EVALUATING BEST MANAGEMENT PRACTICES FOR MINIMIZED

PESTICIDES TRANSPORT WITH THE WEPP-UI MODEL

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the

Degree of Masters of Science

by

Sheila M. Saia

August 2012

ABSTRACT

Best management practice (BMP) tools that provide site-specific water quality suggestions with limited input data are needed to assist consultants and farm advisors as they work to target BMP placement and reduce dissolved and sediment bound pesticide loads from agricultural landscapes. BMP effectiveness largely depends on accurate characterization of dominant regional hydrologic processes. However, most BMP tools capture either infiltration excess runoff or saturation excess runoff but cannot predict a combination of the two. The objective is therefore to develop an easy-to-use BMP tool capable of (1) accurately characterizing the dominant hydrological processes in a region of interest and (2) evaluating the selection and placement of various BMPs for reductions in pesticide loss. In this study we show that the Watershed Erosion Prediction Project University of Idaho (WEPP-UI) model with coupled pesticide module can realistically simulate both types of runoff processes as well as dissolved pesticide loss. Simulated runoff and dissolved atrazine concentrations were compared to observed data from mulch till (MT) and no till (NT) plots in the Goodwater Creek Experimental Watershed (GCEW) of northeastern Missouri. The timing of runoff predictions by WEPP-UI coincided well with observed events and flow magnitudes were between minimum and maximum observations for the two replicate plots. Nash-Sutcliffe efficiencies (NSEs) were 0.84 and 0.79 for MT and NT runoff simulations, respectively. Both simulations and observations showed little difference in total growing season (planting to harvest) runoff between MT and NT. Dissolved atrazine loads were well predicted for MT and NT plots with NSEs of 0.59 and 0.71, respectively. Consistent with observations, seasonal NT atrazine losses were higher than losses from MT plots. Lastly, we also use the WEPP-UI pesticides module to simulate the impact of precision application on dissolved pesticide loss. While this study only focuses on dissolved pesticide transport at the bottom of a

hillslope, the WEPP-UI pesticide transport module can predict loads for a variety of pesticide types from each section of the hillslope (i.e. top-slope, mid-slope, and toe-slope), including sediment bound pesticide loads. Vegetated buffer strip routines are also available but more extensive field monitoring is needed to validate these aspects of the model.

BIOGRAPHICAL SKETCH

Sheila Marie Saia was born in Buffalo, New York on August 21, 1985. She graduated in 2007 with a Bachelors of Science degree in Bioengineering from the University of Binghamton. In the autumn of 2007 she headed to Chiba, Japan where she taught English at a private school for the next year. When she was not teaching, she explored Japan and learned all she could about Japanese culture. Towards the end of her stay, she had the opportunity to attend a G12 International Symposium on Biodiversity and Climate Change. Upon returning home to the US, she accepted an Americorps sponsored volunteering coordinator position at Buffalo Niagara Riverkeeper. Her positive experience at Buffalo Niagara Riverkeeper motivated her to pursue an advanced degree in environmental engineering. Sheila is dedicated to improving water quality through research and community education.

ACKNOWLEDGMENTS

I would like to thank my committee members—Tammo Steenhuis and Todd Walter—for their support and patience. I would also like to thank the other researchers who provided comments on my paper including: Erin Brooks, Zach Easton, Jan Boll, and Claire Baffaut. Many thanks to Erin Brooks for his insights and assistance with the WEPP model. Thank you to my mother for encouraging me to pursue a degree in the sciences and to my father who inspired my love for nature and all things hydrologic. Thank you to my sister for being honest and creative. Thank you to my grandparents—Mary Jane and Harold Hensen as well as James and Norma Saia—who taught me how to work hard and be grateful for what I have. Thanks to the many friends and peers in the US and abroad who have supported me both before and during my time at Cornell. I am especially grateful to have worked with and befriended lab-mates, Christine Baver and Christian Guzman. Thanks to Veronica Morales for continually pushing me to think like a scientist and pursue the endless opportunities available to me. Thanks to the rest of my peers in the Soil & Water Lab. I could not have completed this work without your feedback and friendship. Thanks to all of my climbing friends including Scott Henderson, Megan Mills-Novoa, Andy Castler, Scott Cambo, Johan Ugander, and Julie Muller who always knew when I needed to take a break from research. Thanks to Ben Currens for his positivity and continuous encouragement. Thanks to Jonathan Newman for his critical eye and excitement for all things science. Last but not least, I would like to thank my best friend Shelley Wilson for her love, support, and courageousness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABELS

SECTION 1: INTRODUCTION

Pesticides found in agricultural runoff threaten human health and ecosystems function (Colborn et al. 1993; Solomon et al. 2008; Lerch et al. 2011; USEPA 1996). Atrazine, one of the most commonly used pesticides, has been detected in both surface (Lerch et al. 1998; Scribner et al. 2005) and subsurface water (Sinkevich et al. 2005; Richards et al. 2012) at high enough levels to be of concern. In response, the Natural Resources Conservation Service (NRCS) has approved over 160 BMPs (USDA-NRCS 2006a; 2006b; 2006c) to control runoff, erosion, and associated contaminant transport (D'Arcy & Frost 2001; Merriman et al. 2009). Some BMPs are implemented to reduce contaminant transport to nearby water bodies (Mostaghimi et al. 1997; Miao et al. 2004; Blanco-canqui et al. 2004; Hively et al. 2006; Easton et al. 2008b; Rao et al. 2009). While others serve the dual purpose of trapping sediment and sediment bound contaminants. For example, vegetated buffer strips are designed with the intention to trap sediment bound nutrients (Heathwaite & Johnes 1996) and pesticides (Reichenberger et al. 2007) transported through the buffer via overland flows. No till (NT) farming can improve water quality when used in combination with pesticides that degrade quickly in the environment and can also be easily adsorbed to soil particles or organic matter (e.g. glyphosate; Cerdeira & Duke 2006). However, despite best intentions, not all BMPs perform as expected. Vegetated buffer strips may serve as a phosphorus source rather than sink (Dillaha et al. 1989) and in some locations NT farming has been found to increase pesticide loss (Ghidey et al. 2005; 2010; Needelman et al. 1998).

To target BMP placement and improve overall BMP effectiveness, two dominant runoff processes must be considered: saturation excess overland flow (Dunne 1970; Frankenberger et al. 1999; Walter et al. 2000) and infiltration excess overland flow (Horton 1933). In many areas

of the United States, where the climate is humid and the soil is underlain by a low permeable layer (e.g. claypan, fragipan), infiltration and poor drainage at shallow top-slopes leads to the formation of saturated areas that expand and contract seasonally (McDaniel et al. 2008; Brooks & Boll 2011). As the source of surface runoff, these variable or partial source areas are main contributors to contaminant loss (Gburek et al. 2002; Dunne & Black 1970; Needelman et al. 1998; Walter et al. 2000; Müller et al. 2006, Reichenberger et al. 2007; Bayabil et al. 2010). When BMP selection tools cannot accurately predict major hydrologic flow mechanisms, BMP placement may not be optimal (Tomer & Locke 2011). Consequently, there is a need for a physically-based tool that can accurately classify local hydrology, erosion processes, and pesticide transport potential for various management activities along the hillslope.

Currently in the United States, the two most commonly used tools for evaluating BMP effectiveness are the Soil Water Assessment Tool-SWAT (Arnold et al. 1998) and the Generalized Watershed Loading Function-GWLF (Haith & Shoemaker 1987; Schneiderman et al. 2002). Other popular models such as Agricultural Non-Point Source Pollution-AGNPS (Young et al. 1989), Hydrologic Simulation Program Fortran-HSPF (Bicknell et al. 2001), Agricultural Drainage and Pesticide Transport-ADAPT (Chung et al. 1992), and APEX (Mudgal et al. 2010) have been used to model hydrological processes and BMP impacts. All of these models are based on the assumption that infiltration excess flow is the dominant runoff mechanism. Models based on saturation excess runoff mechanisms include the Soil Moisture Distribution and Routing-SMDR (Zollweg et al. 1996; Frankenberger et al. 1999), Variable Source Area Loading Function-VSLF (Schneiderman et al. 2007; Rao et al. 2009; Easton et al. 2009), and Soil Water Assessment Tool for Variable Source Areas-SWAT-VSA (Easton et al. 2008a). These models

are able to predict saturated areas in the landscape but require advanced training to calibrate for specific regions.

The WEPP model is a process-based hydrology and erosion model that has been tested for a variety of tillage practices (e.g. mulch till, no till, etc.), crop rotations (e.g. corn-soy, winter wheat-barley-pea, etc.), soils, topographies, climates, and other agricultural BMPs (e.g. vegetated buffer strips) in the United States and abroad (Burroughs et al. 1989; Ascough et al. 1997; Larose et al. 2004; Pieri et al. 2007; Greer et al. 2006; Singh et al. 2009). Boll et al. (in review) modified the WEPP model to represent both saturation excess and infiltration excess overland flow. This version of WEPP, called the Watershed Erosion Prediction Project – University of Idaho (WEPP-UI) model, is unique because it can simulate both runoff processes given site-specific rainfall intensities and soil conditions. By combining a simple pesticide runoff model (Steenhuis & Walter 1980) with WEPP-UI, we can simulate both saturation excess and infiltration excess runoff as well as associated pesticide losses. Other commonly used models such as SWAT, ADAPT, APEX, and the Watershed Regression for Pesticides-WARP (Larson et al. 2004) are not capable of predicting pesticide losses when there is saturation excess runoff. Simulating the correct runoff mechanisms is important because the location of runoff generation as well as pesticide loss can vary with saturation excess and infiltration excess runoff.

The purpose of this study is to: (1) couple a pesticide transport module with the WEPP-UI model, (2) evaluate model predictions of runoff and pesticide transport against field data from the Goodwater Creek Experimental Watershed (GCEW) of northeastern Missouri, and (3) use the newly coupled model to explore various precision application scenarios. The WEPP-UI coupled pesticide module is being incorporated into a recently developed, web-based tool called the

12

Hydrologic Characterization Tool (HCT) available at: http://wepp.ag.uidaho.edu/cgi-bin/HCT.pl/ so it can be used by non-experienced modelers (Brooks & Boll 2011).

This study is part of a larger Conservation Effects Assessment Project (CEAP) study funded by the United States Department of Agriculture (USDA) with the goal of developing site-specific, easy-to-use management tools. The web-based HCT can be used by soil and water managers to identify dominant hydrologic flow paths and associated contaminant transport. Here we focus on dissolved and particulate pesticide loss, but we intend to incorporate nitrogen and phosphorus modules into the HCT as well. Additionally, the HCT can be used to test out combinations of BMPs for a given region by varying soils, topography, crop rotations, application amounts, and application location.

SECTION 2: METHODOLOGY

WEPP Model Description

The process-based Watershed Erosion Prediction Project (WEPP) model for hillslopes (Flanagan & Nearing 1995) and watersheds (Baffaut et al. 1997; Ascough et al. 1997) was developed to improve soil and water management and address the limitations of empirical erosion models such as the Universal Soil Loss Equation-USLE (Wischmeier & Smith 1978) and other modified versions (Renard 1991; Renard et al. 1997). See Flanagan et al. (2007) for a complete history of WEPP model development.

Climate data required to run WEPP include: daily maximum and minimum air temperature, daily average wind speed, solar radiation, dew point temperature, and precipitation distribution throughout the day. Climate files can be created by the user or can be generated using the stochastic climate generator-CLIGEN (WEPP Documentation Chapter 2, Nicks et al. 1995). The CLIGEN database includes several thousand sites across the US. WEPP soil routines can be populated using data from the SURRGO soils database (USDA-NRCS 2003). Crop type, tillage type and timing as well as slope are defined by the user.

WEPP simulates interrill erosion due to raindrop impact and thin sheet flow as well as rill erosion—the loss of soil from small channels of concentrated flow (Laflen et al. 1997). Overland flow estimates are calculated by analytically solving the kinematic wave equations (WEPP Documentation Chapter 1, Flanagan & Nearing 1995; Mein & Larson 1973). WEPP as modified by researchers at the University of Idaho (WEPP-UI), simulates converging subsurface lateral flow and generation of perched water above a hydrologically restrictive soil layer (Brooks &

Boll 2011; Boll et al. in review) using a finite difference model. Consequently, WEPP-UI is capable of predicting both infiltration-excess and saturation-excess overland flow. This classification of runoff processes is important in regions like Missouri, where a claypan layer under a pervious surface soil controls subsurface hydrology (Jiang et al. 2007) and leads to the development of saturated areas at the base of the hillslope.

In order to represent variable-source area hydrology, water flow through a hillslope is represented by multiple, hillslope segments called overland flow elements (OFEs). The length of these OFEs is fixed by the user based on natural slope-breaks in the topography of the hillslope or on *a priori* information defined by the user. WEPP-UI outputs include overland flow, lateral flow, percolation, and sediment loss for each OFE. With the site-specific information gained from running different management 'scenarios', a soil and water manager can assess the impact of multiple combinations of BMPs along the length of the hillslope. Potential BMP targeting scenarios may focus on varying tillage practices, crop rotations, buffer size and presence, as well as precision application of pesticide to specific sections of the hillslope when combined with the pesticide module water quality analysis.

WEPP Pesticide Module Description

In this study, we use WEPP-UI outputs as inputs for the pesticide module. The pesticide module is adapted from Steenhuis & Walter (1980) and can predict both dissolved and particulate bound pesticide concentrations in runoff water and in the subsoil. Pesticides are added to the mixing layer (Figure 1) on an application date specified by the user. When it rains, pesticide can be lost with overland flow, lateral flow, and percolating water in proportion to the concentration of

dissolved pesticide within the mixing layer. Dissolved pesticide loss via shallow percolation is added to the mass of pesticide in the deeper transition layer. Dissolved pesticide lost with deep percolation and lateral flow is carried out of the OFE. Dissolved pesticide transport in overland flow and lateral flow from up-slope OFEs is added to the down-slope OFE. Similarly, particulate pesticide lost with overland flow can be transported to down-slope OFEs. See Appendix A for further description of the pesticide module.
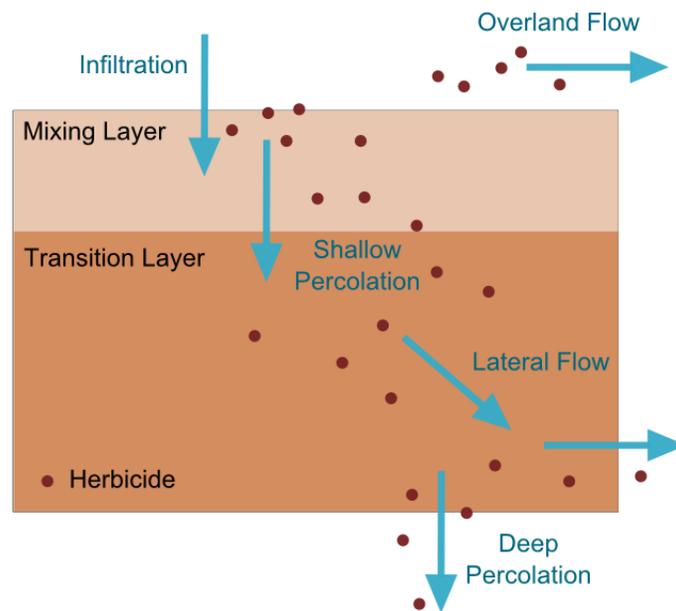
Figure 1. Schematic of hydrological transport processes included in the WEPP-UI pesticide module. Circles indicate dissolved or sediment bound pesticides and the blue arrows indicate water flows.

Site Description

The model was tested in the Goodwater Creek Experimental Watershed (GCEW) of northeastern Missouri. Mulch till (MT) and no-till (NT) management in corn-soybean rotation were evaluated using a randomized complete block design as described in Ghidey et al. (2005). Each plot was 189 m by 18 m and surrounded by a 1.5 m wide berm. Slopes ranged from 0-2% and soils were classified predominantly as poorly drained Mexico silt loam soil (fine smectic, mesic Vertic Epiaqualf) having a claypan horizon at depths of 20, 10, and 30 cm below the soil surface for top-slope, mid-slope, and toe-slope positions, respectively. See Ghidey et al. (2005), Jung et al. (2007), and Jiang et al. (2007) for a detailed description of soil properties and landscape characteristics for the GCEW field site.

The long-term average annual rainfall (1970-2003) for the GCEW was 944 mm (Ghidey et al. 2005). Years 1998 and 2001 were above the long-term average annual rainfall while 1997, 1999, and 2001 were below (Figure 2). Year 1999 was especially dry and had very few observed runoff events (Ghidey et al. 2005).
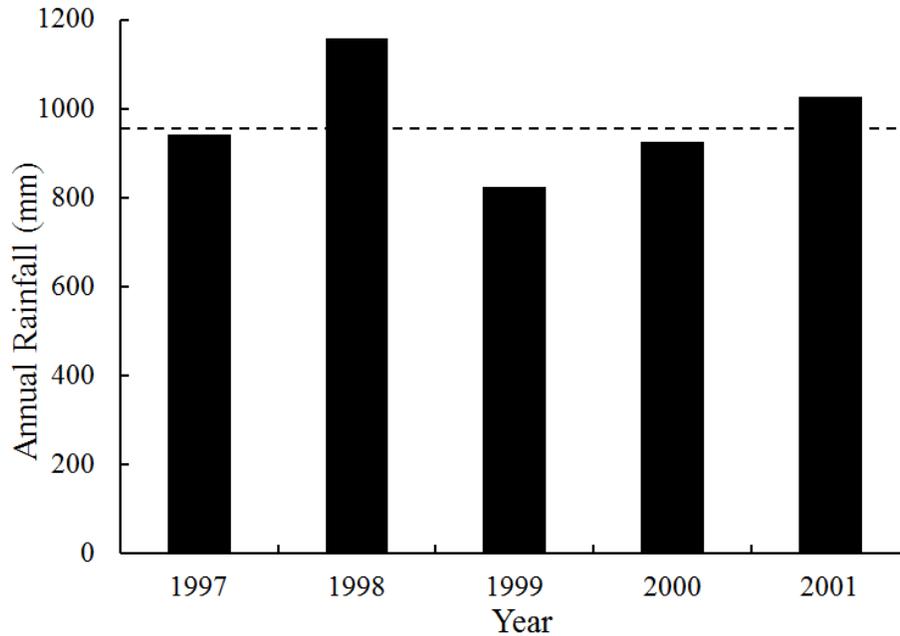
Figure 2. Annual rainfall for mulch till and no till plots (1997-2001) used in simulations 1 through 4. Annual, long-term average rainfall (1970-2003) is indicated by the dotted line at 944 mm.

Model Set-Up

WEPP-UI (Brooks & Boll 2011) was used to compare MT and NT simulations to field data collected by Ghidey et al. (2005). Measured hillslope characteristics were averaged by tillage type and used as inputs for each simulation. Specifically, top-slope, mid-slope, and toe-slopes were set at 0.5, 1.5, and 0.5 %, respectively. Slope lengths were set at 50, 90, and 50 m, for top-slope, mid-slope, and toe-slopes lengths, respectively. Soil bulk density, hydraulic conductivity, field capacity moisture content, wilting point moisture content, percent sand, percent clay, cation exchange capacity, and percent organic matter were taken from the SSURGO soils database

(USDA-NRCS 2003) for a Mexico silt loam as reported in Table 1. Detailed soils data were collected for the GCEW site (Jiang et al. 2007; Jung et al. 2007) but the authors' motivation was to evaluate the model using widely accessible data sources. Restrictive claypan layer depth was set at 20, 20, and 30 cm below the soil surface for the top-slope, mid-slope, and toe-slope, respectively. It should be noted that the simulated mid-slope restrictive layer depth was increased from 10 to 20 cm in order to accommodate tillage routines in the WEPP-UI model. It was assumed that roots could not penetrate the dense clay layer. Vertical saturated hydraulic conductivity below the claypan was calibrated by comparing simulated runoff volume to observed average runoff volume.

Table 1. Mexico silt loam soil properties used in mulch till and no till simulations.

Depth (mm)	ρ_b (g/cm^3)	$k_{sat,v}$[a] (mm/h)	$k_{sat,h}$ Factor	FC	WP	Percent Sand	Percent Clay	Percent OM	CEC (meq/ 100g)
180	1.45	17.4[b]	50	0.29	0.14	9.1	19.5	1.9	13
200	1.55	9.72	50	0.33	0.2	9.4	29.9	1.2	14

Abbreviations: ρ_b (bulk density), k_{sat} (hydraulic conductivity), FC (field capacity water content), WP (wilting point water content), OM (organic matter), CEC (cation exchange capacity)
[a] For MT and NT plots ksat,v below 200 mm was set to 0.05 mm/h.
[b] ksat,v = 17.4 mm/h for MT simulations and 32.4 mm/h for NT simulations.

Simulations mimicked application treatments as described in Ghidey et al. (2005). Application and planting occurred on the same day for all MT and NT plots (Table 2) and atrazine application rate was kept at 2.24 kg/ha for all plots and throughout all five years of the study as reported in Ghidey et al. (2005). Atrazine adsorption coefficient (K_{OC}) was set equal to 200

cm^3/g. Atrazine half-life for the mixing and transition layers of MT and NT plots was set equal to 15 days. Despite differences in decay potential, the half-lives of both layers were assumed equal to ensure model simplicity. See Table 4 for additional model parameters. Because Ghidey et al. (2005) only collected atrazine samples during the corn growing season for alternating plots, we simulated two management schemes for each tillage type (Table 3) and then merged simulation pairs in order to compare MT and NT field data against model outputs for a single year corn rotation without soy years. Similarly, WEPP-UI simulated hydrology was taken from years when the plots were in corn. Atrazine loads (flow-weighed), runoff volumes, and regional climate data were provided by the USDA's Agricultural Research Service – Cropping Systems and Water Quality Research Unit in Colombia, Missouri (Sadler et al. 2006) through the STEWEARDS database (www.ars.usda.gov/watersheds/stewards, accessed on September 15, 2011, Steiner et al. 2009).

Table 2. Application and planting dates given in Ghidey et al. (2005).

Year	Tillage	Plot Numbers	Application and Planting Date
1997	MT	19, 22	May 13
	NT	13, 24	May 12
1998	MT	11, 23	May 21
	NT	18, 21	May 21
1999	MT	19, 22	Jun 03
	NT	13, 24	Jun 03
2000	MT	11, 23	May 16
	NT	18, 21	May 16
2001	MT	19, 22	May 16
	NT	13, 24	May 16

Table 3. Description of WEPP-UI simulations for mulch till and no till runs.

Simulation	Number of Years in Simulation	Tillage	Rotation
1	5	MT	Corn-Soy
2	5	MT	Soy-Corn
3	5	NT	Corn-Soy
4	5	NT	Soy-Corn

Model Evaluation

We compared event-based runoff and dissolved atrazine losses for observed and simulated MT and NT plots from the base of the hillslope. Since runoff and atrazine were only measured during the growing season (May 1 to August 31) for plots in corn rotation, we limit our comparison between simulated and observed results to this time period and crop type. Simulated and average observed events were plotted against each other; a point closest to the one-to-one line indicates a

perfect fit. Simulated points were excluded from analysis if no corresponding observation was available for a particular event. Two different performance metrics were used to evaluate the WEPP-UI pesticide module: the Nash-Sutcliffe Efficiency-NSE (Nash & Sutcliffe 1970) and coefficient of determination (R^2). NSE, given by equation 1, is a dimensionless statistic that depicts how well the observed versus simulated data fits on a one-to-one line. The obs_i variable represents an observed value taken at time i while the sim_i variable represents a simulated value at the same time i. A NSE value of 1.0 indicates a perfect fit and a value below 0 indicates that the average observed value is a better predictor than the simulated value. According to the recommendations of Moriasi et al. (2007), models can be judged 'satisfactory' if NSE > 0.50. The R^2 value indicates the tightness of spread around the linearly regressed line.

$$NSE = 1 - \frac{\sum_{i=1}^{n}(obs_i - sim_i)^2}{\sum_{i=1}^{n}(obs_i - \overline{obs})^2} \qquad (1)$$

SECTION 3: RESULTS AND DISCUSSION

Comparison of Simulations and Observations: Hydrology

Overall the WEPP model predicted that 60% of the total precipitation received on the MT plots would evaporate, 16% would be moved off the plots as surface runoff, 0.02% as interflow, and 24% of the rainfall would be lost to deep percolate through the claypan (Figure 3a). Similar percentages were simulated for the NT plots 61, 15, 0.04, 24 % for evaporation, surface runoff, interflow and deep percolation (Figure 3b).
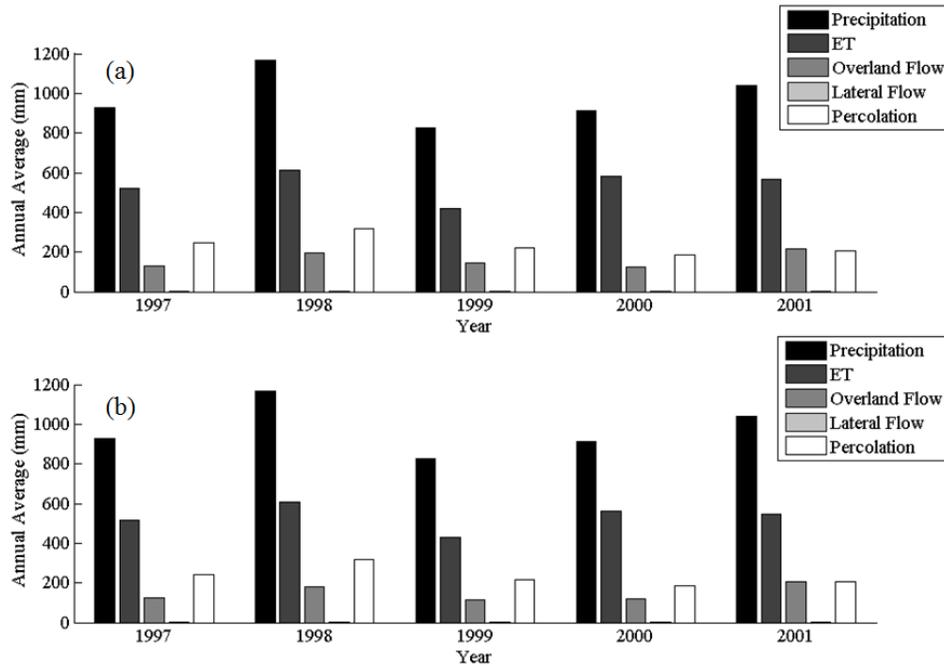
Figure 3. Average annual precipitation, evapotranspiration (ET), overland flow, lateral flow, and percolation for all hillslope sections in (a) mulch till and (b) no till simulations.

Runoff was generated during wet periods when there was significant rainfall (Figure 4). MT plots had 6 observed runoff events on average per year with the minimum and maximum number of events occurring in 1999 and 2001, respectively. NT plots had 5 observed runoff events on average per year with the minimum and maximum number of events occurring in 1999 and both 1998 and 2001. Generally, the predicted values were close to the (sometimes large) range of the two runoff values for replicate field plots (Figure 4 and 5). Runoff simulations indicate a reasonable fit with observed data averaged between the two replicate plots; the NSE for runoff from MT is 0.84 and the NSE for runoff from NT is 0.79 (Table 5). R^2 is above 0.5 for both tillage types (Table 5).
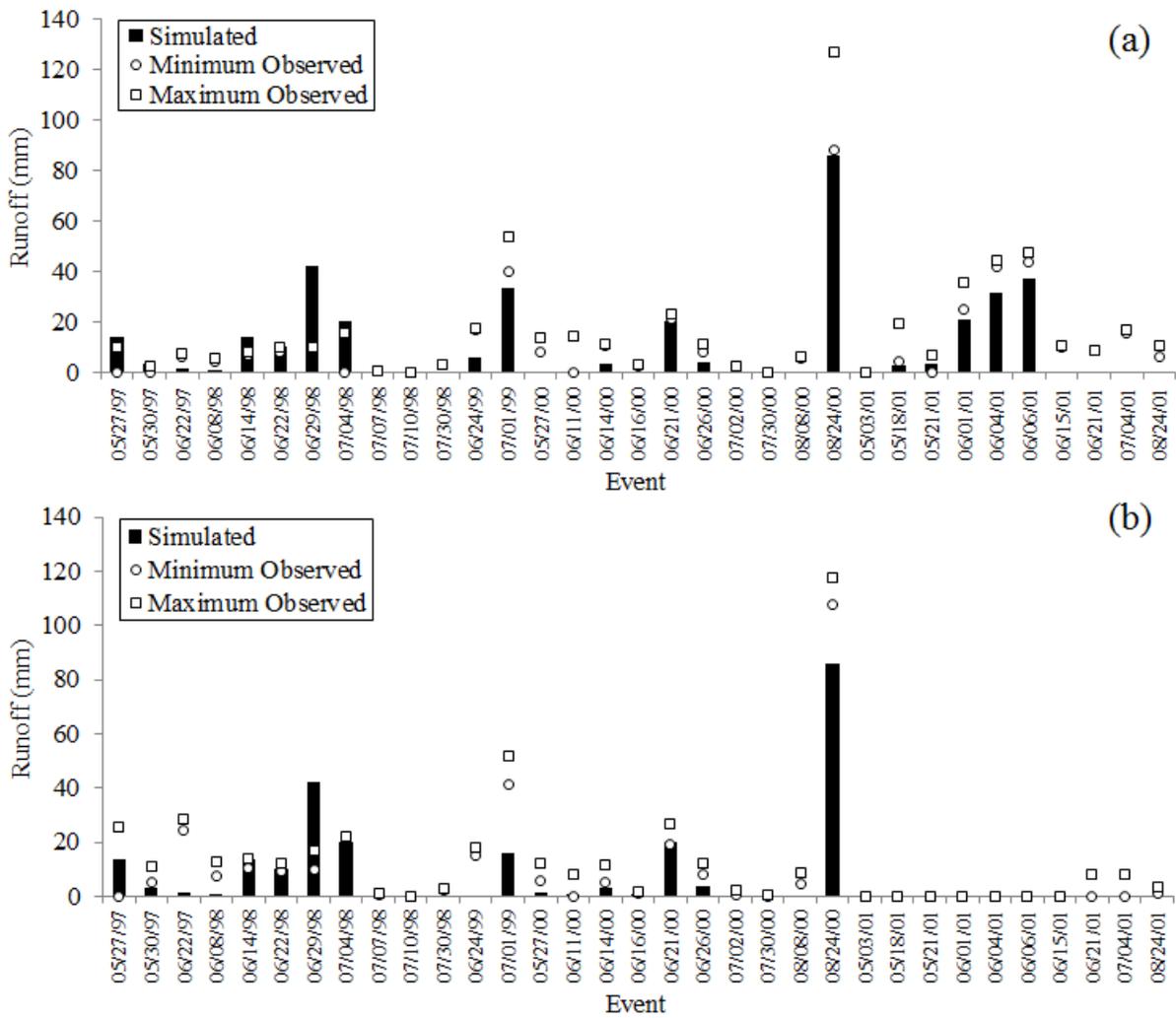
Figure 4. Comparison between seasonal (planting to harvest) simulated and observed runoff over time for (a) mulch till and (b) no till plots. Simulated results are shown as a black bar, minimum observed results are shown as open circles, and maximum observed results are shown as open squares.
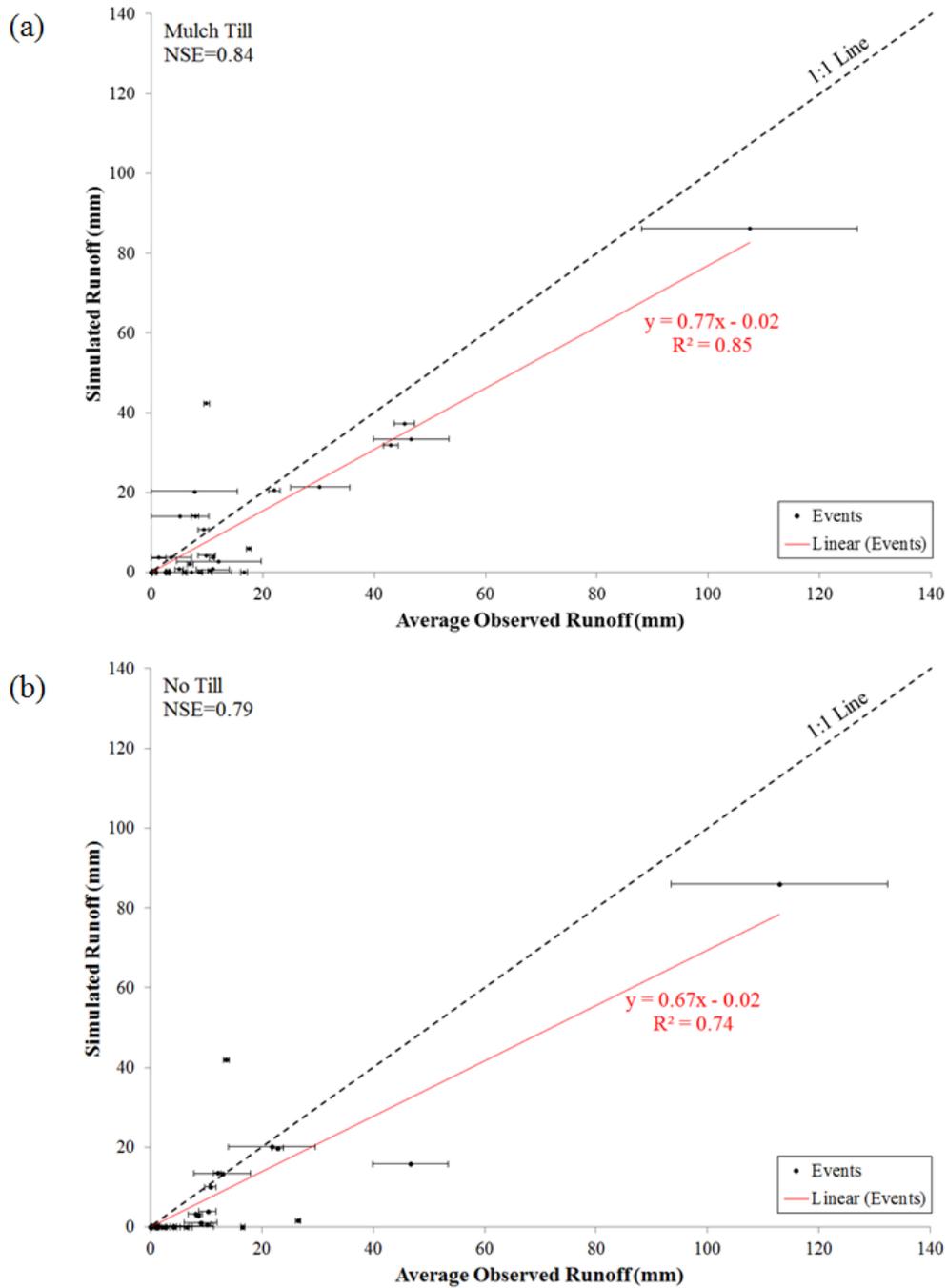
Figure 5. Comparison between seasonal (planting to harvest) simulated and average observed runoff (planting to harvest) in (a) mulch till, and (b) no till plots. Right and left horizontal bounds indicate minimum and maximum field observations. Linear regression line, equation, and R^2 value are show in red.

Jiang et al (2007) calculated hydraulic conductivities of approximately 10 mm/h and 50 mm/h for MT and NT, respectively (Jiang et al., 2007). The soil survey data used for both WEPP-UI simulations gave saturated hydraulic conductivities of 17 and 32 mm/h for the top 18 cm of soil for MT and NT plots, respectively (Table 1). According to break-point precipitation data collected from the field site during the five year study period, the median, mean, and maximum rainfall intensities were equal to approximately 3, 6, and 168 mm/h, respectively. The median and mean are well below the saturated hydraulic conductivities used in WEPP-UI (Table 1). Further analyses of rainfall intensities indicate that saturation excess is the dominant runoff generating mechanism 94 and 98 percent of the time for MT and NT plots, respectively (data not shown). However, the high maximum rainfall intensity over the duration of the study period also points to the importance of runoff generation due to infiltration excess processes; by compliment, 6 and 2 percent of the time for MT and NT plots, respectively.

The spatial differences in the depth to the claypan are likely to describe the large variation in runoff between replicate plots (Figure 5). There are many instances when observed data from one plot has runoff while the companion plot has none. These differences are especially evident during dry periods of the year. During wet periods, the loss of storage depends mostly on evaporation losses. Because evaporation loss is independent of the restrictive layer depth, depletion (i.e the difference between saturation and soil moisture content) should be relatively constant between plots when the soil is wet.

The subsurface lateral flow component was negligible because of the flat slopes (Figure 3). Since the simulated soil depth was shallow over the claypan horizon (20 cm for the top and mid-slope

OFEs, and 30 cm for the toe-slope OFE), and neither much lateral flow nor percolation occurred, the soil could easily fill up with water under a heavy rainfall event. Thus, the runoff is essentially equal to the amount of rainfall minus the volume of storage space in the soil. A simple validation model based on this idea performed similarly well to predict runoff values for the GCEW plots (results not shown). This is not to say WEPP-UI is not useful in this situation. On the contrary, WEPP-UI simulated this situation well and can accommodate more complicated cases where lateral flow influences the drainage at the top-slope and saturation at the toe-slope.

WEPP-UI estimates for evapotranspiration (ET) from MT and NT plots agreed relatively well with annual SWAT based ET estimates for the region (Baffaut, personal communication). SWAT generated ET was approximately 450 mm/y while WEPP-UI average ET for the five year study period was 540 and 532 mm/y for MT and NT plots, respectively. Our findings suggest that WEPP-UI can partially account for the increased water retention of NT fields covered with crop residue. However, the under-estimated runoff values observed in NT simulations may indicate misrepresentation of antecedent moisture conditions. Studies such as Enz et al. (1988) and Wolgast-Broberg (2002), show that large amounts of residue cover associated with NT farming may decrease evaporation from the soil when the plants are small. Since less water is needed to saturate the soil when it is already wet, the remaining water is lost as runoff.

Comparison of Simulations and Observations: Pesticide Transport

Dissolved atrazine concentrations for observed events were compared to simulated concentrations during the growing season from planting to harvest for both MT and NT plots (Figure 6 and 7). Although the differences in runoff was small between runoff generated from

28

NT and MT (compare Figure 4a and 4b), observed pesticide loss was greater for NT than MT. In

order for the model to simulate this we used a mixing depth for the mulch tillage equal to 12 cm

and for the no till profile the mixing layer was set equal to 2 cm for each hillslope section. In

previous studies in upstate New York (Steenhuis & Muck 1988) where the surfaces soils were

highly conductive surface due to the presence of biopores, the depth of the mixing layer was

equal to the depth above the impermeable layer (Table 4). Therefore, 12 cm is a reasonable

estimate for the variable restrictive layers found in Missouri. The 2 cm depth mixing layer for

NT is similar to other studies where either the pores in the topsoil are not inter-connected or the

soil was initially unsaturated (Steenhuis & Walter 1980; Tong et al. 2010).

Table 4. Model settings for mulch till and no till simulations.

Parameter Name	MT	NT
Mixing Layer Depth, h_{hix} (m)	0.12	0.02
K_{OC} (cm^3/g)	200	200
Atrazine half-Life, t_{half} (days)	15	15
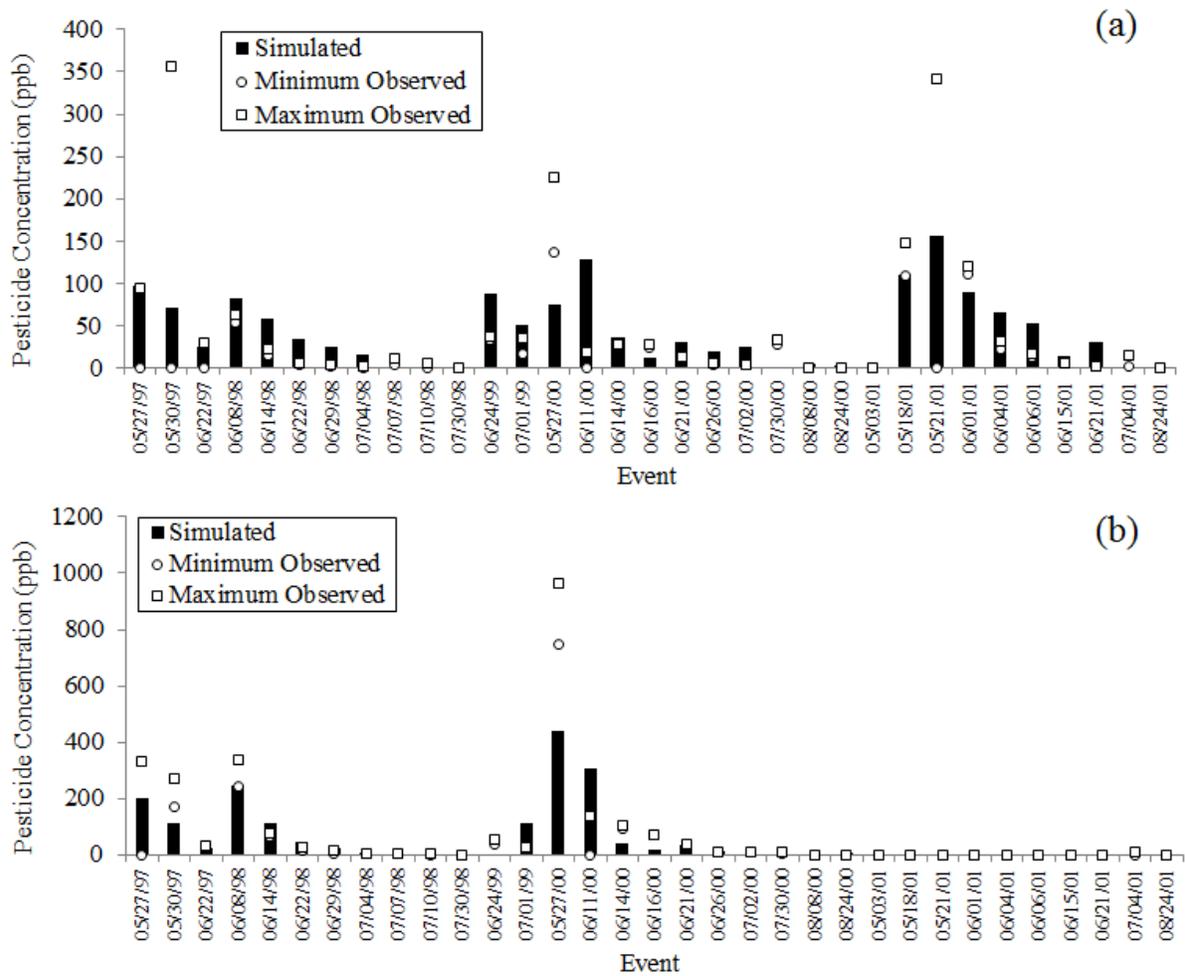
Figure 6. Comparison between seasonal (planting to harvest) simulated and observed atrazine concentration over time for (a) mulch till and (b) no till plots. Simulated results are shown as a black bar, minimum observed results are shown as open circles, and maximum observed results are shown as open squares.

30
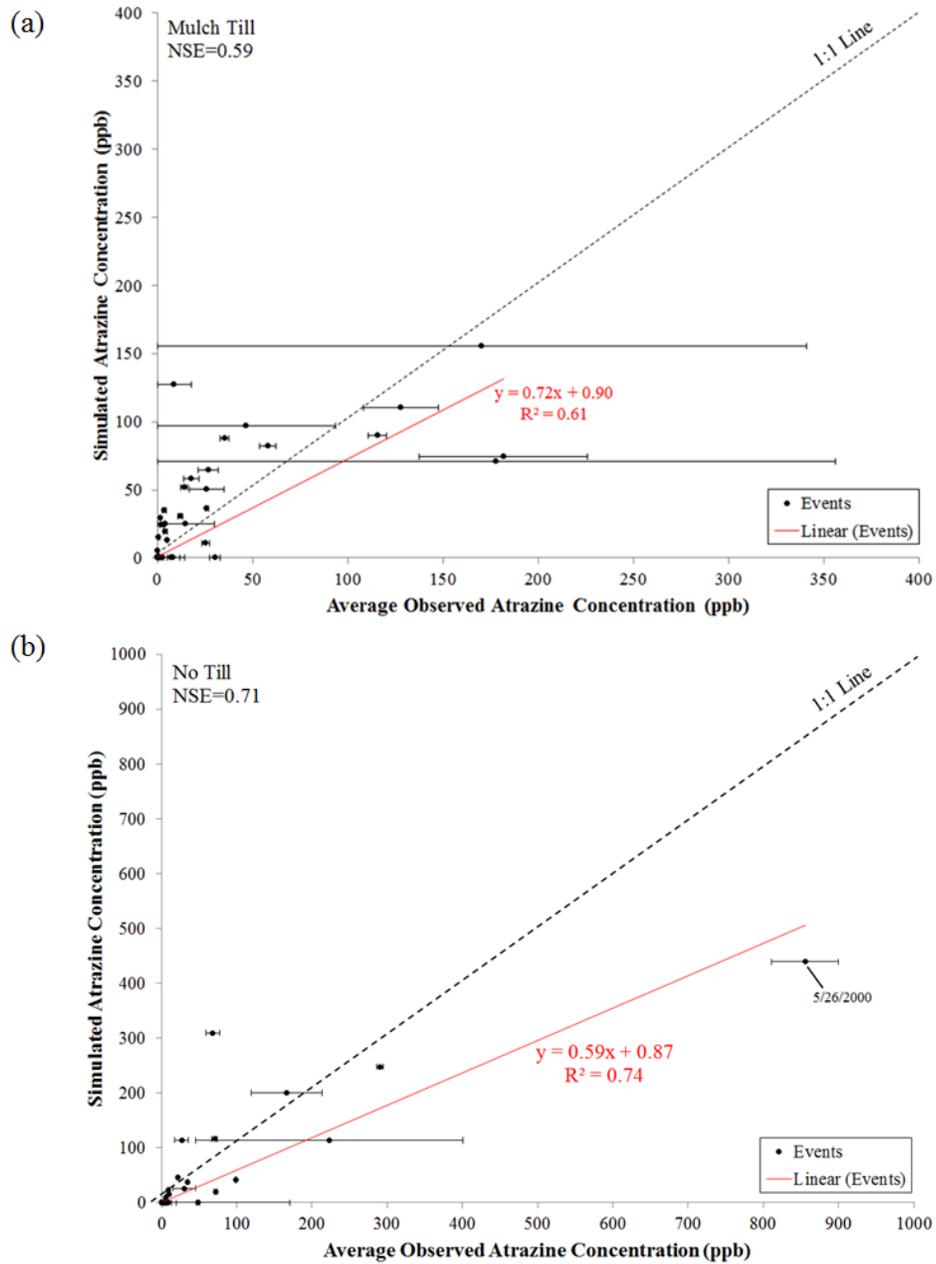
Figure 7. Comparison between seasonal (planting to harvest) simulated seasonal and average observed dissolved atrazine lost in (a) mulch till, and (b) no till plots. Right and left horizontal bounds indicate minimum and maximum field observations. Linear regression line, equation, and R2 value are show in red.

Atrazine concentrations for MT and NT plots were reasonably well predicted (Figures 6 and 7). NSEs were 0.59 and 0.71 for MT and NT plots, respectively (Table 5). For MT plots, the largest atrazine concentrations were slightly under-predicted during the first rainfall event after application (in May) for all five simulation years (see red squares in Figure 8). Given this result, it is obvious that the pesticide may have not been fully mixed into the 12 cm mixing layer assumed for MT simulations. NT simulations show some sensitivity to pesticide losses during the early stages of the growing season, but this pattern is not as pronounced when compared to MT. NSE for atrazine losses from NT plots was 0.71, which indicates a relatively good fit (Table 5). Ghidey and all could not explain the high concentration reported on 5/26/2000 (see Figure 7b) which was under-predicted in by the pesticide module. R^2 values for atrazine concentration lost from MT and NT plots also indicates a relatively good fit (Table 5).
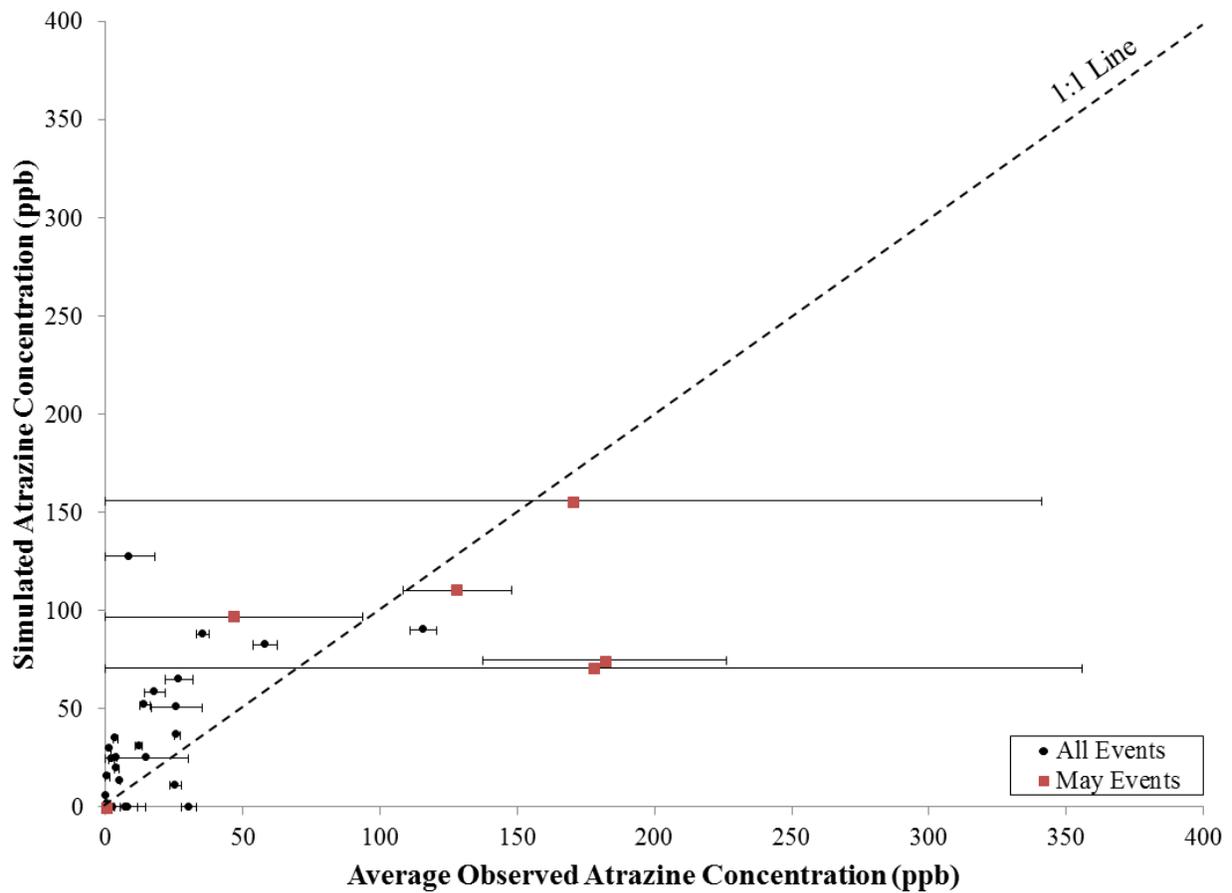
Figure 8. Comparison between seasonal (planting to harvest) simulated and average observed dissolved atrazine lost from mulch till plots. All events occurring in May are shown as red squares. Right and left horizontal bounds indicate minimum and maximum field observations.

Table 5. Model evaluation metrics for dissolved atrazine loads from mulch till and no till plots.

Runoff (mm)		
Tillage	NSE	R^2
MT	0.84	0.85
NT	0.79	0.74

Atrazine Loss (ppb)		
Tillage	NSE	R^2
MT	0.59	0.61
NT	0.71	0.74

Ghidey et al. (2005) only reported the dissolved pesticide loss for MT and NT plots during the growing season (planting to harvest). Given the predicted sediment losses for MT and NT plots, the majority of pesticide lost is in the dissolved form; approximately 99% for both NT and MT plots. Hence any soil and water conservation practice that conserves soil only is not effective for reducing pesticide loss to streams in this region.

Effect of Best Management Practices

To show that the coupled WEPP and pesticides module could be used to evaluate the effect of buffer strips and precision application rate on atrazine loss. We ran the WEPP-UI pesticide module for the scenarios presented in Table 6. All three slopes (top, mid, and toe) were set at a 2% slope and are in corn/soy rotation.

Table 6. WEPP-UI pesticide module scenario description.

Scenario Number*	Depth to Restrictive Layer (cm)	Tillage	Buffer Length (m)	Number of Simulation Years	Application Rate (kg/ha)**	Application OFE
1	20	NT	0	10	2.24	All
2	20	NT	10	10	2.24	All
3	20	NT	0	10	2.24	Top
4	20	NT	0	10	2.24	Middle
5	20	NT	0	10	2.24	Bottom

*All scenarios were simulated for a corn/soy rotation with 2% slopes.
** Application rate for atrazine.

Figure 9a shows the annual average percent pesticide lost via various hydrological processes (dissolved pesticide lost in overland flow, particulate pesticide lost in overland flow, dissolved pesticide lost in lateral flow, and dissolved pesticide lost in percolation) for each hillslope section for scenario 1 and 2. The results of these scenarios show that a large percentage of pesticide are lost in lateral flow from the top of the hillslope (Figure 9a). Also, nearly half of particulate pesticide lost in overland flow comes from the toe-slope (Figure 9a). Looking at Figure 9b, we see the positive impact of the buffer; it is able to trap all of the particulate pesticide lost with overland flow.
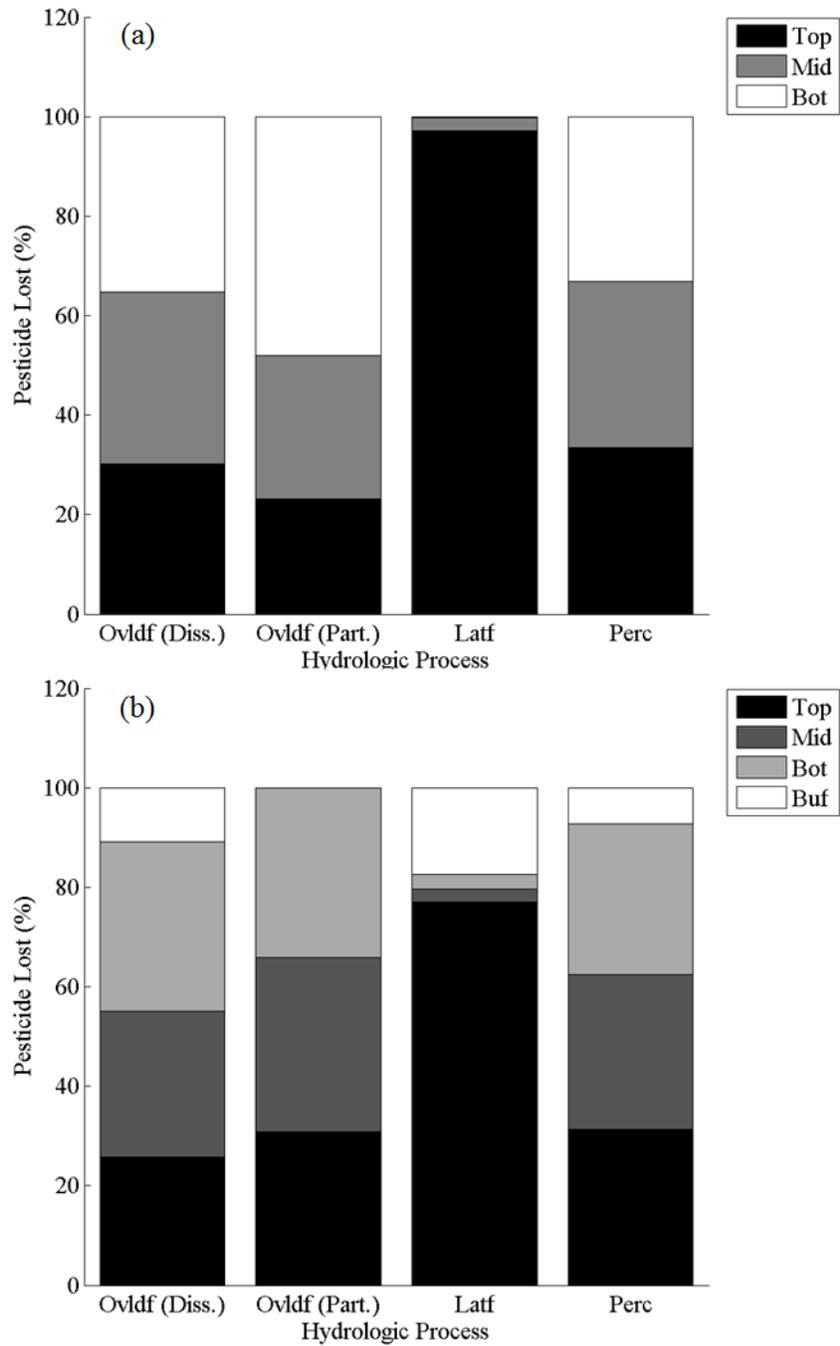
Figure 9. Average annual percent pesticide lost with various hydrologic processes for each section of the hillslope for (a) scenario 1 and (b) scenario 2 as specified in table 6.

Scenarios 3 through 5 demonstrate the impacts of precision pesticide application on water quality. Results show large losses of dissolved pesticide in overland flow originating from the application OFE (Table 7). The largest magnitude of dissolved pesticide is lost with overland flow when atrazine is applied only to the toe-slope section of the hillslope, which is what we would expect given the propensity for water to accumulate at the base of the hillslope (Table 7). Soil and water managers can explore additional scenarios using the WEPP-UI pesticide module. Furthermore, they can use outputs such as these to make scientifically-based decisions on vegetated buffer strip installation and precision pesticide application.

Table 7. Average atrazine losses with various hydrological processes for BMP scenario 1 through 5 (see table 6 for application rates).

	Pesticide Loss (kg/ha)			
	Overland Flow (Dissolved)	Overland Flow (Particulate)	Lateral Flow	Percolation
Scenario 1				
Top	8.0E-02	3.7E-04	3.3E-04	5.5E-03
Middle	9.1E-02	4.6E-04	8.9E-06	5.5E-03
Bottom	9.3E-02	7.6E-04	7.4E-07	5.5E-03
Scenario 2				
Top	8.0E-02	3.7E-04	4.9E-04	8.0E-03
Middle	9.2E-02	4.2E-04	1.6E-05	8.0E-03
Bottom	1.1E-01	4.0E-04	1.9E-05	7.7E-03
Buffer	3.4E-02	0	1.1E-04	1.8E-03
Scenario 3				
Top	8.0E-02	3.7E-04	3.3E-04	5.5E-03
Middle	5.7E-04	2.4E-05	6.2E-08	3.4E-04
Bottom	3.0E-04	3.9E-05	3.2E-09	1.1E-04
Scenario 4				
Top	0	0	0	0
Middle	1.4E-01	6.5E-04	3.1E-04	5.1E-03
Bottom	1.7E-03	1.0E-04	2.3E-08	5.4E-04
Scenario 5				
Top	0	0	0	0
Middle	0	0	0	0
Bottom	1.8E-01	1.0E-03	3.1E-04	4.8E-03

SECTION 4: CONCLUSIONS

Tools are needed to assist soil and water managers as they select appropriate BMPs for a wide variety of landscapes, climates, tillage practices, crops, and contaminant types. The purpose of this study was to test the WEPP-UI based pesticide transport module, which will be incorporated into a recently developed web-based tool called the Hydrological Characterization Tool (HCT) available at: http://wepp.ag.uidaho.edu/cgi-bin/HCT.pl (Brooks & Boll 2011). The pesticide module requires few inputs and predicts runoff and dissolved atrazine loss reasonably well for NT plots in the GCEW. Further hillslope scale field experiments are needed to test the WEPP-UI pesticide module for a wider range of regions and contaminant types. This is especially true given the model's ability to predict sediment bound pesticide loads as well as the impact of buffer strips and precision application. According to Reichenberger et al. (2007), the ability to predict particulate pesticide loss is especially important for highly adsorbed pesticides; those with an adsorption partition coefficient (K_{OC}) value greater than 1000 cm^3/g (e.g. glyphosate, and malathion).

In addition to the pesticides module, phosphorus and nitrogen modules will also be coupled with the HCT. Specifically, these will further assist soil and water managers looking to target BMPs while accurately (and simply) characterizing the dominant hydrologic flow paths in a given region. We also recognize the importance of identifying other barriers to BMP implementation. Future studies may consider incorporating economic and social factors into the HCT for BMP targeting.

APPENDIX A

Pesticide Model Description

The pesticide module was adapted from Steenhuis and Walter's (1980) steady-state model, which can predict dissolved and particulate forms of pesticide transport via runoff from the mixing layer. Dissolved pesticide concentration is calculated from the mass of pesticide in the mixing layer as:

$$C_{D,t} = \frac{M_{m,t}}{h_{mix}(\theta_s + \rho_s k)}$$

where

$C_{D,t}$ = dissolved pesticide concentration at time t

$M_{m,t}$ = pesticide mass in the mixing layer at time t

h_{mix} = mixing layer depth

θ_s = saturated moisture content

ρ_s = soil bulk density

k = adsorption partition coefficient.

The total water depth (i.e., volume) in the mixing layer is equal to

$$w_t = h_{mix}(\theta_s - \theta_t) \qquad \text{(A1)}$$

where

h_{mix} = mixing layer depth

θ_s = soil saturated water content

θ_t = soil water content at time t.

The effective water depth defined as the apparent amount of water needed to obtain the dissolved pesticide concentration in the mixing layer for a pesticide with adsorption partition coefficient k is given as

$$w_e = h_{mix}(\theta_s + \rho_s k) \qquad \text{(A2)}$$

where

k = adsorption partition coefficient as given by equation A3

Where the adsorption partition coefficient is obtained by multiplying the published value of the organic carbon adsorption partition coefficient (K_{OC}) by the organic carbon content of the soil (Sinkevich et al. 2005).

$$k = K_{OC}OC \qquad \text{(A3)}$$

where

K_{OC} = Organic carbon pesticide adsorption coefficient

OC = percent organic carbon.

The effective overland flow due to sediment loss is calculated as

$$Q_{s,t} = Q_t \left(1 + \left(\frac{S_t k}{1-S_t}\right)\rho_w\right) \qquad \text{(A4)}$$

where

Q_t = depth of overland flow at time t

S_t = daily sediment fraction at time t

ρ_w = density of water.

The total mass of pesticide loss in the overland flow can be calculated as

$$P_{OT,t} = M_{m,t}\left(1 - \exp\left(-\frac{Q_{s,t}}{h_{mix}(\theta_s + k\rho_s)}\right)\right) \qquad \text{(A5)}$$

where

R_i = the rainfall depth at time t.

The mass of dissolved and particulate pesticide loss, given respectively by equations A6 and A7, are taken as a proportion of the total pesticide lost.

$$P_{OD,t} = \frac{P_{OT,t}(1-S_t)}{1+S_t(k\rho_w-1)} \qquad \text{(A6)}$$

$$P_{OP,t} = \frac{P_{OT,t}S_t k\rho_w}{1+S_t(k\rho_w-1)} \qquad \text{(A7)}$$

Dissolved pesticide lost in shallow percolation is given as

$$P_{SP,t} = M_{m,t}(1 - \exp\left(-\frac{R_t - ET_t}{h_{mix}(\theta_s + k\rho_s)}\right) \qquad \text{(A8)}$$

where

ET_t = soil water lost due to evapotranspiration at time t.

Dissolved pesticide lost in deep percolation is given as

$$P_{DP,t} = M_{T,t}\left(1 - \exp\left(-\frac{DP_t}{h_{mix}(\theta_s + k\rho_s)}\right)\right) \qquad \text{(A9)}$$

where

$M_{T,t}$= pesticide mass in the transition layer at time t

DP_t= deep percolation depth at time t.

Dissolved pesticide lost in lateral flow is given as

$$P_{L,t} = M_{T,t}\left(1 - \exp\left(-\frac{L_t}{h_{mix}(\theta_s + k\rho_s)}\right)\right) \qquad \text{(A10)}$$

where

L_i= lateral flow depth at time t.

In addition to losses from hydrological processes, pesticide can be lost from the mixing and transition layers due to chemical degradation. This follows an exponential decay as suggested by Ghidey et al. (2005) and is given here as

$$\frac{dM}{dt} = -cM \qquad \text{(A11)}$$

where

$$c = \frac{0.69}{t_{half}}$$

t_{half}= half-life of the pesticide (days).

If we integrate equation A11 with respect to time we get

$$M_{m,t} = (M_{m,t-1}) \exp\left(\frac{-0.69\, t_{step}}{t_{half}}\right) \qquad \text{(A12)}$$

where

$M_{m,i-1}$ = mass of pesticide remaining in the mixing layer at time t-1 (the previous time)

t_{step} = daily time step (equal to 1 day).

Pesticide degradation for the transition layer is given by a similar equation. For further description of the pesticide module see (Steenhuis and Walter 1980).

APPENDIX B

MatLab Code Description and Input Formats

The MatLab code for the pesticide module is structured into five main files. The main pesticide program file ('program_pesticide.m'), the two sub pesticide program files depending on whether a vegetated buffer is present in the run ('sub_program_pesticide.m' and 'sub_program_pesticide_buff.m'), and the two pesticide function files also depending on whether a vegetated buffer is present in the run ('dailyofePestsimTrans.m' and 'dailyofePestsimTransBuff.m'). Seven different text file inputs are needed to initiate each run. Of the seven files, three are from WEPP, including the water balance file ('wat*.txt'), the OFE line summary file ('elem*.txt'), and the plant file ('crop*.txt'). The '*' indicates the number of the WEPP run. The four remaining input files include the user defined input file ('userinput*.txt'), the fixed input file ('fixedinput*.txt'), the file name path file ('filename*.txt'), and the scheduling file ('sched*.txt') for each run. The format for the four non-WEPP input files is shown below (Table AB1 – Table AB4) where OFE 1, OFE 2, OFE 3, and OFE4 correspond to the top, mid, bottom, and buffer slopes, respectively. Only the numerical values highlighted by the text box are required in the corresponding text files.

Table B1. User input file format. Only the numerical values highlighted by the text box are required in the corresponding text files.

A Row #	B User Input (userinput*.txt)	C ofe 1	D ofe 2	E ofe 3	F ofe 4 (buffer)	C
1	Run #	1	1	1		
2	# years in simulation	10	10	10		
3	number of OFE	3	3	3		
4	Depth to Restrictive Layer (cm)	20	20	20		
5	Bulk Density (g/cm^3)* (crop 1)	1.45	1.45	1.45		
6	Bulk Density (g/cm^3)* (crop 2)	1.45	1.45	1.45		
7	Bulk Density (g/cm^3)* (crop 3)	0	0	0		
8	Bulk Density (g/cm^3)* (crop 4)	0	0	0		
9	Theta S (crop 1)	0.45	0.45	0.45		
10	Theta S (crop 2)	0.45	0.45	0.45		
11	Theta S (crop3)	0	0	0		
12	Theta S (crop4)	0	0	0		
13	Theta FC (crop 1)	0.29	0.29	0.29		
14	Theta FC (crop 2)	0.29	0.29	0.29		
15	Theta FC (crop 3)	0	0	0		
16	Theta FC (crop 4)	0	0	0		
17	Theta WP (crop 1)	0.14	0.14	0.14		
18	Theta WP (crop 2)	0.14	0.14	0.14		
19	Theta WP (crop 3)	0	0	0		
20	Theta WP (crop 4)	0	0	0		
21	OM % (crop 1)	1.90	1.90	1.90		
22	OM % (crop 2)	1.90	1.90	1.90		
23	OM% (crop 3)	0	0	0		
24	OM% (crop 4)	0	0	0		
25	Nutrients and herbicide corporated? (yes=1,no=0)	0	0	0		
26	KOC (cm^3/g) (crop 1) = corn	160	160	160		
27	KOC (cm^3/g) (crop 2) = soy	160	160	160		
28	KOC (cm^3/g) (crop 3)	0	0	0		
29	KOC (cm^3/g) (crop 4)	0	0	0		
30	half-life (days) (crop 1) = corn	60	60	60		
31	half-life (days) (crop 2) = soy	60	60	60		
32	half-life (days) (crop 3)	0	0	0		
33	half-life (days) (crop4)	0	0	0		
34	STP (mg/kg)	0	0	0		
35	Nitrate at the surface (ppm)	0	0	0		
36	Soil N (ppm) (if they don't enter something please use 9999)	0	0	0		
37	Buffer (yes=1, no=0)	0	0	0		
38	Length ofe (m)	100.00	100.00	90.00		
39	Width ofe (m)	100.00	100.00	100.00		
40	Berm ofe (m)	0	0	0		
41	Mixing layer depth (m), initial for application day	0.05	0.05	0.05		

Table B2. Fixed input file format. Only the numerical values highlighted by the text box are required in the corresponding text files.

Row #	Fixed Input (fixedinput*.txt)	Only one column (same for all OFE's)
1	Buffer mixing layer depth (m)	0
2	DP Immobilization rate (days)	10
3	Rxn constant for DP and PP (m^3/kg)	0.15
4	P model factor change rate for 10 degC change in soil temp (no units)	20
5	Reference DP plant/soil complex export coefficient (ppm)	1.00E+00
6	P model decay time step (days)	1
7	P model reference soil temp (degC)	20
8	Fraction of active N in manure	0.02
9	Fraction of available mineral N	0.08
10	Fraction of N transferred betw. orgNstab and orgNact pools	1.00E-05
11	Nitrate-ammonium ratio	8
12	N model factor change rate for 10 degC change in soil temp (no units)	3
13	N model reference soil temp (degC)	20
14	Half life of orgNact (days)	182
15	Half life of orgNfrsh (days)	30
16	Half life of orgNstab (days)	730
17	Potential rate of denitrification (kg/(m^2*d))	1.00E-04
18	Saturation activity for moistfact fuction	0.6
19	Specific nitrification rate (1/d)	0.2
20	OrgNstab specific mineralization constant rate (1/d)	7.00E-05
21	Half saturation constant for denitrification (ppm)	10
22	C/N ratio threshold for mobilization/immobilization	10
23	OrgNfrsh decomposition rate (1/d)	0.035
24	Synthesis efficiency constant for cfrsh	0.5
25	C humification fraction	0.2
26	Min soil temp for chem, phys, bio processes (degC)	-2
27	Denitrification threshold water content (=to thetaFC)	0.30
28	deltheta1	0.1
29	deltheta2	0.08
30	N model decay time step (days)	1
31	Fraction of N (by weight) in manure (depends on manure type)	0.017
32	Manure C:N ratio (depends on manure type)	18
33	Stable C fraction of residue (flat, buried, and dead root)	0.4
34	CEC factor	0.15
35	Buffer ofe (number, 0 if none and max OFE if present)	0

Table B3. File name path file format. Only the numerical values highlighted by the text box are required in the corresponding text files.

P	Q	R
Row #	File Name Input (filename*.txt)	Only one column (same for all OFE's)
1	wat file name	C:\Users\Sheila\Documents\MATLAB\wat1.txt
2	crop file name	C:\Users\Sheila\Documents\MATLAB\crop1.txt
3	elem file name	C:\Users\Sheila\Documents\MATLAB\elem1.txt
4	schedule file name	C:\Users\Sheila\Documents\MATLAB\sched1.txt

Table AB4. Schedule file format. Only the numerical values highlighted by the text box are required in the corresponding text files.

V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
Schedule Input (sched*.txt)			Appln Amt (kg/ha)										
Year	Crop ID	Appln Day	OFE1	OFE2	OFE3	OFE4	Planting Day	Harvest Day	Plowing Day	Tillage Depth (m)	Crop Root Depth (m)	Buffer Root Depth (m)	N Fertilizer Code (1=manure, 2=fertilizer)
1	2	132	2.24	2.24	2.24	0	125	284	125	0.05	1.52	0	0
2	3	140	0	0	0	0	132	274	132	0.05	1	0	0
3	2	132	2.24	2.24	2.24	0	125	284	125	0.05	1.52	0	0
4	3	140	0	0	0	0	132	274	132	0.05	1	0	0
5	2	132	2.24	2.24	2.24	0	125	284	125	0.05	1.52	0	0
6	3	140	0	0	0	0	132	274	132	0.05	1	0	0
7	2	132	2.24	2.24	2.24	0	125	284	125	0.05	1.52	0	0
8	3	140	0	0	0	0	132	274	132	0.05	1	0	0
9	2	132	2.24	2.24	2.24	0	125	284	125	0.05	1.52	0	0
10	3	140	0	0	0	0	132	274	132	0.05	1	0	0

MatLab Code

'program.pesticide.m'


```
% This program calculates pesticide transport (dissolved and particulate
% bound) from OFEs for multiple runs and generates a yearly text and bar
% graph for each run. Daily outputs from the daily WEPP hillslope hydrology
% model (i.e. water balance, OFE element, and crop files) must be saved as
% numerical text files prior to running this program.  Parameters and other
% info can be imported from 'info.txt'.

% References: Steenhuis and Walter 1980, Sinkevich et al 2005

%%
% Identify location of WEPP output text files
weppfiles='C:\Users\Sheila\Documents\MATLAB';
% Return a list of user and fixed input files
userinput=fullfile(weppfiles,'userinput*.txt');
fixedinput=fullfile(weppfiles,'fixedinput*.txt');
filenameinputs=fullfile(weppfiles,'filename*.txt');
% Define the user input and fixed input files in the current matlab
% directory
userinputfiles=dir(userinput);
fixedinputfiles=dir(fixedinput);
filenamefiles=dir(filenameinputs);
% Save all outputs to this folder
outputfolder=fullfile('C:\Users\Sheila\Documents\MATLAB\outputs');
% Set current directory (location of all program files)
cd('C:\Users\Sheila\Documents\MATLAB');

%%
for h=1:1:length(userinputfiles)
    % Select the files from the folder for each of k hillslopes
    % User input files
    baseFileNameuser=userinputfiles(h).name;
    fullFileNameuser=fullfile(weppfiles, baseFileNameuser);
    % Fixed input files
    baseFileNamefix=fixedinputfiles(h).name;
    fullFileNamefix=fullfile(weppfiles, baseFileNamefix);
    % File name input files
    baseFileNamefile=filenamefiles(h).name;
    fullFileNamefile=fullfile(weppfiles,baseFileNamefile);

    % User defined inputs
    userinputdata=importdata(fullFileNameuser);
    % Number of OFEs
    numofe=userinputdata(3,1);
    % Fixed inputs
    fixedinputdata=importdata(fullFileNamefix);

    % Run the sub hydrology program to generate bar graphs for each
```

```matlab
% WEPP run.  Note that the sub program that is chosen depends on
% whether the there is a buffer or not.
if userinputdata(37,1)==1 % Buffer
    % Running the sub program pesticide model with buffer
    run('sub_program_pesticide_buff');

    % Generating text files for each run (daily)
    txtstr1=sprintf('day_output');
    txtstr2=sprintf('pestRun%02.0f',h);
    txtfile=[outputfolder,'\',txtstr1,'_',txtstr2,'.txt'];
    dlmwrite(txtfile,dailyPestdata);

    % Generating text files for each run (monthly)
    txtstr1=sprintf('month_output');
    txtstr2=sprintf('pestRun%02.0f',h);
    txtfile=[outputfolder,'\',txtstr1,'_',txtstr2,'.txt'];
    dlmwrite(txtfile,monthlyPestdata);

    % Generating text files for each run (yearly)
    txtstr1=sprintf('yr_output');
    txtstr2=sprintf('pestRun%02.0f',h);
    txtfile=[outputfolder,'\',txtstr1,'_',txtstr2,'.txt'];
    dlmwrite(txtfile,yearlyPestdata);

    % Figures are saved within each run.
else
    % Run the sub program pesticide model
    run('sub_program_pesticide');

    % Generating text files for each run (daily)
    txtstr1=sprintf('day_output');
    txtstr2=sprintf('pestRun%02.0f',h);
    txtfile=[outputfolder,'\',txtstr1,'_',txtstr2,'.txt'];
    dlmwrite(txtfile,dailyPestdata);

    % Generate text files for each run (monthly)
    txtstr1=sprintf('month_output');
    txtstr2=sprintf('pestRun%02.0f',h);
    txtfile=[outputfolder,'\',txtstr1,'_',txtstr2,'.txt'];
    dlmwrite(txtfile,monthlyPestdata);

    % Generate text files for each run (yearly)
    txtstr1=sprintf('yr_output');
    txtstr2=sprintf('pestRun%02.0f',h);
    txtfile=[outputfolder,'\',txtstr1,'_',txtstr2,'.txt'];
    dlmwrite(txtfile,yearlyPestdata);

    % Figures are saved within each run.
end;
end;
```

```
%% Format of Output Text Files

% daily text files (need to divide by area to get cumulative kg/ha)
% colm1: pesticide left in the top layer (kg)
% colm2: pesticide left in the bottom layer (kg)
% colm3: total pesticide lost in overland flow (kg)
% colm4: dissolved pesticide lost in overland flow (kg)
% colm5: adsorbed pesticide lost in overland flow (kg)
% colm6: pesticide lost in lateral flow (kg)
% colm7: pesticide lost in shallow percolation (from top layer into bottom
% layer) (kg)
% colm8: pesticide lost in deep percolation (kg)
% colm10: (buffer runs only) particulate pesticide trapped by buffer
% (kg)


% monthly text files (need to divide by area to get cumulative kg/ha)
% colm1: adsorbed pesticide lost in overland flow (kg)
% colm2: dissolved pesticide lost in overland flow (kg)
% colm3: pesticide lost in lateral flow (kg)
% colm4: pesticide lost in deep percolation (kg)
% colm6: (buffer runs only) particulate pesticide trapped by buffer
% (kg)


% yearly text files (need to divide by area to get cumulative kg/ha)
% colm1: adsorbed pesticide lost in overland flow (kg)
% colm2: dissolved pesticide lost in overland flow (kg)
% colm3: pesticide lost in lateral flow (kg)
% colm4: pesticide lost in deep percolation (kg)
% colm6: (buffer runs only) particulate pesticide trapped by buffer
% (kg)
```

'sub_program_pesticide.m'

```matlab
% sub_program_pesticide.m

% This file is needed to run 'program_pesticide.m'. Basically, it
% initializes matrices and defines parameters based on wat, elem, crop,
% soil, and OFE files loaded in from 'program_pesticide.m'.

% References: Steenhuis and Walter 1980, Sinkevich et al 2005

%% Import wat, crop, elem, sched, filename files.

openfile=fopen(fullFileNamefile);
filenames=textscan(openfile,'%q');
fclose(openfile);
% WEPP water balance files give water depths for hydrologic processes
watdata=importdata(filenames{1}{1});
% WEPP crop files give detailed crop variables
cropdata=importdata(filenames{1}{2});
% WEPP element output files give some crop growth outputs, erosion
% outputs, and runoff
elemdata=importdata(filenames{1}{3});
% WEPP schedule output files give application, planting, harvest dates, and
% maximum rooting depth.
scheddata=importdata(filenames{1}{4});

%% Notes

% Unless marked otherwise these common iterators are used
% i=day number (cumulative over the entire simulation)
% j=month number
% k=year number
% Baseflow is not included here.  The only processes we focus on are
% overland flow (with sediment and without), lateral flow, and percolation.

%% Parameters

% OFE number
ofelst=watdata(:,1);
% OFE lengths (m)
ofeLength=userinputdata(38,1:numofe);
% Sum OFE lengths (m)
ofeLengthsum=sum(ofeLength);

%% Water Balance Output File (wat.txt)

% col1: OFE
% col2: day of the month
% col3: year
% col4: precipitation (snow or rain) (mm)
% col5: rainfall + irrigation + snowmelt (mm)
% col6: daily runoff scaled to single OFE (mm/unit OFE are) = total
% col7: plant transpiration (mm)
```

```matlab
% col8: soil evaporation (mm)
% col9: deep percolation (mm)
% col10: upstream flow (mm)
% col11: lateral flow (mm/unit OFE area)
% col12: total soil water content (mm)
[watrow,watcol]=size(watdata);

%% WEPP Element Output File (elem.txt)

% This file only prints out days when there is a water related event
% (rainfall, snowfall, runoff, etc.).
% colm1: OFE id
% colm2: day of the month (I convert to a cumulative day over the entire
% simulation period - see below)
% colm3: month
% colm4: year
% colm5: precipitation (snow or rain) (mm)
% colm6: runoff (mm)
% colm7: Effective intensity (mm/h)
% colm8: peak runoff (mm/h)
% colm9: Effective duration (h)
% colm10: enrichment ratio
% colm11: Keff (effective hydraulic conductivity of the surface soil - mm/h)
% colm12: Sm (total soil water content - mm)
% colm13: leaf area index (LAI - no units)
% colm14: canopy height (m)
% colm15: canopy cover (%)
% colm16: interrill cover (%)
% colm17: rill cover (%)
% colm18: live biomass (kg/m^2)
% colm19: dead biomass (kg/m^2)
% colm20: Ki (interrill erosion coefficient - )
% colm21: Kr (rill erosion coefficient - )
% colm22: Tcrit? (C)
% colm23: Rill width (m)
% colm24: sediment leaving (kg/m)
[elemrow,elemcol]=size(elemdata);

% Identifying leap years
mlst=[31,28,31,30,31,30,31,31,30,31,30,31];
mlstleap=[31,29,31,30,31,30,31,31,30,31,30,31];
% Month per year
mpy=12;
% Days in year (with leap year)
dayINyearlst=zeros(max(watdata(:,3)),1);
yrs=1:1:max(elemdata(:,4));
for k=1:1:max(elemdata(:,4));
    if mod(yrs(k),4)==0
        dayINyearlst(k)=366;
    else
        dayINyearlst(k)=365;
    end;
end;
% Days in year (with leap year), long list (non-cumulative)
dayINyearlonglst=zeros(watrow,1);
for i=1:1:watrow
```

```matlab
    if mod(watdata(i,3),4)==0
        dayINyearlonglst(i)=366;
    else
        dayINyearlonglst(i)=365;
    end;
end;

% Modify the days so they are cumulative for the year and can be used as a
% id key for combination with other WEPP files.
dlst=zeros(elemrow,1);
for i=1:1:elemrow
    if elemdata(i,3)==1
        dlst(i)=elemdata(i,2);
    elseif elemdata(i,3)>1 && mod(elemdata(i,4),4)==0
        cum=sum(mlstleap(1:(elemdata(i,3))-1));
        dlst(i)=elemdata(i,2)+cum;
    elseif elemdata(i,3)>1 && mod(elemdata(i,4),4)~=0
        cum=sum(mlst(1:(elemdata(i,3))-1));
        dlst(i)=elemdata(i,2)+cum;
    end;
end;

% Add dlst back into the element file
elemdata=[elemdata(:,1),dlst,elemdata(:,3:elemcol)];

% Uses dlst to make the days cumulative for the entire time period
elemid=zeros(elemrow,1);
for i=1:1:elemrow
    if elemdata(i,4)>1
        elemid(i)=elemdata(i,2)+sum(dayINyearlst(1:elemdata(i,4)-1));
    else
        elemid(i)=elemdata(i,2);
    end;
end;

% Add in the daily identifier that is unique for the entire time period
% (e.g. 10 years)
elemdata=[elemdata(:,1),elemid,elemdata(:,3:elemcol)];
[elemrow,elemcol]=size(elemdata);

%% Plant and Residue Output File (crop.txt)

% colm1: OFE
% colm2: day of the year
% colm3: year
% colm4: canopy height (m)
% colm5: canopy cover (%)
% colm6: leaf area index (LAI - no units)
% colm7: rill cover (%)
% colm8: interrill cover (%)
% colm9: crop id #
% colm10: live biomass
% colm11: standing residue mass (kg/m^2)
% colm12: crop id # for the last crop harvested
% colm13: flat residue mass for the last crop harvested (kg/m^2)
% colm14: crop id # for the previous crop harvested
```

```matlab
% colm15: flat residue mass for the previous crop harvested (kg/m^2)
% colm16: crop id # for all previous crops harvested
% colm17: flat residue mass for all previous crops harvested (kg/m^2)
% colm18: buried residue mass for the last crop harvested (kg/m^2)
% colm19: buried residue mass for the previous crop harvest (kg/m^2)
% colm20: buried residue mass for all previous crops harvested (kg/m^2)
% colm21: crop id # for the last crop harvested
% colm22: dead root mass for the last crop harvested (kg/m^2)
% colm23: crop id # for the previous crop harvested
% colm24: dead root mass for the previous crop harvested (kg/m^2)
% colm25: crop id # for all previous crops harvested
% colm26: dead root mass all previous crops harvested (kg/m^2)
% colm27: average temp (C)
% colm28: sediment (kg/m)
cropdata=[watdata(:,1),cropdata];
[croprow,cropcol]=size(cropdata);
% Crop type
croptype=cropdata(:,9);

% Create a daily identifier for the entire run.
srtyr=min(cropdata(:,3));
endyr=max(cropdata(:,3));
cropid=zeros(croprow,1);
for i=1:1:croprow
    if cropdata(i,3)>srtyr
        cropid(i)=cropid(i-numofe)+1;
    else
        cropid(i)=cropdata(i,2);    % Returns first year as is
    end;
end;

% Add in the daily identifier
cropdata=[cropdata(:,1),cropid,cropdata(:,3:cropcol)];

% Add in sediment and soil moisture
a=[elemdata(:,2),elemdata(:,24)];        % Sediment (kg/m)
sednewlst=zeros(croprow,1);              % New matrix (sediment in col 1,
sm in col 2)
for i=1:1:elemrow
    % Sediment
    if a(i,2)>0
        idlook=a(i,1:2);                 % Identify sediment event day
        x=find(cropdata(:,2)==idlook(1),1); % Identify associated row id for
the day (one row for each ofe so have to adjust for this by subtracting 1)
        sednewlst(x+elemdata(i,1)-1)=idlook(2);
    end;
end;
cropdata=[cropdata,sednewlst];
[croprow,cropcol]=size(cropdata);

%% Final Compiled Water Balance File (wat.txt)

% Adding all the crop file data to water balance file
% col1: OFE
% col2: day identifier
% col3: day of the month
```

55

```
% col4: year
% col5: precipitation (snow or rain) (mm)
% col6: rainfall + irrig + snowmelt (mm)
% col7: daily runoff scaled to single OFE (mm/unit OFE are) = total
% col8: plant transpiration (mm)
% col9: soil evaporation (mm)
% col10: deep percolation (mm)
% col11: upstream runoff (mm)
% col12: lateral flow (mm)
% col13: total soil water content (mm)
% col14: sediment (kg/m)
% col15: leaf area index (LAI - no units)
% col16: average temperature (C)
% col17: row id (#)


id=1:1:watrow;
watdata=[watdata(:,1),cropdata(:,2),watdata(:,2:watcol),cropdata(:,28),cropda
ta(:,6),cropdata(:,27),transpose(id)];
[watrow,watcol]=size(watdata);


%% Scheduling File (sched.txt)

% colm1: year
% colm2: crop id # (corresponds to crop type)
% colm3: application date (day of year)
% colm4: application amount for OFE1
% colm5: application amount for OFE2
% colm6: application amount for OFE3
% colm7: application amount for OFE 4 (buffer)
% colm8: plant date (day of year) (only for non-buffer OFEs)
% colm9: harvest date (day of year) (only for non-buffer OFEs)
% colm10: plow date (day of year)
% colm11: tillage depth (m)
% colm12: crop root depth (m)
% colm13: buffer root depth (m)
% colm14: N Fertilizer Code (1=manure, 2=fertilizer)


%% Defining Inputs from WEPP

% Number of simulation days
numsimdays=sum(dayINyearlst);
% Number of years
numyrs=max(elemdata(:,4));
% Day identifier
daylst=watdata(:,2);
% Day of the month
dayofmnthlst=watdata(:,3);
% Year
yrlst=watdata(:,4);
% Precipitation (snow or rain) (m) (non-cumulative)
precip=watdata(:,5)./1000;
% Rainfall+snowmelt+irrigation (m) (non-cumulative)
ris=watdata(:,6)./1000;
% Calculate the area of the ofe
ofeWidth=userinputdata(39,:);
ofeArea=ofeLength.*ofeWidth;
```

```matlab
% Cumulative overland flow passing through the OFE (m)
ovldf=watdata(:,7)./1000;
% Cumulative lateral flow passing through the OFE (m)
latf=watdata(:,12)./1000;
% Percolation (m) (non-cumulative)
perc=watdata(:,10)./1000;
% Cumulative sediment leaving an OFE per width (kg/m)
sedkgm=watdata(:,14);
% Cumulative ediment leaving an OFE (kg)
sed=zeros(watrow,1);
for i=1:1:watrow
    sed(i)=sedkgm(i)*ofeWidth(watdata(i,1));
end;
% Net sediment loss (kg) (non-cumulative)
netsed=zeros(watrow,1);
for i=2:1:watrow
    if watdata(i,1)>1
        mass=sedkgm(i)*ofeWidth(watdata(i,1));
        massprev=sedkgm(i-1)*ofeWidth(watdata(i-1,1));
        netsed(i)=mass-massprev;
    else
        mass=sedkgm(i)*ofeWidth(watdata(i,1));
        netsed(i)=mass;
    end;
end;
% Crop transpiration (m)
cropevap=watdata(:,8)./1000;
% Soil evaporation (m)
soilevap=watdata(:,9)./1000;
% Evapotranspiration (m)
et=cropevap+soilevap;
% Soil moisture (m)
soilmoist=watdata(:,13)./1000;
% Average temp (C)
avgT=watdata(:,16);


%% Parameters

% Initialize parameters for each crop
ps=zeros(max(croptype),numofe);
om=zeros(max(croptype),numofe);
thetaS=zeros(max(croptype),numofe);
thetaFC=zeros(max(croptype),numofe);
koc=zeros(max(croptype),numofe);
thalf=zeros(max(croptype),numofe);
for o=1:1:numofe
    for c=1:1:max(croptype);
        % Soil bulk density (g/cm^3)
        ps(c,o)=userinputdata(5+c-1,o);
        % Convert ps to kg/m^3
        ps(c,o)=ps(c,o)*1000;
        % Water content at saturation, import from WEPP
        thetaS(c,o)=userinputdata(9+c-1,o);
        % Field capacity (thetaFC)
        thetaFC(c,o)=userinputdata(13+c-1,o);
        % Percent organic matter in the mixing layer
```

```matlab
        om(c,o)=userinputdata(21+c-1,numofe);
        % Pesticide (here, atrazine) organic carbon adsorption coefficient
        % (cm^3/g), see Excel spreadsheet for values
        koc(c,o)=userinputdata(26+c-1,o);
        % Convert to m^3/kg
        koc(c,o)=koc(c,o)/1000;
        % Half-life for atrazine (or other pesticide) (days)
        thalf(c,o)=userinputdata(30+c-1,o);
    end;
end;
% Max root depth (m), for each crop
% Second colm is buffer root depth
rootdepthshrt=scheddata(:,12:13);
rootdepth=zeros(croprow,1);
for i=1:1:croprow
    rflag=find(scheddata(:,1)==cropdata(i,3));
    r=rootdepthshrt(rflag,1);
    rsel=unique(r);
    rootdepth(i)=rsel;
end;
% Incorporation (yes=1, no=0);
incorp=userinputdata(25,:);
% Percent organic carbon
oc=(om./100).*0.58;
% Adsorption partition coefficient (see Sinkevich 2005) (m^3/kg)
apc=koc.*oc;
% Density of water (g/cm^3) at 20C
pw=0.9980;
% Convert pw to kg/m^3
pw=pw*1000;
% Time step (for decay, in days)
tstep=1;


%% Water Content

% Also considers soil water content losses due to crop and soil evaporation
% ( residue evaporation is left out)
theta=(soilmoist-et)./rootdepth;


%% Pesticide Application Information (date and amount), Planting Date, and
Harvest Date

% Application location and amount in kg/m^2
applnofe=scheddata(:,4:7)./10000;
% Application and plow dates
applndate=scheddata(:,3);
plowdate=scheddata(:,10);
% Convert date to a cumulative date for the simulation
[sr,sc]=size(scheddata);
applndatecum=zeros(sr,1);
plowdatecum=zeros(sr,1);
for s=1:1:sr
    % Application of pesticide
    if applndate(s)>0
        if scheddata(s,1)==1 % first year
            applndatecum(s)=applndate(s);
```

```matlab
        else % second year and on
            applndatecum(s)=applndate(s)+sum(dayINyearlst(1:scheddata(s,1)-
1));
        end;
    else % just in case
        applndatecum(s)=0;
    end;
    % Plowing (included planting b/c of disruption to soil)
    if plowdate(s)>0
        if scheddata(s,1)==1 % first year
            plowdatecum(s)=plowdate(s);
        else % second year and on
            plowdatecum(s)=plowdate(s)+sum(dayINyearlst(1:scheddata(s,1)-1));
        end;
    else % just in case
        plowdatecum(s)=0;
    end;
end;
plowdepth=scheddata(:,11);

%% Initialize Matrices

% Pesticide in the mixing layer (kg/m^2)
pestleftTop=zeros(numsimdays,numofe);
% Pesticide in the deep layer (kg/m^2)
pestleftBot=zeros(numsimdays,numofe);
% Total pesticide lost in overland flow with sediment (dissolved and sorbed)
pestlostovlfTot=zeros(numsimdays,numofe);
% Dissolved pesticide lost in overland flow pesticide (kg/m^2)
pestlostovlfW=zeros(numsimdays,numofe);
% Sorbed pesticide lost in overland flow pesticide (kg/m^2)
pestlostovlfS=zeros(numsimdays,numofe);
% Pesticide lost due to lateral flow (kg/m^2)
pestlostlatf=zeros(numsimdays,numofe);
% Pesticide lost due to shallow percolation (kg/m^2) (to below mixing layer)
pestlostshalperc=zeros(numsimdays,numofe);
% Pesticide lost due to percolation (kg/m^2)
pestlostperc=zeros(numsimdays,numofe);
% Sediment fraction (kg/kg)
sedfraclst=zeros(numsimdays,numofe);
% Mixing depth
hmix=zeros(numsimdays,numofe);
% W amount of space available to hold dis atrazine in mixing layer
vol=zeros(numsimdays,numofe);
% Evapotranspiration
etlst=zeros(numsimdays,numofe);

%% Daily Simulation

k=1;
for id=numofe+1:1:watrow %start at second day
    i=daylst(id);
    ofe=ofelst(id);
    % Upslope Pesticide Contributions (dissolved P in runnoff and lateral
flow)
    if ofe>1
```

```matlab
        lostPestovldfWUP=pestlostovlfW(i,ofe-1);
        lostPestovldfSUP=pestlostovlfS(i,ofe-1);
        lostPestlaftUP=pestlostlatf(i,ofe-1);
    else
        lostPestovldfWUP=0;
        lostPestovldfSUP=0;
        lostPestlaftUP=0;
    end;
    % Calculate fraction of runoff make up of sediment (kg/kg)
    if ovldf(id)>0
        ovldfvol=ovldf(id)*ofeArea(ofe); % runoff volume
        ovldfmass=ovldfvol*pw;
        sedfrac=sed(id)/ovldfmass;
    else
        sedfrac=0;
    end;
    % Calculate pesticide applied to ofe (kg)
    bermwidth=userinputdata(40,ofe); % m
    width=userinputdata(39,ofe); % m
    applnarea=(width-2*bermwidth)*(ofeLength(ofe)-2*bermwidth);
    % Identify application day for the year
    d=find(scheddata(:,1)==k); % identify position
    allappdays=applndatecum(d(:));
    if intersect(allappdays,i)>0
        % Application day (cumulative)
        applnday=intersect(allappdays,i);
        % Application row
        applnrow=find(applndatecum(:)==applnday);
        % Application amount (convert to kg)
        applnamt=unique(applnofe(applnrow,ofe))*applnarea;
    else
        % Application day (cumulative)
        applnday=0;
        % Application amount (kg)
        applnamt=0;
    end;
    % Define depth of mixing layer
    allplowdays=plowdatecum(d(:));
    if intersect(allplowdays,i)>0
        % Plow day
        plowday=intersect(allplowdays,i);
        % Plow row
        plowrow=find(plowdatecum(:)==plowday);
        % Mixing layer depth (m)
        mixdepth=unique(plowdepth(plowrow));
    else
        % Non plow day
        plowday=0;
        mixdepth=userinputdata(41,ofe);
    end;

    % Non-uniform application with transport between OFEs

out=dailyofePestsimTrans(id,i,k,tstep,apc(croptype(id),ofe),applnamt,mixdepth
,hmix(i-
1,ofe),ps(croptype(id),ofe),pw,thetaS(croptype(id),ofe),rootdepth(id),applnda
y,plowday,thalf(croptype(id),ofe),pestleftTop(i-1,ofe),pestleftBot(i-
```

```matlab
1,ofe),lostPestovldfWUP,lostPestovldfSUP,lostPestlaftUP,ris(id),et(id),ovldf(
id),latf(id),perc(id),sedfrac,ofe);
    pestlostovlfTot(out(3),out(2))=out(5);  % Dissolved and particulate
pesticide lost in overland flow (cumulative)
    pestlostovlfW(out(3),out(2))=out(6);    % Dissolved pesticide lost in
overland flow (cumulative)
    pestlostovlfS(out(3),out(2))=out(7);    % Particulate pesticide lost in
overland flow (cumulative)
    pestlostlatf(out(3),out(2))=out(8);     % Dissolved pesticide lost in
lateral flow (cumulative)
    pestlostshalperc(out(3),out(2))=out(9); % Dissolved pesticide lost in
shallow percolation (to transition layer, not cumulative)
    pestlostperc(out(3),out(2))=out(10);    % Dissolved pesticide lost in
deep percolation (to groundwater, not cumulative)
    pestleftTop(out(3),out(2))=out(11);     % Pesticide left in the mixing
layer
    pestleftBot(out(3),out(2))=out(12)+out(9);  % Pesticide left in the
transition layer
    sedfraclst(i,ofe)=sedfrac;                      % Sediment fraction
    hmix(i,ofe)=mixdepth;                   % mixing depth

vol(i,ofe)=mixdepth*(thetaS(croptype(id),ofe)+apc(croptype(id),ofe)*ps(cropty
pe(id),ofe));
    etlst(i,ofe)=soilevap(id)+cropevap(id);

    % Iterator
    if daylst(id)<=sum(dayINyearlst(1:k))
        k=k+0;
    else
        k=k+1;
    end;
end;


%% Daily

% Save daily data (kg) (need to divide by area to get cumulative kg/ha)
dailyPestdata=[pestleftTop,pestleftBot,pestlostovlfTot,pestlostovlfW,pestlost
ovlfS,pestlostlatf,pestlostshalperc,pestlostperc];

%% Montly

% Month list for the entire simulation
mbiglst=zeros(mpy*numyrs,1);
srt=1;
for k=1:1:numyrs
    if dayINyearlst(k)==365
        mbiglst(srt:srt+mpy-1)=mlst;
    else
        mbiglst(srt:srt+mpy-1)=mlstleap;
    end;
    srt=srt+mpy;
end;
% Cumulative monthly list for the entire simulation
mbiglstcum=zeros(mpy*numyrs,1);
for j=1:1:mpy*numyrs
    if j==1
```

```matlab
            mbiglstcum(j)=mbiglst(1);
        else
            mbiglstcum(j)=sum(mbiglst(1:j));
        end;
end;
% Use mbiglstcum to add up all daily processes
monthlylostPesteros=zeros(mpy*numyrs,numofe);
monthlylostPestovldfnosed=zeros(mpy*numyrs,numofe);
monthlylostPestlatf=zeros(mpy*numyrs,numofe);
monthlylostPestperc=zeros(mpy*numyrs,numofe);
srt=1;
for j=1:1:mpy*numyrs
    for i=1:1:numofe
        monthlylostPesteros(j,i)=sum(pestlostovlfS(srt:mbiglstcum(j),i));

monthlylostPestovldfnosed(j,i)=sum(pestlostovlfW(srt:mbiglstcum(j),i));
        monthlylostPestlatf(j,i)=sum(pestlostlatf(srt:mbiglstcum(j),i));
        monthlylostPestperc(j,i)=sum(pestlostperc(srt:mbiglstcum(j),i));
    end;
    srt=mbiglstcum(j)+1;
end;


% Save montly data (kg) (need to divide by area to get cumulative kg/ha)
monthlyPestdata=[monthlylostPesteros,monthlylostPestovldfnosed,monthlylostPes
tlatf,monthlylostPestperc];


%% Yearly

yearlylostPesteros=zeros(numyrs,numofe);
yearlylostPestovldfnosed=zeros(numyrs,numofe);
yearlylostPestlatf=zeros(numyrs,numofe);
yearlylostPestperc=zeros(numyrs,numofe);
srt=1;
for k=1:1:numyrs
    for i=1:1:numofe

yearlylostPesteros(k,i)=sum(pestlostovlfS(srt:sum(dayINyearlst(1:k)),i));

yearlylostPestovldfnosed(k,i)=sum(pestlostovlfW(srt:sum(dayINyearlst(1:k)),i)
);

yearlylostPestlatf(k,i)=sum(pestlostlatf(srt:sum(dayINyearlst(1:k)),i));

yearlylostPestperc(k,i)=sum(pestlostperc(srt:sum(dayINyearlst(1:k)),i));
    end;
    srt=srt+dayINyearlst(k);
end;


% Save yearly data (kg) (need to divide by area to get cumulative kg/ha)
yearlyPestdata=[yearlylostPesteros,yearlylostPestovldfnosed,yearlylostPestlat
f,yearlylostPestperc];


%% Yearly (for total hillslope, i.e. the last OFE) (kg)

yearlyPesteroshill=zeros(numyrs,1);
yearlyPestovldfnosedhill=zeros(numyrs,1);
```

```matlab
yearlyPestlatfhill=zeros(numyrs,1);
yearlyPestperchill=zeros(numyrs,1);

for k=1:1:numyrs
    yearlyPesteroshill(k)=yearlylostPesteros(k,numofe)-
yearlylostPesteros(k,numofe-1);
    yearlyPestovldfnosedhill(k)=yearlylostPestovldfnosed(k,numofe)-
yearlylostPestovldfnosed(k,numofe-1);
    yearlyPestlatfhill(k)=yearlylostPestlatf(k,numofe)-
yearlylostPestlatf(k,numofe-1);
    yearlyPestperchill(k)=yearlylostPestperc(k,numofe);
end;

% Yearly data at bottom of the hillslope (kg) (need to divide by area to
% get cumulative kg/ha)
yearlyPesthilldata=[yearlyPesteroshill,yearlyPestovldfnosedhill,yearlyPestlat
fhill,yearlyPestperchill];

%% Yearly Fraction (for total hillslope, i.e. the last OFE) (%)

% Make a matrix with total application rates for each OFE for each year
yearlyPestapplnlst=zeros(numyrs,numofe);
for k=1:1:numyrs
    x=find(scheddata(:,1)==k);
    xamt=scheddata(x,4:4+numofe);
    for ofe=1:1:numofe
        yearlyPestapplnlst(k,ofe)=sum(xamt(:,ofe))/10000*ofeArea(ofe);
    end;
end;

yearlyPesteroshillfrac=zeros(numyrs,1);
yearlyPestovldfnosedhillfrac=zeros(numyrs,1);
yearlyPestlatfhillfrac=zeros(numyrs,1);
yearlyPestperchillfrac=zeros(numyrs,1);

for k=1:1:numyrs
    totyrapplnamt=sum(yearlyPestapplnlst(k,:));
    if totyrapplnamt>0
    yearlyPesteroshillfrac(k,1)=yearlyPesteroshill(k)/totyrapplnamt*100;

yearlyPestovldfnosedhillfrac(k,1)=yearlyPestovldfnosedhill(k)/totyrapplnamt*1
00;
    yearlyPestlatfhillfrac(k,1)=yearlyPestlatfhill(k)/totyrapplnamt*100;
    yearlyPestperchillfrac(k,1)=yearlyPestperchill(k)/totyrapplnamt*100;
    else
    yearlyPesteroshillfrac(k,1)=0;
    yearlyPestovldfnosedhillfrac(k,1)=0;
    yearlyPestlatfhillfrac(k,1)=0;
    yearlyPestperchillfrac(k,1)=0;
    end;
end;

% Yearly data at bottom of the hillslope (kg) (need to divide by area to
% get cumulative kg/ha)
```

```matlab
yearlyPesthilldatafrac=[yearlyPesteroshillfrac,yearlyPestovldfnosedhillfrac,y
earlyPestlatfhillfrac,yearlyPestperchillfrac];
```

```matlab
%% Yearly Net Averages by OFE (kg)

yearlylostPesterosavg=zeros(1,numofe);
yearlylostPestovldfnosedavg=zeros(1,numofe);
yearlylostPestlatfavg=zeros(1,numofe);
yearlylostPestpercavg=zeros(1,numofe);

for ofe=1:1:numofe
    if ofe>1
        yearlylostPesterosavg(1,ofe)=(sum(yearlylostPesteros(:,ofe))-
sum(yearlylostPesteros(:,ofe-1)))/numyrs;

yearlylostPestovldfnosedavg(1,ofe)=(sum(yearlylostPestovldfnosed(:,ofe))-
sum(yearlylostPestovldfnosed(:,ofe-1)))/numyrs;
        yearlylostPestlatfavg(1,ofe)=(sum(yearlylostPestlatf(:,ofe))-
sum(yearlylostPestlatf(:,ofe-1)))/numyrs;
        yearlylostPestpercavg(1,ofe)=sum(yearlylostPestperc(:,ofe))/numyrs;
    else
        yearlylostPesterosavg(1,ofe)=sum(yearlylostPesteros(:,ofe))/numyrs;

yearlylostPestovldfnosedavg(1,ofe)=sum(yearlylostPestovldfnosed(:,ofe))/numyr
s;
        yearlylostPestlatfavg(1,ofe)=sum(yearlylostPestlatf(:,ofe))/numyrs;
        yearlylostPestpercavg(1,ofe)=sum(yearlylostPestperc(:,ofe))/numyrs;
    end;
end;

% Yearly average (kg) (need to divide by area to get cummulative kg/ha)
yearlyPestavgdata=[yearlylostPesterosavg;yearlylostPestovldfnosedavg;yearlylo
stPestlatfavg;yearlylostPestpercavg];

%% Yearly Net Averages by OFE (%)

yearlylostPesterosavgfrac=zeros(1,numofe);
yearlylostPestovldfnosedavgfrac=zeros(1,numofe);
yearlylostPestlatfavgfrac=zeros(1,numofe);
yearlylostPestpercavgfrac=zeros(1,numofe);

for ofe=1:1:numofe
    appfrac=sum(sum(yearlyPestapplnlst(:,ofe)));
    if appfrac>0

yearlylostPesterosavgfrac(1,ofe)=yearlylostPesterosavg(1,ofe)/appfrac*100;

yearlylostPestovldfnosedavgfrac(1,ofe)=yearlylostPestovldfnosedavg(1,ofe)/app
frac*100;

yearlylostPestlatfavgfrac(1,ofe)=yearlylostPestlatfavg(1,ofe)/appfrac*100;

yearlylostPestpercavgfrac(1,ofe)=yearlylostPestpercavg(1,ofe)/appfrac*100;
    else
        yearlylostPesterosavgfrac(1,ofe)=0;
```

```matlab
        yearlylostPestovldfnosedavgfrac(1,ofe)=0;
        yearlylostPestlatfavgfrac(1,ofe)=0;
        yearlylostPestpercavgfrac(1,ofe)=0;
    end;
end;


% Yearly average (kg) (need to divide by area to get cummulative kg/ha)
yearlyPestavgdatafrac=[yearlylostPesterosavgfrac;yearlylostPestovldfnosedavgf
rac;yearlylostPestlatfavgfrac;yearlylostPestpercavgfrac];


%% Calculating kg/ha Lost for Each Hillslope Section

pestovldWkgha=zeros(numsimdays,numofe);
pestovldSkgha=zeros(numsimdays,numofe);
pestlatfkgha=zeros(numsimdays,numofe);
pestperckgha=zeros(numsimdays,numofe);


for i=1:1:numsimdays
    for ofe=1:1:numofe
        % dissolved pesticide lost in overland flow
        if ofe==1
            pestovldWkgha(i,ofe)=pestlostovlfW(i,ofe)./ofeArea(ofe).*1e4;
        else
            if (pestlostovlfW(i,ofe)-pestlostovlfW(i,ofe-1))<0
                pestovldWkgha(i,ofe)=0;
            else
                pestovldWkgha(i,ofe)=(pestlostovlfW(i,ofe)-
pestlostovlfW(i,ofe-1))./ofeArea(ofe).*1e4;
            end;
        end;


        % particulate pesticide lost in overland flow
        if ofe==1
            pestovldSkgha(i,ofe)=pestlostovlfS(i,ofe)./ofeArea(ofe).*1e4;
        else
            if (pestlostovlfS(i,ofe)-pestlostovlfS(i,ofe-1))<0
                pestovldSkgha(i,ofe)=0;
            else
                pestovldSkgha(i,ofe)=(pestlostovlfS(i,ofe)-
pestlostovlfS(i,ofe-1))./ofeArea(ofe).*1e4;
            end;
        end;


        % dissolved pesticide lost in lateral flow
        if ofe==1
            pestlatfkgha(i,ofe)=pestlostlatf(i,ofe)./ofeArea(ofe).*1e4;
        else
            if (pestlostlatf(i,ofe)-pestlostlatf(i,ofe-1))<0
                pestlatfkgha(i,ofe)=0;
            else
                pestlatfkgha(i,ofe)=(pestlostlatf(i,ofe)-pestlostlatf(i,ofe-
1))./ofeArea(ofe).*1e4;
            end;
        end;
```

```matlab
        % dissolved pesticide lost percolation (not cummulative)
        pestperckgha(i,ofe)=pestlostperc(i,ofe)./ofeArea(ofe).*1e4;

    end;
end;


% Compute the average for the simulation
pestkgha=zeros(numofe,4);
for ofe=1:1:numofe
    pestkgha(ofe,1)=sum(pestovldWkgha(:,ofe))/numyrs;
    pestkgha(ofe,2)=sum(pestovldSkgha(:,ofe))/numyrs;
    pestkgha(ofe,3)=sum(pestlatfkgha(:,ofe))/numyrs;
    pestkgha(ofe,4)=sum(pestperckgha(:,ofe))/numyrs;

end;

% Convert to a percentage based on the maximum value
pestkghapercent=zeros(numofe,4);
for i=1:1:4
    totalval=sum(pestkgha(:,i));
    pestkghapercent(:,i)=pestkgha(:,i)./totalval*100;
end;

%% Plot Pesticide Losses per OFE

% Saving plots
% Generating bar graphs for each run
plotstr1=sprintf('fig_output');
plotstr2=sprintf('Run%02.0f',h);

% Convert run number to a string
runnum=num2str(h);

bar(pestkgha)
title(['Run  ',runnum],'FontSize',14,'FontName','Times');
ylabel('Pesticide Lost (kg/ha)','FontSize',14,'FontName','Times');
xlabel('OFE','FontSize',14,'FontName','Times');
legend('Dissolved (Ovldf)','Particulate
(Ovldf)','Latf','Perc','Location','Best');
set(gca,'XTickLabel',{'Top','Mid','Bot'},'FontSize',14,'FontName','Times')
colormap gray
box off

figure1=gcf;
plotfile1=[outputfolder,'\',plotstr1,'_pest_',plotstr2];
saveas(figure1,plotfile1,'png');
clf(figure1,'reset');

%% Plot Percent Pesticide Losses per OFE

bar(transpose(pestkghapercent))
title(['Run  ',runnum],'FontSize',14,'FontName','Times');
ylabel('Pesticide Lost (%)','FontSize',14,'FontName','Times');
xlabel('Hydrologic Process','FontSize',14,'FontName','Times');
legend('Top','Mid','Bot','Location','Best');
```

```matlab
set(gca,'XTickLabel',{'Ovldf (Diss.)','Ovldf
(Part.)','Latf','Perc'},'FontSize',14,'FontName','Times')
colormap gray
box off

figure2=gcf;
plotfile2=[outputfolder,'\',plotstr1,'_pest_percent_',plotstr2];
saveas(figure2,plotfile2,'png');
clf(figure2,'reset');
```

'sub_program_pesticide_buff.m'

```
% sub_program_pesticide_buff.m

% This file is needed to run 'program_pesticide.m'. Basically, it
% initializes matrices and defines parameters based on wat, elem, crop,
% soil, and OFE files loaded in from 'program_pesticide.m'.

% References: Steenhuis and Walter 1980, Sinkevich et al 2005

%% Import wat, crop, elem, sched, filename files.

openfile=fopen(fullFileNamefile);
filenames=textscan(openfile,'%q');
fclose(openfile);
% WEPP water balance files give water depths for hydrologic processes
watdata=importdata(filenames{1}{1});
% WEPP crop files give detailed crop variables
cropdata=importdata(filenames{1}{2});
% WEPP element output files give some crop growth outputs, erosion
% outputs, and runoff
elemdata=importdata(filenames{1}{3});
% WEPP schedule output files give application, planting, harvest dates, and
% maximum rooting depth.
scheddata=importdata(filenames{1}{4});

%% Parameters

% OFE number
ofelst=watdata(:,1);
% OFE lengths (m)
ofeLength=userinputdata(38,1:numofe);
% Sum OFE lengths (m)
ofeLengthsum=sum(ofeLength);
% Buffer OFE (# assume last)
buffernum=fixedinputdata(35);

%% Water Balance Output File (wat.txt)

% col1: OFE
% col2: day of the month
% col3: year
% col4: precipitation (snow or rain) (mm)
% col5: rainfall + irrigation + snowmelt (mm)
% col6: daily runoff scaled to single OFE (mm/unit OFE are) = total
% col7: plant transpiration (mm)
% col8: soil evaporation (mm)
% col9: deep percolation (mm)
% col10: upstream flow (mm)
% col11: lateral flow (mm/unit OFE area)
% col12: total soil water content (mm)
[watrow,watcol]=size(watdata);
```

```matlab
%% WEPP Element Output File

% This file only prints out days when there is a water related event
% (rainfall, snowfall, runoff, etc.).
% col1: OFE id
% col2: day of the month (I convert to a cumulative day over the entire
% simulation period - see below)
% col3: month
% col4: year
% col5: precipitation (snow or rain) (mm)
% col6: runoff (mm)
% col7: Effective intensity (mm/h)
% col8: peak runoff (mm/h)
% col9: Effective duration (h)
% col10: enrichment ratio
% col11: Keff (effective hydraulic conductivity of the surface soil - mm/h)
% col12: Sm (total soil water content - mm)
% col13: leaf area index (LAI - no units)
% col14: canopy height (m)
% col15: canopy cover (%)
% col16: interrill cover (%)
% col17: rill cover (%)
% col18: live biomass (kg/m^2)
% col19: dead biomass (kg/m^2)
% col20: Ki (interrill erosion coefficient - )
% col21: Kr (rill erosion coefficient - )
% col22: Tcrit? (C)
% col23: Rill width (m)
% col24: sediment leaving (kg/m)
[elemrow,elemcol]=size(elemdata);

% Identifying leap years
mlst=[31,28,31,30,31,30,31,31,30,31,30,31];
mlstleap=[31,29,31,30,31,30,31,31,30,31,30,31];
% Month per year
mpy=12;
% Days in year (with leap year)
dayINyearlst=zeros(max(watdata(:,3)),1);
yrs=1:1:max(elemdata(:,4));
for k=1:1:max(elemdata(:,4));
    if mod(yrs(k),4)==0
        dayINyearlst(k)=366;
    else
        dayINyearlst(k)=365;
    end;
end;
% Days in year (with leap year), long list (non-cumulative)
dayINyearlonglst=zeros(watrow,1);
for i=1:1:watrow
    if mod(watdata(i,3),4)==0
        dayINyearlonglst(i)=366;
    else
        dayINyearlonglst(i)=365;
    end;
end;
```

```matlab
% Modify the days so they are cumulative for the year and can be used as a
% id key for combination with other WEPP files.
dlst=zeros(elemrow,1);
for i=1:1:elemrow
    if elemdata(i,3)==1
        dlst(i)=elemdata(i,2);
    elseif elemdata(i,3)>1 && mod(elemdata(i,4),4)==0
        cum=sum(mlstleap(1:(elemdata(i,3))-1));
        dlst(i)=elemdata(i,2)+cum;
    elseif elemdata(i,3)>1 && mod(elemdata(i,4),4)~=0
        cum=sum(mlst(1:(elemdata(i,3))-1));
        dlst(i)=elemdata(i,2)+cum;
    end;
end;


% Add dlst back into the element file
elemdata=[elemdata(:,1),dlst,elemdata(:,3:elemcol)];


% Uses dlst to make the days cumulative for the entire time period
elemid=zeros(elemrow,1);
for i=1:1:elemrow
    if elemdata(i,4)>1
        elemid(i)=elemdata(i,2)+sum(dayINyearlst(1:elemdata(i,4)-1));
    else
        elemid(i)=elemdata(i,2);
    end;
end;


% Add in the daily identifier that is unique for the entire time period
% (e.g. 10 years)
elemdata=[elemdata(:,1),elemid,elemdata(:,3:elemcol)];
[elemrow,elemcol]=size(elemdata);


%% Plant and Residue Output File

% col1: OFE
% col2: day of the year
% col3: year
% col4: canopy height (m)
% col5: canopy cover (%)
% col6: leaf area index (LAI - no units)
% col7: rill cover (%)
% col8: interrill cover (%)
% col9: crop id #
% col10: live biomass
% col11: standing residue mass (kg/m^2)
% col12: crop id # for the last crop harvested
% col13: flat residue mass for the last crop harvested (kg/m^2)
% col14: crop id # for the previous crop harvested
% col15: flat residue mass for the previous crop harvested (kg/m^2)
% col16: crop id # for all previous crops harvested
% col17: flat residue mass for all previous crops harvested (kg/m^2)
% col18: buried residue mass for the last crop harvested (kg/m^2)
% col19: buried residue mass for the previous crop harvest (kg/m^2)
% col20: buried residue mass for all previous crops harvested (kg/m^2)
% col21: crop id # for the last crop harvested
```

```matlab
% col22: dead root mass for the last crop harvested (kg/m^2)
% col23: crop id # for the previous crop harvested
% col24: dead root mass for the previous crop harvested (kg/m^2)
% col25: crop id # for all previous crops harvested
% col26: dead root mass all previous crops harvested (kg/m^2)
% col27: average temp (C)
% col28: sediment (kg/m)
cropdata=[watdata(:,1),cropdata];
[croprow,cropcol]=size(cropdata);
% Crop type
croptype=cropdata(:,9);

% Create a daily identifier for the entire run.
srtyr=min(cropdata(:,3));
endyr=max(cropdata(:,3));
cropid=zeros(croprow,1);
for i=1:1:croprow
    if cropdata(i,3)>srtyr
        cropid(i)=cropid(i-numofe)+1;
    else
        cropid(i)=cropdata(i,2);    % Returns first year as is
    end;
end;

% Add in the daily identifier
cropdata=[cropdata(:,1),cropid,cropdata(:,3:cropcol)];

% Add in sediment and soil moisture
a=[elemdata(:,2),elemdata(:,24)];          % Sediment (kg/m)
sednewlst=zeros(croprow,1);                % New matrix (sediment in col 1,
sm in col 2)
for i=1:1:elemrow
    % Sediment
    if a(i,2)>0
        idlook=a(i,1:2);                   % Identify sediment event day
        x=find(cropdata(:,2)==idlook(1),1); % Identify associated row id for
the day (one row for each ofe so have to adjust for this by subtracting 1)
        sednewlst(x+elemdata(i,1)-1)=idlook(2);
    end;
end;
cropdata=[cropdata,sednewlst];
[croprow,cropcol]=size(cropdata);

%% Final Compiled Water Balance File

% Adding all the crop file data to water balance file
% col1: OFE
% col2: day identifier
% col3: day of the month
% col4: year
% col5: precipitation (snow or rain) (mm)
% col6: rainfall + irrig + snowmelt (mm)
% col7: daily runoff scaled to single OFE (mm/unit OFE are) = total
% col8: plant transpiration (mm)
% col9: soil evaporation (mm)
% col10: deep percolation (mm)
```

```matlab
% col11: upstream runoff (mm)
% col12: lateral flow (mm)
% col13: total soil water content (mm)
% col14: sediment (kg/m)
% col15: leaf area index (LAI - no units)
% col16: average temperature (C)
% col17: row id (#)

id=1:1:watrow;
watdata=[watdata(:,1),cropdata(:,2),watdata(:,2:watcol),cropdata(:,28),cropda
ta(:,6),cropdata(:,27),transpose(id)];
[watrow,watcol]=size(watdata);


%% Scheduling File (sched.txt)

% colm1: year
% colm2: crop id # (corresponds to crop type)
% colm3: application date (day of year)
% colm4: application amount for OFE1
% colm5: application amount for OFE2
% colm6: application amount for OFE3
% colm7: application amount for OFE 4 (buffer)
% colm8: plant date (day of year) (only for non-buffer OFEs)
% colm9: harvest date (day of year) (only for non-buffer OFEs)
% colm10: plow date (day of year)
% colm11: tillage depth (m)
% colm12: crop root depth (m)
% colm13: buffer root depth (m)
% colm14: N Fertilizer Code (1=manure, 2=fertilizer)


%% Defining Inputs from WEPP

% Number of simulation days
numsimdays=sum(dayINyearlst);
% Number of years
numyrs=max(elemdata(:,4));
% Day identifier
daylst=watdata(:,2);
% Day of the month
dayofmnthlst=watdata(:,3);
% Year
yrlst=watdata(:,4);
% Precipitation (snow or rain) (m) (non-cumulative)
precip=watdata(:,5)./1000;
% Rainfall+snowmelt+irrigation (m) (non-cumulative)
ris=watdata(:,6)./1000;
% Calculate the area of the OFE
ofeWidth=userinputdata(39,:);
ofeArea=ofeLength.*ofeWidth;
% Cumulative overland flow passing through the OFE (m)
ovldf=watdata(:,7)./1000;
% Cumulative lateral flow passing through the OFE (m)
latf=watdata(:,12)./1000;
% Percolation (m) (non-cumulative)
perc=watdata(:,10)./1000;
% Cumulative sediment leaving an OFE per width (kg/m)
```

```matlab
sedkgm=watdata(:,14);
% Cumulative sediment leaving an OFE (kg)
sed=zeros(watrow,1);
for i=1:1:watrow
    sed(i)=sedkgm(i)*ofeWidth(watdata(i,1));
end;
% Net sediment loss (kg) (non-cumulative)
netsed=zeros(watrow,1);
for i=2:1:watrow
    if watdata(i,1)>1
        mass=sedkgm(i)*ofeWidth(watdata(i,1));
        massprev=sedkgm(i-1)*ofeWidth(watdata(i-1,1));
        netsed(i)=mass-massprev;
    else
        mass=sedkgm(i)*ofeWidth(watdata(i,1));
        netsed(i)=mass;
    end;
end;
% Crop transpiration (m)
cropevap=watdata(:,8)./1000;
% Soil evaporation (m)
soilevap=watdata(:,9)./1000;
% Evapotranspiration (m)
et=cropevap+soilevap;
% Soil moisture (m)
soilmoist=watdata(:,13)./1000;
% Average temp (C)
avgT=watdata(:,16);

%% Parameters

% Initialize parameters for each crop
% Initialize parameters for each crop
ps=zeros(max(croptype),numofe);
om=zeros(max(croptype),numofe);
thetaS=zeros(max(croptype),numofe);
thetaFC=zeros(max(croptype),numofe);
koc=zeros(max(croptype),numofe);
thalf=zeros(max(croptype),numofe);
for o=1:1:numofe
    for c=1:1:max(croptype);
        % Soil bulk density (g/cm^3)
        ps(c,o)=userinputdata(5+c-1,o);
        % Convert ps to kg/m^3
        ps(c,o)=ps(c,o)*1000;
        % Water content at saturation, import from WEPP
        thetaS(c,o)=userinputdata(9+c-1,o);
        % Field capacity (thetaFC)
        thetaFC(c,o)=userinputdata(13+c-1,o);
        % Percent organic matter in the mixing layer
        om(c,o)=userinputdata(21+c-1,numofe);
        % Pesticide (here, atrazine) organic carbon adsorption coefficient
        % (cm^3/g), see Excel spreadsheet for values
        koc(c,o)=userinputdata(26+c-1,o);
        % Convert to m^3/kg
        koc(c,o)=koc(c,o)/1000;
```

```matlab
        % Half-life for atrazine (or other pesticide) (days)
        thalf(c,o)=userinputdata(30+c-1,o);
    end;
end;
% Max root depth (m), for each crop
% Second colm is buffer root depth
rootdepthshrt=scheddata(:,12:13); % convert to m
rootdepth=zeros(croprow,1);
for i=1:1:croprow
    if cropdata(i,9)==max(croptype) % for buffer
        rflag=find(scheddata(:,1)==cropdata(i,3));
        r=rootdepthshrt(rflag,2);
        rsel=unique(r);
        rootdepth(i)=rsel;
    else % for non-buffer crops
        rflag=find(scheddata(:,1)==cropdata(i,3));
        r=rootdepthshrt(rflag,1);
        rsel=unique(r);
        rootdepth(i)=rsel;
    end;
end;
% Incorporation (yes=1, no=0);
incorp=userinputdata(25,:);
% Percent organic carbon
oc=(om./100).*0.58;
% Adsorption partition coefficient (see Sinkevich 2005) (m^3/kg)
apc=koc.*oc;
% Density of water (g/cm^3) at 20C
pw=0.9980;
% Convert pw to kg/m^3
pw=pw*1000;
% Time step (for decay, in days)
tstep=1;


%% Water Content

% Also considers soil water content losses due to crop and soil evaporation
% ( residue evaporation is left out)
theta=(soilmoist-et)./rootdepth;


%% Pesticide Application Information (date and amount), Planting Date, and
Harvest Date

% Application location and amount in kg/m^2
applnofe=scheddata(:,4:7)./10000;
% Application and plow dates
applndate=scheddata(:,3);
plowdate=scheddata(:,10);
% Convert date to a cumulative date for the simulation
[sr,sc]=size(scheddata);
applndatecum=zeros(sr,1);
plowdatecum=zeros(sr,1);
for s=1:1:sr
    % Application of pesticide
    if applndate(s)>0
        if scheddata(s,1)==1 % first year
```

```matlab
            applndatecum(s)=applndate(s);
        else % second year and on
            applndatecum(s)=applndate(s)+sum(dayINyearlst(1:scheddata(s,1)-
1));
        end;
    else % just in case
        applndatecum(s)=0;
    end;
    % Plowing
    if plowdate(s)>0
        if scheddata(s,1)==1 % first year
            plowdatecum(s)=plowdate(s);
        else % second year and on
            plowdatecum(s)=plowdate(s)+sum(dayINyearlst(1:scheddata(s,1)-1));
        end;
    else % just in case
        plowdatecum(s)=0;
    end;
end;
plowdepth=scheddata(:,11);

%% Initialize Matrices

% Pesticide in the mixing layer (kg/m^2)
pestleftTop=zeros(numsimdays,numofe);
% Pesticide in the deep layer (kg/m^2)
pestleftBot=zeros(numsimdays,numofe);
% Total pesticide lost in overland flow with sediment (dissolved and
adsorbed)
pestlostovlfTot=zeros(numsimdays,numofe);
% Dissolved pesticide lost in overland flow pesticide (kg/m^2)
pestlostovlfW=zeros(numsimdays,numofe);
% Sorbed pesticide lost in overland flow pesticide (kg/m^2)
pestlostovlfS=zeros(numsimdays,numofe);
% Pesticide lost due to lateral flow (kg/m^2)
pestlostlatf=zeros(numsimdays,numofe);
% Pesticide lost due to shallow percolation (kg/m^2) (to below mixing layer)
pestlostshalperc=zeros(numsimdays,numofe);
% Pesticide lost due to percolation (kg/m^2)
pestlostperc=zeros(numsimdays,numofe);
% Total pesticide lost (kg/m^2)
totalpestlost=zeros(numsimdays,numofe);
% Sediment fraction (kg/kg)
sedfraclst=zeros(numsimdays,numofe);
% Pesticide trapped in buffer (kg/m^2)
bufferPest=zeros(numsimdays,numofe);
% Mixing depth
hmix=zeros(numsimdays,numofe);

%% Daily Simulation

k=1;
for id=numofe+1:1:watrow %start at second day and end on second to last day
    i=daylst(id);
    ofe=ofelst(id);
    % Upslope Pesticide Contributions (dissolved pesticide and particulate
```

```matlab
    % pesticide in runoff as well as dissolved pesticide in lateral flow)
    if ofe>1
        lostPestovldfWUP=pestlostovlfW(i,ofe-1);
        lostPestovldfSUP=pestlostovlfS(i,ofe-1);
        lostPestlaftUP=pestlostlatf(i,ofe-1);
    else
        lostPestovldfWUP=0;
        lostPestovldfSUP=0;
        lostPestlaftUP=0;
    end;
    % Calculate fraction of runoff make up of sediment (kg/kg)
    if ovldf(id)>0
        ovldfvol=ovldf(id)*ofeArea(ofe); % runoff volume
        ovldfmass=ovldfvol*pw;
        sedfrac=sed(id)/ovldfmass;
    else
        sedfrac=0;
    end;
     % Calculate pesticide applied to ofe (kg/actual applied ofe area in m^2)
    bermwidth=userinputdata(40,ofe); % m
    width=userinputdata(39,ofe); % m
    applnarea=(width-2*bermwidth)*(ofeLength(ofe)-2*bermwidth);
    % Identify application day for the year
    d=find(scheddata(:,1)==k); % identify position
    allappdays=applndatecum(d(:));
    if intersect(allappdays,i)>0
        % Application day (cumulative)
        applnday=intersect(allappdays,i);
        % Application row
        applnrow=find(applndatecum(:)==applnday);
        % Application amount (convert to kg)
        applnamt=unique(applnofe(applnrow,ofe))*applnarea;
    else
        % Application day (cumulative)
        applnday=0;
        % Application amount (kg/m^2)
        applnamt=0;
    end;
    % Define depth of mixing layer
    allplowdays=plowdatecum(d(:));
    if intersect(allplowdays,i)>0
        % Plow day
        plowday=intersect(allplowdays,i);
        % Plow row
        plowrow=find(plowdatecum(:)==plowday);
        % Mixing layer depth (m)
        mixdepth=unique(plowdepth(plowrow));
    else
        plowday=0;
        if ofe==numofe % Buffer
            mixdepth=fixedinputdata(1);
        else
            mixdepth=userinputdata(41,ofe);
        end;
    end;

    % Set ET of buffer equal to previous OFE
```

```matlab
    if ofe==numofe
        etnow=et(id-1);
    else
        etnow=et(id);
    end;

    % Non-uniform application with transport between OFEs

out=dailyofePestsimTransBuff(id,i,k,tstep,apc(croptype(id),ofe),applnamt,mixd
epth,hmix(i-
1,ofe),ps(croptype(id),ofe),pw,thetaS(croptype(id),ofe),rootdepth(id),applnda
y,plowday,thalf(croptype(id),ofe),pestleftTop(i-1,ofe),pestleftBot(i-
1,ofe),lostPestovldfWUP,lostPestovldfSUP,lostPestlaftUP,ris(id),etnow,ovldf(i
d),latf(id),perc(id),sedfrac,ofe,buffernum);
    pestlostovlfTot(out(3),out(2))=out(5);  % Dissolved and particulate
pesticide lost in overland flow
    pestlostovlfW(out(3),out(2))=out(6);     % Dissolved pesticide lost in
overland flow
    pestlostovlfS(out(3),out(2))=out(7);     % Particulate pesticide lost in
overland flow
    pestlostlatf(out(3),out(2))=out(8);      % Dissolved pesticide lost in
lateral flow
    pestlostshalperc(out(3),out(2))=out(9); % Dissolved pesticide lost in
shallow percolation (to transition layer)
    pestlostperc(out(3),out(2))=out(10);     % Dissolved pesticide lost in
deep percolation (to groundwater)
    totalpestlost(out(3),out(2))=out(5)+out(8)+out(10); % Total pesticide
lost (dissolved and particulate)
    pestleftTop(out(3),out(2))=out(11);      % Pesticide left in the mixing
layer
    pestleftBot(out(3),out(2))=out(12)+out(9);  % Pesticide left in the
transition layer
    sedfraclst(i,ofe)=sedfrac;                        % Sediment fraction
    bufferPest(out(3),numofe)=out(13);
    hmix(i,ofe)=mixdepth;

    % Itterator
    if daylst(id)<=sum(dayINyearlst(1:k))
        k=k+0;
    else
        k=k+1;
    end;
end;

%% Daily

% Save daily data (kg) (need to divide by area to get cumulative kg/ha)
dailyPestdata=[pestleftTop,pestleftBot,pestlostovlfTot,pestlostovlfW,pestlost
ovlfS,pestlostlatf,pestlostshalperc,pestlostperc];

%% Montly

% Month list for the entire simulation
mbiglst=zeros(mpy*numyrs,1);
srt=1;
for k=1:1:numyrs
```

```matlab
    if dayINyearlst(k)==365
        mbiglst(srt:srt+mpy-1)=mlst;
    else
        mbiglst(srt:srt+mpy-1)=mlstleap;
    end;
    srt=srt+mpy;
end;
% Cumulative monthly list for the entire simulation
mbiglstcum=zeros(mpy*numyrs,1);
for j=1:1:mpy*numyrs
    if j==1
        mbiglstcum(j)=mbiglst(1);
    else
        mbiglstcum(j)=sum(mbiglst(1:j));
    end;
end;
% Use mbiglstcum to add up all daily processes
monthlylostPesterossum=zeros(mpy*numyrs,numofe);
monthlylostPestovldfnosedsum=zeros(mpy*numyrs,numofe);
monthlylostPestlatfsum=zeros(mpy*numyrs,numofe);
monthlylostPestpercsum=zeros(mpy*numyrs,numofe);
srt=1;
for j=1:1:mpy*numyrs
    for i=1:1:numofe
        monthlylostPesterossum(j,i)=sum(pestlostovlfS(srt:mbiglstcum(j),i));

monthlylostPestovldfnosedsum(j,i)=sum(pestlostovlfW(srt:mbiglstcum(j),i));
        monthlylostPestlatfsum(j,i)=sum(pestlostlatf(srt:mbiglstcum(j),i));
        monthlylostPestpercsum(j,i)=sum(pestlostperc(srt:mbiglstcum(j),i));
    end;
    srt=mbiglstcum(j)+1;
end;


% Save montly data (kg) (need to divide by area to get cumulative kg/ha)
monthlyPestdata=[monthlylostPesterossum,monthlylostPestovldfnosedsum,monthlyl
ostPestlatfsum,monthlylostPestpercsum];


%% Yearly

yearlylostPesterossum=zeros(numyrs,numofe);
yearlylostPestovldfnosedsum=zeros(numyrs,numofe);
yearlylostPestlatfsum=zeros(numyrs,numofe);
yearlylostPestpercsum=zeros(numyrs,numofe);
srt=1;
for k=1:1:numyrs
    for i=1:1:numofe

yearlylostPesterossum(k,i)=sum(pestlostovlfS(srt:sum(dayINyearlst(1:k)),i));

yearlylostPestovldfnosedsum(k,i)=sum(pestlostovlfW(srt:sum(dayINyearlst(1:k))
,i));

yearlylostPestlatfsum(k,i)=sum(pestlostlatf(srt:sum(dayINyearlst(1:k)),i));

yearlylostPestpercsum(k,i)=sum(pestlostperc(srt:sum(dayINyearlst(1:k)),i));
    end;
```

78

```
    srt=srt+dayINyearlst(k);
end;


% Save yearly data (kg) (need to divide by area to get cumulative kg/ha)
yearlyPestdata=[yearlylostPesterossum,yearlylostPestovldfnosedsum,yearlylostP
estlatfsum,yearlylostPestpercsum];

%% Yearly (for total hillslope, i.e. the last OFE) (kg)

yearlyPesteroshill=zeros(numyrs,1);
yearlyPestovldfnosedhill=zeros(numyrs,1);
yearlyPestlatfhill=zeros(numyrs,1);
yearlyPestperchill=zeros(numyrs,1);


for k=1:1:numyrs
    yearlyPesteroshill(k)=yearlylostPesterossum(k,numofe)-
yearlylostPesterossum(k,numofe-1);
    yearlyPestovldfnosedhill(k)=yearlylostPestovldfnosedsum(k,numofe)-
yearlylostPestovldfnosedsum(k,numofe-1);
    yearlyPestlatfhill(k)=yearlylostPestlatfsum(k,numofe)-
yearlylostPestlatfsum(k,numofe-1);
    yearlyPestperchill(k)=yearlylostPestpercsum(k,numofe);
end;


% Yearly data at bottom of the hillslope (kg) (need to divide by area to
% get cumulative kg/ha)
yearlyPesthilldata=[yearlyPesteroshill,yearlyPestovldfnosedhill,yearlyPestlat
fhill,yearlyPestperchill];

%% Yearly Fraction (for total hillslope, i.e. the last OFE) (%)

% Make a matrix with total application rates for each OFE for each year
yearlyPestapplnlst=zeros(numyrs,numofe);
for k=1:1:numyrs
    x=find(scheddata(:,1)==k);
    xamt=scheddata(x,4:4+numofe);
    for ofe=1:1:numofe
        yearlyPestapplnlst(k,ofe)=sum(xamt(:,ofe))/10000*ofeArea(ofe);
    end;
end;


yearlyPesteroshillfrac=zeros(numyrs,1);
yearlyPestovldfnosedhillfrac=zeros(numyrs,1);
yearlyPestlatfhillfrac=zeros(numyrs,1);
yearlyPestperchillfrac=zeros(numyrs,1);


for k=1:1:numyrs
    totyrapplnamt=sum(yearlyPestapplnlst(k,:));
    if totyrapplnamt>0
    yearlyPesteroshillfrac(k,1)=yearlyPesteroshill(k)/totyrapplnamt*100;

yearlyPestovldfnosedhillfrac(k,1)=yearlyPestovldfnosedhill(k)/totyrapplnamt*1
00;
    yearlyPestlatfhillfrac(k,1)=yearlyPestlatfhill(k)/totyrapplnamt*100;
    yearlyPestperchillfrac(k,1)=yearlyPestperchill(k)/totyrapplnamt*100;
```

```matlab
    else
    yearlyPesteroshillfrac(k,1)=0;
    yearlyPestovldfnosedhillfrac(k,1)=0;
    yearlyPestlatfhillfrac(k,1)=0;
    yearlyPestperchillfrac(k,1)=0;
    end;
end;


% Yearly data at bottom of the hillslope (kg) (need to divide by area to
% get cumulative kg/ha)
yearlyPesthilldatafrac=[yearlyPesteroshillfrac,yearlyPestovldfnosedhillfrac,y
earlyPestlatfhillfrac,yearlyPestperchillfrac];

%% Yearly Averages by OFE (kg)

yearlylostPesterosofeavg=zeros(1,numofe);
yearlylostPestovldfnosedofeavg=zeros(1,numofe);
yearlylostPestlatfofeavg=zeros(1,numofe);
yearlylostPestpercofeavg=zeros(1,numofe);

for ofe=1:1:numofe
    if ofe>1
        yearlylostPesterosofeavg(1,ofe)=(sum(yearlylostPesterossum(:,ofe))-
sum(yearlylostPesterossum(:,ofe-1)))/numyrs;

yearlylostPestovldfnosedofeavg(1,ofe)=(sum(yearlylostPestovldfnosedsum(:,ofe)
)-sum(yearlylostPestovldfnosedsum(:,ofe-1)))/numyrs;
        yearlylostPestlatfofeavg(1,ofe)=(sum(yearlylostPestlatfsum(:,ofe))-
sum(yearlylostPestlatfsum(:,ofe-1)))/numyrs;

yearlylostPestpercofeavg(1,ofe)=sum(yearlylostPestpercsum(:,ofe))/numyrs;
    else

yearlylostPesterosofeavg(1,ofe)=sum(yearlylostPesterossum(:,ofe))/numyrs;

yearlylostPestovldfnosedofeavg(1,ofe)=sum(yearlylostPestovldfnosedsum(:,ofe))
/numyrs;

yearlylostPestlatfofeavg(1,ofe)=sum(yearlylostPestlatfsum(:,ofe))/numyrs;

yearlylostPestpercofeavg(1,ofe)=sum(yearlylostPestpercsum(:,ofe))/numyrs;
    end;
end;

% Yearly average (kg) (need to divide by area to get cumulative kg/ha)
yearlyPestavgdata=[yearlylostPesterosofeavg;yearlylostPestovldfnosedofeavg;ye
arlylostPestlatfofeavg;yearlylostPestpercofeavg];

%% Yearly Net Averages by OFE (%)

yearlylostPesterosavgfrac=zeros(1,numofe);
yearlylostPestovldfnosedavgfrac=zeros(1,numofe);
yearlylostPestlatfavgfrac=zeros(1,numofe);
yearlylostPestpercavgfrac=zeros(1,numofe);
```

```matlab
for ofe=1:1:numofe
    appfrac=sum(sum(yearlyPestapplnlst(:,ofe)));
    if appfrac>0

yearlylostPesterosavgfrac(1,ofe)=yearlylostPesterosofeavg(1,ofe)/appfrac*100;

yearlylostPestovldfnosedavgfrac(1,ofe)=yearlylostPestovldfnosedofeavg(1,ofe)/
appfrac*100;

yearlylostPestlatfavgfrac(1,ofe)=yearlylostPestlatfofeavg(1,ofe)/appfrac*100;

yearlylostPestpercavgfrac(1,ofe)=yearlylostPestpercofeavg(1,ofe)/appfrac*100;
    else
        yearlylostPesterosavgfrac(1,ofe)=0;
        yearlylostPestovldfnosedavgfrac(1,ofe)=0;
        yearlylostPestlatfavgfrac(1,ofe)=0;
        yearlylostPestpercavgfrac(1,ofe)=0;
    end;
end;

% Yearly average (kg) (need to divide by area to get cumulative kg/ha)
yearlyPestavgdatafrac=[yearlylostPesterosavgfrac;yearlylostPestovldfnosedavgf
rac;yearlylostPestlatfavgfrac;yearlylostPestpercavgfrac];

%% Calculating kg/ha Lost for Each Hillslope Section

pestovldWkgha=zeros(numsimdays,numofe);
pestovldSkgha=zeros(numsimdays,numofe);
pestlatfkgha=zeros(numsimdays,numofe);
pestperckgha=zeros(numsimdays,numofe);

for i=1:1:numsimdays
    for ofe=1:1:numofe
        % dissolved pesticide lost in overland flow
        if ofe==1
            pestovldWkgha(i,ofe)=pestlostovlfW(i,ofe)./ofeArea(ofe).*1e4;
        else
            if (pestlostovlfW(i,ofe)-pestlostovlfW(i,ofe-1))<0
                pestovldWkgha(i,ofe)=0;
            else
                pestovldWkgha(i,ofe)=(pestlostovlfW(i,ofe)-
pestlostovlfW(i,ofe-1))./ofeArea(ofe).*1e4;
            end;
        end;

        % particulate pesticide lost in overland flow
        if ofe==1
            pestovldSkgha(i,ofe)=pestlostovlfS(i,ofe)./ofeArea(ofe).*1e4;
        else
            if (pestlostovlfS(i,ofe)-pestlostovlfS(i,ofe-1))<0
                pestovldSkgha(i,ofe)=0;
            else
                pestovldSkgha(i,ofe)=(pestlostovlfS(i,ofe)-
pestlostovlfS(i,ofe-1))./ofeArea(ofe).*1e4;
            end;
        end;
```

```matlab
        % dissolved pesticide lost in lateral flow
        if ofe==1
            pestlatfkgha(i,ofe)=pestlostlatf(i,ofe)./ofeArea(ofe).*1e4;
        else
            if (pestlostlatf(i,ofe)-pestlostlatf(i,ofe-1))<0
                pestlatfkgha(i,ofe)=0;
            else
                pestlatfkgha(i,ofe)=(pestlostlatf(i,ofe)-pestlostlatf(i,ofe-
1))./ofeArea(ofe).*1e4;
            end;
        end;

        % dissolved pesticide lost percolation (not cumulative)
        pestperckgha(i,ofe)=pestlostperc(i,ofe)./ofeArea(ofe).*1e4;

    end;
end;

% Compute the average for the simulation
pestkgha=zeros(numofe,4);
for ofe=1:1:numofe
    pestkgha(ofe,1)=sum(pestovldWkgha(:,ofe))/numyrs;
    pestkgha(ofe,2)=sum(pestovldSkgha(:,ofe))/numyrs;
    pestkgha(ofe,3)=sum(pestlatfkgha(:,ofe))/numyrs;
    pestkgha(ofe,4)=sum(pestperckgha(:,ofe))/numyrs;

end;

% Convert to a percentage based on the maximum value
pestkghapercent=zeros(numofe,4);
for i=1:1:4
    totalval=sum(pestkgha(:,i));
    pestkghapercent(:,i)=pestkgha(:,i)./totalval*100;
end;



%% Plot Pesticide Losses per OFE

% Saving plots
% Generating bar graphs for each run
plotstr1=sprintf('fig_output');
plotstr2=sprintf('Run%02.0f',h);

% Convert run number to a string
runnum=num2str(h);

bar(pestkgha)
title(['Run  ',runnum],'FontSize',14,'FontName','Times');
ylabel('Pesticide Lost (kg/ha)','FontSize',14,'FontName','Times');
xlabel('OFE','FontSize',14,'FontName','Times');
legend('Dissolved (Ovldf)','Particulate
(Ovldf)','Latf','Perc','Location','Best');
```

```matlab
set(gca,'XTickLabel',{'Top','Mid','Bot','Buf'},'FontSize',14,'FontName','Time
s')
colormap gray
box off

figure1=gcf;
plotfile1=[outputfolder,'\',plotstr1,'_pest_',plotstr2];
saveas(figure1,plotfile1,'png');
clf(figure1,'reset');

%% Plot Percent Pesticide Losses per OFE

bar(transpose(pestkghapercent))
title(['Run  ',runnum],'FontSize',14,'FontName','Times');
ylabel('Pesticide Lost (%)','FontSize',14,'FontName','Times');
xlabel('Hydrologic Process','FontSize',14,'FontName','Times');
legend('Top','Mid','Bot','Buf','Location','Best');
set(gca,'XTickLabel',{'Ovldf (Diss.)','Ovldf
(Part.)','Latf','Perc'},'FontSize',14,'FontName','Times')
colormap gray
box off

figure2=gcf;
plotfile2=[outputfolder,'\',plotstr1,'_pest_percent_',plotstr2];
saveas(figure2,plotfile2,'png');
clf(figure2,'reset');
```

'dailyofePestsimTrans.m'

```matlab
% Daily Pesticide Transport Function

% This function calculates the change in soil pesticide over time for a thin
% mixing layer at the surface and a thicker transition layer (root
% depth-mixing layer depth=transition layer depth).  These changes as well
% as the losses from the soil pesticide pool are exported as outputs
% expressed in units kg/m^2. Additions and processes included in this
% function are: additions from up-slope OFEs, pesticide application,
% leaching (dissolved and adsorbed pesticide), and decay.

% References: Steenhuis and Walter 1980, Sinkevich et al 2005

function
pesticideDailyDATA=dailyofePestsimTrans(id,day,yr,tstep,apc,applnamt,mixdepth
,hmixprev,ps,pw,thetaS,rootdepth,applnday,plowday,thalf,pestleftTop,pestleftB
ot,lostPestovldfWUP,lostPestovldfSUP,lostPestlaftUP,rain,evap,qovlf,qlatf,qpe
rc,sed,ofe)

% Rename incoming variables
pestleftTopnow=pestleftTop;
pestleftBotnow=pestleftBot;

% Transported Pesticides from up-slope OFEs
pestleftTopnow=pestleftTopnow+lostPestovldfWUP+lostPestovldfSUP;
pestleftBotnow=pestleftBotnow+lostPestlaftUP;

% Plowing day indicates a mixing of pesticide to the deeper transition
% later based on the plow depth, determine fraction that enters by
% comparing the previous mixing depth to the current mixing depth (=plow
% depth)
if day==plowday
    fplow=hmixprev/mixdepth;
    pestleftTopnow=fplow*pestleftTopnow;
    pestleftBotnow=(1-fplow)*pestleftTopnow+pestleftBotnow;
end;

% For calculations with runoff we want smallest mixing depth
hmixnow=min(hmixprev,mixdepth);

% Application of pesticide to mixing layer of soil (=top layer)
if day==applnday
    if applnamt>0
        pestleftTopnow=pestleftTopnow+applnamt;
        % No pesticide lost to decay in the top layer the day it's added
        % but pesticide is lost from the bottom layer
        pestleftBotnow=pestleftBotnow*exp(-0.69*tstep/thalf);
    else
        % Pesticide lost to decay
        pestleftTopnow=pestleftTopnow*exp(-0.69*tstep/thalf);
        pestleftBotnow=pestleftBotnow*exp(-0.69*tstep/thalf);
    end;
```

84

```matlab
else
    % Pesticide lost to decay
    pestleftTopnow=pestleftTopnow*exp(-0.69*tstep/thalf);
    pestleftBotnow=pestleftBotnow*exp(-0.69*tstep/thalf);
end;

if qovlf==0 % No runoff
    %No pesticide lost to overland flow (dissolved or sediment bound)
    %because soil is not saturated
    pestlostovlfTot=0;
    pestlostovlfW=0;
    pestlostovlfS=0;

    %Pesticide lost to shallow percolation (from mixing to deeper layer)
    if (pestleftTopnow>0)&&(rain-evap>0)
        pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
        if pestlostshalperc<=pestleftTopnow
            pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
            pestleftTopnow=pestleftTopnow-pestlostshalperc;
            pestleftBotnow=pestleftBotnow+pestlostshalperc;
        else
            pestleftTopnow=0;
            pestleftBotnow=pestleftBotnow+pestleftTopnow;
            pestlostshalperc=pestleftTopnow;
        end;
    else
        pestlostshalperc=0;
    end;

    %Pesticide lost in lateral flow (from below mixing layer)
    if (pestleftBotnow>0)&&(qlatf>0)
        pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostlatf<=pestleftBotnow
            pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostlatf;
        else
            pestleftBotnow=0;
            pestlostlatf=pestleftBotnow;
        end;
    else
        pestlostlatf=0;
    end;

    %Pesticide lost in percolation (from below mixing layer)
    if (pestleftBotnow>0)&&(qperc>0)
        pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostperc<=pestleftBotnow
            pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostperc;
        else
```

```matlab
            pestleftBotnow=0;
            pestlostperc=pestleftBotnow;
        end;
    else
        pestlostperc=0;
    end;


elseif qovlf>0
    %Overland flow with sediment (m)
    qovlfx=qovlf*(1+(((sed*apc)/(1-sed))*pw));
    %Total pesticide lost
    pestlostovlfTot=pestleftTopnow*(1-exp(-
(qovlfx)/(hmixnow*(thetaS+apc*ps))));
    if pestlostovlfTot<=pestleftTopnow
        pestlostovlfTot=pestleftTopnow*(1-exp(-
(qovlfx)/(hmixnow*(thetaS+apc*ps))));
        pestleftTopnow=pestleftTopnow-pestlostovlfTot;
        %Dissolved pesticide lost in overland flow (kg/m^2)
        pestlostovlfW=(pestlostovlfTot*(1-sed))/(1+sed*(apc*pw-1));
        %Sorbed pesticide lost in overland flow (kg/m^2)
        pestlostovlfS=(pestlostovlfTot*apc*pw*sed)/(1+sed*(apc*pw-1));
    else
        pestlostovlfTot=pestleftTopnow;
        pestleftTopnow=0;
        %Dissolved pesticide lost in overland flow (kg/m^2)
        pestlostovlfW=(pestlostovlfTot*(1-sed))/(1+sed*(apc*pw-1));
        %Sorbed pesticide lost in overland flow (kg/m^2)
        pestlostovlfS=(pestlostovlfTot*apc*pw*sed)/(1+sed*(apc*pw-1));
    end;

    %Pesticide lost to shallow percolation (from mixing to deeper layer)
    if (pestleftTopnow>0)&&(rain-evap>0)
        pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
        if pestlostshalperc<=pestleftTopnow
            pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
            pestleftTopnow=pestleftTopnow-pestlostshalperc;
            pestleftBotnow=pestleftBotnow+pestlostshalperc;
        else
            pestleftTopnow=0;
            pestleftBotnow=pestleftBotnow+pestleftTopnow;
            pestlostshalperc=pestleftTopnow;
        end;
    else
        pestlostshalperc=0;
    end;

    %Pesticide lost in lateral flow (from below mixing layer)
    if (pestleftBotnow>0)&&(qlatf>0)
        pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostlatf<=pestleftBotnow
            pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostlatf;
```

```
        else
            pestleftBotnow=0;
            pestlostlatf=pestleftBotnow;
        end;
    else
        pestlostlatf=0;
    end;

    %Pesticide lost in percolation (from below mixing layer)
    if (pestleftBotnow>0)&&(qperc>0)
        pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostperc<=pestleftBotnow
            pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostperc;
        else
            pestleftBotnow=0;
            pestlostperc=pestleftBotnow;
        end;
    else
        pestlostperc=0;
    end;
end;

pesticideDailyDATA=[id,ofe,day,yr,pestlostovlfTot,pestlostovlfW,pestlostovlfS
,pestlostlatf,pestlostshalperc,pestlostperc,pestleftTopnow,pestleftBotnow];
```

'dailyofePestsimTransBuff.m'

```matlab
% Daily Pesticide Transport Function

% This function calculates the change in soil pesticide over time for a thin
% mixing layer at the surface and a thicker transition layer (root
% depth-mixing layer depth=transition layer depth).  These changes as well
% as the losses from the soil pesticide pool are exported as outputs
% expressed in units kg/m^2. Additions and processes included in this
% function are: additions from up-slope OFEs, pesticide application,
% leaching (dissolved and adsorbed pesticide), and decay.

% References: Steenhuis and Walter 1980, Sinkevich et al 2005

function
pesticideDailyDATA=dailyofePestsimTransBuff(id,day,yr,tstep,apc,applnamt,mixd
epth,hmixprev,ps,pw,thetaS,rootdepth,applnday,plowday,thalf,pestleftTop,pestl
eftBot,lostPestovldfWUP,lostPestovldfSUP,lostPestlaftUP,rain,evap,qovlf,qlatf
,qperc,sed,ofe,buffernum)

% Rename incoming variables
pestleftTopnow=pestleftTop;
pestleftBotnow=pestleftBot;

% Transported Pesticides from up-slope OFEs
pestleftTopnow=pestleftTopnow+lostPestovldfWUP+lostPestovldfSUP;
pestleftBotnow=pestleftBotnow+lostPestlaftUP;

% Plowing day indicates a mixing of pesticide to the deeper transition
% later based on the plow depth, determine fraction that enters by
% comparing the previous mixing depth to the current mixing depth (=plow
% depth)
if day==plowday
    fplow=hmixprev/mixdepth;
    pestleftTopnow=fplow*pestleftTopnow;
    pestleftBotnow=(1-fplow)*pestleftTopnow+pestleftBotnow;
end;

% For calculations with runoff we want smallest mixing depth
hmixnow=min(hmixprev,mixdepth);

% Application of pesticide to mixing layer of soil (=top layer)
if day==applnday
    if applnamt>0
        pestleftTopnow=pestleftTopnow+applnamt;
        % No pesticide lost to decay in the top layer the day it's added
        % but pesticide is lost from the bottom layer
        pestleftBotnow=pestleftBotnow*exp(-0.69*tstep/thalf);
    else
        % Pesticide lost to decay
        pestleftTopnow=pestleftTopnow*exp(-0.69*tstep/thalf);
        pestleftBotnow=pestleftBotnow*exp(-0.69*tstep/thalf);
    end;
```

```matlab
    else
        % Pesticide lost to decay
        pestleftTopnow=pestleftTopnow*exp(-0.69*tstep/thalf);
        pestleftBotnow=pestleftBotnow*exp(-0.69*tstep/thalf);
end;

if qovlf==0 % No runoff
    if ofe<buffernum
        % No pesticide lost to overland flow (dissolved or sediment bound)
        % because soil is not saturated
        pestlostovlfTot=0;
        pestlostovlfW=0;
        pestlostovlfS=0;
        bufferPest=0;
    else
        pestlostovlfW=lostPestovldfWUP;
        pestlostovlfTot=lostPestovldfWUP;
        pestlostovlfS=0;
        bufferPest=lostPestovldfSUP;
    end;

    %Pesticide lost to shallow percolation (from mixing to deeper layer)
    if (pestleftTopnow>0)&&(rain-evap>0)
        pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
        if pestlostshalperc<=pestleftTopnow
            pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
            pestleftTopnow=pestleftTopnow-pestlostshalperc;
            pestleftBotnow=pestleftBotnow+pestlostshalperc;
        else
            pestleftTopnow=0;
            pestleftBotnow=pestleftBotnow+pestleftTopnow;
            pestlostshalperc=pestleftTopnow;
        end;
    else
        pestlostshalperc=0;
    end;

    %Pesticide lost in lateral flow (from below mixing layer)
    if (pestleftBotnow>0)&&(qlatf>0)
        pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostlatf<=pestleftBotnow
            pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostlatf;
        else
            pestleftBotnow=0;
            pestlostlatf=pestleftBotnow;
        end;
    else
        pestlostlatf=0;
    end;

    %Pesticide lost in percolation (from below mixing layer)
```

```
    if (pestleftBotnow>0)&&(qperc>0)
        pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostperc<=pestleftBotnow
            pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostperc;
        else
            pestleftBotnow=0;
            pestlostperc=pestleftBotnow;
        end;
    else
        pestlostperc=0;
    end;


elseif qovlf>0
    %Pesticide lost with sediment in overland flow
    %Calculate sediment concentration in overland flow
    if qovlf>0
        %Overland flow with sediment (m)
        qovlfx=qovlf*(1+(((sed*apc)/(1-sed))*pw));
        %Total pesticide lost
        pestlostovlfTot=pestleftTopnow*(1-exp(-
(qovlfx)/(hmixnow*(thetaS+apc*ps))));
        if ofe<buffernum
            if pestlostovlfTot<=pestleftTopnow
                pestlostovlfTot=pestleftTopnow*(1-exp(-
(qovlfx)/(hmixnow*(thetaS+apc*ps))));
                pestleftTopnow=pestleftTopnow-pestlostovlfTot;
                %Dissolved pesticide lost in overland flow (kg/m^2)
                pestlostovlfW=(pestlostovlfTot*(1-sed))/(1+sed*(apc*pw-1));
                %Sorbed pesticide lost in overland flow (kg/m^2)
                pestlostovlfS=(pestlostovlfTot*apc*pw*sed)/(1+sed*(apc*pw-
1));
                bufferPest=0;
            else
                pestlostovlfTot=pestleftTopnow;
                pestleftTopnow=0;
                %Dissolved pesticide lost in overland flow (kg/m^2)
                pestlostovlfW=(pestlostovlfTot*(1-sed))/(1+sed*(apc*pw-1));
                %Sorbed pesticide lost in overland flow (kg/m^2)
                pestlostovlfS=(pestlostovlfTot*apc*pw*sed)/(1+sed*(apc*pw-
1));
                bufferPest=0;
            end;
        else
            if pestlostovlfTot<=pestleftTopnow
                pestlostovlfTot=pestleftTopnow*(1-exp(-
(qovlfx)/(hmixnow*(thetaS+apc*ps))));
                pestleftTopnow=pestleftTopnow-pestlostovlfTot;
                %Dissolved pesticide lost in overland flow (kg/m^2)
                pestlostovlfW=(pestlostovlfTot*(1-sed))/(1+sed*(apc*pw-1));
                %Sorbed pesticide lost in overland flow (kg/m^2)
                pestlostovlfS=0;
                bufferPest=(pestlostovlfTot*apc*pw*sed)/(1+sed*(apc*pw-1));
            else
                pestlostovlfTot=pestleftTopnow;
```

```matlab
                pestleftTopnow=0;
                %Dissolved pesticide lost in overland flow (kg/m^2)
                pestlostovlfW=(pestlostovlfTot*(1-sed))/(1+sed*(apc*pw-1));
                %Sorbed pesticide lost in overland flow (kg/m^2)
                pestlostovlfS=0;
                bufferPest=(pestlostovlfTot*apc*pw*sed)/(1+sed*(apc*pw-1));
            end;
        end;
    else
        pestlostovlfTot=0;
        pestlostovlfW=0;
        pestlostovlfS=0;
        bufferPest=0;
    end;

    %Pesticide lost to shallow percolation (from mixing to deeper layer)
    if (pestleftTopnow>0)&&(rain-evap>0)
        pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
        if pestlostshalperc<=pestleftTopnow
            pestlostshalperc=pestleftTopnow*(1-exp(-(rain-
evap)/(hmixnow*(thetaS+apc*ps))));
            pestleftTopnow=pestleftTopnow-pestlostshalperc;
            pestleftBotnow=pestleftBotnow+pestlostshalperc;
        else
            pestleftTopnow=0;
            pestleftBotnow=pestleftBotnow+pestleftTopnow;
            pestlostshalperc=pestleftTopnow;
        end;
    else
        pestlostshalperc=0;
    end;

    %Pesticide lost in lateral flow (from below mixing layer)
    if (pestleftBotnow>0)&&(qlatf>0)
         pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
         if pestlostlatf<=pestleftBotnow
            pestlostlatf=pestleftBotnow*(1-exp(-(qlatf)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
            pestleftBotnow=pestleftBotnow-pestlostlatf;
         else
            pestleftBotnow=0;
            pestlostlatf=pestleftBotnow;
         end;
    else
        pestlostlatf=0;
    end;

    %Pesticide lost in percolation (from below mixing layer)
    if (pestleftBotnow>0)&&(qperc>0)
        pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
        if pestlostperc<=pestleftBotnow
            pestlostperc=pestleftBotnow*(1-exp(-(qperc)/((rootdepth-
hmixnow)*(thetaS+apc*ps))));
```

```
            pestleftBotnow=pestleftBotnow-pestlostperc;
        else
            pestleftBotnow=0;
            pestlostperc=pestleftBotnow;
        end;
    else
        pestlostperc=0;
    end;
end;
pesticideDailyDATA=[id,ofe,day,yr,pestlostovlfTot,pestlostovlfW,pestlostovlfS
,pestlostlatf,pestlostshalperc,pestlostperc,pestleftTopnow,pestleftBotnow,buf
ferPest];
```

REFERENCES

Arnold, J.G., R. Srinivasan, R.S. Muttiah and J.R. Williams. 1998. Large Area Hydrologic Modeling and Assessment Part I: Model Development. *Journal of the American Water Resources Association*. 34(1):73-89.

Ascough, J.C., C. Baffaut, M.A. Nearing and B.Y. Liu. 1997. The WEPP Watershed Model: I. Hydrology and Erosion. *Transactions of the ASAE*. 40(4):921-933.

Baffaut, C., M.A. Nearing, J.C. Ascough and B.Y. Liu. 1997. The WEPP Watershed Model: II. Sensitivity Analysis and Discretization on Small Watersheds. *Transactions of the ASAE*. 40(4):935-943.

Blanco-canqui, H., C.J. Gantzer, S.H. Anderson and E.E. Alberts. 2004. Tillage and Crop Influences on Physical Properties for an Epiaqualf. *Soil Science Society of America Journal*. 68(2):567-576.

Boll, J., E.S. Brooks, B. Crabtree and S. Dun. Variable Source Area Hydrology Modeling with the Water Erosion Prediction Project (WEPP) Model. *Journal of Hydrology*. (in review).

Brooks, E.S. and J. Boll. 2011. Building process-based understanding for improved adaptation and management. In *International Symposium on Erosion and Landscape Evolution CD-ROM Proceedings*. Anchorage, Alaska; St. Joseph, Michigan: American Society of Agricultural and Biological Engineers, ASABE. pp. 18-21.

Burroughs, E.R. and J.G. King. 1989. *Reduction of Soil Erosion on Forest Roads Agriculture Reduction of Soil Erosion on Forest*. General Technical Notes. Orgden, UT: Intermountain Research Station.

Cerdeira, A.L. and S.O. Duke. 2006. The current status and environmental impacts of glyphosate-resistant crops: a review. *Journal of Environmental Quality*. 35(5):1633-1658.

Chung, S.O., A.D. Ward and C.W. Schalk. 1992. Evaluation of the hydrologic component of the ADAPT water table management model. *Transactions of the ASAE*. 35(2):571-579.

Colborn, T., F.S. vom Saal and A.M. Soto. 1993. Developmental effects of endocrine-disrupting chemicals in wildlife and humans. *Environmental health perspectives*. 101(5):378-84.

Dillaha, T.A., R.B. Reneau, S. Mostaghimi and D. Lee. 1989. Vegetative Filter Strips for Agricultural Nonpoint Source Pollution Control. *Transactions of the ASAE*. 32(2):513-519.

Dunne, T. 1970. *Runoff production in humid areas*,

Dunne, T. and R.D. Black. 1970. Partial Area Contributions to Storm Runoff in a Small New England Watershed. *Water Resources Research*. 6(5):1296-1311.

D'Arcy, B. and A. Frost. 2001. The role of best management practices in alleviating water quality problems associated with diffuse pollution. *The Science of the Total Environment*. 265(1-3):359-367.

Easton, Z.M., D.R. Fuka, M.T. Walter, D.M. Cowan, E.M. Schneiderman and T.S. Steenhuis. 2008a. Re-conceptualizing the soil and water assessment tool (SWAT) model to predict runoff from variable source areas. *Journal of Hydrology*. 348(3-4):279-291.

Easton, Z.M., M.T. Walter, E.M. Schneiderman, M.S. Zion and T.S. Steenhuis. 2009. Including Source-Specific Phosphorus Mobility in a Nonpoint Source Pollution Model for Agricultural Watersheds. *Journal of Environmental Engineering*. 135(1):25-35.

Easton, Z.M., M.T. Walter and T.S. Steenhuis. 2008b. Combined monitoring and modeling indicate the most effective agricultural best management practices. *Journal of Environmental Quality*. 37(5):1798-1809.

Enz, J.W., L.J. Brun and J.K. Larsen. 1988. Evaporation and energy balance for bare and stubble covered soil. *Agriculture and Forest Meteorology*. 43:59-70.

Flanagan, D.C., J.E. Gilley and T.G. Franti. 2007. Water Erosion Prediction Project (WEPP): Development History, Model Capabilities, and Future Enhancements. *Transactions of the ASABE*. 50(5):1603-1612.

Flanagan, D.C. and M.A. Nearing. 1995. *USDA Water Erosion Prediction Project: Hillslope Profile and Watershed Model Documentation*. West Lafayette, IN: USDA-ARS.

Frankenberger, J.R., E.S. Brooks, M.T. Walter, M.F. Walter and T.S. Steenhuis. 1999. A GIS-based variable source area hydrology model. *Hydrological Processes*. 13(6):805-822.

Gburek, W.J., C.C. Drungil, M.S. Srivivasan, B.A. Needelman and D.E. Woodward. 2002. Variable-source-area controls on phosphorus transport: Bridging the gap between research and design. *Journal of Soil and Water Conservation*. 57(6):534-543.

Ghidey, F., C. Baffaut, R.N. Lerch, N.R. Kitchen, E.J. Sadle and K.A. Sudduth. 2010. Herbicide transport to surface runoff from a claypan soil: Scaling from plots to fields. *Journal of Soil and Water Conservation*. 65(3):168-179.

Ghidey, F., P. Blanchard, R. Lerch, N. Kitchen, E. Alberts and E. Sadler. 2005. Measurement and simulation of herbicide transport from corn phase of three cropping systems. *Soil and Water Conservation Society*. 60(5):260-273.

Greer, R.C., J.Q. Wu, P. Singh and D.K. McCool. 2006. WEPP Simulation of Observed Winter Runoff and Erosion in the U.S. Pacific Northwest. *Vadose Zone Journal*. 5(1):261-272.

Haith, D.A. and L.L. Shoemaker. 1987. Generalized watershed loading functions for stream flow nutrients. *Journal of the American Water Resources Association*. 23(3):471-478.

Heathwaite, A.L. and P.J. Johnes. 1996. Contribution of Nitrogen Species and Phosphorus Fractions To Stream Water Quality in Agricultural Catchments. *Hydrological Processes*. 10(7):971-983.

Hively, W.D., P. Gérard-Marchant and T.S. Steenhuis. 2006. Distributed hydrological modeling of total dissolved phosphorus transport in an agricultural landscape, part II: dissolved phosphorus transport. *Hydrology and Earth System Sciences*. 10(2):263-276.

Horton, R.E. 1933. The role of infiltration in the hydrologic cycle. *Transactions of the American Geophysical Union*. 14:446-460.

Jiang, P., S.H. Anderson, N.R. Kitchen, E.J. Sadler and K.A. Sudduth. 2007. Landscape and Conservation Management Effects on Hydraulic Properties of a Claypan-Soil Toposequence. *Soil Science Society of America Journal*. 71(3):803.

Jung, W.K., N.R. Kitchen, S.H. Anderson and E.J. Sadler. 2007. Crop management effects on water infiltration for claypan soils. *Journal of Soil and Water Conservation*. 62(1):55-63.

Laflen, J.., W.J. Elliot, D.C. Flanagan, C.R. Meyer and M.A. Nearing. 1997. WEPP-predicting water erosion using a process-based model. *Journal of Soil and Water Conservation*. 52(2):96-102.

Larose, M., J.L. Oropeza-Mota, D. Norton, A. Turrent-Fernandez, M. Martinex-Menes, J.A. Pedraza-Oropeza and N. Francisco-Nicolas. 2004. Application of the WEPP model to hillside lands in the Tuxtlas, Veracruz, Mexico. *Agrociencia Volumen*. 38(2):155-163.

Larson, S.J., C.G. Crawford and R.J. Gilliom. 2004. *Development and application of Watershed Regression for Pesticides (WARP) for estimating atrazine concentration distributions in streams*, Reston, Virginia.

Lerch, R.N., P.E. Blanchard and E.M. Thurman. 1998. Contribution of Hydroxylated Atrazine Degradation Products to the Total Atrazine Load in Midwestern Streams. *Environmental Science & Technology*. 32(1):40-48.

Lerch, R.N., E.J. Sadler, K. Sudduth, C. Baffaut and N.R. Kitchen. 2011. Herbicide Transport in Goodwater Creek Experimental Watershed: I. Long-Term Research on Atrazine. *Journal of the American Water Resources Association*. 47(2):209-223.

McDaniel, P.A., M.P. Regan, E. Brooks, J. Boll, S. Barndt, A. Falen, S.K. Young and J.E. Hammel. 2008. Linking fragipans, perched water tables, and catchment-scale hydrological processes. *Catena*. 73(2):166-173.

Mein, R.G. and C.L. Larson. 1973. Modeling infiltration during a steady-state rain. *Water Resources Research*. 9(2):384-394.

Merriman, K.R., M.W. Gitau and I. Chaubey. 2009. A tool for estimating best management practice effectiveness in Arkansas. *Applied Engineering in Agriculture*. 25(2):199-214.

Miao, Z., A. Vicari, E. Capri, F. Ventura, L. Padovani and M. Trevisan. 2004. Organic compounds in the environment: Modeling the effects of tillage management practices on herbicide runoff in northern Italy. *Journal of Environmental Quality*. 33(5):1720-32.

Moriasi, D.N., J.G. Arnold, M.W.V. Liew, R.L. Bingner, R.D. Harmel and T.L. Veith. 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*. 50(3):885-900.

Mostaghimi, S., S.W. Park, R.A. Cooke and S.Y. Wang. 1997. Assessment of management alternatives on a small agricultural watershed. *Water Resources*. 31(8):1867-1878.

Mudgal, A., C. Baffaut, S.H. Anderson, E.J. Sadler and A.L. Thompson. 2010. APEX Model Assessment of Variables Landscapes on Runoff and Dissolved Herbicides. *Transactions of the ASABE*. 53(4):1047-1058.

Müller, K., R. Stenger and a. Rahman. 2006. Herbicide loss in surface runoff from a pastoral hillslope in the Pukemanga catchment (New Zealand): Role of pre-event soil water content. *Agriculture, Ecosystems & Environment*. 112(4):381-390.

Nash, J.E. and J.V. Sutcliffe. 1970. River flow forecasting through conceptual models, part 1—A discussion of principles. *Journal of Hydrology*. 10:282–290.

Needelman, B.A., W.J. Gburek, G.W. Petersen, A.N. Sharpley and P.J.A. Kleinman. 1998. Surface Runoff along Two Agricultural Hillslopes with Contrasting Soils. *Instrumentation*. 68:914-923.

Nicks, A.D., L.J. Lane and G. Gander. 1995. Weather Generator. D. Flanagan & M.A. Nearing, eds. *USDA-Water Erosion Prediction Project: Hillslope Profile and Watershed Model Documentation*.

Pieri, L., M. Bittelli, J. Wu, S. Dun, D. Flanagan, P. Pisa, F. Ventura and F. Salvatorelli. 2007. Using the Water Erosion Prediction Project (WEPP) model to simulate field-observed runoff and erosion in the Apennines mountain range, Italy. *Journal of Hydrology*. 336(1-2):84-97.

Rao, N.S., Z.M. Easton, E.M. Schneiderman, M.S. Zion, D.R. Lee and T.S. Steenhuis. 2009. Modeling watershed-scale effectiveness of agricultural best management practices to reduce phosphorus loading. *Journal of Environmental Management*. 90(3):1385-1395.

Reichenberger, S., M. Bach, A. Skitschak and H.-G. Frede. 2007. Mitigation strategies to reduce pesticide inputs into ground- and surface water and their effectiveness; A review. *The Science of the Total Environment*. 384(1-3):1-35.

Renard, K.., G.R. Foster, G.A. Weesies, D.K. McCool and D.C. Yoder. 1997. Predicting soil erosion by water: A guide to conservation planning with the Revised Universal Soil Loss Equation (RUSLE). *Agricultural Handbook 703*. Washington, DC: U.S. Government Print Office.

Renard, K.G. 1991. RUSLE-Revised Universal Soil Loss Equation. *Journal of Soil and Water Conservation*. 46(1):30-33.

Richards, B.K., S. Pacenka, A.E. Salvucci, S.M. Saia, L.F. Whitbeck and P.M. Furdyna. 2012. Surveying Upstate NY Well Water for Pesticide Contamination : Cayuga and Orange Counties. *Ground Water Monitoring and Remediation*. 23(1):73-82.

Sadler, E.J., R.N. Lerch, E.E. Alberts and T.L. Oster. Long term hydrologic database: Goodwater Creek, Missouri. In *2nd Interagency Conf. on Research in the Watersheds*. Coweeta Hydrologic Laboratory, Otto, NC, :161-168.

Schneiderman, E.M., D.C. Pierson, D.G. Lounsbury and M.S. Zion. 2002. Modeling the hydrochemistry of the Cannonsville Watershed with Generalized Watershed Loading Functions (GWLF). *Journal of American Water Resources Association*. 38(5):1323-1347.

Schneiderman, E.M., T.S. Steenhuis, D.J. Thongs, Z.M. Easton, M.S. Zion, A.L. Neal, G.F. Mendoza and M.T. Walter. 2007. Incorporating variable source area hydrology into a curve-number-based watershed model. *Hydrological Processes*. 21(25):3420-3430.

Scribner, E.A., E.M. Thurman, D.A. Goolsby, M.T. Meyer, W.A. Battaglin and D.W. Kolpin. 2005. *Summary of Significant Results from Studies of Triazine Herbicides and Their Degradation Products in Surface Water, Ground Water, and Precipitation in the Midwestern United States During the 1990s*. Reston, VA: USGS.

Singh, P., J.Q. Wu, D.K. McCool, S. Dun, C.-H. Lin and J.R. Morse. 2009. Winter Hydrologic and Erosion Processes in the U.S. Palouse Region: Field Experimentation and WEPP Simulation. *Vadose Zone Journal*. 8(2):426-436.

Sinkevich, M.G., M.T. Walter, A.J. Lembo, B.K. Richards, N. Peranginangin, S. a. Aburime and T.S. Steenhuis. 2005. A GIS-Based Ground Water Contamination Risk Assessment Tool for Pesticides. *Ground Water Monitoring and Remediation*. 25(4):82-91.

Solomon, K.R., J.A. Carr, L.H. Du Preez, J.P. Giesy, R.J. Kendall, E.E. Smith and G.J. Van Der Kraak. 2008. Effects of atrazine on fish, amphibians, and aquatic reptiles: a critical review. *Critical Reviews in Toxicology*. 38(9):721-772.

Steenhuis, T.S. and R.E. Muck. 1988. Preferred Movement of Nonadsorbed Chemicals on Wet, Shallow, Sloping Soils. *Journal of Environment Quality*. 17(3):376.

Steenhuis, T.S. and M.F. Walter. 1980. Closed Form Solution for Pesticide Loss in Runoff Water. *Transactions of the ASAE*. 23(3):615-628.

Steiner, J.L., E.J. Sadler, G. Wilson, J.L. Hatfield, D. James, B. Vandenberg, J.S. Chen, T. Oster, J.D. Ross and K. Cole. 2009. STEWARDS watershed data system: system design and implementation. *Transactions of the ASABE*. 52(5):1523-1533.

Tomer, M.D. and M.A. Locke. 2011. The challenge of documenting water quality benefits of conservation practices: a review of USDA-ARS's conservation effects assessment project watershed studies. *Water Science & Technology*. 64(1):300.

Tong, J.-X., J.-Z. Yang, B.X. Hu and R. Bao. 2010. Experimental study and mathematical modelling of soluble chemical transfer from unsaturated/saturated soil to surface runoff. *Hydrological Processes*. 24(21):3065-3073.

United States Department of Agriculture, Natural Resources Conservation Service, USDA-NRCS, 2003. Soil Survey Geographic (SSURGO) Database. *Soil Survey Geographic (SSURGO) Database. Washington, DC. Retrieved February 2012, from http://soils.usda.gov/survey/geography/ssurgo/*.

United States Department of Agriculture, Natural Resources Conservation Service, USDA-NRCS. 2006a. National Conservation Practice Standards. Washington, DC: USDA.

United States Department of Agriculture, Natural Resources Conservation Service, USDA-NRCS. 2006b. Urban BMP's – Stream System Protection, Restoration, and Reestablishment. Washington, DC: USDA.

United States Department of Agriculture, Natural Resources Conservation Service, USDA-NRCS. 2006c. *Urban BMP's – Water Runoff Management*. Washington, DC: USDA.

United States Environmental Protection Agency, USEPA. 1996. *Drinking Water Regulations and Health Advisories*, Washington, DC: United States Government Print Office.

Walter, M.T., M.F. Walter, E.S. Brooks, T.S. Steenhuis, J. Boll and K. Weiler. 2000. Hydrologically Sensitive Areas : Variable Source Area Hydrology Implications for Water Quality Risk Assessment. *Journal of Soil and Water Conservation*. 55(3):277-284.

Williams, J.D., S. Dun, D.S. Robertson, J.Q. Wu, E.S. Brooks, D.C. Flanagan and D.K. McCool. 2010. WEPP simulations of dryland cropping systems in small drainages of northeastern Oregon. *Journal of Soil and Water Conservation*. 65(1):22-33.

Wischmeier, W.H. and D.D. Smith. 1978. Predicting rainfall erosion losses-a guide to conservation planning. *Agriculture*. Washington, DC: U.S. Department of Agriculture Science and Education Administration.

Wolgast-Broberg, Å. 2002. Evaporation from a Clay Soil Covered with a Surface Mulch - Three Lysimeter Studies. *Acta Agriculturae Scandinavica, Section B - Soil & Plant Science*. 52(4):121-126.

Young, R.A., C.A. Onstad, D.D. Bosch and W.P. Anderson. 1989. AGNPS: A nonpoint-source pollution model for evaluating agricultural watersheds. *Journal of Soil and Water Conservation*. 44(2):168-173.

Zollweg, J.A. 1994. *Effective use of geographic information systems for rainfall-runoff modeling*. Cornell University, Ithaca, NY.

Zollweg, J.A., W.J. Gburek and T.S. Steenhuis. 1996. SMoRMod - A GIS-integrated rainfall-runoff model. *Transactions of the ASABE*. 39(4):1299-1307.