

**DEVELOPMENT AND TESTING OF A DAILY
WATER-BALANCE MODEL FOR ESTIMATING
EVAPOTRANSPIRATION AND SOIL
WATER CONTENT**

by

ZAIGHAM AKHTAR KAZMI

Bachelor of Agricultural Engineering

Allahabad University

Allahabad, UP, India

1994

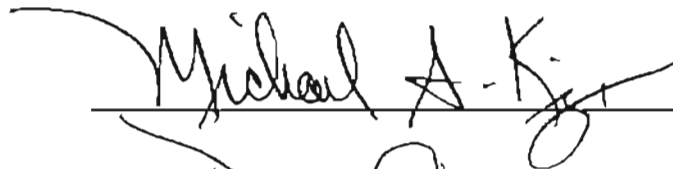
Submitted to the faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirement for
the Degree of
MASTER OF SCIENCE
May 1998

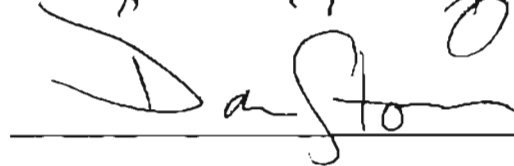
DEVELOPMENT AND TESTING OF A DAILY
WATER-BALANCE MODEL FOR ESTIMATING
EVAPOTRANSPIRATION AND SOIL
WATER CONTENT

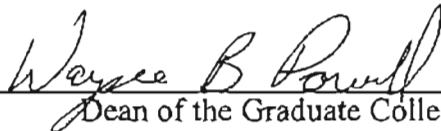
Thesis Approved:



Thesis Advisor







Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Ron Elliott for accepting me as his research assistant and acting as my mentor and friend. His profound knowledge of the issues involved, able guidance and critical analysis was the most important factor in the successful completion of this project. The advice and assistance offered by my other committee members, Drs. Daniel E. Storm and Michael A. Kizer, are also appreciated.

The financial support of the National Science Foundation through EPSCoR Project Number EP 59550478 is gratefully acknowledged.

I am grateful to Mark Shafer and Dr. Gabriel Senay who created the database of weather variables reported by the Mesonet stations. Their efforts made my work much easier. Appreciation is extended to Dr. Gabriel Senay and V. Sridhar for providing additional data and helpful insights. I am also grateful to William Marshall for providing me the lysimeter and vegetation data at the lysimeter sites: Apache, Goodwell and Marena.

And last, but not least, the help of Shellie Miller at critical moments is greatly appreciated.

TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION	1
Objective.....	5
Scope.....	6
2. LITERATURE REVIEW	7
Soil Water Balance Models	7
Evapotranspiration.....	10
Resistance Theory.....	11
Resistance terms.....	13
Scaling up to canopy.....	14
Soil Evaporation	16
Infiltration And Soil Water Movement.....	20
Infiltration	21
Soil Water Movement.....	23
Plant Growth.....	25
3. MODEL DESCRIPTION	28
Overview.....	28
Penman-Monteith Combination Equation.....	31
Vapor Pressure Calculations.....	31
Slope of Vapor Pressure Curve.....	32
Latent Heat of Vaporization.....	33
Psychrometric Constant	33
Air Density.....	33
Net Radiation	34
Soil Heat Flux	37
Aerodynamic Resistance.....	37
Reference Evapotranspiration.....	38

Plant Transpiration.....	39
Soil Evaporation	45
Soil Water Extraction	48
Infiltration.....	53
Drainage.....	55
Plant Growth.....	56

4. MODEL PARAMETERIZATION AND CALIBRATION 60

Model Parameterization.....	61
Weather Variables.....	61
Soil Parameters	63
Vegetation Parameters	66
Leaf area index.....	67
Canopy height.....	68
Critical dates of growth.....	69
Root depth.....	70
Stomatal resistance.....	71
ET Measurement.....	72
Soil Water Content Measurement.....	74
Model Calibration.....	76
Calibration Process	77
Calibration Results for the Development Sites.....	82
Apache	82
Goodwell.....	90
Marena	97
Stillwater	104
Summary of the Calibration.....	104

5. MODEL VALIDATION AND SENSITIVITY ANALYSIS 110

Parameterization of the Validation Sites	111
Vegetation Parameters	112
Leaf area index.....	112
Canopy height.....	113
Root depth.....	113
Critical dates of growth.....	114
Other parameters.....	115

Model Performance	116
Boise City.....	116
Broken Bow	121
Hollis.....	121
Miami.....	130
Summary.....	135
Sensitivity Analysis	135
Soil Parameters	139
Vegetation Parameters	144
SCS Curve Number.....	161
Summary.....	161
6. SUMMARY AND CONCLUSIONS	164
Recommendations.....	169
REFERENCES	171
APPENDICES	180
Appendix A: Input soil data files.....	181
Appendix B: Vegetation parameter input file.....	190
Appendix C: Calibration parameters for 229L sensors	192
Appendix D: Source code	196

LIST OF TABLES

Table	Page
4.1 Description of development sites	60
4.2 Mesonet weather variables used in the model	63
4.3 Texture based soil water parameter values, as used in the model	64
4.4 Hydrology soil groups	66
4.5 Initial estimates of vegetation parameters at the development sites	69
4.6 Depths (mm) and texture of soil layers at development sites	77
4.7 Calibration parameters	78
4.8 Site specific vegetation parameters and their values after calibration	81
4.9 Common parameters and their values after calibration	82
4.10 Comparison between the observed and the predicted ET at the lysimeter sites	85
5.1 Description of validation sites	110
5.2 Depths and texture of soil layers at validation sites	111
5.3 Vegetation parameters at the validation sites	115
5.4 Values of the soil water parameters for loam, used in the sensitivity analysis	137
5.5 Values of the soil water parameters for sandy clay loam, used in the sensitivity analysis	137
5.6 Values of input parameters used in the sensitivity analysis	138

LIST OF FIGURES

Figure	Page
3.1. Stomatal response to solar radiation ($r_{smin} = 120 \text{ s m}^{-1}$; $r_{smax} = 5000 \text{ s m}^{-1}$; $R_{pc} = 100 \text{ W m}^{-2}$)	41
3.2. Stomatal response to vapor pressure deficit ($c_d = 4 \text{ kPa}$)	42
3.3. Stomatal response to soil water content ($\theta_{wp} = 0.19$; $\theta_l = 0.265$)	43
3.4. Moisture availability function for soil layers ($\theta_{fc} = 0.25$)	50
3.5. Extraction function assuming uniform soil water content ($b = 0.0026$; $Z_{root} = 1200 \text{ mm}$)	52
3.6. Plant growth (dates and parameter values are not specific to any particular site)	57
4.1. ET for Jan. 1996 through Sep. 1997 at Apache	83
4.2. Comparison of predicted and observed ET for Jan. 1996 through Sep. 1997 at Apache	84
4.3. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Apache	86
4.4. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Apache	87
4.5. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Apache	88
4.6. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Apache	89
4.7. ET for Jan. 1996 through Sep. 1997 at Goodwell	91
4.8. Comparison of predicted and observed ET for Jan. 1996 through Sep. 1997 at Goodwell	92
4.9. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Goodwell	93
4.10. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Goodwell	94
4.11. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Goodwell	95
4.12. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Goodwell	96
4.13. ET for Jan. 1996 through Sep. 1997 at Marena	98
4.14. Comparison of predicted and observed ET for Jan. 1996 through Sep. 1997 at	

Marena	99
4.15. Soil water content at 5 cm for Jan. 1996 through Sep. 1997 at Marena	100
4.16. Soil water content at 25 cm for Jan. 1996 through Sep. 1997 at Marena	101
4.17. Soil water content at 60 cm for Jan. 1996 through Sep. 1997 at Marena	102
4.18. Soil water content at 75 cm for Jan. 1996 through Sep. 1997 at Marena	103
4.19. Soil water content at 5 cm for Jan. 1996 through Sep. 1997 at Stillwater	105
4.20. Soil water content at 25 cm for Jan. 1996 through Sep. 1997 at Stillwater	106
4.21. Soil water content at 60 cm for Jan. 1996 through Sep. 1997 at Stillwater	107
4.22. Soil water content at 75 cm for Jan. 1996 through Sep. 1997 at Stillwater	108
5.1. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Boise City	117
5.2. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Boise City	118
5.3. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Boise City	119
5.4. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Boise City	120
5.5. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Broken Bow	122
5.6. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Broken Bow	123
5.7. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Broken Bow	124
5.8. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Broken Bow	125
5.9. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Hollis	126
5.10. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Hollis	127
5.11. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Hollis	128
5.12. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Hollis	129
5.13. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Miami	131
5.14. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Miami	132
5.15. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Miami	133
5.16. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Miami	134
5.17. Sensitivity of soil water content to the field capacity (Base = 0.25, 0.23; Low = 0.20, 0.18; High = 0.30, 0.28 for sandy clay loam and loam, respectively)	140

5.18. Sensitivity of soil water content to the wilting point (Base = 0.13, 0.12; Low = 0.08, 0.07; High = 0.18, 0.17 for sandy clay loam and loam, respectively)	141
5.19. Sensitivity of ET to the field capacity (Base = 0.25, 0.23; Low = 0.20, 0.18; High = 0.30, 0.28 for sandy clay loam and loam, respectively)	142
5.20. Sensitivity of ET to the wilting point (Base = 0.13, 0.12; Low = 0.08, 0.07; High = 0.18, 0.17 for sandy clay loam and loam, respectively)	143
5.21. Sensitivity of soil water content of the first layer to the bubbling pressure (Base = 28.08 cm, 11.15 cm; Low = 5.57 cm, 1.63 cm; High = 141.5 cm, 76.40 cm for sandy clay loam and loam, respectively)	145
5.22. Sensitivity of soil evaporation to the bubbling pressure (Base = 28.08 cm, 11.15 cm; Low = 5.57 cm, 1.63 cm; High = 141.5 cm, 76.40 cm for sandy clay loam and loam, respectively)	146
5.23. Sensitivity of ET to the minimum stomatal resistance (Base = 166 s m ⁻¹ ; Low = 124.5 s m ⁻¹ ; High = 207.5 s m ⁻¹)	147
5.24. Sensitivity of soil water content to the minimum stomatal resistance (Base = 166 s m ⁻¹ ; Low = 124.5 s m ⁻¹ ; High = 207.5 s m ⁻¹)	148
5.25. Sensitivity of ET to the maximum LAI (Base = 5; Low = 3.75; High = 6.25)	150
5.26. Sensitivity of soil water content to the maximum LAI (Base = 5; Low = 3.75; High = 6.25)	151
5.27. Sensitivity of ET to the coefficient of the shelter factor (Base = 0.1; Low = 0.05; High = 0.2)	152
5.28. Sensitivity of soil water content to the coefficient of the shelter factor (Base = 0.1; Low = 0.05; High = 0.2)	153

5.29. Sensitivity of ET to the root depth (Base = 100 cm; Low = 75 cm; High = 125 cm)	154
5.29. Sensitivity of soil water content to the root depth (Base = 100 cm; Low = 75 cm; High = 125 cm)	155
5.30. Sensitivity of soil water content to the minimum fractional transpiration factor (Base = 0.5; Low = 0.4; High = 0.6)	157
5.31. Sensitivity of soil water content to the maximum fractional transpiration factor (Base = 0.5; Low = 0.4; High = 0.6)	158
5.32. Sensitivity of soil water content to the date of the start of growth (Base = March 26; Low = March 12; High = April 9)	159
5.33. Sensitivity of soil water content to the date of the start of senescence (Base = March 26; Low = March 12; High = April 9)	160
5.34. Sensitivity of soil water content to the fraction of available soil water content at no stress (Base = 0.5; Low = 0.4; High = 0.6)	162

NOMENCLATURE

α	Albedo
δ	Solar declination, radian
ε_a	Effective emissivity of the atmosphere
ε_{vs}	Effective emissivity of the vegetation and soil
ε'	Net emissivity of the surface
Φ	Latitude, radian
ϕ	Porosity
γ	Psychrometric constant, kPa C ⁻¹
λ	Latent heat of vaporization, MJ kg ⁻¹
Δ	Slope of vapor pressure curve
θ	Volumetric soil water content
θ_{FC}	Soil water content at the field capacity
θ_r	Residual soil water content
θ_s	Soil water content at saturation
θ_i	Threshold soil water content
θ_{WP}	Soil water content at the wilting point
ρ	Air density, kg m ⁻³
σ	Stefan - Boltzman constant (4.90 x 10 ⁻⁹) MJ m ⁻² K ⁻⁴ d ⁻¹
ω_s	Sunset hour angle, radian
Δt	Time interval, d
ΔT_{ref}	Reference sensor response, C
ΔT_{sensor}	Sensor response, C
c_p	Specific heat of moist air, kJ kg ⁻¹ C ⁻¹
d_i	Thickness of layer i
d_r	Relative distance of the earth from the sun
d_u	Zero plane displacement for the wind profile, m
$e^o(T)$	Saturation vapor pressure at temperature T, kPa
f	Cloudiness factor

k	von Karman constant equal to 0.41
k_A	Clearness index
h	Canopy height, m
h_{mn}	Minimum canopy height, m
h_{mx}	Maximum canopy height, m
m	Pore size distribution index
r	Normalized root length density
r_a	Aerodynamic resistance, $s\ m^{-1}$
r_c	Canopy resistance, $s\ m^{-1}$
r_{ms}	Mean leaf stomatal resistance, $s\ m^{-1}$
r_s	Surface resistance, $s\ m^{-1}$
$r_{S,i}$	Stomatal resistance of a layer, $s\ m^{-1}$
r_{smax}	Maximum stomatal resistance $s\ m^{-1}$
r_{smin}	Minimum stomatal resistance, $s\ m^{-1}$
r_{st}	Stomatal resistance, s^{-1}
t	Time, d
x	Measurement height for wind speed, m
x'	Measurement height for humidity and temperature, m
z_i	Depth down to the center of the layer i , mm
z_{om}	Roughness length for momentum transfer, m
z_{ov}	Roughness length for heat and vapor transfer, m
BP	Bubbling pressure, mm
C_r	Coefficient for the root extraction function
C_{sf}	Coefficient for shelter factor
CN	Curve number
D_{mx}	Date when LAI is at its peak
D_{sd}	Date at the start of the dormancy
D_{sg}	Date of the start of the growth

D_{ss}	Date at the start of the senescence
DV	Desorptivity volume, mm d ⁻¹
E_{AD}	Atmospheric demand, mm d ⁻¹
E_{bs}	Bare soil evaporation, mm d ⁻¹
E_s	Actual soil evaporation, mm d ⁻¹
$E_s(t)$	Soil evaporation in time t, mm
ET	Evapotranspiration, mm
ET_o	Grass reference evapotranspiration, mm d ⁻¹
F	Infiltration, mm
F_{ns}	Fraction of available soil water depletion at no stress
Fr	Fractional transpiration factor
Fr_{mn}	Minimum fractional transpiration factor
Fr_{mx}	Maximum fractional transpiration factor
G	Soil heat flux, MJ m ⁻² d ⁻¹
G_{sc}	Solar constant
HU	Heat unit
J	Day of the year (January 1 = 1)
I_{ra}	Initial rainfall abstraction, mm
$K(\theta)$	Hydraulic conductivity at θ , mm d ⁻¹
K_s	Saturated hydraulic conductivity, mm d ⁻¹
LAI	Leaf area index
LAI_{eff}	Effective LAI
LAI_{mx}	Maximum LAI
LAI_{mn}	Minimum LAI
LAI_{MFr}	LAI at maximum fraction cover
M_i	Moisture availability for layer i
P	Atmospheric pressure, kPa
P_c	Precipitation, mm

P_{int}	Interception, mm
P_s	Shelter factor
Q	Drainage volume, mm d ⁻¹
R_A	Daily extra terrestrial radiation, MJ m ⁻² d ⁻¹
R_{ld}	Incoming thermal radiation, MJ m ⁻² d ⁻¹
R_{lu}	Outgoing thermal radiation, MJ m ⁻² d ⁻¹
R_n	Net radiation, MJ m ⁻² d ⁻¹
R_{nl}	Net long wave radiation, MJ m ⁻² d ⁻¹
R_{nlo}	Net long wave radiation for a clear sky, MJ m ⁻² d ⁻¹
R_{ns}	Net short wave radiation, MJ m ⁻² d ⁻¹
R_p	Photosynthetically active radiation, MJ m ⁻² d ⁻¹
R_s	Solar radiation, MJ m ⁻² d ⁻¹
R_{so}	Short wave radiation for a clear sky, MJ m ⁻² d ⁻¹
RO	Runoff volume, mm
S	Potential maximum retention after rainfall begins, mm
S_E	Sorptivity, mm d ^{-1/2}
S_G	Slope of the LAI growth curve during early growth period
S_S	Slope of the LAI growth curve during senescence
T_v	Virtual temperature, K
T_a	Daily average air temperature, C
T_{md}	Daily mean dewpoint temperature
T_{max}	Daily maximum air temperature, C
T_{min}	Daily minimum air temperature, C
Tr	Plant transpiration, mm d ⁻¹
Tr_{fc}	Full cover transpiration, mm d ⁻¹
Tr_{root}	Transpiration taken from the root zone, mm
$TW_{i,j}$	Total water in layer i on day j, mm
VPD	Vapor pressure deficit, kPa

W_i Weighting factor for root extraction in layer i
 Z_{root} Root depth, mm

Chapter 1

INTRODUCTION

Evaporation of water from the earth's surface plays an important role in the energy and water balances at the earth-atmosphere interface. Water is evaporated from the earth's surface through evaporation from wet surfaces and through transpiration from plant tissues. These processes, evaporation and transpiration, are often combined and referred to as evapotranspiration (ET).

An important factor that directly affects evapotranspiration is soil water content in the root zone. Plants transpire well when the soil water content is near field capacity. In general up to 75% of the available soil water, the difference between field capacity and wilting point, can be easily extracted by plants (Punamia and Pande, 1990), but this percentage varies for different combinations of soils and plants. As soil water content decreases, roots are required to exert more force to extract water from the soil. This causes a reduction in the transpiration rate. As soil water content nears the wilting point, the transpiration rate becomes negligible.

The soil water content in the upper layer affects soil evaporation. When the soil is wet, evaporation is controlled by atmospheric factors. But once soil water in the upper layer falls below a certain limit, evaporation is primarily controlled by soil factors.

Knowledge of the soil water content is also important for several other reasons. In general, irrigation is recommended when soil water content nears or falls below the lower

limit of the readily available soil water. Monitoring the soil water in the root zone depth is very helpful in irrigation scheduling. Soil water content is also very important in partitioning the rainfall into infiltration and runoff. Soil with high water content will lead to reduced infiltration and higher runoff. It can increase the potential for erosion and flooding.

Soil moisture content is the most important criterion to evaluate the drought status of any region. The definition of drought may vary from place to place, depending upon the climate, but soil water content is always a key variable in deciding the drought status. Various humanitarian organizations and governments prefer to assess the severity of the drought before they initiate aid to the affected region.

Various instruments are available in the market to measure soil water content at discrete points. But installing such instruments at every desirable point is neither cost effective nor feasible. Intensive field measurement of soil water content, due to equipment and labor requirements, is mostly limited to research applications.

The need for estimates of soil water content has resulted in the development of estimation methods based on meteorological variables, soil hydraulic properties and land cover characteristics. Often soil water balance models are applied to estimate soil water content in the root zone. Generally they are continuous simulation models, which predict soil water content over an extended period of time. A wide variety of models have been developed for this purpose. These models differ in the way they estimate various components of the water balance, their time and space resolution, and the assumptions involved. Most of the models are tailored to the specific region they have been developed for, based on the data availability and climate.

Like any other balance, whether it is mass, volume or energy, water balance models relate inflow, outflow and change in storage:

$$\text{INFLOW} - \text{OUTFLOW} = \text{CHANGE IN STORAGE}$$

The temporal distribution of water in the soil profile results from a complex interaction of many variables related to current and past occurrence of weather, vegetation, soils and management. While soil-water and plant-growth principles have been studied for centuries, only in the last few decades have efforts been made to integrate the processes in the soil-plant-atmosphere system (Jong and Kabat, 1990).

Modern soil water balance models try to simulate the soil-plant-atmosphere system. This system consists of many other complex sub-systems, e.g. evapotranspiration, infiltration, etc. These components of water balance models can be measured, if instruments are available, or estimated using various techniques. Estimation techniques can be either empirical or mechanistic. Empirical models tend to contain simple relationships between one or more variables while the system itself is treated as a “black box”. On the other hand, mechanistic models attempt to simulate physical, chemical and physiological processes that take place within the system.

Empirical models often require less input data. These methods usually have to be calibrated for local conditions. The crop coefficient approach for crop ET and the curve number approach for runoff estimation are examples of empirical methods.

A mechanistic model generally provides a more theoretical description of the system. But these methods normally require numerous variables as inputs. Performance

of these models largely depends upon the accuracy of the set of variables and the associated model parameter estimates.

Different soil water balance models have been developed with a large variation in degree of complexity. Generally soil water balance models are neither purely empirical nor purely mechanistic. Various components of the model are simulated differently based on the data availability and time and other resource constraints. In practice, a process-based water balance model will make use of empirical functions when and where knowledge about a given process is lacking. Selection of the most appropriate method that balances the mechanistic-empirical character of the model is controlled by modeling objectives (Jong and Kabat, 1990), data availability, and other constraints.

Advances in electronic instrumentation, automated sensors and datalogging equipment have led to expanded deployment of automated weather stations. The availability of high quality weather data has allowed more sophisticated methods to be used to estimate some of the components of the water balance models, especially evapotranspiration. In Oklahoma, an extensive network of automated, remote weather stations has been installed which is known as the Mesonet. This consists of 114 weather stations located across the state, which report a variety of meteorological measurements.

The USA is one of those countries which have put forth considerable effort to develop databases of their natural resources. Databases of soil hydraulic properties and land use/land cover information are two examples. Advances in remote sensing have been crucial for the development of land use/land cover databases. This knowledge is very important in better simulation of the soil-plant system. These databases aid greatly in the application of water balance models over larger geographic areas.

The study of soil-plant-atmosphere relations also helps in understanding the atmospheric system and our climate. Many soil-vegetation-atmosphere models and schemes have been incorporated into meteorological models ranging from regional forecasts models to general circulation models (Capehart and Carlson, 1994). It is imperative that the soil-plant-atmosphere system, in general, and ET and soil water content, in particular, be studied thoroughly.

OBJECTIVE

The main objective of this research is to develop and test a point-based water balance model to predict, on a daily basis, soil water content in the root zone depth for non-irrigated grass in Oklahoma. Model results will be compared against field measurements of soil water and ET.

The goal is to develop a fairly generic model, which later can be used for different cover types and over larger geographic areas. Model components will be developed so as to take advantage of the weather, soil and vegetation databases available in Oklahoma.

SCOPE

This model will be tested at a number of Mesonet sites which exhibit diversity in climate, soils and vegetation. During the development process attempts will be made to avoid the use of parameters which are too specific to any particular land cover or region, in order to keep the model as generic as possible. It is anticipated that this model will be interfaced with a geographic information system (GIS) at a later date.

Model components will be physically based, but model practicality and data availability will be important considerations. Because of the natural variability in soil texture with depth, the model will accommodate multiple soil layers. Vegetation cover represented in the model may range from a bare soil to a full cover.

Chapter 2

LITERATURE REVIEW

SOIL-WATER BALANCE MODELS

Soil, vegetation and climate interact in a complex manner to determine the soil water and plant growth. Soil water is one of the most important elements of this system. A number of water balance studies have been conducted through the years for various parts of the world representing different climatic, vegetation and soil regimes (Scanlin, 1994; Yin and Brook, 1992; Motoyama et al., 1986; Clarke and Newson, 1978; Davis, 1971; etc.).

The soil-water balance is comprised of precipitation and irrigation inputs balanced by water outflows in the form of evapotranspiration, runoff, and deep drainage (Neilson, 1995). Many soil-water balance models consider infiltration as their only input (e.g., Capehart and Carlson, 1994), whereas some others also consider seepage and capillary rise (e.g., Wigmosta et al., 1994). Evapotranspiration is usually the dominant component of the outflow. It can be estimated by a number of techniques including the Penman-Monteith combination equation (Monteith, 1981), Shuttleworth-Wallace method (Shuttleworth and Wallace, 1985), Blaney-Criddle method (Blaney and Criddle, 1962),

and others. Other sources of outflow are usually minor in comparison to ET and sometimes ignored. Wigmosta et al. (1994) and Running and Coughlan (1988) do not estimate runoff. Running and Coughlan (1988) compute drainage only after the soil column has reached its water holding capacity. In layered soil-water balance models, movement of water from one layer to another adds one more component (Sharma et al., 1980).

Vegetation affects inflow as well as outflow. The SCS curve number approach, widely used to estimate runoff, depends upon the land cover (SCS, 1972). Among vegetation parameters, leaf area index, canopy height and root depth are the main variables which affect the temporal and spatial distribution of the soil water. Recent workshops on global ecological issues have identified leaf area index as the most important single variable measuring vegetation structure over large areas, and influencing energy and mass exchange (Wittwer, 1983; Botkin, 1986).

Hydrology and vegetation dynamics have, for the most part, been studied independently (Wigmosta et al., 1994). Different hydrology and crop-growth models simulate the soil-plant-atmosphere system in different ways depending upon the objective of the study and the data availability. Most of the hydrology models do not apply an elaborate approach to modeling the vegetation component of this system. They have mostly been concerned with estimating runoff; the representation of vegetation is often reduced to a specification of potential evapotranspiration and highly simplified soil water stress relationships (Wigmosta et al., 1994).

Many of the soil water balance models do not predict changes in vegetation (Wigmosta et al., 1994; Flerchinger et al., 1994; etc.). Recently attempts have been made

to give more attention to the vegetation component of the soil water balance models. Jong and Kabat (1990) synthesized the soil water balance model SWATRE with the crop production model CROPR (Feddes et al., 1978) to produce a new version called SWACROP. The biosphere-atmosphere-transfer-scheme (BATS) model (Dickinson et al., 1986) simulates changes in the fraction cover and LAI.

Those soil water balance models which simulate changes in vegetation do it in different ways. Many of them simulate vegetation growth as a function of temperature as in BATS (Dickinson et al., 1986). Capehart and Carlson (1994) used periodic functions based on the time of the year and type of the plant cover to calculate fraction cover. Feddes et al. (1978) gave a mathematical derivation of the growth rate of the crop as a function of the normalized water use, with the maximum growth rate as the upper limit and the efficiency of utilization of water as the initial slope. This approach was used by Jong and Kabat (1990).

Soil-water balance models differ in many additional ways. These differences range from their component methods to their temporal and spatial resolution. Some models are tailored to specific vegetation types such as FOREST-BGC (Running and Coughlan, 1988). Many others have been developed for specific sites or regions (Flerchinger et al., 1994). Site specific models often use methods based on the availability of the data for that site.

Layered models (Capehart and Carlson, 1994) use root extraction functions. They also apply algorithms to move soil water from one layer to another. Single layered models (Running and Coughlan, 1988) do not have a root extraction function and simply remove the transpiration amount from the root zone depth. Soil water movement is

mainly limited to drainage or deep percolation leaving the root zone. Many continental or regional scale models (Wigmosta et al., 1994; Neilson, 1995) divide the root zone depth into 2 layers. Soil evaporation and grass transpiration are taken from the upper layer whereas woody vegetation can extract water from both or either of the layers.

The FOREST-BGC model (Running and Coughlan, 1988) does not include a soil evaporation term. Wigmosta et al. (1994) and Neilson (1995) estimate soil evaporation, in addition to plant transpiration, and extract it only from the upper soil layer. The plant transpiration is taken from the root zone. Capehart and Carlson (1994) applied a more sophisticated approach. They combined soil evaporation and plant transpiration on the basis of fraction cover to produce a weighted average of the evapotranspiration.

EVAPOTRANSPIRATION

Many methods have been used to estimate evapotranspiration under various surface cover, water availability and meteorological conditions. Early methods consisted of a simple correlation between one or more weather parameters and plant growth. Over time as experience with ET estimation was gained and more accurate and reliable meteorological data became available, existing methods were refined and improved methods were developed.

The physics of evaporation was first explained, to some extent, by Howard Penman (1948). He combined the aerodynamic and thermodynamic aspects of

evaporation to make theoretical estimates of evaporation rates from standard meteorological data, estimates that can be retrospective. He later developed an equation for single leaves that included leaf resistance (Penman, 1953). Later, Monteith (1981) introduced terms to account for the resistances offered to the transfer of sensible heat (r_H) and water vapor (r_v). The resulting equation is known as the Penman-Monteith equation.

Resistance Theory

When water evaporates at the interface between a wet surface and the atmosphere, the transfer of latent and sensible heat takes place. In his analytical work, Monteith (1981) related the transfer of sensible heat to a resistance r_H . Similarly he related the transfer of latent heat to a resistance r_v . If relative humidity at the surface is equal to one, e.g. soil/foilage thoroughly wetted by rain, $r_v \approx r_H$ (Robinson, 1966; Monteith, 1981). When sensible and latent heat exchange occur at a complex surface, e.g. transpiring stomata, it is possible for r_v to exceed r_H (Monteith, 1981).

Individual pores behave as if they were wired in parallel with each other and with the cuticle which they perforate. And this compound physiological resistance of the leaf surface, r_s , can be treated as if it were placed in series with the resistance of the boundary layer, next to the leaf surface, r_a (Cowan, 1972). Ignoring small differences between the boundary layer resistances for heat and water vapor, it follows that with $r_H \approx r_a$, $r_v \approx r_a + r_s$ (Monteith, 1981). Thus, the final equation has two resistance parameters, one to account for resistance offered by atmospheric constraints called aerodynamic resistance (r_a) and another to account for the resistance offered by the surface itself called canopy resistance or surface resistance (r_s).

Shuttleworth and Wallace (1985) developed a method to estimate evapotranspiration from a sparsely vegetated surface using energy balance and mass and heat transfer equations in a manner similar to the development of the Penman-Monteith (PM) equation. They calculated evaporation from soil and transpiration from vegetation separately and then combined them to obtain a weighted evapotranspiration. They introduced some additional resistance terms to account for the aerodynamic mixing in sparse crops and soil evaporation from between and beneath the vegetation. They also considered the amount of sensible heat leaving soil in the case of a sparse canopy. This term causes a reduction in ET. If this sensible heat flux is assumed to be zero, the canopy ET given by the Shuttleworth-Wallace (SW) model is the same as given by the PM method (Shuttleworth and Wallace, 1985).

A comparison study showed the PM model to be less accurate than the SW model for two reasons (Stannard, 1993). First, its big leaf assumption does not hold during dry sunny periods, when a large fraction of the sensible heat comes from the ground. Secondly, immediately after rainfall it can not simulate the large values of bare soil evaporation. This result implies that the performance of the PM model can be improved if it is used to estimate only canopy evapotranspiration and soil evaporation is estimated separately. Canopy transpiration and soil evaporation can then be blended together to get total evapotranspiration. Capehart and Carlson (1994) combined them based on the fractional vegetation cover.

Resistance Terms

Jarvis (1976) suggests that stomatal resistance depends upon specific humidity difference, photosynthetically active radiation (PAR), leaf temperature, leaf water potential and CO₂ concentration. CO₂ has an overall influence on stomatal resistance (Morrison and Gifford, 1983), yet it is excluded from most of the resistance models because of the absence of field instrumentation and data (Wright et al., 1995). Also changes in carbon dioxide concentration are considered to be small (Stannard, 1993). Stewart (1988) simplified the parameterization by substituting solar radiation for PAR and soil water for leaf water potential. Kaufmann (1982), while modeling stomatal resistance, considered leaf temperature and water stress to be secondary factors as their effects occur only when temperature or water stress is extreme. Wright et al. (1995) obtained better results after excluding the temperature term. He was of the opinion that its influence is partly reflected in the humidity difference. Also leaf temperature is a difficult term to measure.

The model proposed by Jarvis (1976) has been used by many researchers to estimate stomatal resistance, the inverse of conductance, with different parameterization. Deardoff (1978) estimated stomatal resistance by correcting species dependent minimum stomatal resistance for the effects of solar radiation and soil water. His model is of the form:

$$r_{st} = r_{smin} [f(R_s) + f(\theta)] \quad (2.1)$$

This model was used by Capehart and Carlson (1994). Dickinson et al. (1986) estimated stomatal resistance for his model, BATS, as the product of a species dependent minimum stomatal resistance and functions of solar radiation, seasonal temperature and

soil water. Noilhan and Planton (1989) used the same model with the addition of a vapor pressure deficit function. Dickinson et al. (1991) proposed a modified model to estimate stomatal resistance as the product of a species dependent minimum resistance and functions of PAR, leaf temperature and humidity difference. Wigmosta et al. (1994) used the same relationship but he also incorporated one more function to account for soil water based on Feddes et al. (1978). The reciprocals of these functions range from 0 to 1. Thus, these variables actually reduce the species dependent maximum conductance depending upon environmental constraints. The species dependent maximum leaf conductance is often defined as the largest value of conductance observed in fully developed, but not senescent, leaves of well-watered plants under optimal climatic conditions, natural outdoor carbon-dioxide concentration and sufficient nutrient supply (Korner et al., 1979). Korner et al. (1979) has reported the maximum leaf conductance values for the diffusion of water vapor in 246 plant species and cultivars belonging to 13 morphologically and/or ecologically comparable plant groups. He has duly adjusted these values to present the values based on total leaf surface area, wherever it was possible.

Scaling up to Canopy

Much has been said about the Penman-Monteith combination equation and its applicability to a real canopy from a theoretical standpoint (Shuttleworth, 1976; Finnigan and Raupach, 1987) and also from an experimental point of view (Lindroth and Halldin, 1986). However it is certain that “the physical meaning of the canopy resistance is not easy to comprehend” (Brutsaert, 1982) and that the classical Penman-Monteith equation

is not sufficient to correctly describe the complexity of the vegetation-atmosphere interaction (Lhomme, 1991).

Lhomme (1991) presents a historical survey of canopy resistance. Different approaches, which are used to scale up stomatal resistance to represent canopy resistance, include a single layer approach based on the 'big leaf assumption' (Monteith, 1973) and a multi-layer approach.

Monteith's single-layer approach is perhaps the simplest of all and is still used widely with one or more modifications. This approach assumes all the leaves as resistors acting in parallel. Thus, canopy resistance can be obtained by dividing the leaf stomatal resistance by leaf area index, i.e. $r_s = r_{ms}/2L$ (Monteith, 1973), where r_{ms} is the mean leaf stomatal resistance for an amphistomatal canopy and L is the leaf area index. Leaf stomatal resistance would be half of the mean stomatal resistance in the case of an amphistomatal canopy (leaves with stomata on both the surfaces). A better procedure might be to divide the canopy into several parallel layers, to calculate for each layer a stomatal resistance $r_{s,i}$ as $r_{ms,i}/2dL_i$, where $r_{ms,i}$ is the mean leaf resistance of a layer i with a partial leaf area index of dL_i , and to interpret the stomatal resistance of the canopy as the effective resistance of the $r_{s,i}$ acting in parallel (Shuttleworth, 1976; Monteith, 1985; Lindroth and Halldin, 1986).

In the multi-layer approach, the stand is treated as a continuous or discrete set of horizontal planes, each one absorbing net radiation and transferring sensible and latent heat (Lhomme, 1991). The discrete or stratified approach conceives of the canopy as being divided into a finite number of layers, each one with a given thickness, and yields linear equations, which are solved by means of matrix methods (Waggoner and

Reifsynder, 1968; Waggoner et al., 1969; Furnival et al., 1975). In the continuous approach (Phillip, 1964; Cowan, 1968; Goudriaan and Waggoner, 1972; Furnival et al., 1975; Perrier, 1976), the canopy is considered to be composed of an infinite number of strata.

Rochette et al. (1991) used different scaling techniques to obtain the canopy resistance of maize. He recommends use of a shelter factor to account for the leaves which are not directly in the sunlight. Mascart et al. (1991) proposes a function based on leaf area index to estimate the shelter factor. He reported that neglecting the shelter factor underestimates the canopy resistance by a factor ranging from 2.3 to 2.4 when the plant density is high ($3 < L < 6$) and by a factor ranging from 0.5 to 2.3 when plant density is low ($1 < L < 3$).

SOIL EVAPORATION

Evaporation from bare soil is a time dependent phenomenon due to progressive drying. Many of the factors influencing soil evaporation are functions of time. When bare soil is thoroughly wetted, the soil surface behaves like water in so far as the relative humidity of air in contact with the surface is 100%. The rate of evaporation can be calculated from the Penman-Monteith equation (Monteith, 1981) and as a matter of observation, is usually very close to the rate for adjacent short vegetation, despite differences in radiative and aerodynamic properties (Monteith, 1981). This process can

not continue indefinitely because the conductivity of soil for water decreases very rapidly as it dries and it is usually only a few days before the rate of evaporation becomes limited by the upward diffusion of liquid water towards the surface (Monteith, 1981).

In his analytical approach to estimate soil evaporation, Monteith (1981) assumed that the evaporation of water takes place from wet soil below a dry soil layer of increasing thickness, treated as isothermal. Since surface resistance comes from this dry layer, it becomes a function of time. Or we can say that surface resistance is proportional to the amount of water evaporated, since depth of the dry layer depends on the amount evaporated. The model, presented by him, estimates cumulative soil evaporation as a function of the square root of time. In his model at time $t = 0$, soil evaporation is a constant determined by the state of the atmosphere (Monteith, 1981).

Evaporation from bare soil is often characterized as occurring in two distinct stages (Ritchie, 1974; Kanemasu et al., 1976; Hanks and Hill, 1980; Ritchie and Johnson, 1990). The first stage, stage 1, is termed as the “energy limited” stage. In this stage there is enough water available in the upper profile of soil to fulfill the requirement of atmospheric demand. During this stage, water is transported to the soil surface at a rate sufficient to supply the potential rate of evaporation, which is, in turn, governed by energy availability at the soil surface. The second stage, stage 2, is termed as the “falling” stage or “soil limited” stage, where hydraulic transport of subsurface water to the soil surface is unable to supply the water at the potential evaporation rate.

As mentioned above, the first-stage soil evaporation is limited by energy availability. Most of the commonly used approaches attempt to estimate it as a function of potential evaporation or potential evapotranspiration. Jensen et al. (1990) recommend

that it be taken as 90 percent of the potential evapotranspiration. Allen et al. (1996a) recommend calculating first stage soil evaporation as 1.15 times the grass reference evapotranspiration.

In theoretical treatments (Rose, 1968), it is usually assumed that h (relative humidity) drops instantaneously from its initial value of unity during the first stage of drying to a fixed value of h_0 at the onset of the second stage where h_0 is determined by atmospheric factors. The theory predicts that the subsequent rate of evaporation is inversely proportional to the square root of elapsed time. Thus during the second stage the cumulative soil evaporation tends to increase with the square root of time for a given soil and evaporation potential (Jensen et al., 1990). The most commonly used approach is to use a constant of proportionality to calculate cumulative evaporation as a function of square root of time. The constant of proportionality depends on the soil characteristics and the soil water content (Black et al., 1969) and time taken is the time after the onset of second stage soil evaporation.

The combination equation given by Monteith, also known as the Penman-Monteith equation, can be used to estimate evaporation from bare soil (Wigmosta et al., 1994). In the case of soil, the specified surface resistance is that of the soil surface. This resistance comes from the progressively drying soil layer (Monteith, 1981). Capehart and Carlson (1994) used the Penman-Monteith equation to estimate bare soil evaporation with canopy resistance replaced by an effective ground resistance.

As water evaporates, the amount of evaporable water in the soil layer decreases. This causes a decrease in conductivity for moisture or, alternatively, an increase in resistance. Allen et al. (1998) presented a mathematical relationship to estimate the

average rate of soil water evaporation occurring during any period based on the assumption that the evaporation rate in the second stage is linearly proportional to the depth of evaporable water remaining.

Most of the models using a two-stage soil evaporation approach move from first stage to second stage once they have evaporated a certain amount of water in the first stage (Jensen et al., 1990; Allen et al., 1998). This amount also depends upon the depth of the soil layer that is contributing towards evaporation. Evaporation will enter into the first stage again if there is sufficient rainfall or irrigation application to bring the amount of water above the threshold for the first-stage soil evaporation.

Another way of characterizing the two stages of soil evaporation is as atmospheric demand and soil desorptivity, respectively. In this approach, the first stage evaporation is estimated as a function of potential evaporation demand. In the second stage, the evaporation is estimated as a nonlinear function of soil water content (Wigmosta et al., 1994). Under this approach the soil evaporation, at any time, is calculated as the minimum of atmospheric demand and desorptivity volume (Wigmosta et al., 1994). Thus soil evaporation moves from one stage to the other depending upon which one would minimize the evaporation.

The thickness of the soil layer contributing to soil evaporation is very important for modeling purposes. Eagleson (1978a) defines this thickness, the penetration depth, as the depth at which surface-induced capillary forces become negligible. It is a function of soil hydraulic and thermal properties (Allen et al., 1996b). For modeling purposes it is usually kept within 20 cm. Allen et al. (1998) recommend taking a value between 10 and 20 cm. They used a depth of 15 cm, for all soil types, to estimate the parameters of their

soil evaporation model (Allen et al., 1998). Dickinson et al. (1986) used a depth of 10 cm in their BATS model. Capehart and Carlson (1994) extracted water for soil evaporation from a depth of 10 cm. Wigmosta et al. (1994) adjusted the depth of the upper soil layer to 20 cm during the calibration of their model.

INFILTRATION AND SOIL-WATER MOVEMENT

A porous medium, such as soil, is an interconnected structure of tiny conduits of various shapes and sizes. Soil moisture movement is determined, to a great extent, by the soil properties. Soil strata with different physical properties may overlay each other, forming horizons. Also, soils exhibit great spatial variability even within a small area. Some of the soil hydraulic parameters, e.g. hydraulic conductivity, vary with soil water content thus introducing time variability.

Attempts to model flow through porous media date back to the 19th century. In 1856 Darcy developed his law of porous medium flow that still forms the basis of many of the soil moisture movement studies. Darcy's law describes a steady uniform flow of constant velocity through a porous medium. Richards (1931) derived his equation based on Darcy's law for change in soil water content through time. This is the governing equation for unsteady unsaturated flow in a porous medium.

Infiltration

Both physically based and empirical infiltration models have been proposed and used in hydrological modeling. Most of the physically based models are based on Richards' (1931) unsaturated flow equation, or can be derived from it. One of the earliest infiltration equations was developed by Horton (1933, 1939), who observed that the infiltration begins at some rate and exponentially decreases until it reaches a constant rate. Eagleson (1970) and Raudkivi (1979) have shown that Horton's equation can be derived from Richards' equation by assuming that the hydraulic conductivity and the soil water diffusivity are constants independent of the moisture content of the soil (Chow et al., 1988). Philip (1957, 1969) solved the Richards' equation under less restrictive conditions by assuming that the hydraulic conductivity and the soil water diffusivity can vary with the moisture content.

Models based on the Richards' equation are more desirable because they use measured, physical parameters and, in general, they are more accurate. But they are difficult to use (James and Larson, 1976). Since the Richards' equation does not have a general analytic solution, it must be solved numerically. Such solutions are complex, use a lot of computer time and require more detailed input data.

Green and Ampt (1911) proposed a simplified picture of infiltration so that an exact analytical solution could be found. They assumed that the wetting front is a sharp boundary dividing soil with a moisture content of θ_i below, from saturated soil with moisture content ϕ (porosity) above. Under a ponded condition, soil water moves vertically downward as piston flow. Thus the wetting front moves down, increasing the

saturation depth. This model estimates potential infiltration rate, i.e., the rate under a ponded condition.

During a rainfall, ponding will take place only if the rainfall intensity is greater than the infiltration capacity of the soil. Prior to ponding, infiltration takes place at a rate equal to the rainfall rate (or application rate). If rainfall continues the potential infiltration rate will decrease due to increasing soil water content, and soon it will be equal to the rainfall rate. This time is termed as the ponding time (Mein and Larson, 1973; Chow et al., 1988). Afterwards infiltration would take place at the potential infiltration rate. Mein and Larson (1973) modified the Green-Ampt equation to estimate infiltration at any time after ponding by offsetting the starting time by an amount equal to the ponding time. Chu (1978) extended the same concept to model infiltration during unsteady rainfall.

The Mein-Larson equation, the modified Green-Ampt equation, does not have a direct solution, as in this equation cumulative infiltration is an implicit function of time. Many attempts have been made to solve this equation. Chu (1978) presented a graphical solution. Chaubey et al. (1994) used an iterative solution in which iteration continued until the difference between successive estimates of cumulative infiltration was less than the stopping criterion. Srivastava et al. (1996) presented a direct approximate solution to the Mein-Larson equation.

Another approach to estimate infiltration is to model runoff and then estimate infiltration based on the knowledge of runoff. One method to estimate runoff is by subtracting the abstraction from the rainfall. Abstraction includes interception of precipitation on vegetation above the ground, depression storage on the ground surface as

water accumulates in hollows over the surface, and infiltration of water into the soil (Chow et al., 1988).

The Soil Conservation Service (1972) of the U.S. Department of Agriculture (USDA) has developed a method for computing abstraction from storm rainfall. The amount of abstraction is calculated as a function of an empirical constant called curve number (CN). Curve numbers depend on soil type and vegetation cover and are listed in many SCS publications and hydrology textbooks. In this approach precipitation must exceed a certain amount before any runoff is generated.

Soil Water Movement

Downward movement of the water within the soil does not cease immediately at the end of the infiltration and may in fact persist for a long time as soil water redistributes within the profile. During redistribution the wetting front continues to move downward and the transmission zone water content decreases (James and Larson, 1976). Downward movement of water, percolation, is often modeled using Darcy's law (examples are Sharma et al. (1980) and Wigmosta et al. (1994)). Sharma et al. (1980) estimated soil water content and hydraulic conductivity for all layers, except the first layer, at their upper and lower boundaries. From these values the hydraulic gradient and average hydraulic conductivity were estimated for all layers. Then the averages of hydraulic conductivity and hydraulic gradient for two layers were used to estimate flow of moisture from one layer to another. Eagleson (1978a) formulated the apparent percolation velocity as a steady gravitational seepage which is simply the value of hydraulic conductivity at the given soil water content.

The same Darcy's law can be applied for horizontal unsaturated flow as well as upward vertical flow (Iwata et al., 1995). According to this law, soil water moves from a higher head to a lower head, where head is the energy of water per unit weight. Darcy's law was extended to unsaturated flow by studying unsaturated hydraulic conductivity. Wyckoff and Botset (1936) showed a relation between hydraulic conductivity and water saturation in their experiments with mixtures of gas and liquid. Brooks and Corey (1964) presented an equation to estimate unsaturated hydraulic conductivity of soil.

The soil water regime may be affected by seepage from adjoining areas, subsurface runoff and capillary rise from a water table, besides other factors. Subsurface runoff and seepage tend to negate the effect of each other. Capehart and Carlson (1994) consider subsurface runoff in their model but acknowledge the fact that, in principle, some water should flow in from adjacent locations.

During inter-storm periods, the soil water regime is mainly affected by unsaturated soil water movement and root extraction for the purpose of transpiration. Since the unsaturated hydraulic conductivity generally decreases radically with decreased soil water content, the root extraction becomes predominant with decreased soil water content (Miyazaki, 1993).

PLANT GROWTH

Vegetative growth is a complex process which depends on many factors including but not limited to climate, soil, water availability, etc. After the germination of seed and emergence of a shoot, the growth depends by and large on the rate of photosynthesis. This growth is often restricted by climatological factors and water and nutrient availability.

Photosynthetically active radiation plays a very important role in the growth of a crop. Monteith (1977) estimated daily potential increase in the biomass of crops based on the interception of PAR (photosynthetically active radiation). Pearson and Ison (1987) described a linear relationship between the growth rate of grassland and the photosynthetically active radiation, assuming that the herbage is intercepting all the radiation and other factors, e.g. temperature, are optimal.

Interception of photosynthetically active radiation depends upon the leaf area index and extinction coefficient. During the early stage of growth, each new leaf that is formed contributes to more light being intercepted so that growth increases even more. Later on leaves will gradually start overshadowing each other, and above a LAI of 3, new leaf area hardly results in any increase in light being intercepted (Goudriaan and van Laar, 1994).

Both the light and temperature environments experienced by the leaf during growth and development affect its size (Charles-Edwards et al., 1986). In studying the relative leaf growth rates of crops of broad bean (*Vicia faba*), Dennett et al. (1978)

reported that the growth rate increased with mean leaf temperature but was not directly related to solar radiation. The rate of expansion of individual leaves is highest 2-5 days after emergence and declines thereafter (Pearson and Ison, 1987). During the period of fastest expansion, rates are highly sensitive to temperature (Thomas and Stoddart, 1984).

Rooting depth normally increases rapidly from the seeding depth to a crop-specific maximum. In many crops the maximum is usually attained well before physiological maturity (Borg and Grimes, 1986). The extent and shape of the root system vary according to soil characteristics such as texture, bulk density (and void ratio), aeration, soil water potential and fertility (Pearson and Ison, 1987).

Often plant growth is simulated based on daily heat unit accumulation or cumulative degree days. It is computed as:

$$HU_k = \left[\frac{T_{mx,k} - T_{mn,k}}{2} \right] - T_{b,j} \quad (2.2)$$

where HU_k is the number of heat units for day k , T_{mx} and T_{mn} are maximum and minimum temperature (C) on day k and T_b is the crop specific base temperature (C) of crop j . No growth occurs at or below T_b .

During early vegetative growth, leaf tip appearance and blade expansion are linear functions of heat unit accumulation (Tollenaar et al., 1979; Watts, 1972). There is an exponential increase in LAI during the early growth stage. In many crops, LAI decreases after reaching a maximum and approaches zero at physiological maturity. In addition, leaf expansion, final LAI, and leaf duration are reduced by stresses (Acevedo et al., 1971; Elk and Harway 1965).

EPIC (Sharpley and Williams, 1990) uses a single heat unit based model to grow all the crops considered. In this model, potential increase in biomass for a day is estimated based on the interception of solar radiation. LAI is simulated as a function of heat units, crop stress, and crop development stages. Crop height is estimated only as a function of heat unit accumulation.

Chapter 3

MODEL DESCRIPTION

OVERVIEW

I am presenting a process based, daily simulation model. This model divides the soil-plant-atmosphere system into components and processes that can be treated individually. The model is modular so that any component can be changed without affecting the other components.

Only rainfall is assumed to be contributing as inflow (irrigation could be treated as “pseudo rainfall”). Rainfall water, after subtracting runoff and canopy interception, is infiltrated into the soil. Runoff is estimated using the curve number approach. After the soil column, down to the root zone depth, has reached saturation, excess water is simply drained out. If any layer is at a moisture content above field capacity, it is drained to the field capacity and the drained water is entered into the layer immediately below the draining layer. Drainage volume is calculated via Darcy's Law assuming a unit hydraulic gradient (Wigmosta et al., 1994). This module runs 24 times on an hourly basis to get daily drainage. This approach is being used to account for the fact that the soil water content varies with the drainage and thus varies the hydraulic conductivity.

In addition to deep drainage, other sources of outflow are transpiration from the plant tissues and evaporation from the upper layer of the soil. These two are estimated separately and then mixed based on the fractional transpiration factor to produce a

weighted average of the evapotranspiration (Capehart and Carlson, 1994). It is assumed that the fraction of the soil exposed to sunlight would contribute towards soil evaporation whereas the fraction covered by the plant canopy would contribute towards transpiration. The maximum fractional transpiration factor is limited to a value less than 1.0 to account for the fact that soil evaporation is never suppressed to zero. This value will differ with vegetation type. Soil evaporation is taken only from the uppermost layer which is 15 cm deep. Water for the transpiration is extracted from the root zone using an exponential weighting function, which varies from a maximum at the surface to a minimum at the bottom of the root zone. The extraction coefficient for each layer also depends on the water content of that layer, as it affects the contribution of individual layers towards total transpiration. A layer at a moisture content near wilting point will release less water and thus more water will be taken from the other layers.

Transpiration is estimated using the Penman-Monteith combination equation. This equation accounts for the energy required to sustain evapotranspiration and a mechanism required to remove the vapor. Daily net radiation and vapor pressure deficit are estimated as recommended by Allen et al. (1994b). Canopy resistance is obtained by scaling up the stomatal resistance.

Soil evaporation is estimated as the minimum of atmospheric demand and the desorption volume. Desorption volume is here defined as the amount of water which the soil would release to a non-restricting atmosphere over a period of time. This approach is similar to the one used by Wigmosta et al. (1994), though the specific methods used to estimate atmospheric demand and desorptivity are different. Atmospheric demand is estimated as 1.15 times grass reference evapotranspiration, as recommended by Allen et

al. (1996a) for estimating first stage soil evaporation. The desorption model follows that presented by Gardner (1959) and used by Eagleson (1978a), Milly (1986) and others. Randall (1989) discusses it in detail.

Land cover characteristics affect the evapotranspiration and the vertical distribution of the soil water. To keep track of vegetation parameters a simple plant growth model is used. Outputs from this model are canopy height, leaf area index, fractional transpiration factor and root depth. The fractional transpiration factor is correlated with the LAI and can be visualized as the fraction of the ground covered by the canopy. For grass, it is assumed that roots do not change through time once they have been established. But canopy height and leaf area index do change through time.

A linear model is used to increase the leaf area index from the start of the growing period to the time it reaches a maximum. The leaf area index is assumed to remain at this maximum until the start of the senescence period. Then it is linearly decreased from the start of the decline to the start of the dormant period. Leaf area index is assumed to remain constant, at its minimum value, throughout the dormant period. Canopy height and fractional transpiration factor are assumed to change in the same fashion as LAI. But fractional transpiration factor often reaches its maximum before LAI. Fractional transpiration factor is assumed to reach its maximum value at a LAI of 3.0, as above this LAI overshadowing of leaves occurs (Mascart et al., 1991; Goudriaan and van Laar, 1994). In the case of grazing and/or mowing, a cut-off LAI is assumed. It is the maximum LAI which would be present throughout the growing period.

PENMAN-MONTEITH COMBINATION EQUATION

The Penman-Monteith combination equation (Monteith, 1981), to estimate evapotranspiration from any surface with a specified surface resistance, is:

$$ET = \frac{\Delta(R_n - G) + \rho c_p (VPD)K / r_a}{\lambda[\Delta + \gamma(1 + r_s / r_a)]} \quad (3.1)$$

where ET is evapotranspiration (mm d⁻¹), R_n is net radiation (MJ m⁻² d⁻¹), G is soil heat flux density (MJ m⁻² d⁻¹), ρ is air density (kg m⁻³), c_p is specific heat of moist air (kJ kg⁻¹ C⁻¹), VPD is vapor pressure deficit (kPa), r_a is aerodynamic resistance (s m⁻¹), λ is latent heat of vaporization (MJ kg⁻¹), Δ is slope of saturation vapor pressure versus temperature curve (kPa C⁻¹), γ is psychrometric constant (kPa C⁻¹), r_s is surface resistance (s m⁻¹) and K is a scaling factor equal to 86400 s d⁻¹.

Vapor Pressure Calculations

The saturation vapor pressure is a measure of the water vapor content of saturated air at a given temperature. Saturation vapor pressure, e^o, at any temperature, T, can be estimated using an equation developed by Tetens (1930) (Allen et al., 1994b):

$$e^o(T) = 0.611 \exp\left(\frac{17.27T}{T + 237.3}\right) \quad (3.2)$$

where T is temperature (C) and e^o(T) is saturation vapor pressure (kPa).

Thus saturation vapor pressure at maximum, minimum, average and mean dewpoint temperature would be calculated as

$$e^{\circ}(T_{\max}) = 0.611 \exp\left(\frac{17.27T_{\max}}{T_{\max} + 237.3}\right) \quad (3.3)$$

$$e^{\circ}(T_{\min}) = 0.611 \exp\left(\frac{17.27T_{\min}}{T_{\min} + 237.3}\right) \quad (3.4)$$

$$e^{\circ}(T_a) = 0.611 \exp\left(\frac{17.27T_a}{T_a + 237.3}\right) \quad (3.5)$$

$$e^{\circ}(T_{\text{md}}) = 0.611 \exp\left(\frac{17.27T_{\text{md}}}{T_{\text{md}} + 237.3}\right) \quad (3.6)$$

where subscripts represent maximum (max), minimum (min), average (a) and mean dew point (md).

Vapor pressure deficit is the difference between the saturation vapor pressure at the given temperature and the actual vapor pressure. For daily calculations, Allen et al. (1994b) recommend the following equation to estimate vapor pressure deficit:

$$\text{VPD} = \frac{e^{\circ}(T_{\max}) + e^{\circ}(T_{\min})}{2} - e^{\circ}(T_{\text{md}}) \quad (3.7)$$

Slope of Vapor Pressure Curve

The slope of the saturation vapor pressure versus temperature relationship can be estimated by differentiating eq 3.2 (Jensen et al., 1990), yielding:

$$\Delta = \frac{4099e^{\circ}(T_a)}{(T_a + 237.3)^2} \quad (3.8)$$

Latent Heat of Vaporization

Latent heat of vaporization is the amount of energy required to change water at a given temperature from the liquid to the vapor phase. It can be estimated as a function of temperature using an equation developed by Harrison (Jensen et al., 1990):

$$\lambda = 2.501 - (0.002361)T_a \quad (3.9)$$

where λ is in MJ kg⁻¹.

Psychrometric Constant

The psychrometric constant (γ) relates the sensible heat gained from moving air to the sensible heat converted into latent heat (Jensen et al., 1990) and can be estimated as:

$$\gamma = \frac{c_p P}{0.622\lambda} \quad (3.10)$$

where γ is in kPa C⁻¹, c_p is specific heat of moist air equal to (0.001013 MJ kg⁻¹ C⁻¹) and P is atmospheric pressure (kPa). Thus

$$\gamma = 0.00163 \frac{P}{\lambda} \quad (3.11)$$

Air Density

Air density can be calculated based on the atmospheric pressure and the virtual temperature as (Jensen et al., 1990):

$$\rho = 3.486 \frac{P}{T_v} \quad (3.12)$$

where ρ is density of moist air (kg m^{-3}), P is atmospheric pressure (kPa), and T_v is virtual temperature (K). The virtual temperature, which accounts for the effects of moisture on buoyancy (Rosenberg et al., 1983), can be calculated as:

$$T_v = \frac{T_a + 273.16}{1 - \frac{0.378e^0(T_{md})}{P}} \quad (3.13)$$

Net Radiation

Net radiation is a measure of the amount of energy available for use at the earth's surface, and provides energy for plant growth, heating of the soil and atmosphere, and evaporation of water. Net radiation is composed of short wave and long wave radiation, and is dependent on reflective properties of the surface, the temperature of the surface and the water vapor content of the atmosphere (Jensen et al., 1990).

Net radiation (R_n) is estimated as the sum of net short wave radiation (R_{ns}) and net long wave radiation (R_{nl}) (Allen et al., 1994b):

$$R_n = R_{ns} + R_{nl} \quad (3.14)$$

where R_n is net radiation ($\text{MJ m}^{-2} \text{d}^{-1}$), R_{ns} is net short wave radiation ($\text{MJ m}^{-2} \text{d}^{-1}$) (positive downward) and R_{nl} is net long wave radiation ($\text{MJ m}^{-2} \text{d}^{-1}$) (positive downward).

Net short wave radiation is estimated as:

$$R_{ns} = (1 - \alpha) R_s \quad (3.15)$$

where R_s is solar radiation ($\text{MJ m}^{-2} \text{d}^{-1}$) and α is surface albedo.

Net long wave radiation is estimated as (Allen et al., 1994b):

$$R_{nl} = R_{ld} - R_{lu} \approx f(\varepsilon_a - \varepsilon_{vs})\sigma \frac{T_{kx}^4 + T_{kn}^4}{2} \quad (3.16)$$

where R_{ld} is incoming thermal radiation emitted by the atmosphere and cloud cover to the earth's surface (downward flux) ($\text{MJ m}^{-2} \text{d}^{-1}$), R_{lu} is outgoing thermal radiation emitted by the vegetation and soil into the atmosphere (upward flux) ($\text{MJ m}^{-2} \text{d}^{-1}$), f is an adjustment for cloud cover or a cloudiness factor, ε_a is effective emissivity of the atmosphere, ε_{vs} is emissivity by vegetation (0.99 – 0.94) and soil (0.98 – 0.80) and approximately equal to 0.98, σ is the Stefan-Boltzmann constant ($4.90 \times 10^{-9} \text{ MJ m}^{-2} \text{K}^{-4} \text{d}^{-1}$), T_{kx} is maximum daily temperature (K) and T_{kn} is minimum daily temperature (K).

Cloudiness factor, f , can be estimated as:

$$f = \frac{R_{nl}}{R_{nlo}} \approx a_c \frac{R_s}{R_{so}} + b_c \quad (3.17)$$

where R_{nlo} is net long wave radiation for a clear sky day ($\text{MJ m}^{-2} \text{d}^{-1}$), R_s is short wave solar radiation ($\text{MJ m}^{-2} \text{d}^{-1}$), R_{so} is short wave solar radiation for a clear sky day ($\text{MJ m}^{-2} \text{d}^{-1}$), and a_c and b_c are calibration parameters. a_c and b_c were set equal to 1.35 and -0.35 , respectively, as recommended by Allen et al. (1994b).

Net emissivity of the surface, ε' , can be estimated as (Allen et al., 1994b):

$$\varepsilon' = (\varepsilon_{vs} - \varepsilon_a) = (a_1 + b_1 \sqrt{e(T_{md})}) \quad (3.18)$$

where $e(T_{md})$ is vapor pressure at mean dewpoint temperature (kPa), and a_1 and b_1 are empirical coefficients. $e(T_{md})$ is estimated using eq 3.6. a_1 and b_1 are taken as 0.34 and -0.14 respectively.

Thus, net long wave radiation is (Allen et al., 1994b):

$$R_{nl} = \left[\left(1.35 \frac{R_s}{R_{so}} - 0.35 \right) \left(0.34 - 0.14 \sqrt{e^{\circ}(T_{md})} \right) \sigma \frac{(T_{sk}^4 + T_{skn}^4)}{2} \right] \quad (3.19)$$

Clear sky short wave radiation can be estimated as (Allen et al., 1996b):

$$R_{so} = k_A R_A \quad (3.20)$$

where k_A is a clearness index equal to $0.75 + 0.00002E$, where E is elevation (m). R_A is daily extraterrestrial radiation ($MJ m^{-2} d^{-1}$).

Daily extraterrestrial radiation, R_A , can be computed for a location as a function of latitude and day of the year using the following equations from Duffie and Beckman (1991) (Allen et al., 1994b):

$$R_A = \frac{(24)(60)}{\pi} G_{sc} d_r [\omega_s \sin(\Phi) \sin(\delta) + \cos(\Phi) \cos(\delta) \sin(\omega_s)] \quad (3.21)$$

where G_{sc} is the solar constant equal to $0.0820 MJ m^{-2} min^{-1}$, d_r is relative distance of the earth from the sun, Φ is latitude of the station in radians (positive for northern hemisphere), δ is declination in radians and ω_s is sunset hour in radians. These parameters are estimated as:

$$d_r = 1 + 0.33 \cos\left(\frac{2\pi}{365} J\right) \quad (3.22)$$

$$\delta = 0.4093 \sin\left(2\pi \frac{284 + J}{365}\right) \quad (3.23)$$

$$\omega_s = \arccos[-\tan(\Phi)\tan(\delta)] \quad (3.24)$$

where J is the day of the year (January 1st=1).

Net radiation, R_n , can now be estimated as:

$$R_n = (1 - \alpha)R_s - \left[\left(1.35 \frac{R_s}{k_A R_A} - 0.35 \right) \left(0.34 - 0.14 \sqrt{e^o(T_{md})} \right) \sigma \frac{(T_{kx}^4 + T_{kn}^4)}{2} \right] \quad (3.25)$$

where R_n is net radiation ($\text{MJ m}^{-2} \text{ d}^{-1}$), R_s is solar radiation ($\text{MJ m}^{-2} \text{ d}^{-1}$), R_A is extra terrestrial radiation ($\text{MJ m}^{-2} \text{ d}^{-1}$), σ is Stefan-Boltzmann constant equal to $4.90 \times 10^{-9} \text{ MJ m}^{-2} \text{ K}^{-4} \text{ d}^{-1}$, $e^o(T_{md})$ is vapor pressure at mean dewpoint temperature (kPa), T_{kx} is maximum daily temperature (K) and T_{kn} is minimum daily temperature (K).

Soil Heat Flux

On a daily basis and assuming an effective soil depth of 0.18 m, soil heat flux (G) can be estimated as (Allen et al., 1994b):

$$G = 0.38(T_{a,i} - T_{a,i-1}) \quad (3.26)$$

where T_a is the mean daily temperature. Subscripts i and i-1 represent the current and the previous day respectively.

Aerodynamic Resistance

Aerodynamic resistance, r_a , is estimated assuming a logarithmic wind profile (Allen et al., 1994b):

$$r_a = \frac{\ln[(x - d_u)/z_{om}]}{u(x)k^2} \ln[(x' - d_u)/z_{ov}] \quad (3.27)$$

where z_{om} is the roughness length for momentum transfer (m), z_{ov} is the roughness length for heat and vapor transfer (m), d_u is the zero plane displacement for the wind profile (m), x is the measurement height for wind speed (m), x' is the measurement height for

humidity and temperature (m), k is the von Karman constant equal to 0.41 and $u(x)$ is the wind speed at height x (m s^{-1}).

The roughness lengths and the displacement height are estimated as:

$$z_{om} = 0.123h \quad (3.28)$$

$$z_{ov} = 0.1z_{om} \quad (3.29)$$

$$d_u = (2/3)h \quad (3.30)$$

where h is canopy height in m.

Surface resistance (r_s) will vary depending upon the vegetation cover but is fixed for the reference ET calculation.

REFERENCE EVAPOTRANSPIRATION

Allen et al. (1994a) defined grass reference ET as the rate of evapotranspiration from a hypothetical reference crop with an assumed crop height of 0.12 m, a fixed surface resistance of 70 s m^{-1} and an albedo of 0.23, closely resembling the evapotranspiration from an extensive surface of green grass of uniform height, actively growing, completely shading the ground and with adequate water.

Thus equation 3.1 can be used to calculate grass reference ET, ET_o , with the following specifications. Canopy height, h , is equal to 0.12 m, surface albedo, α , is equal to 0.23 and surface resistance, r_s , is equal to 70 s m^{-1} .

PLANT TRANSPIRATION

Full cover plant transpiration, Tr_{fc} , is estimated using the Penman-Monteith combination equation, eq 3.1, with cover specific values for canopy height, surface resistance and surface albedo.

The surface albedo is a measure of the reflectivity of the surface to solar radiation. It is dependent on surface vegetative and moisture conditions, and varies for different types of vegetation and for different heights and growth stages of the vegetation (Oke, 1987). For grass and other short vegetation, it varies from 0.15 to 0.25 (Allen et al., 1996b). In this model a constant value of 0.18 is being used for grass, as used by some other modelers (Wigmosta et al. 1994).

Roughness lengths for momentum and vapor, and zero plane displacement height, are estimated using equations 3.28 – 3.30. And then based on these values, aerodynamic resistance is estimated using equation 3.27.

In case of a plant canopy, the surface resistance, r_s in eq 3.1, is substituted by a canopy resistance, r_c . Canopy resistance is computed by scaling the stomatal resistance of individual leaves (Shuttleworth and Wallace, 1985; Lhomme, 1991; Fisher, 1995):

$$r_c = \frac{r_{st}}{2LAI_{eff}} \quad (3.31)$$

where r_c is canopy resistance ($s\ m^{-1}$), r_{st} is stomatal resistance ($s\ m^{-1}$) and LAI_{eff} is the effective leaf area index of the canopy equal to (LAI/P_s) , where P_s is a shelter factor, suggested by Mascart et al. (1991) and computed as:

$$P_s = C_{sf}LAI + 1.2 \quad (3.32)$$

where C_{sf} is a coefficient for the shelter factor.

Stomatal resistance is modeled by multiplying a species dependent minimum stomatal resistance by functions representing environmental and water stresses, which influence stomatal behavior (Stewart, 1988; Dickinson et al., 1991). Only three factors are being considered here; they are solar radiation, vapor pressure deficit and soil water. Thus the stomatal resistance model takes the following form:

$$r_{st} = r_{smin} f(PAR)f(VPD)f(\theta) \quad (3.33)$$

where r_{st} is stomatal resistance ($s\ m^{-1}$), r_{smin} is crop specific minimum stomatal resistance ($s\ m^{-1}$), $f(PAR)$ is a function for stomatal response to solar radiation, $f(VPD)$ is a function for stomatal response to vapor pressure deficit and $f(\theta)$ is a function for stomatal response to soil water stress. The general shape of these functions, as outlined in the equations to follow, is shown in figures 3.1 – 3.3.

These functions are estimated as recommended by Wigmosta et al. (1994). Relations for the first two of these, $f(PAR)$ and $f(VPD)$, are taken from Dickinson et al. (1991) and the third, $f(\theta)$, follows Feddes et al. (1978). The minimum value of these functions is 1. They are computed as the inverse of the original conductance functions as follows:

$$f(PAR)^{-1} = \frac{r_{smin} / r_{smax} + R_p / R_{pc}}{1 + R_p / R_{pc}} \quad (3.34)$$

$$f(VPD)^{-1} = 1 - (VPD/c_d) \quad (VPD \leq c_d) \quad (3.35)$$

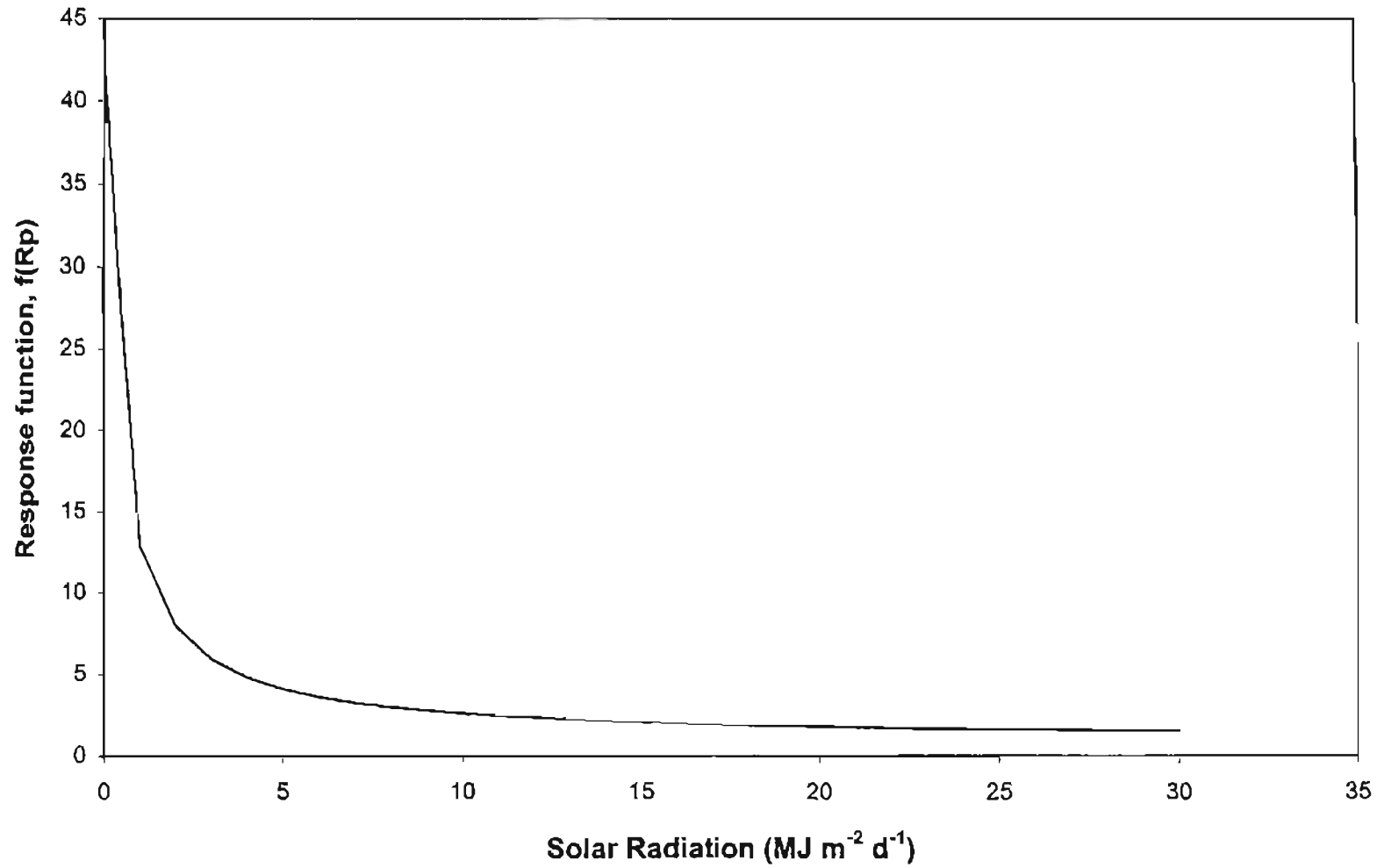


Figure 3.1. Stomatal response to solar radiation ($r_{s\min} = 120 \text{ s m}^{-1}$; $r_{s\max} = 5000 \text{ s m}^{-1}$; $R_{pc} = 100 \text{ W m}^{-2}$)

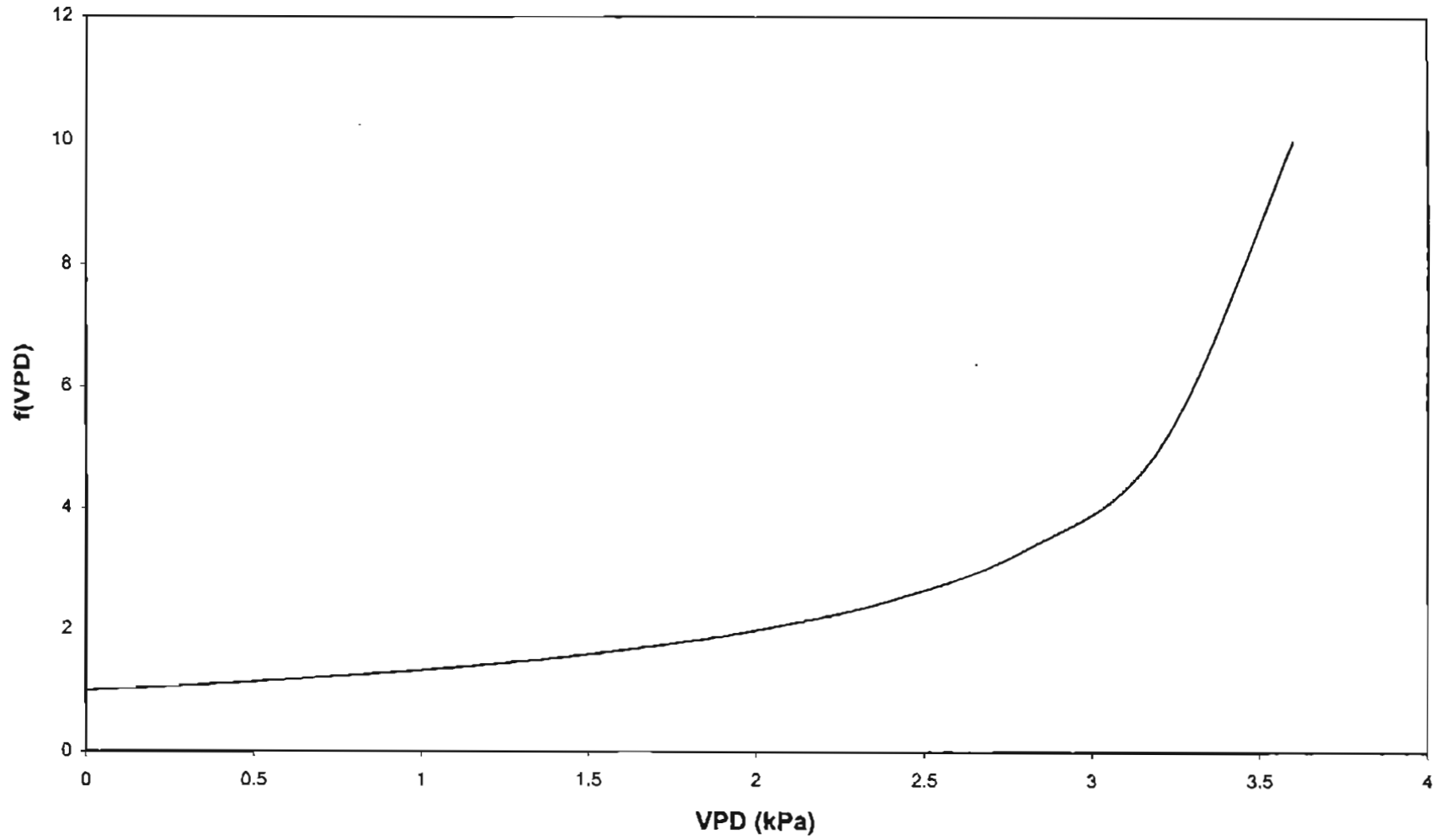


Figure 3.2. Stomatal response to vapor pressure deficit ($c_d = 4$ kPa)

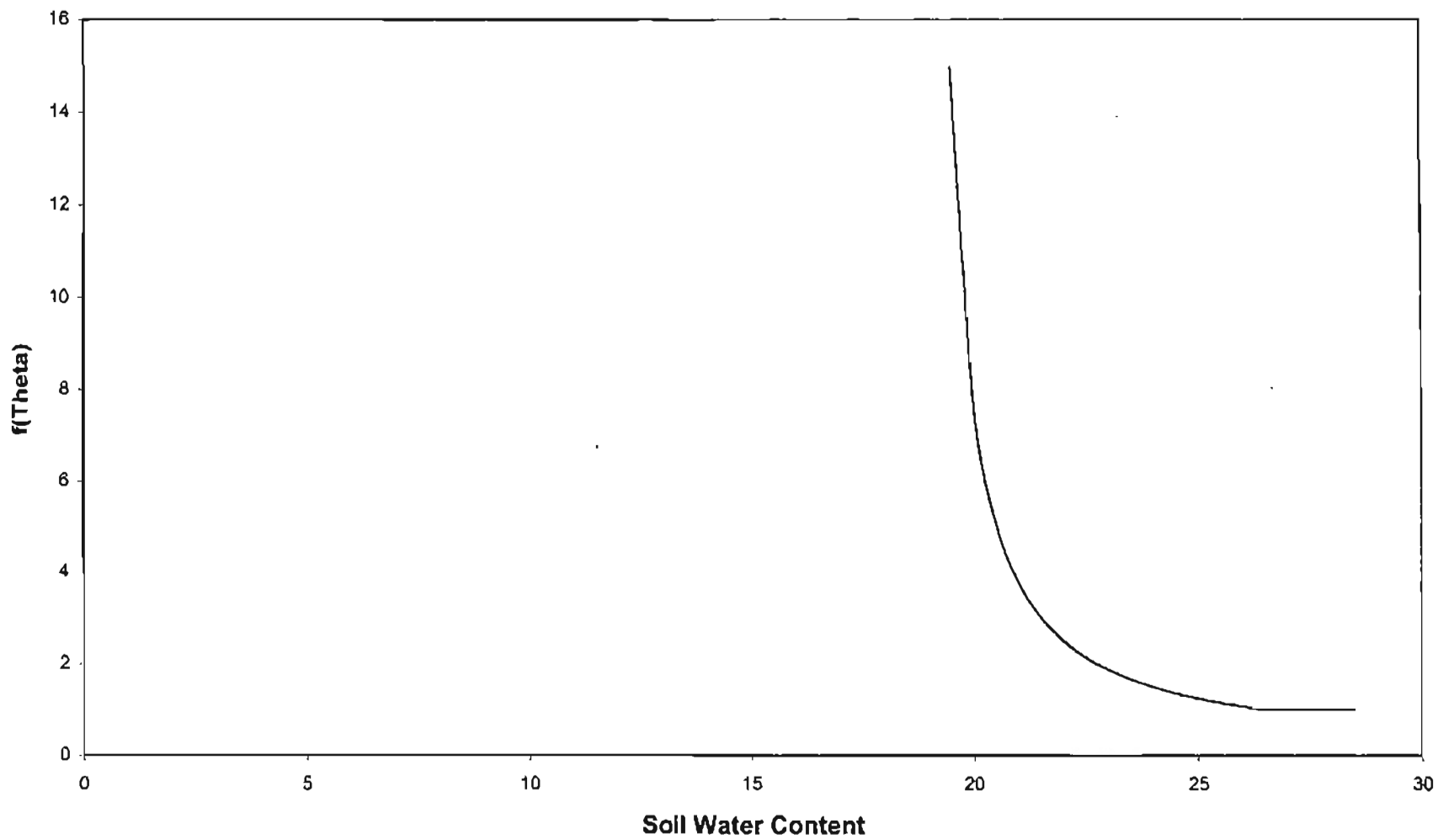


Figure 3.3. Stomatal response to soil water content ($\theta_{WP} = 0.19$; $\theta_t = 0.265$)

$$f(\theta)^{-1} = \frac{\theta - \theta_{wp}}{\theta_t - \theta_{wp}} \quad (\theta_t > \theta > \theta_{wp}) \quad (3.36)$$

where R_p is photosynthetically active radiation ($\text{MJ m}^{-2} \text{ d}^{-1}$), R_{pc} is the light level at which $r_s = 2r_{smin}$ ($\text{MJ m}^{-2} \text{ d}^{-1}$), r_{smin} is species dependent minimum stomatal resistance (s m^{-1}), r_{smax} is the maximum (cuticular) resistance (s m^{-1}), VPD is the vapor pressure deficit (kPa), c_d is the vapor pressure deficit causing stomatal closure ≈ 4 kPa (Wigmosta et al., 1994), θ is the average soil water content in the root zone depth, θ_t is the threshold soil water content above which soil water conditions do not restrict transpiration and θ_{wp} is the plant wilting point. For values of θ greater than θ_t , $f(\theta)^{-1}$ equals 1, whereas for θ less than θ_{wp} , r_{st} is set equal to r_{smax} .

R_p can be approximated as 46% of the incident solar radiation (Skartveit and Olseth, 1994). R_{pc} can be taken as 100 W m^{-2} for grasslands and crops and 30 W m^{-2} for trees (Dickinson et al., 1986).

Threshold soil water content is calculated as:

$$\theta_t = (1 - F_{ns})(\theta_{FC} - \theta_{wp}) + \theta_{wp} \quad (3.37)$$

where F_{ns} is the fraction of the available soil water which can be extracted with negligible stress, θ_{wp} is the plant wilting point and θ_{FC} is field capacity. This equation is often used in crop coefficient based models of ET to account for soil water stress (Allen et al., 1996b). θ , θ_{FC} and θ_{wp} are calculated as average soil water content over the entire root zone depth.

$$\theta = \frac{\sum_{i=1}^{N_{\text{root}}} (\theta_i d_i)}{\sum_{i=1}^{N_{\text{root}}} d_i} \quad (3.38)$$

where N_{root} is the number of layers in the root zone and d_i is the thickness of layer i (mm).

It is assumed that a maximum water stress is reached when the soil water content reaches the wilting point. Since a soil water content below the wilting point is effectively not going to stress the ET process any more, layers at a water content below the wilting point are considered to be at the wilting point, for this calculation.

SOIL EVAPORATION

Soil evaporation is estimated as occurring in two stages. The first stage is termed as climate-controlled and soil evaporation in this stage equals atmospheric demand. The second stage is termed as soil-controlled and soil evaporation in this stage equals desorptivity. So soil evaporation is estimated as the minimum of atmospheric demand and desorptivity volume, following Eagleson (1978a) and Wigmosta et al. (1994). Allen et al. (1996a) recommends the first stage soil evaporation be estimated as:

$$E_{\text{AD}} = 1.15ET_0 \quad (3.39)$$

where E_{AD} is the atmospheric demand for soil evaporation (mm) and ET_0 is grass reference ET (mm).

The second stage soil evaporation is estimated following the approach proposed by Randall (1989):

$$E_s(t + \Delta t) = E_s(t) \sqrt{1 + \Delta t \left(\frac{S_E}{E_s(t)} \right)^2} \quad (3.40)$$

where t is time since the beginning of soil evaporation (d), Δt is the time interval (d), $E_s(t + \Delta t)$ is total soil evaporation in time $t + \Delta t$ (mm), $E_s(t)$ is total soil evaporation in time t (mm) and S_E is sorptivity ($\text{mm d}^{-1/2}$).

Thus desorptivity volume for time interval Δt can be estimated as:

$$DV = E_s(t) \sqrt{1 + \Delta t \left(\frac{S_E}{E_s(t)} \right)^2} - E_s(t) \quad (3.41)$$

where DV is desorptivity volume for time interval Δt (mm).

The sorptivity can be calculated as (Randall, 1989; Wigmosta et al., 1994):

$$S_E = \left[\frac{8\phi K_s BP}{3(1+3m)(1+4m)} \right]^{1/2} \left[\frac{\theta}{\phi} \right]^{(1/2m)+2} \quad (3.42)$$

where S_E is the sorptivity of the soil layer ($\text{mm d}^{-1/2}$), K_s is the saturated hydraulic conductivity of the soil layer (mm d^{-1}), BP is the bubbling pressure (approximately the minimum capillary pressure on the drainage cycle at which a continuous non-wetting phase exists) (mm), m is the pore size distribution index for the soil texture (a characteristic constant being small for media having a wide range of pore sizes and large for media with a relatively uniform pore size), θ is the soil water content of the soil layer and ϕ is the soil porosity.

Sorptivity is calculated based on an average soil water content to account for the fact that the soil water content is not constant throughout the day and sorptivity can change dramatically over the course of a 24-hour period. First atmospheric demand for that day, eq 3.39, and desorptivity volume, based on initial soil water content, are estimated (eq 3.42 and eq 3.41). Then water equal to soil evaporation (eq 3.44) is subtracted from the amount of water in the first layer and the new soil water content is calculated. The average of these two soil water content values is used to calculate the final sorptivity using eq 3.42. Then the final desorptivity volume is estimated based on this new value of sorptivity. Desorptivity volume and atmospheric demand are compared to estimate soil evaporation using equation 3.44.

Time t is reset after each rainfall event and incremented daily. For the first day ($t=1$), the desorptivity volume can be set equal to sorptivity as in general (Randall, 1989):

$$DV = S_E(t)^{1/2} \quad (3.43)$$

Bare soil evaporation is determined as (Wigmosta et al., 1994):

$$E_{bs} = \min (E_{AD}, DV) \quad (3.44)$$

Cumulative soil evaporation is reset after each rainfall event and is incremented daily as:

$$E_s(t) = E_s(t) + E_s \quad (3.45)$$

where E_s is actual soil evaporation (mm) as discussed in the following section..

SOIL WATER EXTRACTION

Once soil evaporation and plant transpiration have been estimated, they are extracted from the root zone. As soil evaporation is estimated assuming bare soil and plant transpiration is estimated assuming full cover, the first step is to mix them based on the fractional transpiration factor.

Thus soil evaporation is calculated as:

$$E_s = E_{bs}(1 - Fr) \quad (3.46)$$

and plant transpiration as:

$$Tr = Tr_{fc}(Fr) \quad (3.47)$$

where Fr is the fractional transpiration factor, E_{bs} is bare soil evaporation (mm), E_s is actual soil evaporation (mm), Tr_{fc} is the full cover transpiration (mm) and Tr is actual transpiration (mm). Now evapotranspiration can be calculated as:

$$ET = E_s + Tr \quad (3.48)$$

where ET is evapotranspiration (mm).

Soil evaporation is taken only from the first layer, which is 150 mm deep. Water for plant transpiration is extracted from the entire root zone using a function based on the root distribution. The extraction of water from individual layers is affected by the water availability in that layer. The extraction function used in this model is:

$$W_i = \frac{(M_i)^n d_i r_i}{\sum_{i=1}^{N_{root}} [(M_i)^n d_i r_i]} \quad (3.49)$$

where W_i is the weighting factor for the i^{th} layer, M_i is the moisture availability function, d_i is the depth of the layer i , r_i is the normalized root length density for the i^{th} layer, N_{root} is the number of soil layers containing roots and n is a stress parameter. n is equal to 2 if soil water content is less than wilting point, otherwise it is equal to 1.

The parameter n is used to modify the moisture availability function when the soil water content falls below the wilting point. This allows the extraction of water from an individual layer to be greatly reduced when the water content falls below the wilting point.

Moisture availability for soil layers containing roots is estimated as (Capehart and Carlson, 1994):

$$M_i = \left[\frac{1}{4} \left\{ 1 - \cos \left(\frac{\theta_i}{\theta_{\text{FC}_i}} \pi \right) \right\}^2 \right] \quad \text{for } \theta_i < \text{FC} \quad (3.50)$$

$$M_i = 1 \quad \text{for } \theta_i \geq \text{FC} \quad (3.51)$$

where M_i is the moisture availability function for layer i and θ_{FC_i} is the field capacity for layer i . Figure 3.4 is a plot of this function for $\theta_{\text{FC}_i} = 0.25$.

The normalized root length density, r , at any point z is estimated as proposed by Zhang et al. (1993):

$$r = a \exp(-bz) \quad (3.52)$$

where b is an empirical constant and a is computed as:

$$a = \frac{b}{1 - \exp(-bZ_{\text{root}})} \quad (3.53)$$

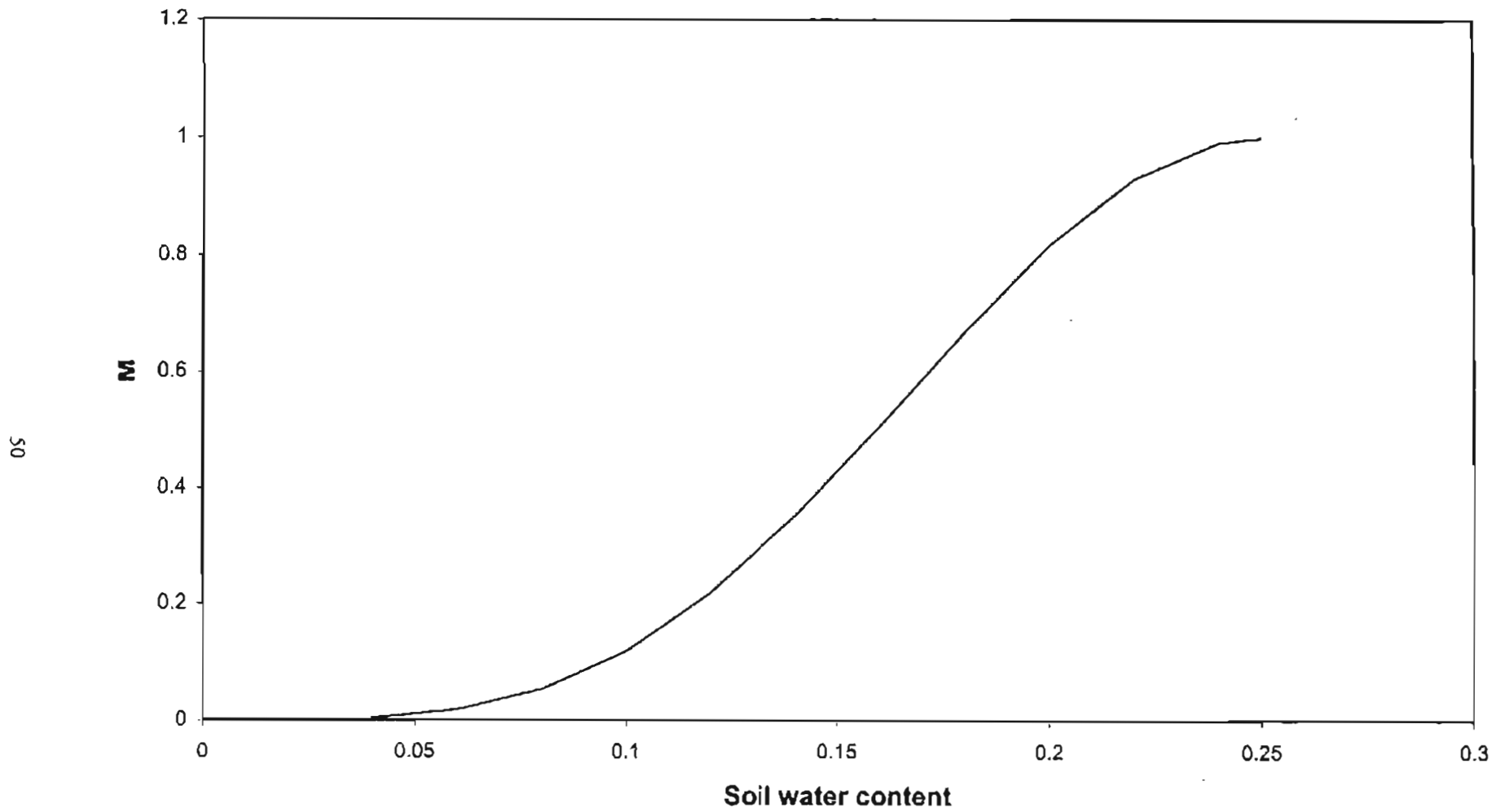


Figure 3.4. Moisture availability function for soil layers ($\theta_{FC} = 0.25$)

where Z_{root} is the root depth. For any layer i , z is the depth down to the center of that layer (if roots extend throughout the layer) or the depth down to the center of the portion of the layer containing roots. Figure 3.5 shows the weighting factor assuming a uniform water content throughout the root zone and a root depth of 1200 mm.

If there is any canopy interception, it should be subtracted from the transpiration before the extraction function is applied:

$$Tr_{\text{root}} = Tr - P_{\text{int}} \quad (3.54)$$

where Tr_{root} is the amount of water to be extracted from the root zone as transpiration (mm) and P_{int} is the canopy interception (mm).

For the first layer

$$TW_{1,j} = TW_{1,j-1} - E_s - Tr_{\text{root}} W_1 \quad (3.55)$$

and for other layers

$$TW_{i,j} = TW_{i,j-1} - Tr_{\text{root}} W_i \quad (3.56)$$

where TW_i is the depth of water in layer i (mm) and subscripts $j-1$ and j represent the previous day and the current day, respectively.

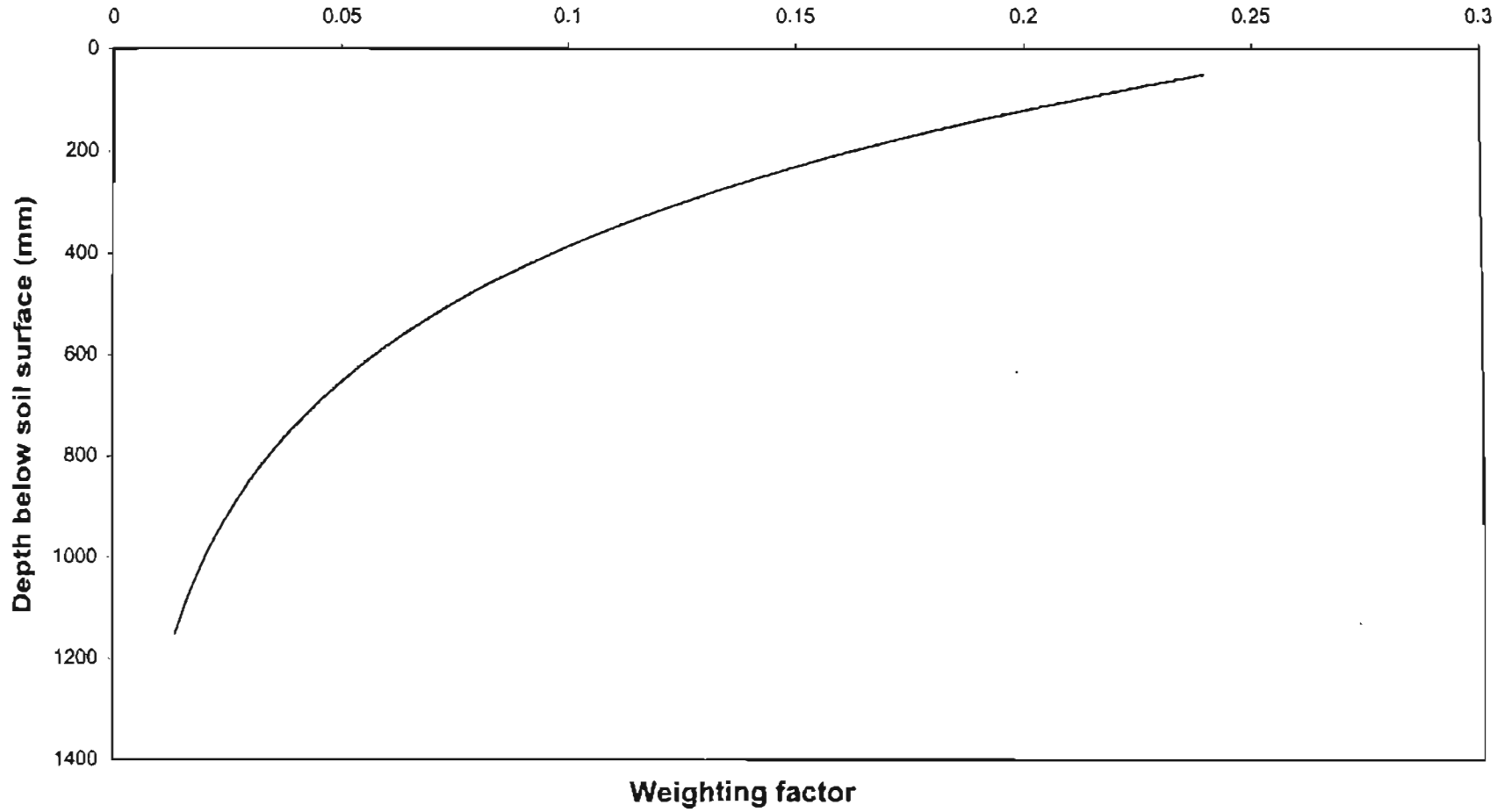


Figure 3.5. Extraction function assuming uniform soil water content ($b = 0.0026$; $Z_{\text{root}} = 1200$ mm)

INFILTRATION

Runoff is estimated using the SCS curve number approach. It is assumed that all the rainfall that is not runoff and that is not intercepted by the canopy, infiltrates into the soil.

$$F = P_c - (RO + P_{int}) \quad (3.57)$$

where F is the depth of the infiltrated water (mm), P_c is the precipitation depth (mm), RO is runoff (mm) and P_{int} is the depth of water intercepted by the canopy (mm).

First the maximum amount which can be intercepted by the canopy (Wigmosta et al., 1994) is calculated as:

$$P_{int} = Fr(0.1 LAI) \quad (3.58)$$

If rainfall is less than this amount, no infiltration takes place. If rainfall exceeds canopy interception, then runoff is estimated. Runoff is estimated based on the CN which is a dimensionless parameter defined in such a way that $0 \leq CN \leq 100$. A high CN means high runoff. CN, for “normal” moisture conditions (i.e. antecedent moisture condition II), is tabulated in many hydrology references, e.g. Chow et al. (1988), but it should be modified for other antecedent moisture conditions (Chow et al., 1988).

For antecedent moisture condition I (dry condition):

$$CN_I = \frac{4.2CN_{II}}{10 - (0.058CN_{II})} \quad (3.59)$$

and for antecedent moisture condition III (wet condition):

$$CN_{III} = \frac{23CN_{II}}{10 + (0.13CN_{II})} \quad (3.60)$$

These moisture conditions are determined based on the total amount of the rainfall occurring in the last five days (Chow et al., 1988). Runoff is estimated as:

$$RO = \frac{(P_c - I_{ra})^2}{P_c + 0.8S} \quad (3.61)$$

where I_{ra} is the initial rainfall abstraction (mm) and S is the potential maximum retention after runoff begins (mm).

Initial rainfall abstraction is comprised of all losses before any runoff is generated. It includes water retained in surface depressions and water intercepted by vegetation, evaporation and infiltration (Rawls et al., 1996). It can be approximated as:

$$I_{ra} = 0.2S \quad (3.62)$$

The parameter S is related to the soil and cover conditions through the curve number, CN (Rawls et al., 1996). This relation is expressed as:

$$S = \frac{2540}{CN} - 25.4 \quad (3.63)$$

The infiltrated water is moved into the soil on a layer by layer basis, i.e., first all water is entered into the first layer and if it becomes saturated then water begins entering the next layer. This process continues until all layers have reached saturation or all water has entered into the root zone. In the former case, the excess water is drained out of the control volume and simply ignored.

DRAINAGE

Infiltration effectively raises soil water content to saturation. During the redistribution process, layers at a moisture content higher than field capacity release water to lower layers, until the draining layers reach field capacity. Drainage volume for each layer is calculated and this volume is entered into the next layer. When moving water into any layer, the moisture content in the receiving layer is checked. If it exceeds porosity, that layer is filled up to porosity and the remaining water is moved into the next layer.

Based on the soil water content, the hydraulic conductivity for each layer is calculated using the Brooks-Corey (1964) model as proposed by Wigmosta et al. (1994):

$$K(\theta_i) = K_{s,i} \left[\frac{\theta_i - \theta_r}{\phi_i - \theta_r} \right]^{\frac{2}{m_i + 3}} \quad (3.64)$$

where $K(\theta)$ is the soil vertical unsaturated hydraulic conductivity (mm hr^{-1}), K_s is the saturated hydraulic conductivity (mm hr^{-1}), θ is the soil water content, θ_r is the residual soil water content (water left in an air dried soil) and ϕ is the porosity. The subscript i refers to the i^{th} layer.

Discharge volume (Q) for each time step, one hour, is calculated assuming a unit hydraulic gradient. Thus

$$Q_i = K(\theta_i) \Delta t \quad (3.65)$$

where Q is discharge volume (mm) and Δt is time interval, equal to 1 hr. An hourly, rather than daily, time step is used here because the water content of draining soil can change markedly over the course of a day.

This water is entered into the next layer. Water is drained from the first layer to the second and from the second layer to the third and so on. This process is repeated 24 times to complete the daily drainage cycle. In case all layers reach field capacity the remaining water is drained out.

PLANT GROWTH

In this model a very simple, linear function is used to grow the plants through time. Plant growth is simulated in terms of canopy height and LAI. Another important output is the fractional transpiration factor, which is assumed to be a linear function of LAI. It is assumed that plant growth, in terms of LAI and canopy height, is linear from the day of the start of the growth to the date LAI reaches its maximum value. Vegetation cover is then constant until senescence starts. Then a linear decrease in LAI and canopy height is simulated. Figure 3.6 shows this pattern with arbitrary values of LAI and fractional transpiration factor.

This approach divides the simulation year in four periods in terms of the vegetative condition. These periods are dormant period, early growth period, peak LAI period and senescence period. Four key dates also known as “critical dates of growth” are

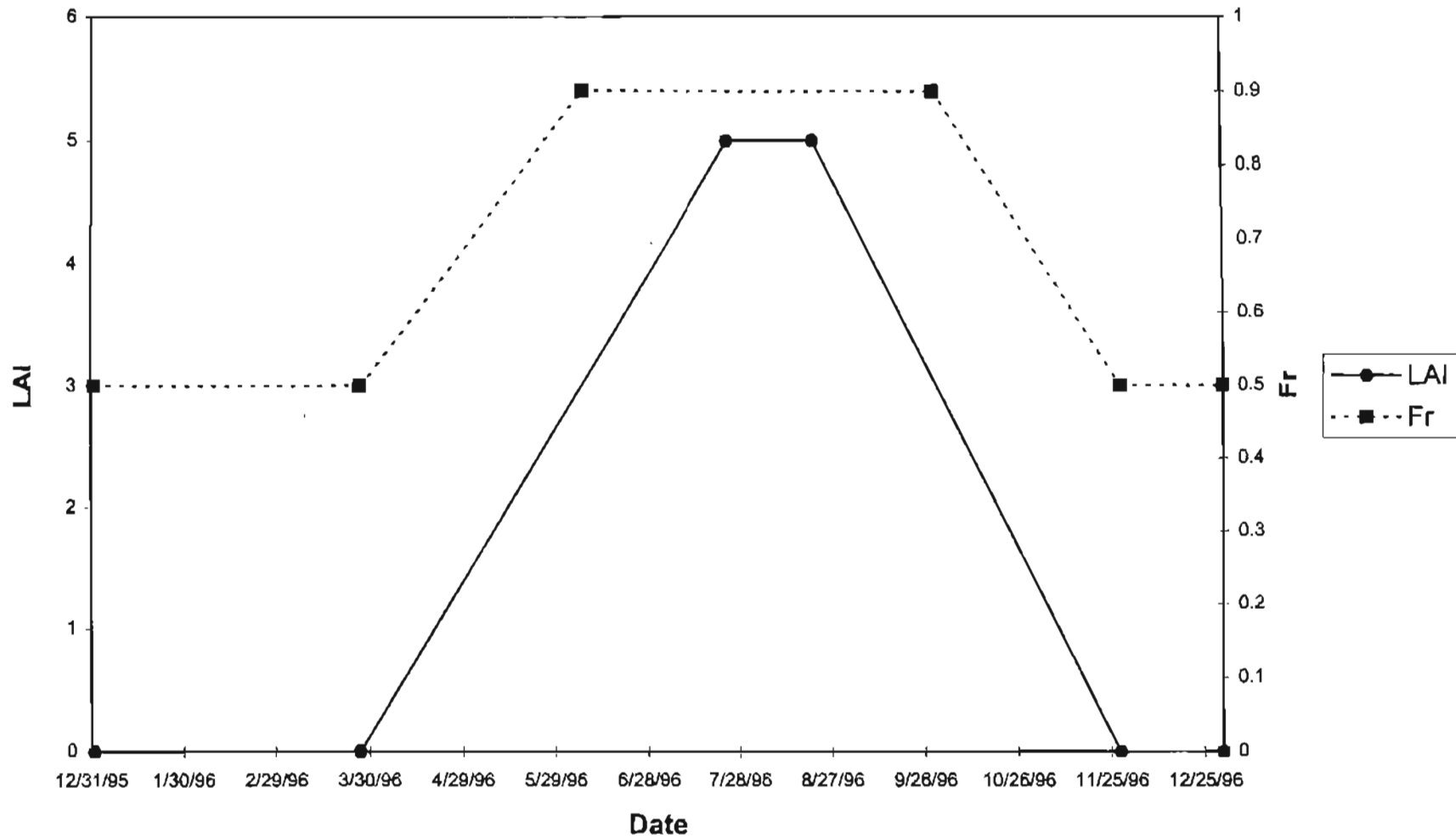


Figure 3.6. Plant growth (dates and parameter values are not specific to any particular site)

required to define the linear functions for LAI. These dates are date of the start of growth (D_{sg}), date when LAI reaches its peak (D_{mx}), date of the start of senescence (D_{ss}) and date of the start of dormancy(D_{sd}). The linear growth functions are calculated as:

$$S_G = \frac{LAI_{mx} - LAI_{mn}}{D_{mx} - D_{sg}} \quad (3.66)$$

$$S_S = \frac{LAI_{mx} - LAI_{mn}}{D_d - D_{ss}} \quad (3.67)$$

where S_G and S_S are slopes during the early growth and the senescence period, LAI_{mx} and LAI_{mn} are the maximum and minimum values of LAI respectively, D_{sg} is the date at the start of the growth, D_m is the date when LAI reaches its maximum, D_{ss} is the date at the start of the senescence and D_d is the date at the start of the dormancy period.

During the initial growing period, until LAI reaches its maximum value

$$LAI_j = LAI_{j-1} + (S_G \Delta t) \quad (3.68)$$

During the senescence period

$$LAI_j = LAI_{j-1} - (S_S \Delta t) \quad (3.69)$$

where Δt is the time interval equal to one day. Subscripts $j-1$ and j refers to the previous day and the current day, respectively.

Canopy height and fractional transpiration factor are simulated as linear functions of LAI, but the fractional transpiration factor is allowed to reach its maximum value before LAI does.

$$h_j = h_{mn} + \frac{LAI_j - LAI_{mn}}{LAI_{mx} - LAI_{mn}} (h_{mx} - h_{mn}) \quad (3.70)$$

$$Fr_j = Fr_{mn} + \frac{LAI_j - LAI_{mn}}{LAI_{MFr} - LAI_{mn}} (Fr_{mx} - Fr_{mn}) \quad \text{for } LAI_j < LAI_{MFr} \quad (3.71)$$

$$Fr_j = Fr_{mx} \quad \text{for } LAI_j \geq LAI_{MFr} \quad (3.72)$$

where h is canopy height (m), h_{mn} is minimum canopy height (m), h_{mx} is maximum canopy height (m), LAI is leaf area index, LAI_{mn} is minimum leaf area index, LAI_{mx} is maximum leaf area index, LAI_{MFr} is the LAI at which the fractional transpiration factor reaches its peak, Fr is the fractional transpiration factor and Fr_{mn} and Fr_{mx} are the minimum and the maximum fractional transpiration factors, respectively. In case of grazing or mowing a threshold LAI is assumed. LAI is not allowed to exceed this value. Thus fraction cover and canopy height, too, remain at a lower level.

Chapter 4

MODEL PARAMETERIZATION AND CALIBRATION

The following eight field monitoring sites were chosen for the model development and its validation: Apache, Boise City, Broken Bow, Goodwell, Hollis, Marena, Miami and Stillwater. The model development and initial testing were accomplished by running the model at the following four sites: Apache, Goodwell, Marena and Stillwater. These sites are geographically diverse and exhibit considerable variation in terms of soil and vegetation (table 4.1). These sites will henceforth be known as the 'development sites'. Later, the model was extended to the other sites, Boise City, Broken Bow, Hollis and Miami, for validation. These sites will henceforth be known as the 'validation sites'.

Table 4.1 Description of development sites.

Site	Latitude (N)	Longitude (W)	Vegetation	Texture Class of the Upper Soil Layer	SCS Curve Number
Apache	34.91	98.29	Bermudagrass	Loamy sand	68
Goodwell	36.60	101.60	Short grass*	Loam	61
Marena	36.06	97.21	Mixed bluestem	Sandy clay loam	74
Stillwater	36.12	97.10	Bermudagrass	Silty clay loam	80

* Predominant species are hairy grama and buffalograss.

The model requires three different types of inputs: weather, soil and vegetation data. Weather data are required to estimate the ET and the inflow to the root zone. Soil data are used to estimate the soil water parameters at various depths. Vegetation data are needed for the plant growth component of the model and for ET estimates.

The model outputs were compared against observed ET and soil moisture data to aid in developing and evaluating the performance of the model. ET was measured at three of the eight test sites (Apache, Goodwell and Marena) using weighing lysimeters installed by Fisher (1995). These sites will henceforth be known as the 'lysimeter sites'. Soil moisture was measured at all eight sites using thermal dissipation sensors (the Campbell Scientific, Inc. 229L Matric Potential Sensor) installed at four different depths: 5 cm, 25 cm, 60 cm and 75 cm.

MODEL PARAMETERIZATION

WEATHER VARIABLES

The Oklahoma Mesonet (Mesonet) is an extensive network of automated weather stations deployed across the state of Oklahoma. The network consists of 114 weather stations operating continuously in diverse, often remote locations, including the eight study sites discussed herein. The Mesonet weather stations make continuous, automatic measurements of a variety of weather and soil parameters and report the data at 15-minute intervals (Elliott et al., 1994). The data are reported via radio telemetry and the Oklahoma Law Enforcement Telecommunications System, with the data from all the

weather stations being ingested at the Oklahoma Climatological Survey located at the University of Oklahoma. The data are then made available for dissemination to users via a computer bulletin-board system and over the Internet. The capabilities and operation of the Oklahoma Mesonet are described in detail by Brock et al. (1995).

Weather data are collected continuously, summarized every five minutes and reported at 15-minute intervals by the Mesonet stations. Summaries of the 5-minute observations have been prepared (M.A. Shafer, Oklahoma Climate Survey, personal communication) to provide daily values of a variety of weather parameters for the 24-hour period, midnight to midnight. These summaries files were converted into a database format by Dr. Gabriel Senay (Oklahoma State University, personal communication). A list of the weather variables used in this model is shown in table 4.2. The data-input module of the model converts English units to SI.

Bad or missing data were flagged as -999 in the data file. There were a few such instances in the databases of 1996 and 1997. Rainfall was found missing for as long a period as 1 month for Boise City (1997). In this case, data from the Boise City cooperative observing station were used. A couple of rainfall values were missing for Miami (1997) too. In this case, the rainfall at the neighboring Mesonet stations, Jay and Vinita, were used to fill in missing rainfall values.

Some information is lost when summarizing rainfall as a daily total. The precipitation rate is important in determining the infiltration rate. An intense rainfall may result in less infiltration than that resulting from a light steady rainfall amounting to the same daily total rainfall. The timing of the rainfall is important in determining when and how much water becomes available for evapotranspiration. Rainfall occurring in the

morning is available for the evapotranspiration during the entire day, while rainfall occurring after sunset will not be subject to appreciable evapotranspiration until the next day when the evaporative energy becomes available. This model assumes that the rainfall takes place at the end of the day. So if rainfall actually occurred at the start of the day, modeled effects on ET and soil water content could be shifted by one day.

Table 4.2 Mesonet weather variables used in the model.

Variable	Symbol	Height of Measurement	Value	Units
Solar radiation	R_s	2 m	24-h total	MJ m^{-2}
Air temperature, Maximum	T_{\max}	1.5 m	24-h maximum	$^{\circ}\text{F}$
Minimum	T_{\min}	1.5 m	24-h minimum	$^{\circ}\text{F}$
Average	T_a	1.5 m	24-h average	$^{\circ}\text{F}$
Dew point temperature (derived)	T_{md}	1.5 m	24-h average	$^{\circ}\text{F}$
Precipitation	P_c	0.5 m	24-h total	inches
Wind	U	2 m	24-h average	mi h^{-1}
Atmospheric pressure	P	0.5 m	24-h average	inches of Hg

SOIL PARAMETERS

Due to the absence of any direct observed data on soil water parameters, texture-based soil parameter values were used in the model (table 4.3). Rawls et al. (1982) have documented mean values of many soil parameters solely based on the soil texture. These parameters include porosity, field capacity, wilting point, residual saturation, saturated

Table 4.3 Texture based soil parameter values (Rawls et al., 1982 and Allen et al., 1996b) as used in the model*.

Soil texture	Porosity ϕ	Field capacity θ_{FC}	Wilting point θ_{WP}	Residual saturation θ_r	Sat. hydraulic conductivity K_s (cm/h)	Bubbling pressure BP (cm)	Pore size distribution index m
Sand	0.437 (0.37-0.50)	0.12 (0.07-0.17)	0.04 (0.05-0.11)	0.02 (0.001-0.039)	21.00	7.26 (1.36-38.74)	0.592 (0.332-1.051)
Loamy sand	0.437 (0.37-0.51)	0.14 (0.11-0.19)	0.06 (0.03-0.10)	0.035 (0.003-0.067)	6.11	8.69 (1.80-41.85)	0.474 (0.271-0.827)
Sandy loam	0.453 (0.35-0.56)	0.23 (0.18-0.28)	0.10 (0.06-0.16)	0.041 (0.0-0.106)	2.59	14.66 (3.45-62.24)	0.322 (0.186-0.558)
Loam	0.463 (0.38-0.55)	0.26 (0.20-0.30)	0.12 (0.07-0.16)	0.027 (0.0-0.074)	1.32	11.15 (1.63-76.40)	0.220 (0.137-0.355)
Silt loam	0.501 (0.42-0.58)	0.30 (0.22-0.36)	0.15 (0.09-0.21)	0.015 (0.0-0.058)	0.68	20.76 (3.58-120.4)	0.211 (0.136-0.326)
Sandy clay loam	0.398 (0.33-0.46)	0.28 -----	0.13 -----	0.068 (0.0-0.137)	0.43	28.08 (5.57-141.5)	0.250 (0.125-0.502)
Clay loam	0.464 (0.41-0.52)	0.32 -----	0.17 -----	0.075 (0.0-0.174)	0.23	25.89 (5.80-115.7)	0.194 (0.100-0.377)
Silty clay loam	0.471 (0.42-0.52)	0.34 (0.30-0.37)	0.19 (0.17-0.24)	0.040 (0.0-0.118)	0.15	32.56 (6.68-158.7)	0.151 (0.090-0.253)
Sandy clay	0.430 (0.37-0.49)	0.32 -----	0.18 -----	0.109 (0.0-0.205)	0.12	29.17 (4.96-171.6)	0.168 (0.078-0.364)
Silty clay	0.479 (0.43-0.53)	0.36 (0.29-0.42)	0.21 (0.14-0.29)	0.056 (0.0-0.136)	0.09	34.19 (7.04-166.2)	0.127 (0.074-0.219)
Clay	0.475 (0.43-0.52)	0.36 (0.32-0.39)	0.21 (0.19-0.24)	0.090 (0.0-0.195)	0.06	37.30 (7.43-187.2)	0.131 (0.068-0.253)

* Single values represent means (geometric means for BP and m). Values in parentheses represent lower and upper limits for field capacity and wilting point, and \pm one standard deviation for other parameters.

hydraulic conductivity, bubbling pressure and pore size distribution index. Estimates for the field capacity and the wilting point were taken from Allen et al. (1996b), table 4.3. Since these parameters, the field capacity and the wilting point, were not given for the sandy clay loam and the clay loam textural classes, they were estimated based on their position in the USDA soil textural triangle (Hillel, 1971). These parameters (the field capacity and the wilting point) were calibrated later.

For the eight study sites, the main source of the soil texture information was data collected by soil scientists with the USDA Soil Conservation Service (R.L. Elliott, Oklahoma State University, personal communication). Based on the field observations of the soil cores taken at the sites, this data set contains information about the texture and the thickness of various layers. Also available were soil particle-size analyses of samples taken by the Mesonet technicians at four depths (5 cm, 25 cm, 60 cm and 75 cm). Where necessary, texture information was extrapolated to estimate the boundaries of the various soil layers. STATSGO (USDA, 1994) data were used as an added source of texture information. Due to its coarse resolution, the use of STATSGO was limited to help in deciding the upper and lower boundaries of the soil layers in a few cases.

For each site, the root zone was divided into various model layers based on the soil texture. The division was made in such a way that the observation points for soil water measurement (see section: "soil water content measurement") fall either at the center of the layer, approximately, or at the boundary of two adjacent layers. The thickness of any layer containing an observation point was limited to 250 mm. These requirements forced, in some cases, a homogeneous thick layer to be sub-divided into smaller layers.

Each soil layer was assigned one of the following 11 major USDA texture classes: Sand, Loamy sand, Sandy loam, Loam, Silt loam, Sandy clay loam, Clay loam, Silty clay loam, Sandy clay, Silty clay and Clay. The texture was assumed to be uniform throughout a layer. Using table 4.3, soil water parameter values for each layer were assigned based on its texture (see appendix A.1 – A.4 for the information on soil texture for each site). In order to estimate the SCS curve number for each site, the 11 texture classes were regrouped into four hydrologic soil groups (Rawls et al., 1996), table 4.4. The curve number for each site was chosen based on the texture of the upper soil layer and vegetation (table 4.1).

Table 4.4. Hydrologic soil groups

Hydrologic soil group	Soil Texture
A	Sand, Loamy sand, Sandy loam
B	Silt loam, Loam
C	Sandy clay loam
D	Clay loam, Silty clay loam, Sandy clay, Silty clay, Clay

VEGETATION PARAMETERS

The important vegetation parameters are minimum and maximum leaf area index (LAI), minimum and maximum canopy height, minimum and maximum fractional transpiration factor, root depth and minimum and maximum stomatal resistance. Other parameters include coefficient for shelter factor, fraction of available soil water depletion

at no stress and coefficient for root extraction function. At the lysimeter sites, vegetation data, LAI and canopy height, are periodically collected (W.R. Marshall, Oklahoma State University, personal communication). The measurements are made inside as well as outside the fence surrounding the lysimeter. Multiple measurements are made in each of these areas so that average values can be estimated and spatial variability taken into account. The frequency of this data collection is approximately once per month. These data were used in determining the parameters for the linear models of LAI, canopy height and fractional transpiration factor (equations 3.66 – 3.71). In 1997, the frequency of data collection was less, so those observations were used only as a guideline.

Canopy height of the vegetation is used in calculating the zero-plane displacement height and roughness lengths. Tall grasses shoot normally 60–120 cm whereas short grasses shoot 15–60 cm (Weaver, 1920). Terrel (1979) describes tall fescue as 200 cm tall grass. Canopy height of a mowed pasture is often taken as 7 cm (Allen et al., 1996b). A canopy height of 20 cm was used for grassland by Capehart and Carlson (1994). EPIC (Sharpley and Williams, 1990) uses maximum canopy heights of 150 cm for range and spring pasture and 120 cm for winter pasture. Fractional transpiration factor is estimated based on LAI. The minimum and the maximum values of this factor are assigned during model calibration.

Leaf area index

The dominant vegetation types at Apache, Goodwell and Marena are Bermudagrass, native short grass and mixed native grasses (predominantly bluestem) respectively. In 1996, average (representative) LAI values during the peak season were

observed as 6.4 at Apache, 3.0 at Goodwell and 4.8 at Marena. Growth started in late March at Marena and Apache. At Goodwell no growth was observed until May. Leaf area index reached its peak in August at Goodwell and Apache. At Marena, peak LAI was observed in late July. Senescence started in August-September and LAI declined to zero by November at all three sites.

The dominant vegetation type at Stillwater is bermudagrass. At this site, the vegetation was parameterized based on the literature and observations at Apache. Grass is periodically mowed at this site. Therefore, it was assigned a maximum LAI of 6 and a cutoff LAI of 2. The cutoff LAI was adjusted during the calibration. The estimates of the minimum and the maximum LAI for all the development sites are listed in table 4.5.

Canopy height

In 1994 and 1995, the mean maximum canopy height reported at Apache was nearly 30 cm and that at Marena was 40 cm. At Goodwell, the mean maximum canopy height was less than 10 cm. In the winter of these years, the canopy height at Apache and Marena had dropped to nearly 10 cm. At Goodwell, it had dropped as low as 1 cm. In 1996, vegetation showed increased growth in terms of canopy height at the lysimeter sites. The mean maximum canopy heights, in 1996, were 30 cm, 50 cm and 80 cm at Goodwell, Apache and Marena, respectively.

At Stillwater, grass does not grow too tall due to the mowing. The observed canopy height in the summer 1997 was in the range of 1-4 cm. Therefore the minimum canopy height was assigned a value as low as 1 cm. The maximum canopy height

(without mowing) was taken as 50 cm, based on the observations made at Apache. The estimates of the minimum and the maximum canopy heights for all the development sites are listed in table 4.5.

Table 4.5. Initial estimates of vegetation parameters at the development sites

Parameter	Apache	Goodwell	Marena	Stillwater
Vegetation	Bermudagrass	Short grass*	Mix. Bluestem	Bermudagrass
LAI _{mn}	0	0	0	0
LAI _{mx}	6.4	3.0	4.8	6
Min. canopy height (cm)	10	1	30	1
Max. canopy height (cm)	50	10	80	50
Root depth (cm)	80	60	80	80
Min. stomatal resistance (s m ⁻¹)	166	166	166	166
Dates of Growth				
Start of growth (D _{sg})	March 26	May 9	March 26	March 26
LAI at peak (D _{mx})	Aug. 15	Aug. 3	July 23	Aug. 15
Start of senescence (D _{ss})	Aug. 31	Sep. 9	Aug. 20	Aug. 31
Start of dormancy (D _{sd})	Nov. 27	Nov. 11	Nov. 4	Nov. 27

* Predominant species are Hairy grama and Buffalograss

Critical dates of growth

The observed data, particularly LAI, indicate that the grass starts to grow in March-April and reaches its peak in July-August. Then after some time LAI starts to decline, in late August or early September, and reaches its initial value, zero in this case, in October-November. The crop growth model requires the following four dates to

simulate the crop growth on a daily basis: the onset of greenness, when LAI reaches its peak, the onset of senescence and the onset of dormancy. Since LAI is measured only once a month, it was quite difficult to estimate these dates. Observed LAI values were interpolated and thus inference was used to estimate these critical dates of growth.

Senay and Elliott (1997) used a combination of high spatial resolution landcover data and high temporal resolution NOAA AVHRR NDVI data to characterize the temporal variability in vegetative activity for major landcover categories in Oklahoma. NDVI (Normalized Difference Vegetation Index) is a measure of the greenness of the vegetation. Temporal plots of NDVI for the entire year for various landcover classes were used to analyze the growth pattern of the vegetation. These data were used to help determine the critical dates of growth. The critical dates of growth for all the development sites, as used in the model, are listed in table 4.5.

Root depth

In this model, root depth is defined as the “working depth” of the roots, as opposed to the maximum depth at which roots are found. This depth contains the majority of the roots and almost all water for ET is extracted from within this depth. Since no observed data were available for root depth, it was inferred based on the values given in the literature and the observed pattern of soil water content.

Weaver (1920) observed roots of little bluestem as deep as 167 cm. But he found that the sod was formed at a depth of 82 cm. The maximum root depth of bermudagrass is reported to be about 200 cm whereas that of tall fescue is about 122 cm (Donald et al.,

1991). Roots of short grass usually go down to 60-70 cm deep (Weaver, 1920). Dickinson et al. (1986) used a depth of 80 cm for all sorts of grasses. Capehart and Carlson (1994) used a rooting depth of 50 cm for grassland. EPIC (Sharpley and Williams, 1990) uses the maximum root depth of alfalfa, rangeland and pasture as 200 cm. Initially a root depth of 80 cm (Dickinson et al., 1986) was used at all development sites except Goodwell. At Goodwell, the main vegetation type is native short grass. Therefore a root depth of 60 cm was used at this site. During the calibration process, values in the range of 50 – 120 cm were tested. The estimates of root depth for all the development sites are listed in table 4.5.

Stomatal resistance

The minimum stomatal resistances were taken from the literature. A wide range of canopy resistance values for grass have been reported in the literature (Fisher, 1995). Fisher (1995) estimated the minimum stomatal resistance as 166 s m^{-1} at Marena and Goodwell, and 250 at Apache. Komer et al. (1979) reported 166 s m^{-1} as the average minimum stomatal resistance for grass. The minimum stomatal resistance values for grass as used in some water balance models are 250 s m^{-1} (Dickinson et al., 1986), 120 s m^{-1} (Wigmosta et al., 1994), 100 s m^{-1} (Capehart and Carlson, 1994), etc. The maximum stomatal resistance was fixed at 5000 s m^{-1} (Wigmosta et al., 1994). The estimates of the minimum stomatal resistance for all the development sites are listed in table 4.5.

Other parameters include coefficient for the root extraction function (C_r), fraction of soil water depletion at no stress (F_{ns}) and minimum and maximum fractional transpiration factor (Fr_{mn} and Fr_{mx}). Zhang et al. (1993) estimated the coefficient (b) for

the root distribution function for peanut. It varied from 0.0019 to 0.0029 through the year. During summer it varied from 0.0022 to 0.0027. Though peanuts are different from grasses, this study gave an idea of a reasonable range for this parameter. Later, various estimates of this parameter were tested during the calibration process and results were analyzed to obtain the best estimate. The fraction of soil water depletion at no stress varies from 0.4 to 0.6 for various vegetation types (Allen et al., 1996b). The initial values for C_p , F_{ns} , Fr_{min} and Fr_{mx} were taken as 0.0027, 0.5, 0.3 and 0.9. The fractional transpiration factor is assumed to be highly correlated with the fraction ground cover i.e. fraction of the ground covered by the vegetation. The estimate for the minimum fraction cover was based on the assumption that even during the dormant period a fraction of the ground is covered by vegetation though it does not transpire. The estimate for the maximum fractional transpiration factor was based on the assumption that even during the peak growing season there is some soil evaporation. Later these values were adjusted during the calibration process.

ET MEASUREMENT

Weighing lysimeters had been installed and calibrated (Fisher and Elliott, 1994; Fisher, 1995) at four Mesonet sites: Apache, Goodwell, Marena and Wister. Of these, three sites were included in this study: Apache, Goodwell and Marena. Lysimeter data are collected remotely via connection to the Mesonet station near each lysimeter. The datalogger at the weather station is programmed to make lysimeter measurements at 30-

second intervals by sending an excitation voltage to each of four loadcells and measuring the signal voltage from the loadcell. At 15-minute intervals, the average of the thirty 30-second measurements taken during the period is calculated and reported for each loadcell.

At the end of each day, a data file containing loadcell voltages is created on the Mesonet computer operated by the Oklahoma Climatological Survey. The data file contains average signal voltages for each loadcell at 15-minute intervals throughout the day. The daily files are then available for downloading. The voltage readings are converted to weights using calibrations determined by Fisher (1995). Due to some temperature effects on the data, daily ET is calculated as the change in average lysimeter weight from sunrise on one day to sunrise on the next day. These daily ET values were obtained from W.R. Marshall (Oklahoma State University, personal communication).

On days with rainfall, many of the reported ET values were negative indicating gain in the weight of the lysimeters. On such days, lysimeter observations of ET are not as reliable because of the difficulty in separating out the rainfall effects. So it was decided not to use the observations on the days of rainfall in model development and evaluation. Since the rainfall data are from midnight to midnight and lysimeter data are from sunrise to sunrise, the rainfall could affect the lysimeter observation of the same day (if it occurs after sunrise) or that of the preceding day (if it occurs before sunrise). Therefore, each day with the rainfall and the preceding day were excluded from the ET data set. Also excluded were the days on which ET measurements reported by the lysimeters exceeded the energy available in the measured incoming solar radiation. In theory, the ET should not exceed the net radiation assuming no advection of sensible heat. Solar radiation, instead of net radiation, was selected as a threshold to avoid discarding too much data.

SOIL-WATER CONTENT MEASUREMENT

The Campbell Scientific, Inc. 229L Matric Potential Sensors have been installed at many of the Mesonet sites to facilitate the study of the behavior of soil-water in the root zone. All the eight study sites have these sensors, in fact presence of these sensors was a criterion for site selection. The sensors were installed at different times at different sites. Some of the study sites had these sensors installed and working in the latter part of 1996, e.g. Marena, whereas some sites got them as late as early 1997, e.g. Hollis.

The sensor operates on the heat dissipation principle. To make a matric potential measurement, the temperature of the sensor is first measured and recorded. A constant current is then supplied to the sensor and the sensor is heated. The current is switched off and the temperature of the sensor after heating is measured and recorded. The temperature rise due to heating is calculated and output as the sensor response. The temperature rise, ΔT , is dependent on the amount of heat dissipated through the sensor ceramic matrix, which is a function of its water content. The water content of the ceramic matrix is a function of the matric potential of the soil.

These instruments were installed at four different depths: 5 cm, 25 cm, 60 cm and 75 cm. Observations of ΔT , at these depths, are reported for every 30-minute interval. Daily files of 30-minute observations are available for downloading. These observations are incorporated into a database in order to simplify data sorting and extraction.

The observations of ΔT can be converted into soil-water potential. The estimates of soil-water potential, in turn, can be transformed into soil-water content. Fisher's

approach (R.L. Elliott, Oklahoma State University, personal communication) was used for this transformation. He first adjusted individual sensor response to a reference sensor response:

$$\Delta T_{ref} = m * \Delta T_{sensor} + b \quad (4.1)$$

where ΔT_{ref} is reference sensor response (C), ΔT_{sensor} is individual sensor response (C), m is slope and b is intercept.

Now these reference sensor responses can be converted into soil-water potential:

$$\psi = \frac{1}{a} \left[\frac{\Delta T_w - \Delta T_d}{\Delta T_{ref} - \Delta T_d} - 0.9 \right]^{1/n} \quad (4.2)$$

where ψ is soil water potential (kPa), ΔT_w equals 4.00 (C), ΔT_d equals 1.45 (C), a equals -0.01 (kPa⁻¹) and n equals 0.77. These parameter values (ΔT_w , ΔT_d , a , and n) resulted from a detailed laboratory calibration procedure.

Soil water content from soil water potential is determined as:

$$\theta = \theta_r + \frac{\theta_s - \theta_r}{\left[1 + (x(-\psi/100))^y \right]^{(1-1/y)}} \quad (4.3)$$

where θ , θ_s and θ_r are soil water content, soil water content at saturation and residual soil water content, respectively and x and y are empirical constants. Fisher used the method of Arya and Paris (1981) to estimate the soil water release characteristics from the particle size distribution information. He then ran a fitting program RETC (van Genuchten et al., 1991) to estimate the parameters required in equation 4.3. These coefficients are given in appendix C. Obviously, the resulting values of θ are, at best, approximations of the actual water contents.

The data set of the estimated soil water contents showed occasional, sudden jumps in either direction. Malfunctioning of the sensors (perhaps due to electronic noise) could be one of the reasons, as these jumps did not necessarily follow rainfall events. The data set was filtered to remove these questionable data. Any 30-minute jump greater than 5% water content and in the absence of rainfall was assumed to be unrealistic and the corresponding value was discarded. If the sensor response remained at this higher/lower level before returning to the “pre-jump” level, these data were removed as well. This problem was worst at Apache, where these criteria led to approximately half of the data being discarded. At other sites, the situation was much better and only a few values were filtered out of the data set.

MODEL CALIBRATION

The model was calibrated at Apache, Goodwell, Marena and Stillwater. ET predictions were compared against lysimeter observations and soil water predictions were compared against 229L Matric Potential Sensor observations. The ΔT observations were converted into soil water content estimates using Fisher’s scheme. At each site, the root zone was divided into various layers, based on the soil texture, in such a way that each of these points fall either at the center of any layer, approximately, or at the boundary of two neighboring layers. In the former case, soil water predictions for that layer were compared only against the observations made within that layer. In the latter case, an average soil water content was calculated for the two layers. The average soil water content, thus computed, was compared against the 229L observations made at the

boundary. Table 4.6 lists, for the development sites, the thickness of various layers and their texture.

Table 4.6. Depths (mm) and texture of soil layers at development sites.

	Apache	Goodwell	Marena	Stillwater
Layer-1	0 – 150 (Loamy sand)	0 – 150 (Silt loam)	0 – 150 (Sandy clay loam)	0 – 150 (Clay loam)
Layer-2	150 – 300 (Sandy loam)	150 – 350 (Silt loam)	150 – 300 (Loam)	150 – 300 (Silt loam)
Layer-3	300 – 550 (Clay loam)	350 – 430 (Silt loam)	300 – 425 (Loam)	300 – 450 (Silt loam)
Layer-4	550 – 650 (Clay loam)	430 – 550 (Silty clay loam)	425 – 550 (Sandy clay loam)	450 – 550 (Silty clay loam)
Layer-5	650 – 700 (Clay loam)	550 – 650 (Silty clay loam)	550 – 650 (Sandy clay loam)	550 – 650 (Silty clay loam)
Layer-6	700 – 800 (Clay loam)	650 – 700 (Silty clay loam)	650 – 700 (Sandy clay loam)	650 – 700 (Silty clay loam)
Layer-7	800 – 900 (Clay loam)	700 – 800 (Silty clay loam)	700 – 800 (Sandy clay loam)	700 – 800 (Silty clay loam)
Layer-8	900 – 1000 (Clay loam)	800 – 900 (Silty clay loam)	800 – 1000 (Sandy clay loam)	800 – 1000 (Silty clay loam)
Layer-9	1000 – 1250 (Clay loam)	900 – 1050 (Silty clay loam)	1000 – 1250 (Sandy clay loam)	1000 – 1250 (Silty clay loam)

CALIBRATION PROCESS

Parameters adjusted during the calibration process include field capacity, wilting point, minimum stomatal resistance, coefficient for shelter factor, cutoff LAI, soil

moisture depletion for no stress, minimum and maximum fractional transpiration factor, root-depth, parameter for root extraction and the critical dates of growth (table 4.7).

Table 4.7. Calibration parameters

PARAMETERS	RANGE	SOURCE FOR RANGE
Field capacity and wilting point	± 3% VWC*	Arbitrary
Root depth, cm	50 – 120	Literature, Observation
Min. stomatal resistance, s m ⁻¹	100 - 250	Literature
Min. fractional transpiration factor	0.4 – 0.6	Cover, observation
Max. fractional transpiration factor	0.8 – 0.9	Cover, observation
Coefficient for shelter factor	0.1 – 0.3	Arbitrary
Cut-off LAI	1.5 – 2.0	Observation
Fraction of ASW** at no stress	0.25 - 0.6	Literature
Parameter for root distribution	0.0020 - 0.0030	Literature
Critical dates of growth		Literature, Observation
Onset of growth	March-April	
Peak	June-August	
Onset of Senescence	Sep.-October	
End of Senescence	Nov.-December	

* Volumetric water content

** Available soil water

An iterative approach was used to calibrate the parameters. First the model was run with the initial estimates of the calibration parameters, as listed in tables 4.3 and 4.5 and discussed in the ‘‘Model Parameterization’’ section. The results were compared with the observations and changes in the parameter estimate were made. Each time the model was run, results were analyzed to make a decision which parameter to change. The estimate for that parameter was changed by a reasonable amount so that it was within the

prescribed limit (table 4.7). The model would be run again and the results would be compared. Then the first parameter would be brought back to its original estimate and another parameter would be changed and the results would be analyzed. Finally it was decided with which parameters to start calibration. Then parameters were adjusted one after another and the results were analyzed. This procedure, first sensitivity analysis and then calibration, was adopted to minimize the possibility of adjusting a parameter which does not really need a change.

First, the model predictions for ET were compared against the observed ET values at the lysimeter sites, and the parameters which affect ET most were selected, e.g. minimum stomatal resistance. Then changes in the estimates of these parameters were made and the model results were compared with the observed ET values. This process continued until a reasonably good match was found. Similarly all the parameters were adjusted one after another.

Then the model results were compared with the 229L observations. This comparison was made to adjust the parameters which affect the soil water profile, e.g. root depth. The procedure was similar to the one adopted for ET comparisons. Root depth was the main parameter adjusted in this part of the calibration, though effects of other parameters were also analyzed and they were adjusted if required. In this part of the calibration, the main aim was to match the pattern of the soil water profile, not the actual estimates of soil water content. This is due to the fact that the “observed” volumetric soil water data are in fact derived (from 229L ΔT observations).

At Apache, the model first underestimated the ET grossly when LAI was at its peak. The drop in ET was caused by the shortage of available soil water in the root zone.

Increasing root depth to 100 cm improved the result. Increasing root depth beyond that did not affect the predictions much. A root depth of 100 cm was used at other sites, except Goodwell, and the results were compared with the observations. At Goodwell, the soil water estimates for 1997 showed very little extraction from the soil at 75 cm. Increasing root depth to 80 cm at this site slightly improved the soil water predictions. Increasing root depth beyond that did not affect the predictions.

The field capacity and the wilting point were adjusted only at the end of the calibration process in an attempt to match the soil water profile. At Apache, the field capacity of sandy clay loam was changed from 28% to 25% and that of loam was changed from 26% to 23%. This greatly improved the predicted soil water profile. It also improved the ET predictions. At Stillwater, the field capacity and the wilting point of the clay loam, the soil texture of the upper layer, were adjusted. The field capacity was reduced from 32% to 31% and the wilting point was reduced from 17% to 14%.

While comparing the soil water predictions with the 229L observations, it was felt, at times, that changes were required in the parameters which affect ET. In such cases, changes in those parameters were made and the ET predictions as well as soil water predictions were compared with the observations. Appropriate parameters were adjusted so that both ET and soil water predictions could be improved. For example, increasing root depth allowed roots to extract water from a greater pool of the water. This changed the stress condition as now stress would be felt later. Thus appropriate changes were required in other parameters, e.g. F_{ns} .

After calibration had been performed at all the development sites, it was found that some parameters varied only slightly from site to site. It was decided to arrive at one

estimate of such parameters at all sites if the model is not highly sensitive to these parameters.

The critical dates of growth were calibrated based on the observed soil water content and ET (table 4.8). Interestingly, the minimum stomatal resistance remained at its initial estimate at the end of the calibration.

Table 4.8. Site specific vegetation parameters and their values after calibration.

SITE	VEGETATION	r_{smin} (s m ⁻¹)	Z_{root} (mm)	DATES OF GROWTH			
				1	2	3	4
Apache	Bermudagrass	166	1000	Mar. 26	Jul. 30	Aug. 31	Nov. 12
Goodwell	Short grass	166	800	May 9	Aug. 3	Sep. 09	Nov. 11
Marena	Mixed bluestem	166	1000	Mar. 26	Jul. 23	Aug. 20	Nov. 04
Stillwater	Bermudagrass	166	1000	Mar. 26	Jul. 30	Aug. 31	Nov. 12

Other parameters were calibrated to arrive at a common estimate for all sites (table 4.9). Minimum and maximum fractional transpiration factor might change from site to site, but it was found that they were quite similar at all the development sites. The coefficient for shelter factor was fixed at 0.1, thus the final equation for the shelter factor is:

$$P_s = 0.1LAI + 1.2 \quad (4.4)$$

Table 4.9. Common parameters and their values after calibration.

PARAMETERS	VALUE
Coefficient for root distribution	0.0026
Fraction of ASW depletion at no stress	0.5
Minimum fractional transpiration factor	0.5
Maximum fractional transpiration factor	0.9
Coefficient for shelter factor	0.1

CALIBRATION RESULTS FOR THE DEVELOPMENT SITES

Apache

At Apache, the model tended to underestimate the ET, especially in 1996 when LAI was at its peak (figure 4.1). Predicted versus observed ET is plotted in figure 4.2 and the associated statistical parameters are presented in table 4.10. The low coefficient for the regression slope and the comparison between average predicted and observed ET are further evidence of the model's underestimation. In June 1996, the model shows a drop in the ET. It is due to the shortage of water which has raised the canopy resistance. But no such drop was observed in the lysimeter data. The average predicted ET, 2.20 mm, is comparable with the average observed ET, 2.77 mm. But the average absolute deviation is 1.39 mm. This indicates that there were also periods of overestimation (figures 4.1 and 4.2).

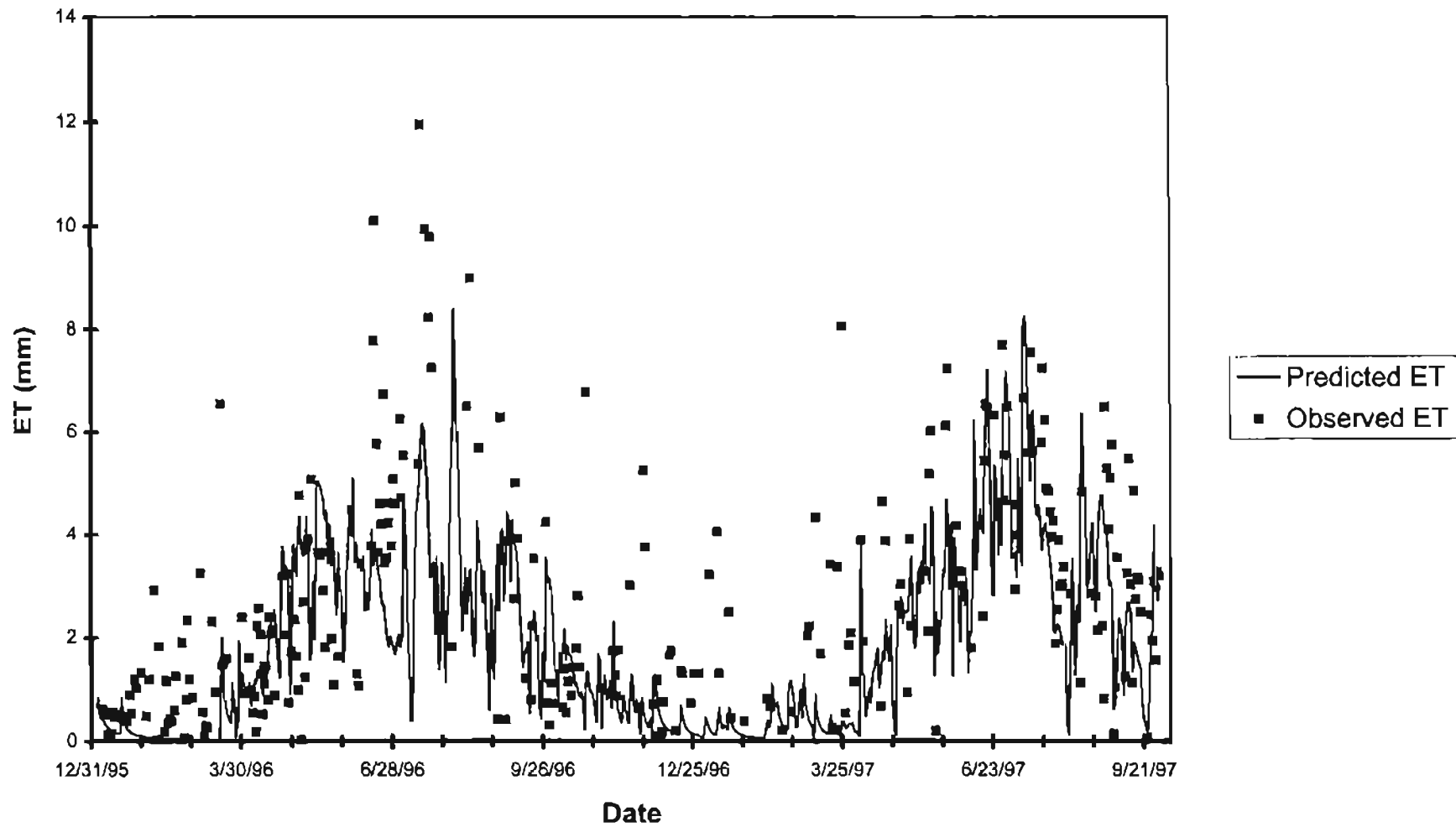


Figure 4.1. ET for Jan. 1996 through Sep. 1997 at Apache

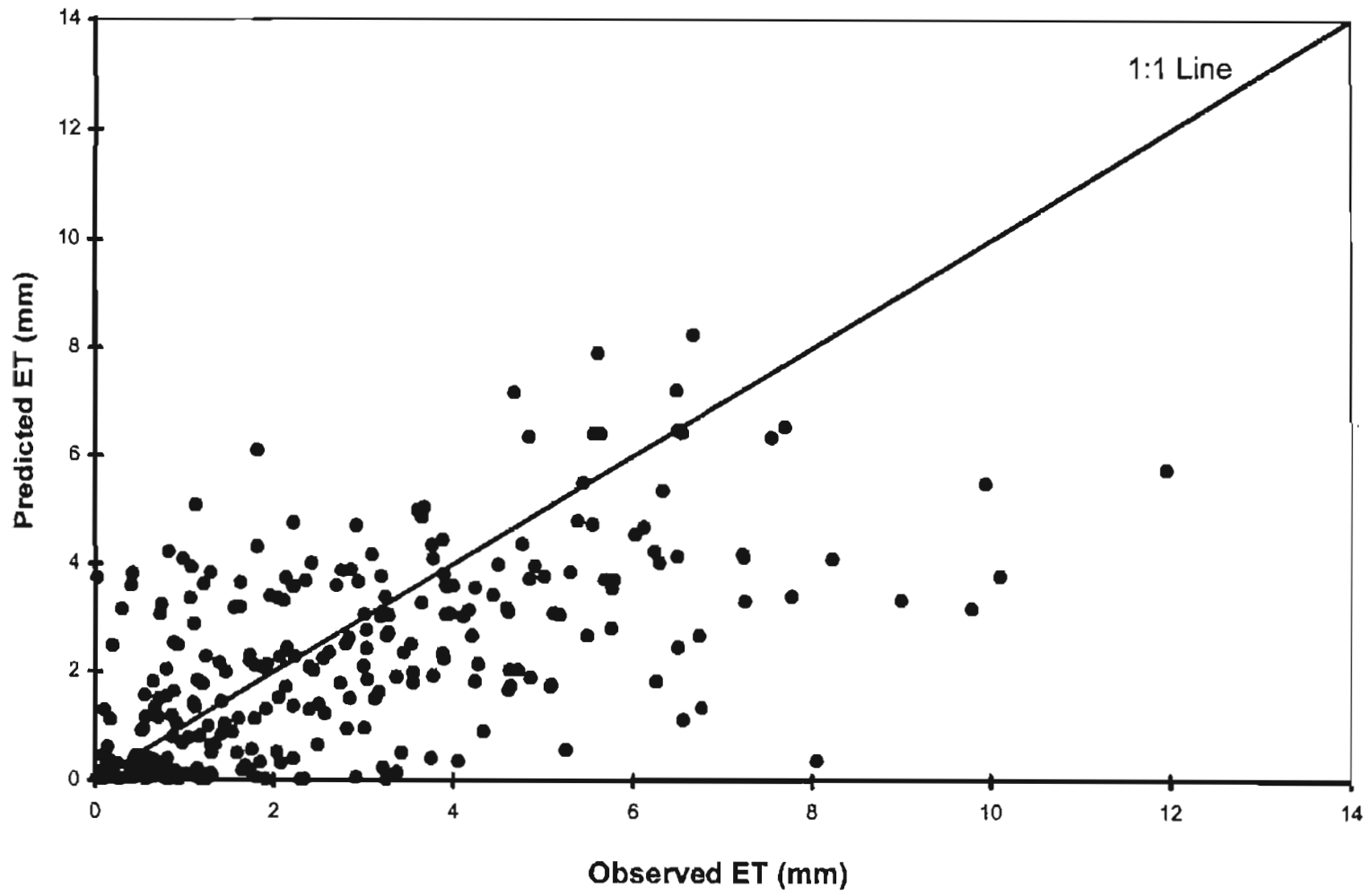


Figure 4.2. Comparison of predicted and observed ET for Jan. 1996 through Sep. 1997 at Apache

Table 4.10. Comparison between the observed and the predicted ET at the lysimeter sites.

Statistics	Apache	Goodwell	Marena
Intercept (mm)	0.85	-0.09	1.08
Slope	0.49	0.83	0.76
r ²	0.37	0.66	0.35
Standard error (mm)	1.44	1.00	1.76
Average			
Observed ET (mm)	2.77	1.95	1.95
Predicted ET (mm)	2.20	1.52	2.57
Median			
Observed ET (mm)	2.22	1.33	1.33
Predicted ET (mm)	2.00	0.85	2.06
Absolute deviation			
Maximum (mm)	7.67	5.37	6.912
Minimum (mm)	0.00	0.00	0.00
Average (mm)	1.39	0.82	1.34

Model predictions of soil water were compared with the 229L observations (figures 4.3 through 4.6). Since data collection started towards the end of the year 1996, only data from 1997 were used for this comparison. For the 1997 simulation, layers were initialized with the moisture contents estimated by the model at the end of the year 1996. This resulted in some layers being initialized at a moisture content less than their field capacity. The rainfall in February raised the water content of these layers to field capacity (figures 4.5 and 4.6). But the observed data do not show any such increase in the water content except for the first layer. 229L observations present a moisture profile which more or less remained constant through out the root zone, except the first layer, in the

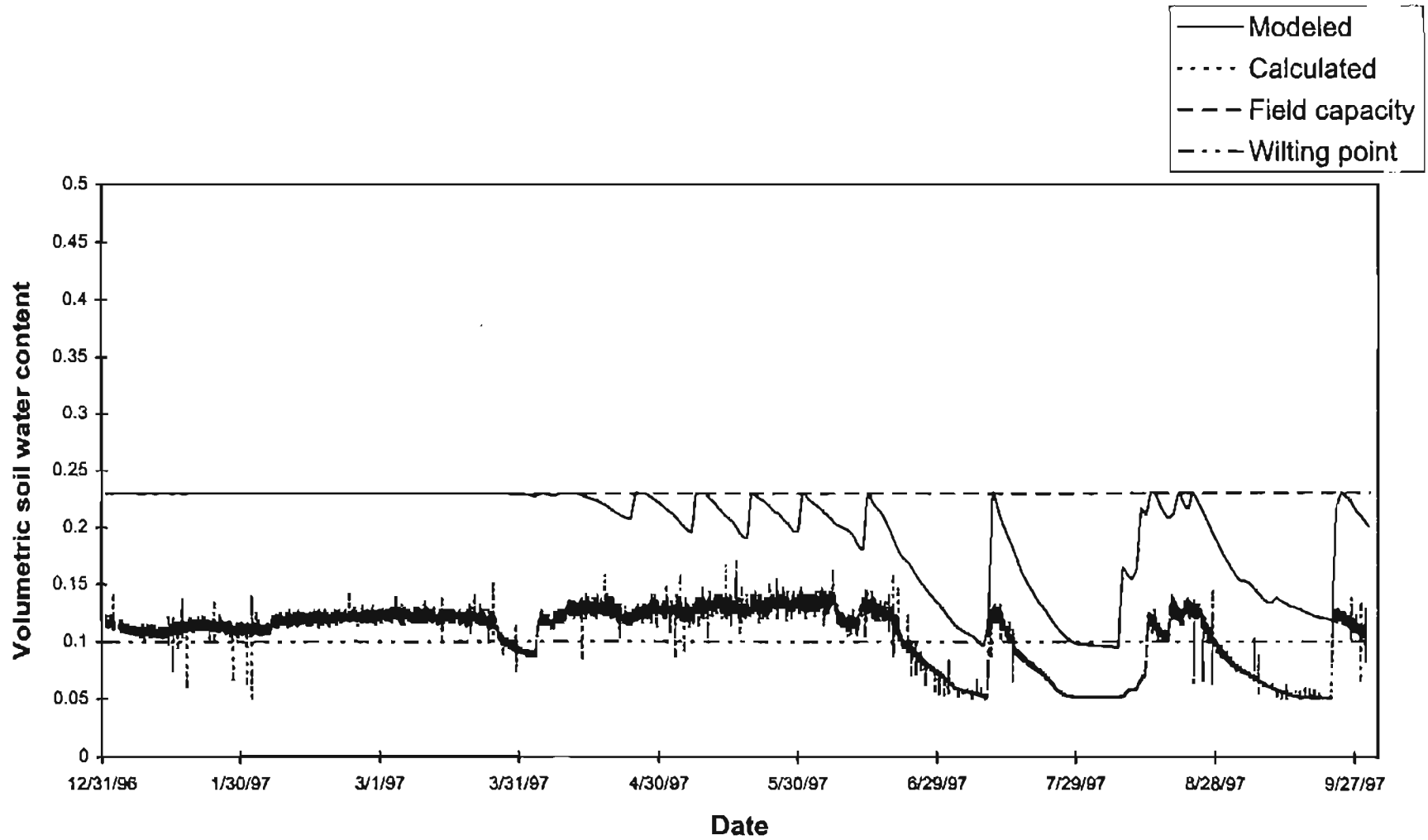


Figure 4.3. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Apache

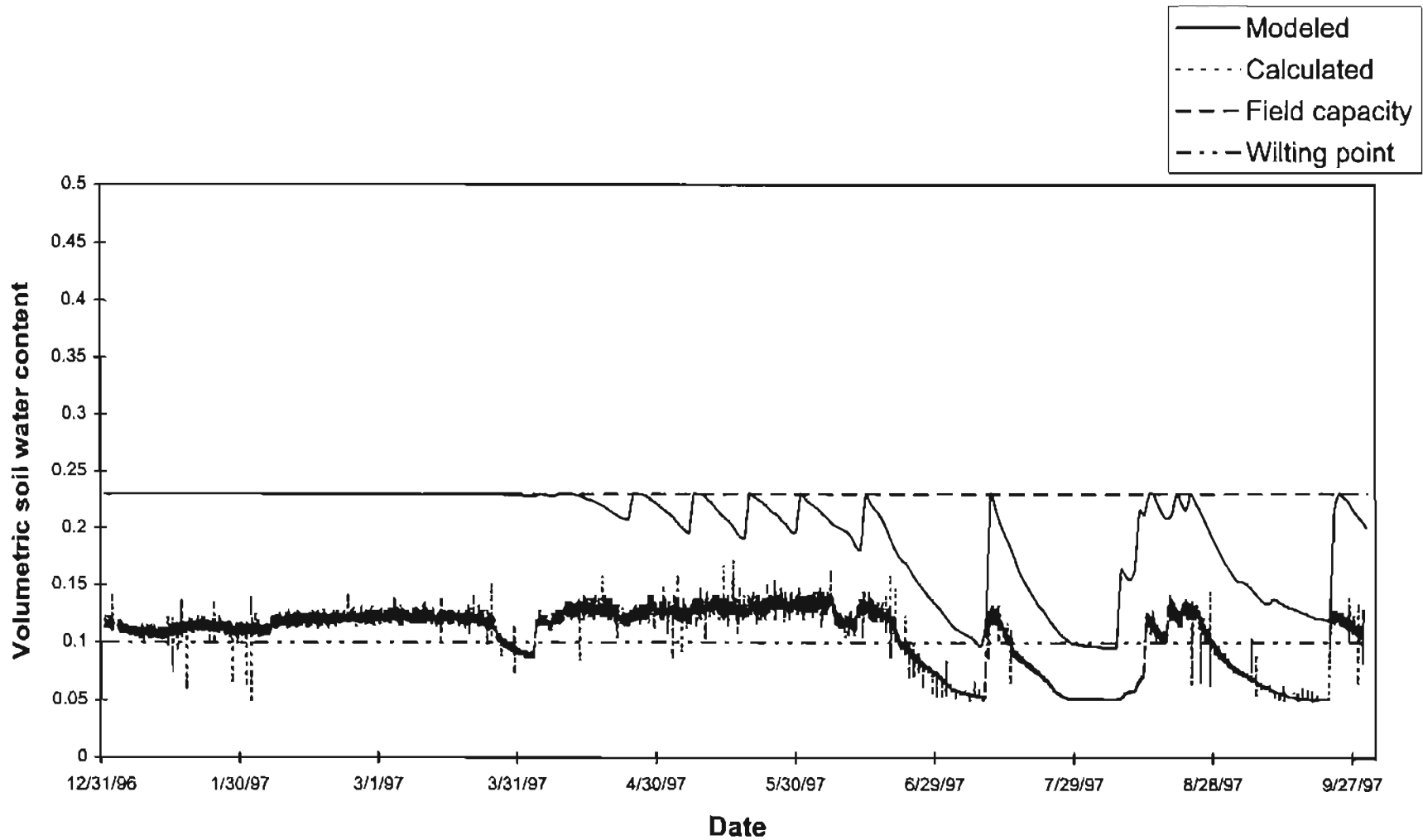


Figure 4.4. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Apache

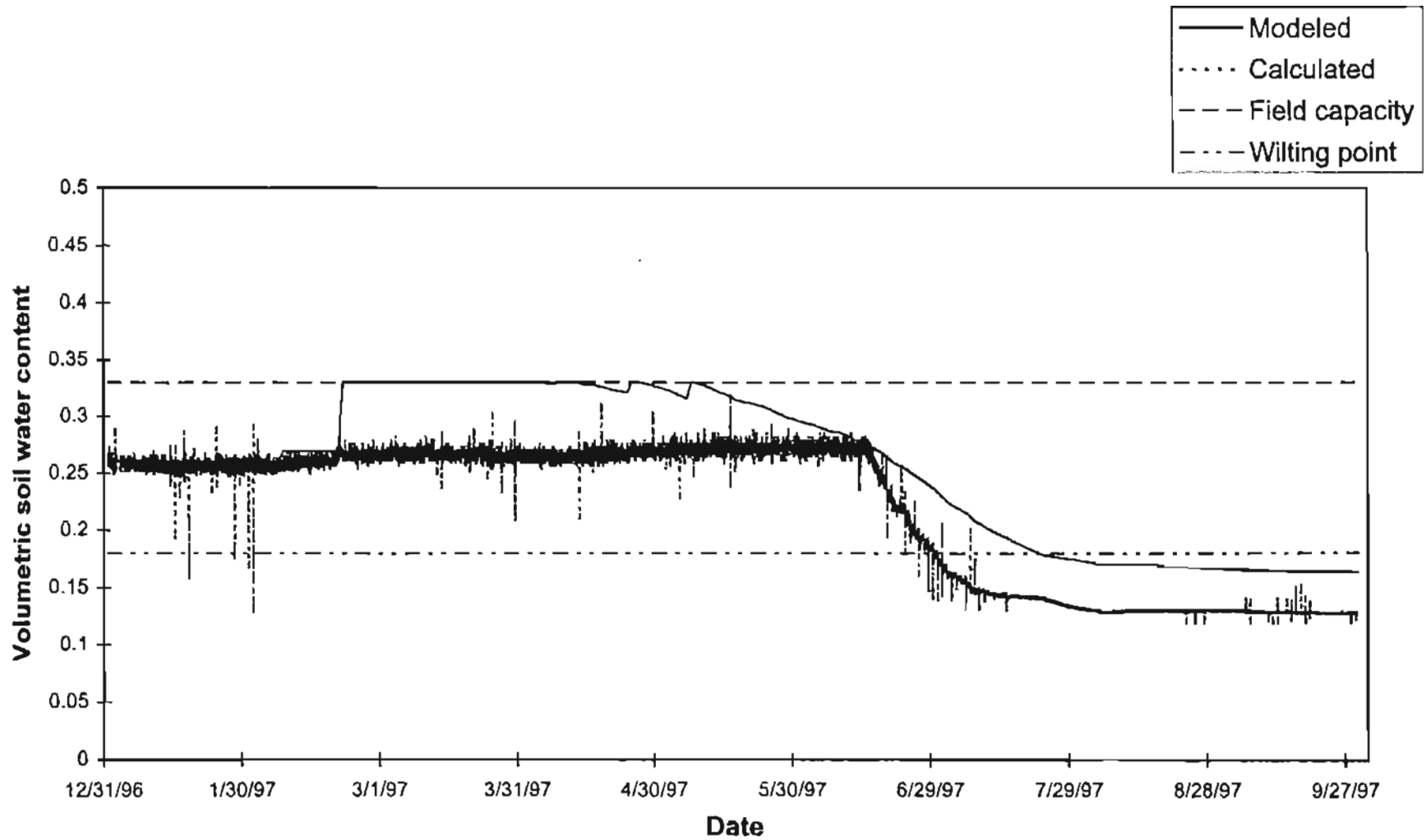


Figure 4.5. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Apache

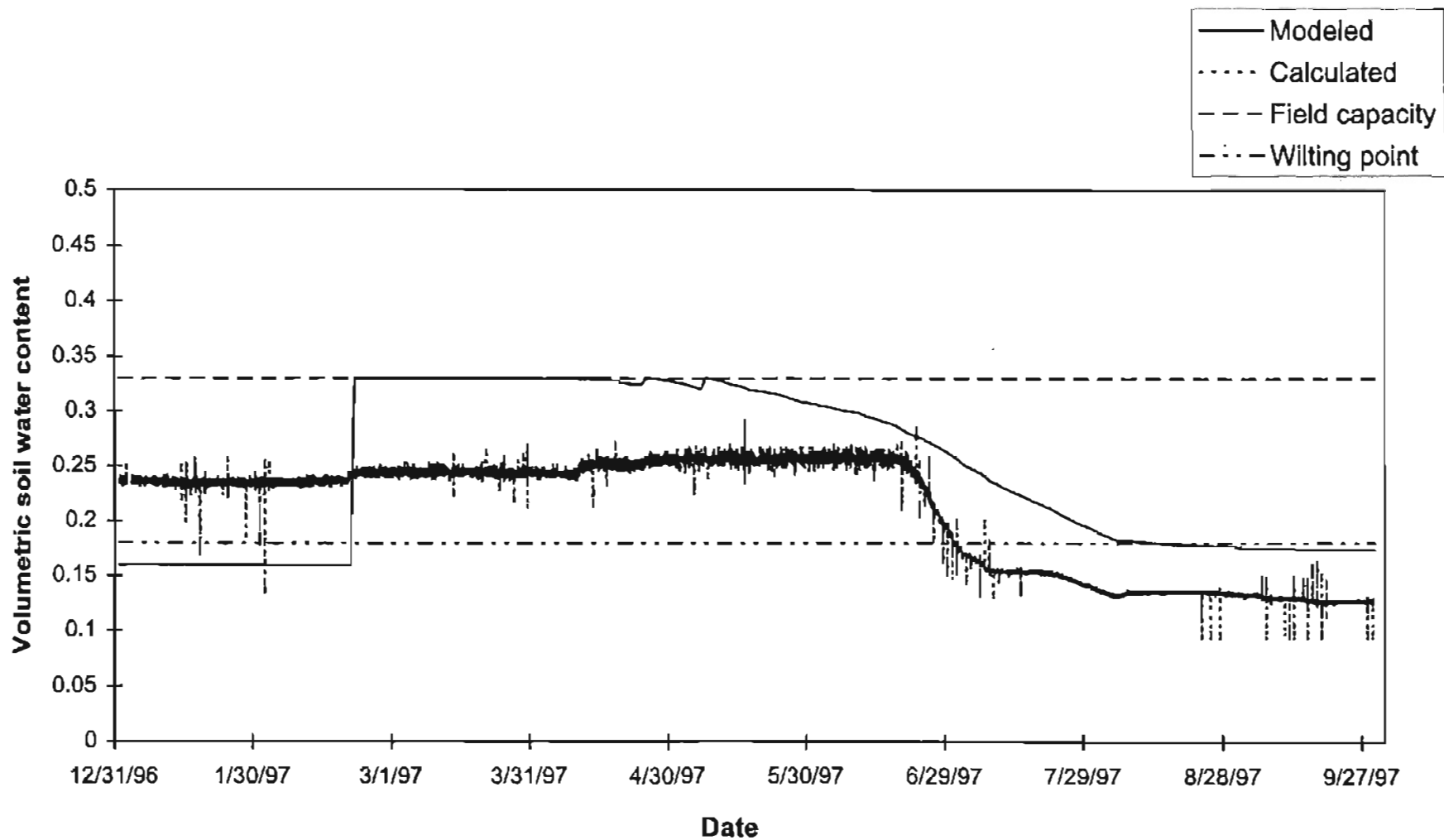


Figure 4.6. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Apache

dormant period. The soil water content of the lower layers, especially at 60 cm and 75 cm, begin to decrease later than predicted by the model.

Goodwell

At Goodwell, model predictions compare very well with the observed lysimeter ET measurements (figures 4.7 and 4.8). This fact is also evident from the regression results (low intercept, high slope and high r^2 (table 4.10)). The average predicted ET, 1.52 mm, and the average observed ET, 1.95 mm, are comparable (table 4.10). The average absolute deviation, 0.82 mm, is low as compared to other sites (table 4.10).

Soil water predictions were compared with the 229L observations. The observations show the first layer nearly static until the fourth week of April (figure 4.9) though it is expected to be more dynamic due to soil evaporation. From the middle of May to the first week of August 1996 observations do not show any major increase in the soil water content of the first layer, whereas records show many instances of rainfall within this period. The model predicted a more dynamic first layer. The model prediction compares well with the observations at 25 cm (figure 4.10). The predictions follow the trend of the observations throughout the simulation period. Since a shorter root depth (80 cm) was used at Goodwell, not much water was extracted from deeper layers. The predictions show a small but steady decrease in the soil water contents at 60 cm and 75 cm (figures 4.11 and 4.12). The observations at 60 cm do show a decrease in the soil water content but it is more rapid. Also the observations at this depth show an increase in the soil water content in the last week of April, which is not matched by the predictions. The observations at 75 cm show minor fluctuations in the soil water regime throughout the simulation period (figure 4.12). Since the fluctuation in this layer is not accompanied

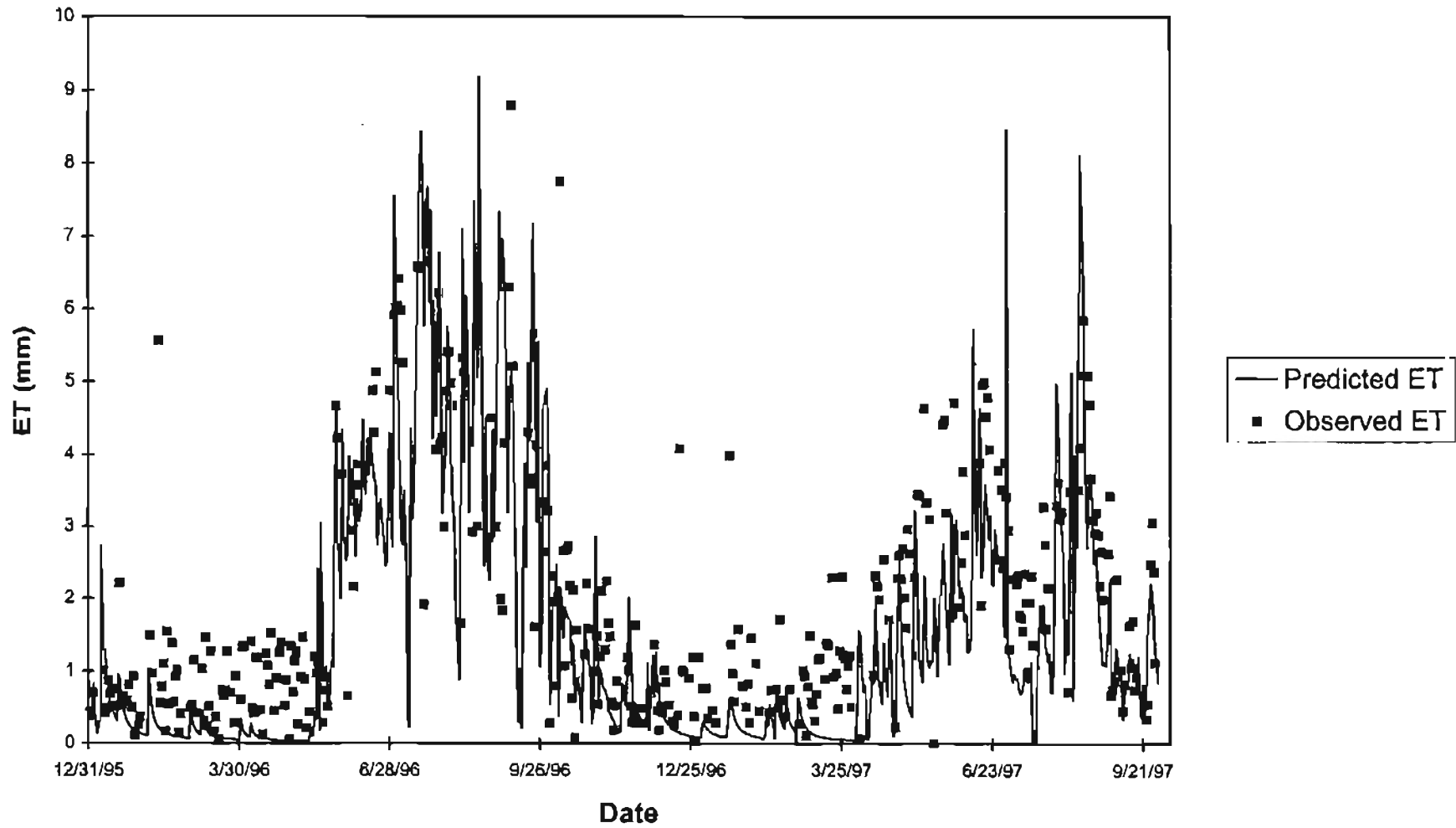


Figure 4.7. ET for Jan. 1996 through Sep. 1997 at Goodwell

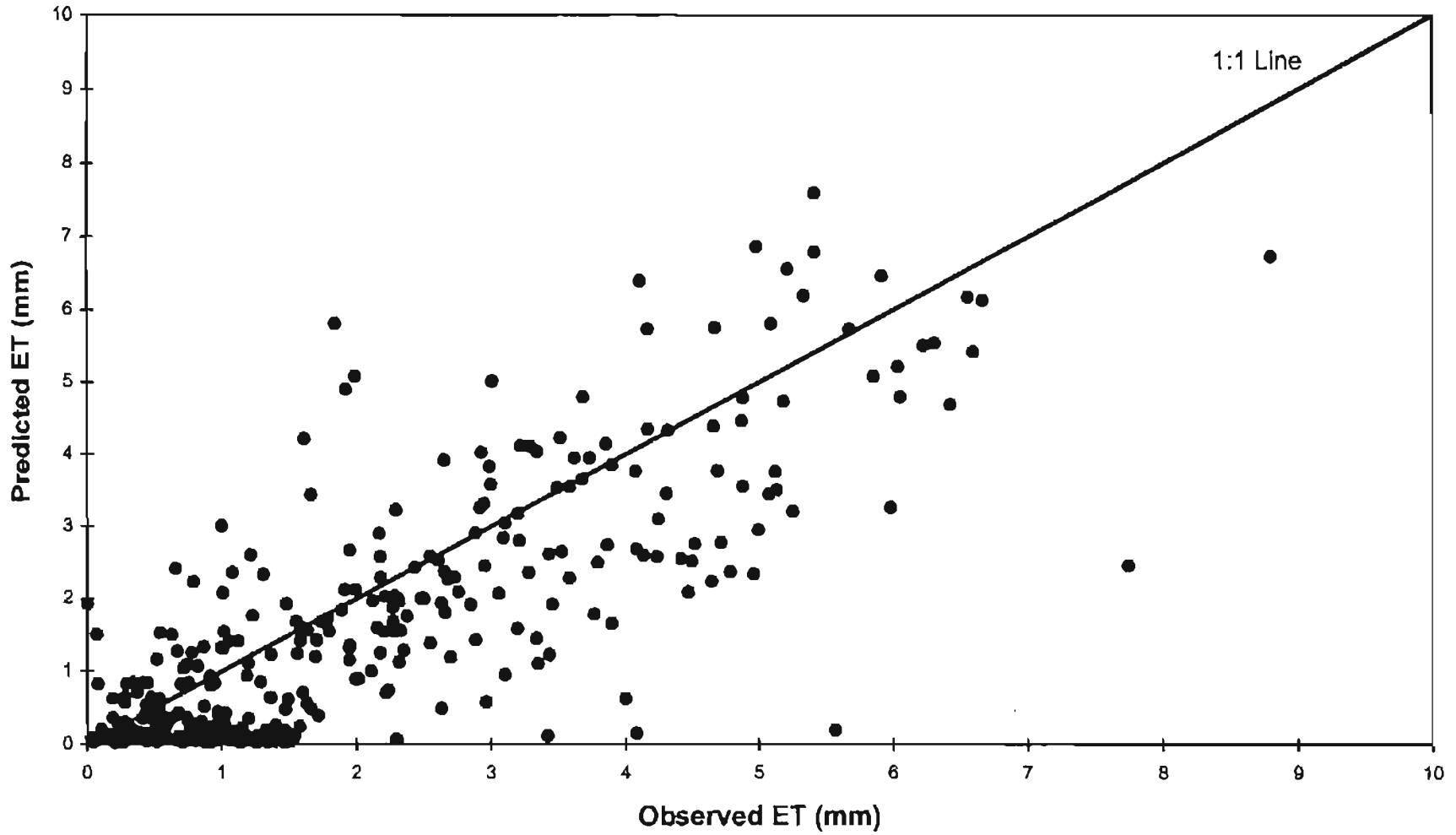


Figure 4.8. Comparison of predicted and observed ET for Jan. 1996 through Sep. 1997 at Goodwell

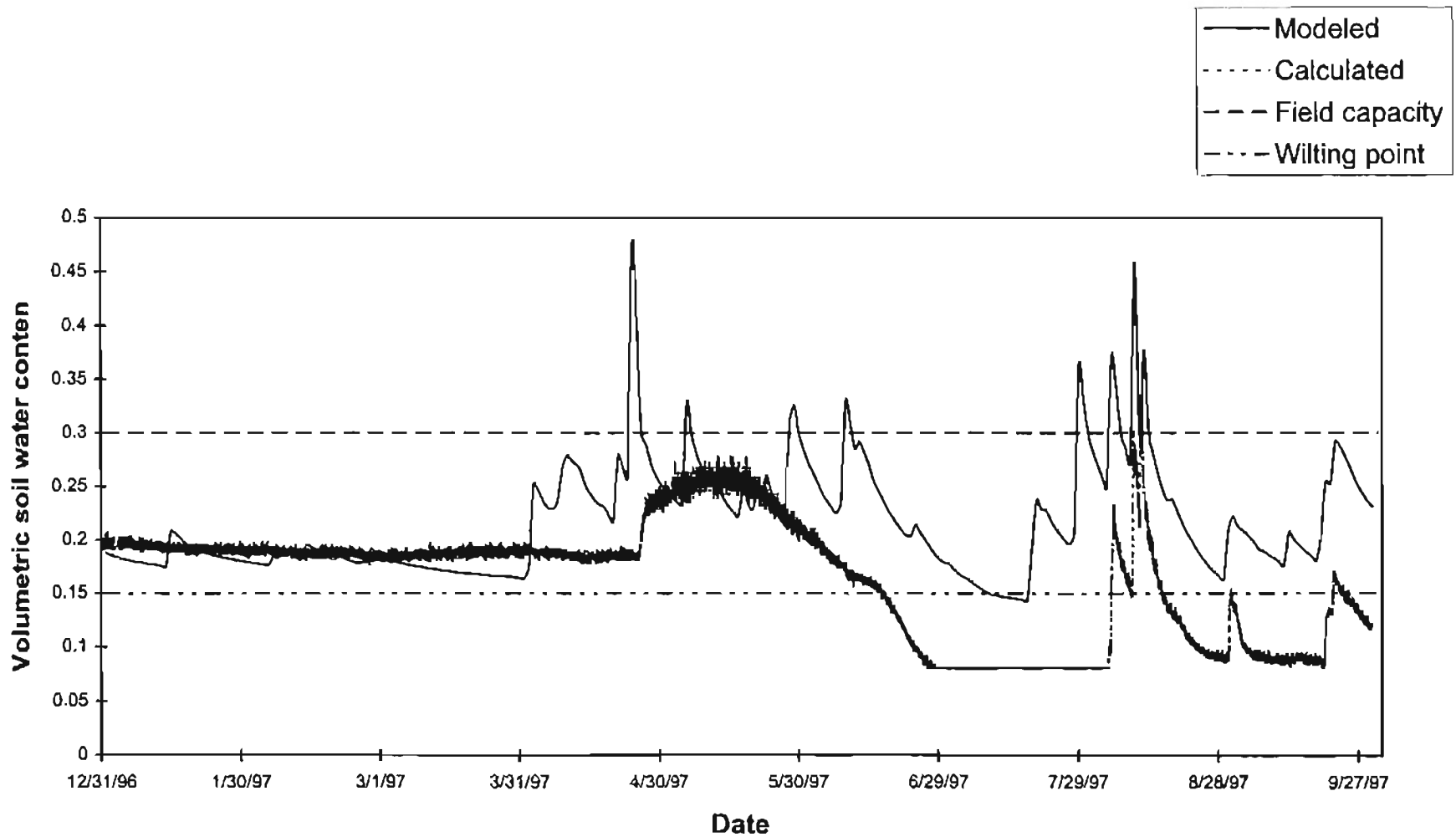


Figure 4.9. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Goodwell

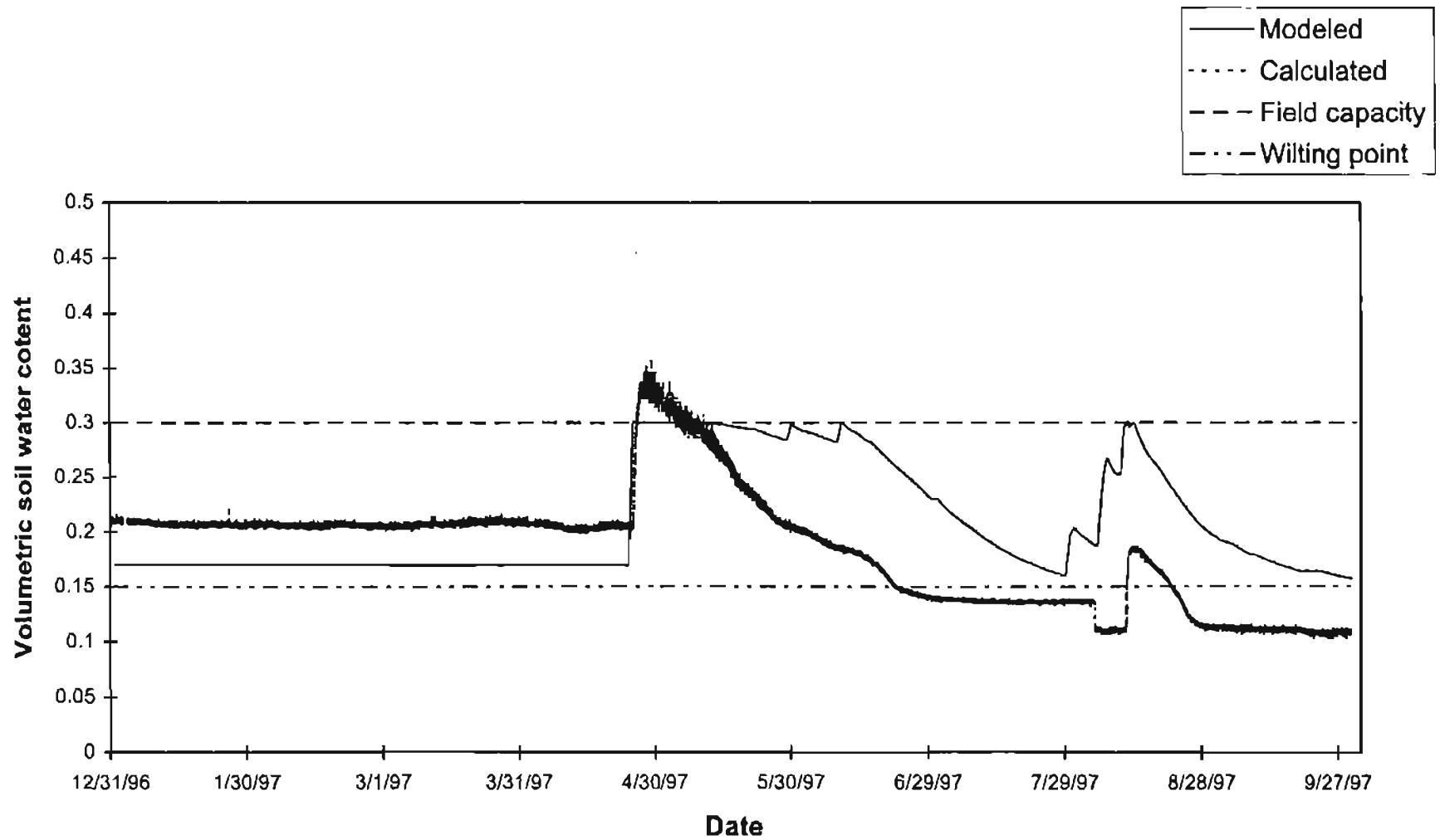


Figure 4.10. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Goodwell

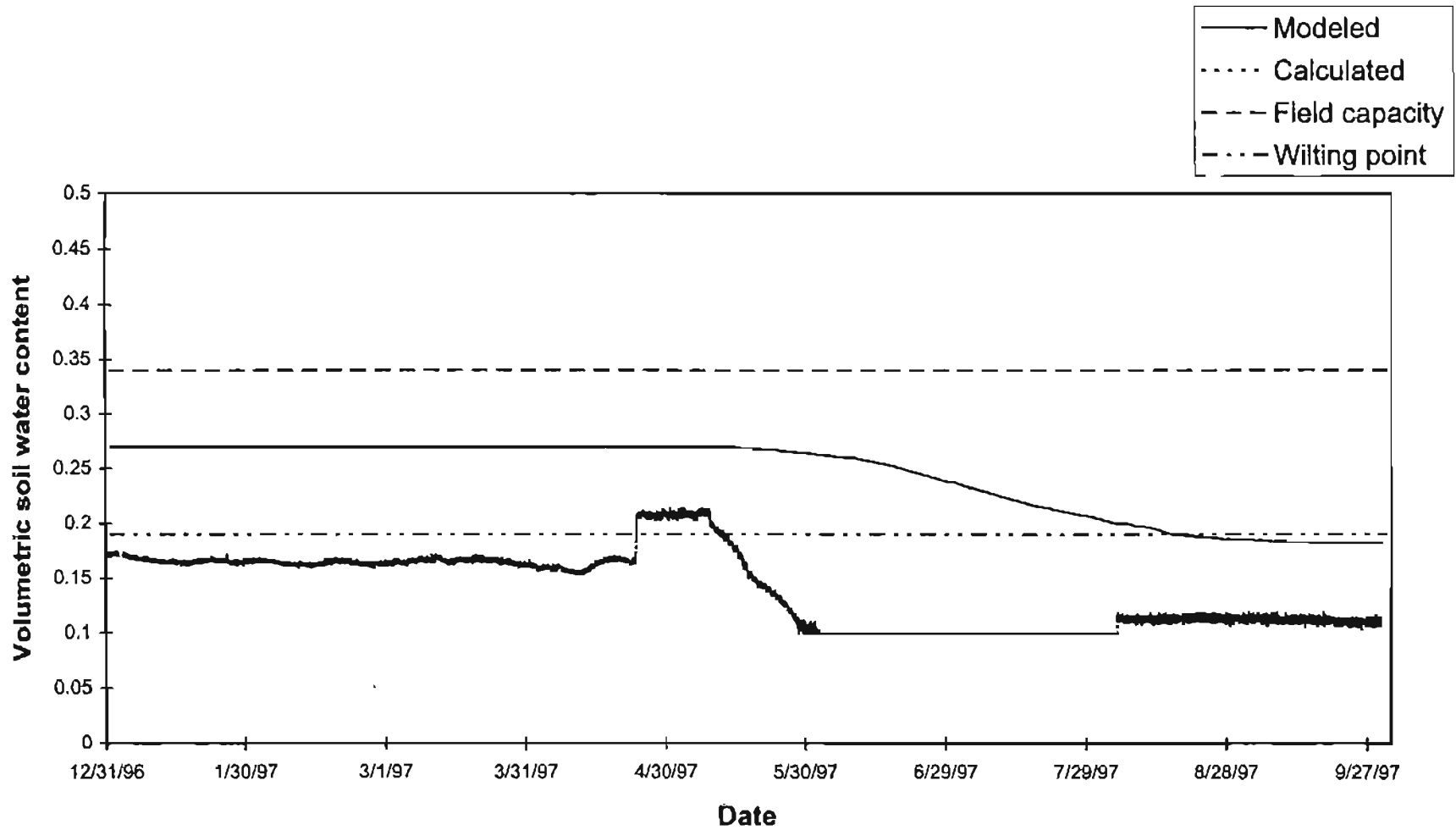


Figure 4.11. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Goodwell

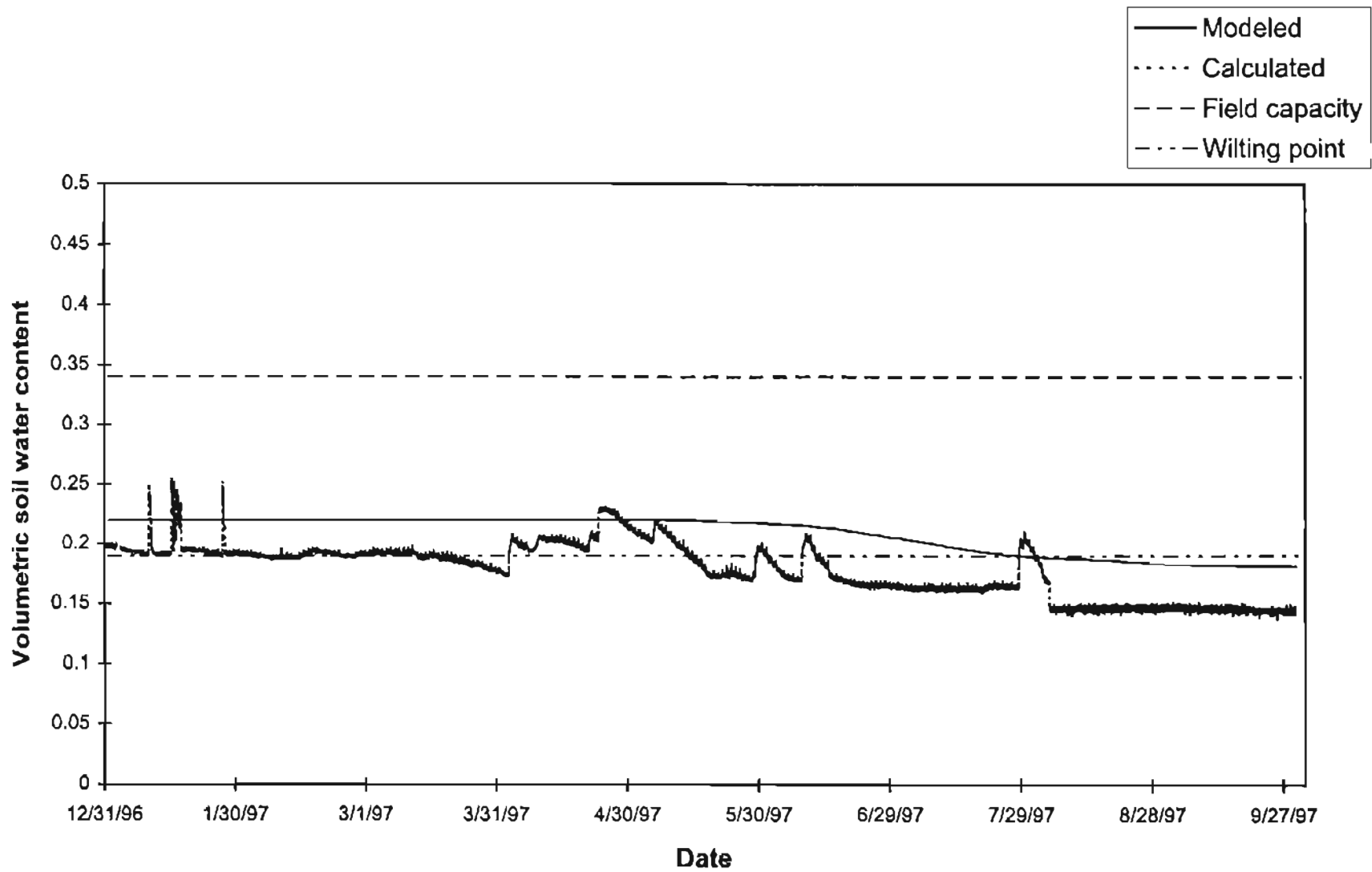


Figure 4.12. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Goodwell

by the similar fluctuations in the upper layers, it seems unlikely that it could be due to infiltration.

Marena

At Marena, the model was successful in simulating the trend in the ET (figure 4.13), though the absolute values did not match that well. The one-to-one plot (figure 4.14) showed that the model tended to overestimate the ET. The average predicted ET, 2.57 mm, was higher than the average observed ET, 1.95 mm (table 4.10). Though the coefficient for the regression slope is less than one, 0.76, the intercept is quite high, 1.08 (table 4.10). In the last week of August 1996, the model predicted a drop in the ET due to soil water stress (figure 4.13). ET increased only after the soil received water on the 15th and the 16th of September. But during the same period, many of the observed ET values were considerably higher (in the range of 4.5-8.8 mm). In 1997, model predictions were in good agreement with the observations during the early growing period. In July 1997, the model predictions were considerably higher than the observations. The overall average absolute deviation, 1.34 mm, is similar to that at Apache.

Since 229L data collection had started by the end of July, both 1996 and 1997 observations were used for the comparison and analysis. The predictions for soil water compared very well with the measured data for both 1996 and 1997 (figures 4.15 through 4.18). The model predictions follow more or less the observed pattern. The major discrepancy seemed to be on June 29, 1997. The observations indicate that the rain on

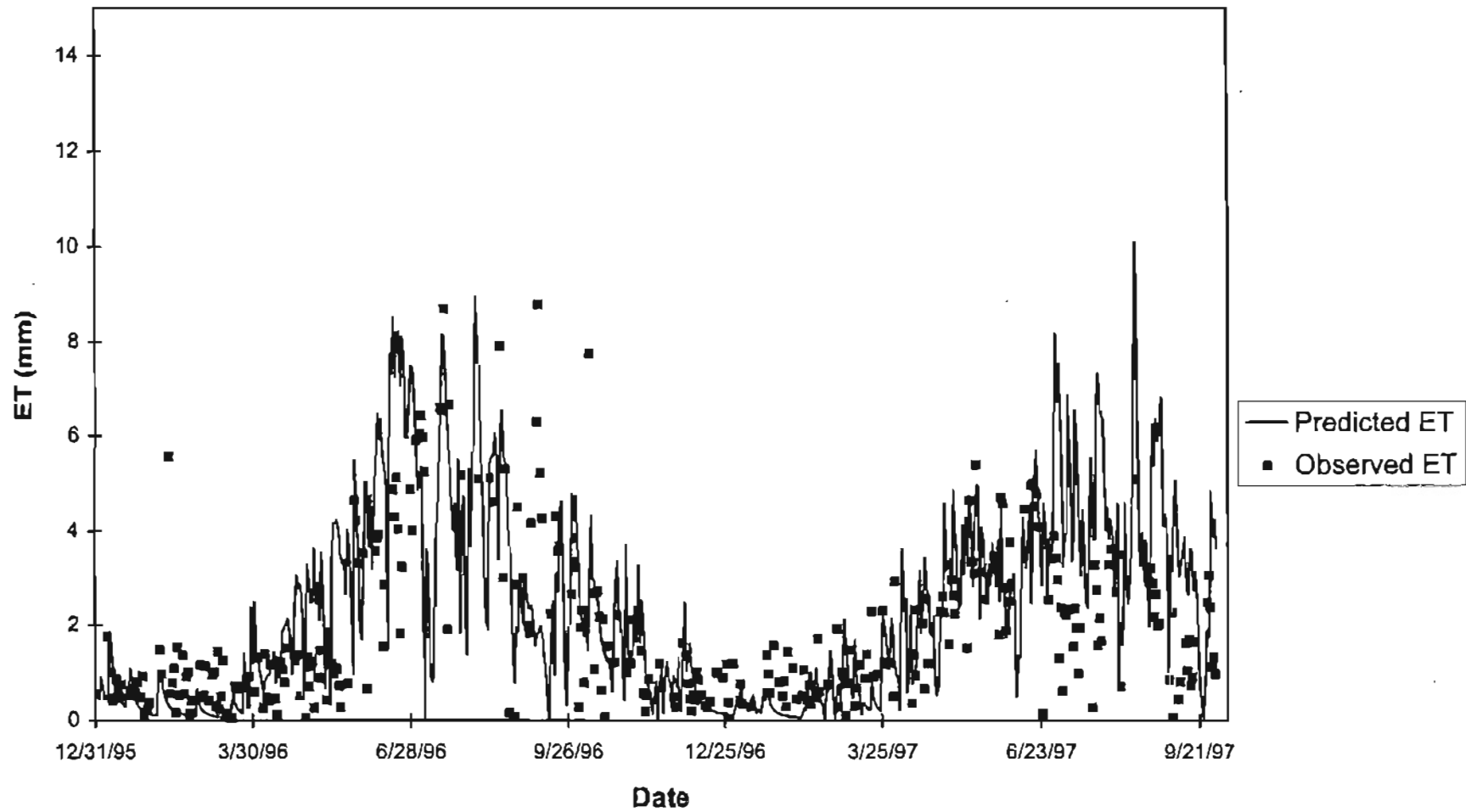


Figure 4.13. ET for Jan. 1996 through Sep. 1997 at Marena

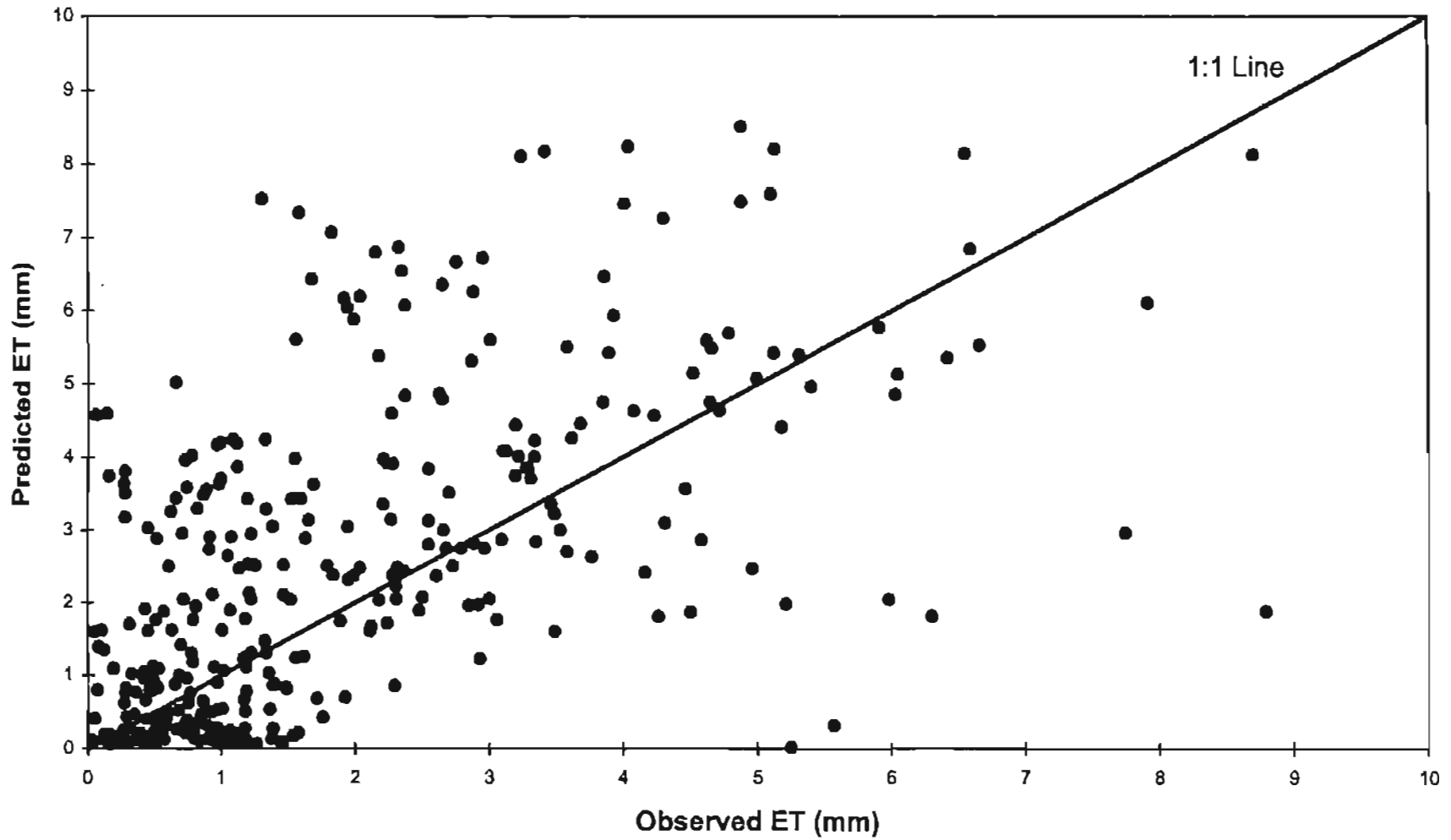


Figure 4.14. Comparison of predicted and observed ET for Jan. 1996 through Sep. 1997 at Marena

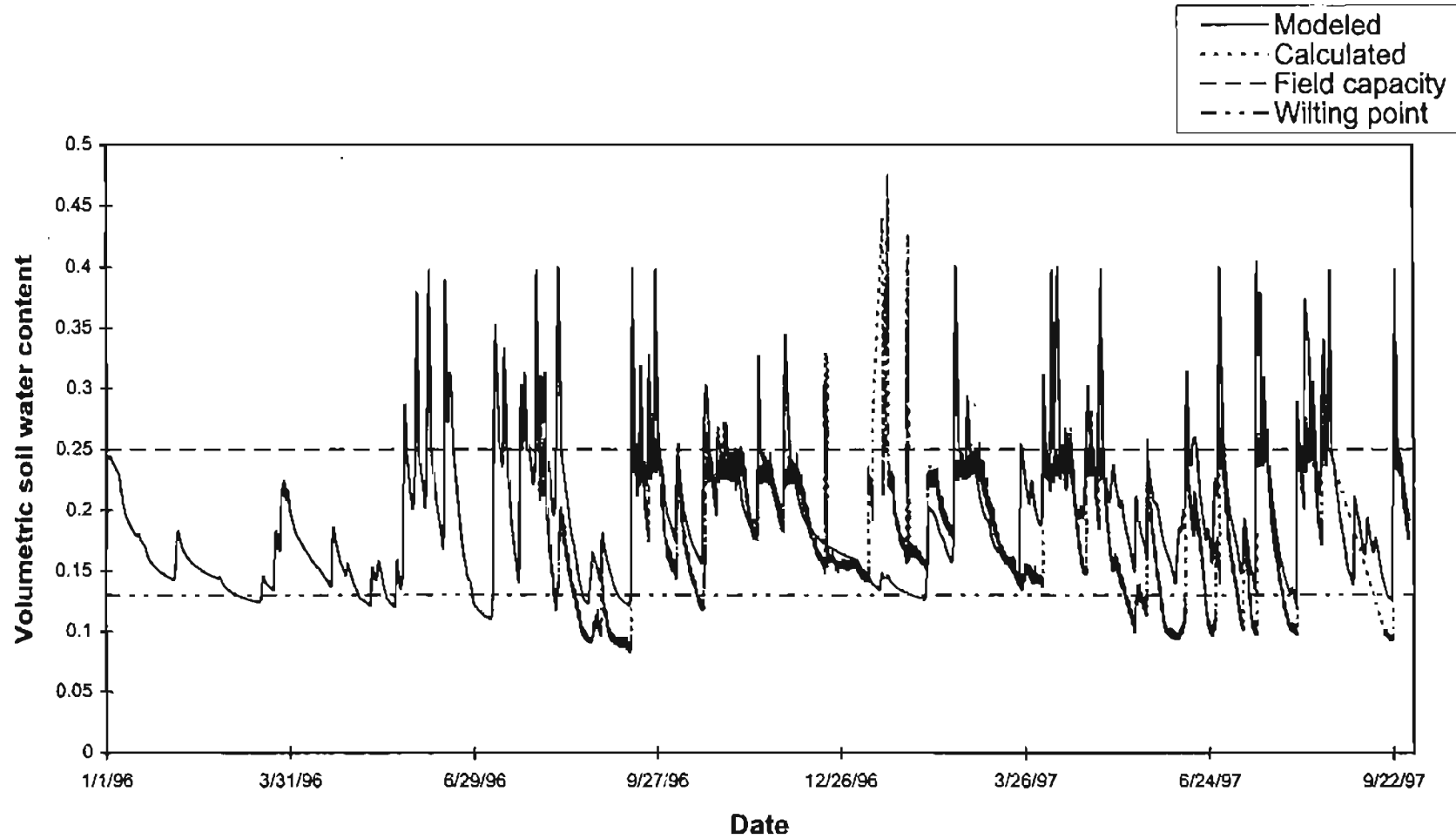


Figure 4.15. Soil water content at 5 cm for Jan. 1996 through Sep. 1997 at Marena

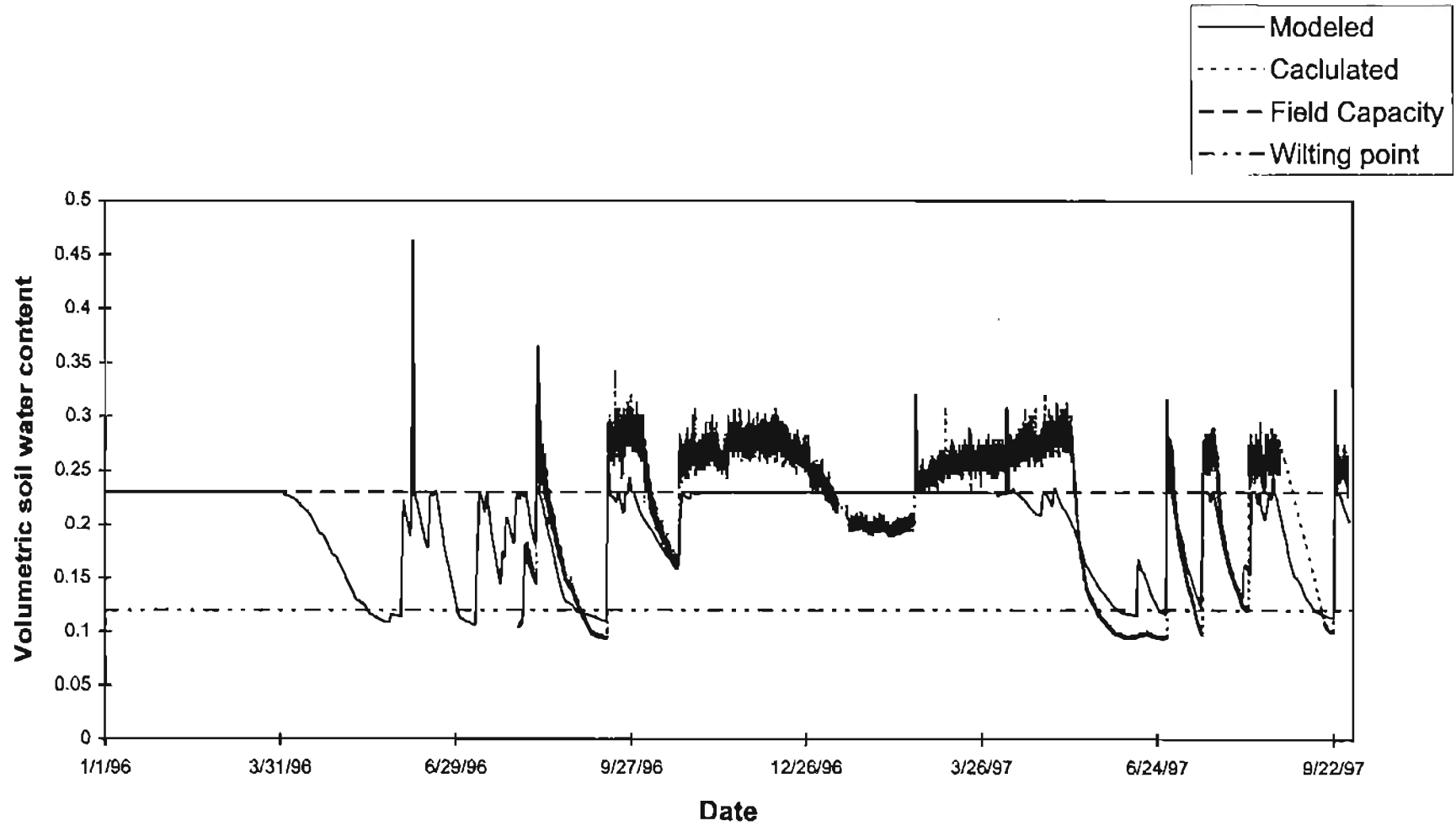


Figure 4.16. Soil water content at 25 cm for Jan. 1996 through Sep. 1997 at Marena

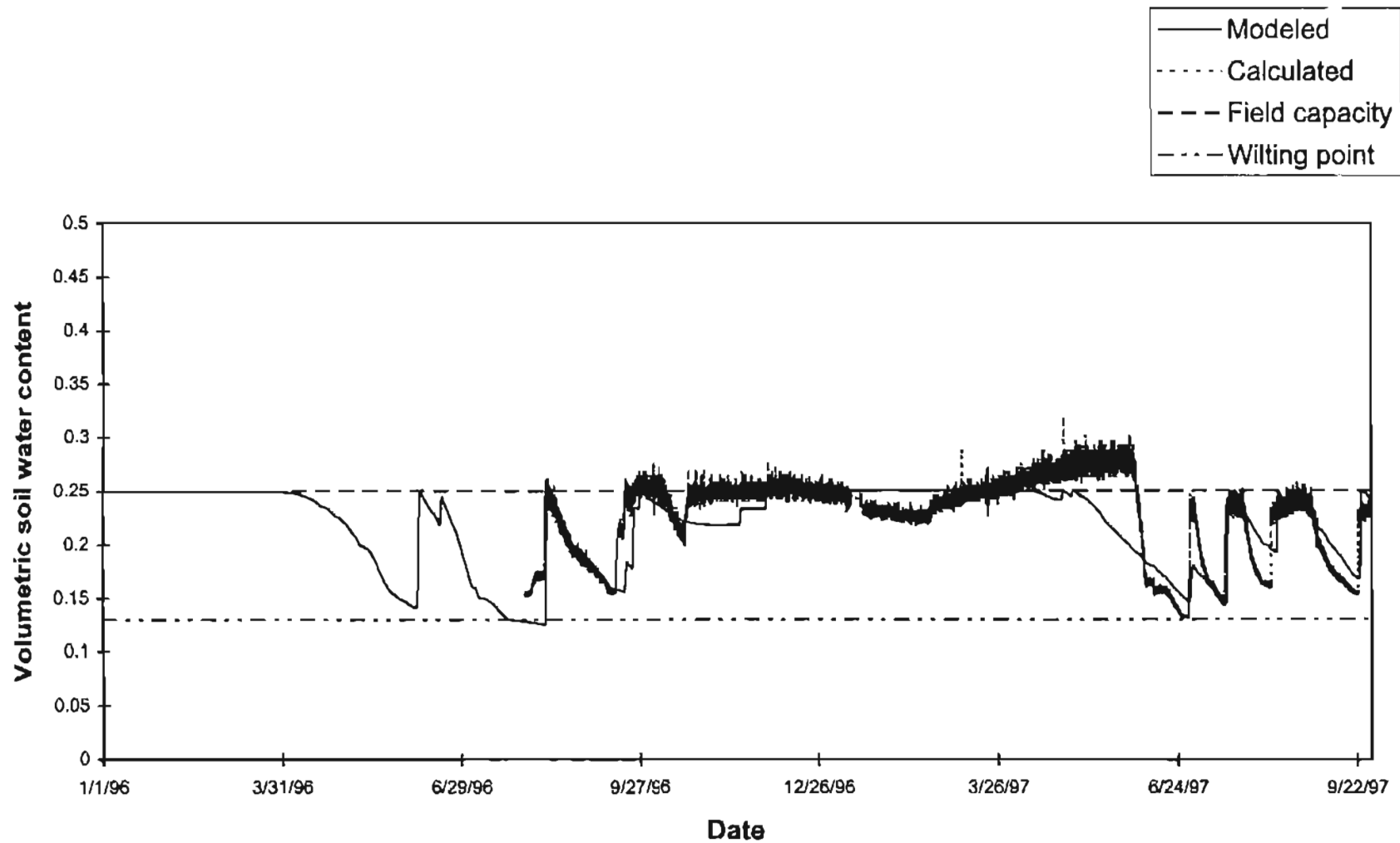


Figure 4.17. Soil water content at 60 cm for Jan. 1996 through Sep. 1997 at Marena

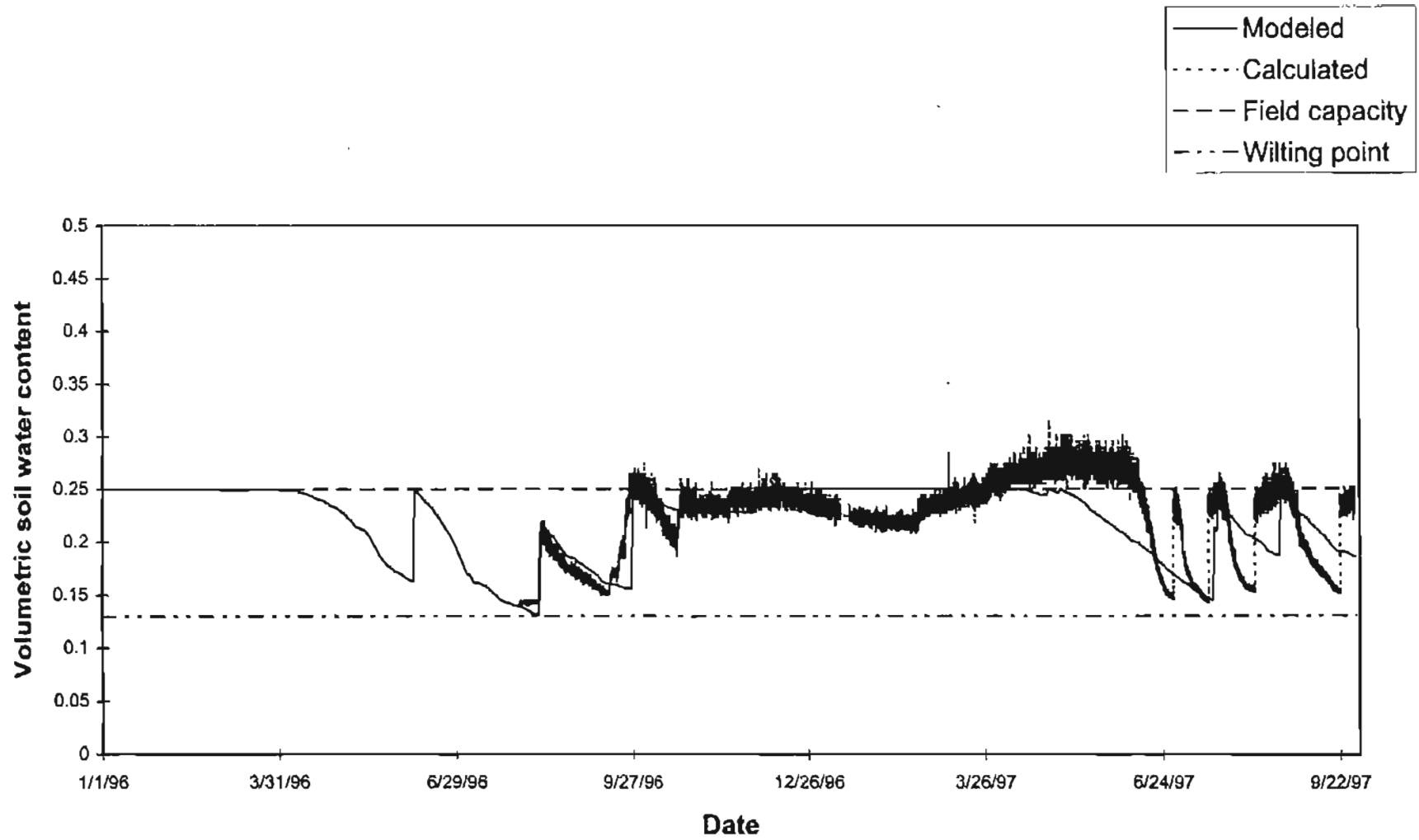


Figure 4.18. Soil water content at 75 cm for Jan. 1996 through Sep. 1997 at Marena

that day had increased the soil water content down to a depth of 75 cm, at least, but in the model the soil water content increased only down to a depth of 60 cm.

Stillwater

For Stillwater, only soil water observations were available. Both 1996 and 1997 soil water observations were included since data collection at this site had started by the end of July and thus enough observations were available in 1996. The time series trends of the predicted soil water were in good agreement with the observed soil water at all four depths, except for some minor deviations (figures 4.19 - 4.22). But the quantity by which the soil water content decreases and increases differs considerably. The drops in the observed water contents were steeper than the drops in the predicted. 229L observations indicated a higher field capacity at 25 cm than the field capacity being used in the model. Adjusting the field capacity at this site did not improve the model predictions appreciably.

Summary Calibration

During the calibration phase, attempts were made to match the ET observations and the pattern of the soil-water variation at four different depths in the root zone. Model predicted ET matched well with the observed ET at Goodwell. At other sites, the agreement was not as close. The model did not simulate the high ET very well, especially when LAI was at its peak. The predicted soil water profile matched with the observed profile fairly well, but some discrepancies were observed. At all sites, the observations showed soil at 5 cm drier than the prediction (figures 4.3, 4.9, 4.15 and 4.19), especially

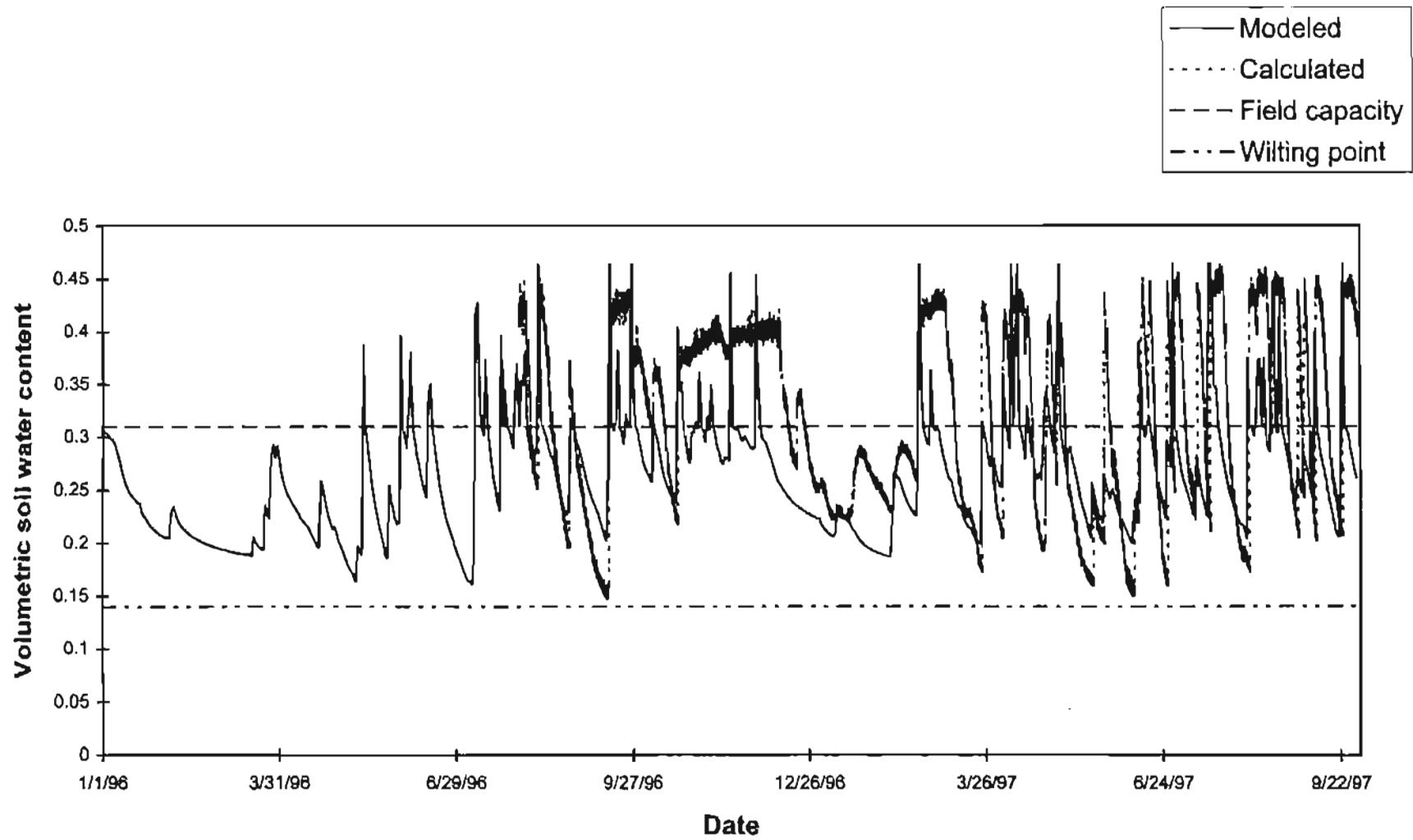


Figure 4.19. Soil water content at 5 cm for Jan. 1996 through Sep. 1997 at Stillwater

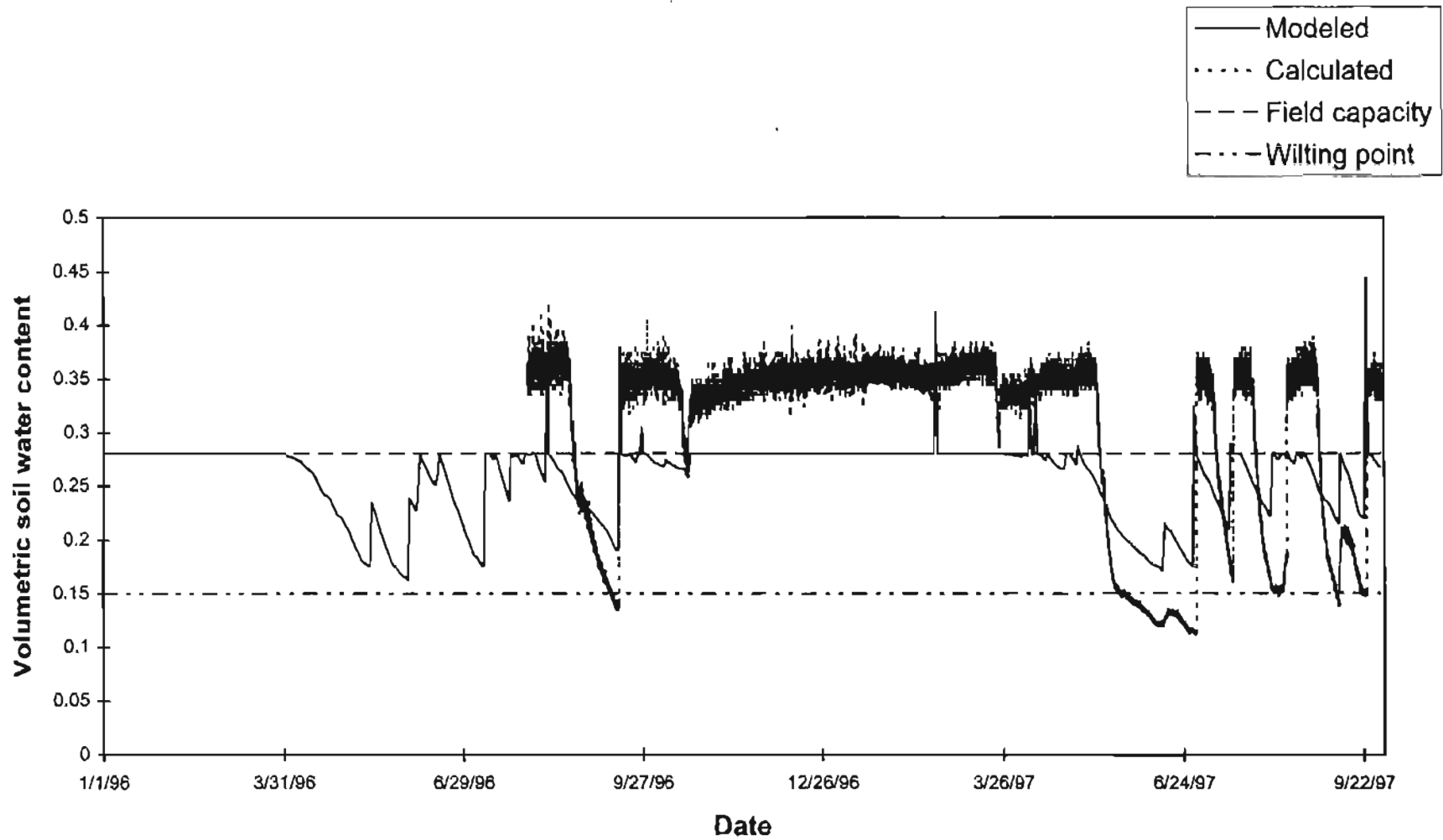


Figure 4.20. Soil water content at 25 cm for Jan. 1996 through Sep. 1997 at Stillwater

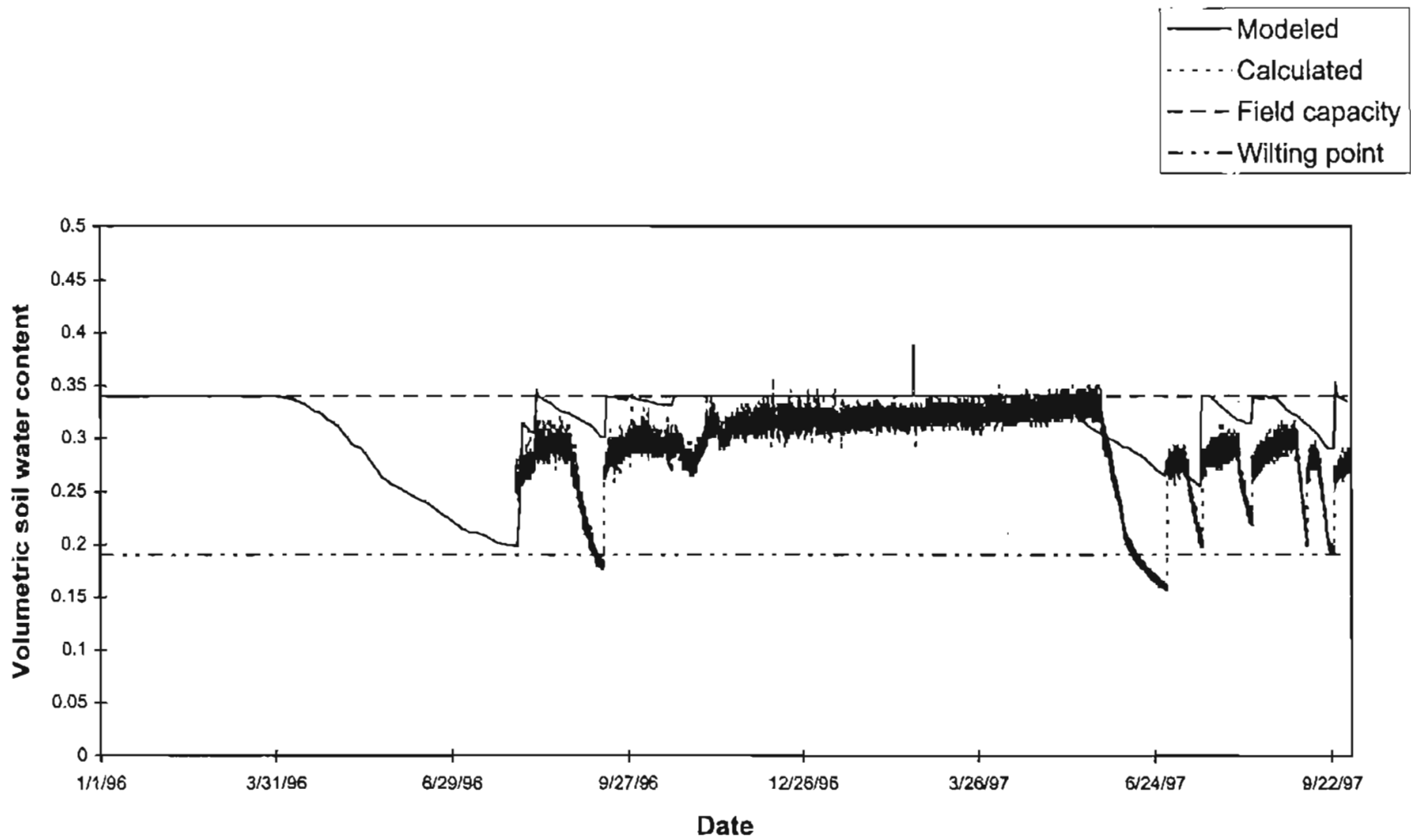


Figure 4.21. Soil water content at 60 cm for Jan. 1996 through Sep. 1997 at Stillwater

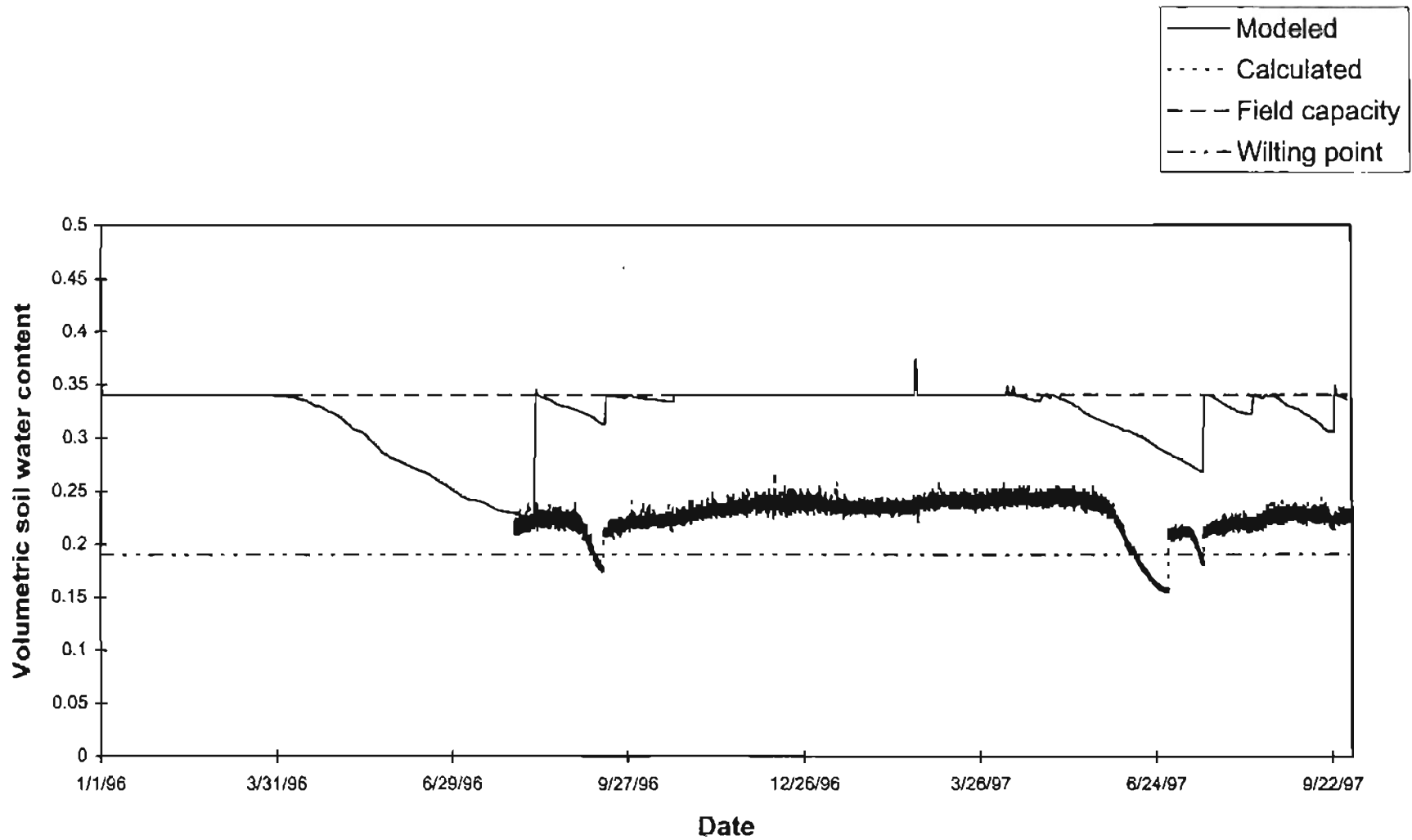


Figure 4.22. Soil water content at 75 cm for Jan. 1996 through Sep. 1997 at Stillwater

during the dry periods. The predictions are for a 15 cm thick layer, whereas the observations are point observations at a depth of 5 cm. It is very likely that the upper part of the upper soil layer gets drier than the soil beneath it. At all sites, the predictions were not in particularly good agreement with the observations at lower depths.

In evaluating model performance, it should be noted that the model is using literature-based estimates for the soil water parameters. This parameter uncertainty leads to uncertainty in model output. The estimates for the field capacity and the wilting point are especially important as they determine the upper and the lower limits of the soil-water in the root zone. Also, it should be noted that the observed soil water data are actually rough estimates derived from the 229L measurements, so a comparison of trends is more valid than a comparison of magnitudes.

Chapter 5

MODEL VALIDATION AND SENSITIVITY ANALYSIS

After the initial testing and parameterization at the development sites, the model's performance was evaluated at validation sites. The validation sites are Boise City, Broken Bow, Hollis and Miami. These sites exhibit considerable variation in terms of vegetation and soil type (table 5.1), and also climate.

Table 5.1 Description of validation sites.

Site	Latitude (N)	Longitude (W)	Vegetation	Soil Texture of the Upper Layer	SCS Curve Number
Boise City	36.69	102.50	Short grass*	Clay loam	74
Broken Bow	34.01	94.61	Tall fescue	Loam	61
Hollis	34.69	99.83	Mixed grass	Silty clay	80
Miami	36.89	94.84	Tall fescue	Silt loam	61

* Predominant species are blue grama and hairy grama

At each site, the root zone was divided into various layers in the same way as done at the development sites, to accommodate the varying texture information and soil water observation points. Table 5.2 provides information on the soil layers as used in the model.

Table 5.2. Depths (mm) and texture of soil layers at validation sites.

	Boise City	Broken Bow	Hollis	Miami
Layer-1	0 – 150 (Clay loam)	0 – 150 (Sandy loam)	0 – 150 (Silty clay loam)	0 – 150 (Silt loam)
Layer-2	150 – 300 (Clay loam)	150 – 350 (Sandy loam)	150 – 200 (Silty clay loam)	150 – 350 (Silt loam)
Layer-3	300 – 450 (Silty clay loam)	350 – 560 (Sandy loam)	200 – 300 (Silty clay)	350 – 500 (Silt loam)
Layer-4	450 – 550 (Silty clay loam)	560 – 650 (Clay)	300 – 550 (Silty clay)	500 – 550 (Clay)
Layer-5	550 – 650 (Silty clay loam)	650 – 730 (Clay)	550 – 650 (Silty clay)	550 – 650 (Clay)
Layer-6	650 – 700 (Silty clay loam)	730 – 760 (Sandy clay)	650 – 700 (Silty clay)	650 – 700 (Clay)
Layer-7	700 – 800 (Silty clay loam)	760 – 900 (Sandy clay)	700 – 800 (Silty clay)	700 – 800 (Clay)
Layer-8	800 – 1050 (Silty clay loam)	900 – 1100 (Sandy clay)	800 – 900 (Silty clay)	800 – 900 (Clay)
Layer-9	--	--	900 – 1100 (Silty clay)	900 – 1100 (Clay)

PARAMETERIZATION OF THE VALIDATION SITES

Meteorological variables were measured directly by the Mesonet weather station at these sites. Soil water parameters were estimated based on the texture information of the soil at that site (table 4.3). The data collection procedures for the weather and the soil parameters are described in chapter 4.

VEGETATION PARAMETERS

Vegetation at the validation sites was parameterized based on the calibration results from the development sites, pertinent literature and observation. The dominant vegetation types at Boise City, Broken Bow, Hollis and Miami are short grass, tall fescue, mixed native (bluestem) and tall fescue, respectively. The predominant species at Boise City are blue grama and hairy grama. Boise City was parameterized based on the observations at Goodwell, for the vegetation at both sites is comparable and both are in the Oklahoma Panhandle. None of the development sites has tall fescue as its main vegetation. Sites with tall fescue were parameterized based on the literature and limited observations made during Mesonet site visits in the summer of 1997.

Leaf area index

The maximum LAI for tall grass species generally varies between 5 and 6 (Dickinson et al., 1986; Neilson, 1995; etc.). The maximum LAI for short grass species generally varies between 1 and 2 (Dickinson et al., 1986; Capehart and Carlson, 1994; Neilson, 1995; etc.). The observed maximum leaf area index at the development sites, except Goodwell, was between 5 and 6.4. At Goodwell where short grass is the main vegetation, the observed maximum LAI was 3. The maximum LAI at Broken Bow, Hollis and Miami was set equal to 6, as used by Dickinson et al. (1986) for tall grass species. The maximum LAI at Boise City was set equal to 3, based on the observations at Goodwell. The minimum LAI was set to 0 at all sites.

Canopy height

Tall grasses shoot normally 60 and 120 cm whereas short grasses shoot 15 and 60 cm (Weaver, 1920). Terrel (1979) describes tall fescue as 200 cm tall grass. A canopy height of 20 cm was used by Capehart and Carlson (1994) for grassland. In 1997, the observed canopy heights at lysimeter sites were 30 to 38 cm at Apache (last week of June), about 25 cm at Goodwell (first week of June) and about 40 cm at Marena (first week of July).

The model requires the average canopy height (representative of the site) at the peak period, not the height of the tallest plants comprising the canopy. Therefore the maximum canopy heights for the parameterization purpose were inferred from the observations at the development sites, observations at the validation sites and the photographs of these sites on the Internet. The minimum and the maximum canopy heights at Boise City were selected as 1 cm and 35 cm, respectively, based on the observations at Goodwell. The minimum and the maximum canopy heights at Hollis were taken as 30 cm and 80 cm, respectively, based on the observations at Marena in 1996. The minimum and the maximum canopy heights for tall fescue, at Broken Bow and Miami, were taken as 10 cm and 100 cm, respectively.

Root depth

Root depth was calibrated to 80 cm at Goodwell and 100 cm at the other development sites. Based on this calibration, the root depth was taken as 80 cm at Boise City and 100 cm at Hollis.

Roots of tall fescue usually grow less than some other grasses. The maximum root depth of tall fescue was reported as 122 cm whereas that of bermudagrass was 200 cm (Donald et al., 1991). Weaver (1920) reported a working depth of meadow fescue as 45 cm. Milthorpe and Moorby (1979) presented a root profile for tall fescue in which root densities declined from 63 cm³/cm³ in the topsoil to 3.6 cm³/cm³ at the 50-60 cm depth. The observed soil moisture profile at Broken Bow and Miami showed water extraction from a depth of 75 cm. Based on these pieces of information, the root depth of tall fescue, at Broken Bow and Miami, was taken as 80 cm.

Critical dates of growth

The observed growth pattern at all development sites, except Goodwell, was in good agreement with the NDVI pattern from satellite imagery. At Goodwell, the growth started very late. Therefore the critical dates of growth for Boise City were selected from the calibration results at Goodwell. For other sites, NDVI data were used. As explained in chapter 4, Senay and Elliott (1997) have characterized the temporal variability in vegetative activity for major landcover categories in Oklahoma using a combination of high spatial resolution landcover data and high temporal resolution NOAA AVHRR NDVI data. They estimated the critical dates of growth for these major landcover classes. Their estimates were used to characterize the vegetation at the validation sites, except for Boise City.

Table 5.3. Vegetation parameters at the validation sites.

Parameter	Boise City	Broken Bow	Hollis	Miami
Vegetation	Short grass	Tall fescue	Mixed native	Tall fescue
LAI _{min}	0	0	0	0
LAI _{max}	3	6	6	6
Min. canopy height (cm)	1	10	30	10
Max. canopy height (cm)	35	100	80	100
Root depth (cm)	80	80	100	80
Dates of Growth				
Start of growth (D _{sg})	May 09	March 26	March 26	March 26
At max. LAI (D _{mx})	Aug. 03	July 23	July 23	July 23
Start of senescence (D _{ss})	Sep. 09	Aug. 25	Aug. 25	Aug. 25
Start of dormancy (D _{sd})	Nov. 11	Nov. 27	Nov. 27	Nov. 27

Other parameters

Values for other parameters were taken directly from the calibration results (tables 4.8 and 4.9). The minimum stomatal resistance is 166 s m^{-1} , the coefficient for the root extraction is 0.0026, the fraction of available soil water depletion at no stress is 0.5, the coefficient for shelter factor is 0.1 and the minimum and the maximum fractional transpiration factors are 0.5 and 0.9. Curve numbers were estimated based on the soil texture of the upper layer and the vegetation (table 4.1).

MODEL PERFORMANCE

The performance of the model was tested by running it at the validation sites. Simulation was performed for January 1997 through September 1997. First, the model was run using 1996 data and results from this simulation were used to initialize the soil water content in 1997. Only 229L observations are available at these sites. So soil water predictions from the model were compared against the soil water values estimated from the ΔT observations.

Boise City

The results are shown in figures 5.1 through 5.4. The rainfall data from 04/23/1997 to 05/29/1997 were missing for this site. Data from the Boise City cooperative observing station were used for this period. The model results matched the observations well at 5 cm and 25 cm (figures 5.1 and 5.2). At 60 cm and 75 cm, the simulated profile did not match well with the observed profile, especially in the case of high rainfall events (figures 5.3 and 5.4). This site received good rainfall during the second week of August (about 69 mm in 5 days). The observations showed an increase in the soil water content down to 75 cm. But the simulated profile showed no increase in the soil water content at 60 cm and below. This suggests that the modeled available water capacities were considerably different than those in the field. At 60 cm and 75 cm, the model predicted a continuous gradual decrease in the soil water, whereas observations showed fluctuations.

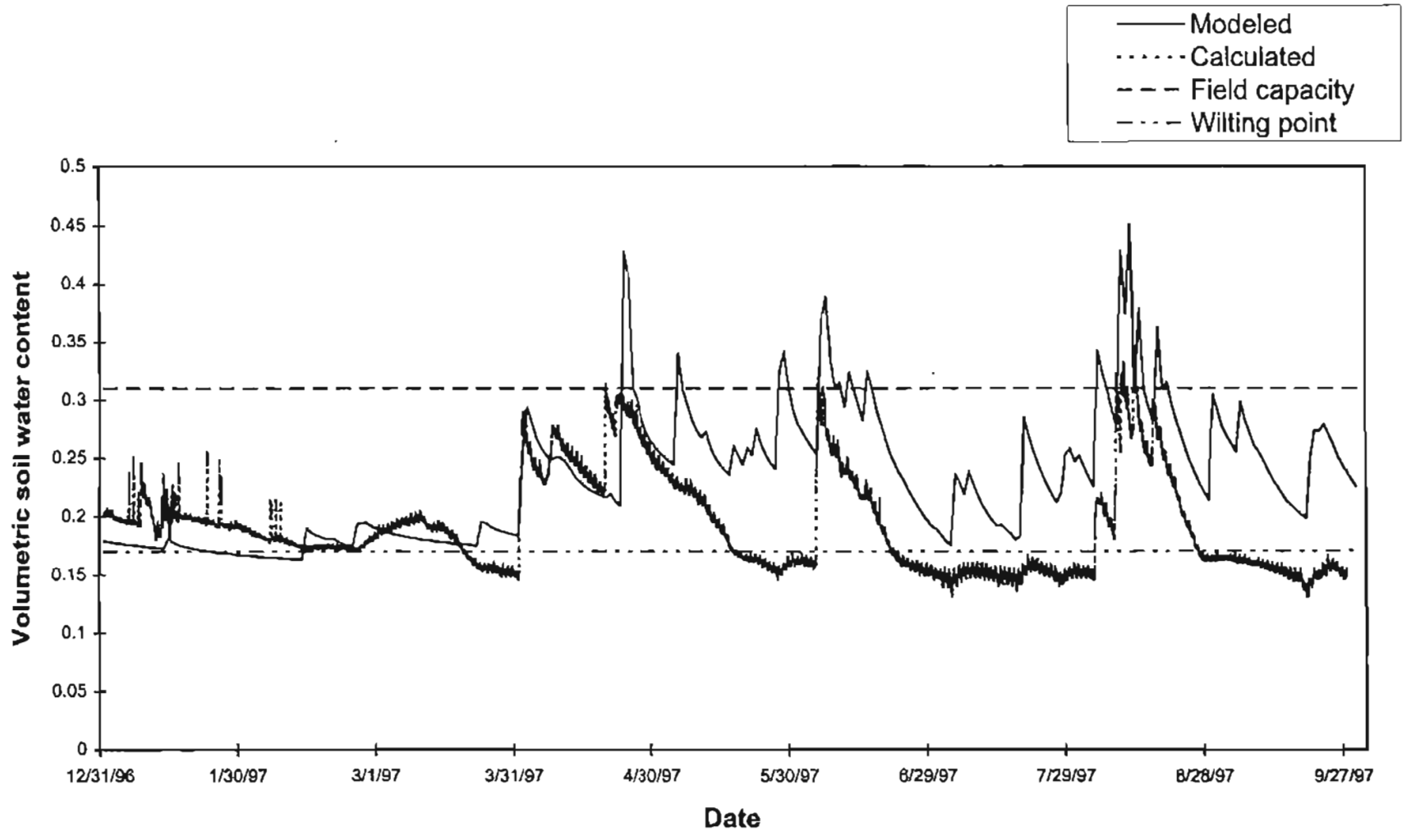


Figure 5.1. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Boise City

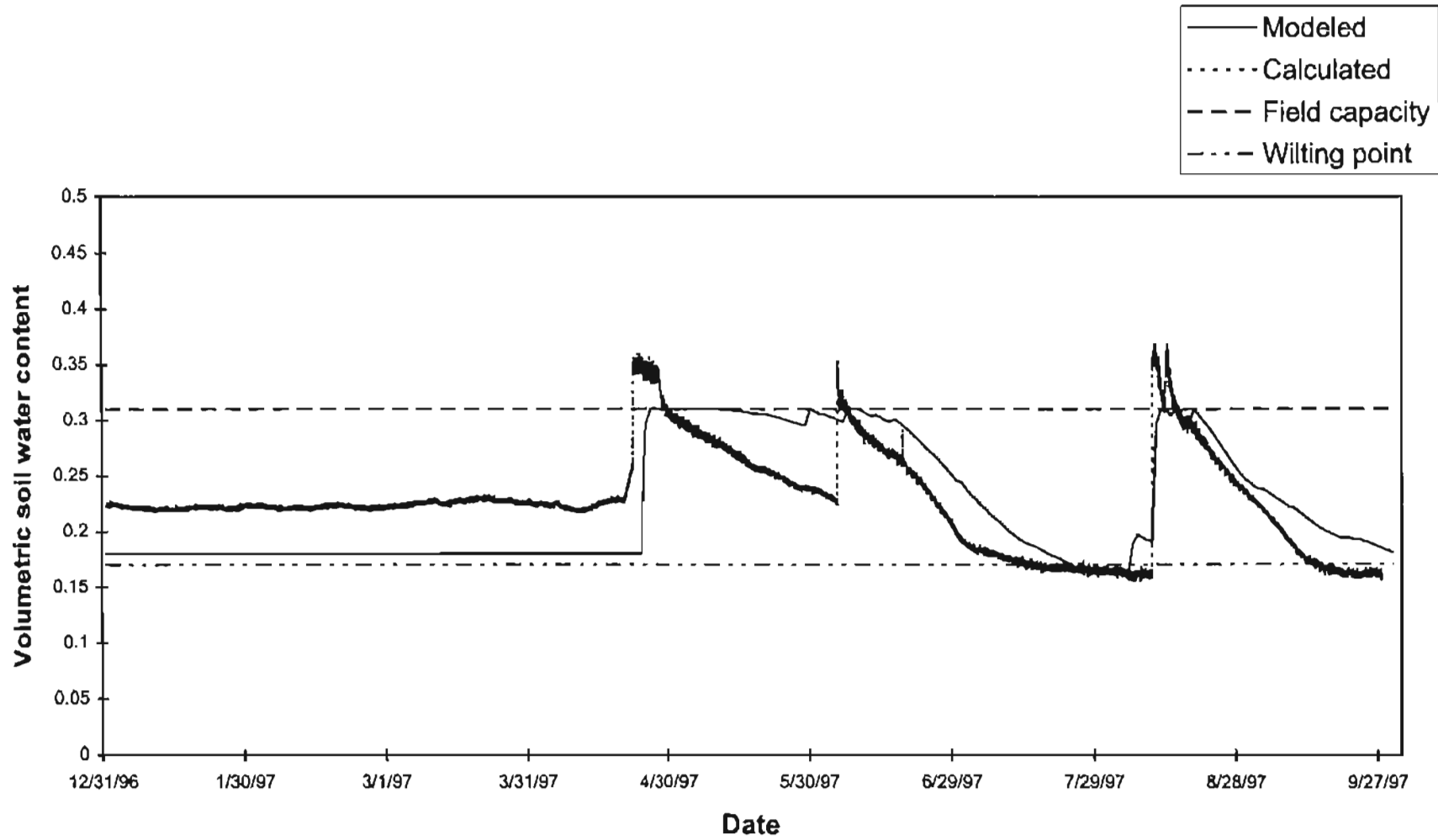


Figure 5.2. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Boise City

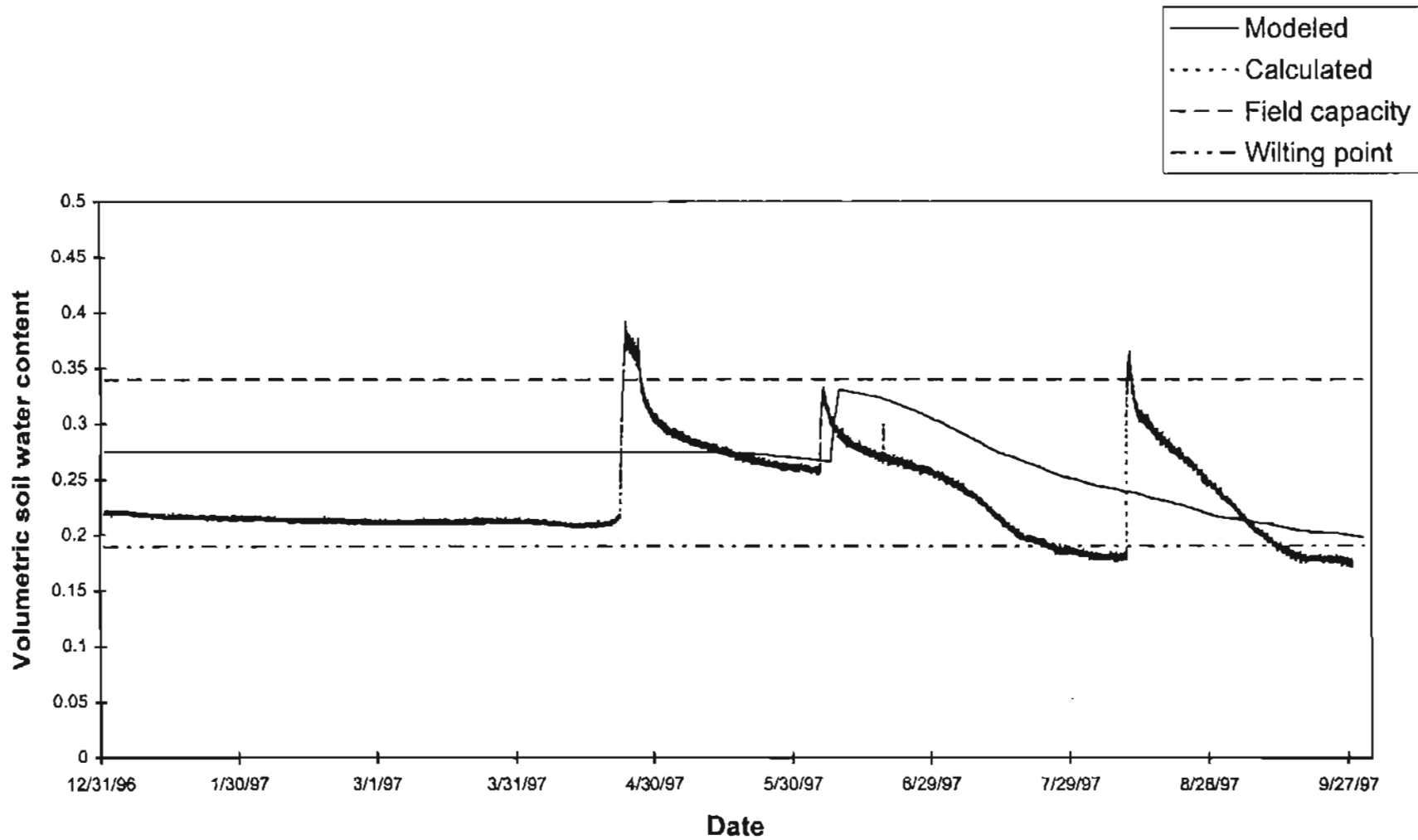


Figure 5.3. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Boise City

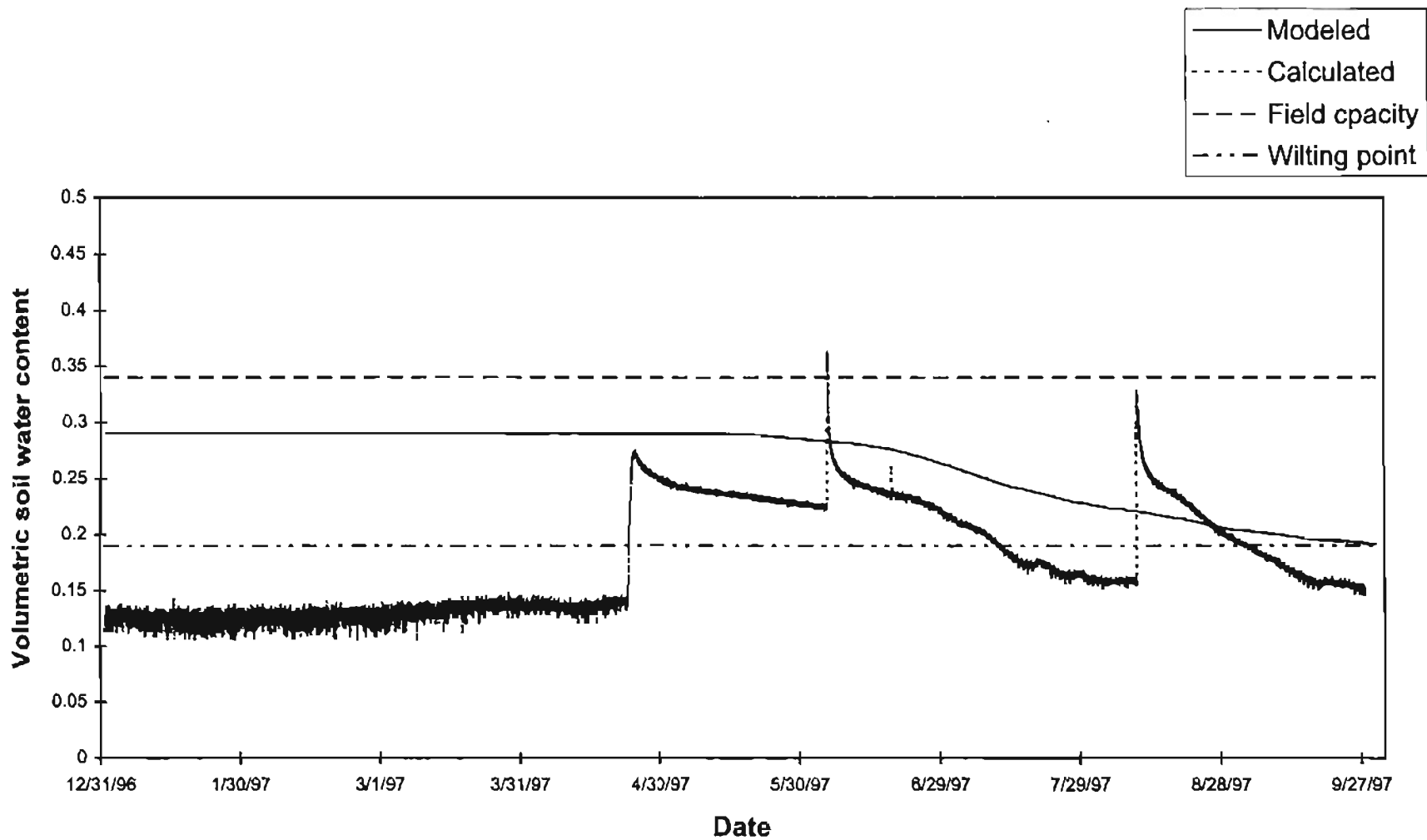


Figure 5.4. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Boise City

Broken Bow

The observations at Broken Bow indicated no major changes in the soil water regime in the first three months, approximately, at all depths (figures 5.5 through 5.8). The dormant vegetation might have formed mulch over the soil surface which prevented soil evaporation and maintained a high water content. Then the soil water content, at 5 cm and 25 cm, dropped rather suddenly (figures 5.5 and 5.6). At 60 cm and 75 cm the observed soil water content remained at its initial value until June 7 and then dropped rapidly (figures 5.7 and 5.8). The model did simulate the observed drops in the soil water regime but it could not match the steep slope of the observed drops. At the shallower depths (5 cm and 25 cm), it appears that the modeled range between field capacity and wilting point is considerably smaller than the range in the calculated water content.

Hollis

The simulation results at Hollis matched very well with the observed pattern at the first two observation points, 5 cm and 25 cm (figures 5.9 and 5.10). Almost all peaks and drops in the soil water regime were captured by the simulation results. At 60 cm, the model predicted an almost continuous gradual drop in the soil water content during the growing period. But the observations indicated a sudden rise in the soil water content three times during this period (figure 5.11). At 75 cm, the observations twice indicated a sudden rise in the soil water content, which is not matched by the simulation results (figure 5.12). Although rainfall was received during both of these periods, the increase in the observed soil water content appears to be inconsistently high. Especially during the

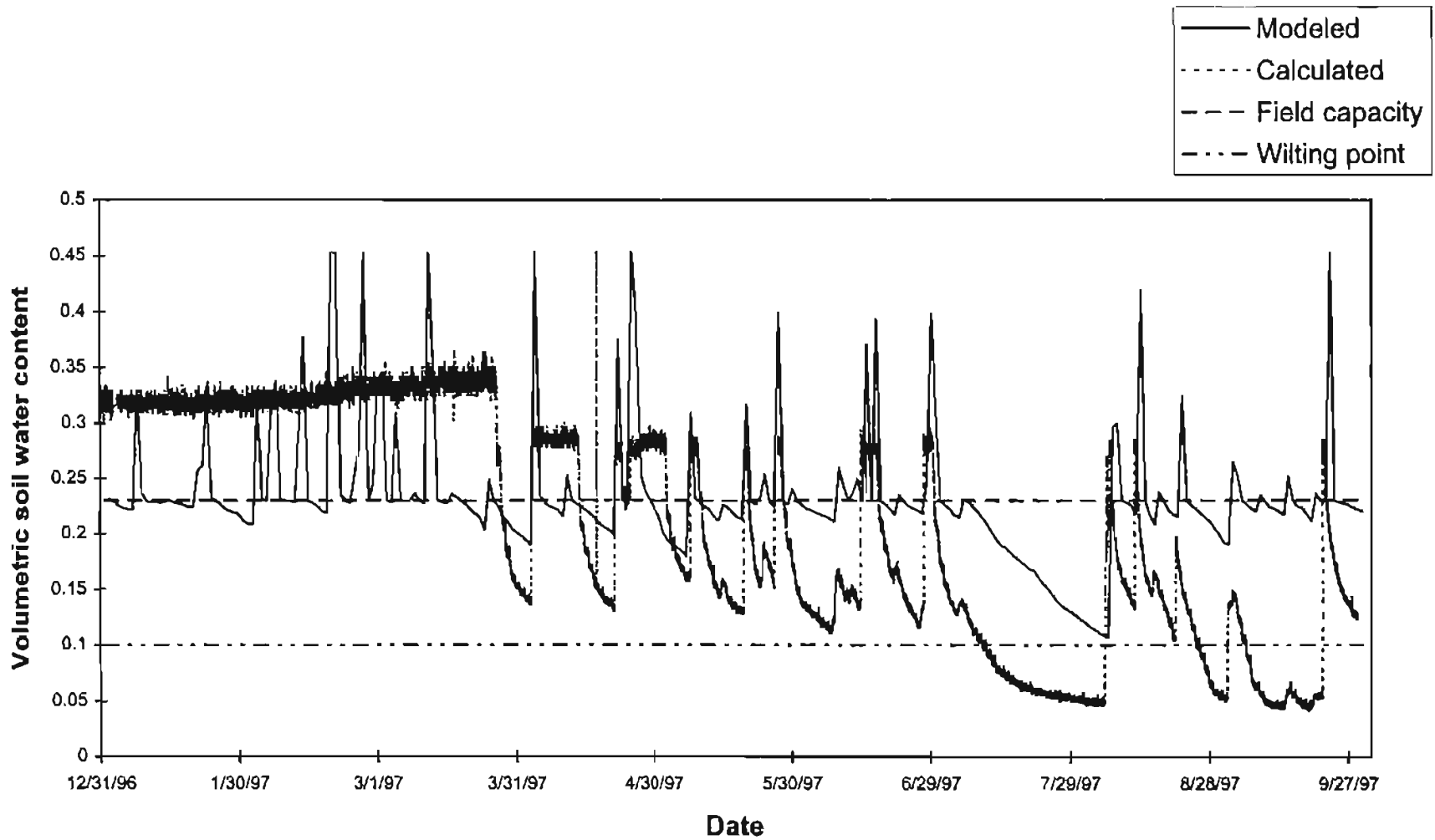


Figure 5.5. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Broken Bow

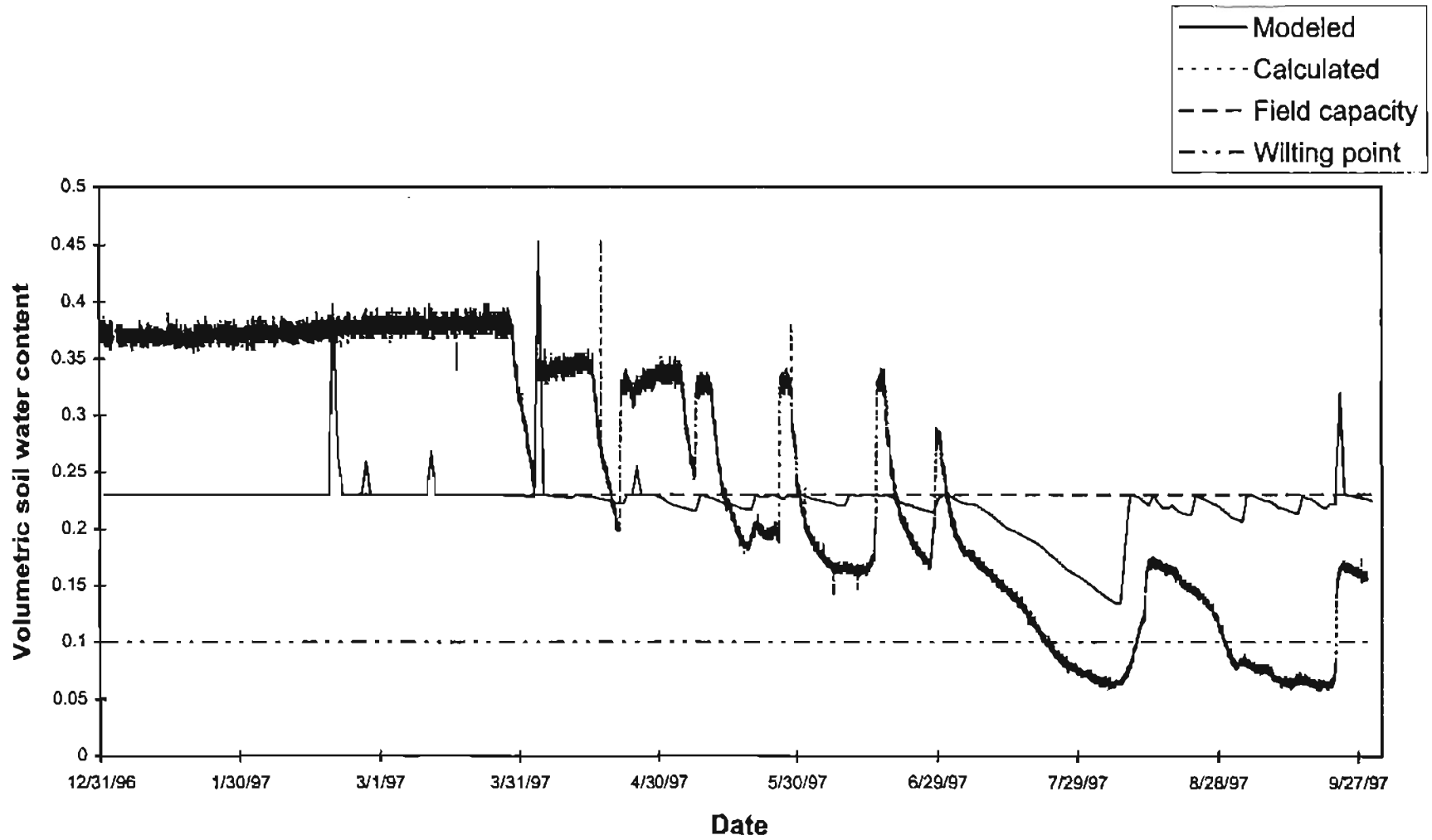


Figure 5.6. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Broken Bow

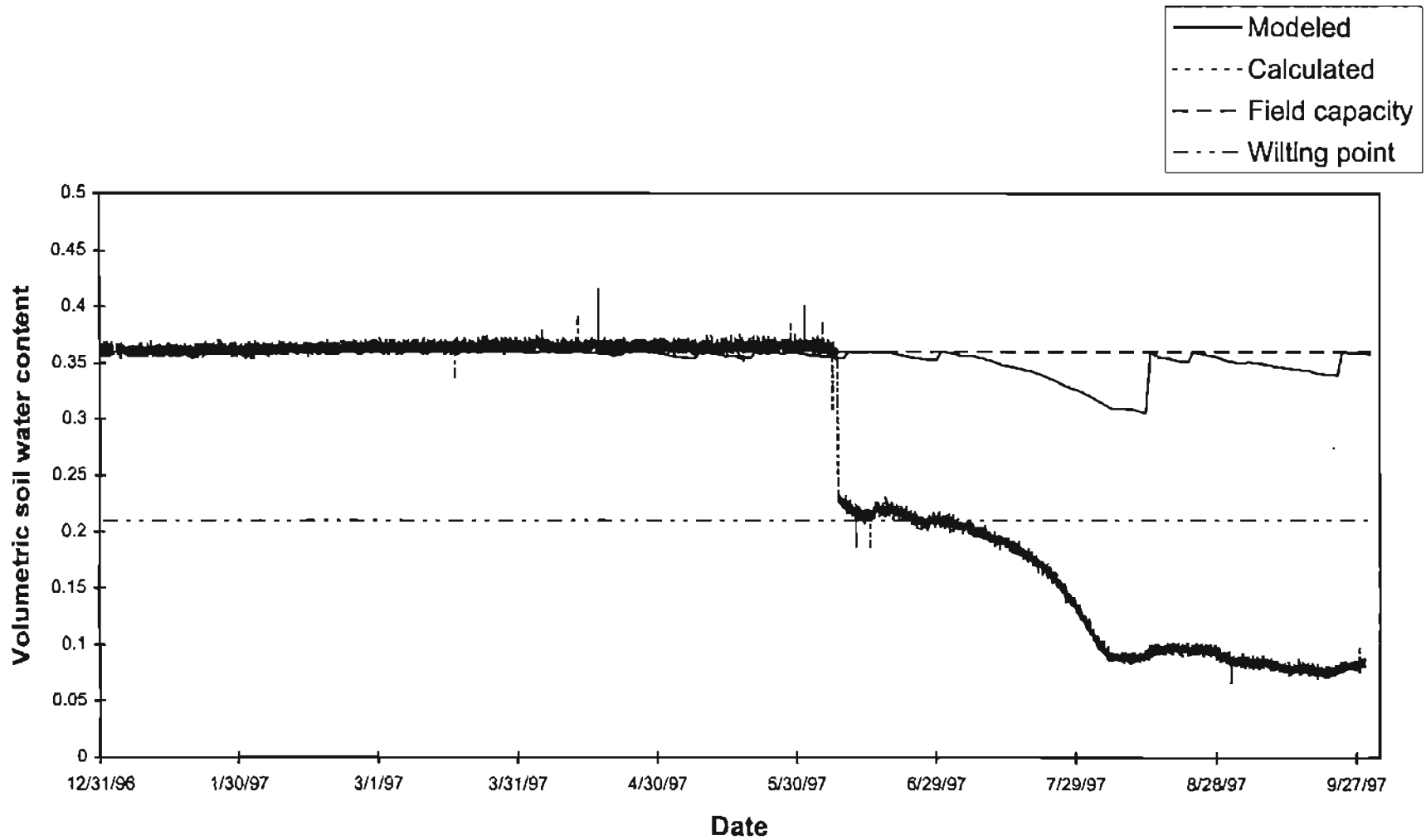


Figure 5.7. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Broken Bow

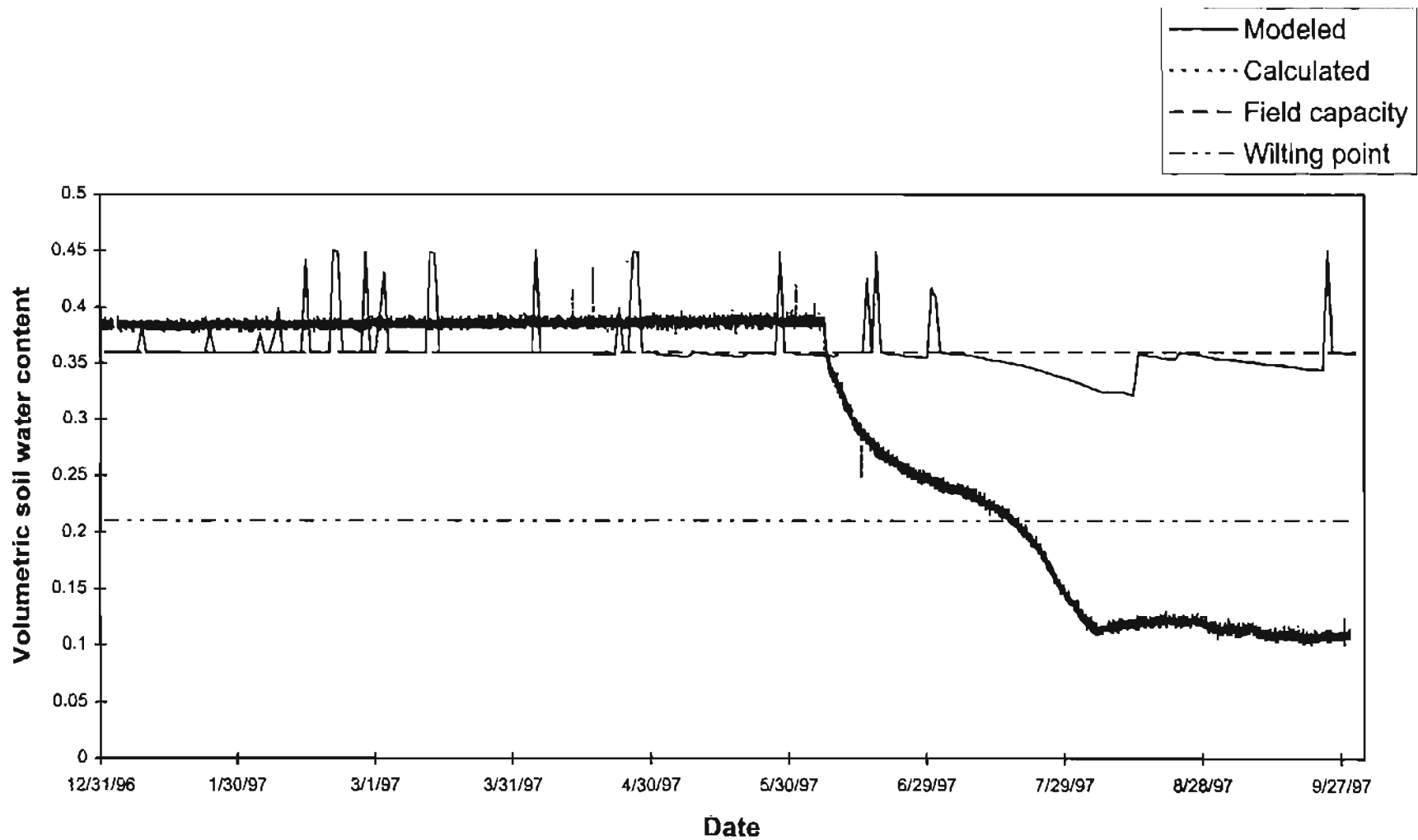


Figure 5.8. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Broken Bow

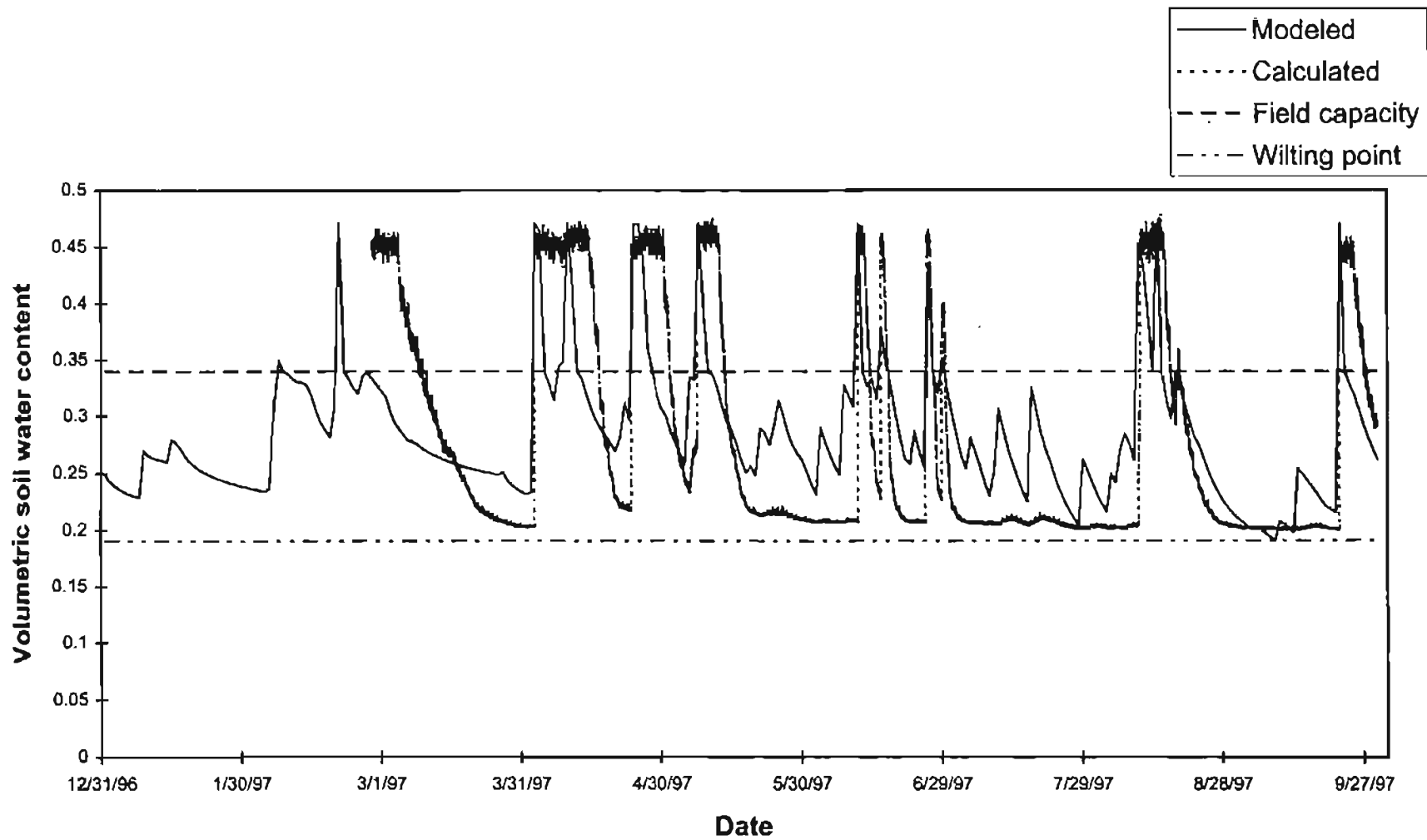


Figure 5.9. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Hollis

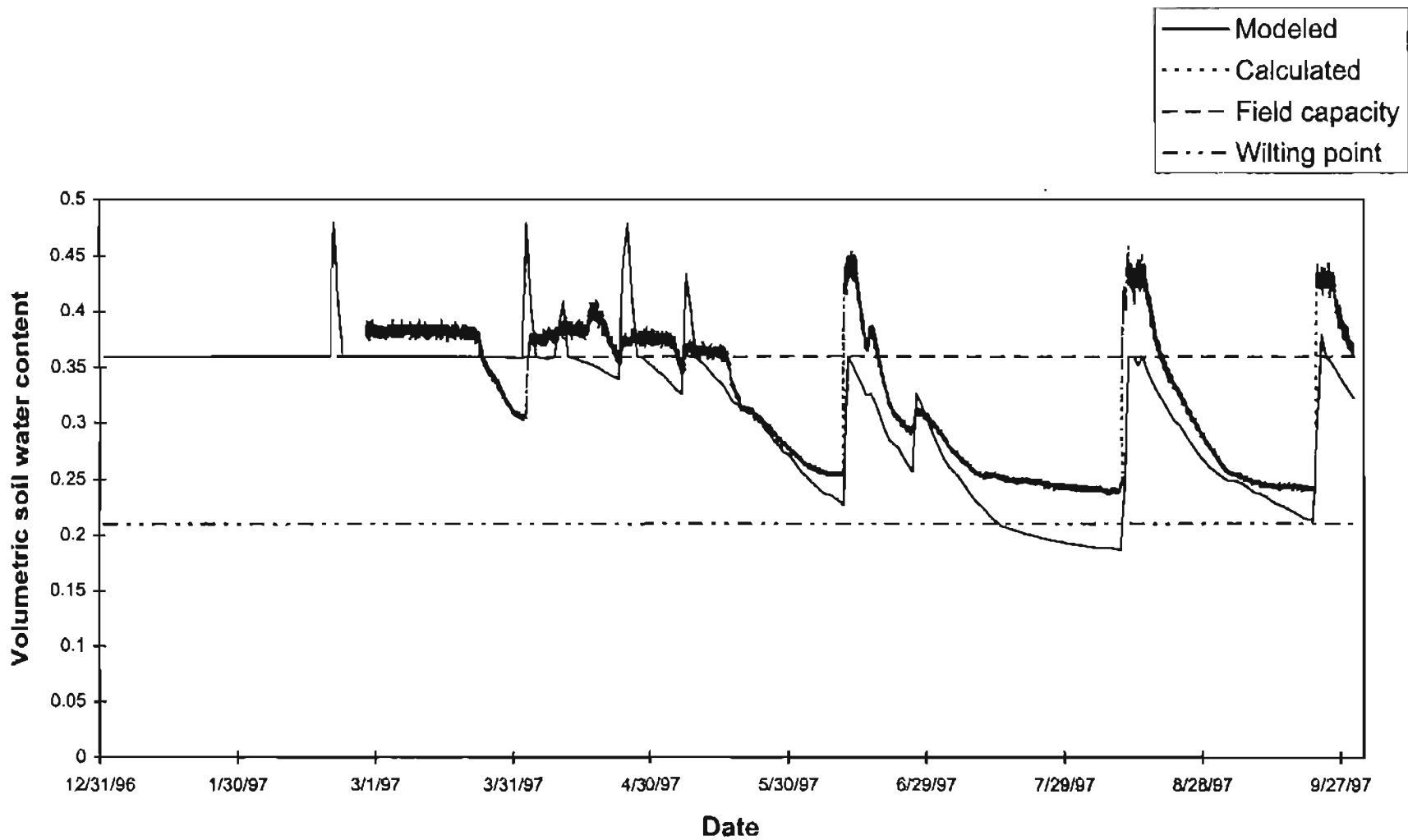


Figure 5.10. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Hollis

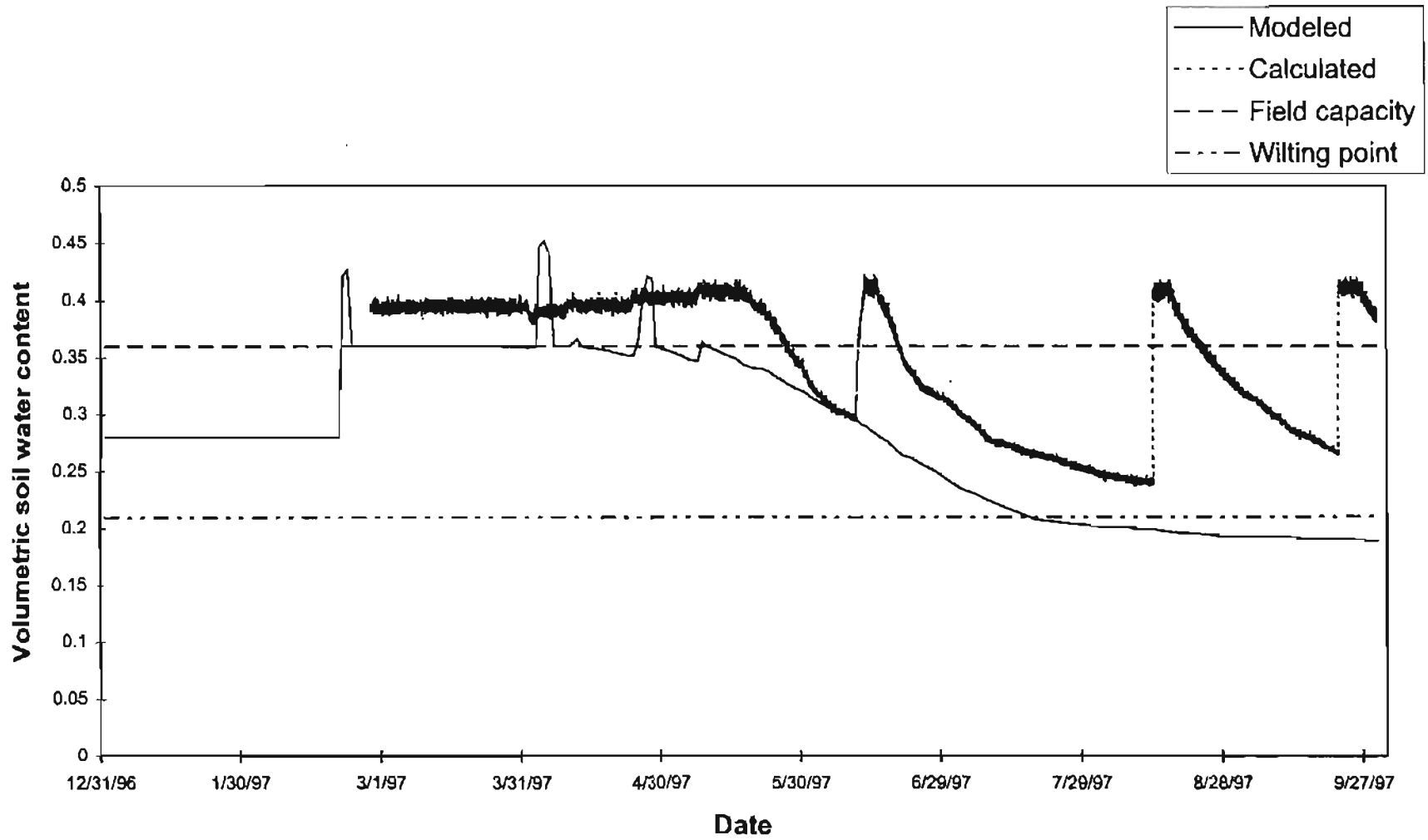


Figure 5.11. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Hollis

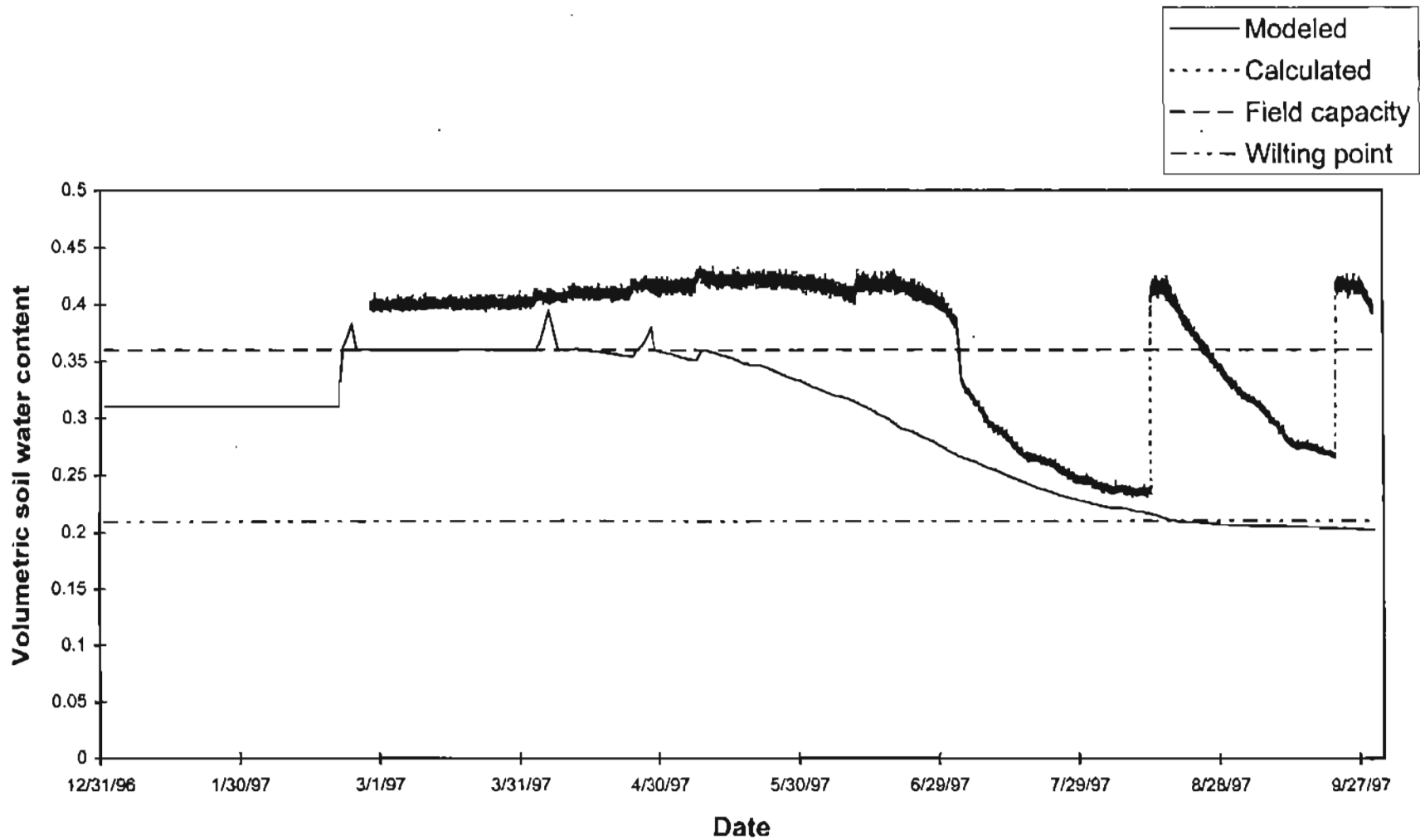


Figure 5.12. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Hollis

first jump, rainfall was insufficient to support this much increase in the soil water content. The observations show an increase of 25% at 5 cm, 18% at 25 cm, 17% at 60 cm and 18% at 75 cm. To increase the water content of a 750 mm deep soil column by 18%, 135 mm of water are required. The rainfall during this period, 8/9/1997 – 8/16/1997, is only 78 mm. This provides further evidence of the uncertainty in the process of converting observed ΔT values to water contents.

Miami

The results at Miami matched the observed pattern of soil water in most cases (figures 5.13 through 5.16). The major discrepancy is the low initialization of the first soil layer. The soil water content in the first layer was initialized at 22%, but the calculations show the soil water content at the beginning of the year to be about 44% (figure 5.13). The next two layers (25 cm and 60 cm) also were initialized at lower soil water contents but the difference was not as great. The model underestimated the soil water content at these depths consistently for the first four months of the year. The soil water content at 75 cm was initialized at 36% and the observed value too was 36%. The simulation results at this depth matched very well with the observed soil water, throughout the simulation period. At 5 cm and 25 cm depths, the observed soil water profile showed steep drops during the growing period. These drops did appear in the simulated profile, but the changes were not as dramatic as in the observed profile.

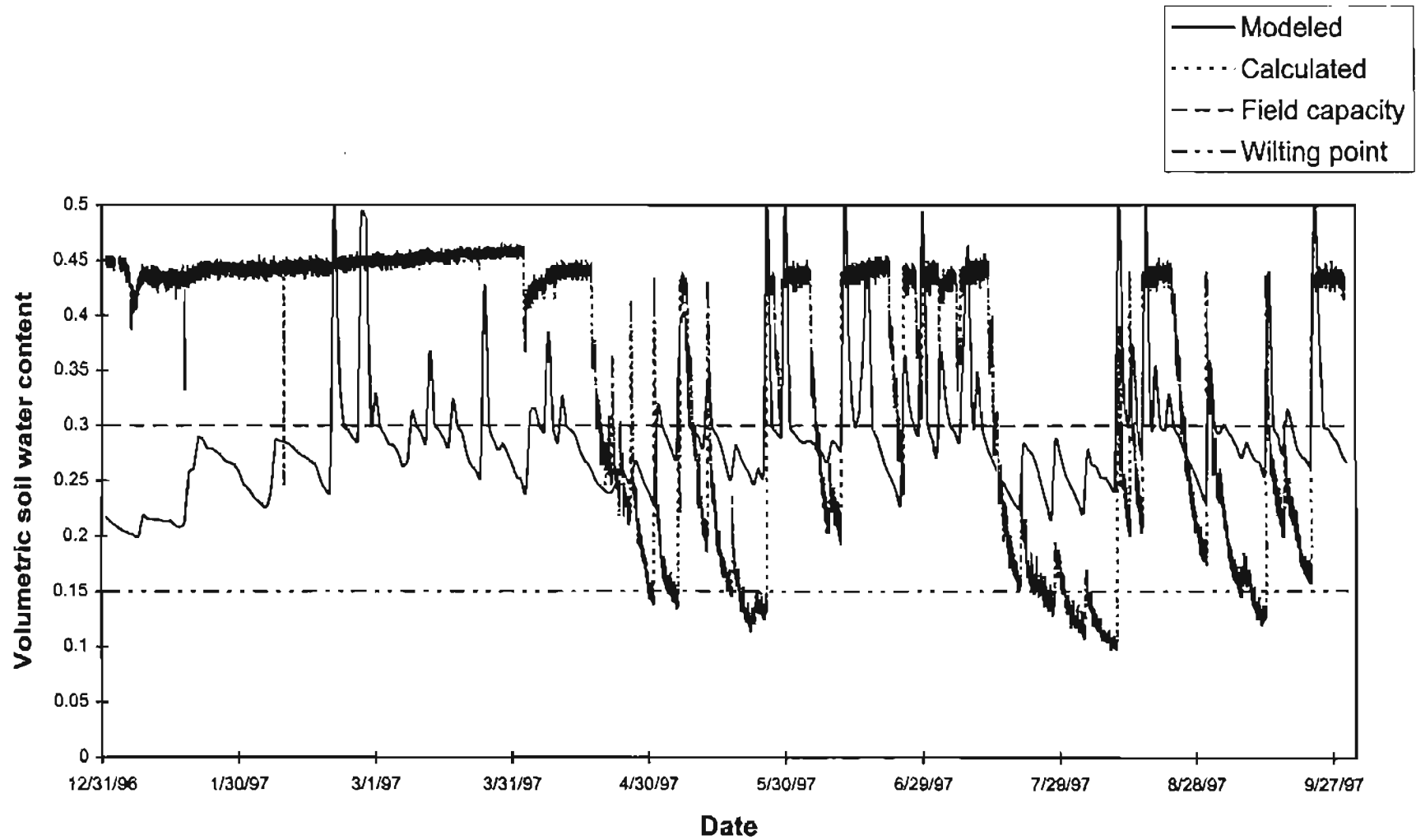


Figure 5.13. Soil water content at 5 cm for Jan. 1997 through Sep. 1997 at Miami

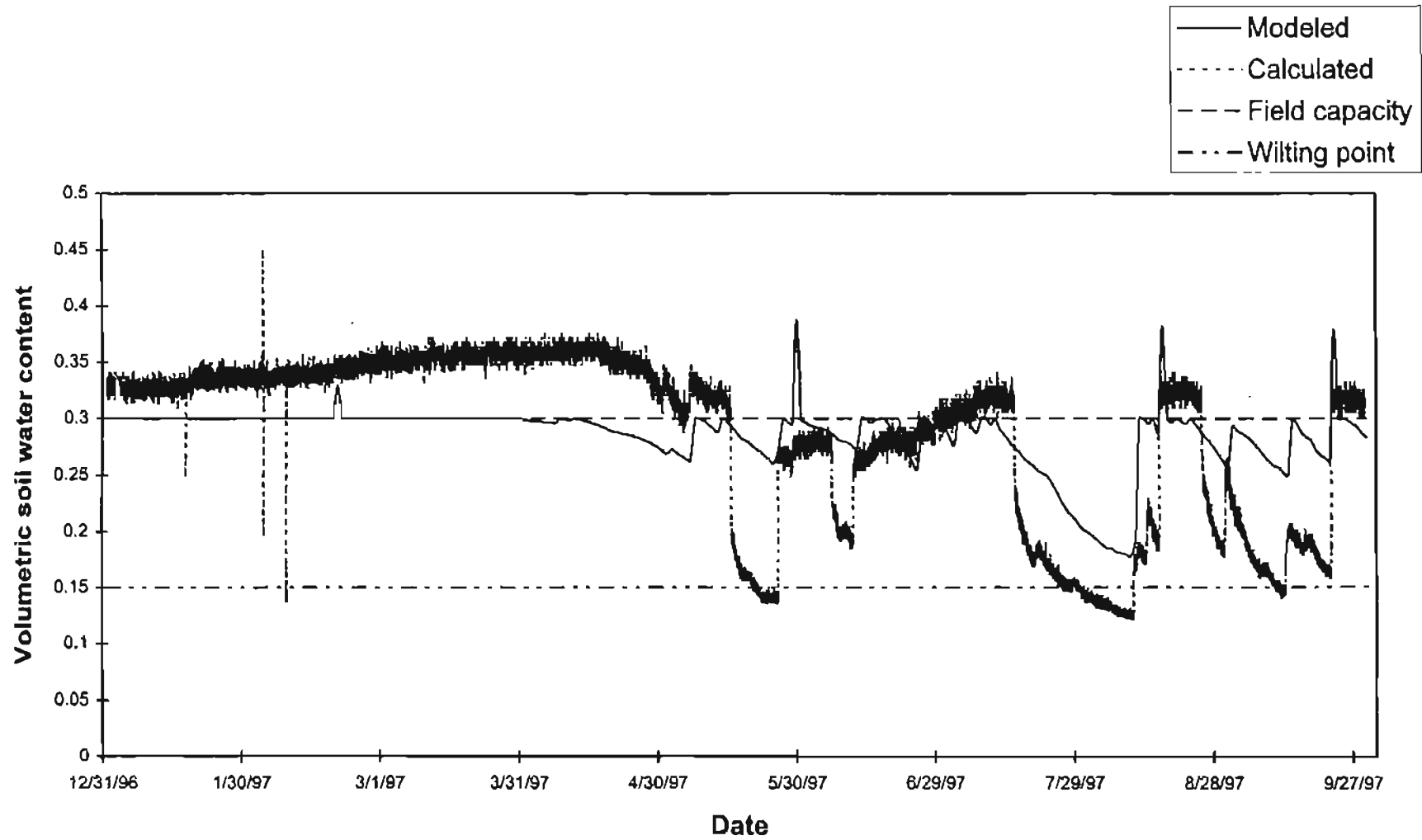


Figure 5.14. Soil water content at 25 cm for Jan. 1997 through Sep. 1997 at Miami

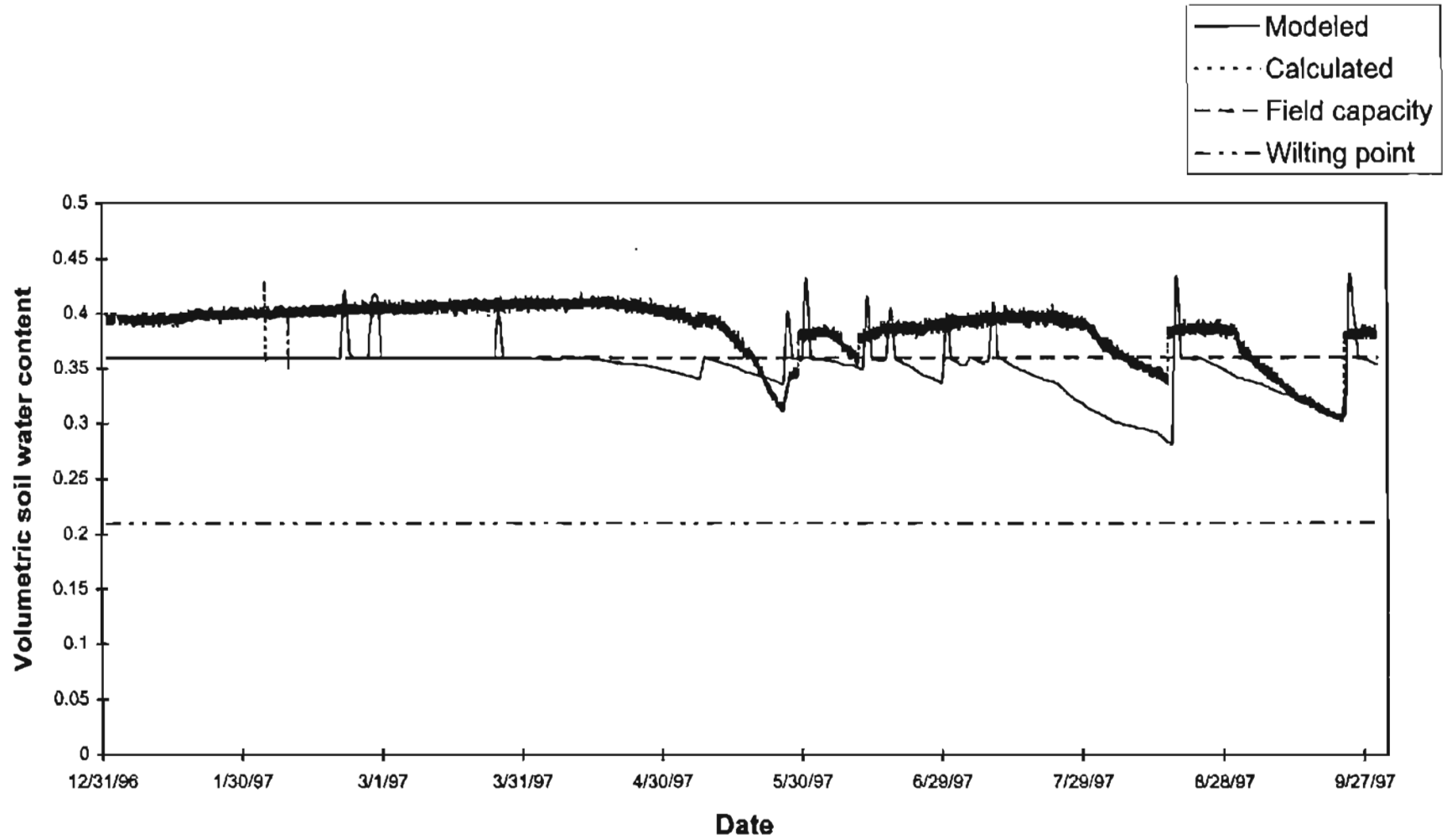


Figure 5.15. Soil water content at 60 cm for Jan. 1997 through Sep. 1997 at Miami

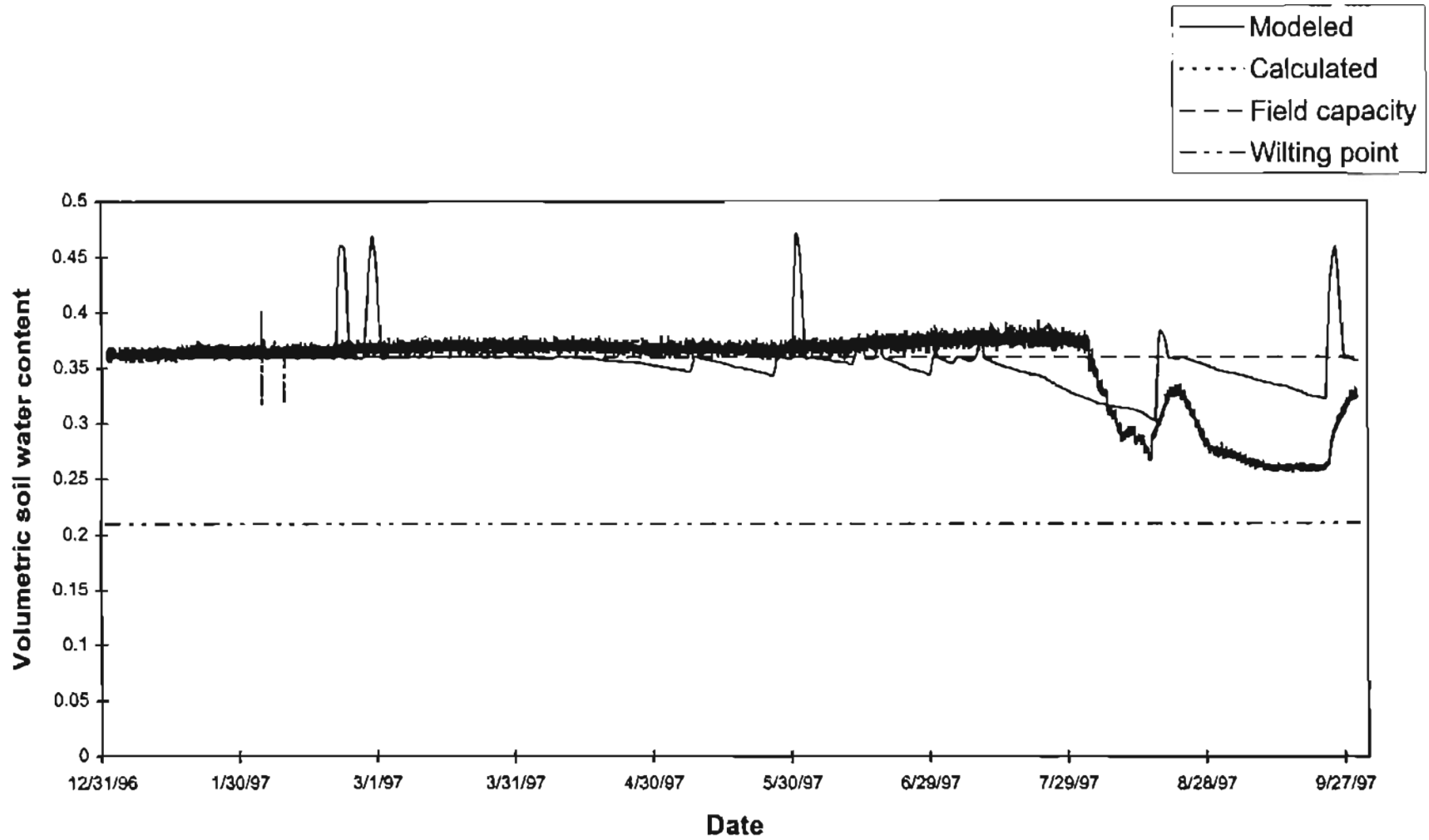


Figure 5.16. Soil water content at 75 cm for Jan. 1997 through Sep. 1997 at Miami

Summary

As pointed out previously, the 229L sensor provides a ΔT output which can be calibrated to soil matric potential. The extension to volumetric water content is considerably more tenuous. The observed (calculated) data can, at best, be used to study the pattern of the soil water. The model results were compared with the soil water content values estimated from the ΔT observations. Overall at the validation sites, the model did a good job of simulating the behavior of the soil water in the upper layers. But its performance did not appear to be as good at lower depths.

SENSITIVITY ANALYSIS

An analysis was performed to determine the sensitivity of the model to changes in the model parameter values. This would help the users of this model to know which parameters require the most attention and the most accurate estimates, and which parameters could be estimated less accurately without greatly affecting the performance of the model. The sensitivity analysis was performed on all soil and vegetation parameters.

Marena was the site selected for the sensitivity analysis. The sensitivity analysis was performed using the 1996 data. First, the model was run with the "best set" of input data as obtained after the calibration. The resulting output was considered to be the "base" case. Then each parameter was varied (high and low), while keeping the other parameters unchanged. The results from the high and the low estimates of each parameter were compared against the base results. During the sensitivity analysis, the effects of a

parameter on ET and soil water predictions were presented with time-series plots. In some cases the ET results from the low and the high estimates were plotted against the ET results from the base estimate to better visualize the deviations. The soil water predictions were presented as average volumetric soil water content in a control volume of 125 cm (base estimate for the root depth was 100 cm).

The estimates for the soil water parameters, excluding the field capacity and the wilting point, were their mean values, as used in the model development (table 4.3, page). The values were geometric means for the bubbling pressure and the pore size distribution index. They were varied \pm one standard deviation from their mean. Saturated hydraulic conductivity was varied \pm 25% as standard deviation, or any other measure of variability was not available for this parameter. The base estimates for the field capacity and the wilting point were obtained from the calibration results. They were varied from their base estimates by \pm 5% water content. The soil textures reported at Marena are sandy clay loam and loam. The base, high and low estimates of the soil water parameters for these textures are given in tables 5.4 (loam) and 5.5 (sandy clay loam).

The base estimates for other parameters were obtained from the calibration results. The parameter values were varied over different ranges depending on the parameter selected. The base estimates and the range over which they were varied are given in table 5.6.

Table 5.4. Values of the soil water parameters for loam, used in the sensitivity analysis.

Soil water parameter	Base estimate	Range	
		Low	High
Porosity	0.463	0.380	0.550
Field capacity	0.23	0.18	0.28
Wilting point	0.12	0.07	0.17
Residual saturation	0.027	0.0	0.074
Saturated hydraulic conductivity (cm h ⁻¹)	1.32	0.99	1.65
Bubbling pressure (cm)	11.15	1.63	76.40
Pore size distribution index	0.220	0.137	0.355

Table 5.5. Values of the soil water parameters for sandy clay loam, used in the sensitivity analysis.

Soil water parameter	Base estimate	Range	
		Low	High
Porosity	0.398	0.330	0.460
Field capacity	0.25	0.20	0.30
Wilting point	0.13	0.08	0.18
Residual saturation	0.068	0.0	0.137
Saturated hydraulic conductivity (cm h ⁻¹)	0.43	0.32	0.54
Bubbling pressure (cm)	28.08	5.57	141.5
Pore size distribution index	0.250	0.125	0.502

Table 5.6. Values of input parameters used in the sensitivity analysis.

Parameter	Base	Range	
	Estimate	Low	High
SCS Curve Number	74	65	85
Maximum LAI	5	3.75	6.25
LAI at max. fractional transpiration factor	3	2	4
Coefficient of shelter factor	0.1	0.05	0.2
Root depth (cm)	100	75	125
Minimum canopy height (cm)	30	10	50
Maximum canopy height (cm)	80	60	100
Minimum stomatal resistance (s m ⁻¹)	166	124.5	207.5
Maximum stomatal resistance (s m ⁻¹)	5000	2500	7500
Min. fractional transpiration factor	0.5	0.4	0.6
Max. fractional transpiration factor	0.9	0.8	1.0
Fraction of ASW depletion at no stress	0.5	0.4	0.6
Critical dates of growth			
Start of growth (D _{sg})	March 26	March 12	April 9
At max. LAI (D _{mx})	June 23	June 9	July 7
Start of senescence (D _{ss})	August 20	August 6	September 3
Start of dormancy (D _{sd})	November 4	October 21	Nov. 18

Some parameters were varied $\pm 25\%$ of their base values, e.g. maximum LAI, minimum stomatal resistance etc. Some others were varied over a reasonable range determined by literature or observation, e.g. fraction of available soil water (ASW) depletion at no stress (Allen et al., 1996b). Critical dates of growth were varied by ± 14 days.

Soil parameters

Among the soil parameters, the field capacity and the wilting point have the greatest effect on the performance of the model. These parameters determine the upper and the lower levels of the soil water content in the root zone excluding the first layer which can dry below the wilting point. As expected, increasing the field capacity raised the soil water content in the root zone whereas decreasing it lowered the soil water content (figure 5.17). Since the initial soil water content was assumed to be the field capacity, the three profiles are separated from the very first day. The wilting point too has the same effect on the soil water profile (figure 5.18) but the profiles separated after transpiration had started. Due to different wilting point values, the water stress on the transpiration started at different times and thus the three cases became separated from one another. If there is enough rainfall to raise the soil water content to the field capacity in all three cases, it would bring the soil water content to the same points, as field capacity and other soil parameters are the same. But they would again separate as soon as water stress occurred.

The field capacity and the wilting point have no direct effect on ET prediction. Indirectly, they affect ET predictions by determining when the water stress begins. A high estimate of the field capacity would delay the occurrence of water stress and thus would cause ET to be higher than for the base case. A low estimate of the field capacity would let the water stress occur earlier and thus would cause ET to be lower (figure 5.19). On the other hand a high estimate of the wilting point would let the water stress occur earlier whereas a low estimate would delay it (figure 5.20).

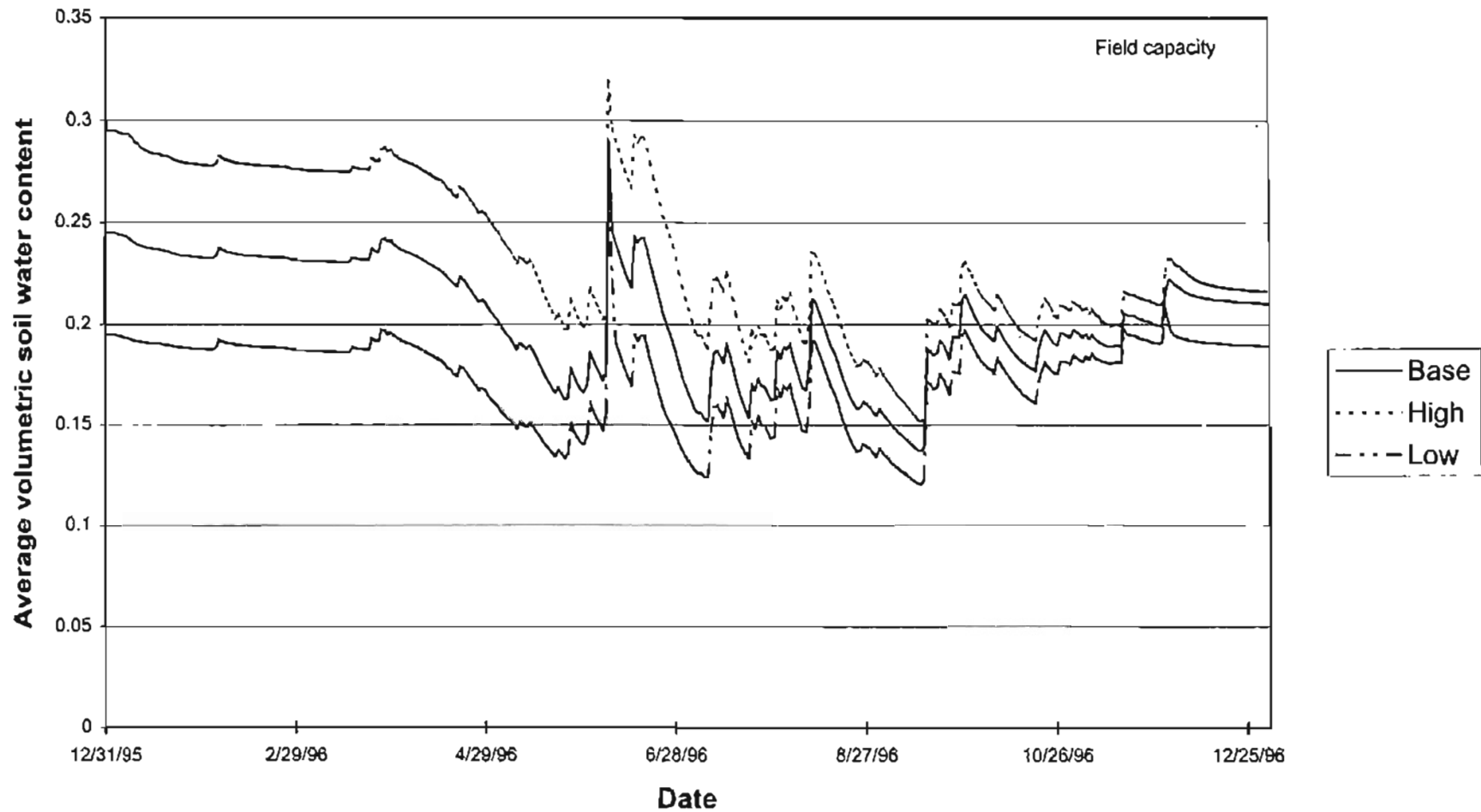


Figure 5.17. Sensitivity of soil water content to the field capacity (Base = 0.25, 0.23; Low = 0.20, 0.18; High = 0.30, 0.28 for sandy clay loam and loam, respectively)

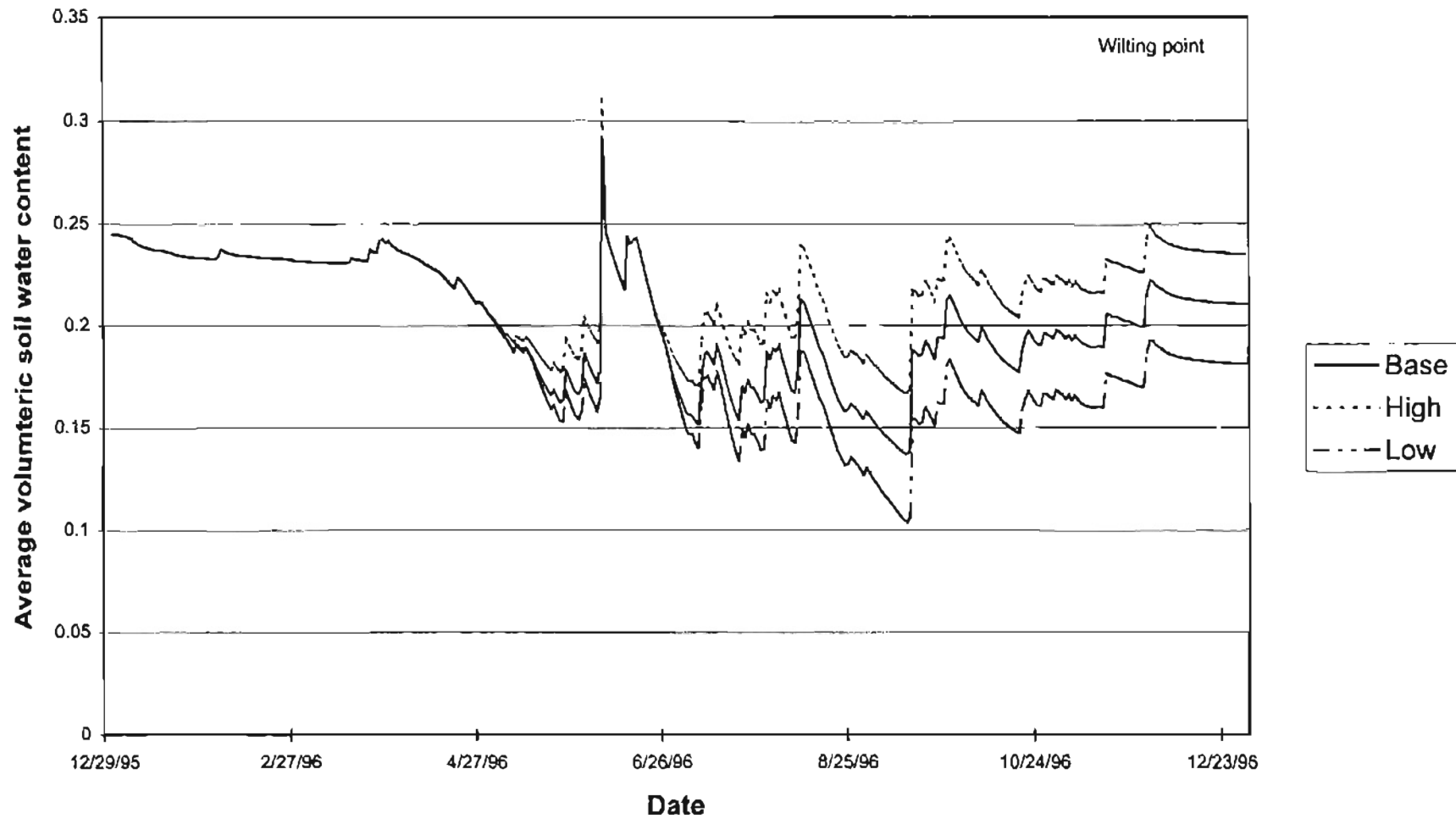


Figure 5.18. Sensitivity of soil water content to the wilting point (Base = 0.13, 0.12; Low = 0.08, 0.07; High = 0.18, 0.17 for sandy clay loam and loam, respectively)

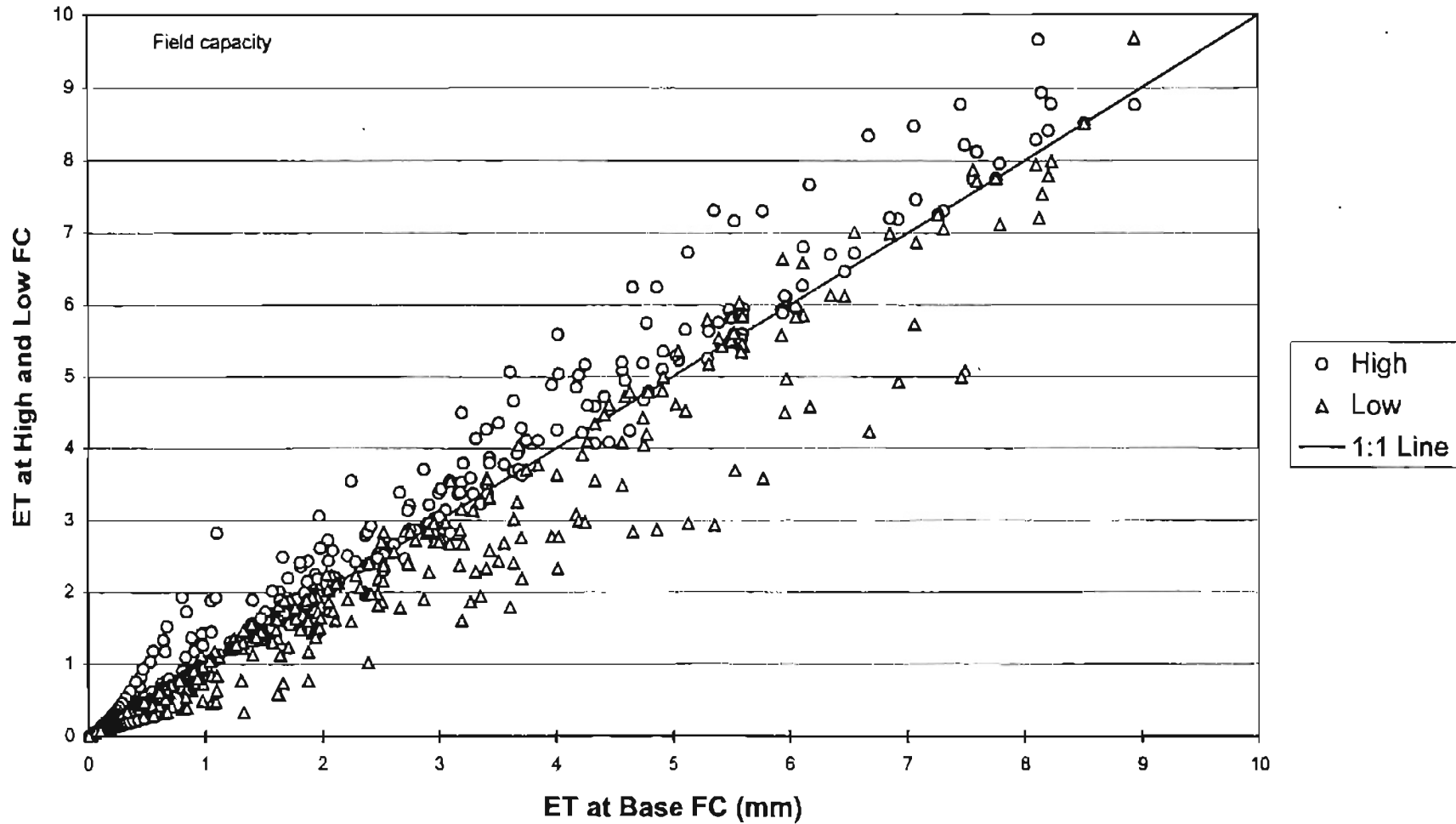


Figure 5.19. Sensitivity of ET to the field capacity (Base = 0.25, 0.23; Low = 0.20, 0.18; High = 0.30, 0.28 for sandy clay loam and loam, respectively)

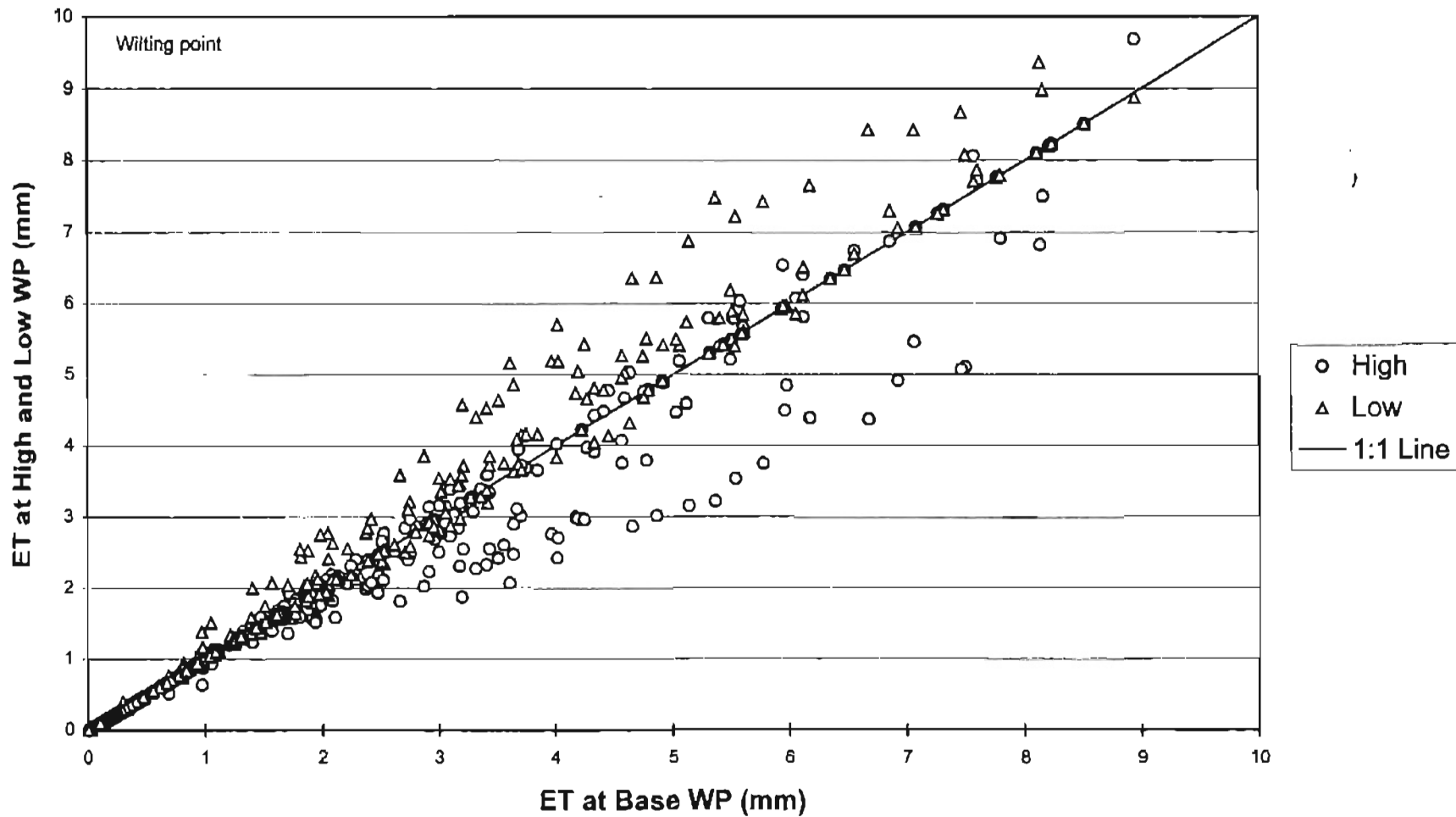


Figure 5.20. Sensitivity of ET to the wilting point (Base = 0.13, 0.12; Low = 0.08, 0.07; High = 0.18, 0.17 for sandy clay loam and loam, respectively)

Other soil water parameters are used in estimating the desorption volume for calculating soil evaporation. They primarily affect the modeled soil water profile only in dormant periods when transpiration is zero and soil evaporation is high. In active growth periods, soil evaporation relative to transpiration and thus the effect of these soil water parameters is minimal. Since soil evaporation takes place from a 15 cm thick soil layer out of the 125 cm thick root zone, its effect is very little on the overall soil water content. Therefore the effects of these parameters were studied only with regards to soil evaporation and the soil water content in the first layer.

The effect of bubbling pressure on the soil water is shown in figure 5.21. Its effect on soil evaporation is shown in figure 5.22. As expected, the soil water parameters had negligible effects on the predictions during the active growing period. Though changes in the soil water content of the first layer are significant during the dormant period, effects on evaporation were negligible. The sensitivity of the model to other soil water parameters is similar or even less.

Vegetation parameters

The minimum stomatal resistance seems to be the most important vegetation parameter in terms of model sensitivity. Its effect on the ET is shown in figure 5.23. As expected a high estimate of this parameter would tend to decrease the ET. Its effect is most pronounced under conditions of high ET. Its consequence on the soil water in the root zone is shown in figure 5.24. A high estimate of this parameter would increase the

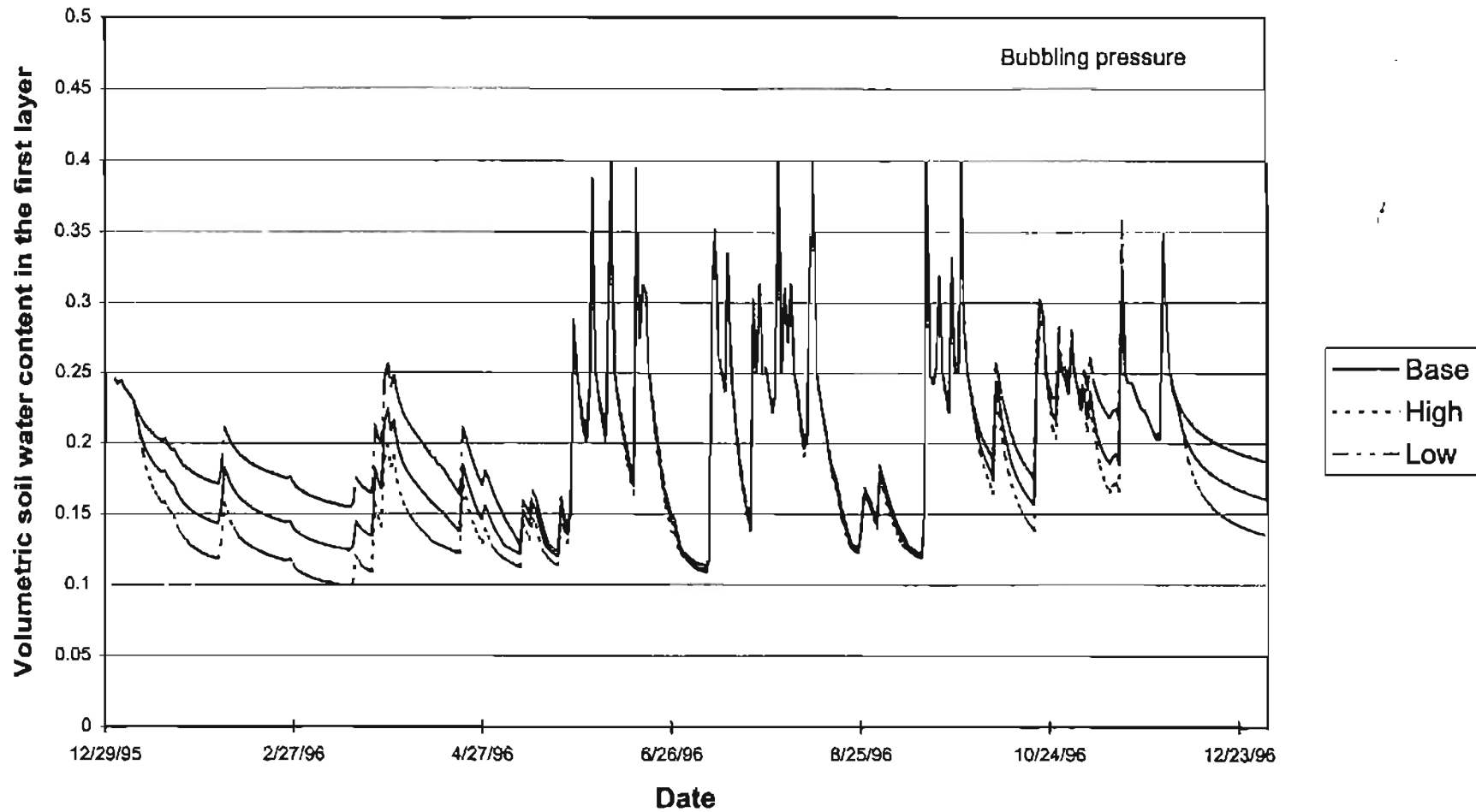


Figure 5.21. Sensitivity of soil water content of the first layer to the bubbling pressure (Base = 28.08 cm, 11.15 cm; Low = 5.57 cm, 1.63 cm; High = 141.5 cm, 76.40 cm for sandy clay loam and loam, respectively)

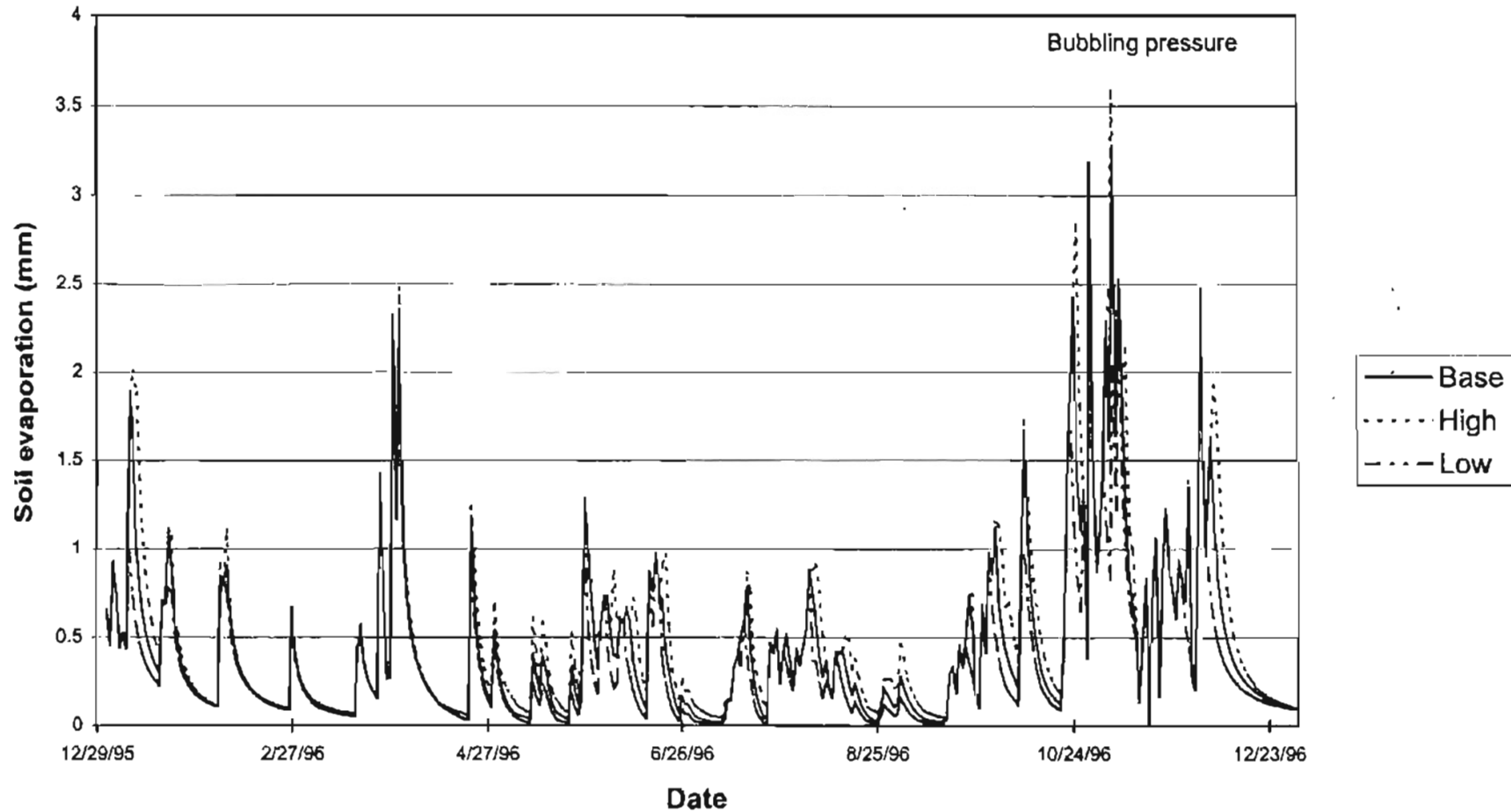


Figure 5.22. Sensitivity of soil evaporation to the bubbling pressure (Base = 28.08 cm, 11.15 cm; Low = 5.57 cm, 1.63 cm; High = 141.5 cm, 76.40 cm for sandy clay loam and loam, respectively)

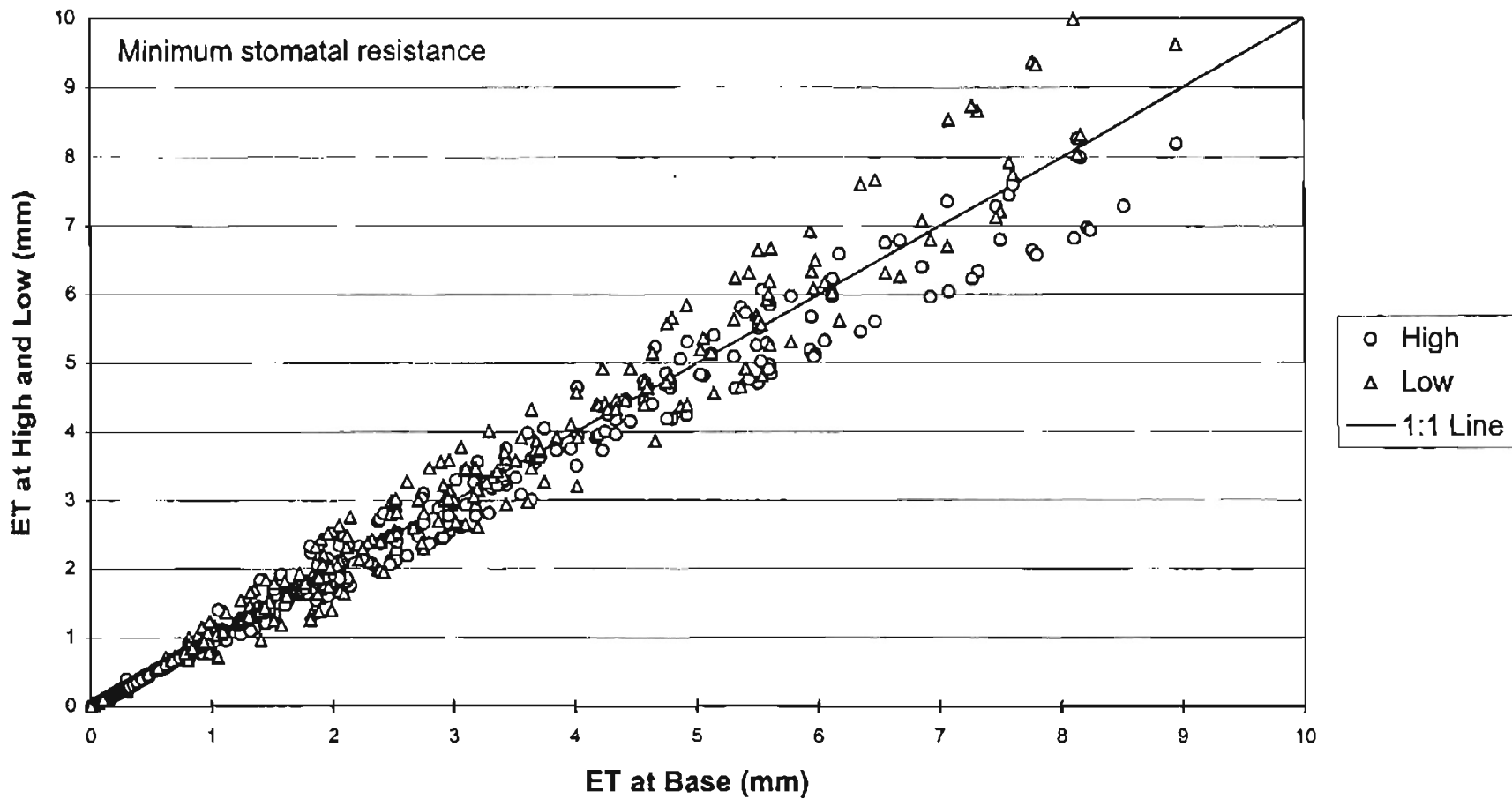


Figure 5.23. Sensitivity of ET to the minimum stomatal resistance (Base = 166 s m^{-1} ; Low = 124.5 s m^{-1} ; High = 207.5 s m^{-1})

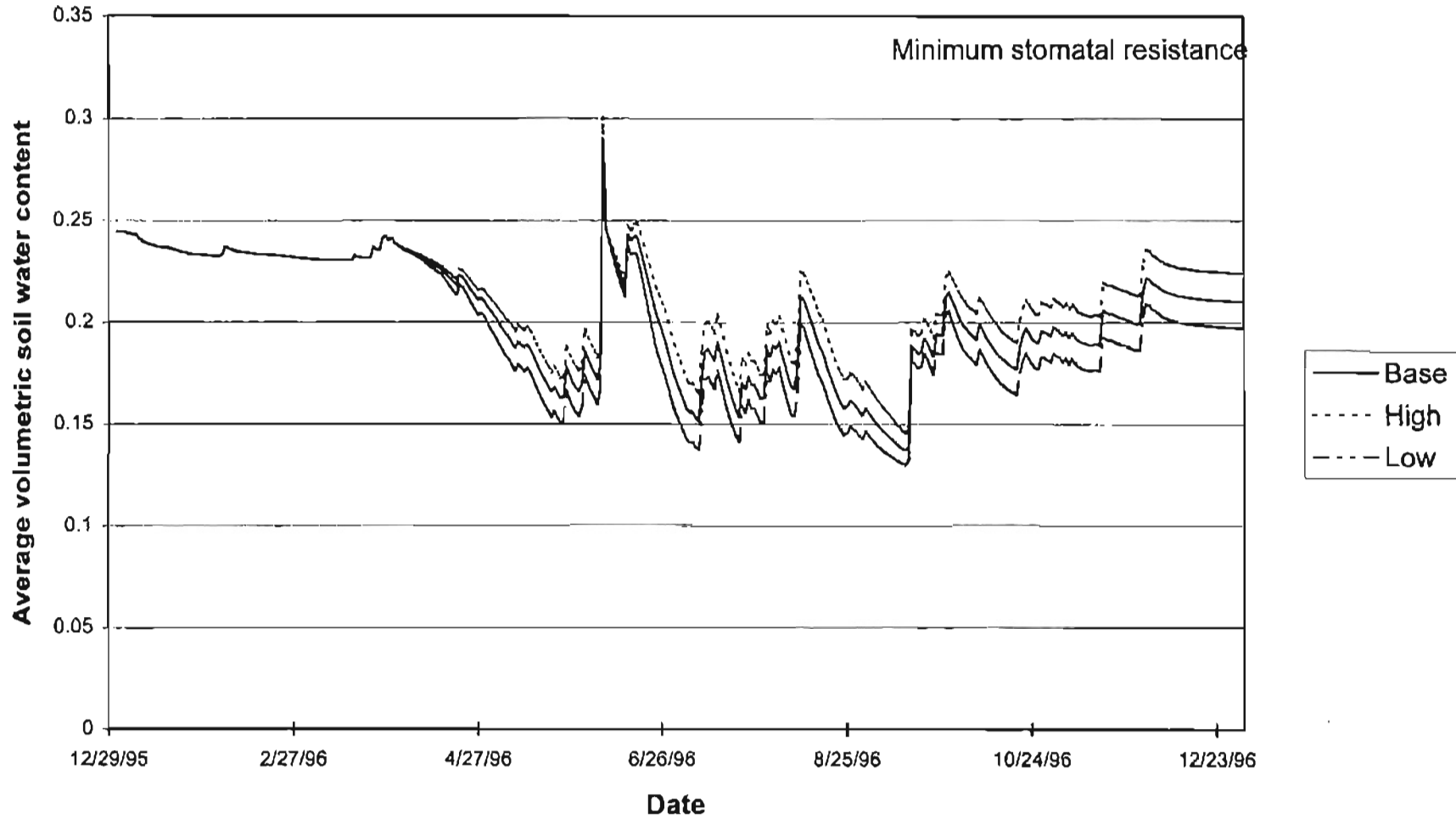


Figure 5.24. Sensitivity of soil water content to the minimum stomatal resistance (Base = 166 s m^{-1} ; Low = 124.5 s m^{-1} ; High = 207.5 s m^{-1})

soil water in the root zone by decreasing the ET. The maximum stomatal resistance has negligible effect on the model's performance.

Another important parameter is the maximum LAI. A change in the estimate of this parameter will also change the early-growth and the senescence slope of the LAI-time curve. Its effect on the predictions is shown in figures 5.25 and 5.26. A high estimate of this parameter increases the ET estimate. It affects the ET estimate more at low LAI than at high LAI. The reason is that the effective LAI changes very little at the higher LAI values due to the shelter factor. The model results show that the model is sensitive to the coefficient of the shelter factor (figures 5.27 and 5.28). Any change in this coefficient would affect the effective LAI, and would thus affect the ET prediction. A high coefficient results in a higher shelter factor and thus decreases the effective LAI. This in turn causes less ET. The effect of an increase in the estimate of this parameter can be nullified to some extent by an increase in the estimate of the maximum LAI.

Root depth becomes important under the water stressed condition. Its effects on the ET and the soil water content are shown in figures 5.29 and 5.30. Under such a condition, deeper roots allow access to a larger reservoir of soil water and therefore maintain higher ET rates. If there is no lack of soil water, increasing root depth would have no impact but decreasing it may cause water stress and thus may lower the ET. In figure 5.30, the effect of changing root depth on the soil water content is shown. Here the soil water content values correspond to root depths (i.e., depths of 75 cm, 100 cm and 125 cm), rather than a fixed control depth. The soil water content for the low case appears to be higher after rainfall events. That is because of its smaller thickness. After rainfall events, the top 75 cm of soil for the base or high case may be at the same water content as

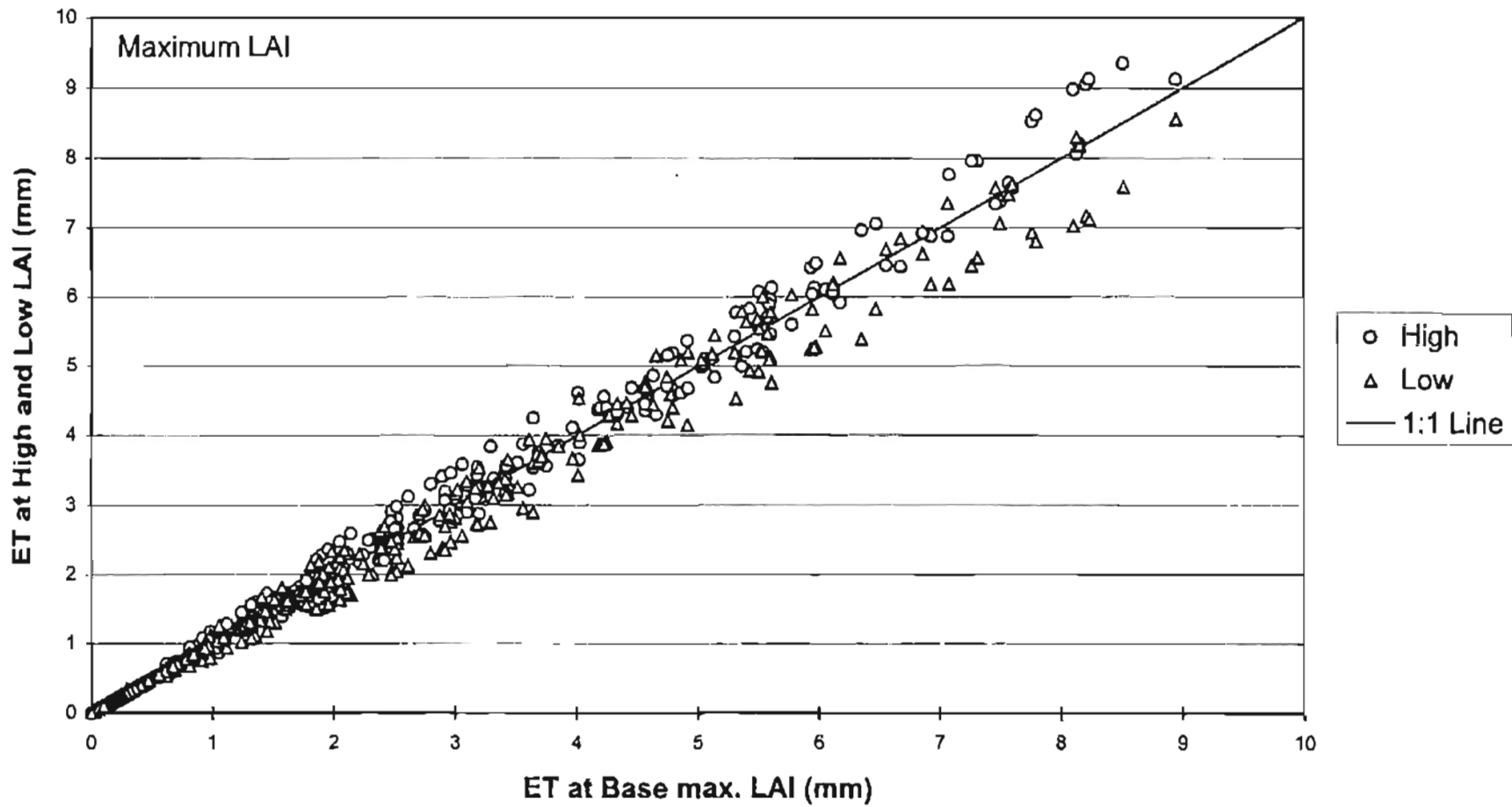


Figure 5.25. Sensitivity of ET to the maximum LAI (Base = 5; Low = 3.75; High = 6.25)

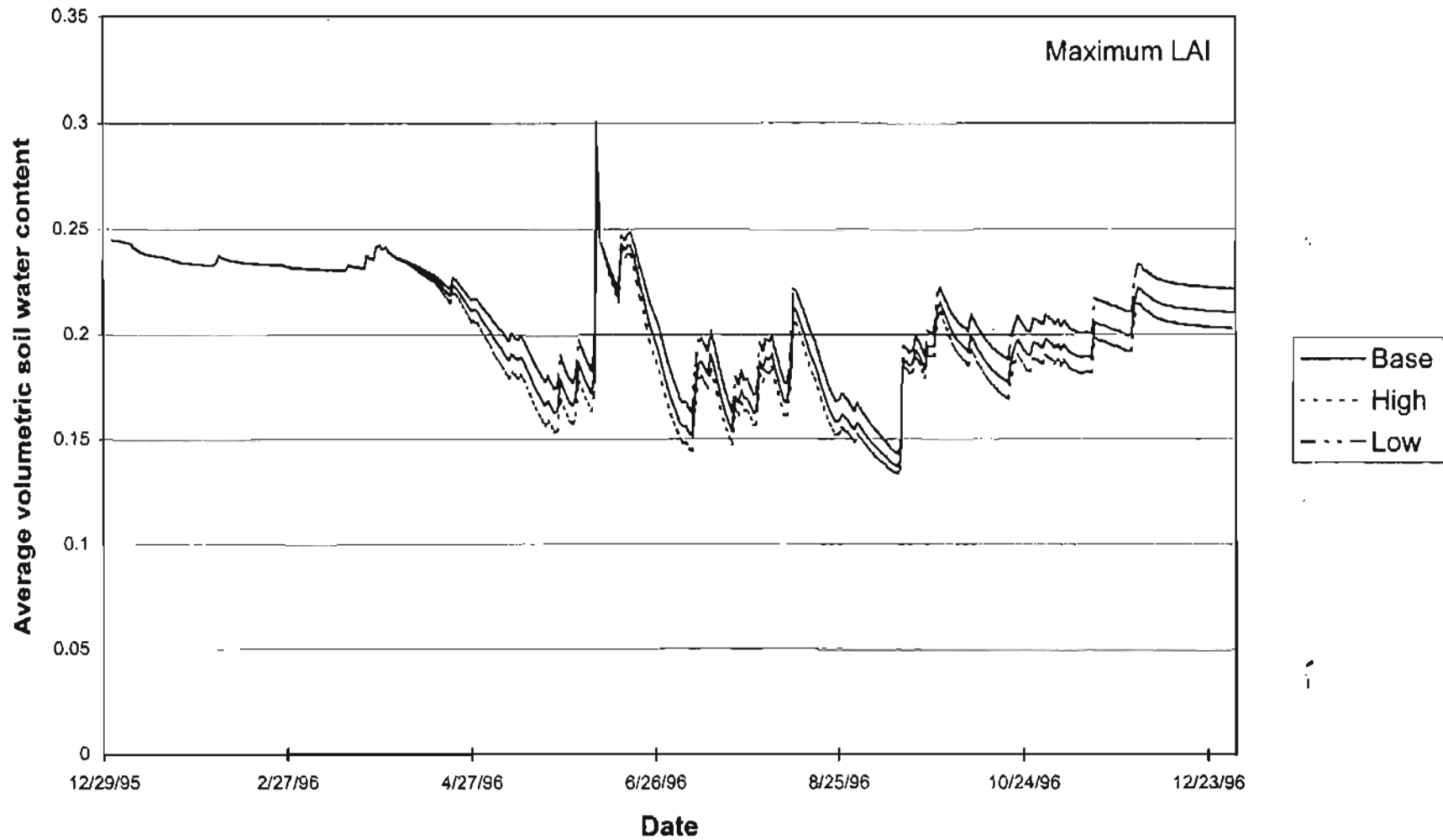


Figure 5.26. Sensitivity of soil water content to the maximum LAI (Base = 5; Low = 3.75; High = 6.25)

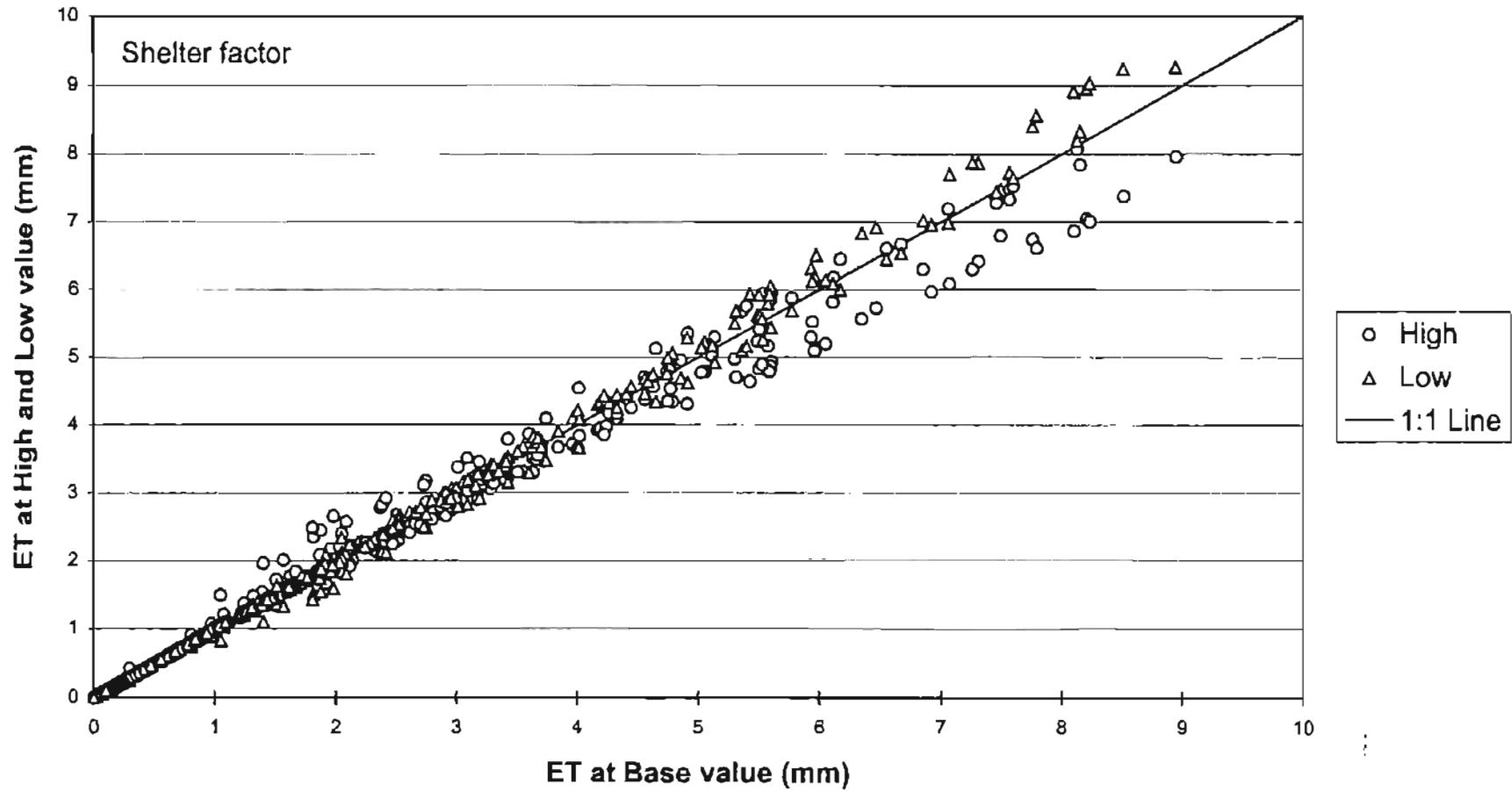


Figure 5.27. Sensitivity of ET to the coefficient of the shelter factor (Base = 0.1; Low = 0.05; High = 0.2)

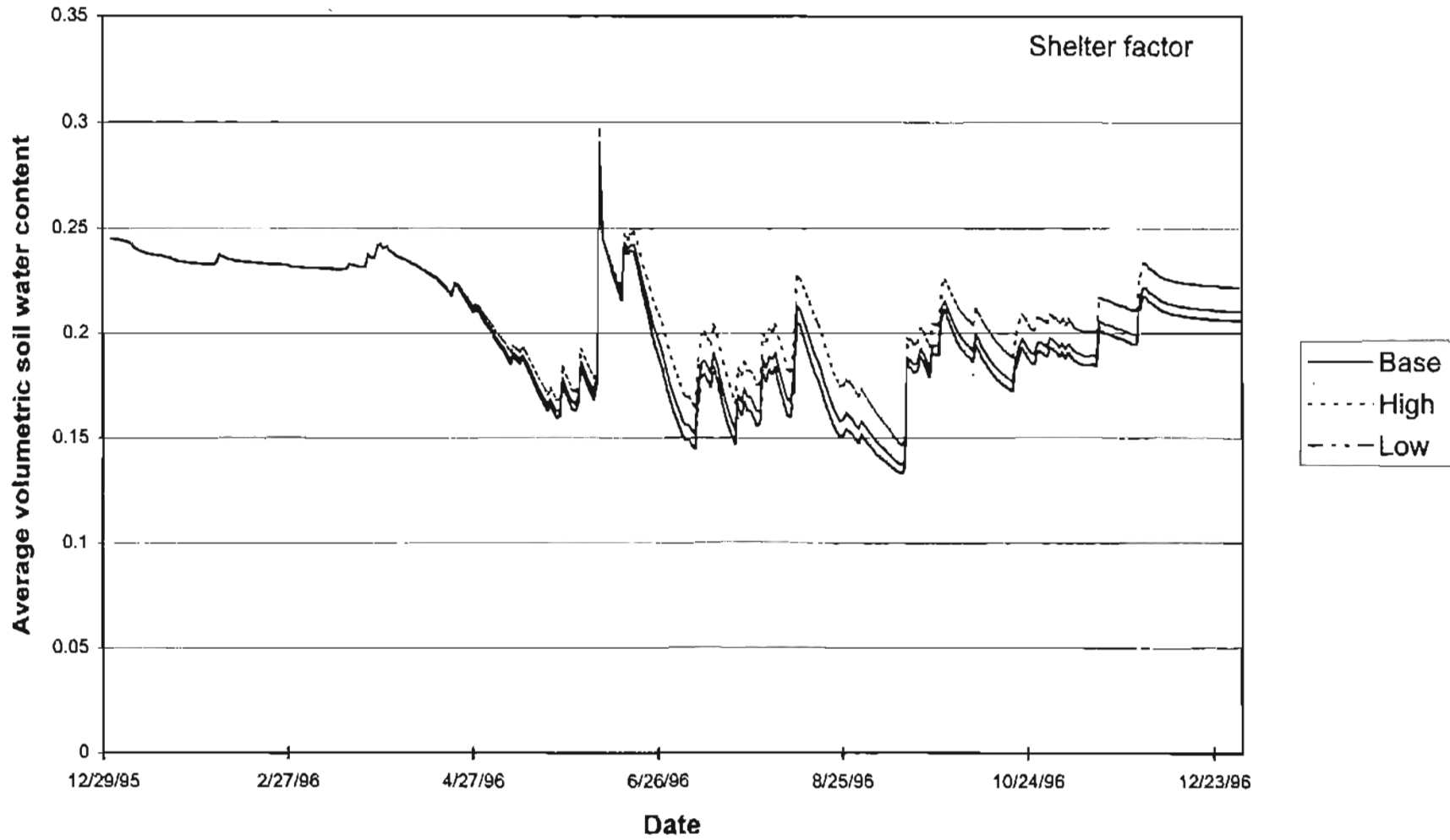


Figure 5.28. Sensitivity of soil water content to the coefficient of the shelter factor (Base = 0.1; Low = 0.05; High = 0.2)

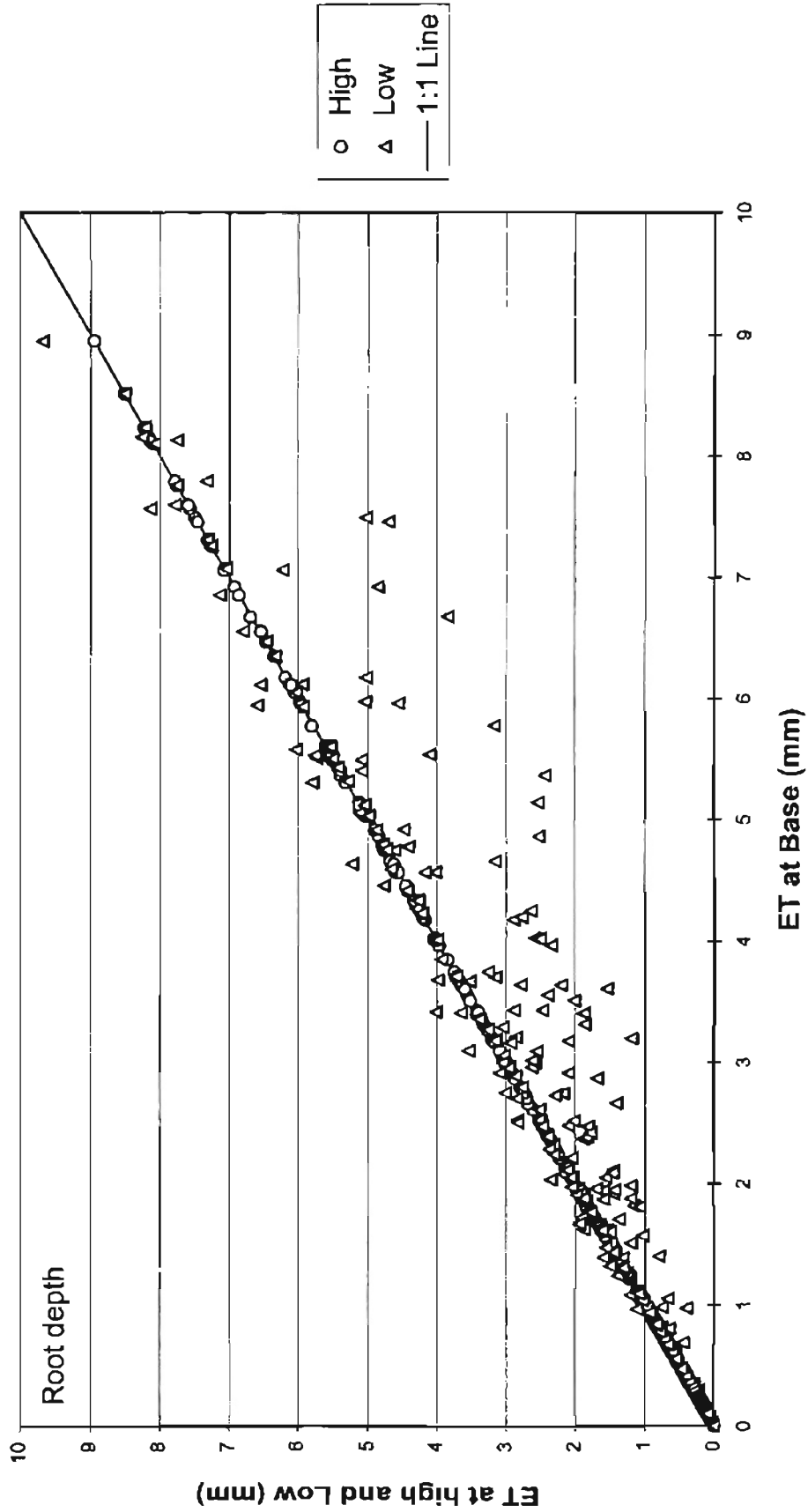


Figure 5.29. Sensitivity of ET to the root depth (Base = 100 cm; Low = 75 cm; High = 125 cm)

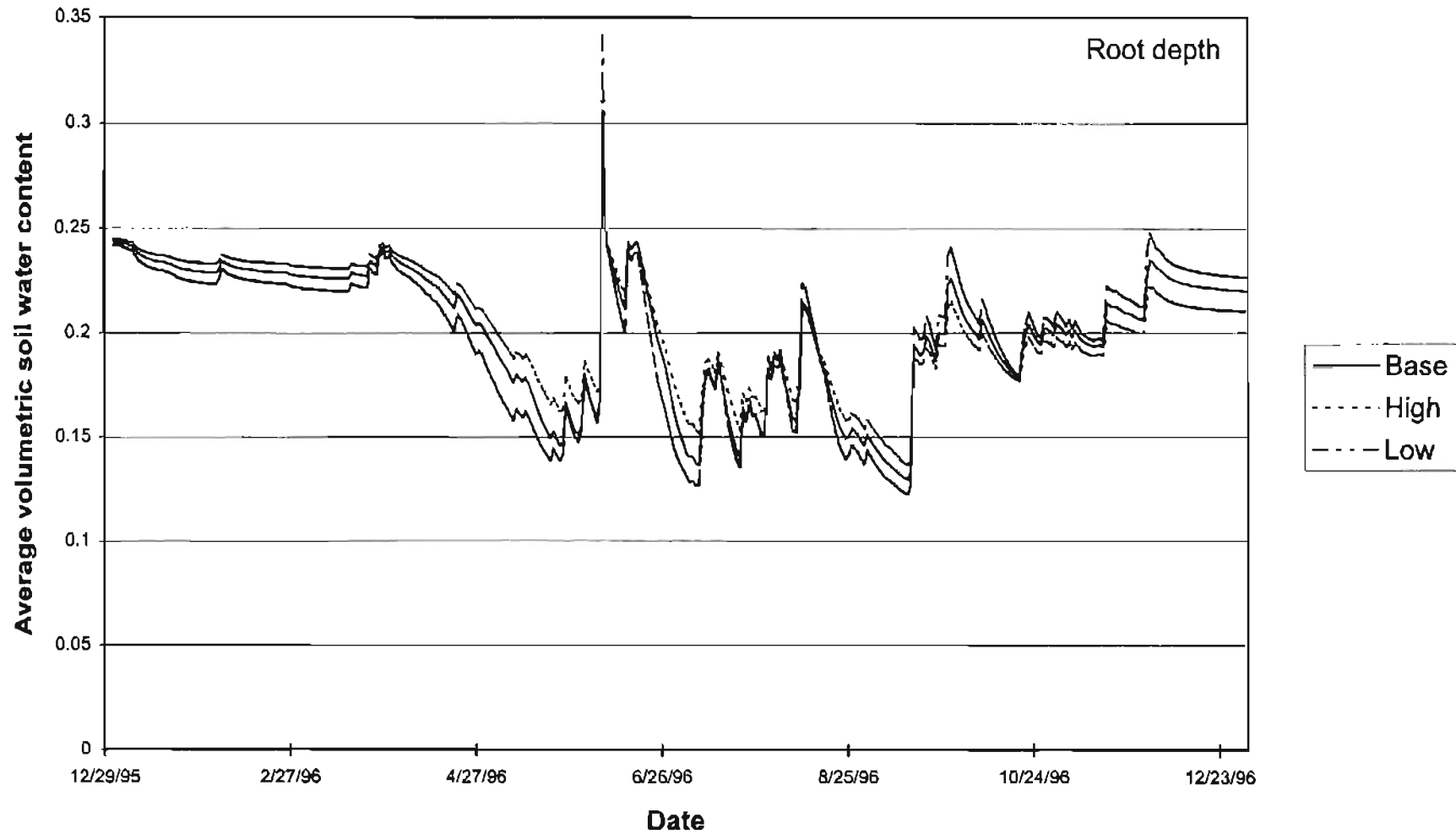


Figure 5.30. Sensitivity of soil water content to the root depth (Base = 100 cm; Low = 75 cm; High = 125 cm)

for the low case, but the average soil water content in a 100 cm or 125 cm depth may be less.

The minimum fractional transpiration factor weights the bare soil evaporation in the dormant period, whereas the maximum fractional transpiration factor determines the weighting factors between evaporation and transpiration during the peak of the growing season. Therefore the minimum fraction cover affects (slightly) the model results in the dormant period (figure 5.31), whereas the maximum fraction cover has a small effect on the model results in the peak season (figure 5.32).

The set of the critical dates of growth affects the timings of changes in ET and soil water. The first two dates, the date of the start of growth and the date at which maximum LAI is reached, affect the active growth whereas the other two dates, the date of the start of senescence and the date of the start of dormancy, affect the senescence. Any change in any of these dates would be shown as offsets in the time-series plots. The offset may not persist if there is sufficient rainfall later to raise the soil water content to field capacity. For example, in figure 5.33 the difference has disappeared due to a heavy rainfall, but the difference persists in figure 5.34 as there was not sufficient rainfall in the dormant period to raise the soil water content to field capacity. Other vegetation parameters, e.g. the minimum and the maximum canopy heights, have negligible effect on the model predictions.

The fraction of available soil water (ASW) depletion determines when water stress would begin. It affects the ET by causing stress to occur earlier or later. A high fraction would cause the stress to occur later. But the net effect on the ET estimation over

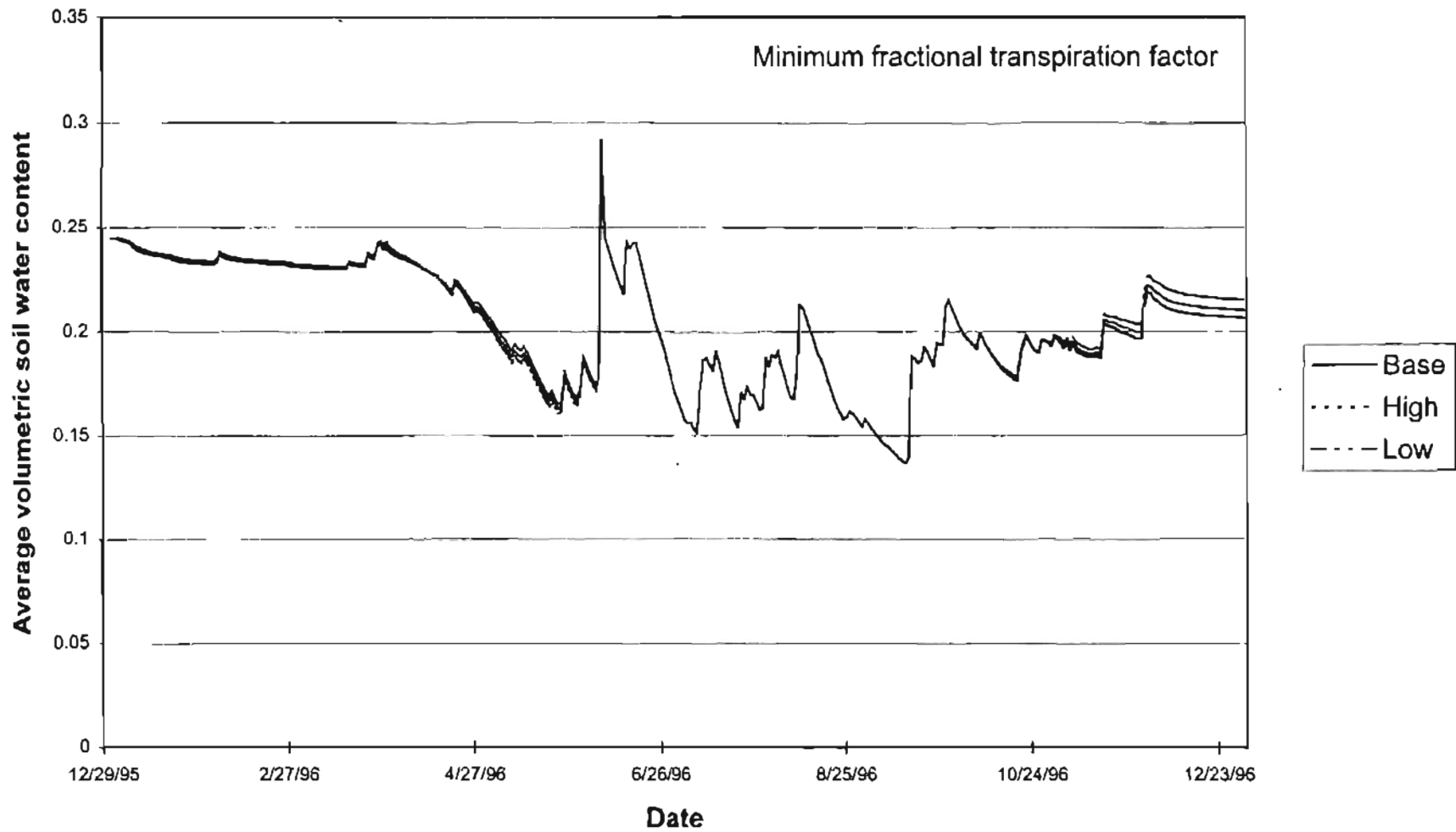


Figure 5.31. Sensitivity of soil water content to the minimum fractional transpiration factor (Base = 0.5; Low = 0.4; High = 0.6)

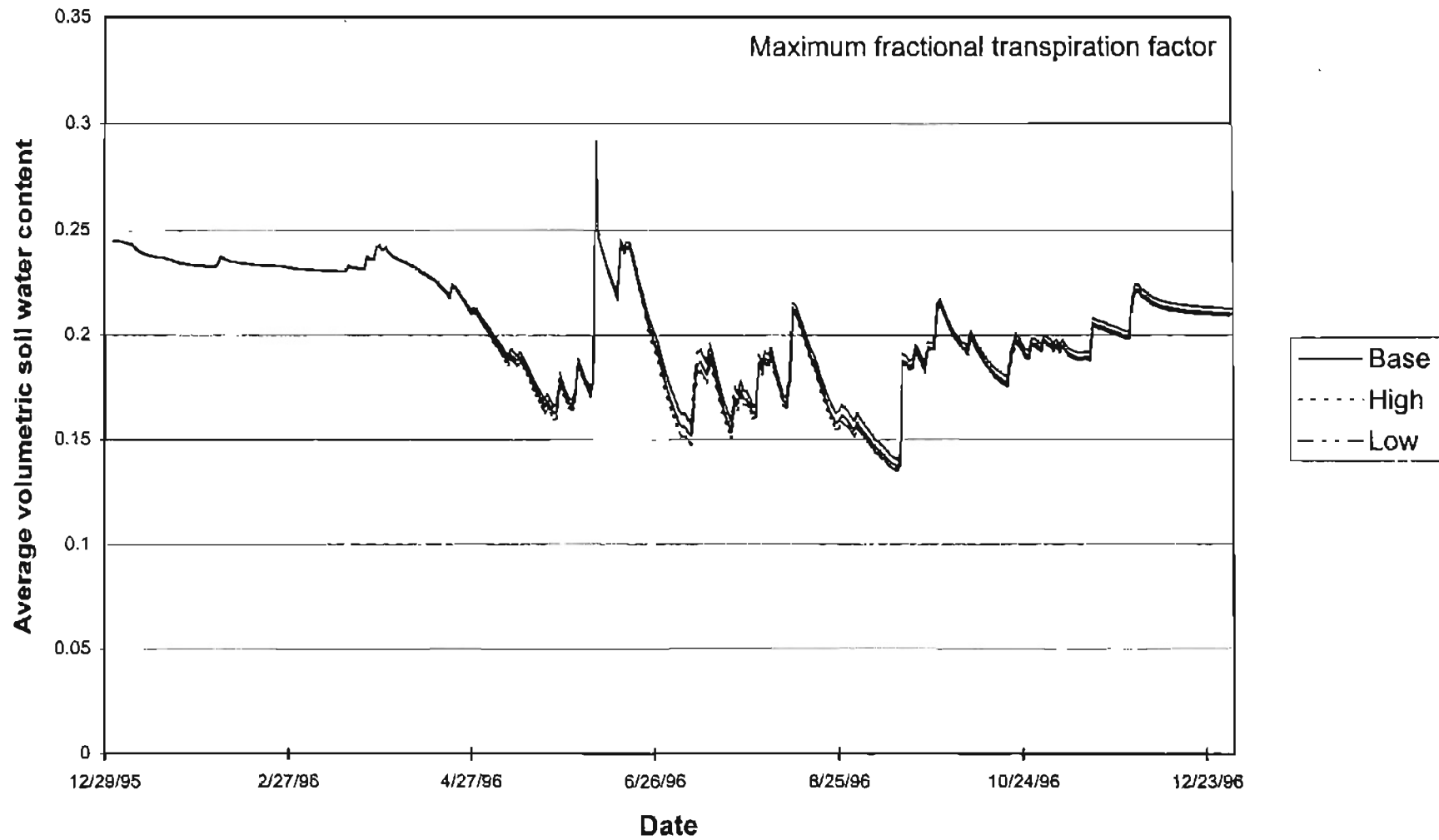


Figure 5.32. Sensitivity of soil water content to the maximum fractional transpiration factor (Base = 0.5; Low = 0.4; High = 0.6)

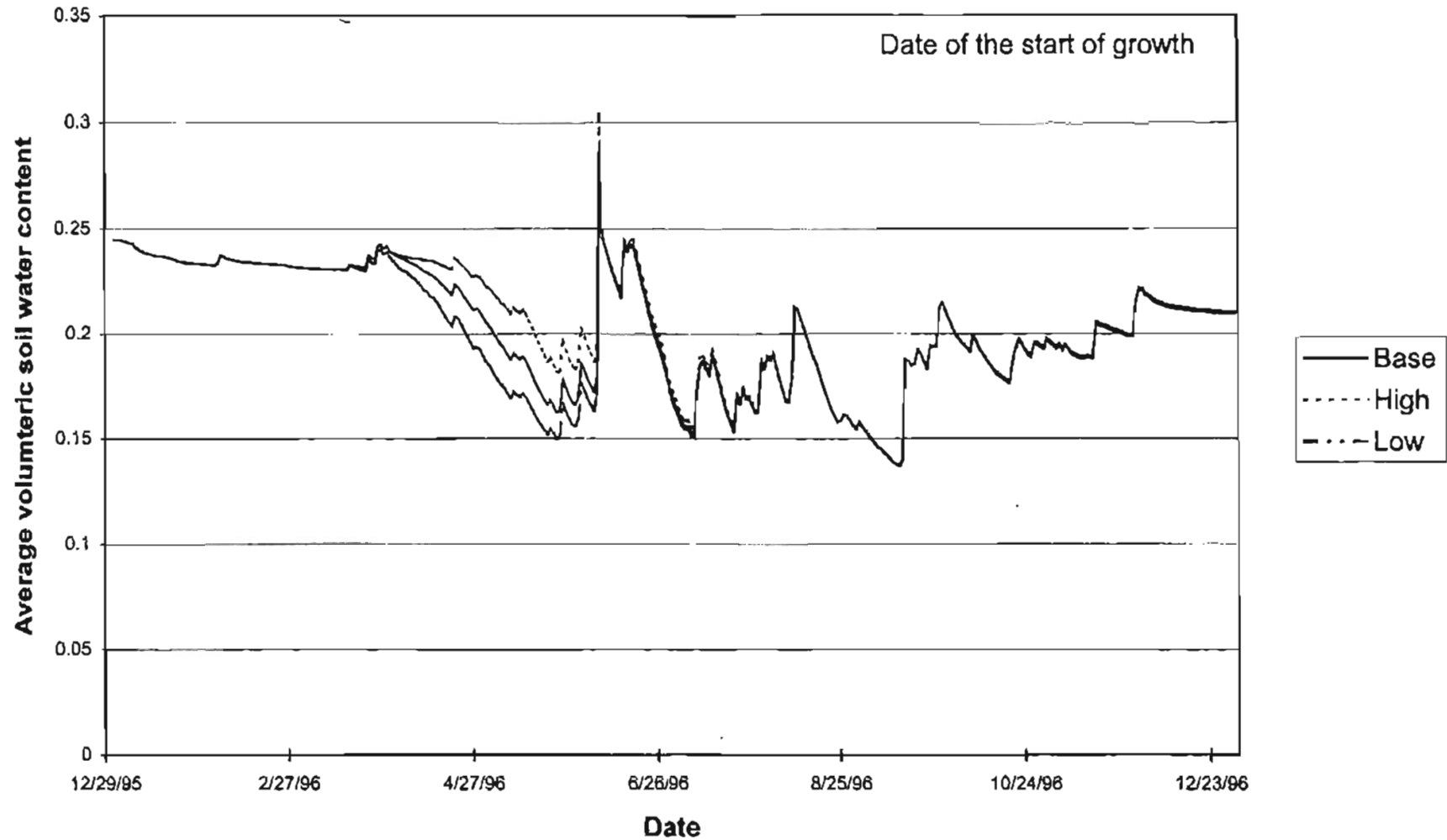


Figure 5.33. Sensitivity of soil water content to the date of the start of growth (Base = March 26; Low = March 12; High = April 9)

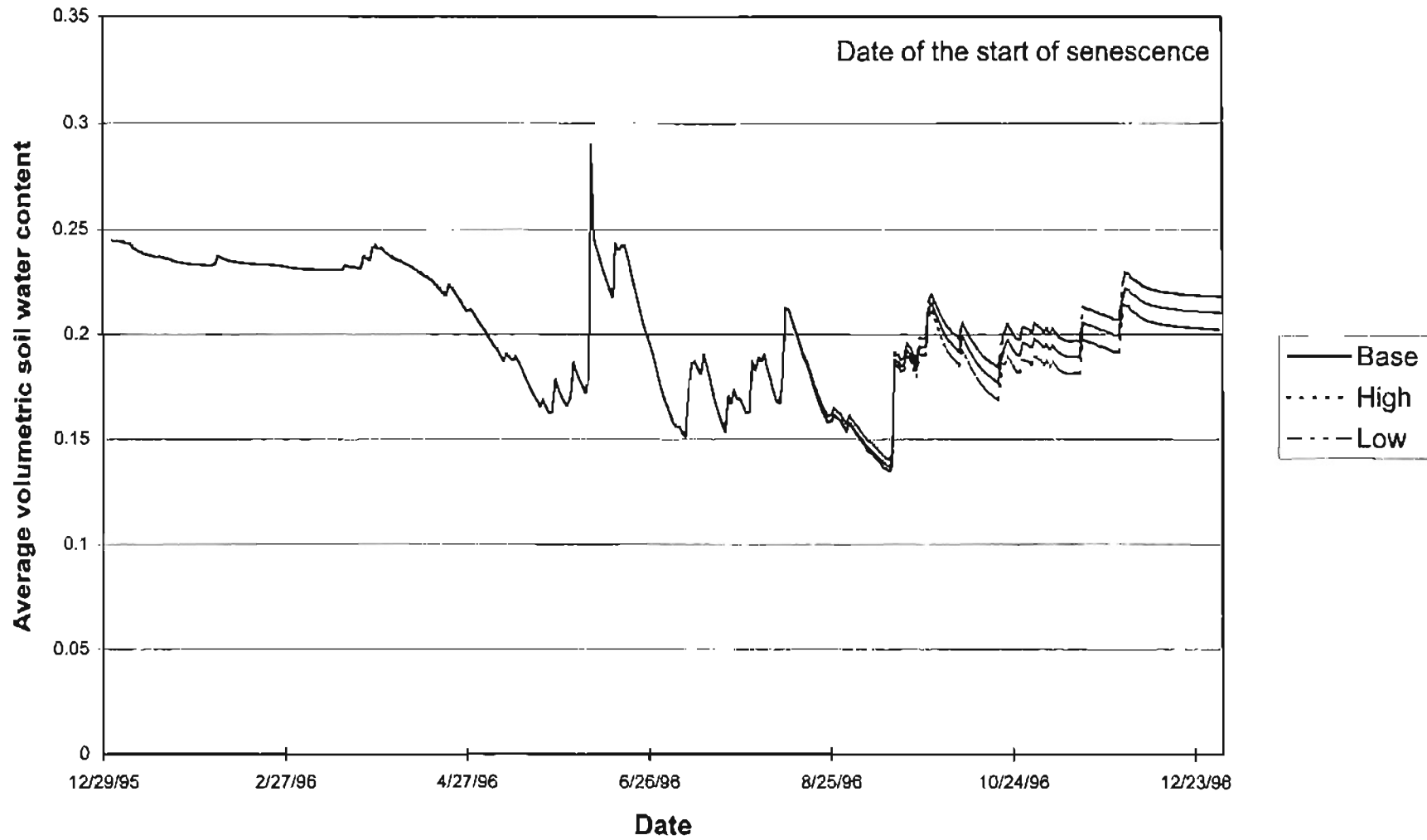


Figure 5.34. Sensitivity of soil water content to the date of the start of senescence (Base = March 26; Low = March 12; High = April 9)

a period of days is not significant. It may cause an offset in the modeled soil water time series which may persist in the absence of rainfall (figure 5.35).

SCS Curve Number

The SCS curve number affects the infiltration. The greater the curve number the less the infiltration. But over the given range, its effect was not found to be very significant. If rainfall is sufficiently high, even a higher curve number causes enough infiltration to raise the soil water content to the porosity and thus there won't be any effect on the modeled soil water content. If the rainfall amount is small, almost zero runoff would be generated as the initial rainfall abstraction is quite high even at the base and the high estimates of the curve number, 74 and 85 respectively.

Summary

Field capacity, wilting point, the minimum stomatal resistance and the maximum LAI were found to be the most important parameters in terms of model sensitivity. Field capacity and wilting point determine the high and the low limit of the soil water content in the root zone depth. The minimum stomatal resistance is the most critical parameter in the ET component of the model. Thus its effect on the overall performance of the model is significant. The maximum LAI affects the ET mostly when LAI is low, by changing the slope of the early-growth and the senescence slope. At higher values its effect is insignificant as effective LAI changes very little at higher LAI values due to the shelter factor.

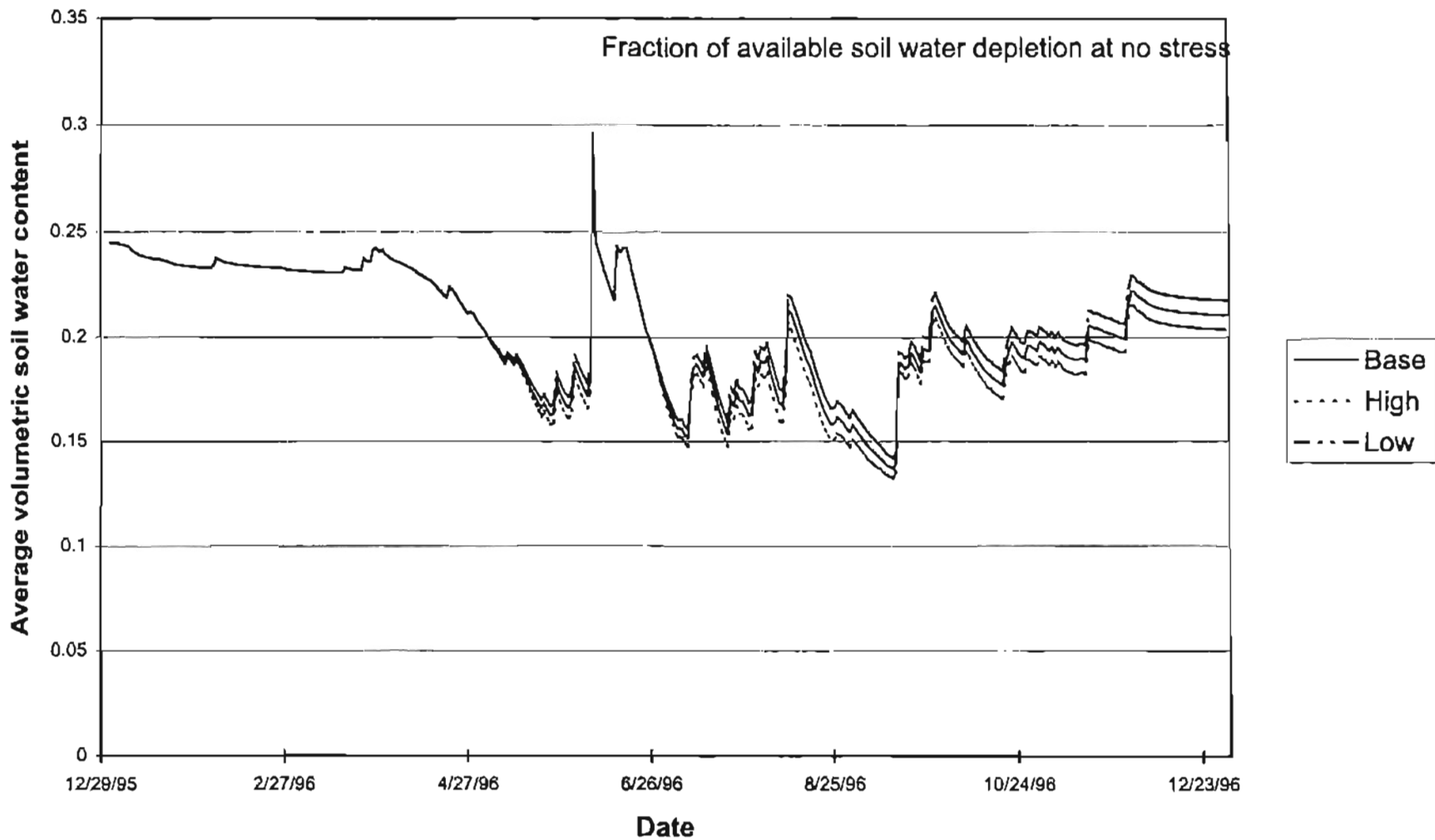


Figure 5.35. Sensitivity of soil water content to the fraction of available soil water content at no stress (Base = 0.5; Low = 0.4; High = 0.6)

Root depth is important under water stressed conditions. Under such a condition, deeper roots allow access to a larger reservoir of soil water and therefore maintain higher ET rates.

Chapter 6

SUMMARY AND CONCLUSIONS

Evapotranspiration (ET) is a major component of both the energy balance and the hydrologic balance at the earth's surface. An important factor that directly affects ET is soil moisture content in the root zone. Knowledge of the soil moisture content is also important for several other reasons, such as irrigation scheduling and runoff prediction.

Various instruments are available on the market to measure soil water content at discrete points. But installing such instruments at every desirable point is neither cost effective nor feasible. The need for estimates of soil moisture content has resulted in the development of estimation methods based on meteorological variables, soil hydraulic properties and land cover characteristics. Often soil water balance models are applied to estimate the moisture content of the root zone. ET algorithms are important components of such models.

The main objective of this research was to develop and test a point-based water balance model to predict, on a daily basis, soil moisture content in the root zone for non-irrigated grass in Oklahoma. In an attempt to develop a fairly generic model, site-specific calibration was minimized. The model was developed and tested at eight Mesonet sites across the State of Oklahoma and representing various soil textures and vegetation types. These sites are Apache, Boise City, Broken Bow, Goodwell, Hollis, Marena, Miami and Stillwater.

The soil-water balance is comprised of precipitation and irrigation inputs balanced by water outflows in the form of evapotranspiration, runoff, and deep drainage. Many soil water balance models have been developed to predict soil water content in the root zone. These models differ in their component methods and data requirements. Most of the hydrology models do not apply an elaborate approach to modeling the vegetation component of the soil-plant-atmosphere system. Recently attempts have been made to give more attention to the vegetation component of the soil water balance models. The component methods of soil water balance models vary from physical methods to fairly empirical approaches depending upon the data availability.

This model was developed so as to take advantage of the weather, soil and vegetation databases available in Oklahoma. This led to a fairly physically-based model, though some components are empirical. The model is process based and uses a daily time-step. This model divides the soil-plant-atmosphere system into components and processes that can be treated individually. The model is modular so that any component can be changed without affecting the other components. Components of this model include plant transpiration, soil evaporation, infiltration, drainage and plant growth.

Transpiration is estimated using the Penman-Monteith combination equation. This equation accounts for the energy required to sustain evapotranspiration and a mechanism required to remove the vapor. Canopy resistance is obtained by scaling up the stomatal resistance.

Soil evaporation is estimated as the minimum of atmospheric demand and the desorption volume. Desorption volume is the amount of water which the soil would

release to a non-restricting atmosphere over a period of time. Atmospheric demand is estimated as 1.15 times grass reference evapotranspiration.

The soil evaporation is estimated as occurring from a bare soil whereas plant transpiration is estimated assuming full cover. Then these two are mixed based on the fractional transpiration factor to produce a weighted average of the evapotranspiration. The fractional transpiration factor is a function of ground cover and is assumed to be highly correlated with the LAI. The maximum fractional transpiration factor is limited to a value less than 1.0 to account for the fact that soil evaporation is never suppressed to zero. At all the study sites, the minimum fractional transpiration factor was set greater than zero to account for the fact that dormant grasses cover a part of the ground even though they may not transpire.

Soil evaporation is taken only from the uppermost soil layer, which is 15 cm deep. Water for transpiration is extracted from the root zone using an exponential weighting function, which varies from a maximum at the surface to a minimum at the bottom of the root zone. The extraction coefficient for each layer also depends on the water content of that layer, as it affects the contribution of individual layers towards total transpiration. A layer at a moisture content near wilting point will release less water and thus more water will be taken from the other layers.

To keep track of vegetation parameters a simple plant growth model is used. Outputs from this model are canopy height, leaf area index, fractional transpiration factor and root depth. The fractional transpiration factor is correlated with the LAI and can be visualized as the fraction of the ground covered by the canopy. For grass, it is assumed that roots do not change through time once they have been established. But canopy height

and leaf area index do change through time. A linear model is used to increase the leaf area index from the start of the growing period to the time it reaches a maximum. The leaf area index is assumed to remain constant until the start of the senescence period. Then it is linearly decreased from the start of the decline to the start of the dormant period. Leaf area index is assumed to remain constant, at its minimum value, throughout the dormant period. Canopy height and fractional transpiration factor are assumed to change in the same fashion as LAI. But fractional transpiration factor often reaches its maximum value before LAI. Fractional transpiration factor is assumed to reach full cover at a LAI of 3.0. In case of grazing and/or mowing a cut-off LAI value is assumed. It is the maximum LAI which would be present throughout the growing period.

The model was initially developed and calibrated at the following four sites: Apache, Marena, Goodwell and Stillwater. Data for January 1996 through September 1997 were used for the development and the calibration of the model. The weather data were obtained from the Oklahoma Mesonet. The vegetation parameters were estimated based on field observations or literature. The estimates for the soil water parameters were obtained from the literature based on the texture of the various layers. The main source of the soil texture information was data collected by soil scientists with the USDA NRCS. Other sources were results from particle-size analyses of samples taken at the sites, and the STATSGO database.

Lysimeter observations at Apache, Marena and Goodwell and soil water estimates at all four sites were used to help in developing and calibrating the model. The soil water estimates were obtained from the thermal dissipation sensors installed at many of the Mesonet sites including all the eight study sites. These sensors were installed at four

depths: 5 cm, 25 cm, 60 cm and 75 cm. They report the difference in the temperature, ΔT , of the sensor before and after heating. The ΔT values were converted into soil water potential and then into volumetric soil water content. So these are indeed estimates of soil water and not direct observations.

The calibrated model performed well in predicting soil water content at all four development sites. The ET predictions matched well with the observed ET at Goodwell. The comparison was not as good at other lysimeter sites (Apache and Marena). At Apache, the model tended to underestimate the ET whereas at Marena it overestimated the ET.

The performance of the model was independently evaluated at the following four sites: Boise City, Broken Bow, Hollis and Miami. Only soil water estimates, from the ΔT observations, were available for the comparison. Only 1997 data were used for the comparison purpose as all these validation sites got thermal sensors installed and working either at the end of 1996 or at the beginning of 1997.

The model did a fairly good job of simulating soil water content in the upper layers. At almost all sites, the simulated soil water matched very well with the field observations at the 5 cm and 25 cm depths. The model's performance is generally not as good at greater depths in the profile (60 cm and 75 cm). The observed profile sometimes shows an increase in the water content at lower depths after heavy rainfall events, which is not captured by the simulated profile.

This result could be attributed to the self-correcting behavior of the upper layers due to the high dynamism involved there. At lower depths, the modeling job becomes more difficult due to the lack of knowledge about the rooting depths of various vegetation

classes and the extraction behavior of their roots. The wrong assumptions regarding the estimates for the soil water parameters of the upper layers, especially the field capacity and the wilting point, might actually affect the results more at the deeper points than at the upper points.

The overall model performance is directly related to the estimates of the parameters input to the model. The model predictions for ET are highly sensitive to the minimum stomatal resistance and those for soil water content are highly sensitive to the field capacity and the wilting point. The field capacity and the wilting point determine the upper and the lower limit of the predicted soil water profile. Among other vegetation parameters, the maximum LAI, the coefficient of the shelter factor and the root depth are the most important. The coefficient of the shelter factor is important as it affects the effective LAI. The soil water parameters, other than the field capacity and the wilting point, can be estimated based on the soil texture as the model is not very sensitive to these parameters.

Recommendations

- 1) The extraction function of this model has the flexibility to accommodate different vegetation types. The coefficient of root distribution function should be calibrated separately for various vegetation classes.
- 2) More work is required to parameterize the fractional transpiration factor.
- 3) The shelter function is arbitrary. It should be improved. Different vegetation types might require different types of shelter functions.

- 4) The model's performance is highly sensitive to the field capacity and the wilting point estimates. Efforts should be made to obtain as accurate estimates as possible.
- 5) Better field measurements of soil water are required to evaluate the performance of this model.
- 6) Further validation of the model is required at additional sites and for longer time periods.
- 7) Further work should be done to quantify the relative sensitivity of model parameters.
- 8) This model can be integrated with a geographic information system and used to predict ET and soil water over larger areas.

REFERENCES

- Acevedo, E., T.C. Hsiao and D.W. Henderson. 1971. Immediate and subsequent growth responses of maize leaves to changes in water status. *Plant Physiol.* 48:631-636.
- Allen, R.G., L.S. Pereira, D. Raes and M. Smith. 1998. *Evapotranspiration and Crop Water Requirements*. FAO-5X (in preparation).
- Allen, R.G., W.O. Pruitt, J.A. Businger, L.J. Fritschen, M.E. Jensen and F.H. Quinn. 1996b. *Evaporation and Transpiration*. *Hydrology Handbook*. ASCE Manuals and Reports on Engineering Practice No. 28:125-249.
- Allen, R.G., M. Smith, L.S. Pereira and A. Pereira. 1994a. An update for the definition of reference evapotranspiration. *ICID Bulletin* 43(2):1-34.
- Allen, R.G., M. Smith, L.S. Pereira and A. Pereira. 1994b. An update for the calculation of Reference Evapotranspiration. *ICID Bulletin* 43(2):35-92.
- Allen R.G., M. Smith, W.O. Pruitt and L.S. Pereira. 1996a. Modifications to the FAO crop coefficient approach. *Evapotranspiration and Irrigation Scheduling*. *Proceed. of the Inter. Conf. ASAE*:124-132.
- Arya, L.M. and J.F. Paris. 1981. A physicoempirical model to predict the soil moisture characteristic from particle-size distribution and bulk density data. *Soil Sci. Soc. Am. J.* 45:1023-1030.
- Black, T.A., W.R. Gardner and G.W. Thurtell. 1969. The prediction of evaporation, drainage, and soil water storage for a bare soil. *Soil. Sci. Soc. Am. Proc.* 33:655-660.
- Blaney, H.F. and W.D. Criddle. 1962. *Determining consumptive use and irrigation water requirements*. USDA-ARS Technical Bulletin 1275. Washington, D.C.: U.S. Government Printing Office.
- Borg, H. and D.W. Grimes. 1986. Depth development of roots with time: an empirical description. *Trans. ASAE* 29(1):194-197.
- Botkin, D.B. 1986. *Remote sensing of the biosphere*. Report of the Committee on Planetary Biology. National Research Council, National Academy of Sciences, New York.
- Brock, F.V., K.C. Crawford, R.L. Elliott, G.W. Cuperus, S.J. Stadler, H.L. Johnson and M.D. Eilts. 1995. The Oklahoma Mesonet: A technical overview. *J. of Atmospheric And Oceanic Technology* 12(1):5-19.
- Brooks, R.H. and A.T. Corey. 1964. Hydraulic properties of porous media. *Hydrol. Pap.* 3. Colo. State Univ., Fort Collins.

- Brutsaert, W.H. 1982. *Evaporation into the Atmosphere*. New York : Reidel.
- Capehart, W.J. and T.N. Carlson. 1994. Estimating near-surface soil moisture availability using a meteorologically driven soil-water profile model. *Journal of Hydrology* 160:1-20.
- Charles-Edwards, D.A., D. Doley and G.M. Rimmington. 1986. *Modeling Plant Growth and Development*. New York: Academic Press.
- Chaubey, I., D.R. Edwards, T.C. Daniel and P.A. Moore. 1994. Modeling nutrient transport in vegetative filter strips, Paper No. 94-2149. St. Joseph, Mich.:ASAE.
- Chow, V.T., D.R. Maidment and L.W. Mays. 1988. *Applied Hydrology*. McGraw-Hill Book Company.
- Chu, S.T. 1978. Infiltration during an unsteady Rain. *Water Resources Res.* 14(3):461-466.
- Clarke, R.T., and M.D. Newson. 1978. Some detailed water balance studies of research catchments. *Proc. R. Soc., Lond. A.* 363:21-42.
- Cowan, I.R. 1968. Mass, heat and momentum exchange between stands of plants and their atmospheric environment. *Q. J. R. Meteorol. Soc.* 94:523-544.
- Cowan, I.R. 1972. Mass and heat transfer in laminar boundary layers. *Agric. Met.* 10:311-329.
- Davis, D.J. 1971. A water balance on a small agricultural watershed, Latah county, Idaho. M.Sc. Thesis, Agricultural Engineering Department, University of Idaho, Moscow. Oct.
- Deardoff, J.W. 1978. Efficient prediction of ground surface temperature and moisture, with inclusion of a layer of vegetation. *J. Geophys. Res.* 83:1889-1903.
- Dennet, M.D., J.R. Milford and J. Elston. 1978. The effect of temperature on the relative leaf growth rate of crops of *Vicia faba* L. *Agril. Meteorol.* 19:505-514.
- Dickinson, R.E., A. Henderson-Sellers, C. Rosenzweig and P.J. Sellers. 1991. Evapotranspiration models with canopy resistance for use in climate models. *Agric. For. Meteorol.* 54:373-388.
- Dickinson, R.E, A. Henderson-Sellers, P.J. Kennedy and M.F. Wilson. 1986. Biosphere-atmosphere transfer scheme (BATS) for the NCAR Community Climate Model. NCAR/TN 275+STR, Natl. Cent. For Atmos. Res., Boulder, Colo.
- Donald, M.B., C.S. Hoveland and G.D. Lacefield. 1991. *Southern Forages*. Atlanta: The Potash and Phosphate Institute (PPI) and the Foundation for Agronomic Research (FAR).

- Duffie, J.A. and W.A. Beckman. 1991. *Solar Engineering of Thermal Processes*. 2nd Ed. New York: John Wiley & Sons.
- Eagleson, P.S. 1970. *Dynamic Hydrology*. New York: McGraw-Hill.
- Eagleson, P.S. 1978a. Climate, soil, and vegetation, 3, a simplified model of soil moisture movement in the liquid phase. *Water Resources Research* 14(5):722-730.
- Eagleson, P.S. 1978b. Climate, soil, and vegetation, 4, the expected value of annual evapotranspiration. *Water Resources Research* 14(5):731-739.
- Elk, K. and J.J. Hanway. 1965. Some factors affecting development and longevity of leaves of corn. *Agron. J.* 57:7-12.
- Elliott, R.L., F.V. Brock, M.L. Stone and S.L. Harp. 1994. Configuration decisions for an automated weather station network. *Applied Engineering in Agriculture* 10(1):45-51.
- Feddes, R.A., P.J. Kowalik and H. Zaradny. 1978. *Simulation of field water use and crop yield*. New York: John Wiley.
- Finnigan, J.J. and M.R. Raupach. 1987. Modern theory of transfer in plant canopies in relation to stomatal characteristics. In *Stomatal Function*, ed. Zeiger et al., 385-429. Stanford: Stanford University Press.
- Fisher, D.K. and R.L. Elliott. 1994. Experiences with remote, low-cost, weighing lysimeters. Paper No. 94-2077. Kansas City, Missouri:ASAE.
- Fisher, D.K. 1995. *Evapotranspiration measurements and resistance parameter estimation under range/pasture conditions in Oklahoma*. Ph.D. Thesis, Oklahoma State University.
- Flerchinger, G.N., K.R. Cooley, C.L. Hanson, M.S. Seyfried and J.R. Wight. 1994. A lumped parameter water balance of a semi-arid watershed. In *International Summer Meeting, Kansas City Missouri*. ASAE, 2950 Niles Rd. St. Joseph, MI. Paper No. 94-2133.
- Furnival, G.M., P.E. Waggoner and W.E. Reifsynder. 1975. Computing the energy budget of a leaf canopy with matrix algebra and numerical integration. *Agric. Meteorol.* 14:405-416.
- Gardner, W.R. 1959. Solution of the flow equation for the drying of soils and other porous media. *Soil Sci. Soc. Amer. Proc.* 23:183-187.
- Goudriaan, J. and H.H. van Laar. 1994. *Modeling potential crop growth processes, current issues in production ecology*. Kluwer Academic Publishers, The Netherlands.

Goudriaan, J. and P.E. Waggoner. 1972. Simulating both aerial microclimate and soil temperature from observations above the foliage canopy. *Neth. J. Agric. Sci.* 20:104-124.

Green, W.H. and G. A. Ampt. 1911. Studies on soil physics, part 1, the flow of air and water through soils. *J. Agric. Sci.* 4(1):1-24.

Hanks, R.J. and R.W. Hill. 1980. Modeling crop response to irrigation in relation to soils, climate and salinity. International Irrigation Information Center, No. 6. New York: Pergamon Press.

Hillel, D. 1971. *Soil and Water: Physical principles and processes*. New York: Academic Press.

Horton, R.E. 1933. The role of infiltration in the hydrologic cycle. *Trans. Am. Geophys. Union* 14:446-460.

Horton, R.E. 1939. Analysis of runoff plot experiments with varying infiltration capacity. *Trans. Am. Geophys. Union* 20:693-711.

Iwata, S., T. Tabuchi and B.P. Warkentin. 1995. *Soil-Water Interactions. Mechanisms and Applications*. New York: Marcel Dekker Inc.

James, G.J. and C.L. Larson. 1976. Modeling infiltration and redistribution of soil water during intermittent application. *Trans. ASAE* vol:482-488.

Jarvis, P.G. 1976. The interpretation of the variations in leaf water potential and stomatal conductances found in canopies in the field. *Philos. Trans. R. Soc. London, Ser. B* 273:593-610.

Jensen, M.E., R.D. Burman and R.G. Allen. 1990. *Evapotranspiration and Irrigation Water Requirements*. ASCE Manuals and Reports on Engineering Practice No. 70.

Jong, R.D. and P. Kabat. 1990. Modeling water balance and grass production. *Soil Science Society of America Journal* 54(6): 1725-1732.

Kanemasu, E.T., L.R. Stone and W.L. Powers. 1976. Evapotranspiration model tested for soybean and sorghum. *Agron. J.* 68:569-572.

Kaufmann, Merrill R. 1982. Leaf conductance as a function of photosynthetic photon flux density and absolute humidity difference from leaf to air. *Plant Physiology* 69:1018-1022.

Korner, C., Judith A.S. and H. Bauer. 1979. Maximum leaf diffusive conductance in vascular plants. *Photosynthetica* 13(1):45-82.

Lhomme, J.P. 1991. The concept of canopy resistance: historical survey and comparison of different approaches. *Agric. For. Meteorol.* 54:227-240.

Lindroth, A. and S. Halldin. 1986. Numerical analysis of pine forest evaporation and surface resistance. *Agric. For. Meteorol.* 38:59-79.

Mascart, P., O. Taconet, J. Pinty and M.B. Mehrez. 1991. Canopy resistance formulation and its effect in mesoscale models: A HAPEX perspective. *Agric. For. Meteorol.* 54:319-351.

Mein, R.G. and C.L. Larson. 1973. Modeling infiltration during a steady rain. *Water Resources Research* 9:384-394.

Milly, P.C.D. 1986. An event based simulation model of moisture and energy fluxes at a bare soil surface. *Water Resources Research* 22(12):1680-1892.

Milthorpe, F.L. and J. Moorby. 1979. *An Introduction to Crop Physiology*, 2nd Ed. Cambridge University Press.

Miyazaki, T. 1993. *Water Flow in Soils*. New York: Marcel Dekker Inc.

Monteith, J.L. 1973. *Principles of Environmental Physics*. Arnold, London.

Monteith, J.L. 1977. Climate and the efficiency of crop production in Britain. *Phil. Trans. Res. Soc. London Ser. B.* 281:277-329.

Monteith, J.L. 1981. Evaporation and surface temperature. *Q. J. R. Meteorol. Soc.* 107:1-27.

Monteith, J.L. 1985. Evaporation from land surfaces: Progress in analysis and prediction since 1948. *Advances in Evapotranspiration*. National Conf., ASAE:4-12. Chicago, IL.

Morrison, J.I.L. and R.M. Gifford. 1983. Stomatal sensitivity to carbon dioxide and humidity. *Plant Physiol.* 71:789-796.

Motoyama, H., D. Kobayahi and K. Kojima. 1986. Water balance and runoff analysis at a small watershed during snow-melting season. In *Proceedings of the Cold Regions Hydrology Symposium*, ed. D.L. Kane, 297-304. Bethesda, MD. American Water Resources Association.

Neilson, R.P. 1995. A model for predicting continental-scale vegetation distribution and water balance. *Ecological Applications* 5(2):362-385.

Noilhan, J. and S. Planton. 1989. A simple parameterization of land surface processes for meteorological models. *Monthly Weather Rev.* 117:536-549.

- Oke, T.R. 1987. *Boundary Layer Climates*. Mathuen and Company, London.
- Pearson, C.J. and R.L. Ison. 1987. *Agronomy of grassland systems*. New York: Cambridge University Press.
- Penman, H.L. 1948. Natural evaporation from open water, bare soil and grass. *Proc. R. Soc. A.* 193:120-145.
- Penman, H.L. 1953. The physical bases of irrigation control. Rep. 13th Int. Hort. Cong. 2:913-923.
- Perrier, A., Etude. 1976. et essai de modelisation des echanges de masse et d'energie au niveau des couverts vegetaux. These de Doctorat d'Etat, University de Paris 6, Pairs.
- Philip, J.R. 1957. The theory of infiltration:1. The infiltration equation and its solution. *Soil Sci.* 83(5):345-357.
- Philip, J.R. 1964. Sources and transfer processes in the air layers occupied by vegetation. *J. Appl. Meteorol.* 3:390-395.
- Philip, J.R. 1969. Theory of infiltration. In *Advances in Hydrosience* vol. 5, ed. V.T. Chow, 215-269.
- Punamia, B.C. and B.B.L. Pande. 1990. *Irrigation and water power engineering*. Delhi, India: Standard publishers distributors.
- Randall, J.C. 1989. Liquid moisture redistribution: hydrologic simulation and spatial variability. *Unsaturated Flow in Hydrologic Modeling*, 127-160.
- Raudkivi, A.J. 1979. *Hydrology*. Oxford: Pergamon Press.
- Rawls, W.J., D.L. Brakensiek and K.E. Saxton. 1982. Estimation of soil properties. *Trans. of ASAE* 25:1316-1320.
- Rawls, W.J., D. Goldman, J.A. Van Mullen and T.J. Ward. 1996. Infiltration. *Hydrology Handbook*. ASCE Manuals and Reports on Engineering Practice No. 28, 75-124.
- Richards, L.A. 1931. Capillary conduction of liquids through porous mediums. *Physics* 1:318-333.
- Ritchie, J.T. 1974. Evaluating irrigation needs for southeastern U.S.A. In *Proc. Irrig. and Drainage Spec. Conf.*, ASCE. 262-273.
- Ritchie, J.T. and B.S. Johnson. 1990. Soil and plant factors affecting evaporation. In *Irrigation of Agricultural Crops*, ed. Stewart, B.A. and D.R. Neilson, 363-390. *Agronomy Series 30*. Am. Soc. Agron.

- Robinson, G.D. 1966. Another look at some problems of the air-sea interface. *Quart. J. R. Met. Soc.*, 92:451-465.
- Rochette, P., E. Pattey, R.L. Desjardins, L.M. Dwyer, D.W. Stewart and P.A. Dube. 1991. Estimation of maize (*Zea mays* L.) canopy conductance by scaling up leaf stomatal conductance. *Agric. For. Meteorol.* 54:241-261.
- Rose, D.A. 1968. Water movement in porous materials: III Evaporation of water from soil. *Brit. J. Appl. Phys.* 1:1779-1791.
- Rosenberg, N.J., B.L. Blad and S.B. Verma. 1983. *Microclimate: The Biological Environment*. New York: John Wiley & Sons.
- Running, S.W. and J.C. Coughlan. 1988. A general model of forest ecosystem processes for regional applications hydrologic balance, canopy gas exchange and primary production processes. *Ecological Modeling* 42:125-154.
- Scanlin, B.R. 1994. Water and heat fluxes in desert soils: 1. Field studies. *Water Resour. Res.* 30(3):709-719.
- Senay, G.B. and R.L. Elliott. 1997. NDVI as a means of characterizing temporal variability in landcover for use in ET modeling. Paper No. 97-2004. Minneapolis, Minnesota: ASAE.
- Sharma, P.B.S., N.H. Rao and K.V.P. Rao. 1980. Calculation of water balance in the crop root zone by computer. *J. of Hydrology* 45:123-131.
- Sharpley, A.N. and J.R. Williams. 1990. EPIC – erosion/productivity impact calculator: model documentation. USDA ARS Technical Bulletin Number 1768.
- Shuttleworth, W.J. 1976. A one-dimensional theoretical description of the vegetation-atmosphere interaction. *Boundary-Layer Meteorol.* 10:273-302.
- Shuttleworth, W.J. and J.S. Wallace. 1985. Evaporation from sparse crops - an energy combination theory. *Q. J. R. Meteorol. Soc.* 111:839-855.
- Skartveit, A. and J.A. Olseth. 1994. Luminous efficacy models and their applications for calculations of photosynthetically active radiation. *Solar energy* 52(5):391-399.
- Soil Conservation Service. 1972. *National Engineering Handbook*, section 4, Hydrology. U.S. D.A., Washington, D.C.
- Srivastava, P., T.A. Costello and D.R. Edwards. 1996. A direct approximate solution to the modified Green-Ampt infiltration equation. *Trans. ASAE* 39(4):1411-1413.

Stannard, David I. 1993. Comparison of Penman-Monteith, Shuttleworth-Wallace, and Modified Priestly-Taylor evapotranspiration models for wild land vegetation in semiarid rangeland. *Water Resources Research* 29(5):1379-1392.

Stewart, J.B. 1988. Modeling surface conductance of pine forest. *Agric. For. Meteorol.* 43:19-35.

Terrel, E.E. 1979. Taxonomy, morphology, and phylogeny. In *Tall Fescue: Agron. Monogr.* 20, ed. R.C. Buckner and L.P. Bush, 31-39. ASA, CSSA and SSSA, Madison, WI.

Thomas, H. and J.L. Stoddart. 1984. Kinetics of leaf growth in *Lolium temulentum* at optimal and chilling temperatures. *Annals of Botany* 53:341-347.

Tollenaar, M., T.B. Daynard and R.B. Hunter. 1979. Effect of temperature on rate of leaf appearance and flowering date of maize. *Crop Sci.* 19:363-366.

USDA NRCS. 1994. State Soil Geographic (STASGO) Data Base. National Soil Survey Center. Miscellaneous Publication Number 1492.

van Genuchten, M.T., F.J. Leij, and S.R. Yates. 1991. The RETC code for quantifying the hydraulic functions of unsaturated soils. EPA/600/2-91/065. Office of Research And Development, U.S. Environmental Protection Agency. Ada, Oklahoma.

Waggoner, P.E. and W.E. Reifsynder. 1968. Simulation of the temperature, humidity and evaporation profiles in a leaf canopy. *J. Appl. Meteorol.* 7:400-409.

Waggoner, P.E., G.M. Furnival and W.E. Reifsynder. 1969. Simulation of the microclimate in a forest. *For. Sci.* 15:37-45.

Watts, W.R. 1972. Leaf extension in *Zea mays*. II. Leaf extension in response to independent variation of the temperature of the apical meristem, of the air around the leaves, and of the root zone. *J. Exp. Bot.* 23:713-721.

Weaver, J.E. 1920. The ecological relations of roots. The Carnegie Institution of Washington, Washington. Publication Number 286.

Wigmosta, M.S., L.W. Vail and D.P. Lettenmaier. 1994. A distributed hydrology-vegetation model for complex terrain. *Water Resources Research* 30(6):1665-1679.

Wittwer, S. (Chair). 1983. Land related global habitability science issues. NASA Tech. Mem. 85841.

Wright, I.R., A.O. Manzi and H.R. da Rocha. 1995. Surface conductance of Amazonian pasture: model application and calibration for canopy climate. *Agric. For. Meteorol.* 75:51-70.

Wyckoff, R.D. and H.G. Botset. 1936. The flow of gas liquid mixtures through unconsolidated sands. *Physics* 7:325-345.

Yin, Z.Y. and G.A. Brook. 1992. Evaporation in the Okefenokee Swamp watershed: a comparison of temperature and water balance methods. *J. Hydrol.* 131:293-312.

Zhang, J., R.L. Elliott and D.L. Ketring. 1993. Root distribution models applied to peanuts. *Trans. ASAE* 36(3):727-734.

APPENDICES

APPENDIX A

Input soil data files

The following files contain the soil information used as model inputs. The first four files contain information for the development sites whereas the last four correspond to the validation sites. The abbreviations are:

NumLayer	Number of soil layers
FC	Field Capacity
WP	Wilting Point
POR	Porosity
KS	Saturated hydraulic conductivity (cm h ⁻¹)
BP	Bubbling Pressure (cm)
m	Pore size distribution index
THETAR	Residual soil moisture
THK	Thickness of the soil layers (m)

AI

SiteName APAC

NumLayer 9

Elevation 440

Latitude 34.91

Longitude -98.29

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.14	0.06	0.437	6.11	8.69	0.474	0.04
0.150	0.23	0.10	0.453	2.59	14.66	0.322	0.04
0.250	0.33	0.18	0.464	0.23	25.89	0.194	0.08
0.100	0.33	0.18	0.464	0.23	25.89	0.194	0.08
0.050	0.33	0.18	0.464	0.23	25.89	0.194	0.08
0.100	0.33	0.18	0.464	0.23	25.89	0.194	0.08
0.100	0.33	0.18	0.464	0.23	25.89	0.194	0.08
0.100	0.33	0.18	0.464	0.23	25.89	0.194	0.08
0.250	0.33	0.18	0.464	0.23	25.89	0.194	0.08

A2

SiteName GOOD

NumLayer 9

Elevation 996

Latitude 36.60

Longitude -101.60

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.30	0.15	0.501	0.68	20.76	0.211	0.02
0.200	0.30	0.15	0.501	0.68	20.76	0.211	0.02
0.080	0.30	0.15	0.501	0.68	20.76	0.211	0.02
0.120	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.050	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.150	0.34	0.19	0.471	0.15	32.56	0.151	0.04

A3

SiteName MARE

NumLayer 9

Elevation 331

Latitude 36.06

Longitude -97.21

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.25	0.13	0.398	0.43	28.08	0.250	0.07
0.150	0.23	0.12	0.463	1.32	11.15	0.220	0.03
0.125	0.23	0.12	0.463	1.32	11.15	0.220	0.03
0.125	0.25	0.13	0.398	0.43	28.08	0.250	0.07
0.100	0.25	0.13	0.398	0.43	28.08	0.250	0.07
0.050	0.25	0.13	0.398	0.43	28.08	0.250	0.07
0.100	0.25	0.13	0.398	0.43	28.08	0.250	0.07
0.200	0.25	0.13	0.398	0.43	28.08	0.250	0.07
0.250	0.25	0.13	0.398	0.43	28.08	0.250	0.07

A4

SiteName STILL

NumLayer 9

Elevation 272

Latitude 36.12

Longitude -97.10

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.31	0.14	0.464	0.23	25.89	0.190	0.08
0.150	0.28	0.15	0.501	0.68	20.76	0.211	0.02
0.150	0.28	0.15	0.501	0.68	20.76	0.211	0.02
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.050	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.200	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.250	0.34	0.19	0.471	0.15	32.56	0.151	0.04

A5

SiteName BOIS

NumLayer 9

Elevation 1267

Latitude 36.69

Longitude -102.50

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.31	0.17	0.464	0.23	25.89	0.190	0.08
0.150	0.31	0.17	0.464	0.23	25.89	0.190	0.08
0.150	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.050	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.050	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.050	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.250	0.34	0.19	0.471	0.15	32.56	0.151	0.04

A6

SiteName BBOW

NumLayer 8

Elevation 113

Latitude 34.01

Longitude -94.61

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.23	0.10	0.453	2.59	14.66	0.322	0.04
0.200	0.23	0.10	0.453	2.59	14.66	0.322	0.04
0.210	0.23	0.10	0.453	2.59	14.66	0.322	0.04
0.090	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.080	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.030	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.140	0.32	0.18	0.430	0.12	29.17	0.168	0.11
0.200	0.32	0.18	0.430	0.12	29.17	0.168	0.11

A7

SiteName HOLL

NumLayer 9

Elevation 498

Latitude 34.69

Longitude -99.83

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.050	0.34	0.19	0.471	0.15	32.56	0.151	0.04
0.100	0.36	0.21	0.479	0.09	34.19	0.127	0.06
0.250	0.36	0.21	0.479	0.09	34.19	0.127	0.06
0.100	0.36	0.21	0.479	0.09	34.19	0.127	0.06
0.050	0.36	0.21	0.479	0.09	34.19	0.127	0.06
0.100	0.36	0.21	0.479	0.09	34.19	0.127	0.06
0.100	0.36	0.21	0.479	0.09	34.19	0.127	0.06
0.200	0.36	0.21	0.479	0.09	34.19	0.127	0.06

A8

SiteName MIAM

NumLayer 9

Elevation 248

Latitude 36.89

Longitude -94.84

THIK	FC	WP	POR	KS	BP	m	THETAR
0.150	0.30	0.15	0.501	0.68	20.76	0.211	0.02
0.200	0.30	0.15	0.501	0.68	20.76	0.211	0.02
0.150	0.30	0.15	0.501	0.68	20.76	0.211	0.02
0.050	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.100	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.050	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.100	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.100	0.36	0.21	0.475	0.06	37.30	0.131	0.09
0.200	0.36	0.21	0.475	0.06	37.30	0.131	0.09

APPENDIX B

Vegetation parameter input file

SITE	α	Γ_{smin}	Γ_{smax}	h_{mx}	Z_{root}	LAI_{mx}	Fr_{mx}	LAI_{cutoff}	D_{sg}	D_{mx}	D_{ss}	D_{sd}
STILL	0.18	166	5000	0.50	1.0	6.0	0.90	2.0	03/26	07/30	08/31	11/12
MARE	0.18	166	5000	0.80	1.0	5.0	0.90	5.0	03/26	07/23	08/20	11/04
BBOW	0.18	166	5000	1.00	0.8	6.0	0.90	6.0	03/26	07/23	08/25	11/27
BOIS	0.18	166	5000	0.35	0.8	3.0	0.90	3.0	05/09	08/03	09/09	11/11
HOLL	0.18	166	5000	0.80	1.0	6.0	0.90	6.0	03/26	07/23	08/25	11/27
MIAM	0.18	166	5000	1.00	0.8	6.0	0.90	6.0	03/26	07/23	08/25	11/27
APAC	0.18	166	5000	0.50	1.0	6.4	0.90	6.4	03/26	07/30	08/31	11/12
GOOD	0.18	166	5000	0.35	0.8	3.0	0.90	3.0	05/09	08/03	09/09	11/11

APPENDIX – C

Calibration parameters for 229L sensors

Sensor and soil parameters required to estimate soil water content from the sensor response, ΔT .

Common Sensor Coefficients These coefficients are constant for all sensors:

$$\Delta T_d = 4.00 \text{ C}$$

$$\Delta T_w = 1.45 \text{ C}$$

$$a = -0.01 \text{ kPa}^{-1}$$

$$n = 0.77$$

Site Specific Parameters

C1 - Apache

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
Apache	5	0.396	0.024	26.928	3.230	0.997	-0.51
Apache	25	0.415	0.050	66.064	1.871	0.988	-0.058
Apache	60	0.434	0.119	31.810	1.667	0.897	0.397
Apache	75	0.434	0.087	156.882	1.298	0.936	0.083

C2 - Broken Bow

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
BBOW	5	0.453	0.000	34.960	1.364	0.96	0.107
BBOW	25	0.453	0.000	20.224	1.375	1.005	0.299
BBOW	60	0.415	0.000	7.173	1.396	0.932	0.377
BBOW	75	0.434	0.000	6.848	1.344	0.925	0.367

C3 - Boise City

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
BOIS	5	0.471	0.122	229.953	1.268	0.931	-0.114
BOIS	25	0.472	0.112	122.482	1.243	1.131	-0.384
BOIS	60	0.453	0.102	70.954	1.228	1.131	-0.408
BOIS	75	0.453	0.096	61.022	1.245	1.001	-0.058

C4 – Goodwell

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
GOOD	5	0.453	0.079	53.571	1.446	1.654	-1.57
GOOD	25	0.453	0.101	52.295	1.433	1.064	-0.382
GOOD	60	0.434	0.096	35.471	1.393	0.272	-0.547
GOOD	75	0.434	0.085	818.626	1.174	1.111	-0.353

C5 - Hollis

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
HOLL	5	0.509	0.197	14.778	1.814	0.933	0.19
HOLL	25	0.528	0.209	40.222	1.388	1.057	-0.106
HOLL	60	0.472	0.110	41.491	1.192	1.008	-0.296
HOLL	75	0.491	0.000	51.921	1.132	0.992	-0.035

C6 – Marena

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
MARE	5	0.472	0.080	122.193	1.439	1.041	0.114
MARE	25	0.453	0.085	47.446	1.625	1.044	-0.401
MARE	60	0.434	0.119	48.507	1.640	0.955	0.114
MARE	75	0.434	0.121	40.556	1.857	0.938	0.296

C7 – Miami

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
MIAM	5	0.472	0.038	5.946	2.129	1.102	0.238
MIAM	25	0.472	0.023	9.291	1.753	0.948	0.389
MIAM	60	0.472	0.134	15.051	1.317	0.891	0.413
MIAM	75	0.472	0.140	25.589	1.314	1.068	-0.047

C8 - Stillwater

SITE	DEPTH (cm)	θ_s	θ_r	x (1/bar)	y	m	b
Stillwater	5	0.490	0.087	17.530	1.338	0.991	-0.251
Stillwater	25	0.453	0.088	23.509	1.637	0.998	0.137
Stillwater	60	0.415	0.093	26.112	1.574	1.021	0.098
Stillwater	75	0.415	0.085	197.139	1.329	0.941	0.264

APPENDIX D

Source code

```

////////////////////////////////////
////////

```

```

//////// This class contains land cover and soil information for any
//////// particular cell. It contains methods to calculate ET based on trans-
//////// piration, soil evaporation , see modules cropet and SoilEvap.
//////// It resets the soil water content in each layer based on ET
//////// and Infiltration, see Infiltration module for calculations on
//////// infiltraion.
////////
////////////////////////////////////

```

```

#ifndef __CELL_H__
#define __CELL_H__
#include "soilayer.h"
#include "landcovr.h"
class Cell
{
private: LandCoverPara Crop;
        int NumLayers, NumOfActiveLayers; //Active Layers mean layers which contain roots
        SoilLayer *layer;
        //Latitude and longitude are in radians
        float CN, Elevation, Latitude, Longitude, Drain, ET, Se; //this value will
be set based on soil type and land cover
        float *AC;
        float ThetaT, ThetaT4RG; //threshold for soil moisture stress
        char SiteName[5];

public: Cell();
        Cell(int); // constructor will initialize the number of layers
        //Copying constructor
        Cell(const Cell&);
        void SetNumOfActiveLayers(void);
        void SetCN(void)
        {
//set cn on the basis of SoilPara of first layer and land cover
            CN = (float)80;
        };
        void SetCN(float cn) {CN = cn;};
        void SetThetaT(void);
        void SetThetaT4RG(void);
        void SetPosition(float, float, float); //Elev, lat, long
        float DrainagePerHour(int);
        void SetDrain(float tDrain){Drain = tDrain;};
        void DrainWater();
        void AddWaterToNextLayer(int, float);
        void InfiltrateWater(float);
        int CalcSoilMoist(float, float, float); //arguements are CROPET, SE &
Intercept
        int SetLayersPara(char *); //it will read the soil properties from the file
layer.dat

        float GetCN(void) {return CN;};
        float GetTW(void); //return total soil water
        float GetTawForEt(void); //return total available water in active layers
        float MoveWaterUp(int, float); // Moves water in the upper layer
        float GetDepth(void); // return total depth in mm
        float GetTheta(void); //return average theta

```

```

float GetThetaT(void){ return ThetaT;};
float GetThetaT4RG(void){ return ThetaT4RG;};
float GetThetaXcLayer1(void);
float GetFC(void); //return average field capacity
float GetWP(void); //return average wilting point
float GetDrain(void){return Drain;};
float GetET(void){return ET;};
float GetElevation(void){return Elevation;};
float GetLatitude(void){return Latitude;};
float GetSe(void) {return Se;};
int GetNumLayers(void) {return NumLayers;};
//get the SoilLayer objects by reference to access their members
SoilLayer& Getlayer(int);
LandCoverPara& GetCrop(void){return Crop;};
//Assignment operator
const Cell& operator=(const Cell&);
float GetAc1(void){ return AC[0];};
float GetAc2(void){return AC[1];};
float GetAc3(void){return AC[2];};
~Cell(); //destructor
};
#endif

////////////////////////////////////
////////
//////// This module calculates plant transpiration assuming full cover
//////// For calculations, see header and source files for pet which is
//////// used to estimate ref. ET
//////// This class contains equations to estimate resistances.
////////
////////////////////////////////////
#include "pet.h"
#include "cell.h"
// this is derived class, pet is the base class
//it inherits all other functions except calc_ra()
//and calc_rc()

class cropet : public pet
{
protected : float rsmín, rsmáx, rs, PAR, THETA, THETAT, FC, WP;
float RPC, RP, cd, LAI, Ps, FPAR, FTHETA, FVPD, CH;
public: cropet(Cell &);
void SetSmValues(Cell &); //Sets soil moisture values
virtual void SetValue(MetValue&, CropPara&, float);
virtual void calc_ra(); //virtual as it will overwrite calc_ra() in pet
virtual void calc_rc(); // virtual as it will overwrite calc_rc() in pet
float GetTheta(void) {return THETA;};
float GetWp(void) {return WP; };
};

////////////////////////////////////
////////
//////// This class contains information about the land cover. It
//////// is plant growth module and estimates growth in plant.
//////// on a daily basis.
////////
////////////////////////////////////

```

```

#ifndef __LANDCOVR__
#define __LANDCOVR__
#ifndef __FSTREAM_H__
#include "fstream.h"
#endif
#include "Mdate.h"

struct CropPara
{
    float CH, RD, LAI, FR;
};

class LandCoverPara
{
private: int land_cover;
        int SmsCounter;
        float ALPHA, BEETA, RSMIN, RSMAX, CHMX, RDMX, LAIMX, FRMX, LAIatMFR;
        float MinLai, CUTOFF_LAI, MinFr, MinCh;
        float Slope1, Slope2;
        char Date1[6], Date2[6], Date3[6], Date4[6];
        BOOL Season; // true for active and false for dormant
        CropPara CurrentCropPara;
        CIDate pDate[4];

public: //ifstream CropIn;
        int SetLandCoverPara(char *, char*); //arguements are file name and sitename
        void SetMinFr(float tMinFr){MinFr = tMinFr;};
        void SetMinCh(float tMinCh){MinCh = tMinCh;};
        void SetBeeta(float tBeeta){BEETA = tBeeta;};
        float GetAlpha(void){ return ALPHA;};
        float GetBeeta(void){ return BEETA;};

        float GetRsMin(void){ return RSMIN; };
        float GetRsMax(void){ return RSMAX; };
        float GetChMx(void) { return CHMX; };
        float GetRdMx(void) { return RDMX; };
        float GetLaiMx(void){ return LAIMX;};
        float GetFrMx(void) { return FRMX; };
        float GetMinFr(void){ return MinFr;};
        BOOL GetSeason(void) { return Season; };
        CropPara& GetCropPara(void){return CurrentCropPara;};

        void SetCropPara(float, float, float, float, float);
        void SetpDate(int, char*); //first argument is date # and 2nd is date
        //call this function only after setting dates
        void CalcSlopes(void);
        const char* GetPDate(int n){return pDate[n-1].GetDate();};
        void GrowCrop(CIDate);
        void CheckSms(float, float); //check for soil moisture stress
        void GrowRoot(void);
        //These two values are changed based on current LAI
        float ChangeFr(float);
        float ChangeCh(float);
};

#endif
////////////////////////////////////
// CTheta.h

```

```

////////////////////////////////////
#ifndef __CTHETA_H
#define __CTHETA_H
#include <iostream.h>
class CTheta
{
private: char date[10];
        float *Theta;
        int NumLayers;
public: CTheta(int); //constructor will allocate memory for Theta
        CTheta(); //default counstructor
        //copy constructor
        CTheta(const CTheta&);
        void SetTheta(int, float);
        float GetTheta(int n){return Theta[n-1]; };
        void SetDate(const char *);
        friend ostream& operator << (ostream&, CTheta&);
        CTheta& operator=(const CTheta&);
};
#endif
// watbalView.h : interface of the CWatbalView class
//
////////////////////////////////////
#include <strstrea.h>
#include "metvalue.h"
#include "Cell.h"
#include "watbalDoc.h"
class CWatbalView : public CScrollView
{
protected:
    char *WeatherFile;
    char *CropFile;
    char *LayerFile;
    char SiteName[5];
    int NumOfLayers, NumOfSimulation;
    char Date1[11], Date2[11], Date3[11], Date4[11];
    char *InitTheta;
    COLORREF Color[10];
    //File flags
    BOOL wfFlag, cfFlag, lfFlag;

    //Run flag, true if run is over
    BOOL RunIsOver;
    enum {ViewText, ViewChart} eViewType;
    // create from serialization only
    CWatbalView();
    DECLARE_DYNCREATE(CWatbalView)
// Attributes
public:
    CWatbalDoc* GetDocument();
    void FileInput(char *);
    int GetRunInfo(Cell&);
    int GetNumOfLayers(void){return NumOfLayers;};
    int GetNumOfSimulations(void){return NumOfSimulation;};
    void RunTheModel();
    void TextOut(CDC*);

```



```

        void ChartOut(CDC*);
        void ChangeColor(int); //argument is the index of color
// Operations
public:
// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CWatbalView)
    public:
    virtual void OnDraw(CDC*); // overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    protected:
    virtual void OnInitialUpdate(); // called first time after construct
    //}AFX_VIRTUAL
// Implementation
public:
    virtual ~CWatbalView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
protected:
// Generated message map functions
protected:
    //{AFX_MSG(CWatbalView)
    afx_msg void OnInputWeather();
    afx_msg void OnUpdateInputWeather(CCmdUI* pCmdUI);
    afx_msg void OnInputCrop();
    afx_msg void OnUpdateInputCrop(CCmdUI* pCmdUI);
    afx_msg void OnInputLayer();
    afx_msg void OnUpdateInputLayer(CCmdUI* pCmdUI);
    afx_msg void OnRun();
    afx_msg void OnOutChart();
    afx_msg void OnOutText();
    afx_msg void OnUpdateOutChart(CCmdUI* pCmdUI);
    afx_msg void OnUpdateOutText(CCmdUI* pCmdUI);
    afx_msg void OnLButtonDbClick(UINT nFlags, CPoint point);
    //}AFX_MSG
    DECLARE_MESSAGE_MAP()
        friend class CMainFrame;
};
#ifdef _DEBUG // debug version in watbalView.cpp
inline CWatbalDoc* CWatbalView::GetDocument()
    { return (CWatbalDoc*)m_pDocument; }
#endif
//
//
//
//
// This module estimates infiltration using curve number approach.
// CN for any cell is defined in the file cell.
//
//
//
#include "rain.h"
#ifdef _RAIN_H_
#define _RAIN_H_
#endif
#include "cell.h"

```

```

//This class calculates Infiltration based on Curve Number
class Infiltration
{
//THETA is soil moisture in different layers
//SMD is soil moisture deficit in different layers
//D is depth of different layers
//D5RAIN is an object of Rain_Array type to hold 5 days' rainfall
//RAIN5 is the sum of previous 5 days rainfall
//AMC is antecedent moisture condition
//SMX is maximum rainfall abstraction under dry condition
//S is maximum rainfall abstraction under prevailing moisture condition
//F is cumulative infiltration
private: float CN, THETAS, THETAA, P, RAIN5, PMAXINT, PINT, FR, LAI, SMX, S,
        IRA, RO, INFLOW, F, SMDB, SMDB, DA, DB;
        RainArray D5RAIN;
        int AMC;
        BOOL Season; // True for active and false for dormant
public: void SetValue(Cell, float);
        void CalcIntercept(void);
        void UpdateRain5(void); //updates the value of rain5 everyday
        void GetAmc(void); //needs Rain5
        void CalcCn(void); //needs amc and cn
        float GetInflow(void);
        float GetRain5(void);
        float GetIntercept(void){return PINT;};
        void CalcAbstraction(void);
        void CalcInfiltration(void);
        float Calculations(void); //calls all other fundtions in right order return
INFLOW
};
#ifdef __LANDCOVR__
#define __LANDCOVR__
#ifdef __FSTREAM_H__
#include "fstream.h"
#endif
#include "Mdate.h"

struct CropPara
{
    float CH, RD, LAI, FR;
};

class LandCoverPara
{
private: int land_cover;
        int SmsCounter;
        float ALPHA, BEETA, RSMIN, RSMAX, CHMX, RDMX, LAIMX, FRMX, LAIatMFR;
        float MinLai, CUTOFF_LAI, MinFr, MinCh;
        float Slope1, Slope2;
        char Date1[6], Date2[6], Date3[6], Date4[6];
        BOOL Season; // true for active and false for dormant
        CropPara CurrentCropPara;
        CIDate pDate[4];
//ifstream CropIn;
public: //LandCoverPara();

```

```

        int SetLandCoverPara(char *, char*); //arguments are file name and sitename
void SetMinFr(float tMinFr){MinFr = tMinFr;};
        void SetMinCh(float tMinCh){MinCh = tMinCh;};
        void SetBeeta(float tBeeta){BEETA = tBeeta;};
        float GetAlpha(void){ return ALPHA;};
        float GetBeeta(void){ return BEETA;};
float GetRsMin(void){ return RSMIN; };
float GetRsMax(void){ return RSMAX; };
float GetChMx(void) { return CHMX; };
float GetRdMx(void) { return RDMX; };
float GetLaiMx(void){ return LAIMX;};
float GetFrMx(void) { return FRMX; };
        float GetMinFr(void){ return MinFr;};
        BOOL GetSeason(void) { return Season; };
        CropPara& GetCropPara(void){return CurrentCropPara;};
void SetCropPara(float, float, float, float, float);
        void SetpDate(int, char*); //first argument is date # and 2nd is date
        //call this function only after setting dates
        void CalcSlopes(void);
        const char* GetPDate(int n){return pDate[n-1].GetDate();};
        void GrowCrop(CIDate);
        void CheckSms(float, float); //check for soil moisture stress
        void GrowRoot(void);
        //These two values are changed based on current LAI
        float ChangeFr(float);
        float ChangeCh(float);
};

#endif
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////// This class is responsible for the input of daily meteorological
////////// variables. It opens the data file, takes values and make unit
////////// conversions.
//////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#ifndef __METVALUE_H__
#define __METVALUE_H__
// these are meteorological values
//these values will be input daily
#ifndef __FSTREAM_H__
#include <fstream.h>
#endif
#include "Mdate.h"

class MetValue
{
private:float TMX, TMN, TA, TMD, PC, P, ux, RS;
        char SiteName[15];
        CIDate mDate;
        ifstream MetIn;
public: MetValue();
        int SetValue(void); // use fin to input values directly and then assign them to variables
        int OpenFile(char *);
        void CloseFile(void);
        CIDate GetDate(void){return mDate;};
        char* GetCarDate(void){return mDate.GetDate();};
};

```

```

float GetTMX(void) {return TMX ;};
float GetTMN (void) {return TMN;};
float GetTA(void) {return TA;};
float GetTMD(void) {return TMD;};
float GetPC(void) {return PC;};
float GetP (void) {return P;};
float Getux(void) {return ux;};
float GetRS(void) {return RS;};

//Assignment operator
//MetValue& operator=(const MetValue&);

};
#endif
////////////////////////////////////
////////////////////////////////////
////////// This module estimates reference ET. All the equations to estimate
////////// tranpiration are outlined here. This module is used in croPET too.
//////////
////////////////////////////////////
////////////////////////////////////

#ifndef _METVALUE_H_
#include "metvalue.h"
#endif
#include "Cell.h"

// This class calculates reference ET using Allen approach
class pet
{
protected: float RS, G, ux, Ta, TaPrev, TMX, TMN, TMD, P, Rn, SVP, SVP_TMX, SVP_TMN,
SVP_TMD;
float Tv, DEL, LH, GAMMA, Cp, AD, VP, ALPHA, VPD, K, x0, x1, du,
zom;
float zov,ra, rc, cpet;
int J;

public:
virtual void SetValue ( MetValue& dailymet1)
{
RS = dailymet1.GetRS(); // solar radiation
ux = dailymet1.Getux(); // wind
Ta = dailymet1.GetTA(); // mean temp
TMX = dailymet1.GetTMX(); // max. temp.
TMN = dailymet1.GetTMN(); // min. dew point temp
TMD = dailymet1.GetTMD(); // mean dew point temp.
P = dailymet1.GetP(); // atmospheric pressure
x0 = (float)2.0; // measurement height in m for wind speed
x1 = (float)1.5; // measurement height in m for humidity and temp
Cp = (float)1.013;
ALPHA = (float)0.23; // Standard definiton for referencet ET assumes albedo of 0.23
J = dailymet1.GetDate().GetJulianDay();
};
void SetTaPrev(float tTaPrev){TaPrev = tTaPrev;}; // Previous day
average temp. for estimating soil heat flux
float calc_svp (float);
virtual void calc_ra ();
virtual void calc_rc ()
{

```

```

    rc = (float)70.0;                //Alien approach assumes a canopy resistance of
    };                               // 70s/m
    void CalcValues(float, float); //arguments are elevation and latitude
    void CalcResist();
    float CalcPet(Cell);
        float GetRn(){return Rn;};
        float GetLh(){return LH;};
        float GetRc(){return rc;};
        float GetRa(){ return ra;};
        float GetVPD(){ return VPD;};

};

//this class will hold the previous 5 days' rainfall
class RainArray
{
private: float rain[5];
public: RainArray();
        void Update(float); //Updates array at each input
        float GetSum();    //Returns the sum of previous 5 days' rainfall
};

// RunDialog.h : header file
//
////////////////////////////////////////////////////////////////////
// RunDialog dialog
#include "LandCovr.h"
class RunDialog : public CDialog
{
// Construction
public:
    RunDialog(CWnd* pParent = NULL); // standard constructor
// Dialog Data
   //{{AFX_DATA(RunDialog)
    enum { IDD = IDD_RUN_INFO };
    CString m_Date1;
    CString m_Date2;
    CString m_Date3;
    CString m_Date4;
    CString m_InitTheta;
    int      m_NumOfSimulation;
    float    m_CurrentFr;
    float    m_CurrentLai;
    float    m_CurrentRd;
    float    m_CurrentCh;
    int      m_NumLayers;
    float    m_CurveNumber;
    CString  m_Year;
    float    m_MinFr;
    float    m_Beeta;
    //}}AFX_DATA
    LandCoverPara DlgCrop;

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(RunDialog)
protected:

```

```

        virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //}}AFX_VIRTUAL
    BOOL OnInitDialog();
// Implementation
protected:
    // Generated message map functions
    //{{AFX_MSG(RunDialog)
    afx_msg void OnChangeCurrentLai();
    afx_msg void OnChangeSimYear();
    afx_msg void OnChangeMinfr();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
////////////////////////////////////////////////////////////////////
//!!!! This class is responsible for estimating soil evaporation.
//!!!! Vairable layera contains values for the soil water parameters for the
//!!!! first layer.
////////////////////////////////////////////////////////////////////
#ifdef SOLLAYER_H
#define SOLLAYER_H
#endif
class SoilEvapo
{
    //thetaa is soil moisture in upper layer of 15cm depth
    // TSE is total soil eveporation after application or precipitaion
    //SE is soil evaporation on day k
    //TDV is total desorptivity volume after application or precipitation
    //DV is desorptivity volume for day k
    //ADSE is atmospheric demand for soil evaporation which is 1.15 times of potential evaporation
private: float THETAA, TSE, SE, TDV, DV, ADSE, rain;
        SoilLayer layera;
        int t;

public: SoilEvapo(SoilLayer); // constructor sets TSE to 12 arbitrarily to have reasonable DV
on first day

        void SetValue(float, float, SoilLayer);
        void SetTse(float); //arguement is fraction cover
        float CalcSp (float); // calculates sorptivity takes theta and SoilPara as
argument

        float CalcDv(float); // calculates and returns desorptivity volume
        float CalcSE (void); //calcualtes soil evaporation
        float GetSE(void); //returns soil evaporation
        float GetDV(void);
        float GetTSE(void);
};
////////////////////////////////////////////////////////////////////
//!!!!
//!!!! This class contains information about a particular soil layer
//!!!! Important information include soil water content and depth of
//!!!! that layer besides estimates of soil water parameters (SoilValues)
//!!!! for that layer.
//!!!!
////////////////////////////////////////////////////////////////////
#ifdef __SOILLAYER__
#define __SOILLAYER__
#include "soil.h"

```

```

class SoilLayer
{
protected: float THETA, DEPTH;
           SoilPara SoilValues;
public:   SoilLayer();
           SoilLayer(float);
           void SetDepth(float); //parameter is depth of the layer in meter
           float GetWater(void);
           float GetAvWater(void); //gives difference between Wilting point and
Current Theta
           float GetWD(void); //gives difference between water at porosity and
current theta, mm
           float GetTHETA(void);
           float GetDEPTH(void) {return DEPTH;}; //DEPTH is in mm
           void SetTHETA(float);
           float CalcExtCoef(void);
           int WiltingPoint();
           //return SoilValues which can be used to access the members SoilPara class
           SoilPara& GetSoilValues(void){return SoilValues;};
};
#endif

```

```

////////////////////////////////////
//////////
////////// This class contains estimates of soil water parameters e.g. porosity,
////////// saturated hydraulic conductivity etc.
//////////
////////////////////////////////////

```

```

#ifndef __SOIL_H_
#define __SOIL_H_
// these values will be constant
// these values depend upon soil texture
class SoilPara
{
private:int soil_type;
           //FC, POR, WP, THETAR, m are unitless
           //BP is in cm
           //KS is in cm/hr
           float FC, KS, POR, BP, WP, THETAR, m;
public: SoilPara(void){};
           void SetValues(float, float, float, float, float, float, float);
           void SetValues(int);
           float GetFC(void) {return FC;};
           float GetKS(void) { return KS;};
           float GetPOR(void) {return POR;};
           float GetBP(void) { return BP;};
           float GetWP (void) {return WP;};
           float Getm (void) {return m;};
           float GetTHETAR(void) {return THETAR;};
};
#endif

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Class definitions and methods
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "Cell.h"
#include <fstream.h>
#include <math.h>

// these are definitions for class Cell

//constructor will initialize the number of layers
Cell :: Cell()
{
    NumLayers=1;
    NumOfActiveLayers=0;
    layer = new SoilLayer;
    Drain = (float)0.0;
}
Cell :: Cell(int n)
{
    NumLayers = n;
    NumOfActiveLayers = 0;
    layer = new SoilLayer[NumLayers];
    for (int i=0; i<NumLayers; i++)
        layer[i].SetTHETA((float)0.20);
    AC = new float[NumLayers];
    Drain = (float)0.0;
};
Cell::Cell(const Cell& tCell)
{
    Crop=tCell.Crop;
    NumLayers=tCell.NumLayers;
    NumOfActiveLayers = tCell.NumOfActiveLayers;
    layer = new SoilLayer[NumLayers];
    AC = new float[NumLayers];
    for(int i=0; i<NumLayers; i++)
    {
        layer[i]=tCell.layer[i];
        AC[i] = tCell.AC[i];
    }
    Drain = tCell.Drain;
    CN=tCell.CN;
    Elevation=tCell.Elevation;
    Latitude=tCell.Latitude;
    Longitude=tCell.Longitude;
}

//destructor
Cell::~Cell()
{
    delete [] layer;
    delete [] AC;
}

void Cell::SetNumOfActiveLayers(void)
{
    float TempDepth = Crop.GetCropPara().RD;
}

```



```

    NumOfActiveLayers = 0;
    //Find out number of active layers
    while(TempDepth >= 0.0 && NumOfActiveLayers < NumLayers)
    {
        TempDepth=layer[NumOfActiveLayers].GetDEPTH();
        NumOfActiveLayers++;
    }
}

int Cell :: SetLayersPara(char *LayerFileName)
{
    char *fline;
    fline = new char [10];
    float data[8];
    ifstream fin(LayerFileName, ios::nocreate);

    if(!fin)
        return 0;

    //Read off sitename
    fin>>fline>>SiteName;
    //read number of layers
    fin>>fline>>NumLayers;
    //read elevation, latitude and longitude
    fin>>fline>>Elevation;
    fin>>fline>>Latitude;
    fin>>fline>>Longitude;
    Latitude=Latitude*(float)3.14/180;
    Longitude=Longitude*(float)3.14/180;

    // read off the characters
    for (int i=0; i<8; i++)
        fin >> fline;
    delete [] fline;

    //read the data
    for (int k=0; k<NumLayers; k++)
    {
        for (int j=0; j<8; j++)
            fin >> data[j];

        layer[k].SetDepth(data[0]);

        //arguements are FC, WP, POR, KS, BP, m, THETAR
        layer[k].GetSoilValues().SetValues(data[1], data[2],
data[3], data[4], data[5], data[6], data[7]);
    }

    //close the file
    fin.close();

    return 1;
}

```

```

    };
void Cell::SetPosition(float tElev, float tLat, float tLong)
{
    Elevation=tElev;
    Latitude=tLat*(float)3.14/180;
    Longitude=tLong*(float)3.14/180;
}
float Cell :: DrainagePerHour(int i)
{
    float WaterTemp, WT;
    float Ki, ScaleFactor, ThetaTemp;
    double Power;
    Power = 2/(layer[i].GetSoilValues().Getm()+3);
    ThetaTemp = ((layer[i].GetTHETA()-
layer[i].GetSoilValues().GetTHETAR())/(layer[i].GetSoilValues().GetPOR()-
layer[i].GetSoilValues().GetTHETAR()));
    ScaleFactor = (float)pow((double)ThetaTemp, Power);
    Ki = (layer[i].GetSoilValues().GetKS()*ScaleFactor; //Ki is in cm/hr
    Ki = Ki*10; //unit changed to mm
    WaterTemp=layer[i].GetWater();
    WT=WaterTemp-((layer[i].GetSoilValues().GetFC()*(layer[i].GetDEPTH()));
    if(Ki>WT)
        Ki=WT;
    WaterTemp=Ki;
    ThetaTemp=WaterTemp/layer[i].GetDEPTH();
    layer[i].SetTHETA(ThetaTemp);
    return Ki;
}

void Cell :: DrainWater(void)
{
    float WaterDrained;
    for(int Hour=0; Hour<24; Hour++)
    {
        for(int x=0; x<NumLayers; x++)
        {
            if(layer[x].GetTHETA() > layer[x].GetSoilValues().GetFC())
                WaterDrained=DrainagePerHour(x);
            else
                WaterDrained=(float)0.0;
            if(x<(NumLayers-1) && WaterDrained >0.0)
                AddWaterToNextLayer(x+1, WaterDrained);
        }
    }
}

//first arguement is layer number into water is to be added
void Cell::AddWaterToNextLayer(int x, float water)
{
    float WaterTemp, ThetaTemp, WD;
    int Layer, LastLayer;
    Layer=x; //index 0 upward
    LastLayer = NumLayers-1; //index 0 upward
    WD = layer[Layer].GetWD();
    if(water > WD)
    {

```

```

layer[Layer].SetTHETA(layer[Layer].GetSoilValues().GetPOR());
water=water - WD;
if(Layer < LastLayer)
    AddWaterToNextLayer(Layer+1, water);
else
    if(Layer == LastLayer)
        Drain = water;
return;
}
else
    if(water <= WD && water > 0.0)
    {
        WaterTemp=layer[Layer].GetWater() + water;
        ThetaTemp=WaterTemp/layer[Layer].GetDEPTH();
        layer[Layer].SetTHETA(ThetaTemp);
        return;
    }
}
float Cell::MoveWaterUp(int x, float Water)
{
    float WaterGiven, WaterTemp, ThetaTemp, WaterInLayer;
    int Layer, LastLayer;
    Layer = x;
    LastLayer = NumLayers-1;
    if(Layer <= LastLayer)
    {
        WaterInLayer = layer[Layer].GetWater();
        //If there is water in this layer,move it up
        if(Water <= WaterInLayer)
        {
            WaterGiven=Water;
            ThetaTemp = (WaterInLayer-WaterGiven)/layer[Layer].GetDEPTH();
            layer[Layer].SetTHETA(ThetaTemp);
        }
        else
        {
            //If this is last layer, move whatever is available
            if(Layer == LastLayer)
            {
                if(!layer[Layer].WiltingPoint())
                {
                    WaterTemp = layer[Layer].GetAvWater();
                    layer[Layer].SetTHETA(layer[Layer].GetSoilValues().GetWP());
                    WaterGiven = WaterTemp;
                }
                else
                    WaterGiven = (float)0.0;
            }
            //Ask next layer for water
            else
            {
                WaterTemp = Water - WaterInLayer;
                WaterTemp = MoveWaterUp(Layer+1, WaterTemp);
                WaterInLayer = WaterInLayer+WaterTemp;
                if(Water <= WaterInLayer)

```

```

        WaterGiven = Water;
    else
        WaterGiven = WaterInLayer;
        WaterInLayer = WaterInLayer - WaterGiven;
        ThetaTemp = WaterInLayer/layer[Layer].GetDEPTH();
        layer[Layer].SetTHETA(ThetaTemp);
    }
}
else
    WaterGiven = (float)0.0;
return WaterGiven;
}

void Cell::InfiltrateWater(float tInflow)
{
    float WaterTemp, ThetaTemp, WD;
    WD = layer[0].GetWD();
    if(tInflow > WD)
    {
        layer[0].SetTHETA(layer[0].GetSoilValues().GetPOR());
        tInflow=tInflow-WD;
        AddWaterToNextLayer(1, tInflow);
        return;
    }
    else
        if(tInflow <= WD && tInflow > 0.0)
        {
            WaterTemp=layer[0].GetWater()+tInflow;
            ThetaTemp=WaterTemp/layer[0].GetDEPTH();
            layer[0].SetTHETA(ThetaTemp);
            return;
        }
}

int Cell :: CalcSoilMoist( float CROPET1, float SE1, float Intercept)
{
    //ACs are availability coefficients for each layers
    //TWs and AWs are total and available water respectively in each
    layer
    float *TW, *TD, *DC, *rz, Ap, ThetaTemp, SumOfExtCoef,
    RootDepth, WaterTemp;
    float FR1 = Crop.GetCropPara().FR;
    float Beeta = Crop.GetBeeta();
    TW = new float [NumLayers];
    TD = new float [NumLayers]; //depth down to the center of the
    layer
    DC = new float [NumLayers]; //denominator of extraction
    function
    rz = new float [NumLayers]; //exponential extraction coefficient
    SumOfExtCoef=(float)0.0;
    RootDepth = Crop.GetCropPara().RD;
    Ap = (float)(Beeta/(1-exp(-(Beeta*RootDepth)))); //Extraction
    parameter alpha

```

```

for (int i=0; i<NumOfActiveLayers; i++)
{
    TW[i] = layer[i].GetWater();
    TD[i]=(float)0.0;
}
//Total depth TD is depth down to the center of the layer
TD[0]=layer[0].GetDEPTH()/2;
for (int x=1; x<NumOfActiveLayers; x++)
{
    for(int y=0; y<x; y++)
        TD[x]+=layer[y].GetDEPTH();
    if(x<NumOfActiveLayers-1)
        TD[x]=TD[x]+(layer[x].GetDEPTH()/2);
    else
        if(x==NumOfActiveLayers-1)
            TD[x]=TD[x] + ((RootDepth-TD[x])/2);
}
for (int z=0; z<NumOfActiveLayers; z++)
    rz[z] = (float)(Ap*exp(-(Beeta*TD[z])));
for(int v=0; v<NumOfActiveLayers; v++)
    DC[v]
layer[v].CalcExtCoef()*layer[v].GetDEPTH()*rz[v];
//This sets the values of ACs

for (int j=0; j<NumOfActiveLayers; j++)
    SumOfExtCoef+=DC[j];

for (int k=0; k<NumOfActiveLayers; k++)
    AC[k] = DC[k]/SumOfExtCoef;
Se = (1-FR1)*SE1;
CROPET1 = FR1*CROPET1;

CROPET1 = CROPET1 - Intercept; //Intercept is not taken from
root
    if(CROPET1 > GetTawForEt())
        CROPET1=GetTawForEt();
    ET=CROPET1+Se+Intercept;
    TW[0] = TW[0] - (Se+(AC[0]*CROPET1));
    ThetaTemp = TW[0]/layer[0].GetDEPTH();
    if (ThetaTemp < layer[0].GetSoilValues().GetTHETAR())
    {
        WaterTemp
(layer[0].GetSoilValues().GetTHETAR()-layer[0].GetTHETA()*layer[0].GetDEPTH());
        WaterTemp = MoveWaterUp(1, WaterTemp);
        ThetaTemp
(WaterTemp+layer[0].GetWater())/layer[0].GetDEPTH();
    }
    layer[0].SetTHETA(ThetaTemp);

// Remove water from layers one by one
//Set to wilting point if smc falls below wilting point

for (int m=1; m<NumOfActiveLayers; m++)
{
    TW[m] = TW[m] - (AC[m]*CROPET1);

```

```

        layer[m].SetTHETA(TW[m]/layer[m].GetDEPTH());
        //f (layer[m].WiltingPoint())
        //{
        //    WaterTemp
        //    =
(layer[m].GetSoilValues().GetWP()-layer[m].GetTHETA())*layer[m].GetDEPTH();
        //    WaterTemp = MoveWaterUp(m+1, WaterTemp);
        //    ThetaTemp
        //    =
(WaterTemp+layer[m].GetWater())/layer[m].GetDEPTH();
        //    layer[m].SetTHETA(ThetaTemp);
        //}
    }
    delete [] TW;
    delete [] TD;
    delete [] DC;

    return 1;
}

SoilLayer& Cell :: Getlayer(int n)
{
    n = n-1;
    return layer[n];
}

float Cell :: GetTW(void)
{
    // Calculates total water in the cell in mm
    float Water = (float)0.0;
    for(int i=0; i<NumLayers; i++)
        Water += layer[i].GetWater();
    return Water;
}

float Cell::GetTawForEt(void)
{
    //Calculate total available water for ET
    float TawForEt = (float)0.0;
    for(int i=0; i<NumOfActiveLayers; i++)
        TawForEt += layer[i].GetAvWater(); //min. value returned by GetAvWater() is 0
    return TawForEt;
}

float Cell :: GetDepth(void)
{
    float Depth = (float)0.0;
    for (int i=0; i<NumLayers; i++)
        Depth += layer[i].GetDEPTH();
    return Depth; // depth is in mm
};

float Cell :: GetTbeta(void)
{
    float Water, Depth, Theta;
    Water=(float)0.0;
    Depth=(float)0.0;
    Theta=(float)0.0;

    for(int i=0; i<NumOfActiveLayers; i++)
    {
        //SMC below WP is taken as WP
    }
}

```

```

//As the the effect must have been felt at WP
if(layer[i].WiltingPoint()
Water +=
(layer[i].GetSoilValues().GetWP()*layer[i].GetDEPTH());
else
Water+=layer[i].GetWater();
}
for (int j=0; j<NumOfActiveLayers; j++)
Depth += layer[j].GetDEPTH();

Theta = Water/Depth;
return Theta;
};

float Cell :: GetThetaXcLayer1(void)
{
float Water, Depth, Theta;
Water=(float)0.0;
Depth=(float)0.0;
Theta=(float)0.0;

for(int i=1; i<NumOfActiveLayers; i++)
Water += layer [i].GetWater();
for (int j=1; j<NumOfActiveLayers; j++)
Depth += layer[j].GetDEPTH();

Theta = Water/Depth;
return Theta;
}

float Cell :: GetFC(void)
{
float CellFc = (float)0.0;
float Depth = (float)0.0;
float *LayerFc;
LayerFc = new float [NumOfActiveLayers];
//Don't Exclude first layer
for (int i=1; i<NumOfActiveLayers; i++)
LayerFc[i] =
(layer[i].GetSoilValues().GetFC()*(layer[i].GetDEPTH()));
for (int j=1; j<NumOfActiveLayers; j++)
CellFc += LayerFc[j];
for (int k=1; k<NumOfActiveLayers; k++)
Depth += layer[k].GetDEPTH();
CellFc = CellFc/Depth;
delete [] LayerFc;
return CellFc;
};

float Cell :: GetWP(void)
{
float CellWp = (float)0.0;
float Depth = (float)0.0;
float *LayerWp;
LayerWp = new float [NumOfActiveLayers];
for (int i=1; i<NumOfActiveLayers; i++)
LayerWp[i] =
layer[i].GetSoilValues().GetWP()*layer[i].GetDEPTH();
for (int j=1; j<NumOfActiveLayers; j++)
CellWp += LayerWp[j];
for (int k=1; k<NumOfActiveLayers; k++)

```

```

        Depth += layer[k].GetDEPTH();
        CellWp = CellWp/Depth;
        delete [] LayerWp;
        return CellWp;
    }

void Cell::SetThetaT()
{
    float FC, WP, Fns;
    FC = GetFC(); // Average values
    WP = GetWP();
    Fns= (float)0.50; //Depletion fraction for no moisture stress
    ThetaT = (1-Fns)*(FC-WP)+WP; //40% of moisture can be extracted without stress
}

void Cell::SetThetaT4RG()
{
    float FC, WP, Fns;
    FC = GetFC(); // Average values
    WP = GetWP();
    Fns= (float)0.70; //Depletion fraction at which roots starts growing
    ThetaT4RG = (1-Fns)*(FC-WP)+WP;
}

const Cell& Cell::operator=(const Cell& tCell)
{
    Crop=tCell.Crop;
    NumLayers=tCell.NumLayers;
    NumOfActiveLayers = tCell.NumOfActiveLayers;
    CN=tCell.CN;
    for(int i=0; i<NumLayers; i++)
        layer[i]=tCell.layer[i];
    return *this;
}

#include "stdafx.h"
#include "cropet.h"
#include <math.h>

cropet :: cropet(Cell &cell1)
{
    rsmn = cell1.GetCrop().GetRsMin();
    rsmax = cell1.GetCrop().GetRsMax();
};

void cropet::SetSmValues(Cell &cell1)
{
    FC = cell1.GetFC(); // Average values
    WP = cell1.GetWP();
    THETA_T = cell1.GetThetaT(); //Threshold soil water above which no stress is felt
}

void cropet :: SetValue (MetValue& dailymet1, CropPara& TValue, float THETA1)
{
    RS = dailymet1.GetRS();
    ux = dailymet1.Getux();
    Ta = dailymet1.GetTA();
    TMX = dailymet1.GetTMX();
    TMN = dailymet1.GetTMN();
    TMD = dailymet1.GetTMD();
    P = dailymet1.GetP();
}

```



```

        x0 = (float)2.0;          // measurement height in m for wind speed
        x1 = (float)1.5;          // measurement height in m for
humidity and temp
        Cp = (float)1.013;
        ALPHA = (float)0.18;     // an average value
        LAI = Tvalue.LAI;
        Ps = (float)((0.1*LAI)+ 1.2); //Shelter factor
        CH = Tvalue.CH;
        THETA = THETA1;
    };

//here displacement height and roughness length have changed
void cropet::calc_ra(void)
    {
        float k;
        zom = (float)0.123*CH;
        zov = (float)0.1*zom;
        du = (float)(2/3)*CH;
        k = (float)0.41;
        ra = (float)((log((x0 - du)/zom))*(log((x1 -
du)/zov))/(ux*k*k));
    };
//canopy resistance is calculated based on Dickinson model
void cropet::calc_rc(void)
    {
        float Ft;
        RPC = (float)8.64; // 100 W/m2 for grasses
        cd = (float)4.0;
        RP = RS/2;
        FPAR =
1/(((rsmmin/rsmax)+(RP/RPC))/(1+(RP/RPC)));
        if(FPAR < 1.0)
            FPAR = (float)1.0;
        if(THETA < THETA1)
        {
            Ft = ((THETA - WP)/(THETA1 - WP));
            if(Ft > 0.0)
                FTHETA = 1/Ft;
            else
                FTHETA = float (-999.0); //if
Theta < WP
        }
        else
            FTHETA = (float)1.0; //if Theta >
Theta1
        if(VPD < cd)
        {
            FVPD = 1/(1- (VPD/cd));
            if(FVPD < 1.0)
                FVPD = (float)1.0;
        }
        else
            FVPD = float(-999.0);
        if(FTHETA == -999.0 || FVPD == -999.0)
            rc = (float)-999.0;
        else
    }

```

```

    {
        rs = rsmIn*(FPAR*FVPD*FTHETA);
        if (rs > rsmax)
            rs = rsmax;
        if(LAI == 0.0)
            rc=(float)-999.0; //Flag for 0.0
    }
}

LAI
else
    rc = Ps*(rs/(2*LAI));
}
}

```

```

//CTheta.cpp
//*****
#include "stdafx.h"
#include "ctheta.h"
CTheta::CTheta(int n)
{
    NumLayers = n;
    Theta = new float[n];
    for (int i=0; i<n; i++)
        SetTheta(i, (float)0.2);
}
CTheta::CTheta(const CTheta& tTheta)
{
    strcpy(date, tTheta.date);
    NumLayers = tTheta.NumLayers;
    Theta = new float[NumLayers];
    for(int i=0; i<NumLayers; i++)
        Theta[i]=tTheta.Theta[i];
}

void CTheta::SetTheta(int n, float th)
{
    Theta[n] = th;
}

void CTheta::SetDate(const char *tdate)
{
    strcpy(date, tdate);
};

ostream& operator<< (ostream& os, CTheta& th)
{
    //first write today's date
    os<<tth.date;
    for (int i=0; i<tth.NumLayers; i++)
        os<<'\t'<<tth.Theta[i];
    os<<endl;
    return os;
}

CTheta& CTheta::operator=(const CTheta& tTheta)
{
    strcpy(date, tTheta.date);
    NumLayers=tTheta.NumLayers;
    Theta=new float [NumLayers];
    for(int i=0; i<NumLayers; i++)

```

```

        Theta[i]=tTheta.Theta[i];
    return *this;
}
#include "stdafx.h"
#include "cell.h"
#include "infiltr.h"

void Infiltration::SetValue( Cell cell1, float RAIN)
{
    CN = cell1.GetCN();
    THETAS =
cell1.Getlayer(1).GetSoilValues().GetPOR();
    FR = cell1.GetCrop().GetCropPara().FR;
    LAI =
cell1.GetCrop().GetCropPara().LAI;
    THETAA =
cell1.Getlayer(1).GetTHETA();
    P = RAIN;
    Season = cell1.GetCrop().GetSeason();
};

void Infiltration::CalcIntercept(void)
{
    //for grass (Wigmosta, 1994)
    PINT = FR*(float)0.1*LAI;
};

void Infiltration::UpdateRain5(void)
{
    RAIN5 = (float)0;
    RAIN5 = D5RAIN.GetSum();
    D5RAIN.Update(P);
}

void Infiltration::GetAmc(void)
{
    if (Season == FALSE) //season is dormant
    {
        if (RAIN5 < 12.7) //0.5 inches = 12.7 mm
        {
            AMC = 1;
            return;
        }
        else
            if (RAIN5 < 27.94) //1.1 inches =
27.94mm
            {
                AMC = 2;
                return;
            }
            else
                AMC = 3;
    }
    else //season is active
    {
        if (RAIN5 < 35.56)
        {
            AMC = 1;
            return;
        }
    }
}

```

```

    }
    else
    if (RAINS < 53.34)
    {
        AMC = 2;
        return;
    }
    else
    AMC = 3;
}
};

void Infiltration::CalcCn(void)
{
    if (AMC == 1)
    CN =(float) ((4.2*CN)/(10-(0.058*CN)));
    else
    if (AMC == 2)
    CN = CN;
    else
    if (AMC == 3)
    CN =
(float)((23*CN)/(10+(0.13*CN)));
};

void Infiltration::CalcAbstraction(void)
{
    SMX = ((25400/CN)-254); //SMX is in
mm
S = SMX;
//S = SMX*(THEETA S -
THEETA A)/THEETA S);
};

void Infiltration::CalcInfiltration(void)
{
    if (S <= 0.0)
    S = (float) 0.0;
    IRA =(float) 0.2*S;
};

if (P > IRA)
{
    RO = (float)((P - IRA)*(P-IRA))/(P +
(0.8*S));
    INFLOW = (P - RO - PINT);
}
else
INFLOW = (P - PINT);
if(INFLOW<0.0)
    INFLOW=(float)0.0;
};

float Infiltration::Calculations(void)
{
    CalcIntercept();
    GetAmc();
    CalcCn();
    CalcAbstraction();
    CalcInfiltration();
    return INFLOW;
};

```

```

float Infiltration::GetInflow(void)
{
    return INFLOW;
};

float Infiltration::GetRain5(void)
{
    return RAIN5 ;
};

#include "stdafx.h"
#include <iostream.h>
#include <fstream.h>
#include "landcovr.h"
int LandCoverPara::SetLandCoverPara(char *CropFileName, char* tSite)
{
    char line[100];
    char Site[5];
    LAIatMFR = (float)3.0;
    ifstream fin(CropFileName, ios::nocreate);
    if (!fin)
        return 0;
    fin.getline(line, 100);
    do
    {
        fin>>Site>>ALPHA>>RSMIN>>RSMAX>>CHMX>>RDMX>>LAIMX;
        fin>>FRMX>>CUTOFF_LAI>>Date1>>Date2>>Date3>>Date4;
    } while((strcmp(Site, tSite)!=0) && !fin.eof());
    fin.close();
    if(LAIatMFR > LAIMX)
        LAIatMFR = LAIMX;
    RDMX = RDMX*1000; //depths are in mm
    MinFr = (float)0.5;
    MinLai = (float)0.0;
    MinCh = (float)0.0;
    if(strcmp(Site,tSite)!=0)
        return -1;
    else
        return 1;
}

void LandCoverPara::SetCropPara( float tRD, float tLAI, float tCH, float tFR, float tMinFr)
{
    CurrentCropPara.CH=tCH;
    MinCb = tCH;
    CurrentCropPara.RD=tRD*1000; // all depths are in mm
    CurrentCropPara.LAI=tLAI;
    CurrentCropPara.FR=tFR;
    MinFr = tMinFr;
    MinLai=tLAI;
    Season = FALSE;
}

void LandCoverPara::SetpDate(int n, char* tDate)
{
    n=1;
}

```

```

        pDate[n].SetDate(tDate);
    }

//call this function only after you have set the four dates
void LandCoverPara::CalcSlopes(void)
{
    Slope1 = (LAIMX-MinLai)/(pDate[1]-pDate[0]);
    Slope2 = (LAIMX-MinLai)/(pDate[3]-pDate[2]);

    //It brings down the third point on LAI curve
    //Senescence is delayed
    if(CUTOFF_LAI != LAIMX)
    {
        int temp = (int)((LAIMX-CUTOFF_LAI)/Slope2);
        pDate[2]=pDate[2]+temp;
    }
}

void LandCoverPara::GrowCrop(CIDate tdate)
{
    // Dormant season
    if(tdate < pDate[0] || tdate == pDate[0])
    {
        Season = FALSE;
        return;
    }
    //First slope
    if(tdate > pDate[0] && (tdate < pDate[1] || tdate == pDate[1]))
    {
        CurrentCropPara.LAI+=Slope1;

        //Sets LAI to cutoff value in case of mowing/grazing
        if(CurrentCropPara.LAI > CUTOFF_LAI)
            CurrentCropPara.LAI = CUTOFF_LAI;
        CurrentCropPara.FR=ChangeFr(CurrentCropPara.LAI);
        CurrentCropPara.CH=ChangeCh(CurrentCropPara.LAI);
        //GrowRoot();
        Season = TRUE;
        return;
    }
    //Constant line
    if(tdate > pDate[1] && (tdate < pDate[2] || tdate == pDate[2]))
    {
        //In case of mowing/grazing growth would continue even in this period
        if(CurrentCropPara.LAI < LAIMX)
        {
            CurrentCropPara.LAI+=Slope1;
            if(tdate.GetDay() == 15 || tdate.GetDay() == 30)
            {
                if(CurrentCropPara.LAI > CUTOFF_LAI)
                    CurrentCropPara.LAI=CUTOFF_LAI;
            }
            CurrentCropPara.FR=ChangeFr(CurrentCropPara.LAI);
            CurrentCropPara.CH=ChangeCh(CurrentCropPara.LAI);
        }
        //GrowRoot();
    }
}

```

```

        Season = TRUE;
        return;
    }
    //Second slope
    if(tdate > pDate[2] && tdate < pDate[3])
    {
        CurrentCropPara.LAI=Slope2;
        if(CurrentCropPara.LAI < MinLai)
            CurrentCropPara.LAI = MinLai;
        CurrentCropPara.FR=ChangeFr(CurrentCropPara.LAI);
        CurrentCropPara.CH=ChangeCh(CurrentCropPara.LAI);
        Season = TRUE;
        return;
    }

    if(tdate == pDate[3]) //Last day of active growth period
    {
        CurrentCropPara.LAI = MinLai;
        CurrentCropPara.FR = ChangeFr(CurrentCropPara.LAI);
        CurrentCropPara.CH = ChangeCh(CurrentCropPara.LAI);
        Season = FALSE;
        return;
    }
    if (tdate > pDate[3])
    {
        if(Season == FALSE)
            return;
        else
        {
            CurrentCropPara.LAI = MinLai;
            CurrentCropPara.FR = ChangeFr(CurrentCropPara.LAI);
            CurrentCropPara.CH = ChangeCh(CurrentCropPara.LAI);
            Season = FALSE;
        }
        return;
    }
}
void LandCoverPara :: CheckSms(float tTheta, float tThetaT4RG)
{
    if (tTheta < tThetaT4RG)
        SmsCounter++;
    else
        SmsCounter = 0;
}
//this function is used only in growing season
void LandCoverPara :: GrowRoot(void)
{
    if(CurrentCropPara.RD < RDMX)
        if(SmsCounter > 8)
            CurrentCropPara.RD = CurrentCropPara.RD+(float)10.0;
};

float LandCoverPara::ChangeFr(float tLai)
{
    float NFr;
    NFr = MinFr + ((tLai - MinLai)/(LAIatMFR - MinLai))*(FRMX - MinFr);
}

```

```

        if(NFr > FRMX)
            NFr = FRMX;
        return NFr;
    }
float LandCoverPara::ChangeCh(float tLai)
{
    float tCh = MinCh + ((tLai-MinLai)/(LAIMX-MinLai))*(CHMX-MinCh);
    return tCh;
}
#include "stdafx.h"
#include "metvalue.h"
#include <iostream.h>

// these are definitions for Class MetValue
int MetValue::OpenFile(char *WeatherFileName)
{
    char line[100];
    MetIn.open(WeatherFileName, ios::nocreate);
    if (!MetIn)
        return 0;

    //read first line of the data file
    MetIn.getline(line, 100);
    return 1;
}

void MetValue::CloseFile(void)
{
    MetIn.close();
}

int MetValue :: SetValue (void)
{
    float MetData[8];
    char date[9];
    //First it try to read
    MetIn>>SiteName;
    //Check for the EOF
    if(MetIn.eof())
        return (-1);

    MetIn>>date;
    mDate.SetDate(date);
    for (int i=0; i < 8; i++)
        MetIn>>MetData[i];
    //assign data to variables
    if(MetData[0] != -999.0)
    {
        TMX = MetData[0];
        TMX = (((TMX-32)*5)/9);
    }
    if(MetData[1] != -999.0)
    {
        TMN = MetData[1];
        TMN = (((TMN-32)*5)/9); // changing
    }
}

unit from F to C

```



```

        if(MetData[2] != -999.0)
        {
            TA = MetData[2];
            TA = (((TA-32)*5)/9);
        }
        if(MetData[3] != -999.0)
        {
            TMD = MetData[3];
            TMD = (((TMD-32)*5)/9);
        }
        if(MetData[4] != -999.0)
        {
            PC = MetData[4];
            PC = (PC*25.4); //Changing unit from
inch to mm
        }
        if(MetData[5] != -999.0)
        {
            P = MetData[5];
            P = (P*3.3864); // changing unit from
inch to kPa
        }
        if(MetData[6] != -999.0)
        {
            ux = MetData[6];
            ux = (ux*0.44704); // changing unit from
mph to mps
        }
        if(MetData[7] != -999.0)
            RS = MetData[7];
        return 1;
    };
//Assignment operator
//will copy everything except ifstream
//MetValue& MetValue::operator=(const MetValue& tMetValue)
//{
//    TMX=tMetValue.TMX;
//    TMN=tMetValue.TMN;
//    TA= tMetValue.TA;
//    TMD=tMetValue.TMD;
//    PC= tMetValue.PC;
//    P = tMetValue.P;
//    ux= tMetValue.ux;
//    RS= tMetValue.RS;
//    mDate=tMetValue.mDate;
//    strcpy(SiteName, tMetValue.SiteName);
//    return *this;
//}
#include "stdafx.h"
#include "pet.h"
#include <math.h>

//calculating saturation vapor pressure
float pet::calc_svp (float TEMP)
{
    float svp1;

```

```

        svp1=0.611*(exp((17.27*TEMP)/(TEMP + 237.3)));
        return svp1;
    };
// calculating aerodynamic resistance
void pet::calc_ra ()
{
    float k;
    float assum_ch; //FAO method assumes a crop height of 12 cm
    assum_ch = 0.12;
    zom = 0.123*assum_ch;
    zov = 0.1*zom;
    du = (2/3)*assum_ch;
    k = 0.41;
    ra = (float)((log((x0 - du)/zom))*(log((x1 - du)/zov))/(ux*k*k));
};
//calculating required parameters of PM model
void pet::CalcValues(float tElev, float tLat)
{
    float SVP_TA;
    double Gsc, Dr, omegas, phi, eita, Ka, Ra, Rso, tRn, Tkn,
Tkn, ted;

    const double stef_boltz = 4.9E-9;
    LH = 2.501 - (0.002361*Ta);
    SVP_TMX = calc_svp(TMX);
    SVP_TMN = calc_svp(TMN);
    SVP_TMD = calc_svp(TMD);
    SVP_TA = calc_svp(Ta);
    SVP = (SVP_TMX + SVP_TMN)/2;
    VP = SVP_TMD;
    DEL = (4099*SVP_TA)/((Ta+237.3)*(Ta+237.3));
    VPD = SVP - VP;
    if(VPD < 0.0)
        VPD = 0.0;
    GAMMA = (Cp*P)/(0.622*LH);
    Tv = (Ta+273.3)*(1-0.378*(VP/P));
    AD = 3.486*(P/Tv);
    //calculations for Net Radiation
    phi = (double)tLat;
    Tkn = (double)TMN+273.3;
    Tkn = (double)TMN+273.3;
    ted = (double)SVP_TMD;
    Gsc=0.082;
    Dr=1+0.33*cos((2*3.14*J)/365);
    eita=0.4093*sin((2*3.14*(284+J))/365);
    omegas=acos(-(tan(phi)*tan(eita)));
    Ka=0.75+0.00002*tElev;

    Ra=((24*60)/3.14)*Gsc*Dr*((omegas*sin(phi)*sin(eita))+(cos(phi)*cos(eita)*sin(omegas)));
    Rso=Ka*Ra;
    G = 0.38*(Ta - TaPrev);
    tRn=((1-ALPHA)*RS)-((1.35*(RS/Rso)-0.35)*(0.34-
(0.14*sqrt(ted)))*stef_boltz*((pow(Tkn,4.0)+pow(Tkn,4.0))/2));
    if(tRn>0.0)
        Rn=(float)tRn;
    else
        Rn=0.0;
}

```

```

        CalcResist();
    }
    // This function calculates resistance parameters
    void pet::CalcResist()
    {
        calc_ra();
        calc_rc();
    }

//calculating evaporation
float pet::CalcPet(Cell tCell)
{
    long int SecInDay;
    SecInDay = 86400;
    //Units are as:
    //Slope of saturation vapor pressure-temperature curve,
    DELTA :- kPa/C

    //Net radiation, Rn:- MJ/sq.m
    //Air density, AD:- kg/cu.m
    //Specific heat, Cp:- MJ/kgC
    //Vapor pressure deficit, VPD:- kPa
    //Resistances, rc & ra:- s/m
    //Latent heat, LH:- MJ/kg
    //Psychrometric constant, GAMMA:- kPa/C
    //Evapotranspiration, cpet:- mm
    CalcValues(tCell.GetElevation(), tCell.GetLatitude());
    if(rc == -999.0)
        return (float)0.0;
    else
        cpet = ((DEL*(Rn-G)) +
(AD*Cp*VPD*SecInDay)/ra)/(LH*(DEL + GAMMA*(1 + (rc/ra))));
    TaPrev = Ta;
    return cpet;
}

#include "stdafx.h"
#include "rain.h"
    RainArray::RainArray() //constructor, initializes values for rain
    {
        for (int i=0; i<5; i++)
            rain[i]=0.0;
    }

void RainArray::Update(float PC)
{
    for (int i=4; i>0; i--)
        rain[i]=rain[i-1];
    rain[0]=PC;
}

float RainArray::GetSum()
{
    float RainSum;
    RainSum=0.0;
    for (int i=0; i<5; i++)
        RainSum=RainSum+rain[i];
    return RainSum;
}
}
// RunDialog.cpp : implementation file
//

```

```

#include "stdafx.h"
#include "watbal.h"
#include "RunDialog.h"
#include <strstream>
#include <stdlib.h>
#include <iomanip.h>
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
////////////////////////////////////
// RunDialog dialog

RunDialog::RunDialog(CWnd* pParent /*=NULL*/)
: CDialog(RunDialog::IDD, pParent)
{
   //{{AFX_DATA_INIT(RunDialog)
    m_Date1 = _T("");
    m_Date2 = _T("");
    m_Date3 = _T("");
    m_Date4 = _T("");
    m_InitTheta = _T("");
    m_NumOfSimulation = 0;
    m_CurrentFr = 0.0f;
    m_CurrentLai = 0.0f;
    m_CurrentRd = 0.0f;
    m_CurrentCh = 0.0f;
    m_NumLayers = 0;
    m_CurveNumber = 0.0f;
    m_Year = _T("");
    m_MinFr = 0.0f;
    m_Beeta = 0.0f;
    //}}AFX_DATA_INIT
}

void RunDialog::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(RunDialog)
    DDX_Text(pDX, IDC_DATE1, m_Date1);
    DDV_MaxChars(pDX, m_Date1, 11);
    DDX_Text(pDX, IDC_DATE2, m_Date2);
    DDV_MaxChars(pDX, m_Date2, 11);
    DDX_Text(pDX, IDC_DATE3, m_Date3);
    DDV_MaxChars(pDX, m_Date3, 11);
    DDX_Text(pDX, IDC_DATE4, m_Date4);
    DDV_MaxChars(pDX, m_Date4, 11);
    DDX_Text(pDX, IDC_INITIAL_SMC, m_InitTheta);
    DDX_Text(pDX, IDC_NumOfSimulation, m_NumOfSimulation);
    DDV_MinMaxInt(pDX, m_NumOfSimulation, 1, 366);
    DDX_Text(pDX, IDC_CURRENT_FR, m_CurrentFr);
    DDV_MinMaxFloat(pDX, m_CurrentFr, 0.f, 10.f);
    DDX_Text(pDX, IDC_CURRENT_LAI, m_CurrentLai);
    DDV_MinMaxFloat(pDX, m_CurrentLai, 0.f, 12.f);
    DDX_Text(pDX, IDC_CURRENT_RD, m_CurrentRd);
}

```

```

    DDV_MinMaxFloat(pDX, m_CurrentRd, 0.f, 2.f);
    DDX_Text(pDX, IDC_CURRENT_CH, m_CurrentCh);
    DDV_MinMaxFloat(pDX, m_CurrentCh, -1.f, 10.f);
    DDX_Text(pDX, IDC_NUM_LAYERS, m_NumLayers);
    DDV_MinMaxInt(pDX, m_NumLayers, 0, 15);
    DDX_Text(pDX, IDC_CN, m_CurveNumber);
    DDV_MinMaxFloat(pDX, m_CurveNumber, 0.f, 100.f);
    DDX_Text(pDX, IDC_SIM_YEAR, m_Year);
    DDV_MaxChars(pDX, m_Year, 6);
    DDX_Text(pDX, IDC_MINFR, m_MinFr);
    DDV_MinMaxFloat(pDX, m_MinFr, 0.f, 1.f);
    DDX_Text(pDX, IDC_BEETA, m_Beeta);
    //}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(RunDialog, CDialog)
    //{{AFX_MSG_MAP(RunDialog)
    ON_EN_CHANGE(IDC_CURRENT_LAI, OnChangeCurrentLai)
    ON_EN_CHANGE(IDC_SIM_YEAR, OnChangeSimYear)
    ON_EN_CHANGE(IDC_MINFR, OnChangeMinfr)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
BOOL RunDialog::OnInitDialog()
{
    char Year[5];
    CDialog::OnInitDialog();
    m_CurrentRd=(float)1.00;
    m_CurrentCh=(float)0.3;
    m_CurrentLai=(float)0.0;
    m_MinFr = DlgCrop.GetMinFr();
    m_CurrentFr = DlgCrop.ChangeFr(m_CurrentLai);
    m_CurveNumber = (float)74.0;
    m_Beeta = (float)0.0026;
    itoa(1996, Year, 10);
    m_Year=Year;
    UpdateData(FALSE);
    return TRUE;
}
////////////////////////////////////
// RunDialog message handlers
void RunDialog::OnChangeCurrentLai()
{
    float fLai, fFr;
    char sLai[10], sFr[10];
    //Get Current Ch values
    GetDlgItemText(IDC_CURRENT_LAI, sLai, 10);
    //convert it to float
    istringstream istr(sLai,5);
    istr>>fLai;
    //Calculate fraction cover
    fFr = DlgCrop.ChangeFr(fLai);
    //convert it back to string
    ostringstream ostr(sFr, 10);
    ostr<<setprecision(3)<<fFr<<ends;
    //display it as fraction cover
    SetDlgItemText(IDC_CURRENT_FR, sFr);
}

```

```

        //Assign values to variables for possible DDX
        m_CurrentLai = fLai;
        m_CurrentFr = fFr;
    }
void RunDialog::OnChangeSimYear()
{
    char tYear[5];
    GetDlgItemText(IDC_SIM_YEAR, tYear, 5);
    m_Date1.SetAt(6,tYear[2]);
    m_Date1.SetAt(7,tYear[3]);
    m_Date2.SetAt(6,tYear[2]);
    m_Date2.SetAt(7,tYear[3]);
    m_Date3.SetAt(6,tYear[2]);
    m_Date3.SetAt(7,tYear[3]);
    m_Date4.SetAt(6,tYear[2]);
    m_Date4.SetAt(7,tYear[3]);
    m_Year=tYear;
    UpdateData(FALSE);
}
void RunDialog::OnChangeMinfr()
{
    float fLai, fFr;
    char sLai[10], sFr[10];
    //Set MinFr value
    GetDlgItemText(IDC_MINFR, sFr, 10);
    istringstream istr1(sFr,5);
    istr1>>fFr;
    DlgCrop.SetMinFr(fFr);
    //Assign values to variables for possible DDX
    m_MinFr = fFr;

    //Get Current Lai values
    GetDlgItemText(IDC_CURRENT_LAI, sLai, 10);
    //convert it to float
    istringstream istr2(sLai,5);
    istr2>>fLai;
    //Calculate fraction cover
    fFr = DlgCrop.ChangeFr(fLai);
    //convert it back to string
    ostringstream ostr(sFr, 10);
    ostr<<setprecision(3)<<fFr<<ends;
    //display it as fraction cover
    SetDlgItemText(IDC_CURRENT_FR, sFr);
    //Assign values to variables for possible DDX
    m_CurrentFr = fFr;
}
#include "stdafx.h"
#include "soil.h"
// This is to set parameter values for soil in each layer
void SoilPara::SetValues(float tFC, float tWP, float tPOR=0.5, float tKS=0.23, float tBP=25.89, float
tm=0.259, float tTHETAR = 0.04)
{
    FC = tFC;
    WP = tWP;
    POR = tPOR;
}

```

```

        KS = tKS;
        BP = tBP;
        m = tm;

    THETAR = tTHETAR;
};

// this will set soil properties based on texture information
void SoilPara::SetValues(int t_soil_type)
{
    // check into table and set the values
};

#include "stdafx.h"
#include <math.h>
#include "soilayer.h"
#include "soilevap.h"

SoilEvapo::SoilEvapo(SoilLayer tlayera)
{
    THETA_A = tlayera.GetTHETA();
    TSE = (float)2.0;
}

void SoilEvapo::SetValue (float ADSE1, float rain1, SoilLayer tlayera)
{
    ADSE = ADSE1;
    rain = rain1;
    tlayera = tlayera;
    THETA_A = tlayera.GetTHETA();
    DV = (float)0.0;
};

float SoilEvapo::CalcSp (float theta1 )
{
    float theta_tmp1;
    float SP, SP1, SP2, SP3, COEF, thetar;
    theta_tmp1 = theta1;
    // KS is in cm/hr and BP is in cm
    //factor 24.0 is used to convert KS into
    cm/day
    SP1 =
    ((8*tlayera.GetSoilValues().GetPOR()*tlayera.GetSoilValues().GetKS()*(float)24.0*tlayera.GetSoilVal
    ues().GetBP()/(3*(1+(3*tlayera.GetSoilValues().Getm()))*(1+(4*tlayera.GetSoilValues().Getm()))));
    SP2 = (float)sqrt(SP1);
    COEF =
    ((1/(2*tlayera.GetSoilValues().Getm()))+2);
    thetar =
    (theta_tmp1/tlayera.GetSoilValues().GetPOR());
    SP3 = (float)pow(tetar, COEF);
    SP = SP2*SP3; //unit is
    cm/(day^0.5)
    SP = SP*10; //cm is
    changed to mm
    // so SP is in mm/(day^0.5)
    return SP;
}

//This function calculates desorptivity volume for each day
float SoilEvapo::CalcDv (float THETA1)

```

```

float SP_TMP, DV_TMP, dt, temp_se, temp_sqr;
SP_TMP = CalcSp(THETA1);
dt = (float)1.0; // dt is time in days
if (TSE > 0.0)
{
    temp_sqr =
    temp_se = (float)sqrt(temp_sqr);
    TDV = TSE*temp_se;
    DV_TMP = TDV - TSE;
}
else
    DV_TMP=SP_TMP; //to safeguard
return DV_TMP;
};

float SoilEvapo::CalcSE (void)
THETA_TMP2, SE_TEMP;
approximation for DV
((THETA_A*layera.GetDEPTH()) - SE_TEMP)/layera.GetDEPTH(); //theta at the end of day 2nd
approx
THETA_TMP2)/2;
ADSE
if (DV <= ADSE)
SE = DV;
else
SE = ADSE;
return SE;
};

void SoilEvapo::SetTse(float tFr)
{
float Es;
Es = SE*(1-tFr);
if (rain > 0.0) //will set TSE to zero after every
rainfall
{
TSE = Es;
}
else
{
TSE = TSE+Es;
};
}

```



```

    }

float SoilEvapo::GetSE()
    { return SE;};

float SoilEvapo::GetTSE(void)
    { return TSE;};

float SoilEvapo::GetDV(void)
    { return DV ;};

#include "stdafx.h"
#include "soilayer.h"
#include <math.h>

SoilLayer::SoilLayer()
{
    THETA=(float)0.25;
    DEPTH=(float)0.20;
}
SoilLayer::SoilLayer(float tTheta)
{
    THETA=tTheta;
    DEPTH= (float)0.2;
}

void SoilLayer::SetDepth( float DEPTH1)
{
    DEPTH = DEPTH1*1000; //conversion from m to
mm
}

int SoilLayer::WiltingPoint()
{
    if(THETA < SoilValues.GetWP())
    return 1;
    else
    return 0;
};

float SoilLayer::GetWater(void)
{
    float TWA;
    TWA = THETA*DEPTH;
    return TWA;
};

float SoilLayer::GetTHETA(void)
{
    return THETA;
}

float SoilLayer::GetWD(void)
{
    float WD;
    WD = (SoilValues.GetPOR()-
THETA)*DEPTH;

    if (WD < 0.0)
        return (float)0.0;
    else
    return WD;
}

```

```

};

float SoilLayer::GetAvWater(void)
{
    float AvWater;
    AvWater = (THETA - SoilValues.GetWP())*DEPTH;
    if(AvWater <0.0)
        AvWater = (float)0.0;
    return AvWater;
}

void SoilLayer::SetTHETA(float T)
{
    THETA = T;
}

float SoilLayer::CalcExtCoef(void)
{
    double NTheta, CosTheta, ExtCoef;
    if(THETA >= SoilValues.GetFC())
        return (float)1.0;
    else
    {
        NTheta = (THETA *3.14)/SoilValues.GetFC();
        CosTheta = cos(NTheta);
        ExtCoef = (1-CosTheta)*(1-CosTheta)/4.0;
        //Modify coef. if theta has reached below wilting point
        if(WiltingPoint())
            ExtCoef = ExtCoef*ExtCoef;
        return (float)ExtCoef;
    }
}

// watbalView.cpp : implementation of the CWatbalView class
//
#include "stdafx.h"
#include "afxdlgs.h"
#include "watbal.h"
#include "GetFileDialog.h"
#ifdef _STRSTREA_H_
#include <strstrea.h>
#endif
#include <iomanip>
#include <direct.h>
#include "watbalDoc.h"
#include "watbalView.h"
#include "RunDialog.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
#include "run.h"
#include <afxtempl.h>
#include "ctheta.h"
int rndto(float);
////////////////////////////////////
// CWatbalView

```

```

// Define filters for use with the File dialog box
const char FileDialogFilter[]="Text Files (*.txt)|*.txt| Data Files\
(*.dat)|*.dat|All Files (*.*)|*.*|";
const char FileDialogExt[] = "dat";
IMPLEMENT_DYNCREATE(CWatbalView, CScrollView)
BEGIN_MESSAGE_MAP(CWatbalView, CScrollView)
   //{{AFX_MSG_MAP(CWatbalView)
    ON_COMMAND(IDC_INPUT_WEATHER, OnInputWeather)
    ON_UPDATE_COMMAND_UI(IDC_INPUT_WEATHER, OnUpdateInputWeather)
    ON_COMMAND(IDC_INPUT_CROP, OnInputCrop)
    ON_UPDATE_COMMAND_UI(IDC_INPUT_CROP, OnUpdateInputCrop)
    ON_COMMAND(IDC_INPUT_LAYER, OnInputLayer)
    ON_UPDATE_COMMAND_UI(IDC_INPUT_LAYER, OnUpdateInputLayer)
    ON_COMMAND(ID_RUN, OnRun)
    ON_COMMAND(ID_OUT_CHART, OnOutChart)
    ON_COMMAND(ID_OUT_TEXT, OnOutText)
    ON_UPDATE_COMMAND_UI(ID_OUT_CHART, OnUpdateOutChart)
    ON_UPDATE_COMMAND_UI(ID_OUT_TEXT, OnUpdateOutText)
    ON_WM_LBUTTONDOWN()
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////
// CWatbalView construction/destruction
CWatbalView::CWatbalView()
{
    WeatherFile = new char[50];
    CropFile = new char[50];
    LayerFile = new char[50];
    //sets flag to zero
    wfFlag = FALSE;
    cfFlag = FALSE;
    lfFlag = FALSE;
    RunIsOver=FALSE;
    Color[0] = RGB(255,0,0);
    Color[1] = RGB(0,255,0);
    Color[2] = RGB(0,0,255);
    Color[3] = RGB(128,128,0);
    Color[4] = RGB(128,0,128);
    Color[5] = RGB(100,100,100);
    Color[6] = RGB(250,100,0);
    Color[7] = RGB(0,250,250);
    Color[8] = RGB(150,0,250);
    Color[9] = RGB(50,200,50);
}
CWatbalView::~~CWatbalView()
{
    delete[] WeatherFile;
    delete[] CropFile;
    delete[] LayerFile;
}
BOOL CWatbalView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs
    BOOL bPreCreated = CScrollView::PreCreateWindow(cs);
}

```

```

        return bPreCreated;
    }
    ///////////////////////////////////////////////////////////////////
    // CWatbalView drawing
    void CWatbalView::OnDraw(CDC* pDC)
    {
        CWatbalDoc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);
        if(eViewType == ViewText)
            TextOut(pDC);
        if(eViewType == ViewChart)
            ChartOut(pDC);
    }
    void CWatbalView::OnInitialUpdate()
    {
        CScrollView::OnInitialUpdate();
        CSize sizeTotal;
        // TODO: calculate the total size of this view
        sizeTotal.cx = 100;
        sizeTotal.cy = 100;
        SetScrollSizes(MM_TEXT, sizeTotal);
    }
    ///////////////////////////////////////////////////////////////////
    // CWatbalView diagnostics
    #ifdef _DEBUG
    void CWatbalView::AssertValid() const
    {
        CScrollView::AssertValid();
    }
    void CWatbalView::Dump(CDumpContext& dc) const
    {
        CScrollView::Dump(dc);
    }
    CWatbalDoc* CWatbalView::GetDocument() // non-debug version is inline
    {
        ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CWatbalDoc));
        return (CWatbalDoc*)m_pDocument;
    }
    #endif // _DEBUG
    ///////////////////////////////////////////////////////////////////
    // CWatbalView message handlers
    void CWatbalView::FileInput(char *FileName)
    {
        //declare this char s as global or
        //return copy of this instead of addrss
        //ostream fstr(FileName, 50);
        int nModal;
        //char s[50];
        //Create a dialog box to input filename
        CGetFileDlg FileDialog(TRUE, NULL, NULL,
        OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, FileDialogFilter);
        nModal = FileDialog.DoModal();
        if (nModal == IDOK)
        {
            // Take filename and read data here
            CString csTemp = FileDialog.GetPathName();

```

```

        strcpy(fileName, csTemp);
    }
}

void CWatbalView::OnInputWeather()
{
    // Create a dialog box to input filename
    FileInput(WeatherFile);
    wfFlag = TRUE;

    // Do processing here
}

void CWatbalView::OnUpdateInputWeather(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
}

void CWatbalView::OnInputCrop()
{
    // Create a dialog box to input filename
    FileInput(CropFile);
    cfFlag = TRUE;

    // Do processing here
}

void CWatbalView::OnUpdateInputCrop(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
}

void CWatbalView::OnInputLayer()
{
    FileInput(LayerFile);
    lfFlag = TRUE;
    ifstream filein(LayerFile, ios::nocreate);
    if(filein)
    {
        char *junk=new char[10];
        filein>>junk>>SiteName;
        filein>>junk>>NumOfLayers;
        filein.close();
        delete[] junk;
    }
}

void CWatbalView::OnUpdateInputLayer(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
}

void CWatbalView::RunTheModel()
{
    CTheta tTheta(NumOfLayers);
    Cell Cell11(NumOfLayers);
    MetValue DailyMet;
}

```

```

pet FaoPet;
Infiltration CnInfil;

//Position of elements of ListOfTheta
//temporary variables
float ADSE1, SOILPEV, SE, EVAPOT, Inflow, CellTheta;
//Pointer to document
CWatbalDoc* pDoc = GetDocument();

//set layers para from LayerFile
if(Cell11.SetLayersPara(LayerFile) == 0)
{
    MessageBox("Could not set Layer's parameters",
               "File open error",
               MB_ICONEXCLAMATION);
    DestroyWindow();
}

//set crop data from CropFile
if(!Cell11.GetCrop().SetLandCoverPara(CropFile, SiteName))
{
    MessageBox("Could not set Crop's parameters",
               "File open error",
               MB_ICONEXCLAMATION);
    DestroyWindow();
}

//Open weather file
if (Dailymet.OpenFile(WeatherFile) == 0)
{
    MessageBox("Could not open weather file",
               "File open error",
               MB_ICONEXCLAMATION);
    DestroyWindow();
}

//Set parameters of cell
Cell11.SetCN();
SoilEvapo SolEvap(Cell11.Getlayer(1)); //sets the first layer and TSE
//Cell11.GetCrop().SetCropPara( 1.0, 1.0, 0.2,0.2);
cropet grasset(Cell11);
//Get Initial values and other parameters
//Then run the model
if(GetRunInfo(Cell11))
{
    RunIsOver=FALSE;
    int Result;
    float f;
    float Intercept = (float)0.0; //Canopy interception
    char *theta = new char[5];
    Cell11.GetCrop().CalcSlopes();
    theta = strtok(InitTheta, ",");
    f = (float)atof(theta);
    Cell11.Getlayer(1).SetTHETA(f);
    for(int m=1; m<NumOfLayers||theta==NULL; m++)
    {
        theta = strtok(NULL, ",");
        f = (float)atof(theta);
    }
}

```

```

        Cell11.Getlayer(m+1).SetTHETA(f);
    }
    //Set some values
    Cell11.SetNumOfActiveLayers(); //based on root depth as taken from the RunInfo
dialog
    Cell11.SetThetaT();
    grasset.SetSmValues(Cell11); //this sets FC,WP and THETA values
    // Create stream for output
    ofstream fout("C:\\zak\\Project\\Result\\Result.txt");
    if (!fout)
    {
        MessageBox("Could not create output file Result.txt",
            "File open error",
                MB_ICONEXCLAMATION);
        SendMessage(WM_COMMAND, ID_APP_EXIT);
    }

    //Create stream for intermediate files
    ofstream OptOut("C:\\zak\\Project\\Result\\Intermed.txt");
    if (!OptOut)
    {
        MessageBox("Could not create output file Intermed.txt",
            "File open error",
                MB_ICONEXCLAMATION);
    }

    //Empty the List in case it is not the first run
    if(!pDoc->ListOfTheta.IsEmpty())
        pDoc->ListOfTheta.RemoveAll();
    //Tell user the model will write output to Result.txt
    MessageBox(" Result would be written to Result.txt.\nIntermediate file is Intermed.txt.",
        "Result Info",
            MB_ICONEXCLAMATION);

    // Put the headings
    fout<<"SITE"<<"\t"<<SiteName<<endl;
    fout<<"DATE";
    for (int x=1; x<=NumOfLayers; x++)
        fout<<"\t"<<"THETA-"<<x;
    fout<<endl<<setprecision(4);

    OptOut<<"SITE"<<"\t"<<SiteName<<endl;

    OptOut<<"DATE"<<"\t"<<"ADSE"<<"\t"<<"Trans"<<"\t"<<"Rn(mm)"<<"\t"<<"SE"<<"\t"<
<"RAINFALL"<<"\t"<<"INFLOW";
    OptOut<<"\t"<<"LAI"<<"\t"<<"FR"<<"\t"<<"Rc"<<"\t"<<"ET"<<endl;
    OptOut<<setprecision(4);
    //////////////////////////////////////
    //////////////////////////////////////
    ////////////////////////////////////// Here the model run starts //////////////////////////////////////
    //////////////////////////////////////
    //////////////////////////////////////
    for (int i = 0; i<NumOfSimulation; i++)
    {
        //Cell11.SetDrain((float)0.0);
        Result = Dailymet.SetValue();
        if(Result == -1)

```

```

        {
            MessageBox("End of File reached.", "Run_Error",
MB_ICONEXCLAMATION);
            break;
        }
        FaoPet.SetValue(Dailymet);
        if(i == 0)
            FaoPet.SetTaPrev(Dailymet.GetTA());

//calculates atmospheric demand

ADSE1 = (float)1.15*FaoPet.CalcPet(Cell11);

        CellTheta = Cell11.GetTheta();
        grasset.SetValue (Dailymet, Cell11.GetCrop().GetCropPara(), CellTheta);
        if(i == 0)
            grasset.SetTaPrev(Dailymet.GetTA());
            if(grasset.GetTheta() > grasset.GetWp())
                EVAPOT = grasset.CalcPet(Cell11);
        else
            EVAPOT = (float)0.0;
//Calculates Soil Evaporation as min(atmospherics demand, desorptivity)
        SolEvap.SetValue(ADSE1, Dailymet.GetPC(), Cell11.Getlayer(1));
        SOILPEV = SolEvap.CalcSE();
        SolEvap.SetTse(Cell11.GetCrop().GetCropPara().FR);
//Calculates water taken from each layer on the basis of fraction cover
        if(!Cell11.CalcSoilMoist(EVAPOT, SOILPEV, Intercept))
        {
            //handle the error
        }
            SE = Cell11.GetSe();
            Cell11.DrainWater();

        CnInfil.SetValue(Cell11, Dailymet.GetPC());
        CnInfil.UpdateRain5();
        if(Dailymet.GetPC()>0.0)
        {
            Inflow = CnInfil.Calculations();
            if(Inflow > 0.0) //Infiltrate only Inflow is greater than zero
                Cell11.InfiltrateWater(Inflow);
                Intercept = CnInfil.GetIntercept();
        }
        else
        {
            //In case there is no rainfall, inflow and intercept both are set to zero
            Inflow = (float)0.0;
            Intercept = (float)0.0;
        }

//write output to file and
//update List
        fout<<Dailymet.GetCarDate();
            tTheta.SetDate(Dailymet.GetCarDate());
            for(int m=1; m<=NumOfLayers; m++)
            {
                tTheta.SetTheta(m-1, Cell11.Getlayer(m).GetTHETA());
                fout<<"\t"<<Cell11.Getlayer(m).GetTHETA();
            }

```



```

    }
    fout<<endl;
    pDoc->ListOfTheta.AddTail(tTheta);

OptOut<<Dailymet.GetCarDate()<<'t'<<ADSE1<<'t'<<EVAPOT<<'t'<<(grasset.GetRn()/grasset.
GetLh());
    OptOut<<'t'<<SE<<'t';
    OptOut<<Dailymet.GetPC()<<'t'<<Inflow<<'t';
    OptOut<<Cell11.GetCrop().GetCropPara().LAI<<'t';

OptOut<<Cell11.GetCrop().GetCropPara().FR<<'t'<<grasset.GetRc()<<'t'<<Cell11.GetET()<<end
l;

    // Crop growth function
    //Cell11.GetCrop().CheckSms(Cell11.GetTheta(), Cell11.GetThetaT4RG());
    Cell11.GetCrop().GrowCrop(Dailymet.GetDate());
    //Set values as they might have changed if roots have grown
    //Cell11.SetNumOfActiveLayers();
    //Cell11.SetThetaT();
    //Cell11.SetThetaT4RG();
};
//close the open files
Dailymet.CloseFile();
fout.close();
OptOut.close();
MessageBox("Run is over",
    "Finish Info",
    MB_ICONEXCLAMATION);
RunIsOver=TRUE;
OnOutText();
}
}

void CWatbalView::OnRun()
{
    if (cfFlag == FALSE || lfFlag == FALSE || wfFlag == FALSE)
        MessageBox("Sorry, you did not input files",
            "File Input Error",
            MB_ICONEXCLAMATION);
    else
    {
        CString Info;
        Info.Format("WeatherFile=%s \n LayerFile=%s \n CropFile=%s \n", \
WeatherFile, LayerFile, CropFile);
        int result=AfxMessageBox(Info, MB_OKCANCEL);
        if(result == IDOK)
            RunTheModel();
    }
}
int CWatbalView::GetRunInfo(Cell& cell)
{
    RunDialog dlg;
    NumOfSimulation=366;
    char line[100];
    char junk[6];

```

```

    char Site[5];
    ifstream fin(CropFile, ios::nocreate);
    if (!fin)
        return 0;
    fin.getline(line, 100);
    do
    {
        fin>>Site;
        for(int i=0; i<8; i++)
            fin>>junk;
        fin>>Date1>>Date2>>Date3>>Date4;
    }
    while((strcmp(Site, SiteName)!=0) && !fin.eof());
    fin.close();
    strcat(Date1, "/96");
    strcat(Date2, "/96");
    strcat(Date3, "/96");
    strcat(Date4, "/96");
    InitTheta = new char[6*NumOfLayers];
    strcpy(InitTheta, "0.25");
    for (int i=1; i<NumOfLayers; i++)
        strcat(InitTheta, ", 0.25");
    dlg.m_InitTheta=InitTheta;
    dlg.m_NumOfSimulation=NumOfSimulation;
    dlg.m_NumLayers=NumOfLayers;
    dlg.m_Date1=Date1;
    dlg.m_Date2=Date2;
    dlg.m_Date3=Date3;
    dlg.m_Date4=Date4;
    dlg.DlgCrop=cell.GetCrop();
    if(dlg.DoModal() == IDOK)
    {
        //Important dates to grow the crop
        //get char* from CString
        cell.GetCrop().SetpDate(1, dlg.m_Date1.GetBuffer(9));
        cell.GetCrop().SetpDate(2, dlg.m_Date2.GetBuffer(9));
        cell.GetCrop().SetpDate(3, dlg.m_Date3.GetBuffer(9));
        cell.GetCrop().SetpDate(4, dlg.m_Date4.GetBuffer(9));
        dlg.m_Date1.ReleaseBuffer();
        dlg.m_Date2.ReleaseBuffer();
        dlg.m_Date3.ReleaseBuffer();
        dlg.m_Date4.ReleaseBuffer();
        //Current Values
        cell.GetCrop().SetCropPara(dlg.m_CurrentRd,
        dlg.m_CurrentCh, dlg.m_CurrentFr, dlg.m_MinFr);
        dlg.m_CurrentLaf,

        cell.GetCrop().SetBeeta(dlg.m_Beeta); //Extraction coefficient
        cell.SetCN(dlg.m_CurveNumber);
        NumOfSimulation=dlg.m_NumOfSimulation;
        strcpy(InitTheta, dlg.m_InitTheta.GetBuffer(20));
        dlg.m_InitTheta.ReleaseBuffer();
        return 1;
    }
    else
        return 0;
}

```

```

void CWatbalView::OnOutChart()
{
    if(!RunIsOver)
    {
        MessageBox("First run the model", "Output error",
            MB_ICONEXCLAMATION);
        return;
    }
    eViewType = ViewChart;
    Invalidate();
}

void CWatbalView::OnOutText()
{
    if(!RunIsOver)
    {
        MessageBox("First run the model", "Output error",
            MB_ICONEXCLAMATION);
        return;
    }
    eViewType = ViewText;
    Invalidate();
}

void CWatbalView::TextOut(CDC* tpDC)
{
    if(!RunIsOver)
        return;
    TEXTMETRIC tm;
    tpDC->GetTextMetrics(&tm);
    //Pointer to document
    CWatbalDoc* pDoc = GetDocument();
    long int TextHt = tm.tmHeight+tm.tmExternalLeading;
    int y = pDoc->ListOfTheta.GetCount()*(TextHt)+(5*TextHt)+20;
    CSize ViewSize;
    ViewSize.cx=400;
    ViewSize.cy=y;
    SetScrollSizes(MM_TEXT, ViewSize);
    POSITION pos = pDoc->ListOfTheta.GetHeadPosition();
    tpDC->TextOut(150, 10, "RESULT");
    char s[100];
    ostream ostr(s, 100);
    ostr<<"DATE"<<" ";
    for (int i=1; i<=NumOfLayers; i++)
        ostr<<"THETA-"<<i<<" ";
    ostr<<ends;
    tpDC->TextOut(10, 10+TextHt+10, s);
    TextHt=10+2*TextHt+10;

    for (int x=0; x<NumOfSimulation && x<pDoc->ListOfTheta.GetCount(); x++)
    {
        ostr.seekp(ios::beg);
        CTheta tTheta=pDoc->ListOfTheta.GetNext(pos);
        ostr<<tTheta;
        ostr<<ends;
    }
}

```

```

        tDC->TextOut(10, TextHt, s);
        TextHt=TextHt+tm.tmHeight+tm.tmExternalLeading;
    }
}
void CWatbalView::ChartOut(CDC* tDC)
{
    //ClientDC dc(this);
    //CDC* pDC = (CDC*)&dc;
    CWatbalDoc* pDoc = GetDocument();
    CPoint point;
    char s[9];
    ostringstream str(s,9);
    CSize sizeTotal;
    sizeTotal.cx = 800;
    sizeTotal.cy = 1200;
    SetScrollSizes(MM_TEXT, sizeTotal);
    TEXTMETRIC tm;
    tDC->GetTextMetrics(&tm);
    long int txtHt = tm.tmHeight+tm.tmExternalLeading;
    long int y=txtHt;
    //setting the scale
    float xscale, yscale, theta;
    xscale = (float)(600.0-80.0)/NumOfSimulation;
    yscale = (float)(380.0-40.0)/pDoc->GetHighestTheta();
    tDC->TextOut(10, 210, "SMC");
    tDC->TextOut(290, 400, "DATE");
    CPen* MyPen;
    tDC->MoveTo(80,40);
    tDC->LineTo(80,380);
    tDC->LineTo(600,380);
    POSITION pos;
    int count = pDoc->ListofTheta.GetCount();
    for(int x=1; x<=NumOfLayers; x++)
    {
        MyPen = new CPen;
        MyPen->CreatePen(PS_SOLID, 1, Color[x-1]);
        tDC->SelectObject(MyPen);
        str.seekp(ios::beg);
        str<<"THETA-"<<x<<ends;
        tDC->SetTextColor(Color[x-1]);
        point.x=620;
        point.y=y;
        tDC->DPtoLP(&point);
        tDC->TextOut(point.x, point.y, s);
        pos=pDoc->ListofTheta.GetHeadPosition();
        CTheta tTheta = pDoc->ListofTheta.GetNext(pos);
        theta = tTheta.GetTheta(x);
        tDC->MoveTo(80, rndto(380-(theta*yscale)));
        for(int i=1; i<NumOfSimulation && i<count; i++)
        {
            point.x = 80+rndto(i*xscale);
            point.y = rndto(380-(theta*yscale));
            tDC->LineTo(point);
            tTheta=pDoc->ListofTheta.GetNext(pos);
            theta = tTheta.GetTheta(x);
        }
    }
}

```

```

        tDC->SelectStockObject(BLACK_PEN);
        y+=tHt;
        delete MyPen;
    }
}
int rndto(float flt)
{
    if(((flt - int(flt))>0.5))
        return int(flt)+1;
    else
        return int(flt);
}
void CWatbalView::OnUpdateOutChart(CCmdUI* pCmdUI)
{
    if(eViewType==ViewChart)
        pCmdUI->SetCheck(1);
    else
        pCmdUI->SetCheck(0);
}
void CWatbalView::OnUpdateOutText(CCmdUI* pCmdUI)
{
    if(eViewType==ViewText)
        pCmdUI->SetCheck(1);
    else
        pCmdUI->SetCheck(0);
}
void CWatbalView::OnLButtonDbkClk(UINT nFlags, CPoint point)
{
    if(eViewType != ViewChart)
        return;
    CPoint PointOnChart;
    POSITION pos;
    CWatbalDoc* pDoc = GetDocument();
    //setting the scale
    float xscale, yscale, theta;
    xscale = (float)(600.0-80.0)/NumOfSimulation;
    yscale = (float)(380.0-40.0)/pDoc->GetHighestTheta();
    int count = pDoc->ListOfTheta.GetCount();
    for(int x=1; x<=NumOfLayers; x++)
    {
        pos=pDoc->ListOfTheta.GetHeadPosition();
        CTheta tTheta = pDoc->ListOfTheta.GetNext(pos);
        theta = tTheta.GetTheta(x);
        for(int i=1; i<NumOfSimulation && i<count; i++)
        {
            PointOnChart.x = 80+rndto(i*xscale);
            PointOnChart.y = rndto(380-(theta*yscale));
            if(point == PointOnChart)
            {
                ChangeColor(x);
                return;
            }
            tTheta=pDoc->ListOfTheta.GetNext(pos);
            theta = tTheta.GetTheta(x);
        }
    }
}

```

```
        }  
    }  
}  
void CWatbalView::ChangeColor(int x)  
{  
    CColorDialog ColorDlg;  
    if(ColorDlg.DoModal() == IDOK)  
    {  
        Color[x-1] = ColorDlg.GetColor();  
        Invalidate();  
    }  
}
```

VITA

Zaigham A. Kazmi

Candidate for the Degree of

Master of Science

**Thesis: DEVELOPMENT AND TESTING OF A DAILY WATER-BALANCE
MODEL FOR ESTIMATING EVAPOTRANSPIRATION AND SOIL WATER
CONTENT**

Major Field: Biosystems Engineering

Biographical:

Personal Data: Born in Allahabad, U.P., India, on April 10, 1972.

Education: Graduated (10th) from Y.H.I.C., Allahabad, India in July 1987; passed Intermediate (12th) from Allahabad Agricultural Institute, Allahabad, India in July 1989; received Bachelor of Technology degree in Agril. Eng. from Allahabad Agricultural Institute, Allahabad, India in July 1994. Completed the requirements for the Master of Science degree with a major in Biosystems Engineering at Oklahoma State University in May 1998.

Experience: Employed by Oklahoma State University, Department of Biosystems and Agricultural Engineering as a graduate research assistant, 1996 to present.

Professional Membership: ASAE (The Society for Engineering in Agricultural, Food, and Biological systems), AWRA (American Water Resource Association), Alpha Epsilon (The Honor Society of Agricultural Engineering), Gamma Sigma Delta (The Honor Society of Agriculture).