC-View: Program Code for Output Options Tool

```
' Epic.outputOptions
' This Avenue Script allows a user to define output options
' by providing an Output Options data entry user interface.
' Mahesh Rao. Dated 5/24/96
'
command = _exeDir+"outputop.exe"
system.execute(command)
' Disable Output Options menu option and enable Run Simulator menu option.
_opOptionEnableFlag = 0
_runEpicEnableFlag = 1

'Outputop.frm
' This VB form allows a user to select a list of variables for output.
' Also, a user can opt to select daily, monthly, yearly, annual or all these output
' files to be generated by EPIC. These are stored into "prnt.utl" file which are later loaded
' into "prnt5300.dat" file.
```

Option Explicit

```
Dim fileNum As Integer
Dim cellDatasetDir As String
Dim epicOutputDir As String
Dim soilDir As String
Dim epicDir As String
Dim outputVars(1 To 150) As String
Dim totVarsSel As Integer

Private Sub cmdCancel_Click()
    Dim resp As Integer

    resp = MsgBox("Do you wish to close?", vbYesNo + vbQuestion, "EPIC-View")
    If resp = vbYes Then
        End
    End If
End Sub

Private Sub cmdOk_Click()
    Dim Counter As Integer
    Dim ctr As Integer
    Dim resp As Integer
    Dim outputStr As String

    Counter = 0
    totVarsSel = 1
    outputStr = ""
    'Restore old settings/store new settings as choosen by a user.
    If (lstOpVars.SelCount > 0 Or optOldVal.Value = True) Then
```

```

resp = MsgBox("Do you wish to close?", vbYesNo + vbQuestion, "EPIC-View")
If resp = vbYes Then
    If optNewVal.Value = True Then
        fileNum = FreeFile
        'Writing a string for output files depending upon user choice.
        Open cellDatasetDir + "outfiles.dat" For Output As fileNum
        If Not (optDaily.Value = False And optMonthly.Value = False And optYearly = False _
            And optAnnual.Value = False And optAllFiles.Value = False) Then
            If optDaily.Value = True Then 'For daily output.
                outputStr = "-epd "
            ElseIf optMonthly.Value = True Then 'For monthly output.
                outputStr = "-epm "
            ElseIf optYearly.Value = True Then 'For yearly output.
                outputStr = "-epy "
            ElseIf optAnnual.Value = True Then 'For annual output.
                outputStr = "-epa "
            ElseIf optAllFiles.Value = True Then
                outputStr = "-ep " 'For all above outputs.
            End If
            Print #fileNum, outputStr
        Else 'For none of above outputs.
            Print #fileNum, "NONE"
        End If
        Close fileNum
        fileNum = FreeFile
        'Creating the output variables file.
        Open cellDatasetDir + "prnt.utl" For Output As fileNum
        'Writing for default values of daily and monthly outputs.
        Print #fileNum, "KD(1)  0"
        Print #fileNum, "KM(1)  0"
        'Writing the codes for all output variables selected.
        Do
            If lstOpVars.Selected(Counter) = True Then
                ctr = 1
                'Loop to get the variable's code.
                Do
                    If outputVars(ctr) = lstOpVars.List(Counter) Then
                        Exit Do
                    End If
                    ctr = ctr + 1
                Loop While ctr < 150
                If ctr < 150 Then
                    Print #fileNum, "KY(" + Trim(Str(totVarsSel)) + ")  " + Str(ctr)
                    totVarsSel = totVarsSel + 1
                End If
            End If
            Counter = Counter + 1
        Loop While (Counter < lstOpVars.ListCount And totVarsSel < 30)
        'If less than 30 variables were selected, write '0' for all
        'remaining variable places.
        If totVarsSel < 30 Then
            Do

```

```

        Print #fileNum, "KY(" + Trim(Str(totVarsSel)) + ") 0"
        totVarsSel = totVarsSel + 1
    Loop While totVarsSel <= 30
    End If
    Close fileNum
End If
End
End If
Else 'In case, no variable is selected.
    If optNewVal.Value = True Then
        resp = MsgBox("The output variables are not selected!", vbInformation, "EPIC-View")
    End If
End If
End Sub

```

```

Private Sub Form_Load()
    Dim Counter As Integer
    Dim fileName As String
    Dim ctr As Integer
    'Get directory paths.
    fileNum = FreeFile
    Open "c:\EVPaths.txt" For Input As fileNum
    Input #fileNum, cellDatasetDir, epicOutputDir, soilDir, epicDir
    Close fileNum

    'Open output variables file and create a list of output variables.
    fileNum = FreeFile
    fileName = epicDir + "opvarlst.dat"
    Open fileName For Input As fileNum
    Counter = 1
    totVarsSel = 0
    txtTotSel.Text = Str(lstOpVars.SelCount)
    Do
        If EOF(fileNum) Then
            Exit Do
        End If
        Input #fileNum, ctr, outputVars(Counter)
        lstOpVars.AddItem outputVars(Counter)
        Counter = Counter + 1
    Loop While Not (EOF(fileNum) And Counter > 150)
    Close fileNum

End Sub

```

```

Private Sub lstOpVars_Click()
    Dim resp As Integer

    txtTotSel.Text = (lstOpVars.SelCount)
    If lstOpVars.SelCount > 30 Then
        resp = MsgBox("More than 30 output variable(s) have been selected! Last selected variable(s) will be ignored.", vbCritical, "EPIC-View")
    End If

```


End Sub

Private Sub lstOpVars_DblClick()

Dim resp As Integer

txtTotSel.Text = (lstOpVars.SelCount)

If lstOpVars.SelCount > 30 Then

resp = MsgBox("More than 30 output variable(s) have been selected! Last selected variable(s) will be ignored.", vbCritical, "EPIC-View")

End If

End Sub

Private Sub optNewVal_Click()

optDaily.Enabled = True

optMonthly.Enabled = True

optYearly.Enabled = True

optAnnual.Enabled = True

optAllFiles.Enabled = True

lblOpVars.Enabled = True

lblTotSel.Enabled = True

lstOpVars.Enabled = True

Frame2.Enabled = True

End Sub

Private Sub optOldVal_Click()

optDaily.Enabled = False

optMonthly.Enabled = False

optYearly.Enabled = False

optAnnual.Enabled = False

optAllFiles.Enabled = False

lblOpVars.Enabled = False

lblTotSel.Enabled = False

lstOpVars.Enabled = False

Frame2.Enabled = False

End Sub

APPENDIX II.g

EPIC-View: Program Code for Model Run Tool

```

' Epic.runEpic
' This Script creates various batch files to complete the cell specific
' input datasets and then invoke EPIC on all of selected cells' input
' datasets and finally invoke the parser to create a comma delimited
' file from the EPIC output files.
' Mahesh Rao. Dated 2/28/96
'
' Check if any cells are selected to run EPIC.
cellsSelected= _selectCellsList.Count
if(0=cellsSelected)then
  MsgBox.Error("There are no cells selected to run the model.", "EPIC-View")
  exit
end
'
theTable = av.GetProject.FindDoc(_mainTable)
theTableWin=theTable.GetWin
theVTab = theTable.GetVTab
myVTab = theVTab.GetSelection
if (0=theVTab.GetSelection.Count) then
  MsgBox.Error("There are no cells selected.", "EPIC-View")
  exit
end
'
' Removes any fields joined to the current table
if (theVTab.IsBase.Not) then
  av.GetProject.SetModified(true)
end
theVTab.UnjoinAll
'
' Removes the added table "form.prs" from project.
theProject = av.GetProject
theTable = theProject.FindDoc(_resultsTable)
if(nil <> theTable) then
  theProject.RemoveDoc(theTable)
end
'
' Removes the added themes.
epicProject=av.getProject
fieldView=epicProject.FindDoc(_mainView)
if(fieldView.GetThemes.Count > 1) then
  totalThemes = fieldView.GetThemes.Count
  while(totalThemes > 1)
    themesList = fieldView.GetThemes
    for each aTheme in themesList
      if(aTheme.GetName <> _mainTheme) then
        fieldView.DeleteTheme(aTheme)
        totalThemes = totalThemes - 1
        break
      end
    end
  end
end

```

```

        end
    end
end
end
'
' If a user selected to have daily, monthly, yearly, annual of all of these
' EPIC output files, a string is written to a file "outfiles.dat" which
' is read here and appropriate command to run EPIC is formulated.
userString=""
userPreference=( _cellDatasetDir+"outfiles.dat").AsFileName
if(File.Exists(userPreference)) then
    stringFile=TextFile.Make(userPreference, #FILE_PERM_READ)
    while(stringFile.IsAtEnd.Not)
        listChar = stringFile.ReadElt
        if(listChar = 10.AsChar.AsString) then
            break
        else
            userString = userString + listChar
        end
    end
end
if (userString = "NONE") then ' If no output file is required.
    userString=""
end
end
'
' Creating various batch files for running.
makeDataset=( _cellDatasetDir.AsString+"create.bat").AsFileName
createFile = TextFile.Make(makeDataset, #FILE_PERM_WRITE)
runEpic=( _cellDatasetDir.AsString+"runepic.bat").AsFileName
runEpicFile = TextFile.Make(runEpic, #FILE_PERM_WRITE)
selectedCell=( _cellDatasetDir.AsString+"selected.cll").AsFileName
selectedFile = TextFile.Make(selectedCell, #FILE_PERM_WRITE)
path=_epicOutputDir
selectedFile.Write(path, path.Count)
selectedFile.WriteElt(_newLineChar)
'
' Show status bar
av.ShowMsg("creating files...")
canceled = False
av.ShowStopButton
statusIndex = 0
av.SetStatus (statusIndex)
totFiles=_selectCellsList.Count
statusIncrement = 100 / totFiles
'
' Writing commands in the batch files.
for each cellId in _selectCellsList
    command=_epicDir+"ewq "+userString+_cellDatasetDir+"form"+cellId+" "+_epicOutputDir
    selectedFile.Write(cellId, cellId.Count)
    selectedFile.WriteElt(_newLineChar)
    runEpicFile.Write(command, command.Count)
    runEpicFile.WriteElt(_newLineChar)
end

```

```

command=_epicDir+"util epic "+_cellDatasetDir+"form"+cellId+".dat @"+_
_cellDatasetDir+"form"+cellId+".utl"
createFile.Write(command, command.Count)
createFile.WriteElt(_newlineChar)

statusIndex = statusIndex + statusIncrement
continued = av.SetStatus (statusIndex)
if(Not continued) then
  canceled = true
  break
end
end

if(canceled) then
  av.ShowMsg("Process interrupted.")
else
  av.ShowMsg("Created command file for running simulator.")
end

outputOptionString=(_epicDir+"util prnt prnt5300.dat @"+_cellDatasetDir+"prnt.utl").AsFileName
if (File.Exists((_cellDatasetDir+"prnt.utl").AsFileName)) then
  createFile.Write(outputOptionString.AsString, outputOptionString.AsString.Count)
  createFile.WriteElt(_newlineChar)
end
createFile.Write(runEpic.AsString, runEpic.AsString.Count)
createFile.WriteElt(_newlineChar)
createFile.Close
command=_exeDir+"parse.exe"
runEpicFile.Write(command, command.Count)
runEpicFile.WriteElt(_newlineChar)
runEpicFile.Close
selectedFile.Close
_epicDir.AsFileName.setCWD

```

' Invoking a waitshell to run EPIC.
command = _exeDir+"DSETMAKE.EXE Beavis Epic.returnToAV"
system.execute(command)
' Disabling Run Simulator menu option.
_runEpicEnableFlag = False
' Enabling the Display Map menu option.
_displayEnableFlag=True

' Silly.bas

' This VB program allows a called program (dsetmake.exe) from ArcView to be executed in a shell.

Option Explicit

```

Global Text1 As TextBox
Global fileNum As Integer
Global cellDatasetDir As String
Global epicOutputDir As String

```

Global soilDir As String
Global epicDir As String

Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModule%)

```

-----
' Beavis
' Date: February 22, 1996
-----
' Subroutine: Beavis
' - Use waitshell to make a synchronous call
'
Sub Beavis(ArgPassed As String)
    Dim fileName As String

    ' Call the batch file to run in shell.
    fileName = cellDatasetDir + "create.bat"
    WaitShell (fileName)
End Sub

-----
' Main
' Date: February 22, 1996
-----
' Main Subroutine:
' - Accepts command-line arguments and calls
' - the corresponding subroutine, or returns
' - an error message.
'
Sub Main()

    ReDim Argv(3) As String
    Dim SpcPos, OldPos As Integer
    Dim Argc As Integer
    Dim i, x As Integer
    Dim Text1 As TextBox

    Set Text1 = Form1!Text1

    Debug.Print Command
    MsgBox "Silly Main: Command = " & Command

    ' Store Command-Line Args in Argv Array.
    OldPos = 1
    Argc = 1
    SpcPos = InStr(OldPos, Command, " ")
    While (SpcPos)
        Argv(Argc) = Mid(Command, OldPos, SpcPos - OldPos)
        OldPos = SpcPos + 1
        SpcPos = InStr(OldPos, Command, " ")
        Argc = Argc + 1
    
```

```

Wend
Argv(Argc) = Mid(Command, OldPos, 1 + Len(Command) - OldPos)

Debug.Print "Arguments: "
For i = 1 To Argc
    Debug.Print Argv(i)
Next i

fileNum = FreeFile
Open "c:\EVPaths.txt" For Input As fileNum
Input #fileNum, cellDatasetDir, epicOutputDir, soilDir, epicDir
Close fileNum

Select Case Argv(1)

    Case "Beavis"
        Debug.Print "Function BEAVIS."
        Beavis Argv(2)

    Case Else    ' Must be something else.
        Debug.Print "Error."

End Select
End Sub

'-----
' WaitShell
' Date: February 22, 1996
'-----
' Subroutine: WaitShell
' - Make a synchronous call.
'-----
Sub WaitShell(ByVal AppName As String)
    Dim hMod As Integer
    hMod = Shell(AppName, 1)
    If (Abs(hMod) > 32) Then
        While (GetModuleUsage(hMod))
            DoEvents
        Wend
    Else
        Debug.Print "Unable to start " & AppName
    End If
End Sub

```

APPENDIX II.h

EPIC-View: Parser Program Code

```

// Parse.c
/*****
*
*   This program creates a file with comma delimited records which can be loaded back into
*   ArcView as a table.
*
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Globals!!
char lastChar, CELL_ID_FIELD[20], PRS_FILE[20];
long filePosition=0;
int newLines=0, print=0, firstFiveFields=1, doubleQuotes=0;
int ignoreMoreSpaces=0, numOfCommas=0;
void insertComma(FILE *, char);
int writeToFile(char *, char *, int);
void displayQuit(void);

void main()
{
    int flag=1, retVal;
    FILE *fp, *fpath;
    char path[80], fsel[80], id[10];
    char cellDir[20], epicOutputDir[20], soilDir[20], epicDir[20], exeDir[20], baseData[20];

    strcpy(fsel, "c:\\EVPpaths.txt");
    if(!(fpath = fopen(fsel, "r")))
    {
        printf("File %s not found. Aborting...\n", fsel);
        exit(0);
    }
    fscanf(fpath, "%s%s%s%s%s%s%s%s", cellDir, epicOutputDir, soilDir, epicDir, exeDir,
           baseData, CELL_ID_FIELD, PRS_FILE);
    strcpy(fsel, cellDir);
    strcat(fsel, "selected.cll");
    if(!(fp = fopen(fsel, "r")))
    {
        printf("File %s not found. Aborting...\n", fsel);
        exit(0);
    }
    fscanf(fp, "%s", path);

    while(!feof(fp))
    {
        fscanf(fp, "%s", id);

```

```

        if(strlen(id)==0)break;
        retVal = writeToFile(path, id, flag);
        if(retVal == -1)
        {
            printf("Error encountered while parsing. Interrupted in
middle!\nAborting...\n");
            displayQuit();
            fclose(fp);
            fclose(fpath);
            exit(0);
        }
        flag=0;
        strcpy(id, "");
    }
    printf("Successfully completed parsing.\n");
    displayQuit();
    fclose(fp);
    fclose(fpath);
}

```

```

/*****
*   writeToFile()
*
*   This function creates single file with comma delimited records from file(s) created
*   by EPIC as output.
*
*****/

```

```

int writeToFile(char *path, char *cellId, int headerFlag)
{
    int firstTime=1;
    long pos=-2;
    FILE *fileIn, *fileOut;
    char ch=' ', fin[80], fout[80];

    sprintf(fin, "%sform%s.sum", path, cellId);
    sprintf(fout, "%s%s", path, PRS_FILE);
    if(!(fileIn = fopen(fin, "r")))
    {
        printf("File %s not found. Aborting...\n", fin);
        exit(0);
    }

    if(headerFlag)
        fileOut= fopen(fout, "w");
    else
        fileOut= fopen(fout, "a");

    if(headerFlag)
        fprintf(fileOut, "%s\n", CELL_ID_FIELD);
    else

```



```

{
    while(fgetc(fileIn) != '\n')
        if(!feof(fileIn))
        {
            printf("File %s is empty!\n", fin);
            return -1;
        }
    newLines++;
}
while(!feof(fileIn))
{
    lastChar=ch;
    fscanf(fileIn, "%c", &ch);
    if (firstTime && newLines)
    {
        fprintf(fileOut, "%d,", atoi(cellId));
        firstTime=0;
        ungetc(ch, fileIn);
        ch=' ';
    }
    else
        insertComma(fileOut, ch); // Create a ',' delimited file.
    if(ch == '\n') newLines++;
}
fclose(fileIn);
fclose(fileOut);
return 1;
}

/*****
*   insertComma()
*
*   This function processes each character read from the input file(s) and takes action
*   such as inserting ',', ignoring space, writing the character read, etc. depending upon
*   various factors such as the character read, previous read character, etc.
*
*****/
void insertComma(FILE *out, char ch)
{
    if(lastChar=='\n')//reset number of commas added.
        numOfCommas=0;
    if( (doubleQuotes) && (ch != ' ') ) //update fileptr if in middle of a quote.
        filePosition--;
    if( (ch=="") && (doubleQuotes == 1) ) // Inserting ','
    {
        ignoreMoreSpaces=0;
        doubleQuotes=0;
        fprintf(out, "\",");
        numOfCommas++;
        filePosition=0;
    }
}

```

```

else if( (ch=="") && (doubleQuotes == 0) ) // Register first "
{
    if(lastChar != "" && lastChar != ' ' && lastChar != '\n')
    {
        ignoreMoreSpaces=0;
        doubleQuotes=1;
        fprintf(out, "%c", ch);
        filePosition=0;
    }
    else
    {
        ignoreMoreSpaces=0;
        doubleQuotes=1;
        fprintf(out, "%c", ch);
        filePosition=0;
    }
}
else if( (ch=="\n") && (doubleQuotes == 1) ) // Missing second "
{
    ignoreMoreSpaces=0;
    doubleQuotes=0;
    filePosition--; //displacement for an extra ' ' added.
    fseek(out, filePosition, SEEK_CUR); //wrap back and write \n.
    fprintf(out, "%c", ch);
}
else if( (ch=="\n") && (doubleQuotes == 0) && (lastChar == "" ) // To avoid writing ' ' after last
// field.
{
    ignoreMoreSpaces=0;
    doubleQuotes=0;
    filePosition--; //displacement for an extra ' ' added.
    fseek(out, filePosition, SEEK_CUR); //wrap back and write \n.
    fprintf(out, "%c", ch);
}
else if( (ch == ' ') && /*(newLines > 0) &&*/ (!ignoreMoreSpaces) //Insert ' ' for numerical
fields.
&& (doubleQuotes==0) && (lastChar != '\n') && (lastChar != ' ') && (lastChar !=
""))
{
    ignoreMoreSpaces=1;
    fprintf(out, ",");
    numOfCommas++;
}
else if(lastChar == ' ' && ch == '.') // Add 0 if a float starts with a decimal pt only.
{
    ignoreMoreSpaces=0;
    fprintf(out, "0%c", ch);
}
else if((ch != ' ') && !(ch == '\n' && lastChar == '\n') ) // Ignore all other spaces.
{
    ignoreMoreSpaces=0;

```

```

        fprintf(out, "%c", ch);
    }
}

/*****
 *   displayQuit()
 *
 *   This function displays a message on the screen to prompt the users to close the dos
 *   shell window.
 *****/
void displayQuit(void)
{
    printf("\n\n          *****\n");
    printf("          *\n");
    printf("          * PLEASE CLOSE THIS WINDOW BY *\n");
    printf("          *\n");
    printf("          * CLICKING ON THE EXIT BUTTON. *\n");
    printf("          *\n");
    printf("          *****\n");
}

```

APPENDIX II.i

EPIC-View: Program Code for Display Tool

```

' Epic.displayMap
' This Avenue script loads into the project the results table created by the parser, joins it with the main
' theme's attribute table and creates new themes depending upon the output variables selected
' by a user. The themes are displayed in the current field view.
' Mahesh Rao. Dated 5/14/96
,
,
' Add the comma delimited file created by parser as a new table in the project and join it with
' the main attribute table on Cell Id.
theTable = av.GetProject.FindDoc(_mainTable)
theTableWin=theTable.GetWin
,
' Removes any fields joined to the current table
theVTab = theTable.GetVTab
if (theVTab.IsBase.Not) then
  av.GetProject.SetModified(true)
end
theVTab.UnjoinAll
,
resTable = av.GetProject.FindDoc(_resultsTable)
if(resTable = nil) then
  f=(_epicOutputDir+_resultsTable).AsFileName
  v = VTab.Make(f, FALSE, FALSE)
  if (v.HasError) then
    MsgBox.Error("The file "" + f.GetBaseName + "" is not valid.", "")
  else
    t = Table.Make(v)
    t.SetName(v.GetName)
    tField = t.GetVTab.FindField(_cellIdFld)
    t.SetActiveField(tField)
    tableField = theTable.GetVTab.FindField(_cellIdFld)
    theTable.SetActiveField(tableField)
    theTable.GetVTab.Join(tableField,t.GetVTab,tField)
  end
end
else
  resField = resTable.GetVTab.FindField(_cellIdFld)
  resTable.SetActiveField(resField)
  tableField = theTable.GetVTab.FindField(_cellIdFld)
  theTable.SetActiveField(tableField)
  theTable.GetVTab.Join(tableField,resTable.GetVTab,resField)
end
' Replicate main theme into new themes depending upon output variables selected.
epicProject=av.getProject
fieldView=epicProject.FindDoc(_mainView)
' To make sure that only main theme is active.
for each aTheme in fieldView.GetThemes
  if(aTheme.GetName = _mainTheme) then
    aTheme.SetActive(True)

```

```

else
  aTheme.SetActive(False)
end
end
fieldView.CopyThemes
' Initialize theme colors.
resTable = av.GetProject.FindDoc(_resultsTable)
r1 = 200
g1 = 200
b1 = 250
r2 = 250
g2 = 150
b2 = 150
for each aField in theTable.GetVTab.GetFields
  fieldExists = resTable.GetVTab.FindField(aField.GetAlias)
  if( (fieldExists <> nil) And (fieldExists.GetAlias <> _cellIdFld) And (fieldExists.GetAlias <> "")
    And (fieldExists.GetAlias <> "Years") And (fieldExists.GetAlias <> "field") )then
    fieldView.Paste
    for each aTheme in fieldView.GetThemes
      if(aTheme.GetName = _mainTheme) then
        aTheme.SetActive(True)
      else
        aTheme.SetActive(False)
      end
    end
    resultTheme = fieldView.FindTheme(_mainTheme)
    resultTheme.SetName(aField.GetAlias)
    resultLegend = resultTheme.GetLegend
    resField = resultTheme.GetFTab.FindField(aField.GetAlias)
    resultLegend.Quantile(resultTheme.GetFTab, resField, 5)
    resultLegend.SetField(resField)
    startColor = Color.Make
    endColor = Color.Make
    startColor.SetRgbList({r1, g1, b1}) '200, 200, 250
    endColor.SetRgbList({r2, g2, b2}) '250, 150, 150
    resultLegend.RampColors(startColor, endColor)
    ' Change colors for the next theme.
    r1 = r1 + 30
    b1 = b1 + 20
    g1 = g1 + 10
    if (r1 > 255) then
      r1 = 0 + (r1 - 255)
    end
    if (b1 > 255) then
      b1 = 0 + (b1 - 255)
    end
    if (g1 > 255) then
      g1 = 0 + (g1 - 255)
    end
    r2 = r2 + 70
    b2 = b2 + 60
    g2 = g2 + 50
  end
end

```

```

if (r2 > 255) then
  r2 = 0 + (r2 - 255)
end
if (b2 > 255) then
  b2 = 0 + (b2 - 255)
end
if (g2 > 255) then
  g2 = 0 + (g2 - 255)
end
resultTheme.SetVisible(False)
if (resultTheme.Is( FTHEME )) then
  sel = resultTheme.GetFTab.GetSelection
  sel.ClearAll 'Clear all selections of added themes.
  resultTheme.GetFTab.UpdateSelection
end
end
end

' Epic.DispChart
' This script displays Chart as per user choices.
' Dated 5/19/96
'
' Create an output variables list.
outputVarList = List.Make
theTable = av.GetProject.FindDoc(_mainTable)
resTable = av.GetProject.FindDoc(_resultsTable)
for each aField in resTable.GetVTab.GetFields
  if( (aField.GetAlias <> _cellIdFld) And (aField.GetAlias <> "")
    And (aField.GetAlias <> "Years") And (aField.GetAlias <> "Botch.da" ) )then
    outputVarList.Add(aField.GetAlias)
  end
end
' Display the chart properties option.
aChart = Chart.MakeUsingDialog(resTable.GetVTab)
if(aChart <> nil) then
  aChart.GetWin.Open
end

' Epic.displayTable
' This script opens and displays the results table.
' Dated 5/23/96
'
epicProject=av.getProject
resTable=epicProject.FindDoc(_resultsTable)
resWin=resTable.GetWin
if (resWin.IsOpen.Not)then
  resWin.Open
else
  resWin.Activate
end

```

APPENDIX II.j

EPIC-View: Miscellaneous Program Codes

```
' Epic.startUp
' This script creates global variables for various directory paths set by user.
' Mahesh Rao. Dated 5/15/96

' Global variables used for enabling various menu options.
_selectCellsList = List.Make
_newLineChar = 10.AsChar
_displayEnableFlag = False
_soilDataIsEnabled = False
_constEnableFlag = 0
_soilEnableFlag = False
_runEpicEnableFlag = 0
_opOptionEnableFlag = 0
_userSoilIsAbsent = True
_totalCells = 1
pathsFile = "c:\EVPaths.txt".AsFileName
if(File.Exists(pathsFile).Not) then
  labelList = List.Make
  labelList.Add("Cell Dataset Directory:")
  labelList.Add("Epic Output Directory:")
  labelList.Add("Soil Data Directory:")
  labelList.Add("EPIC Directory:")
  labelList.Add("EXE Directory Name:")
  labelList.Add("Base Dataset Name:")
  labelList.Add("Cell Id Field Name:")
  labelList.Add("Results Table Name:")
  labelList.Add("Main Attribute Table Name:")
  labelList.Add("Main Field View Name:")
  labelList.Add("Main Theme Name:")
  labelList.Add("Main Theme Path:")

  defaultList = List.Make
  defaultList.Add("c:\EPICView\Temp")
  defaultList.Add("c:\EPICView\Temp")
  defaultList.Add("c:\EPICView\Soil")
  defaultList.Add("c:\epic5300")
  defaultList.Add("c:\EPICView\EXEDir")
  defaultList.Add("const.dat")
  defaultList.Add("Hru2_")
  defaultList.Add("form.prs")
  defaultList.Add("Attributes of Hru2")
  defaultList.Add("Botchlet 1/4 section")
  defaultList.Add("Hru2")
  defaultList.Add("c:\EPICView\Hru")

  userList=MsgBox.MultiInput("Globals Initialization:", "EPIC-View", labelList, defaultList)
```

```

if(userList = nil) then
  _cellDatasetDir = defaultList.Get(0)
  _epicOutputDir = defaultList.Get(1)
  _soilDir = defaultList.Get(2)
  _epicDir = defaultList.Get(3)
  _exeDir = defaultList.Get(4)
  _baseDataset = defaultList.Get(5)

  ' New additions
  _cellIdFld = defaultList.Get(6)
  _resultsTable = defaultList.Get(7)
  _mainTable = defaultList.Get(8)
  _mainView = defaultList.Get(9)
  _mainTheme = defaultList.Get(10)
  _mainThemePath= defaultList.Get(11)

else
  _cellDatasetDir = userList.Get(0)
  _epicOutputDir = userList.Get(1)
  _soilDir = userList.Get(2)
  _epicDir = userList.Get(3)
  _exeDir = userList.Get(4)
  _baseDataset = userList.Get(5)

  ' New additions
  _cellIdFld = userList.Get(6)
  _resultsTable = userList.Get(7)
  _mainTable = userList.Get(8)
  _mainView = userList.Get(9)
  _mainTheme = userList.Get(10)
  _mainThemePath= userList.Get(11)
end

'write to paths file.
pathFile = TextFile.Make(pathsFile, #FILE_PERM_WRITE)
pathFile.Write(_cellDatasetDir, _cellDatasetDir.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_epicOutputDir, _epicOutputDir.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_soilDir, _soilDir.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_epicDir, _epicDir.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_exeDir, _exeDir.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_baseDataset, _baseDataset.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_cellIdFld, _cellIdFld.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_resultsTable, _resultsTable.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_mainTable, _mainTable.Count)
pathFile.WriteElt(_newLineChar)

```



```

pathFile.Write(_mainView, _mainView.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_mainTheme, _mainTheme.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Write(_mainThemePath, _mainThemePath.Count)
pathFile.WriteElt(_newLineChar)
pathFile.Close
' Create a new view at the time of installation.
fieldView = View.Make
theSrcName = SrcName.Make( _mainThemePath+" polygon" )
if (theSrcName = nil) then
  msgbox.Error( "Invalid SrcName", "" )
  exit
end
mainTheme = Theme.Make( theSrcName )
mainTheme.SetActive(True)
mainTheme.SetVisible(True)
mainThemeLegend = mainTheme.GetLegend
aField = mainTheme.GetFTab.FindField("Elev")
mainThemeLegend.Interval(mainTheme.GetFTab, aField, 5)
mainThemeLegend.SetField(aField)
mainThemeLegend.RampColors(Color.GetBlue, Color.GetCyan)
mainTheme.SetName(_mainTheme)
fieldView.AddTheme(mainTheme)
fieldView.SetName(_mainView)
epicProject = av.GetProject
epicProject.AddDoc(fieldView)
mainTheme.EditTable
av.GetProject.Save
else
  eachItem=""
  items=0
  pathFile = TextFile.Make(pathsFile, #FILE_PERM_READ)
  while(pathFile.IsAtEnd.Not)
    aChar = pathFile.ReadElt
    if(aChar = 10.AsChar.AsString) then
      if(items = 0)then
        _cellDatasetDir = eachItem
      elseif(items = 1)then
        _epicOutputDir = eachItem
      elseif(items = 2)then
        _soilDir = eachItem
      elseif(items = 3)then
        _epicDir = eachItem
      elseif(items = 4)then
        _exeDir = eachItem
      elseif(items = 5)then
        _baseDataset = eachItem
      elseif(items = 6)then
        _cellIdFld = eachItem
      elseif(items = 7)then
        _resultsTable = eachItem
    end
  end
end

```

```

elseif(items = 8)then
  _mainTable = eachItem
elseif(items = 9)then
  _mainView = eachItem
elseif(items = 10)then
  _mainTheme = eachItem
elseif(items = 11)then
  _mainThemePath = eachItem
end
eachItem=""
items = items + 1
else
  eachItem = eachItem + aChar
end
end 'end of while loop
end ' End of main if condition
'
' Check if user specified soil is present (used for update property
' of soil data tool option). Also calculate the total number of cells
' present in the gridded coverage.
'
theTable = av.GetProject.FindDoc(_mainTable)
if(nil=theTable)then
  MsgBox.Error("The table: "+_mainTable+", not found.", "Epic")
  exit
end
theTableWin=theTable.GetWin
if (theTableWin.IsOpen.Not)then
  theTableWin.Open
else
  theTableWin.Activate
end
theTableWin.Minimize
'
theVTab = theTable.GetVTab
soilField = theVTab.FindField("Series")
if(soilField <> nil)then
  for each rec in theVTab
    soilSeries = theVTab.ReturnValueString(soilField, rec)
    if(soilSeries.IsNull.Not)then
      _userSoilIsAbsent = False
    end
    _totalCells=_totalCells+1
  end
else
  for each rec in theVTab
    _totalCells=_totalCells+1
  end
end
end
theTableWin.Close

```

```

' Epic.updateConstData
' To enable Constant Data Tool menu option.
' Dated 5/22/96
,
if (_constEnableFlag = 2)then
  Self.SetEnabled(True)
  exit
elseif (_constEnableFlag = 1 And _userSoilsAbsent.Not)then
  Self.SetEnabled(True)
  exit
else
  Self.SetEnabled(False)
  exit
end

```

```

' Epic.updateGISselection
' To update the menu option Spatial Data.
' Dated 2/27/96
,
epicProject=av.GetProject
fieldView=epicProject.FindDoc(_mainView)
,
,
if(nil=fieldView) then
  Self.SetEnabled(False)
  exit
elseif(fieldView.IsActive) then
  layersTheme=fieldView.FindTheme(_mainTheme)
  if(nil=layersTheme) then
    Self.SetEnabled(False)
    Exit
  elseif(layersTheme.IsVisible) then
    Self.SetEnabled(True)
    exit
  else
    Self.SetEnabled(False)
    exit
  end
else
  Self.SetEnabled(False)
  Exit
end
end

```

```

' Epic.updateRunEpic
' To enable Run Epic menu option.
' Dated 5/22/96
,
if(_runEpicEnableFlag = 1) then
  Self.SetEnabled(True)
  exit
else

```

```

Self.SetEnabled(False)
exit
end

```

```

' Epic.updateDispChart
' To update the menu option Chart in Display
' Dated 5/22/96
'
epicProject=av.getProject
ResTable=epicProject.FindDoc(_resultsTable)
if(resTable <> nil) then
  loadedResultsTable = True
else
  loadedResultsTable = False
end
fieldView=epicProject.FindDoc(_mainView)
if( (loadedResultsTable) And (_displayEnableFlag) ) then
  Self.SetEnabled(True)
  exit
else
  Self.SetEnabled(False)
  exit
end

```

```

' Epic.updateDispMap
' To update the menu option Map in Display
' Dated 5/22/96
'
epicProject=av.getProject
fieldView=epicProject.FindDoc(_mainView)
if( (fieldView.GetThemes.Count = 1) And (_displayEnableFlag) ) then
  Self.SetEnabled(True)
  exit
else
  Self.SetEnabled(False)
  exit
end

```

```

' Epic.updateDispTable
' To update the menu option Table in Display
' Dated 5/22/96
'
epicProject=av.getProject
ResTable=epicProject.FindDoc(_resultsTable)
if(resTable <> nil) then
  loadedResultsTable = True
else
  loadedResultsTable = False
end

```

```
fieldView=epicProject.FindDoc(_mainView)
if( (loadedResultsTable) And (_displayEnableFlag) ) then
  Self.SetEnabled(True)
  exit
else
  Self.SetEnabled(False)
  exit
end
```

```
' Epic.updateOpOption
' To enable Output Options menu option.
' Dated 5/22/96
'
```

```
if(_opOptionEnableFlag = 1) then
  Self.SetEnabled(True)
  exit
else
  Self.SetEnabled(False)
  exit
end
```

```
' Epic.updateSoilData
' To enable Soil Data Tool menu option.
' Dated 5/22/96
'
```

```
if(_userSoilsAbsent And _soilEnableFlag)then
  Self.SetEnabled(True)
  _soilsEnabled=True
  exit
else
  Self.SetEnabled(False)
  exit
end
```

APPENDIX III.a

EPIC-View: Base Data Input Values for the Various Parameters

Parameter	Description	Value
ACW	Wind erosion adjustment factor 0, wind erosion shut off; 1, normal wind erosion considered	1
BUS (1-4)	Four Parameter estimates for MUSI erosion equation *	0
CO2*	Carbon-dioxide conc. in the atmosphere Default to 350 ppm (current level)	350
CSALT*	Conc. of salt in irrigation water For future use in salinity submodel. Default to 0.	0
DRV	Equation for water erosion Options [0-5]; Refer User's Guide for equations Default 2 for USLE	2
IDA	Beginning day of simulation	1
IET	Potential Evapotranspiration equation * Options [0-4], Refer User's guide for equations. Default 0 for Penman-Monteith equation	0
IGRAF	Graphic display on/off 0, display off; 1, display on	0
IHUS	Automatic heat unit scheduling 0, Normal operation; 1, automatic heat unit scheduling	0
IMO	Beginning month of simulation	1
ISCN	Stochastic CN Estimator code*	0
ISTA	Static soil code* 0, Varying soil profile; 1, static soil	0
ITYP	Peak rate estimation code* Options [0-4], Refer User's guide	3
IYR	Beginning Year of simulation	91
LPYR	Leap year considered* 0, consider; 1, ignore	0
NBYR	Number of years of simulation duration	5
PEC	Erosion control practice factor 0, Total erosion control; 1, no erosion control practices	1
RTN*	Number of years of cultivation before simulation starts	100
CF*	Climatic factor for wind erosion Refer wind erosion component (EPIC manual, Vol. I) for values	0
IGN	Number of times the random number generator cycles*	0
IGSD*	Day weather stops generating the same weather	0
NGN	Weather input code Options [0-5, 23, 2345] Refer User's guide 0, generates all weather ; 2345 read all weather	0
RCN*	Average concentration of N in rainfall	0.8
SNO*	Water content of snow at start of simulation	0
SWV*	Power of modified exponential distribution of wind speed. Range (0.3 - 0.5). Default to 0.5 as recommended.	0.5

Parameter	Description	Value
ANG*	Clockwise angle of field length from north (deg.)	0
APM*	Peak runoff rate - Rainfall energy adjustment factor Value of 1 gives satisfactory results. Default to 1	1
CHD	Channel Depth	0
CHL*	Distance from outlet to most distant point on watershed	0
CHN*	Channel roughness factor (Manning's N)	0
CHS*	Average channel slope	0
FL*	Field length (Km or miles)	0
FW*	Field width	0
SN*	Surface roughness factor (Manning's N). Refer User's guide for suggested values.	0.09
STD*	Standing dead crop residue (T/ha or t/ac)	
WSA	Watershed drainage area (ha) $WSA = \text{resolution}^2 / 10000$	
YLT	Latitude of watershed (degrees); for daylength estimation	35.21

* Default to zero to let EPIC estimate the value (Dumesnil 1993)

APPENDIX III.b

EPIC-View: Management Data Values for the Various Parameters

Parameter	Description	Value
ARMN*	Minimum single application for automatic irrigation If unknown or rigid irrigation selected (NIRR= 1), set ARMN =0	0
ARMX*	Maximum single application volume for automatic irrigation (mm). This is the amount applied if rigid automatic irrigation is selected. 0 if unknown	0
BFT	N stress factor to trigger automatic fertilizer	0
BIR	Water stress factor to trigger irrigation automatically. For example, 0.85 means that irrigation will be triggered when the biomass production on that day is less than 85% of the potential biomass that could have been produced had water been available	0
DRT	Time required for drainage system to reduce plant stress 0, if drainage not considered	0
EFI	Irrigation runoff ratio	0
FMX	Maximum annual N fertilizer application 0, defaults to 200 kg ha ⁻¹	0
FNP	Amount of fertilizer (IDFT) per automatically scheduled application 0, for manual fertilizer option	0
IDFT	Fertilizer ID # for fertigation or automatic fertilization 0, no automatic fertilization or fertigation	0
IDR*	Drainage code	0
IFA*	Manual fertilizer application interval	0
IFD*	Furrow dike code	0
IFFR*	Automatic fertilizer rigidity code	0
IRI*	Minimum application interval for automatic irrigation.	0
IRR*	Irrigation code Options 0-3. Botchlet has center pivot sprinkler, 1	1
LM*	Liming code	0
NIRR	Rigidity of irrigation code 0, flexible application; 1. rigid application	1
NRO	Crop rotation duration Range (1-30 years)	1
VIMX	Maximum irrigation volume for each crop 24 in. (600mm) assumed to be applied for each crop per year	600
COD	Management operation code Specify type of tillage, field equipment characteristics stored in EPIC's tillage file (CLASSTILL.DAT). Refer User's guide (Section D.3)	&
CRP	Crop ID # Specify type of crop from EPIC's crop file (USDACROP.DAT)	&
DAY	Day of the operation	&
FAP	Fertilizer application rate (kg ha ⁻¹)	&
FN	Fertilizer ID #	&
GRZ	Grazing duration in days	&

Parameter	Description	Value
HUSC	Heat unit scheduling. Time of this operation as a fraction of the growing season or of the year. If no crop is growing; fraction of annual heat units accumulated using 0° as the base temperature	&
IA	Irrigation volume (mm) Specify for manual irrigation	&
MON	Month of the operation	&

& - These parameter values are specific to the management operation. See Appendix III.c for the management practices data specific to the farm.

APPENDIX III.c

EPIC-View: Botchlet Farm Management Data - Corn Soybean Rotation under Conventional Tillage System

MONTH	DAY	OPERATION DESCRIPTION
3	21	Field chiseled.
3	26	Field disked.
3	28	Field fertilized. Anhydrous ammonia applied at the rate of 125 lb ac ⁻¹ depth of 6 in.
4	5	Field disked.
4	12	Used Sweep implement on the field.
4	12	Field fertilized. 100 lb ac ⁻¹ of 10-20-20 at a depth of 6 in..
4	12	Planted corn. Potential heat units for maturity =1466
4	12	Pesticide applied. Dual at the rate of 1.87lb ac ⁻¹ . Pest control factor 1.0
4	13	Field irrigated. 0.75 in. of water applied.
4	25	Field hoed
5	18	Field fertilized. Anhydrous ammonia applied at the rate of 125lb ac ⁻¹ depth of 6 in..
5	19	Field irrigated. 0.75 in. of water applied.
5	25	Field cultivated.
5	27	Field irrigated. 1.2 in. of water applied.
6	17	Field irrigated. 1.2 in. of water applied.
6	30	Field irrigated. 1.2 in. of water applied.
7	10	Field irrigated. 1.2 in. of water applied.
7	20	Field irrigated. 1.2 in. of water applied.
8	1	Field irrigated. 1.2 in. of water applied.
8	11	Field irrigated. 1.2 in. of water applied.
8	21	Field irrigated. 1.2 in. of water applied.
9	17	Harvest

MONTH	DAY	OPERATION DESCRIPTION
3	18	Field chiseled.
3	23	Field disked.
4	4	Field disked.
4	10	Soybean planted. Potential heat units to reach maturity = 1385
4	10	Pesticide applied. Dual at the rate of 1.87lb ac ⁻¹ . Pest control factor 1.0
4	11	Field irrigated. 0.3 in. of water applied.
5	22	Field cultivated.
5	25	Field irrigated. 0.5 in. of water applied.
6	15	Field irrigated. 0.5 in. of water applied.
6	29	Field irrigated. 0.5 in. of water applied.
7	10	Field irrigated. 0.5 in. of water applied.
7	20	Field irrigated. 0.5 in. of water applied.
8	1	Field irrigated. 0.5 in. of water applied.
8	11	Field irrigated. 0.5 in. of water applied.
8	21	Field irrigated. 0.5 in. of water applied.
9	2	Harvested.

APPENDIX III.d

EPIC-View: Botchlet Farm Management Data - Corn Soybean Rotation under Minimum Tillage System

MONTH	DAY	OPERATION DESCRIPTION
3	26	Field disked
3	28	Pesticide applied. Dual applied at the rate of 1.95lb ac ⁻¹ . Pest control factor 1.0
3	30	Field fertilized. Anhydrous ammonia applied at the rate of 150 lb ac ⁻¹ depth of 6in.
4	8	Planted corn. Potential heat units for maturity =1466
4	8	Pesticide applied. Dual applied at the rate of 1.95lb ac ⁻¹ . Pest control factor 1.0
4	9	Field irrigated. 0.75 in. of water applied.
5	14	Field fertilized. Anhydrous ammonia applied at the rate of 150 lb ac ⁻¹ depth of 6in..
5	15	Field irrigated. 0.75 in. of water applied.
5	20	Field irrigated. 1.5 in. of water applied.
6	10	Field irrigated. 1.5 in. of water applied.
6	23	Field irrigated. 1.5 in. of water applied.
7	3	Field irrigated. 1.5 in. of water applied.
7	13	Field irrigated. 1.5 in. of water applied.
7	23	Field irrigated. 1.5 in. of water applied.
8	3	Field irrigated. 1.5 in. of water applied.
8	13	Field irrigated. 1.2 in. of water applied.
9	9	Harvested.

MONTH	DAY	OPERATION DESCRIPTION
3	23	Field disked.
3	28	Pesticide applied. Dual at the rate of 1.95 lb ac ⁻¹ . Pest control factor 1.0
4	6	Field fertilized. 50lb ac ⁻¹ of 18-46-0 at a depth of 4 in..
4	6	Soybean planted. Potential heat units to reach maturity = 1385
4	6	Pesticide applied. Dual at the rate of 1.95 lb ac ⁻¹ . Pest control factor 1.0
4	7	Field irrigated. 0.3 in. of water applied.
5	15	Field irrigated. 0.75 in. of water applied.
6	10	Field irrigated. 0.75 in. of water applied.
6	24	Field irrigated. 0.75 in. of water applied.
7	5	Field irrigated. 0.75 in. of water applied.
7	15	Field irrigated. 0.75 in. of water applied.
7	26	Field irrigated. 0.75 in. of water applied.
8	7	Field irrigated. 0.75 in. of water applied.
8	17	Field irrigated. 0.3 in. of water applied.
8	25	Harvested.

VITA ²

Mahesh N Rao

Candidate for the Degree of

Doctor of Philosophy

**Thesis: A GIS-BASED MODELING APPROACH FOR IMPLEMENTATION OF
SUSTAINABLE FARM MANAGEMENT PRACTICES: APPLICATION TO
FIELD LEVEL NITROGEN MODELING**

Major Field: Environmental Science

Biographical:

**Personal Data: Born in Madras, India, on October 24, 1965, the son of Sharmila N.
and M. Narayan. Rao**

**Education: Graduated from Little Flower Junior College, Hyderabad, India in
August 1982; received Bachelor of Science and Master of Science in
Agriculture from Andhra Pradesh Agricultural University, Hyderabad in August
1986 and December 1988. Completed the requirements for the Doctor of
Philosophy degree with a major in Environmental Science at Oklahoma State
University in December 1996.**

**Professional Membership: Soil and Water Conservation Society, Association of
American Geographers.**