SOME NEW DEVELOPMENTS ON TWO SEPARATE TOPICS: STATISTICAL CROSS VALIDATION AND FLOODPLAIN MAPPING

BY

Jude H. Kastens

Submitted to the graduate degree program in Mathematics and the
Graduate Faculty of the University of Kansas
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy.

David Lerner
Chairperson*

James Church

Ben Cobb

Stephen Egbert

Don Huggins

Erik Van Vleck

Date Defended July 1, 2008

The Dissertation Committee for Jude H. Kastens certifies
that this is the approved version of the following dissertation:

SOME NEW DEVELOPMENTS ON TWO SEPARATE TOPICS:
STATISTICAL CROSS VALIDATION AND FLOODPLAIN MAPPING

Committee:

David Lerner
Chairperson*

James Church

Ben Cobb

Stephen Egbert

Don Huggins

Erik Van Vleck

Date Approved: July 1, 2008

Acknowledgments

I would like to thank David Lerner (my advisor) and the rest of the KU Mathematics Department for their support over the years. David never gave up on me despite my floundering around with many different research topics before finding my focus. David and committee members Jim Church and Ben Cobb provided valuable insight and encouragement as well as excellent guidance for my research into statistical cross validation. Conversations with committee member Erik Van Vleck were very helpful in getting me to "cut to the chase" on my floodplain mapping research.

Also I would like to thank Ed Martinko and the rest of the Kansas Biological Survey (KBS) for providing me with employment throughout my Ph.D. program. Without this job, none of this would have been possible. And this is not just from a financial standpoint, but also from a research standpoint—many of the core ideas behind this dissertation stemmed from my research at KBS and interactions with my fellow KBS employees. In particular I would like to thank committee member Don Huggins, who was involved with my floodplain mapping research from the beginning and essentially exposed me to the problem. With his experience and keen insight, his role as a sounding board was invaluable to help me smooth out the kinks of the FLDPLN model. Committee member Steve Egbert has been extremely supportive in our recent pursuit of applications for the model, which are finding much success. Many conversations with (and materials from) Bob Everhart were invaluable in helping me to better understand the workings of existing floodplain models. Last but certainly not least, Kevin Dobbs has been a tremendous help throughout many aspects of this research. Kevin's help has been instrumental in validating, demonstrating, and improving the FLDPLN model, and his commitment toward establishing applications for the model in many ways have exceeded my own.

Thank you mom (Marilyn) for your nurturing kindness and the many weekends you've spent taking care of my kids so that I could work on research. Thank you dad (Terry) for setting the bar high and introducing me to cross validation, as well as our many valuable conversations about modeling and prediction. Thank you brother (Dietrich) for blazing a path in front of me, overcoming obstacles that I could learn to avoid and achieving success that I could emulate. Thanks to all three of you for setting a good example and teaching me self-reliance.

This work is dedicated to my wife Tracey and our three children, Marlowe, Quincy, and Elliot. Tracey is a wonderful mother who has been very supportive over the years. Thanks to her, I've been able to earn a Ph.D. while working full-time and raising a family. Our kids are a blessing, a marvelous distraction that helps keep me both grounded and excited. My family is my source of inspiration. Everything that I do is for them. Maybe now we can take a REAL vacation.

Abstract

This dissertation describes two unrelated threads of research. The first is a study of *cross validation* (CV), which is a data resampling method. CV is used for model ranking in model selection and for estimating expected prediction error of a model. A review of three resampling methods is provided in Chapter 1. Chapter 2 contains results from simulations that examine various properties of CV, in particular the use of CV for model selection in small sample settings as well as the expected value of the delete-*d* cross validation statistic.

The second research thread is described in Chapter 3, where a new, physically-based computational model (called FLDPLN, or "Floodplain") for mapping potential inundation extents (floodplains) using gridded topographic data is introduced. Due to the parametric economy of FLDPLN, this model has significant advantages over existing methods such as hydrodynamic models. The model is validated using imagery from an actual flood event.

Chapter 1. A Review of Resampling Methods

Chapter Summary

In model selection, the primary task is to assign competing models a "fitness" value by which the models can be ranked and superior models identified. As more models are considered, competition bias becomes more of a problem in that chance can increasingly influence the outcome. Bolstered by theoretical and empirical results, robust data resampling methods have become popular in model selection to mitigate the effects of competition bias. Resampling methods find another (not unrelated) area of applicability in the general statistical modeling situation when sample size gets small and distributional uncertainty increases, and robust estimates for prediction error are needed. The bootstrap and cross validation, preceded in development by the jackknife, are two of the most commonly employed resampling methods when addressing problems of the nature just described. In this context, the bootstrap involves repeatedly drawing (with replacement) new samples from the data and computing the statistic of interest to obtain an approximate empirical distribution (from which inferences can be made) for that statistic. Cross validation involves repeated systematic splitting of the data into two subsets, building a statistical model using one subset and evaluating predictive ability of the model using the other subset. The jackknife uses a regimented resampling approach similar to cross validation but otherwise is like the bootstrap in design. This chapter provides an introduction to the jackknife, the bootstrap, and cross validation in the context of ordinary least squares linear regression modeling.

1.1. Introduction

With the advent of high-speed computers, data reuse methods have come into favor in the statistical modeling community. In particular, difficult questions such as model selection (which requires ranking competing models) and small sample model evaluation are being addressed using methods such as the bootstrap and cross validation. Both tasks require a robust method for assessing general model predictive ability, and are related for this reason (Davison & Hinkley 1997, p.290). Distributional assumptions become more dubious as (*i*) more models are considered, leading to greater chance of competition bias when ranking competing models using some pre-specified "fitness" measure; and (*ii*) sample size gets smaller, leading to increased variance of the sample itself when considering it as a draw from the underlying distribution. Both of these situations promote the tendency toward results that overfit the data, which occurs when noise, or unexplained variation in the dependent variable, excessively influences model parameter estimation through chance correspondence with variation in the independent variable(s). That said, as all theoretical results to be discussed are asymptotic[1] in nature, the "small sample" questions will have to remain in the background, with scant illumination, while model selection comes to the forefront.

In recent decades, practitioners forced to deal with the problems mentioned above frequently turn to robust methods that are more data driven than reliant on computations explicitly dependent on distributional assumptions. The jackknife, the

[1] In applied statistics, "asymptotic results" typically refer to theoretical findings that depend on unlimited sample size, i.e., $n \rightarrow \infty$.

bootstrap, and cross validation are three of the most frequently encountered robust resampling methods. Though the jackknife is less frequently used in the context of data modeling in favor of one of the other two methods, it did provide the foundation from which the other two methods were spawned, and warrants presentation for this reason.

The interest of the author in resampling strategies involves their use in statistical modeling, so that will be the framework in which the methods are presented. In spite of this restriction, the general applicability of the methods for parameter or model characterization should remain clear. Furthermore, the author also has an interest in small sample modeling ($n \in \{4,\ldots,20\}$, say), where ordinary least squares (OLS) linear regression remains the most commonly used modeling tool. Thus the presentation framework is further constrained to the OLS regression modeling arena, but this also happens to be one of the more heavily explored contexts with respect to the resampling methods that comprise the focus of this chapter.

What immediately follows is mathematical stage-setting regarding OLS regression modeling, as it typically appears in both theory and practice. Much of the text regarding the comparison between errors (ε) and residuals (e) that appears in the next section is lifted nearly verbatim or paraphrased from Cook & Weisberg (1982, pp.10-11). After presenting the basics of OLS regression modeling, three resampling methods are introduced in the following order: the jackknife, the bootstrap, and cross validation. Some theoretical and practical aspects of the methods will be highlighted

as the methods are presented. The chapter concludes with a summary of the presentation.

1.2. Basics of Ordinary Least Squares Linear Regression Modeling

The following is the typical set-up surrounding OLS linear regression modeling in the context of parametric theory, at least in cases where model errors are presumed to be second order independent and identically distributed (IID). Let $p \leq n$ be positive integers (n = sample size, p = number of model parameters) and let I_n (or just I, dropping the subscript) denote the identity matrix.

Let $Y = X\beta + \varepsilon$ be the general linear statistical model, where $\varepsilon \sim (0, \sigma^2 I)$ [or equivalently $Y \sim (X\beta, \sigma^2 I)$]. Assume that:

- $X \in \mathbb{R}^{n \times p}$ is of full rank (this condition will be made more stringent below), with each row corresponding to an observation of the p explanatory variables
- $Y \in \mathbb{R}^{n \times 1}$ = n-vector of responses (observations of the dependent variable)
- $\beta \in \mathbb{R}^{p \times 1}$ = p-vector of responses (unobservable "true" model parameters)
- $\varepsilon \in \mathbb{R}^{n \times 1}$ = n-vector of unobservable errors with the indicated distributional properties

1.2.1. Linear Algebra

Let X and Y be defined as above. Assume that X, sometimes referred to as the *design matrix*, has the property that all p-row submatrices of X (say X_S, where $S \subseteq N = \{1,\ldots,n\}$, $|S| = p$) are linearly independent. Note that this is more restrictive than the above assertion that X be of full rank. This condition will be convenient in the context of resampling methods in the OLS regression setting. Let $||\cdot||$ denote the l^2 norm in $\mathbb{R}^{n\text{x}1}$, so that $||a||^2 = a^{\mathrm{T}}a$ for $a \in \mathbb{R}^{n\text{x}1}$.

Consider the orthogonal decomposition $Y = \hat{Y} + e = VY + (\mathrm{I} - V)Y$, where $\hat{Y} = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y = X\hat{\beta} = VY$ is the projection of Y onto the column space of X [Col(X)] and $e = Y - \hat{Y} = Y - VY = (\mathrm{I} - V)Y$ the projection of Y onto the null space of X^{T} [Null(X^{T})].

Projection matrix $V = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}} \in \mathbb{R}^{n\text{x}n}$ is often referred to as the "hat" matrix for X (because it maps Y into $\hat{Y}$), and $\hat{\beta} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y \in \mathbb{R}^{p\text{x}1}$ is the least-squares regression parameter vector. $\hat{\beta}$ provides an estimate for the unknown β defined above. Likewise, e (residuals) will serve as a proxy for ε (errors), which is also unknown.

1.2.2. Statistics

The above material outlines the standard OLS linear regression modeling setup. Each row of the system [X Y] corresponds to a single observation of independent (X) and dependent (Y) variable values. Each column of X corresponds to a particular independent variable, which may or may not have predictive power when estimating Y.

In order to make inferences about the expected accuracy of the linear model $Y \approx X\hat{\beta}$, two assumptions regarding ε have been imposed. First it is assumed that $E[\varepsilon] = 0$, implying that fluctuations of Y about $X\beta$ have 0 mean. This indicates the model $X\beta$ is unbiased for Y (i.e., $E[X\beta] = E[Y]$). The second assumption is that $\text{Var}[\varepsilon] = \sigma^2 I_n$, which says that the elements of ε are of constant variance and are statistically independent (have 0 covariance). Together, these conditions indicate that the elements of ε are IID up to second order, i.e., $\varepsilon \sim (0, \sigma^2 I_n)$.

Under these error assumptions, it can be shown that $E[\hat{\beta}] = \beta$ and $\text{Var}[\hat{\beta}] = \sigma^2(X^T X)^{-1}$. The latter equation is a generalization of the standard formula for variance about the mean, which states that $\text{Var}[\overline{Y}] = \sigma^2/n$ (define X to be a column vector of ones to obtain the mean model $\hat{\beta} = \overline{Y}$). In fact, $\hat{\beta}$ can be shown to be the best linear unbiased estimate for β (Seber & Lee 2003; pp.42-43). If we further tighten our restrictions to $\varepsilon \sim N(0, \sigma^2 I)$, then $\hat{\beta}$ is the maximum likelihood estimate as well as the most efficient estimate for β (Seber & Lee 2003; pp.49-50).

To determine the appropriateness of the linear regression model for a given problem, it is necessary to determine if the assumptions about the errors are

reasonable. Since the error vector ε is unobservable, this must be done indirectly using residual vector e.

From above, we have:

$$e = Y - \hat{Y} = Y - VY = (\mathrm{I} - V)Y.$$

Substituting $X\beta + \varepsilon$ for Y, we get

$$e = (\mathrm{I} - V)(X\beta + \varepsilon) = (\mathrm{I} - V)\varepsilon. \qquad (1.1)$$

Thus if $\varepsilon \sim (0, \sigma^2\mathrm{I})$, then $e \sim (0, \sigma^2(\mathrm{I} - V))$ [$(\mathrm{I} - V)$ is idempotent], and the variation in e is controlled by V.

Note that $V \in \mathbb{R}^{nxn}$ is symmetric ($V = V^{\mathrm{T}}$) and idempotent ($V = V^2$), and is the linear transform that orthogonally projects any n-vector onto $\mathrm{Col}(X)$. $(\mathrm{I} - V)$ has the same properties, except that it orthogonally projects any n-vector onto $\mathrm{Null}(X^{\mathrm{T}})$. Since V is idempotent and symmetric it follows that V is invariant (up to rearrangement of columns) under non-singular reparametrizations, which are equivalent to changing the basis of $\mathrm{Col}(X)$. This property implies that, aside from computational concerns, collinearity between the columns of X is irrelevant to understanding how V (and thus e) behaves. On the other hand, such collinearity can have undesirable effects if one is trying to evaluate the statistical behavior of $\hat{\beta}$ since

$\text{Var}[\hat{\beta}] = \sigma^2(X^{\text{T}}X)^{-1}$. See Cook & Weisberg (1982, pp.12-15) for a succinct discussion regarding how the elements of V can be used to reveal point-specific data characteristics that can possibly influence model performance (e.g., outlier detection). For a much broader perspective on applied linear regression analysis in general, two excellent reference texts are Draper & Smith (1998) and Seber & Lee (2003).

If a variable (column) is added to X, then almost always we find $e^{\text{T}}e$, the squared length of the projection of Y onto $\text{Null}(X^{\text{T}})$, gets smaller. If we remove a column, then $e^{\text{T}}e$ generally gets larger. However, when considering the first and second order statistics of the linear regression model, adding a variable will generally decrease the bias and increase the variance of future predicted values (Miller 2002, p.5; Seber & Lee 2003, pp.394-397). The tradeoff between these two quantities (bias and variance of prediction) underlies much of the uncertainty in linear regression modeling.

The above discussion helps detail some of the low-order statistical behavior of the standard linear regression situation under the assumption of second order IID errors. A large body of linear regression modeling theory hangs on the ability to use e to make inferences regarding model efficacy, through its connection to ε. However, as effective degrees of freedom decrease (i.e., when more models are considered during a model selection exercise, or when sample size decreases in the general situation), uncertainty in this connection increases as effects of model overfitting begin to creep in and e increasingly strays from ε. This situation leads practitioners to seek alternative, robust methods for assessing model efficacy that do not depend so

heavily on the distributional assumptions regarding ε, which generally can never be known. Data resampling provides one particular framework for developing such methods. Shortly we will turn our attention to resampling methods, but first we must introduce some basic concepts in regression error.

1.2.3. Regression Error

Define the *mean squared error of regression* (MSE), which is the maximum likelihood estimate of error variance (σ^2) under normal error assumptions:

$$\text{MSE} = (1/n)\left|\left|Y - X\hat{\beta}\right|\right|^2 = (1/n)\left|\left|e\right|\right|^2 = (e^{\mathrm{T}}e)/n.$$

The least squares regression estimate $\hat{\beta}$ for β minimizes MSE, which frequently underlies ordering statistics used for ranking competing models. Upon accounting for parameter estimation bias (i.e., the almost inevitable overfitting of the sample data by $\hat{\beta}$ during regression), we can define the *expected error of regression* (REG), which provides an unbiased estimate of error variance (σ^2) under the standard, second-order IID error assumptions:

$$\text{REG} = (n/(n-p))\text{MSE} = (1/(n-p))\left|\left|Y - X\hat{\beta}\right|\right|^2.$$

We now prove that REG provides an unbiased estimate for σ^2 in our setting. This proof is a simplification of the proof appearing in Seber & Lee (2003; Theorem 3.3, pp.44-45).

THEOREM 1.1: Let X, Y, V, ε, and e be defined as above, with $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2 I$. Then $E[REG] = \sigma^2$.

PROOF: From (1.1) we have $e = (I - V)\varepsilon$. Since $(I - V)$ is symmetric and idempotent, we have:

$$REG = (e^T e)/(n - p) = (\varepsilon^T (I - V)^T (I - V)\varepsilon)/(n - p) = (\varepsilon^T (I - V)\varepsilon)/(n - p).$$

A special case of Theorem 1.5 in Seber & Lee (2003, p.9) is now needed. This theorem states that if W is an n-by-1 vector of random variables such that $E[W] = 0$ and $Var[W] = \sigma^2 I$, and A is an n-by-n symmetric matrix, then $E[W^T A W] = \sigma^2 tr(A)$. In light of this result, we have

$$E[\varepsilon^T (I - V)\varepsilon] = \sigma^2 tr(I - V).$$

Proposition (A.6.2) in Seber & Lee (2003, p.464) states that if P is a projection matrix, then $tr(P) = rank(P)$. Thus we have $tr(I - V) = (n - p)$, from which it follows that

$$E[REG] = E[\varepsilon^T(I - V)\varepsilon]/(n - p) = \sigma^2(n - p)/(n - p) = \sigma^2. \qquad QED$$

Both MSE and REG are multiples of e^Te, which is the sum of squared errors of the regression. Since e^Te almost always decreases when additional variables are included in a model, using MSE for model ranking almost always favors the higher dimensional model. Though the unbiasedness of REG mitigates this problem, it is still easily influenced by overfitting (i.e., chance covariation of the error vector ε with one or more variables in X) when engaging in either model optimization or model selection. Such behavior suggests that better model selection criteria (of which many are presently available) are needed beyond these standard estimates of model error variance.

1.3. Resampling Methods

Resampling methods involve data reuse. In the present context, the sample at hand is used to define an empirical distribution function (EDF), with each point assigned the same probability mass. This EDF is presumed to be a proxy for the underlying probability distribution function (PDF) from which the initial sample was taken. This presumption underlies the "plug-in principle" that is often mentioned in bootstrapping texts, and obviously loses legitimacy with decreasing sample size just like other asymptotically motivated results.

We will look at three resampling strategies, in the following order: (*i*) the jackknife, (*ii*) the bootstrap, and (*iii*) cross validation. As will be discussed, the bootstrap is essentially a useful generalization of the jackknife. Cross validation (CV) has similarities with the jackknife in its resampling strategy, but fundamentally departs from both the jackknife and the bootstrap in that it is grounded in prediction rather than parameter estimation. Thus we discuss CV last, even though CV preceded the bootstrap in the literature. Due to the prevalence of both the bootstrap and CV in practical settings, more attention will be given to these two methods than to the jackknife.

The formal history of the jackknife extends back to the mid-1900s (Quenouille 1949 and 1956; Tukey 1958; Gray & Schucany 1972). Cross validation was introduced in the mid-1970s (Allen 1974; Stone 1974; Geisser 1975), first as a special case (delete-1 CV) and then in its more general form (delete-*d* CV). Shortly thereafter, in the late 1970s the bootstrap was conceptually formalized (Efron 1979 and 1982).

As will be explained, the jackknife and CV sample the EDF without replacement in order to define "new" samples. The bootstrap, on the other hand, will define "new" samples by sampling the EDF with replacement. CV is the only method of the three that attempts to simulate out-of-sample behavior, an important trait when attempting to assess model predictive properties.

1.3.1. The Jackknife

The *jackknife* statistic (a moniker coined by Tukey) was introduced by Quenouille (1949) for the purpose of reducing bias when estimating serial correlation in time series. Later, Quenouille (1956) somewhat generalized the definition so that it served the purpose of reducing bias of some desired parameter estimate. Tukey (1958) was the first to apply the concept to variance estimation. Gray & Schucany (1972) proposed a more fully generalized statistical concept and provided the first comprehensive overview. Shao & Tu (1995), working from an alternative generalization of the jackknife, present a more contemporary, thorough examination of the subject, largely from a measure-theortic perspective.

Let $\hat{\theta}_1$ and $\hat{\theta}_2$ be two estimates for statistic θ, and suppose $R \neq 1$. The general form (Gray & Schucany 1972. p.2) for the jackknife estimate of θ obtained from $\hat{\theta}_1$ and $\hat{\theta}_2$ is given by $G\left(\hat{\theta}_1, \hat{\theta}_2\right) = \dfrac{\hat{\theta}_1 - R\hat{\theta}_2}{1 - R}$. The parameter R is dependent on the statistic being estimated as well as its distribution. However, it is also a function of sample size n.

Define $\hat{\theta}_{(j)}$ to be the estimate for θ obtained by excluding the jth data point. Then, in the case where $\hat{\theta}_1 = \hat{\theta}$ is an estimate for θ derived from the full sample, and $\hat{\theta}_2 = \overline{\hat{\theta}}_{(\cdot)} = \frac{1}{n}\sum_{j=1}^{n}\hat{\theta}_{(j)}$ is the average of the "leave one out" estimates for θ, then it is common to set $R(n) = \dfrac{n-1}{n}$. This gives the bias-reduced *jackknife estimator of* θ:

$$G_n\left(\hat{\theta}_1,\hat{\theta}_2\right)=J\left(\hat{\theta}\right)=n\hat{\theta}-(n-1)\bar{\hat{\theta}}_{(\cdot)}. \qquad (1.2)$$

This specification, which was first described in Quenouille (1956), can be shown to eliminate first order bias in the $1/n$ power series representation for θ (Gray & Schucany 1972, p.7; Efron 1982, p.5-6; Shao & Tu 1995, p.5). For instance, suppose we are estimating population variance from a sample $\{x_1,\ldots,x_n\}$, where the x_j are IID N(μ, σ^2), using the biased maximum likelihood estimate $\hat{\sigma}^2 = \frac{1}{n}\sum_{j=1}^{n}\left(x_j - \bar{x}\right)^2$ for σ^2. Then, evaluating (1.2) using this formula we obtain $J\left(\hat{\sigma}^2\right) = \frac{1}{n-1}\sum_{j=1}^{n}\left(x_j - \bar{x}\right)^2$, which is the unique minimum variance unbiased estimate (UMVUE) of σ^2.

Shao & Tu (1995), building from work found primarily in Wu (1986, 1990) and Shao & Wu (1989), generalize (1.2) to a resampling procedure involving data subsets (drawn without replacement) of size less than or equal to $n - 1$. Using the regression setup and notation from above, with S^C as the complement of index set $S \subset N = \{1,\ldots,n\}$, statistics of the form $\hat{\theta} = \hat{\theta}_{n-d,S} = \hat{\theta}_{n-d}\left([X\ \ Y]_j, j \in S^C\right)$ are repeatedly computed and used. Here we see that $\hat{\theta}$ is estimated using the $r = n - d$ observations that remain after observations indexed by set S are removed. The formula for the *delete-d jackknife variance estimator for* $\hat{\theta}$ (Shao & Tu 1995, p.50) is given by

$$J_d\left(\hat{\theta}\right) = \frac{r}{\binom{n}{d}} \sum_{\substack{S \subset N \\ |S|=d}} \left(\hat{\theta}_{r,S} - \frac{1}{\binom{n}{d}} \sum_{\substack{R \subset N \\ |R|=d}} \hat{\theta}_{r,R} \right)^2 . \qquad (1.3)$$

Setting $d = 1$, the *delete-1 jackknife variance estimator for* $\hat{\theta}$ is given by

$$J_1\left(\hat{\theta}\right) = \frac{n-1}{n} \sum_{j=1}^{n} \left(\hat{\theta}_{(j)} - \frac{1}{n} \sum_{k=1}^{n} \hat{\theta}_{(k)} \right)^2 . \qquad (1.4)$$

Note that we can write (1.2) as $J\left(\hat{\theta}\right) = \frac{1}{n} \sum_{j=1}^{n} \left(n\hat{\theta} - (n-1)\hat{\theta}_{(j)} \right) = \frac{1}{n} \sum_{j=1}^{n} \tilde{\theta}_j$, where the $\tilde{\theta}_j$ are referred to as the *jackknife pseudovalues*. Treating the $\tilde{\theta}_j$ as IID sample points (Shao & Tu 1995, pp.6-7), and assuming that $\tilde{\theta}_j$ has approximately the same variance as $\sqrt{n}\hat{\theta}$ (Shao & Tu 1995, p.6 and p.68), we can estimate the variance of $\hat{\theta}$ by $J_1\left(\hat{\theta}\right) = \frac{1}{n(n-1)} \sum_{j=1}^{n} \left(\tilde{\theta}_{(j)} - J\left(\hat{\theta}\right) \right)^2$. This equation can be rearranged to give (1.4).

$J_1\left(\hat{\theta}\right)$ does not always provide a consistent estimator[2]. A classic example of this situation involves using (1.4) to estimate the population median (Efron 1982, p.16). Generally speaking, the less smooth (i.e., continuously variable with n) the sample statistic, the larger the value for d is required for consistency of (1.3) (Shao & Wu 1989; Shao & Tu 1995, p.69).

[2] An estimator $\hat{\theta}_n$ for θ is consistent if, for all $\varepsilon > 0$, we have $\lim_{n \to \infty} P\left[\left| \hat{\theta}_n - \theta \right| < \varepsilon \right] = 1$.

The appearance of inconsistencies among delete-1 jackknife estimators provided some motivation for the development of the bootstrap, which alleviates some of these problems. In the context of linear statistics, the jackknife can even be considered as a linear approximation to the bootstrap (Efron & Tibshirani 1993, p.146). The bootstrap is itself a rather intuitive device that is easily implemented and broadly applicable to numerous situations in applied statistics. Its emergence onto the scene following the development of the jackknife is quite logical.

1.3.2. The Bootstrap

Development of the *bootstrap* is primarily attributed to statistician Bradley Efron (1979, 1982). Since that time, numerous papers and texts have emerged centered on this topic. For a general, comprehensive overview of the subject, see Efron & Tibshirani (1993). See Shao & Tu (1995) for an intense, measure-theortic examination of the bootstrap. Davison & Hinkley (1997) is a useful text that explores a number of characteristics and variants of the bootstrap largely from a practical perspective.

As previously mentioned, the bootstrap is premised on the *plug-in principle*, which says that statistics (or parameters) calculated using resamples from the empirical distribution function (EDF) defined by a sample provide estimates for sample statistics that might be obtained from the probability distribution function (PDF) of the population from which the sample was drawn. Assign a probability mass of $1/n$ to each observation $[X\ Y]_j$, which is a row of $[X\ Y]$. One then repeatedly

resamples from the EDF, estimating the statistic of interest at each so-called bootstrap sample (typically these consist of *n* points, like the original sample). After sufficiently many bootstrap samples have been evaluated, an approximated EDF for the desired statistic is obtained. Statistical properties (e.g., mean, variance, confidence intervals) associated with the statistic of interest are then estimated using the EDF emerging from the bootstrap exercise.

Let $\hat{Y}_{S^*} = X_{S^*}\left(X_{S^*}^{\mathrm{T}} X_{S^*}\right)^{-1} X_{S^*}^{\mathrm{T}} Y_{S^*}$, where S^* is a length-*n* index set drawn with replacement from *N*. To ensure unique computability of $\hat{Y}_{S^*}$, S^* must contain at least *p* distinct values. Ignoring this last comment, the *nonparametric bootstrap estimate for the variance of* ε is given by

$$BS\left(X,Y\right) = \frac{1}{2^n}\sum_{j=1}^{2^n}\left\|\hat{Y}_{S_j^*} - \frac{1}{2^n}\sum_{k=1}^{2^n}\hat{Y}_{S_k^*}\right\|^2 . \tag{1.5}$$

This equation is referred to as the *ideal bootstrap*. The S_j^* each refer to a particular (though not necessarily unique) bootstrap sample. Alternatively, (1.5) can be expressed as

$$BS\left(X,Y\right) = \frac{1}{\binom{2n-1}{n}}\sum_{j=1}^{\binom{2n-1}{n}}\omega_j\left\|\hat{Y}_{S_j^*} - \frac{1}{\binom{2n-1}{n}}\sum_{k=1}^{\binom{2n-1}{n}}\omega_k\hat{Y}_{S_k^*}\right\|^2 . \tag{1.6}$$

In this definition (which can be found in Efron & Tibshirani (1993, p.49)), the weight ω_j is the multinomial probability of occurrence of the jth distinct sample. The S_j^* now each refer to a particular unique bootstrap sample. It is easy to show that the number of possible unique bootstrap samples S^* of size n that can be drawn from N is $\binom{2n-1}{n}$ (Feller 1957, II.5).

To implement (1.6) so that all of the $\hat{Y}_{S^*}$ are well defined, one would have to exclude all bootstrap samples that contain fewer than p unique elements. Due to effects of model overfit in situations where S^* contains only p or slightly more than p unique elements, one might also consider discarding these bootstrap samples as well. See Shao & Tu (1995, p.291) for a simple rule designed to address bootstrap sampling concerns in the context of regression modeling. The exhaustive bootstrap of (1.6) is largely a theoretical construct that is rarely implemented in practice due to the intense computation required and diminishing returns observed as one increases the number of bootstrap samples considered. Questions regarding restrictions on S^* are typically not addressed in the literature beyond imposing some sort of representational balance of the data points among the S^* considered.

In applications, (1.6) is typically approximated by

$$BS^*\left(X,Y\right) = \frac{1}{b}\sum_{j=1}^{b}\left\|\hat{Y}_{S_j^*} - \frac{1}{b}\sum_{k=1}^{b}\hat{Y}_{S_k^*}\right\|^2, \qquad (1.7)$$

where b is some arbitrary number of index sets S^* typically numbering in the 10s-1000s. Also, (1.7) is commonly multiplied by $b/(b-1)$ to conform to the usual unbiased estimate of variance (Efron & Tibshirani 1993, p.47)).

Suppose that all the $\hat{\beta}_j^*$ (the bootstrap estimate for β using the bootstrap sample indexed by S_j^*) are calculated and retained. The set $\{\hat{\beta}_j^*\}$ forms an empirical distribution for $\hat{\beta}^*$, i.e., the plug-in estimate for the actual distribution of β, which can be used for confidence interval estimation for the model parameters. For example, after sorting the individual $(\hat{\beta}_j^*)_k$ (the bootstrap estimates for the kth parameter of β), one can locate the values at 5% and 95% (known as *bootstrap percentiles*; Efron & Tibshirani 1993, p.168) to set the endpoints for a 90% confidence interval for $(\hat{\beta}^*)_k$, which is presumed to reflect the same information regarding $(\beta)_k$ based on the plug-in principle.

So far the discussion has centered on the bootstrap approach known as *bootstrapping pairs*, as in data pairs $[X\ Y]_j$, where $[X\ Y]_j$ is the jth row of the system $[X\ Y]$. There is another procedure referred to as *bootstrapping residuals* that uses the values of e to define an empirical distribution, drawn from (with replacement) to form e^*. The e^* are added to the original $X\hat{\beta}$ to form "new" observations of the dependent variable: $Y^* = X\hat{\beta} + e^*$. The system $[X\ Y^*]$ is then treated as data and modeled as before, relevant model statistics are retained, and the procedure is iterated until a sufficiently resolved EDF appears for the quantities of interest. Sometimes the

residuals are "standardized" beforehand to mitigate effects of bias in *e* as well as potential heteroscedasticity of the errors (Wu 1986; Shao 1996; Miller 2002, p.152).

The definition of $\hat{Y}_{S^*}$ given above reveals that the bootstrap estimate for error variance given in (1.7) is an in-sample statistic, and thus has a downward bias (called *expected excess error*) as an estimate for σ^2. Also, just like with MSE and REG, higher dimensional models automatically will be favored if one uses (1.7) by itself for model ranking. Fortunately the situation can be addressed, as the expected excess error can be estimated and partially accounted for via a simple adjustment to (1.7). Let $\hat{\beta}^*$ denote the bootstrap estimate for β. Then compute

$$\mathrm{E}[(Y^* - X^*\hat{\beta}^*)^{\mathrm{T}}(Y^* - X^*\hat{\beta}^*) - (Y - X\hat{\beta}^*)^{\mathrm{T}}(Y - X\hat{\beta}^*)]/n. \qquad (1.8)$$

This expression provides an estimate for the bias of the bootstrap estimate of error variance. In (1.8), the bootstrap coefficients are being applied to both the original data and the bootstrap sample, and the difference between MSE values calculated from these two model applications provides an "observation" of expected excess error. After processing all of the bootstrap samples, (1.8) can be estimated. To reduce the bias of the bootstrap from (1.7), one then subtracts (1.8) from (1.7) (Efron & Tibshirani 1993, p.132; Shao & Tu 1995, p.304; Davison & Hinkley 1997, p.296). Due to negative effects of potential imbalance of the data observations among the bootstrap samples, Efron & Tibshirani (1993, p.132) present a modification (referred to as the *better bootstrap bias estimate*) to (1.8) to generally help stabilize the

outcome without compromising accuracy. In Shao (1996), the author uses the second term of equation (1.8) (namely, $E[(Y - X\hat{\beta}^*)^T(Y - X\hat{\beta}^*)]/n$) to rank competing models (for more on this result, look ahead to footnote 6).

It is remarkable how many variations on the bootstrap theme exist in the literature, which is a testament to its flexibility and general usefulness as a tool in applied statistics. However, when it comes to matters of model selection and small sample predictive modeling, it is still somewhat lacking in appeal due to its "in-sample" features: the data used to estimate model parameters are also used to assess the accuracy of the estimated model. Thus bootstrap estimates are subject to *selection bias*, a concept discussed thoroughly in Miller (2002). There have been attempts to remedy this situation. In particular, there is "the .632 estimator" of Efron (1983) and the later modification that is "the .632+ estimator" introduced in Efron & Tibshirani (1997). However, these methods stray heavily from the standard bootstrap that is presented here (merely borrowing the bootstrap resampling strategy), and actually incorporate aspects of delete-1 cross validation to achieve the goal of better estimating the error rate of prediction. On that note, we now turn our attention to cross validation.

1.3.3. Delete-d Cross Validation

The method of *cross validation* is a systemization of the more general concept known as *data splitting*. Data splitting entails splitting a data set into two parts, one to be used for model parameter estimation (learning) and the other to be used for

model performance evaluation (testing).[3] The sizes of the two data subsets depend on sample size as well as model complexity, but no precise theoretical guidelines are available to assist the practitioner in making the decision of how to split the data.

Cross validation sprang from the statistical philosophy of *predictivism*, the fundamental tenet of which purports that the primary assessment of a model should be based on the model's predictive capabilities. Model parameters, which serve as the primary analytical focus in many studies, take a back seat to model performance.

Three papers provided some of the early groundwork for cross validation. Allen (1974) introduced the prediction sum of squares (PRESS) statistic, which involves sequential prediction of single observations using models estimated from the full data absent the data point to be predicted. This method is frequently referred to as "hold one out", or "delete-1" cross validation. Stone (1974) examined the use of delete-1 cross validation (CV(1)) methods for regression coefficient "shrinker" estimation. Geisser (1975) presented one of the first introductions of a multiple observation holdout sample reuse method similar to delete-*d* cross validation (CV(*d*)), which is a generalization of CV(1).[4] One of the first major practical implementations of CV appeared in Breiman *et al*. (1984), where "V-fold cross validation" is offered as a way to assess accuracy during optimization of classification and regression tree

[3] Some in the artificial neural network community (and elsewhere where iterative model optimization is required) have taken the notion of data splitting one step further by partitioning the data into three subsets—one used for model estimation, one used for halting the optimization routine, and one used for post-optimization model evaluation.

[4] Geisser's "radical" suggestion to consider holding out more than one sample at a time was met with skepticism even among those in his own camp. Stone, in his rejoinder to the discussion following Stone (1974), dismisses Geisser's approach: "*For my swan-song of independent thought, however, I conjecture that Geisser is on the wrong track in leaving out more than one item at a time, that whatever diluted optimality theorems exist in this area will require the n – 1 split.*"

models. Generally applicable only when sample size is large, this involves partitioning the dataset into V subsets of nearly equal size, and then sequentially treating each subset as a holdout set in a CV exercise.

Define $\hat{Y}_S = X_S \hat{\beta}\left(S^C\right) = X_S \left(X_{S^C}^{\mathrm{T}} X_{S^C}\right)^{-1} X_{S^C}^{\mathrm{T}} Y_{S^C}$, where index set S is drawn from $N = \{1,\ldots,n\}$ without replacement, and $S^C = N \setminus S$. Note that in this definition, observations of Y indexed by S are being "predicted" with a model constructed using only observations indexed by S^C. Define the *delete-d cross validation estimate for the variance of* ε to be the quantity

$$CV(d) = \frac{1}{\binom{n}{d}} \sum_{\substack{S \subset N \\ |S|=d}} \left(Y_S - \hat{Y}_S\right)^{\mathrm{T}} \left(Y_S - \hat{Y}_S\right) = \frac{1}{\binom{n}{d}} \sum_{\substack{S \subset N \\ |S|=d}} \left\| Y_S - \hat{Y}_S \right\|^2 . \qquad (1.9)$$

This formula can be found in Zhang (1993), McQuarrie & Tsai (1998, p.255), and Seber & Lee (2003, p.405). With $\hat{\beta} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$, and applying the Sherman-Morrison-Woodbury formula to obtain the relationship $Y_S - \hat{Y}_S = \left(I - X_S \left(X^{\mathrm{T}}X\right)^{-1} X_S^{\mathrm{T}}\right)^{-1} \left(Y_S - X_S\hat{\beta}\right)$, we can write (1.9) as

$$CV(d) = \frac{1}{\binom{n}{d}} \sum_{\substack{S \subset N \\ |S|=d}} \left\| \left(I - X_S \left(X^{\mathrm{T}}X\right)^{-1} X_S^{\mathrm{T}}\right)^{-1} \left(Y_S - X_S\hat{\beta}\right) \right\|^2 . \qquad (1.10)$$

Note that (1.10) does not require the calculation of $\hat{Y}_S$, and overall is much less computationally expensive than (1.9) (Zhang 1993).

It is instructive to demonstrate a simple application of CV. Let us evaluate the expected value of the CV(1) estimate for sample variance, which is the same as estimating σ^2 using the mean model (X = column of ones, so that $\hat{\beta} = \overline{Y}$) in the OLS regression context.

For $j = 1,\ldots,n$, let $y_j \sim (\mu, \sigma^2)$ be independent observations of some random variable. Define the "hold one out" sample mean $\overline{y}_j = \frac{1}{n-1}\sum_{i \neq j} y_i$. Using the formula for variance about the mean, we have $\overline{y}_j \sim (\mu, \sigma^2/(n-1))$. Since y_j and $\overline{y}_j$ are independent random variables, we have CV(1) residuals $e_j = y_j - \overline{y}_j \sim (0, \mathrm{Var}(y_j) + \mathrm{Var}(\overline{y}_j))$, which implies that

$$\mathrm{E}[\mathrm{CV}(1)] = \mathrm{Var}(e_j) = \sigma^2 \frac{n}{n-1}.$$

Not surprisingly, we find the CV(1) estimate for sample variance to be upwardly biased due to its 'out-of-sample' character. Due to the statistical complexity of (1.9), however, no one has yet derived a general formula for E[CV(d)] like the one above.

Some asymptotic properties for CV(d) have been established (Zhang 1993; Shao 1993 and 1997). Besides these studies, theoretical results have been largely

confined to the case $d = 1$, which is more mathematically tractable than cases for which $d > 1$. Numerous authors have discussed and examined the properties of CV(1) specifically in the context of model selection (e.g., Hjorth 1994; McQuarrie & Tsai 1998; Miller 2002). These studies and others have shown that in spite of the merits of using CV(1), this method does not always give rise to good estimates of prediction error, nor does it always perform well in optimal model identification simulation studies when compared to other direct methods such as information criteria[5] (e.g., McQuarrie & Tsai 1998). The consensus is that CV(1) has a tendency in many situations toward selection of overly complex models, i.e., it does not sufficiently penalize for overfitting.

The inconsistency of CV(1) in a general model selection scenario has been shown in Shao (1993). In this work, a pool of candidate predictors (columns of $X \in \mathbb{R}^{nxp}$, $p \leq n$, p fixed) is given, along with the quantity to be predicted (Y). Some of the predictors in X may not be related to Y, and these predictors would be expected to have 0 coefficients in β when included in a linear model. Define the optimal model to be the column subset of X containing only the predictors with non-zero coefficients in β, and rank the different models using CV(d). Under some fairly weak asymptotic assumptions, the requirements that $d/n \rightarrow 1$ and $n - d \rightarrow \infty$ as $n \rightarrow \infty$ are shown to be necessary and sufficient for the asymptotic consistency of using CV(d) for optimal

[5] "Information criteria" generally consist of a log-likelihood function and complexity penalty parameter.

model selection in this situation[6], assuming that the optimal model is not the full model (the one constructed using all of the predictors). The second condition ($n - d \rightarrow \infty$) is needed to ensure that a correct model (i.e., one containing all of the predictors with non-zero coefficients) is selected, while the first condition ($d/n \rightarrow 1$) is needed so that the model selected is of minimal size. The conclusion to be drawn from Shao (1993) is that for model ranking in model selection, a value for d that is an appreciable fraction of sample size n is preferred. However, no specific guidance is provided, as the finite sample situation is inconsequential to the asymptotic result. For example, setting $d = \text{ceil}[n - n^{\alpha}]$, $0 < \alpha < 1$, satisfies Shao's two conditions, yet imposes no certain constraint on what values for d are desirable.

Fixing $d = d_0$ but letting n grow, we witness the basis of much of the asymptotic theory, namely that CV(d_0) becomes numerically indistinguishable from CV(1), which eventually becomes indistinguishable from REG and MSE as n increases and the so-called "error curve" defined by CV(d) gets flatter near d_0. As discussed in Shao (1993, 1996), asymptotic equivalence of CV(1) to the delete-1 jackknife, the standard bootstrap, and other model selection methods such as Mallows's C_p (Mallows 1973) and the Akaike information criteria (Akaike 1973), has also been established. By allowing d to increase at a rate $d/n \rightarrow a < 1$, Zhang (1993) shows that CV(d) and a particular form of the mean squared prediction error (Shibata

[6] Shao later proved a similar consistency result for the bootstrap in the same model selection context (Shao 1996), after demonstrating the inconsistency of using $E[(Y - X\hat{\beta}^*)^T(Y - X\hat{\beta}^*)]$ for optimal model selection with bootstrap samples of size n. The author showed that if bootstrap sample size m was selected so that $m \rightarrow \infty$ and $m/n \rightarrow 0$, these conditions were necessary and sufficient to ensure asymptotic consistency for optimal model selection.

1984; this is a generalization of the "final prediction error" of Akaike 1970) are asymptotically equivalent under certain constraints.

Shao (1997) summarizes most of the above findings by developing a general framework from which to view the situation, resulting in three classes of methods characterized by asymptotic behavior. With respect to CV(d), Shao (1997) shows (*i*) that the condition $d/n \to 0$ is useful in situations where there do not exist fixed dimension correct models; (*ii*) that the condition $d/n \to 1$ is useful in situations where there do exist fixed dimension correct models; and (*iii*) that the condition $d/n \to a \in (0,1)$ is a compromise between the other two conditions, but that its asymptotic performance is not as good as the other conditions in their respectively appropriate situations.

1.4. Conclusion

We have introduced three resampling methods, primarily in the context of OLS linear regression modeling. The jackknife was developed as a tool for bias reduction and variance estimation. The bootstrap, characterized by a more general resampling strategy than the jackknife, presents an improvement over the jackknife, as it can accommodate more situations than the jackknife. Cross validation provides an alternative resampling method premised on out-of-sample prediction.

Attention was given to the presented resampling methods due to their robustness and subsequent applicability in model selection as well as in the general small sample statistical modeling situation. Ultimately, model characterization

involves balancing model generality and model specificity (or simplicity and complexity, or parsimony and goodness-of-fit, or variability and bias, etc.). Asymptotic theory indicates that when using cross validation or the bootstrap for model ranking, the smaller the data subsets that are used for parameter estimation during resampling, then the former properties are emphasized (generality, simplicity, parsimony, less variability). On the other hand, if models are based on larger subsets of the data during resampling, then results will generally favor the latter properties (specificity, complexity, goodness-of-fit, less bias).

Page left intentionally blank.

Chapter 2. Testing Properties of Cross Validation with Simulation

Chapter Summary

Despite numerous empirical studies and theoretical developments exploring properties of delete-*d* cross validation, the small sample behavior of the CV(*d*) statistic is largely undocumented. Using simulation, estimates for the unknown quantity E[CV(*d*)] are examined. Simulation results suggest general formulas for E[CV(*d*)] involving rational scalar multiples of σ^2, with the scalar values increasing with model complexity. Also, a two-point instability in the E[CV(*d*)] error curve is observed in all examined situations involving a model that includes at least one random-valued predictor. This phenomenon, which is compatible with the inferred formulas for E[CV(*d*)], introduces two points of increasing instability at the two largest possible *d* values. Finally, results from the E[CV(*d*)] simulation are connected back to theory.

"All possible subsets" and "fixed dimension" model selection simulations are then used to demonstrate that one important asymptotic model selection result involving linear regression and CV(*d*) is influential already in the smallest sample setting. During these simulations, a few other statistics useful for model selection are also examined to provide a gauge for CV(*d*) model selection rates. Considering the results obtained from these model selection simulations, a "divide and conquer" model selection strategy is proposed for general "all possible subsets" model selection problems.

2.1. A Difficult Modeling Problem

Suppose a practitioner is faced with the following modeling problem setup:

- Small sample of independent observations ($n < 20$)
- Each observation consists of values from some fixed, large number ($O(10^2)$) of candidate predictors and a single response value
- High correlation between most candidate predictor pairs
- No obvious way to eliminate candidate predictors from consideration

As a means for expressing relationships between predictors and response, the practitioner will use the following standard linear statistical model forms:

F1: $\hat{Y}_1 = \hat{\beta}_{01} + \hat{\beta}_{11} X_{11}$

F2: $\hat{Y}_2 = \hat{\beta}_{02} + \hat{\beta}_{12} X_{12} + \hat{\beta}_{22} X_{22}$

The immediate task for this practitioner is to identify some small number (tens to hundreds) of 1- and 2-predictor subsets from the candidate predictor set that will produce the "best" models of the forms F1 and F2 above, using ordinary least squares (OLS) regression to estimate parameter values. For this application, "best" models are those that have the smallest expected mean-squared prediction error when used to make a single response prediction at some unknown future observation of the predictor values. Once this future observation of the predictor values is realized, these best models are individually evaluated at the future observation to create a set of

response predictions. Next, the practitioner computes the average prediction from this set of response predictions.[1] This average prediction is then released to the public well in advance of measurement of the actual response.

Now suppose the practitioner must independently repeat this model selection and implementation exercise 11,000 times per year, using a different data set for each repetition. This describes the modeling problem that the author has faced annually since 2002 as sole administrator of the nationwide crop yield forecasting program at the Kansas Applied Remote Sensing (KARS) Program. For the KARS crop yield forecasting program, response values are annual, final estimated harvested crop yield values generated and distributed by the United States Department of Agriculture (USDA). Candidate predictors (available going back to 1989) are derived from biweekly time-series satellite data collected prior to and during each crop's respective growing season. KARS issues crop yield predictions at multiple times during the year, for multiple crops and multiple spatial scales, resulting in approximately 11,000 unique crop yield forecasts per year.

More than 130,000 models of form F2 must be examined during final season predictions for each (crop, region)-pairing, which is indicative of the author's need for an automated, judicious method of model ranking to identify best models. The author must be assured that only in a marginal number of instances might models be selected that will produce illogical predictions (unbelievably small or large crop

[1] The practitioner is using a "combined forecasting" approach, whereby forecasts from multiple best models are averaged to create a single forecast. This technique is used to help reduce the error variance of the released forecast.

yield). Additionally, the author must estimate in advance a general "expected error" for each eventual state-level forecast that will be made, to provide reasonable "+/- one standard deviation" confidence intervals for these forthcoming predictions. Since program inception, the author has used delete-*d* cross validation for both model selection and estimation of expected prediction error.

Cross validation (CV) is a data resampling method that uses *data splitting*. To describe data splitting, suppose that each data observation consists of a response value (the dependent variable) and its corresponding predictor values (the independent variables) that will be used in some specified model for the response. The data observations are split into two subsets. One subset (the *training set*) is used for model parameter estimation. The complementary subset (the *testing set*) is then used to compute an "out-of-sample" model accuracy statistic, typically mean squared error. For *delete-d cross validation*, all possible data splits with testing sets that contain d observations are evaluated. The statistic that results from this computational effort is denoted CV(d).

For example, suppose the sample size is $n = 10$, and we want to evaluate CV(d) values for a particular two-parameter model using two pre-specified predictors. Then $d_{max} = n - p = 8$ is the largest d value for which CV(d) can be computed, because one needs at least two observations in each training set to estimate the two model parameters. For testing set size $d = 1$, there are 10-choose-1 = 10 possible unique data splits that must be evaluated to compute CV(1). There are 10-choose-2 = 45 possible splits when $d = 2$, 10-choose-3 = 120 possible splits when $d =$

3, and so on. Consider the case with $d = 3$. For each split, the two model parameters are estimated using the seven observations in the training set. The model is then used to generate an "out-of-sample" prediction for each of the three observations in the testing set. These predictions are differenced from their respective observed values, and the squared error is computed and retained from each prediction. After performing the necessary computations for each of the 120 unique splits, this results in 3*120 = 360 out-of-sample predictions, with equal representation for each data observation (360/10 = 36 out-of-sample predictions per observation).

The "out-of-sample" aspect of CV(d) is relevant to any modeling problem where overfitting is a concern, such as small sample problems and other problems with low degrees of freedom. *Overfitting* refers to the phenomenon by which model parameter values are influenced by chance (i.e., non-systematic) covariation between predictor values and response values, typically associated with system noise or other sources of non-pertinent variation in the data. The effect of overfitting is an illusory reduction of the estimated error variance of the model while simultaneously reducing the model's general utility for prediction. This effect can produce a bias toward selection of higher dimensioned models, which are generally more susceptible to overfitting. This is not meant to imply that CV(d) completely overcomes overfitting, but it seems to mitigate this problem. For example, see Kastens *et al.* (2005), where even CV(1) demonstrates an ability to identify two dimensional models as generally preferable to three dimensional models in a rigorous crop yield forecasting exercise with sample size $n = 11$. Other research (e.g., Zhang 1993, Shao 1993) indicates that

using $d > 1$ (but not too large) might be generally preferable to $d = 1$. Consequently, the user's choice for which d to use constitutes an important question.

Turning attention back to the KARS crop yield forecasting program, the reader might ask, "Why must the author go through all this trouble?" First, KARS had a strong need for a fully generalized, purely automated statistical forecasting procedure, so that the program could be confidently and efficiently maintained (and possibly expanded) into the future without requiring a great deal of administrator oversight. Second, in order to build a reliable forecasting "track record" to establish program credibility, it was important that objective, repeatable methods were used that could be applied consistently year after year. Indeed, the program has been successful. Without going into detail, the six-year (2002-2007) accuracy of KARS predictions released almost one month in advance of comparable USDA forecasts is on par with the accuracy of those later-released USDA forecasts (e.g., see Watts *et al.* 2005 for some three-year accuracy statistics).

In addition to the above constraints, important details regarding the use of satellite imagery for crop yield forecasting have not been described, and these details affected the choice of statistical methods on which the program could be based. Elaboration on the full breadth of the program is beyond the scope of this thesis, since the program is being used merely as an example illustrating the author's motivation for studying CV(d). For more information on the use of satellite data for crop yield forecasting, the reader is referred to Kastens *et al.* (2005).

As noted above, an important question facing the author (as well as other practitioners using CV(d) for either prediction error estimation or model selection) regards which choice of d should be used. Due to the evident success of the CV(d) methodology used in the KARS crop yield forecasting program, initially the author's intent was to use the large attendant databases to investigate this question in a small sample setting. If one could identify some statistical tendencies of CV(d) using real data (preferred d for model selection or prediction error estimation), such results could provide generally useful heuristic recommendations. However, the many complexities of the data and the yield modeling problem precluded this possibility in any convenient, meaningful fashion. Rather, computer simulation was used instead to investigate small-sample tendencies of the statistic in idealized settings. The principal question to be addressed is the effect of different choices of d on prediction error estimation and model selection.

2.2. Previous Research

Three popular papers provided some of the early groundwork for cross validation. Allen (1974) introduced the prediction sum of squares (PRESS) statistic, which involves sequential prediction of single observations using models estimated from the full data absent the data point to be predicted. Stone (1974) examined the use of delete-1 cross validation methods for regression coefficient "shrinker" estimation. Geisser (1975) presented one of the first introductions of a multiple observation holdout sample reuse method similar to delete-d cross validation. One of

the first major practical implementations of CV appeared in Breiman *et al*. (1984), where "V-fold cross validation" is offered as a way to internally estimate model accuracy during optimization of classification and regression tree models. Generally applicable only when sample size is large, this involves partitioning the dataset into V subsets of nearly equal size, and then sequentially treating each subset as a holdout set in a CV computation.

Numerous authors have discussed and examined the properties of CV(1) specifically in the context of model selection (e.g., Hjorth 1994; McQuarrie & Tsai 1998; Miller 2002). These studies and others have established that, in spite of the merits of using CV(1), this method does not always perform well in optimal model identification studies when compared to other direct methods such as information criteria (e.g., McQuarrie & Tsai 1998). The consensus is that CV(1) has a tendency in many situations to select overly complex models; i.e., it does not sufficiently penalize for overfitting (Davison & Hinkley 1997, p. 303).

Asymptotic equivalence of CV(1) to the delete-1 jackknife, the standard bootstrap, and other model selection methods such as Mallows's C_p (Mallows 1973) and the Akaike information criteria (AIC; Akaike 1973), has been established (see Shao 1993, 1997 and references therein). By allowing d to increase at a rate $d/n \rightarrow a < 1$, Zhang (1993) shows that CV(d) and a particular form of the mean squared prediction error (Shibata 1984; this is a generalization of the "final prediction error" of Akaike 1970) are asymptotically equivalent under certain constraints.

The inconsistency of CV(1) in a general model selection scenario has been shown in Shao (1993). Many researchers have explored CV(d) for one or more d values for actual and simulated case studies involving model selection (e.g., Zhang 1993; Shao 1993; McQuarrie & Tsai 1998), but not to the extent of exposing any general, finite-sample statistical tendencies of CV(d) as a function of d.

2.3. Delete-d Cross Validation in Ordinary Least Squares Regression

To define the CV(d) statistic used in OLS linear regression settings, let $p < n$ be positive integers and let I_k denote the k-by-k identity matrix. Let $X \in \mathbb{R}^{n\mathrm{x}p}$, $Y \in \mathbb{R}^{n\mathrm{x}1}$, $\beta \in \mathbb{R}^{p\mathrm{x}1}$, and $\varepsilon \in \mathbb{R}^{n\mathrm{x}1}$, where β and ε are unknown and $\varepsilon \sim (0, \sigma^2\mathrm{I}_n)$. Also, let β and ε be such that $Y = X\beta + \varepsilon$ is the "true" linear statistical model relating response values in Y to predictor values in X. As usual in these problems, each row of the matrix [X Y] corresponds to a data observation (p predictors and one response), and each column of X corresponds to a particular predictor. Assume that each p-row submatrix of X has full rank, a necessary condition for CV(d) to be computable for all $d = 1,\ldots,n-p$. This is a reasonable assumption when data observations are presumed to be independently sampled. Let S be an arbitrary subset of $N = \{1,2,\ldots,n\}$, and let $S^C = N \setminus S$. Let X_S denote the row subset of X indexed by S, and define $\hat{\beta}(S)$ to be the OLS parameter vector estimated using X_S (just $\hat{\beta}$ if $X_S = X$). Define

$\hat{Y}_S = X_S \hat{\beta}\left(S^C\right) = X_S \left(X_{S^C}^{\mathrm{T}} X_{S^C}\right)^{-1} X_{S^C}^{\mathrm{T}} Y_{S^C}$. Let $\left|\left|\cdot\right|\right|$ denote the l^2 norm and let $\left|\cdot\right|$ denote set cardinality. Then the *delete-d cross validation statistic* is given by

$$\mathrm{CV}\left(d\right) = \binom{n}{d}^{-1} d^{-1} \sum_{|S|=d} \left\| Y_S - \hat{Y}_S \right\|^2 . \qquad (2.1)$$

This equation can be found in Zhang (1993), McQuarrie & Tsai (1998, p. 255), and Seber & Lee (2003, p. 405). As can be seen from (2.1), the form of CV(d) is that of a "mean squared error", which is common for error variance estimators.

The distinguishing feature of CV(d) is that all model predictions (entries of $\hat{Y}_S$) used in the formula are technically generated "out-of-sample". Indeed, this is the primary appeal of CV(d) for practitioners, that with this attribute it will provide a more believable general estimate for expected prediction error of the model (whose final parameters are estimated using all of the observations) than traditional "in-sample" error variance estimators. The best known such "in-sample" estimator is the *expected error of regression*, which is given by

$$\mathrm{REG} = \left|\left| Y - X\hat{\beta} \right|\right|^2 / (n - p) \qquad (2.2)$$

and has the property that E[REG] = σ^2.

The motive behind using CV(d) for model ranking is now apparent: If an estimator provides a useful estimate for expected prediction error, then it makes sense to rank models according to this statistic, giving preference to models with smaller CV(d) values for a particular user-determined d.

As previously noted, the principal question when using CV(d) for either prediction error estimation or model selection regards which value for d should be used. From a practical perspective, existing theoretical developments (all of which pertain to asymptotic behavior as $n \rightarrow \infty$) are at best marginally useful in resolving this question for small sample settings. Here is the dilemma: Use of small d values in CV(d) can lead to overly optimistic prediction error assessment and bias toward selection of higher-dimensioned models; use of large d values can lead to overly pessimistic prediction error assessment and bias toward selection of lower-dimensioned models.

The hope is that for a given modeling problem, there is some "optimal d" to use for generally even-handed prediction error assessment, or which most fairly compares models of various dimension. However, there is no guarantee that such a d value even exists. Furthermore, if it does exist, it is not necessary that "optimal d" will be the same for the related but distinct tasks of prediction error estimation and model ranking. Finally, even if a single "optimal d" exists applicable for both prediction error estimation and model selection, this value could well exhibit substantial dependence on the distributional structure of the data, enough so that making any general recommendations for "which d should be used" is not possible.

The previous paragraph raises big questions regarding the use of CV(d) in applied statistics, questions for which we may never obtain satisfactory answers. Certainly these matters will not be resolved in this thesis. The fact is that we know almost nothing about the small sample use of the statistic, not even its expected value. With scant direction on how one should begin to study this situation, it is a bit like the proverbial eating of the elephant: "How do you eat an elephant? One bite at a time." In the following sections, the author attempts to take a few bites using simulation studies.

The objective of the first simulation is to estimate values for E[CV(d)] in an attempt to expose general expressions for this statistic, in terms of n, p, d, and σ^2. Theorists have so far been unsuccessful in identifying any such equation. Results from this simulation have indirect relevance for the prediction error estimation problem. Generally speaking, it is reasonable to assume that prediction error is a dilation of σ^2, which characterizes the identified forms for E[CV(d)] uncovered by the author. The resulting equations are then linked back to theory.

In the second simulation, optimal model selection rates using CV(d) for model ranking are simulated in a contrived, small sample setting. The objective is to determine, for different sample sizes, which d values produce CV(d) that are most successful at optimal model identification. In particular, the author wishes to determine if small sample results exhibit behavior reflective of the most interesting asymptotic model selection result from the literature.

2.4. Simulating the Expected Value for CV(d)

2.4.1. Problem Background

As previously noted, there is no general formula for E[CV(d)] as a function of n, p, d, and σ^2. In fact, only one attempt has been made at hypothesizing an approximate equation of this nature (Shao & Tu 1995, p.309). Here the authors suggest that

$$\mathrm{E}\left[\mathrm{CV}(d)\right] \approx \mathrm{E}\left[\hat{\sigma}^2_{n-d}\right] = \sigma^2\left(1+\frac{p}{n-d}\right). \tag{2.3}$$

This expression implies that CV(d) provides an estimate for $\hat{\sigma}^2_{n-d}$, where $\hat{\sigma}^2_{n-d}$ refers to the squared prediction error when making a prediction for a future observation *at a design point* (row of predictor matrix X) and X contains $n - d$ independent observations (rows).

To explain the equality in (2.3), let x_f be some new observation of predictor values, where x_f is identical to some row of predictor matrix X. Suppose $Y \sim \mathrm{N}(X\beta, \sigma^2\mathrm{I}_n)$, and let y_f be the future observation of the dependent variable at x_f. Then we have

$$\mathrm{E}\left[\left\|y_\mathrm{f} - x_\mathrm{f}\hat{\beta}\right\|^2\right] = \sigma^2 + \sigma^2 x_\mathrm{f}\left(X^\mathrm{T}X\right)^{-1}x_\mathrm{f}^\mathrm{T} \tag{2.4}$$

The source for the second term on the RHS of (2.4) is sample bias attributable to x_f as a single observation from sample X. If we compute the average of (2.4) over all rows in X, then we obtain the formula on the RHS of (2.3) (Shao & Tu 1995, p307).

Simulation results described in this section indicate that the approximation for E[CV(d)] in (2.3) is exact for the (somewhat) trivial mean model. Further simulation results reveal that this approximation is incorrect when the model contains a random valued predictor, and becomes worse as an increasing number of random valued predictors are included in the model.

Those studying asymptotic properties for CV(d) generally fail to distinguish cases in which one of the predictors is an *intercept*, which characterizes the vast majority of applied linear models. The intercept is typically represented in X as a column of ones, and its presence allows a model to not have to pass through the origin (i.e., $\hat{Y}$ does not have to be 0 when all of the independent variable values are 0). Perhaps this neglect is permissible in asymptotic studies where p is free to grow unbounded along with n (such as in Shao 1993), if one assumes that the distinct effects of using an intercept generally become negligible as p increases. Theoretical developments applicable for small samples cannot be afforded this luxury, and two cases must be considered—those that include an intercept as a predictor, and those that do not. Simulation results for E[CV(d)] clearly indicate the necessity for this dichotomy.

2.4.2. Results

Using the normal random number generator in MATLAB®, values for CV(d) were simulated for numerous cases with $n \in \{4,\ldots,20\}$, $p \in \{1,\ldots,n\text{-}2\}$, error $\varepsilon_j \sim_{\text{IID}} N(0, \sigma^2)$ and $X_j \sim_{\text{IID}} N(0, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2,\ldots, \sigma_p^2)$ and $j \in \{1,\ldots,n\}$. All σ and σ_k were independently and randomly sampled from the truncated distribution $10^{N(0,1)} \cap [10^{-3},10^3]$, with $10^{N(0,1)}$ sample values outside the interval $[10^{-3},10^3]$ snapped to the appropriate interval endpoint. Separate cases were considered using models with and without an intercept as a predictor. For a particular (n,p), after simulating approximately 20,000-100,000 values for CV(d) for all possible d, average simulated CV(d) values were computed to provide simulated E[CV(d)]. Upon inspection, simulated E[CV(d)] were found to follow rational number pattern sequences clear enough to conjecture general formulas for E[CV(d)] dependent on n, p, d, and σ^2.

An apparently related outcome is the identification of a two-point region of instability of the E[CV(d)] error curve for any tested model that includes a random valued predictor. Specifically, simulation results reveal two points of increasing instability at E[CV($d_{\max} - 1$)] and E[CV($d_{\max}$)], where $d_{\max} = n - p$. This distinct result was unexpected, nowhere anticipated in the CV(d) literature. The term "increasing instability" is apt because the coefficient of variation (= standard deviation / mean) calculated for the simulated CV(d) values is stable for $d < d_{\max} - 1$, but increasingly blows up (along with E[CV(d)]) at $d = d_{\max} - 1$ and $d = d_{\max}$. The reason for this phenomenon is, at present, an interesting open question. These

simulation results are consistent with the mathematical breakdown of the E[CV(d)] formulas given in Conjectures 2.1 and 2.2 at the two largest d values.

To gauge the accuracy of the conjectured formulas for E[CV(d)], the author used an absolute percent error statistic defined by

$$\text{APE} = 100 * \frac{\left|\text{simulated E[CV}(d)] - \text{predicted E[CV}(d)]\right|}{\text{simulated E[CV}(d)]} \qquad (2.5)$$

Begin with the simplest case, where the only predictor is the intercept. Suppose $X = \mathbf{1}^{n\text{x}1}$ (an n-vector of ones), so that the linear regression model under investigation is the mean model (so called because $\hat{Y} = \hat{\beta} = \overline{Y}$). Then the function underlying the mean squared model error is $(1/n)\left|\left|Y - \hat{Y}\right|\right|^2 \approx \text{E}[(y - \text{E}[y])^2] = \text{Var}[y]$, where y is a random variable with the response distribution. The true expected value for (2.1) under the mean model is given in

THEOREM 2.1: Suppose $X = \mathbf{1}^{n\text{x}1}$, and let $Y = [y_j] \in \mathbb{R}^{n\text{x}1}$ be such that the $y_j \sim_{\text{IID}} (\mu, \sigma^2)$. Then, for $d = 1, \ldots, d_{\max}$, the expected value for CV(d) is given by

$$\text{E}\left[\text{CV}(d)\right] = \sigma^2\left(1 + \frac{1}{n-d}\right). \qquad (2.6)$$

PROOF: Define $\hat{Y}_S = X_S \hat{\beta}\left(S^C\right)$, where $|S| = d$. Then, for $d = 1,\ldots, d_{\max}$, we will show that the expected value for a single summand term of (2.1) is given by

$$\mathrm{E}\left[\tfrac{1}{d}\left\|Y_S - \hat{Y}_S\right\|^2\right] = \sigma^2\left(1 + \frac{1}{n-d}\right).$$

The d-by-1 vector $Y_S - \hat{Y}_S$ has components of the form $y_j - \overline{Y}_{S^C}$, where y_j is a "deleted" observation (entry in Y_S) and $\overline{Y}_{S^C}$ is the sample mean of $(n - d)$ Y-values in Y_{S^C} that were not deleted. Since y_j and $\overline{Y}_{S^C}$ are statistically independent and have the same expected value (μ), we have

$$\mathrm{E}\left[\left(y_j - \hat{Y}_S\right)^2\right] = \mathrm{E}\left[\left(y_j - \overline{Y}_{S^C}\right)^2\right] = \mathrm{Var}\left[y_j - \overline{Y}_{S^C}\right] = \mathrm{Var}\left[y_j\right] + \mathrm{Var}\left[\overline{Y}_{S^C}\right] = \sigma^2 + \frac{\sigma^2}{n-d}.$$

Since y_j is an arbitrary element of hold-out set Y_S, we have

$$\mathrm{E}\left[\left\|Y_S - \hat{Y}_S\right\|^2\right] = \mathrm{E}\left[\sum_{j\in S}\left(y_j - \overline{Y}_{S^C}\right)^2\right] = \sum_{j\in S}\mathrm{E}\left[\left(y_j - \overline{Y}_{S^C}\right)^2\right] = d\sigma^2\left(1 + \tfrac{1}{n-d}\right).$$

Because of the linearity of E[·], (2.6) immediately follows from this derivation, which applies to an arbitrary split of the dataset. QED

Different results appear when simulating models that include at least one random valued predictor. Two findings are notable: (a) distinct but related patterns for E[CV(d)] emerge when considering models consisting entirely of random valued predictors and those that use an intercept; and (b) two points of increasing instability appear at E[CV($d_{\max} - 1$)] and E[CV($d_{\max}$)]. The existence of the two-point instability appears to be robust to increasing dimensionality. Result (a) is expressed in the Conjectures 2.1 and 2.2, which also happen to provide implicit support for the observation stated in (b) via their singularities at $d = d_{\max} - 1$. For Conjectures 2.1 and 2.2, suppose that $Y \sim \mathrm{N}(X\beta, \sigma^2 \mathrm{I}_n)$, with β the assumed "true" linear model coefficient vector.

CONJECTURE 2.1: Let X be the n-by-p design matrix, where $p < n - 2$. Let X_j (the jth row of X) be such that $X_j \sim_{\mathrm{IID}} \mathrm{N}(\mathbf{0}, \Sigma)$, where $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ and $j \in \{1,\ldots,n\}$. Then, for $d = 1, \ldots, d_{\max} - 2$, the expected value for CV(d) is given by

$$\mathrm{E}\left[\mathrm{CV}\left(d\right)\right] = \sigma^2 \left(1 + \frac{p}{n - d - p - 1}\right). \qquad (2.7)$$

In the search for this equation, the author scrutinized simulated values for E[CV(d)], examining a variety of cases. Using the approximation in (2.3) as a starting point for exploring possible forms for the RHS of (2.7), the author eventually arrived at (2.7) through trial and error.

At $d = d_{max} - 2 = n - p - 2$ (the largest d value for which Conjecture 2.1 applies), (2.7) reduces to $E[CV(d_{max} - 2)] = \sigma^2(1 + p)$. At $d = d_{max} - 1$ (the first point of the two-point instability in the CV(d) error curve), (2.7) has a singularity. Though (2.7) and (2.3) are similar, the inclusion of $-p$ in the denominator of the dilation factor in (2.7) presents an obvious disagreement that becomes increasingly substantial as p increases. For example, the largest value that (2.3) can achieve is $2\sigma^2$, realized at $d = d_{max}$. Compare this to the maximum E[CV(d)] value $\sigma^2(1 + p)$, realized by (2.7) at $d = d_{max} - 2$.

Figure 2.1 shows results for the case $(n,p) = (10,1)$, with a single random valued predictor used in the model. The simulated E[CV(d)] error curve is displayed along with corresponding predicted E[CV(d)] error curves obtained using (2.7) and (2.3), so that all three error curves can be examined simultaneously. Note the congruity between simulated E[CV(d)] and predicted E[CV(d)] from Conjecture 2.1, and the widening (with d) disparity between simulated E[CV(d)] and predicted E[CV(d)] from the approximation provided in (2.3). Also note the blow-up in simulated E[CV(d)] at the two largest d values, reflecting the previously described two-point instability of the E[CV(d)] error curve when at least one random valued predictor is used in the model.

Figure 2.2(a) shows results for the case $(n,p) = (10,2)$, using a model with two random valued predictors. Figure 2.3(a) shows results for the case with $(n,p) = (20,8)$, using a model with eight random valued predictors. The same observations noted above for Figure 2.1 apply to Figure 2.2(a) and 2.3(a).

Now consider the case where an intercept is included in the linear model.

CONJECTURE 2.2: Let $X = [\mathbf{1}\ X_{RV}]$ be the n-by-p design matrix, where $p < n - 2$, and the first column of X is an intercept. Let X_j (the jth row of X_{RV}) be such that $X_j \sim_{IID} N(\mathbf{0}, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2,\ldots, \sigma_{p-1}^2)$ and $j \in \{1,\ldots,n\}$. Then, for $d = 1,\ldots, d_{max} - 2$, the expected value for CV(d) is given by

$$\mathrm{E}\left[\mathrm{CV}(d)\right] = \sigma^2\left(1+\frac{p}{n-d-p-1}\cdot\left(1-\frac{1}{n-d}\cdot\frac{2}{p}\right)\right). \qquad (2.8)$$

In the search for this equation, the author once again scrutinized simulated values for E[CV(d)], examining a variety of cases. This time, (2.7) was used as a starting point for exploring possible forms for the RHS of (2.8). Specifically, the author reasoned that substitution of an intercept for a random valued predictor reduces overall model complexity, suggesting that the E[CV(d)] expression for models that include an intercept might take the form of a dampened version of (2.7). Indeed, after much trial and error, this was found to be the case once the RHS of (2.8) was "discovered".

Like equation (2.7), at $d = d_{max} - 1$, (2.8) has a singularity. Note that (2.8) constitutes a downward adjustment of (2.7). Apparently, the $1/(n\text{-}d)$ term provides an adjustment for the reduced model complexity when substituting an intercept for a

random-valued predictor. The $2/p$ term dampens the adjustment as p gets larger and the general effect of this substitution on the model becomes less pronounced.

Figure 2.2(b) shows results for the case $(n,p) = (10,2)$, using a model with an intercept and one random valued predictor. Figure 2.3(b) shows results for the case $(n,p) = (20,8)$, using a model with an intercept and seven random valued predictors. The same general observations noted above for Figure 2.1 apply to Figures 2.2(b) and 2.3(b).

Simulation strongly supports the validity of the Conjectures. Graphical evidence for this assertion can be seen in Figures 2.1-2.3. APE values (2.5) computed comparing (2.7) and (2.8) to corresponding simulated E[CV(d)] were generally $O(10^{-2})$ to $O(10^{-1})$. To provide a gauge for these error magnitudes, E[REG] values were also simulated and compared to the known value of σ^2. APE values from this comparison were also generally $O(10^{-2})$ to $O(10^{-1})$, indicating that rounding error was solely responsible for the slight differences observed between simulated E[CV(d)] and predicted E[CV(d)] from (2.7) and (2.8).

As a final note, unlike E[CV(d)], CV(d) values from a single data sample will not necessarily be increasing in d. However, simulation results indicate that such behavior is exceptional. For example, testing 20,000 iterations with $(n,p) = (20,8)$ and no intercept resulted in three cases with CV(2)<CV(1). One of these cases even exhibited decreasing CV(d) from $d = 1$ to 4.

2.4.3. Connecting Simulation Results Back to Theory

Equation (2.3) was examined because it was the only explicitly stated estimate for E[CV(*d*)] found in the literature. This expression gives the *mean squared error of prediction* (MSEP) for using a linear regression model to make a prediction for some future observation *at a design point*. However, this is not an accurate characterization for CV(*d*), which is clear from the E[CV(*d*)] simulation results. Rather, the random subset design used for making "out-of-sample" predictions when computing the CV(*d*) statistic is more logically associated with the MSEP for using a linear regression model to make a prediction for some future observation *at a random X value*.

In Miller (2002), an expression is derived for the MSEP in the random X case, using a model with an intercept and predictor variables independently sampled from some fixed multivariate normal distribution. Miller credits this result to Stein (1960), but uses a derivation from Bendel (1973). Let row vector x_f contain the predictor values for some random future observation, with response value y_f. With this setup, we have $\mathrm{Var}[\hat{\beta}\,] = \sigma^2(X^{\mathrm{T}}X)^{-1}$, and we can write MSEP as

$$\begin{aligned} MSEP = E\left[\left\|y_f - x_f\hat{\beta}\right\|^2\right] &= E\left[\left\|y_f - x_f\beta\right\|^2\right] + E\left[\left\|x_f\beta - x_f\hat{\beta}\right\|^2\right] \\ &= \sigma^2 + \sigma^2 E\left[x_f\left(X^{\mathrm{T}}X\right)^{-1}x_f^{\mathrm{T}}\right] \\ &= \sigma^2\left(1 + \tfrac{1}{n} + E\left[\tilde{x}_f\left(\tilde{X}^{\mathrm{T}}\tilde{X}\right)^{-1}\tilde{x}_f^{\mathrm{T}}\right]\right). \end{aligned} \qquad (2.9)$$

In (2.9), "~" denotes that sample means have been removed, and predictor vectors are of length $p-1$. The "$1/n$" term in the dilation factor accounts for the variance of the intercept parameter estimated in the model.

Let Σ denote the covariance matrix for the X variables, so that the variance of future $\tilde{x}_f$'s will be $(1+1/n)\Sigma$ after removing the sample means. We can estimate Σ by

$$V = \left(\tilde{X}^T \tilde{X}\right)/\left(n-1\right), \tag{2.10}$$

which gives us

$$\tilde{x}_f \left(\tilde{X}^{\mathrm{T}} \tilde{X}\right)^{-1} \tilde{x}_f^{\mathrm{T}} = \frac{n+1}{n\left(n-1\right)} t V^{-1} t^{\mathrm{T}}.$$

In this expression, t is a vector of statistics with zero mean and covariance matrix Σ. $tV^{-1}t$ is a Hotelling T^2-statistic, which is a generalization of Student's t-statistic that is used in multivariate hypothesis testing. The quantity

$$(n-p+1)T^2/((p-1)(n-1))$$

is known to follow an F-distribution given by $F(p-1, n-p+1)$. Using the fact that the expected value of $F(v_1,v_2)$ is $v_2/(v_2-2)$, Miller (2002) obtains

$$E\left[\tilde{x}_f\left(\tilde{X}^{\mathrm{T}}\tilde{X}\right)\tilde{x}_f^{\mathrm{T}}\right]=\frac{n+1}{n(n-1)}\cdot\frac{(n-1)(p-1)}{n-p+1}\cdot\frac{n-p+1}{n-p-1}=\frac{(n+1)(p-1)}{n(n-p-1)}.$$

Substituting this expression into (2.9), we get

$$MSEP=\sigma^2\left(1+\frac{1}{n}+\frac{(n+1)(p-1)}{n(n-p-1)}\right). \tag{2.11}$$

If we replace $n - d$ with n in (2.8), then it is easy to show that (2.11) and (2.8) are equivalent.

Following Miller's derivation for the case using a model with an intercept, we can also derive an expression equivalent to (2.7) from Conjecture 2.1 (i.e., the "no intercept" case). First, the "1/n" term in (2.9) is no longer needed because all predictor and response variables are distributed with 0 mean, and no intercept is used in the model. Second, the expression used to estimate the covariance matrix V in (2.10) should be divided by n rather than $n - 1$. With these changes, we now have

$$x_f\left(X^{\mathrm{T}}X\right)^{-1}x_f^{\mathrm{T}}=\frac{1}{n}tV^{-1}t^{\mathrm{T}},$$

where $t = x_f$ and the T^2-statistic $tV^{-1}t^{\mathrm{T}}$ is such that $(n - p + 1)T^2/(pn) \sim F(p, n - p + 1)$. Using this setup, we determine that

$$E\left[x_f \left(X^{\mathrm{T}} X \right)^{-1} x_f^{\mathrm{T}} \right] = \frac{1}{n} \cdot \frac{np}{n-p+1} \cdot \frac{n-p+1}{n-p-1} = \frac{p}{n-p-1}.$$

Substituting this result into the intermediate expression on the second line of (2.9), we get

$$MSEP = \sigma^2 \left(1 + \frac{p}{n-p-1} \right). \qquad (2.12)$$

(2.12) is identical to (2.7), if we substitute n for $n - d$ in (2.7).

2.4.4. Conclusions and Future Directions

Conjectures 2.1 and 2.2 constitute the first proposed general formulas for E[CV(d)]. The link established between (2.7) and (2.8) and the random-X MSEP described in Miller (2002) indicates that Conjectures 2.1 and 2.2 generalize to multivariate normal X and $\varepsilon \sim (0, \sigma^2)$. In support of the error generalization, simulations using $\varepsilon \sim$ U($-a$,a), with $a \in [10^{-3},10^{3}]$, produce APE values on the same order as those observed using normally distributed ε.

Regarding the normality constraint on X, if we independently sample predictor values from U($-\sqrt{12}/2,\sqrt{12}/2$), then simulated E[CV(d)] values are less than the conjectured values. For example, with (n,p) = (20,8) and no intercept, E[CV(d)] values range from 2.8%-9.6% smaller than the conjectured formula in (2.7) as d

increases from 1 to (d_{max}–2), but simulated E[REG] values are unchanged (as expected, since E[REG] is independent of predictor distribution). Therefore, unlike some of the more general properties for OLS linear regression, E[CV(d)] depends on predictor distribution. Further theoretical development and additional simulation work would help expose this dependency.

Finally, the two-point instability phenomenon (which did not depend on predictor distribution) also warrants serious investigation, one that should begin by examining the development of the Hotelling T^2-statistic. The two-point instability indicates that OLS linear regression models fit using just 1 or 0 degrees of freedom must be unique in some way, compared to models fit using 2 or more degrees of freedom.

2.5. Simulating Optimal Model Selection Rates for CV(d)

2.5.1. Problem Background

The most groundbreaking result using CV(d) for asymptotic model selection appeared in Shao (1993). In this work, a pool of candidate predictors (columns of $X \in \mathbb{R}^{nxp}$, for some fixed p) is given, along with the quantity to be predicted (Y). Some of the predictors in X may not be related to Y, and these predictors would be expected to have 0 coefficients in β when included in a linear model. Associating models with column subsets of X, define the *optimal model* to be the column subset of X containing only the predictors with non-zero coefficients in β, and rank the different possible models using CV(d). Under some reasonable assumptions, the requirements

that $d/n \rightarrow 1$ and $n - d \rightarrow \infty$ as $n \rightarrow \infty$ are shown in Shao (1993) to be necessary and sufficient for the consistency of using CV(d) for optimal model selection, assuming that the optimal model is not the *full model* (the one constructed using all of the predictors).

The conclusion to be drawn from Shao (1993) is that when using CV(d) for model ranking in model selection, and a finite dimension optimal model is assumed (Shao 1997), a value for d that is an appreciable fraction of sample size n is preferred. Shao's result was interesting because it countered conventional wisdom of the time (largely driven by computational pragmatism) that heavily favored study of CV(1). However, no specific guidance is provided for practitioners using CV(d) for model selection, because the finite sample situation is inconsequential to the asymptotic result. For example, setting $d = \text{ceil}[n - n^{\alpha}]$, $0 < \alpha < 1$, satisfies Shao's two asymptotic criteria (namely, $d/n \rightarrow 1$ and $n - d \rightarrow \infty$ as $n \rightarrow \infty$), yet imposes no certain constraint on what values for d are desirable. The objective of the next simulation was to see if behavior reflective of Shao's criteria is indeed observable in the smallest sample setting.

2.5.2. "All Possible Subsets" Model Selection

Consider the equation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$, with $x_1, x_2, \varepsilon \sim_{\text{IID}} \text{N}(0,1)$ and $\beta_j \in \mathbb{Z}_2 = \{0,1\}$ for $j = 0,1,2$. Assume that at least one $\beta_j \neq 0$, so that there are seven unique coefficient vectors $\beta = [\beta_0\ \beta_1\ \beta_2]^{\text{T}}$. This setup is used to simulate data

observations $[x_1\ x_2\ y]$ for each of these particular coefficient vectors. n such observations using a particular coefficient vector are used to fill rows of the n-by-3 matrix $[X_1\ X_2\ Y]$, using sample sizes $n = 4,\ldots,20$. The resulting matrix provides a simulated data sample associated with the particular coefficient vector.

It is helpful to adopt the following naming convention to identify which coefficient vector is being used to simulate data samples (the "D" stands for data):

D1: $\beta = [1\ 0\ 0]^T$

D2: $\beta = [0\ 1\ 0]^T$

D3: $\beta = [0\ 0\ 1]^T$

D4: $\beta = [1\ 1\ 0]^T$

D5: $\beta = [1\ 0\ 1]^T$

D6: $\beta = [0\ 1\ 1]^T$

D7: $\beta = [1\ 1\ 1]^T$

Define the *pool of candidate predictors* to be $X_{POOL} = \{\mathbf{1}, X_1, X_2\}$, where the first element represents an intercept. X_{POOL} has seven unique non-empty subsets, each of which characterizes a candidate model (i.e., models defined using "all possible subsets" of candidate predictors are being considered). Let X_α be the predictor matrix associated with candidate model α, where $\alpha \in \mathbb{Z}_2^3$ is a binary-valued 3-vector indicating inclusion (1) or exclusion (0) of the individual members of X_{POOL}

in X_α. For example, $\alpha = (1,0,1)$ implies that $X_\alpha = [\mathbf{1}\ X_2]$. It is helpful to adopt the following naming convention to identify which predictors are being used in a model (the "M" stands for model):

M1: $\alpha = (1,0,0)$

M2: $\alpha = (0,1,0)$

M3: $\alpha = (0,0,1)$

M4: $\alpha = (1,1,0)$

M5: $\alpha = (1,0,1)$

M6: $\alpha = (0,1,1)$

M7: $\alpha = (1,1,1)$

Note the obvious correspondence between the data labels and the model labels. Using the labels provided above, consider the following data-model array:

D1: ***M1***, M2, M3, M4, M5, M6, M7

D2: M1, ***M2***, M3, M4, M5, M6, M7

D4: M1, M2, M3, ***M4***, M5, M6, M7

D6: M1, M2, M3, M4, M5, ***M6***, M7

D7: M1, M2, M3, M4, M5, M6, ***M7***

Each row of the array is associated with a particular data type, indicated on the left side of the array. Due to the distributional indistinction between X_1 and X_2, D3 and D5 cases are (for all practical purposes) redundant with D2 and D4 cases, respectively, and are thus excluded from direct consideration. Models outlined with a box denote *correct models* with respect to the specified data type (i.e., models that include all predictors in X_{POOL} contributing to Y). Correct models are "Class II" models in accordance with Shao (1993). The boxed model with bold, italicized text denotes the *optimal model*, which is the correct model with the smallest dimension (i.e., it includes only predictors in X_{POOL} contributing to Y). Models not outlined fall in Class I, which contains all *incorrect models* (i.e., models missing at least one predictor in X_{POOL} contributing to Y).

For each simulated data type, the author sought to determine the d-specific rates of optimal model selection using CV(d) to rank the seven candidate models, for sample sizes $n = 4,\ldots,20$. Call the d value exhibiting the highest rate of optimal model selection *optimal d*, or d_{opt}. The objective was to track the movement of d_{opt} as n increases, first for each particular data type, and then for arbitrary data type. Then, a qualitative assessment can be made regarding whether or not d_{opt} varies with n in a manner reflective of the asymptotic model selection criteria identified in Shao (1993) (namely, $d/n \to 1$ and $n - d \to \infty$ as $n \to \infty$). This behavior is referred to as "fall back", considering that one would like to observe d_{opt} generally increasing at the same rate as n (reflecting the criterion $d/n \to 1$ as $n \to \infty$), but with occasional bouts

of "fall back" where d_{opt} is non-increasing for consecutive n values (reflecting the criterion $n - d \to \infty$ as $n \to \infty$).

2.5.3. Results and Key Findings

Model selection histograms (MSHs) are displayed in Figure 2.4. Each colored curve in Figure 2.4 is a MSH, which shows simulated optimal model selection rates (probabilities) plotted against d. Optimal model selection rates using REG for model ranking also were simulated to provide a familiar comparative gauge for optimal model selection rates using CV(d). For convenience, REG results are plotted at $d = 0$. To obtain these curves, 12,000-40,000 iterations were evaluated for each sample size, for each of the five non-redundant data types.

Figures 2.4(a)-(e) show results obtained using specific optimal model types, ordered by increasing optimal model complexity. One model is more complex than another if it (*i*) has a larger p or (*ii*) has the same p but the less complex model includes an intercept and the more complex model includes only random-valued predictors. In particular, note the trajectories for d_{opt} (identified by diamond-framed black dots) in each subplot as n increases.

When minimally complex M1 is the optimal model (Figure 2.4(a)), $d_{opt} = d_{max}$ for all n. This "mean model" case is not specifically exempted in Shao (1993), and the non-conformity of d_{opt} to the finite-sample "fall back" criteria described in the previous section may be cause for concern. However, the author believes that this result is more likely due to the lack of consideration of the intercept as a predictor that

is characteristic of asymptotic studies (including Shao 1993). When maximally complex M7 is the optimal model (Figure 2.4(e)), $d_{opt} = 1$ for all n. This "full model" case is specifically exempted in Shao (1993), and thus the non-conformity of d_{opt} to the finite-sample "fall back" criteria does not present a conflict. Otherwise, "fall back" behavior generally can be observed for the remaining cases M2-M6 (Figures 2.4(b)-(d)), with an increased amount of "fall back" occurring as optimal model complexity increases.

More interesting than the optimal model-specific results is the average result, whereby any one of the seven candidate models is equally likely to be optimal. MSHs for this more general situation are shown in Figure 2.4(f). The observation of "fall back" behavior of d_{opt} as n increases in Figure 2.4(f) provides a convincing illustration that the asymptotic model selection criteria identified in Shao (1993) are indeed influential already in this smallest sample setting.

To test the robustness of the results shown in Figure 2.4(f), the linear independence assumption of x_1 and x_2 was relaxed. Simulation results were recreated using x_1 and x_2 such that both were N(0,1) but Corr[x_1, x_2] ≈ 0.8. The average MSH across all seven data types is shown in Figure 2.5. Using correlated random valued predictors (*i*) leads to a pronounced decrease in optimal model selection rates using REG or CV(d); (*ii*) markedly flattens the CV(d) optimal model selection rate curves, reducing the advantage exhibited by CV(d_{opt}) and points nearby; and (*iii*) expedites the "fall back" effect characterizing the best d as n increases. Most of the loss summarized in observations (*i*) and (*ii*) is attributable to the very poor performance

(not shown) of CV(d) for cases when M6 or M7 is the optimal model, which are the two models that include both X_1 and X_2.

In addition to the focus on the behavior of d_{opt} with respect to the asymptotic model selection criteria found in Shao (1993), optimal model selection rates for REG were simulated for comparison to CV(d) optimal model selection rates. As a matter of convenience, optimal model selection rates for REG are plotted at $d = 0$ in Figure 2.4(a)-(f). If one observed REG generally outperforming CV(d), then this would provide evidence that CV(d) provides no better tool for model ranking than REG, which is better understood and much easier to compute. This was not found to be the case. REG tended to perform better than CV(d) for the smallest sample sizes, for cases where one of the higher dimensioned models was the optimal model (Figures 2.4(c)-(e)). However, when each of the seven candidate models was equally likely to be the optimal model (which was the most general situation studied), then for every sample size at least CV(d_{opt}) was better at identifying the optimal model than REG (Figure 2.4(f)). This observation also held when examining correlated predictors (Figure 2.5). Furthermore, the gap between optimal model selection rates using CV(d_{opt}) and REG widened with increasing sample size, with an increasing number of d values resulting in CV(d) that outperformed REG.

2.5.4. Fixed Dimension Model Selection

Implicit to the "all possible subsets" model selection problem is the *fixed dimension model selection* (FDMS) problem, which refers to the problem of ranking

models with the same dimension. Fewer statistics are available for resolving FDMS problems than "all possible subsets" model selection problems because such statistics can no longer utilize differences in dimensionality to assist with model assessment.

In this section, we use simulation to test whether or not Shao's "all possible subsets" model selection result manifests itself in small sample FDMS problems in a manner similar to the "all possible subsets" simulation examined earlier. We also examine results from a similar FDMS simulation that allows for consideration of another statistic that can be used for FDMS problems, which we now describe.

Let $\hat{Y}$ denote the "full model" response predictions for Y, and let $\hat{Y}_\alpha$ denote the response predictions using some candidate model labeled by α, and which has p predictors. Let X_α denote the predictor matrix for candidate model α. Besides CV(d), few other statistics are available that can be applied to FDMS problems and possibly produce a ranking distinct from simple l^2 model fit (sum of squared errors, or SSE). For example, Mallows' C_p

$$C_p = \frac{\left\| Y - \hat{Y}_\alpha \right\|^2}{\left\| Y - \hat{Y} \right\|^2} - n + 2p$$

and the Akaike information criterion (AIC)

$$\text{AIC} = n \ln\left(\left\| Y - \hat{Y}_\alpha \right\|^2 \right) + 2p$$

are equivalent to SSE in FDMS, because their model complexity penalty parameters (the last term in each expression) are constant for fixed p. The same is true for most other information criteria, such as the Beyesian information criterion (BIC; Schwarz 1978) and the corrected AIC (AIC_c; McQuarrie & Tsai 1998). The exception with the simplest form is the Fisher information criterion (FIC; Wei 1992), which uses a data-dependent complexity penalty parameter. The FIC for model α is given by

$$FIC = \frac{\left\| Y - \hat{Y}_\alpha \right\|^2}{\left\| Y - \hat{Y} \right\|^2} + \ln\left(\det\left(X_\alpha^{\mathrm{T}} X_\alpha\right)\right). \qquad (2.13)$$

Like many other information criteria, the FIC requires an estimate for "full model" error (see the denominator of the fraction in (2.13)), so that the FIC is only applicable for model selection problems where the number of candidate predictors (pool size v) is less than the number of observations (n). This presents a pool size constraint for model selection problems that use the FIC.

We describe results from two related FDMS simulation studies, denoted FDMS-3 and FDMS-2. In FDMS-3, small sample optimal model selection rates using SSE, FIC, and CV(d) are simulated and compared to determine which of these statistics is best capable of optimal model identification when the model dimension is fixed. This simulation is subject to the pool size constraint (namely, $v<n$) imposed by the FIC (see the "FIC Applicable" region in Table 2.1). The FDMS-2 simulation is

an extension of the FDMS-3, dropping consideration of the FIC so that the pool size constraint could be ignored. All (n,v)-pairs represented in Table 2.1 were examined for FDMS-2. The distributional constraints imposed on the X variables and errors ε during the main "all possible subsets" simulation were used here. Approximately 12,000 model selection iterations were evaluated for each (n,v,p) case, both with and without an intercept.

Six different optimal model forms were examined that contained from 1 to 3 random valued predictors, with or without an intercept, so that p ranged from 1 to 4:

$$\{x_1,\ 1+x_1,\ x_1+x_2,\ 1+x_1+x_2,\ x_1+x_2+x_3,\ 1+\ x_1+x_2+x_3\}.$$

Let p denote model dimension, and define the number of random valued predictors in the model as $p_{RV} = p$-1 or p, depending on whether or not an intercept is used. Given a particular optimal model form and simulated data set (predictor pool $\{X_1, X_2, \ldots X_v\}$ and error ε), the first p_{RV} predictors were substituted into the optimal model form and added to ε to provide simulated response values Y. For these simulations, use of the intercept was fixed so that either an intercept was used in the optimal model and all candidate models, or it was not used at all. When no intercept was used, this allowed the maximum pool size to be increased by one considered (see "RV" entries in Table 2.1).

A variety of small sample sizes (n = 6:14), candidate predictor pool sizes (v = 4:16), and model dimensions (p = 1:4) were considered for these FDMS studies. For

example, if $v = 14$ and $p_{RV} = 3$, then there are 14-choose-3 = 364 candidate models to evaluate (which includes the one "correct" model).

To look for "fall back" behavior, we must obtain a single d_{opt} value for each n and examine the trajectory of d_{opt} as n increases. Therefore, it was necessary to aggregate results across different pool size values v. Simple averaging was used for this purpose. Pool size has the obvious influence that optimal model selection rates decline as pool size increases. This is because more models must be considered, which increases competition bias. For example, the FDMS-2 MSHs shown in Figure 2.6 generally portray lower model selection rates than the FDMS-3 MSHs shown in Figure 2.7. This is because in FDMS-2, in addition to considering the same pool sizes used in FDMS-3, larger pool sizes were also evaluated (see Table 2.1).

An examination of the simulation results displayed as MSHs in Figures 2.6 and 2.7 leads to the following observations:

- [FDMS-2 and FDMS-3] Shao's "fall back" behavior for d_{opt} is observed to some degree for each optimal model form, except the simplest one (x_1; see Figures 2.6(a) and 2.7(a)). The condition $d/n \rightarrow 1$ as $n \rightarrow \infty$ becomes more pronounced with increasing model complexity.
- [FDMS-2 and FDMS-3] SSE outperforms CV(d_{opt}) in every instance. Looking at ratios of optimal model selection rates (Figure 2.8), there is no indication that CV(d_{opt}) will eventually overtake SSE as sample size increases, regardless of model complexity. However, looking at differences in optimal

model selection rates (Figure 2.9), CV(d_{opt}) rates may eventually catch up to (i.e., become indistinguishable from) SSE rates as n increases.

- [FDMS-3] SSE outperforms FIC in nearly every instance (Figure 2.7). As model complexity increases, FIC begins to outperform SSE for the very smallest sample sizes. Looking at the trend in the ratio (Figure 2.8) and difference (Figure 2.9) between these two model selection rates, it appears that SSE will maintain its dominance over FIC indefinitely as sample size increases, regardless of model complexity.
- [FDMS-3] As sample size increases, CV(d_{opt}) eventually outperforms FIC. For the smallest sample sizes, FIC outperforms CV(d). The more complex the model form, the greater the n at which CV(d_{opt}) first outperforms FIC. See Figures 2.6-2.9.

2.5.5. Conclusions and Future Directions

The main objective of the model selection simulations was to see if small sample behavior of d_{opt} reflected the asymptotic model selection criteria identified in Shao (1993). This was achieved. Consequently, practitioners may want to consider d values that are an appreciable fraction of n when faced with "all possible subsets" or "fixed dimension" model selection problems. Additional simulations using larger sample sizes, alternative predictor and error distributions, and larger predictor pools (more candidate models) would help expose the general small-sample behavior of the CV(d) statistic when used for model selection. Also, it would be useful to simulate

model selection rates of alternative model selection statistics (such as various information criteria) in the "all possible subsets" simulation framework examined here, to compare against the model selection rates observed using CV(d).

The other notable result from the "all possible subsets" model selection simulation was that using CV(d) for model ranking is generally preferable to using REG, so long as a judicious choice of d is made. On the other hand, when considering FDMS problems, SSE was generally found to provide the most effective model selection statistic when compared to FIC or CV(d). Considering these two outcomes, this suggests that a "divide and conquer" strategy might be more effective in resolving an "all possible subsets" problem than directly competing all possible models. By "divide and conquer", it is meant to subdivide the "all possible subsets" problem into a collection of FDMS sub-problems. Use simple SSE to select the best model within each FDMS sub-problem, and then use a different statistic (such as CV(d) or some information criterion) to select among the resulting set of differently dimensioned, "best" models to identify the overall optimal model.

Tables and Figures

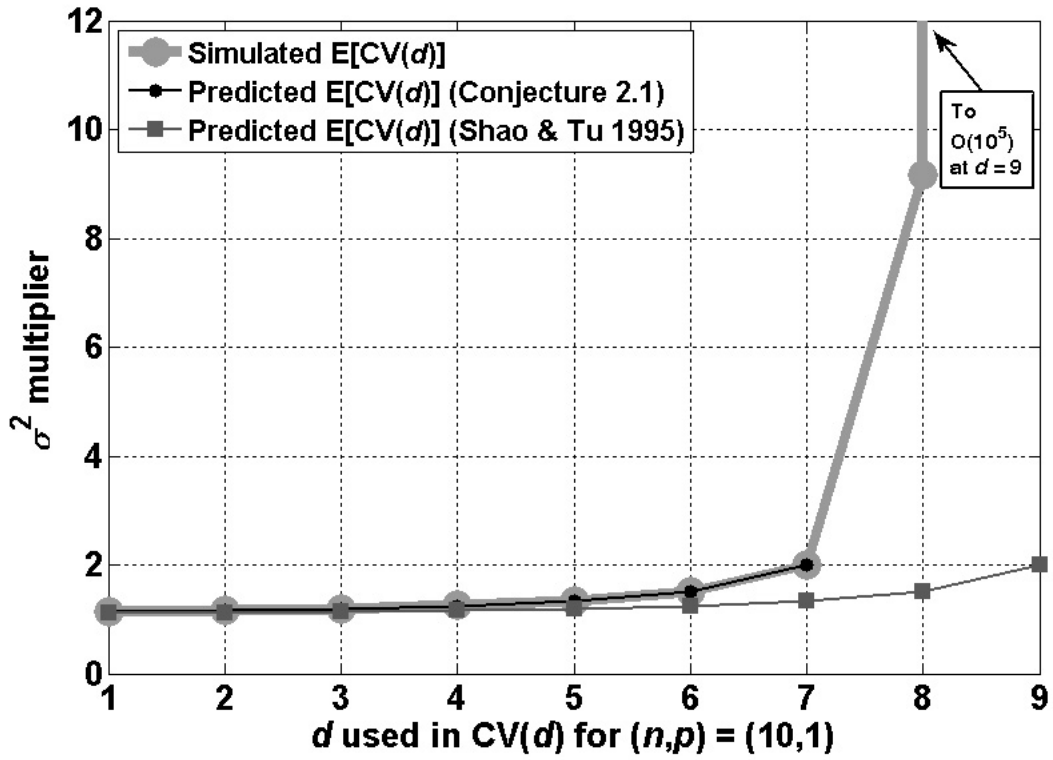


Figure 2.1. A comparison between simulated and predicted E[CV(d)] error curves using the linear model with a single random valued predictor X_1, for sample size n = 10. Note the good correspondence between the simulated E[CV(d)] values and the predicted values from Conjecture 2.1. Also note how simulated E[CV(d)] values blow up at $d = d_{\max} - 1 = 8$ and $d = d_{\max} = 9$. This abrupt behavior change in simulated E[CV(d)] is compatible with the d-value limitations of Conjectures 2.1 and 2.2, which are both inapplicable for the two largest d values.

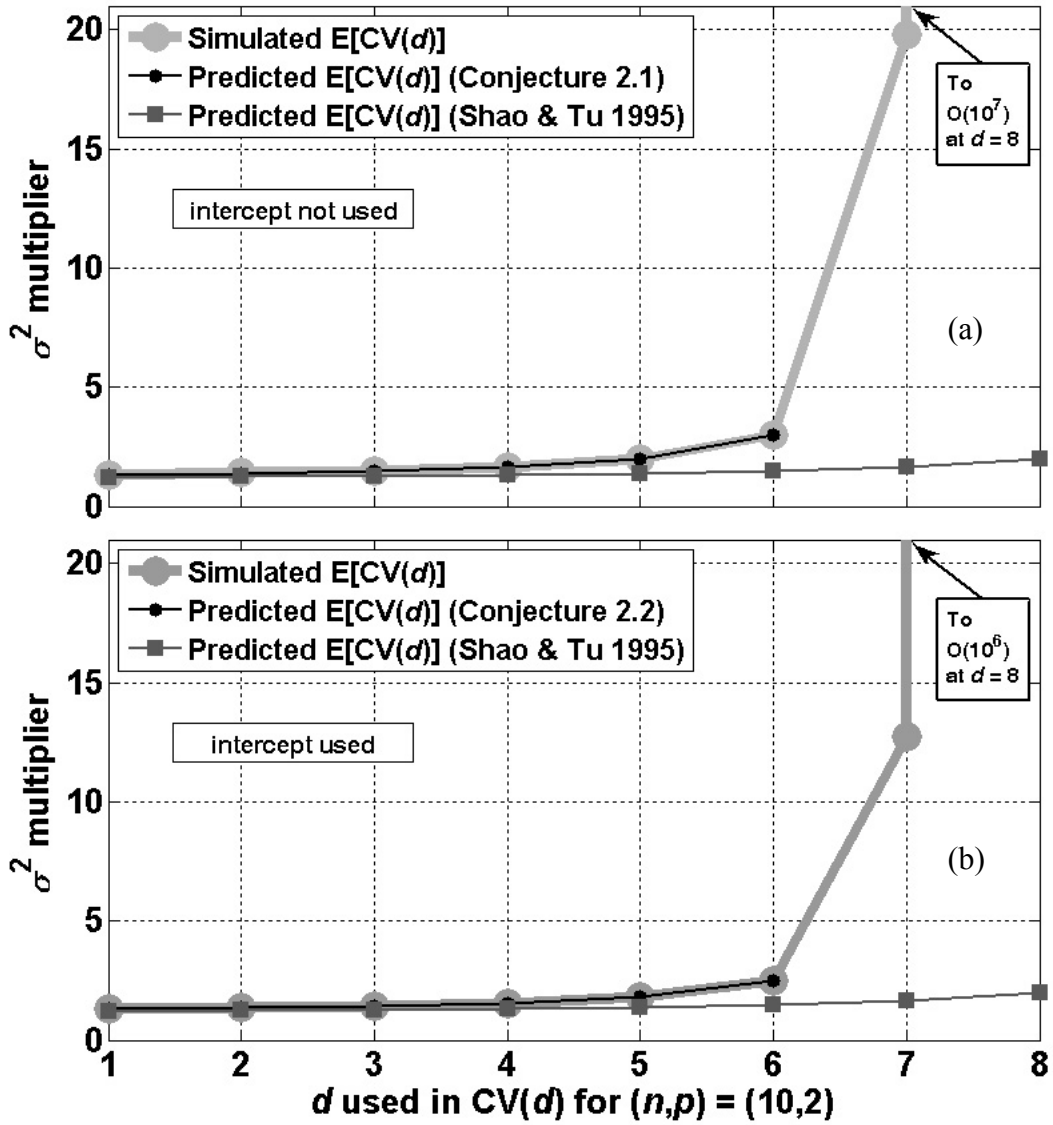


Figure 2.2. A comparison between simulated and predicted E[CV(d)] error curves using the linear model with predictors (a) [X_1 X_2] and (b) [**1** X_1], for sample size n = 10. Note the good correspondence between the simulated E[CV(d)] values and the predicted values from Conjectures 2.1 and 2.2. Also visible is the two-point instability, with simulated E[CV(d)] blowing up at $d = d_{\max} - 1 = 7$ and $d = d_{\max} = 8$.

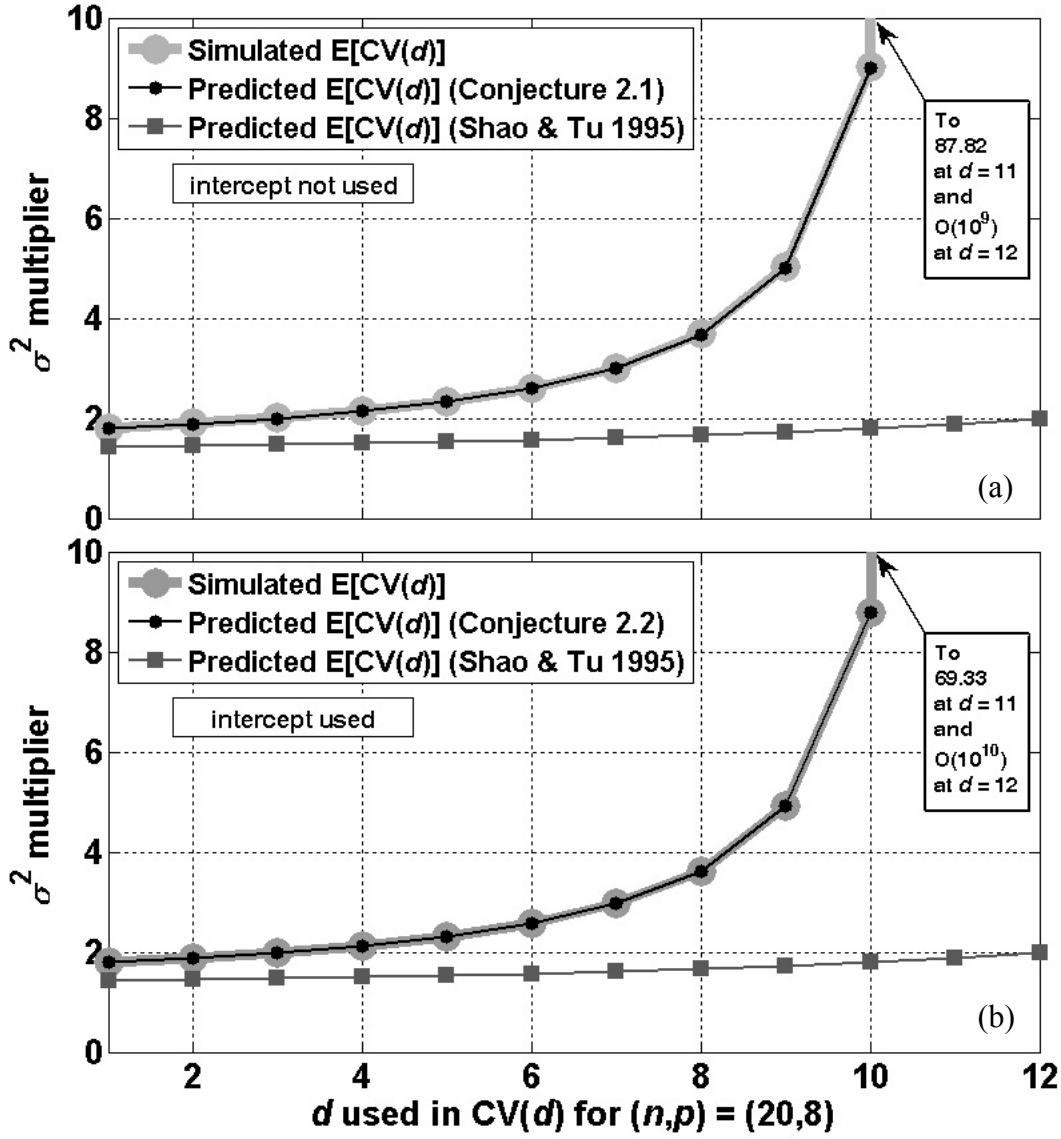


Figure 2.3. A comparison between simulated and predicted E[CV(*d*)] error curves using the linear model with predictors (a) [$X_1 \ldots X_8$] and (b) [**1** $X_1 \ldots X_7$], for sample size $n = 20$. Note the good correspondence between the simulated E[CV(*d*)] values and the predicted values from Conjectures 2.1 and 2.2. Also visible is the two-point instability, with simulated E[CV(*d*)] blowing up at $d = d_{\max} - 1 = 11$ and $d = d_{\max} = 12$.

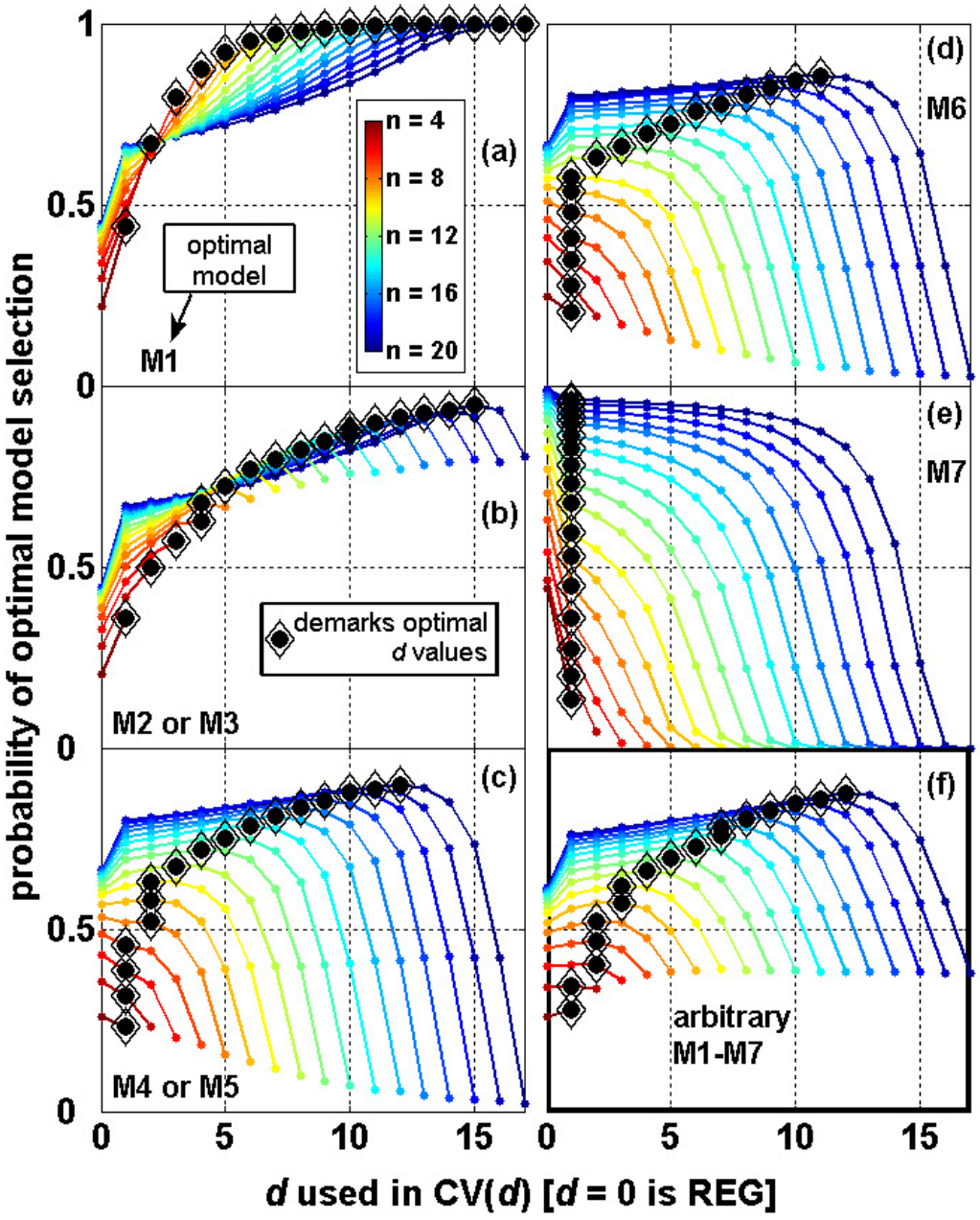


Figure 2.4. The Model Selection Histograms (MSHs) for $n = 4,\ldots,20$ are shown, using results obtained from the "all possible subsets" model selection simulation with optimal model (a) M1, (b) M2 or M3, (c) M4 or M5, (d) M6, or (e) M7. Subplot (f) shows the general MSHs averaged across the seven different optimal model types, reflecting the case whereby any one of the seven models is equally likely to be the optimal model.

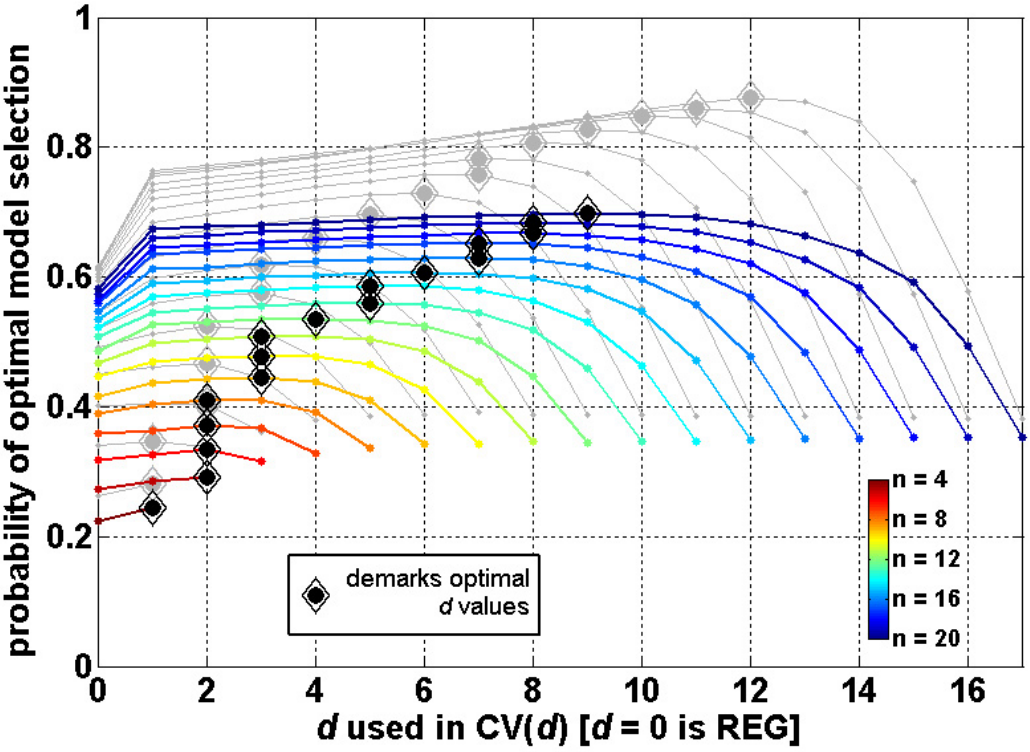
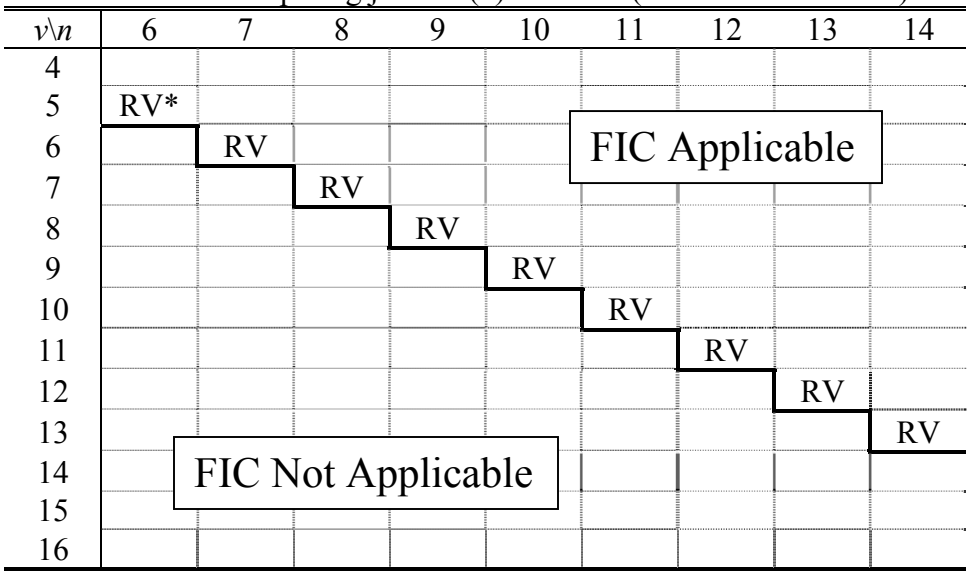


Figure 2.5. The colored lines show, for $n = 4,\ldots,20$, the average MSHs that result when any one of the seven models is equally likely to be the optimal model. For these MSHs, x_1 and x_2 were simulated such that $\text{Corr}[x_1, x_2] \approx 0.8$. Data from the uncorrelated case (see Figure 2.4(f)) are shown in light gray for comparison.

Table 2.1. Examined (n,v) = (sample size, pool size) pairs for the FDMS simulations. When considering the FIC (simulation FDMS-3), only cases to the upper right of the diagonal divider ($v<n$) could be considered due to the pool size constraint imposed by the form of the FIC. All cases were considered when comparing just CV(d) and SSE (simulation FDMS-2).



* "RV" denotes cases where FIC is applicable only for model forms with no intercept.

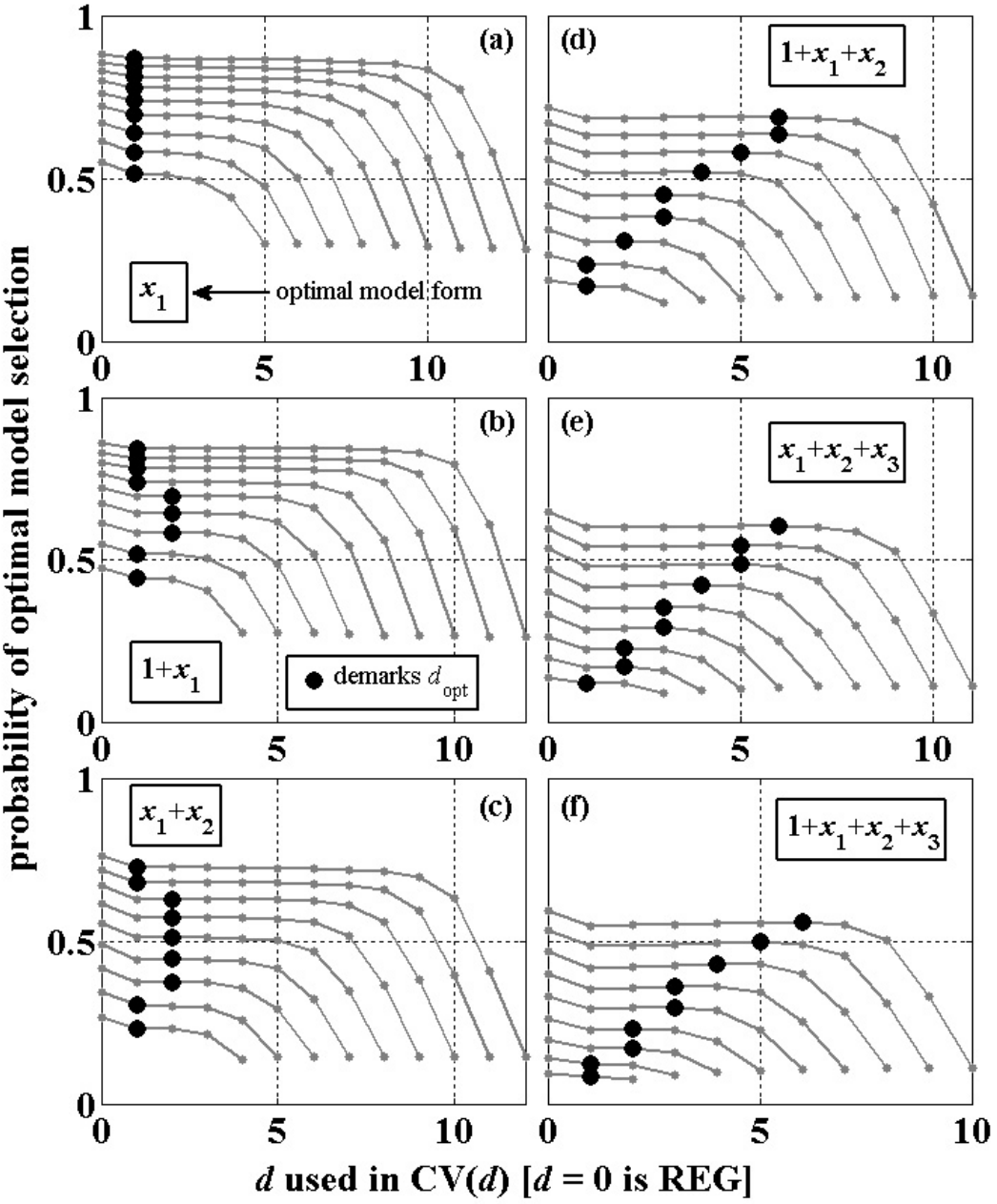


Figure 2.6. Model selection histograms for FDMS-2. In each subplot, results are shown for $n = 6$ (shortest curve) to $n = 14$ (longest). Each curve depicts the average MSH observed over the pool size range $v = 4{:}16$. The number of terms in the "optimal model form" shown in each subplot gives the model dimension p of the FDMS problem at hand, explaining the scaling of d in each plot (recall that the upper bound for d is $d_{max} = n - p$). For convenience, REG results are plotted at $d = 0$.

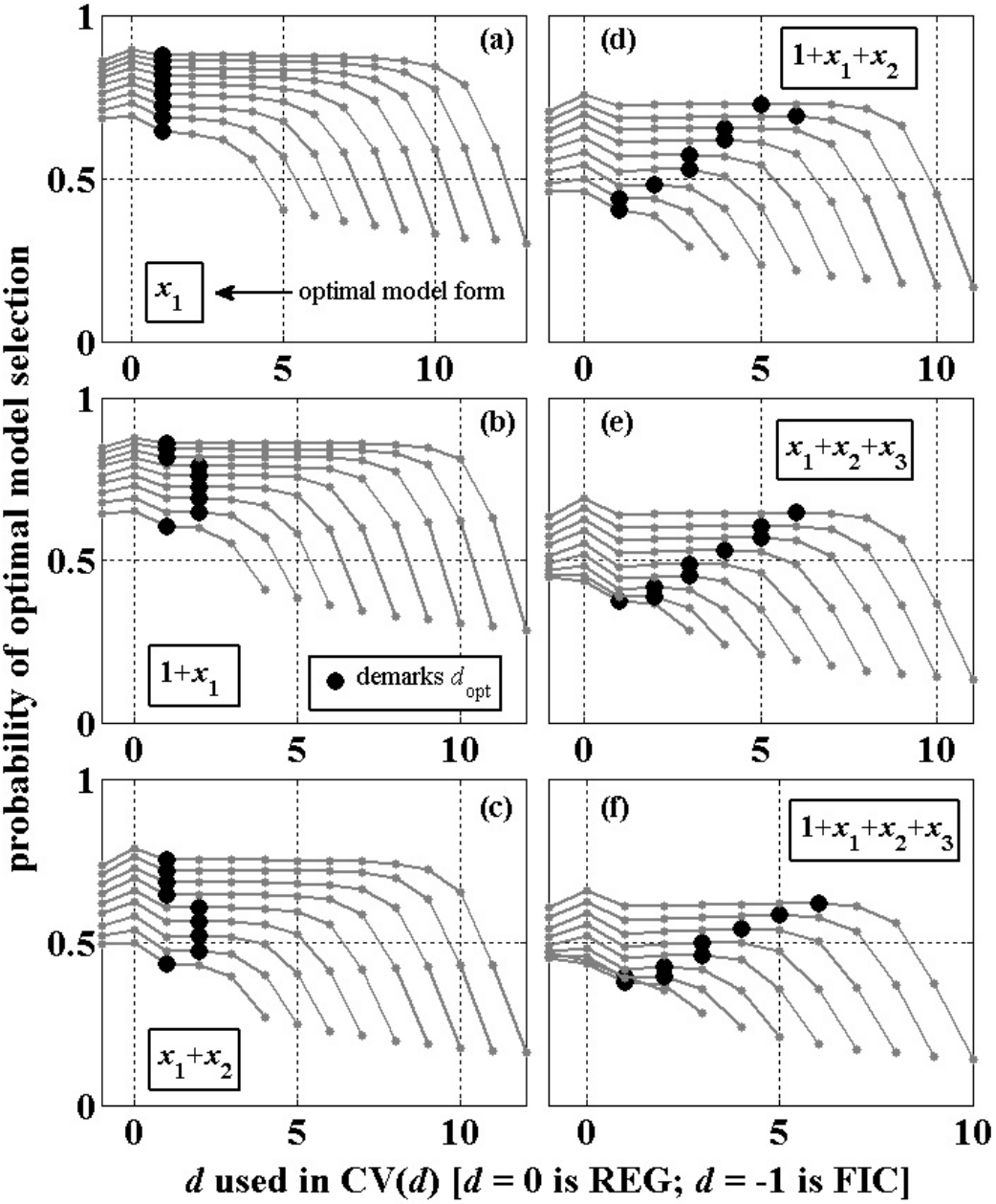


Figure 2.7. Model selection histograms for FDMS-3. In each subplot, results are shown for $n = 6$ (shortest curve) to $n = 14$ (longest). Each curve depicts the average MSH observed over the available "FIC Applicable" pool size range shown in Table 2.1. For convenience, REG results are plotted at $d = 0$ and FIC results are plotted at $d = -1$.

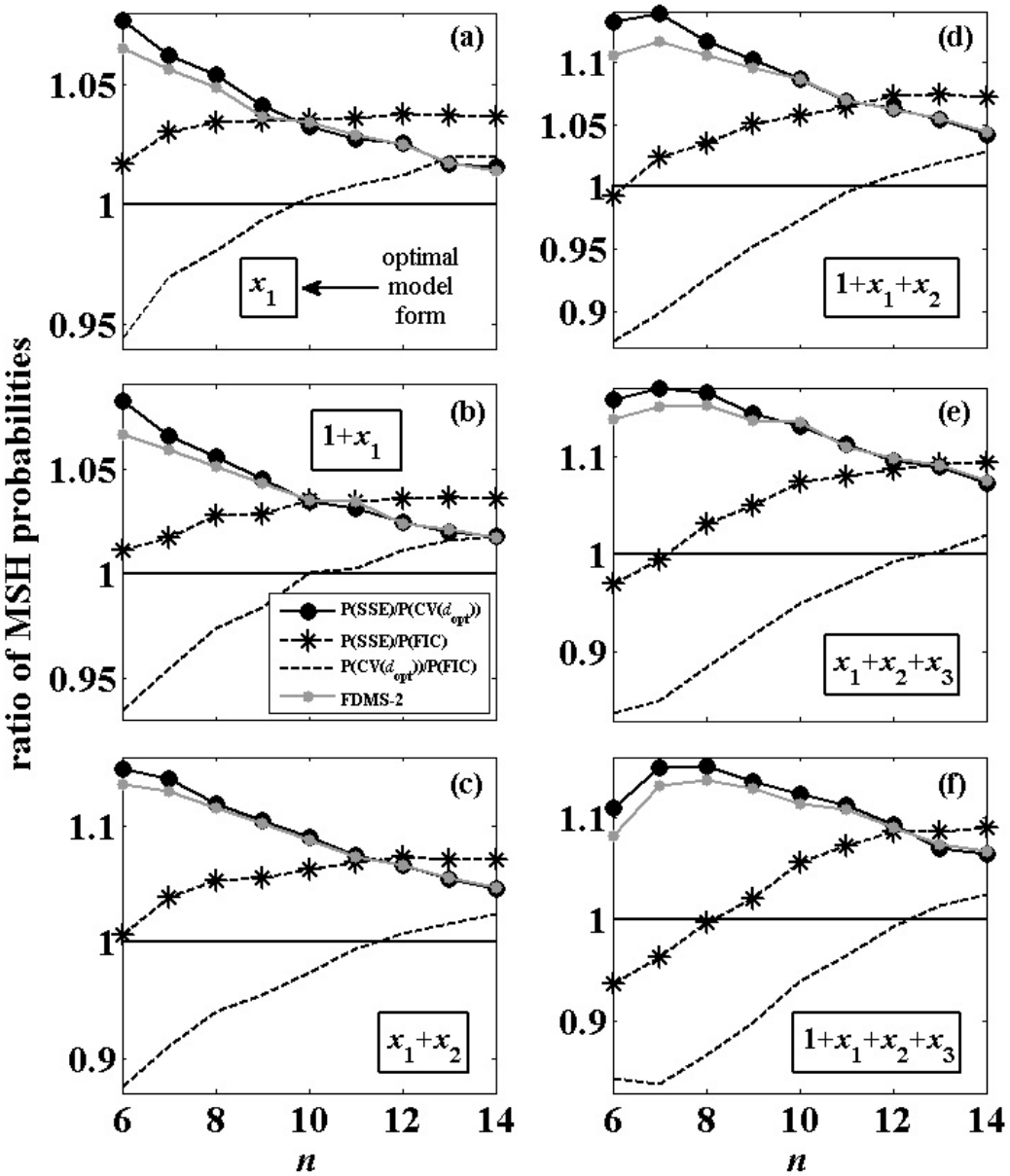


Figure 2.8. To examine the relative asymptotic tendencies of the optimal model selection rates (denoted P(·) in the legend) of the different statistics used for model selection, several ratios of these rates were computed and plotted here. Each of the black curves depicts a specific optimal model selection probability ratio computed using results shown in Figure 2.7. The curve labeled "FDMS-2" uses the ratio P(SSE)/P(CV(d_{opt})) determined from the values shown in Figure 2.6. Note that FIC probabilities are surpassed by REG and CV(d_{opt}) probabilities as sample size grows. Also, SSE probabilities are always better than CV(d_{opt}) probabilities, but the gap closes as sample size increases.

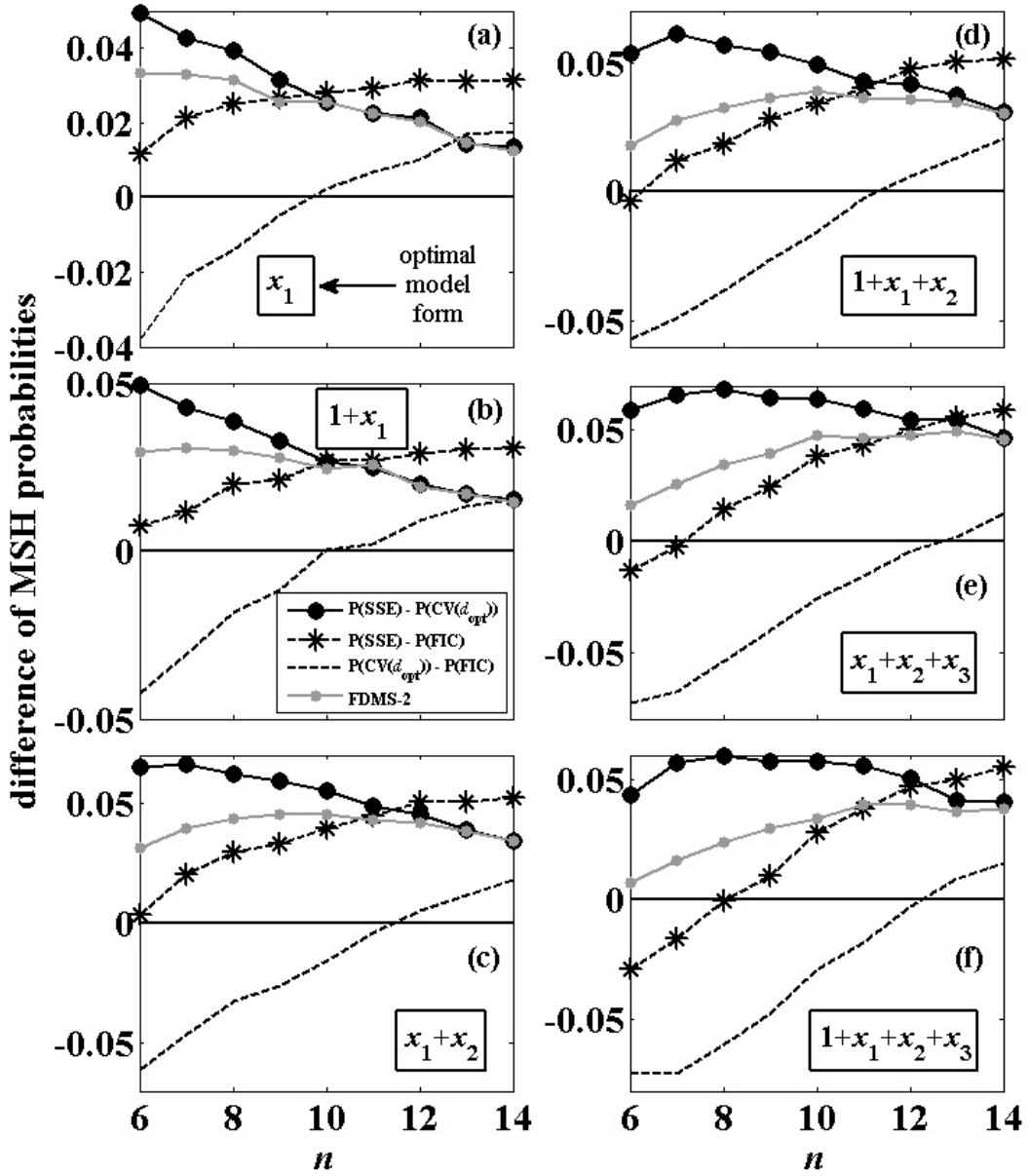


Figure 2.9. Same as Figure 2.8, but differences between (rather than ratios of) optimal model selection rates are shown. These differences exhibit the same general patterns as the ratios shown in Figure 2.7. Note that FIC probabilities are surpassed by REG and CV(d_{opt}) probabilities as sample size grows. That the difference between SSE and CV(d_{opt}) rates is generally decreasing for larger sample sizes provides evidence that these quantities might be asymptotically consistent with respect to optimal model selection rate.

Page left intentionally blank.

Chapter 3. A New Method for Floodplain Modeling

Chapter Summary

Based on simplistic pixel-level surface flow properties derived from gridded elevation data, the FLDPLN ("floodplain") model was developed to estimate floodplain extent as a function of floodwater depth. The FLDPLN model is described in this paper. The model has significant advantages over existing methods such as traditional hydrodynamic models. For example, the FLDPLN model is nearly automated and has few input requirements. FLDPLN can be used to identify and map historic floodplains (river valleys), a capability demonstrated using several examples. Also, FLDPLN can be used to estimate inundation extent for major, sustained flood events. This is demonstrated in a validation study in which the FLDPLN model was used to analyze 130 km of stream length from a forked river network in western Missouri that experienced extreme flooding in 2007. Using the maximum mean daily water surface elevation values recorded at three gaging stations during flood crest (one from each branch of the forked system), the model predicted the flood extent with 87.2% accuracy. This result is outstanding compared to similar studies reported in the literature using hydrodynamic models.

3.1. Introduction

The delineation[1] of floodplains and flood prone areas within them for major river reaches is necessary and desirable at some level of accuracy to address a number of needs. Floodplain areas adjacent to and within nearly all major municipalities have been mapped using Federal Emergency Management Agency (FEMA) methods, other hydrologic and hydraulic methods, or Natural Resources Conservation Service soils maps. FEMA maps are often dated and restricted to very localized floodplain areas because of their high cost. More rapid and affordable methods are necessary to identify flood prone areas over extensive river reaches to provide information about flood potentials that can affect floodplain development by individuals, and private and public organizations.

All rivers reside within a floodplain that contains the extent of their historical meanders and bank overflows. For rivers in shallow-slope landscapes that are prone to periodic flooding (such as those occurring in Kansas), the floodplain usually appears as a topographically distinct landscape feature commonly referred to as a "river valley". The river valley is a slowly evolving (and therefore relatively static) landscape feature, the result of the integration over time of fluvial (flowing water) erosion processes that contributed to the valley's formation. In light of this description, the term "historic floodplain" will be used synonymously with "river valley".

[1] In flood mapping, "delineation" always refers to the two-dimensional, "flat map" representation (i.e., the *xy*-projection) of the areal extent.

In this paper, the term "floodplain" will also be used in reference to a more detailed definition, whereby a particular floodplain extent is associated with a particular floodwater depth. The "floodwater depth" (or "flood depth", or "water depth", or later, "depth to flood") value assigned to an off-stream point refers to the depth of water required above normal local stream flow to inundate that off-stream point. Given some flood depth h, the subset of the historic floodplain with floodwater depth values $\leq h$ comprises the floodplain associated with flood depth h.

3.2. Existing Methods

Identifying a floodplain for a particular stream (or river) using topographic data is complicated by local topographic variability and the downhill trajectory of the stream course. While the historic floodplains of many rivers (especially large rivers) are visually recognizable in most aerial photography and digital images, the actual delineation of floodplains and flood-prone areas is achieved using one of three methods: (i) manual delineation (visual interpretation); (ii) statistical topographic detrending; and (iii) evaluating a hydrodynamic model. We now provide brief descriptions for these methods.

3.2.1. Manual Floodplain Delineation

Option (i) is only possible in cases where sufficient (for the user's purposes) portions of the floodplain boundary corresponding to upland valley walls are visibly recognizable in the topographic map. The historic floodplain is generally the area

contained by the upland valley walls, and as such can be defined by drawing or digitizing the most visual breaks between the flat lowland adjacent the river course and the upland areas. Figure 3.1 shows two examples with topographically distinct river valleys. This graphic delineation process is often facilitated by rather abrupt lowland to upland changes in land cover and topography. However, this procedure is imprecise and difficult to replicate due to frequent interpretational uncertainties (such as what to do at floodplain confluences and other flat areas).

3.2.2. Detrending Topographic Data for Floodplain Identification

Option (ii) refers to applied statistical approaches for detrending the topographic map near a stream, so that off-stream elevation values are made relative to nearby stream elevation. If one assigns an elevation value of zero to all points on the stream, then relative elevation values from nearby off-stream points can be assumed to reflect the floodwater depth above normal flow required for inundation of these points.

This approach is popular in ecological studies, where relative elevation provides a potentially useful explanatory variable (e.g., see Lea & Diamond 2006, Turner *et al.* 2004, Poole *et al.* 2000). In these and other studies, relative elevation is used as an indicator of fluvial "connectedness" of off-stream points to the stream or stream network under investigation. Since flood frequency varies inversely with floodwater depth, locations with smaller relative elevation values are expected to be more frequently "reconnected" with the nearby stream via floodwaters. Connectivity

information is useful for the study of nutrient cycling and other processes characteristic of river-floodplain ecosystems, along with the study of the effects of disturbance (and restoration) of such systems caused by human activity (Sparks *et al.* 1990, Bayley 1995, Buijse *et al.* 2002, Tockner *et al.* 2002).

The advantage of purely statistical detrending approaches is that they are easy to apply and can produce useful, visually appealing results. In addition to providing a hydrologic connectivity index, the floodplain extent for various floodwater depth values can be estimated using level sets from the relative elevation map. The disadvantage is that these approaches are highly subjective and do not consider any topographic information beyond spatial proximity and elevation. The output will depend heavily on how the reference stream elevation values are spatially extrapolated for subtraction from the elevation map, a process that becomes increasingly indeterminate for locations farther from the set of reference points. Proper study area delineation is also necessary, to confine the analysis to regions that drain into the target stream. Otherwise, areas that do not drain into the target stream will be assigned meaningless relative elevations values. For example, a previously delineated river valley boundary was used to define the study area in Lea & Diamond (2006). Because no actual surface flow information is used, output from this method is of limited utility for use in surface hydrology studies.

3.2.3. Hydrodynamic Flood Extent and Floodplain Estimation

Option (iii) refers to the various, widely used hydrologic and hydraulic models, which require numerous input variables and boundary condition specifications. First, a hydrologic model is used to simulate local stream discharge rates. These discharge rates are then fed into a hydraulic model, which is used for flood extent estimation. Due to the physical basis and implementation success of hydrodynamic models, professionals and governmental agencies concerned with estimating observed or potential flood extents use such models almost exclusively. The cost for such detailed studies can be more than $8000 per mile (Lear *et al.* 2000), and thus the mapping of long stream segments can become prohibitively expensive. Successful implementation of hydraulic models is greatly facilitated by detailed geometric information obtained from manual ground surveys at floodplain cross sections. However, the recent and increasing availability of high resolution elevation data ($\leq$ 2 m horizontal resolution, with $\pm$ 10 cm vertical accuracy) should help alleviate the need for ground survey data, helping to reduce the overall cost for such models.

To better understand the complexities associated with the hydrodynamic modeling approach to the flood extent estimation problem, it is helpful to describe, using general terms, how such a solution is typically obtained. First the user designates a study area containing a stream segment for which a particular flood extent or general floodplain is to be estimated. A hydrologic surface runoff response model (e.g., the U.S. Army Corps of Engineer's Hydrologic Engineering Center's

Hydrologic Modeling System, or HEC-HMS; see USACE HEC 2000 for technical details) is then evaluated using precipitation intensity and duration parameters describing some recorded storm or a hypothetical design storm. The output from the hydrologic model is an estimate for the stream *hydrograph*, which shows stream discharge rate as a function of time for some location on the stream. For example, the "100-year" design storm should produce a maximum stream discharge rate that is expected to have a 1% annual probability of being observed or exceeded. Lateral (from the stream channel) floodwater extent is then estimated using a hydraulic model, for which the hydrograph provides necessary boundary condition information. If the flood extent for a particular storm event is desired, then the entire hydrograph is used as a time-dependent, variable flow boundary condition representing rising and falling floodwaters. If a general floodplain (e.g, the "100-year floodplain") is desired, then the peak discharge from the hydrograph provides a constant flow boundary condition. The latter approach is typically used for most engineering applications, including design implementation and FEMA flood studies.

The U.S. Army Corps of Engineer's Hydrologic Engineering Center's River Analysis System (HEC-RAS) software is commonly used for hydraulic modeling in the U.S. (see USACE HEC 2002 for technical details). In fact, use of the HEC-RAS model is currently a requirement for flood studies approved by FEMA. HEC-RAS employs one-dimensional, open channel flow models (in open channel flow, gravity is the only force that can cause flow). These models are parametrized and solved for a sequence of coupled, strategically positioned, user-specified, piecewise linear

floodplain cross sections. Cross sections should be roughly orthogonal to elevation contour lines, and thus approximately will follow elevation gradient lines.[2] The cross sections should partition the floodplain into regions of constant *flow regime*, which refers to the steady-state behavior of the longitudinal (as opposed to cross-sectional) water surface profile within each particular region. Three flow regimes are generally considered, each requiring an increasingly complex hydraulic model solution: (1) uniform flow (water depth and velocity are constant), (2) gradually varied flow (water depth and velocity change gradually with distance), and (3) rapidly varied flow (RVF; water depth and velocity change abruptly with distance). RVF is sometimes observed, for example, at abrupt flow path constrictions/expansions (e.g., near weirs and other partial impoundments), at hydraulic jumps (e.g., at the bases of dam spillways and waterfalls, where high flow velocity and relatively shallow depth meets low flow velocity and large depth), and at river confluences. Rivers exhibiting more frequent variations in flow regime require the specification of more cross sections for accurate representation.

With respect to the dynamic quantities upon which hydraulic models are based, the cross sections are coupled by scalar equations reflecting conservation of mass (i.e., discharge continuity, or "discharge in = discharge out"; this is a consequence of fluid incompressibility) and conservation of energy (USACE HEC 2002, p.2-2). Friction and other energy loss terms are also typically considered. The

[2] Ideally, cross sections will correspond to elevation contours on the floodwater surface to be modeled. This is because the water surface elevation is treated as a constant in the hydraulic model solution for each cross section.

more complicated, vector-based momentum equation (USACE HEC 2002, p.2-16) is used as needed, especially between cross sections bounding RVF flow regimes (USACE HEC 2002, p.1-2).

Both the energy and momentum equations originate from Newton's second law of motion (Henderson 1966, p.5). In particular, hydrodynamic equations for open channel (or free surface) flow are most generally developed using Navier-Stokes equations (Hervouet 2007, Section 2.2). Finally, the more tractable, depth-averaged version of these equations (which are known as St. Venant's equations; Hervouet 2007, Section 2.3) are widely used in 1-D open channel flow models including HEC-RAS and the U.S. National Weather Service's FLDWAV model (Fread & Lewis 1988, Fread 1993). To justify the depth averaging, fluid flow is assumed to be characterized by "long wave-shallow depth", which has the unfortunate effect that swelling is not well represented using St. Venant's equations (Hervouet 2007, p.25).

In addition to the physical equations, other empirical equations (i.e., stage-discharge relationships based on direct measurement) are also used where needed, to provide simple, reasonable approximations to expected flow behavior near structures such as weirs, culverts, and bridge supports. After all of the boundary conditions have been specified, solutions are worked upstream from cross section to cross section in the case of *subcritical* flow (i.e., where gravitational forces exceed inertial forces). Likewise, solutions are worked downstream in the case of *supercritical* flow (i.e., where inertial forces exceed gravitational forces).

After solving the system of coupled cross-section equations, a single floodwater surface elevation value for each cross section is returned by the model (Figure 3.2). Each floodwater surface elevation value is then used to estimate the floodwater extent at its respective cross section, by identifying the portion of the cross section with elevation values below the estimated water surface elevation. Graphically in $\mathbb{R}^3$, each of these cross section-specific, floodwater surface extent estimates generally will appear as a piecewise linear segment with z coordinate equal to the floodwater surface elevation. Next, the cross-sectional floodwater surface extent estimates are spatially interpolated between cross sections. The resulting, three-dimensional floodwater surface then is overlaid on the available topographic data. The portion of the estimated floodwater surface where floodwater elevation exceeds terrain elevation defines the flood extent estimate, so that portions of the estimated floodwater surface occurring below the terrain are discarded. If the user determines that the result is unsatisfactory, then the boundary conditions are amended and the model is re-evaluated. This process is repeated as needed until the user determines that the model has produced an acceptable, physically reasonable result. If the peak hydrograph value from the 100-year design storm was used for hydraulic model development, then the extent of the inundated area predicted by the model is defined to be the 100-year floodplain.

Prior to the current release, HEC-RAS modeling software was incapable of directly solving problems involving unsteady flows. "Unsteady flow" refers to cases where flow velocity and depth vary with time, such as occurs with the passage of a

flood wave through the system. Instead, users relied (and still largely do) on iterative steady flow approximations at different constant flow values, reasoning that velocity and depth changes are sufficiently slow to permit this approach. This is done to approximate unsteady flow situations like floods or dam breaches. Otherwise, most applications are only concerned with the maximum possible inundation extents, so analysis at only the peak flow is necessary.

Topographically driven, two-dimensional diffusion wave models recently have been proposed for use in flood modeling. One example is the LISFLOOD-FP model (Bates & de Roo 2000, Horritt & Bates 2001). LISFLOOD-FP uses a 1-D hydrodynamic representation of channel flow linked to a simple model for flow between spatial grid cells in the floodplain between cross sections. The JFLOW model, which has a similar design, was introduced in Bradbrook *et al.* (2004), and was further examined and described in Bradbrook *et al.* (2005) and Bradbrook (2006). These quasi-2-D hydrodynamic models are less constrained than pure 1-D hydrodynamic models, and instead better utilize the full resolution of the available topographic information. In Bradbrook (2006), the author indicates that JFLOW can help improve 1-D hydrodynamic models when the two approaches are used in tandem.

Hydrodynamic models are used to estimate the temporal evolution of floodwater extent given a set of boundary conditions characterizing a particular flood scenario. The floodplain, however, is a relatively static landscape feature, the result of the integration over time of fluvial processes that contributed to the floodplain's

formation. Consequently, there is no reason why floodplain identification should necessarily require hydrodynamic model evaluation (a similar conclusion is reached in Bates & De Roo 2000). On the other hand, though the floodplain is viewed statically in options (i) [manual delineation] and (ii) [statistical topographic detrending], both approaches are highly subjective and neither directly considers surface hydrology. In this chapter, the author introduces a physically-based, static model for floodplain identification that overcomes the problems of options (i) and (ii), yet avoids the many complexities of hydrodynamic modeling.

There is not yet any clear connection between the new model developed by the author, dubbed the FLDPLN ("floodplain") model, and dynamic open channel flow models. The FLDPLN model was specifically designed for application to discrete elevation data sampled on a regular grid, and does not, to the author's knowledge, correspond to a discretization of some prior theoretical model.

3.3. A New Method to Address User Needs

Over the past several years, the author has developed a new computational model for the purpose of estimating steady state flood (or floodplain) extent as a function of water depth. This model (FLDPLN) is referred to as "computational" because it is actually an iterative algorithm designed specifically to apply to gridded elevation data. Besides producing detailed, useful, and accurate results, the most exciting characteristics of the FLDPLN model are that it is nearly automated and has few input requirements. The FLDPLN model is a substantial extension of Jenson &

Domingue (1988), where the authors introduced conceptually simple methods for gradient approximation and stream network delineation using gridded elevation data.

This research is motivated by three distinct needs of the end-user community, for which existing methods are generally insufficient in some capacity. Foremost is the need among State agencies for rapid flood extent estimation during and following a flood event, which would greatly assist emergency response activities. There is also a need among State agencies for inexpensive estimation of potential dam breach inundation extent, for general hazard assessment. Finally, there is a need among the ecology research community for identification and detailed mapping of historic floodplain extents and the provision of a meaningful hydrologic connectivity index.

3.3.1. The Need for Rapid Flood Extent Estimation

Record rainfall events in Southeast Kansas in late June-early July of 2007 caused severe, widespread flooding that led to Federal disaster declarations in 20 counties. In addition to confirming the value of myriad geographic information system data layers and pre- and post-flooding satellite and aerial imagery, these events highlighted deficiencies in available information. One key layer that is necessary for effective disaster preparedness and response is inundation extent for a given water level. Although accurate and useful inundation extent information derived from Landsat and ASTER satellite data was produced by scientists from the Kansas Applied Remote Sensing Program (KARS) for some of the affected area, coverage was limited by the timing and location of the satellites' orbital paths. Also,

over a week had passed before satellite imagery could be acquired and processed, and that can be considered a favorable scenario given potential orbital constraints and cloud cover.

Simultaneous to the State-coordinated, image-based assessment of the flooded area, hydrologists at the United States Geological Survey (USGS) were using an *ad hoc* method based on local level-sets to estimate the flood extent[3]. Reasonable flood extent estimates were released more than a week after flood crest, but covered only select portions of the affected area. In October 2007, a detailed study supported by the USACE to analyze the flood event was released. This report contained an estimate for the flood perimeter that was based on surveyed high water marks and digital elevation contours (USACE 2007, p.4-1). To the author's knowledge, no one has attempted to model the inundation extent from this major flood event using hydrodynamic methods.

Officials from the Kansas Department of Emergency Management and The Adjutant General's Department (which oversees the Kansas National Guard) have since met with KARS scientists (including the author) and indicated that real-time flood extent estimation would greatly benefit their flood assessment and response activities.[4] Such estimates would assist State emergency response planning and

[3] Personal communication May 2008, Kevin Dobbs of KARS with Carol Mladinich of the USGS Rocky Mountain Geographic Science Center in Lakewood, CO.

[4] Toward this end, in January 2008 the Kansas GIS Policy Board funded a KARS proposal (http://da.ks.gov/gis/documents/KARS-KBS_Inundation_Proposal_18dec07.pdf) to use the FLDPLN model to create a floodplain database covering the 20 counties in southeast Kansas that received Federal disaster declarations due to extensive flooding in Summer 2007. Upon completion, this database can be used to generate real-time flood extent estimates using as little information as a single point location from the floodwater's edge. Expansion of this project is pending.

actions, allowing for more efficient allocation of resources to areas where they are most needed (such as where to direct National Guard assistance and emergency supplies). Additionally, real-time flood extent estimates can be overlaid on existing spatial population and structure data layers to allow for precise and immediate estimates of human and property impact. Such impact estimates are useful for decision-making regarding disaster response and resource allocation, as well as for determining State and Federal disaster declarations, which are based on human and monetary thresholds.

3.3.2. The Need for Inexpensive Dam Breach Inundation Modeling

Final developments for the FLDPLN model were motivated by the recent push within the Kansas Water Office (KWO) to obtain reasonable estimates for potential dam breach inundation extent for the state's many impoundments (5,784 recorded in the USACE National Inventory of Dams). Though hydrodynamic models have been extended for this purpose, KWO has funded pilot studies exploring this option and determined that a more cost-effective solution is needed. Using reservoir volume information and output from the FLDPLN model, the author has developed a simple method for detailed estimation of dam breach inundation extent. This development was possible because the model outputs are sufficient to propagate a wave front down the set of stream pixels, using downstream spatial steps as a proxy for time steps and a water volume conservation constraint.

The FLDPLN model assigns to each point in the floodplain a *flood source pixel* (FSP). The FSP for a floodplain pixel is the stream pixel from which floodwaters originate that would inundate that pixel at the shallowest floodwater depth (this is the *depth to flood*, or DTF, value that the model determines for each floodplain pixel). Floodplain pixels thus can be binned by FSP value. Using these bins and pixel-level DTF information to estimate water column volume above each pixel as a function of flood depth, a histogram can be generated for each stream pixel that shows floodplain volume as a function of flood depth. With knowledge of a reservoir's volume, and making some simple assumptions about maximum depth, wave front decay, and flood wave propagation, the author was able to develop a dam breach inundation model using the information contained in these histograms. Detailed description of this method is beyond the scope of this thesis, and will be presented in a forthcoming journal article.

3.3.3. The Need for Floodplain Mapping in Ecology Studies

While recent, final developments for the FLDPLN model were motivated by practical needs of State agencies, initial model developments were motivated by academic research needs. In 2004, Environmental Protection Agency (EPA) funding was approved for a proposed research project at the Central Plains Center for Bioassessment (CPCB)[5] to assess wetlands for water quality and biological diversity. The focus of the study was on wetlands occurring in approximately 850 km of the

[5] Like KARS, CPCB is a research subunit at the Kansas Biological Survey (KBS), which is located at the University of Kansas.

Missouri River valley between Sioux City, IA, and St. Louis, MO. Don Huggins was the Principle Investigator on this project, which is still ongoing.

For the project, CPCB scientists required a delineation of the boundary of the Missouri River valley in the study area. By spatially intersecting the bounded area with the National Wetlands Inventory database[6], CPCB researchers could then identify which wetlands should be considered for the study. Dr. Huggins was aware of the substantial difficulties and uncertainties associated with using existing methods for identifying historic floodplain extent for a study area of this large magnitude. Through prior collaboration with the author, Dr. Huggins also was aware that the author then had just completed initial developments for the FLDPLN model. Consequently, Dr. Huggins contacted the author to see if he could apply the FLDPLN model to the study area, to supply the CPCB research team with this much-needed data layer. The application was successful, and the resulting floodplain map is still being used for ongoing project studies.

Besides delineating the boundary of the historic floodplain, the CPCB project has also benefited from the physically-based flood depth values assigned to floodplain pixels during implementation of the FLDPLN model. Currently CPCB researchers are using this information as a hydrologic "connectivity index" relating floodplain locations with the main flow channel, in a manner similar to that described in Section 3.1.2.

[6] http://wetlandsfws.er.usgs.gov/nwi/

3.3.4. Some General Remarks about the New Method

Unlike traditional hydrodynamic models, FLDPLN has no dynamic, or time dependence. Rather, it is a static model driven purely by topographic data. It just so happens that a static floodplain characterized by a particular floodwater depth is not much different from the inundation extent from a sustained flood event that realizes the same floodwater depth. This observation is demonstrated in a validation study, which establishes the utility of FLDPLN for flood extent estimation. The validation study examines a major flood event from Summer 2007 in extreme eastern Kansas and western Missouri, spanning 130 km of river length.

The author has established an asymptotic convergence theorem (described in the Appendix) for the principle design feature of the model (namely, *backfill flooding*) for an idealized flow channel. The author also was able to establish a "nested floodplain" property for the full FLDPLN model. To test implementation of the full model, several examples were evaluated using FLDPLN for historic floodplain identification using real topographic data. The algorithm, the theorems, the examples, and the validation study are presented in this chapter.

3.4. Elements of the FLDPLN Model: Backfill and Spillover Flooding

The FLDPLN model is based on the assumption that the floodwater path from floodwater source point P to floodplain point Q can be characterized using two fundamental components (illustrated in Figure 3.3): *backfill flooding* and *spillover flooding*. Backfill flooding approximates floodwater swelling, and is based on the

simplistic notion that "water seeks its own level". Nearly all of the floodplain area identified in FLDPLN is specified using backfill flooding. Spillover flooding establishes new floodwater routes in the floodplain, and is based on the simplistic notion that "water flows downhill". Floodwater rerouting occurs when floodwaters breach a topographic flow divide (often a ridgeline), spilling across the divide (over the ridge) to define a new floodwater flow path.

Figures 3.4-3.6 show how backfill and spillover flooding occur at the pixel level. Backfill flooding is determined using the gradient direction field to "back into" a pixel's upstream watershed (Figure 3.4). However, without a method for defining new floodwater routes, backfill flooding hangs up whenever floodwaters encounter a flow divide. We can remedy this problem by implementing spillover flooding to breach flow divides (such as ridgelines) and create new floodwater propagation paths (Figure 3.5). If floodwaters encounter multiple flow divides, sometimes multiple spillover flooding steps are required to properly specify the new floodwater route (Figure 3.6). Figure 3.7 shows a plan view diagram illustrating instances of backfill and spillover flooding. **The strategy underlying the FLDPLN model is to backfill flood using small flood depth increments (to simulate floodwater swelling), applying spillover flooding between each step (to simulate floodwater rerouting).**

Consider the lateral floodplain cross section diagram shown in Figure 3.8. Figure 3.9(a) shows how a traditional hydrodynamic solution might appear for this cross section. Figure 3.9(b) shows how this cross section would be flooded one-

dimensionally using backfill and spillover flooding[7]. An appealing attribute of the FLDPLN model is that it is isotropic, operating independently of stream or floodplain orientation. For example, the conceptual, one-dimensional behavior of the FLDPLN model along a longitudinal floodplain cross section (Figure 3.10) is identical to its behavior along a lateral floodplain cross section (Figure 3.9(b)).

To gain a static (time independent), two-dimensional perspective of the floodplain, it is helpful to consider the notion of *potential flood extent* (PFE) for an idealized channel. Consider the *pitched channel* (surface S) shown in Figure 3.11(a). Here we see two half-planes forming a V-shaped drainage channel. To create S, two planes P1 and P2 parallel to the y-axis but such that $(\partial z/\partial x)_{P1} = 1 = -(\partial z/\partial x)_{P2}$ were used to construct horizontal channel surface $C = \max\{P1,P2\}$ (Figure 3.11(b)). Next, a plane L (the landscape plane; Figure 3.11(c)) parallel to the x-axis and such that $\partial z/\partial y = 1/2$ was added to C to construct $S = C + L$. The slope of S in the x direction defines the local topography gradient (detail scale). The slope of S in the y direction defines the landscape gradient (trend scale).

Let r be a point along the channel bottom of S. Let z_r be the elevation at r, and let h be some flood depth at r. Construct a channel-wide dam running along the two cross-sectional elevation gradient lines emanating from r, such that the top of the dam is at elevation $z_r + h$. Let the reservoir fill completely, and then consider the extent (surface area) of the reservoir formed behind the dam (Figure 3.12). The

[7] The DTF contours will largely level out when upstream and downstream stream points are also considered and FLDPLN is applied in two dimensions. This will increase the cross-sectional resemblance between FLDPLN model solutions and hydrodynamic model solutions.

reservoir boundary away from the dam will coincide with the elevation contour at $z_r + h$. The reservoir extent (surface area) provides a reasonable characterization for the upstream limit of potential swelling (backwater effects, or backfilling) realized by a flood with depth h at r. We will call this type of flooding *backfill flooding*. Using topographic data and an estimated gradient direction field, the proposed *backfill flood algorithm* (BFA) is designed to estimate the extent of the reservoir just described. The BFA is a critical component of the FLDPLN model.

Now instantly remove the dam, releasing the waters trapped behind it. Gravity-induced flow will distribute these waters laterally and downstream from r. We will call this type of flooding *spillover flooding*. In this event, the spillover flood extent will inevitably converge on the downstream drainage channel. The rate of convergence will depend on the landscape roughness and local topography gradients, but this dependence is not clearly defined. Define the *potential flood extent for point r at depth h*, or $\text{PFE}_r(h)$, to be the area surrounding r generated by taking the union of the reservoir extent and the spillover flood extent realized upon removal of the dam. See Figures 3.13(a)-(b) for possible extents for $\text{PFE}_r(h)$, using different flood depths and different rates of convergence for the spillover flood extent. The FLDPLN model, applied to r using flood depth h, provides an estimate for $\text{PFE}_r(h)$.

Suppose one can estimate the PFE for each point from a sequence of points along a stream segment R. Compute the union of the point-specific PFEs, retaining the minimum flood depth value in areas where point-specific PFEs overlap. The result is the $\text{PFE}_R(h)$, which is the *potential flood extent for segment R at depth h*.

Then $PFE_R(h)$ provides a reasonable conceptual definition for the floodplain of R associated with water depth h. See Figures 3.13(c)-(d) for possible extents for $PFE_R(h)$, using different flood depths and different rates of convergence for the point-specific, spillover flood extent estimates. The FLDPLN model, applied to R using flood depth h, provides an estimate for $PFE_R(h)$.

It is useful to provide a non-technical description of the FLPLN algorithm at this point:

i) Initialize the depth-0 floodplain to be the stream segment. Initialize flood depth $h = dh$, for some depth increment dh.

ii) Use the topography and the gradient direction field to backfill flood outward from the floodplain boundary to depth h. Add these points to the current floodplain.

iii) Locate points on the current floodplain boundary where spillover flooding will occur. Determine the "spillover flood depth" for each spillover point.

iv) Use the gradient direction field to determine new floodwater routes originating from the spillover flood points. Halt each route when it returns to the main channel downstream, or when it returns to the current floodplain, or when it reaches the study area boundary, whichever comes first.

v) Backfill flood each new floodwater route to its respective spillover flood depth.

vi) Add the newly flooded points to the current floodplain. Since these new points largely will have resulted from backfill flooding, it is possible that additional points will now be present on the floodplain boundary that require spillover flooding.

vii) Repeat steps *(iii) – (vi)* until the steady-state is reached.

viii) Increase h if necessary, and go back to step *(ii)*.

3.5. Definitions and Pixel-Level Parameters

A *raster* is rectangular array of square grid cells (pixels) to which scalar values are assigned. For example, a digital image is a raster. Each pixel has four edges and four vertices. Pixels that share an edge or vertex are called *neighboring pixels*, so that a pixel can have up to eight neighbors (four adjacent and four diagonal). A raster topographic dataset is generally referred to as a *digital elevation model*, or DEM. Pixel values in a DEM typically represent bare-Earth (i.e., ground surface) elevation, which will be the case here. The FLDPLN model specifically is designed to use DEM data. For a succinct discussion about the problems associated with digital topographic representation, see Carter (1988).

A *path* through a raster is an ordered sequence (list) of pixels such that (*i*) a pixel can only appear one time on the list (i.e., the path is not self-intersecting), and (*ii*) any two consecutive pixels in the sequence are neighboring pixels. A subset P of a raster is *connected* if for any two pixels $p_1, p_2 \in P$, there is a path between p_1 and p_2 contained in P.

Additional definitions:

- The *neighborhood* $N(p_0)$ for pixel $p_0 \in P$ (where P is a connected subset of a raster) is the set of pixels in $\{P \setminus \{p_0\}\}$ that neighbor p_0. Thus $|N(p_0)|$ can vary from 1 to 8, assuming P consists of more than one pixel.
- For subset $Z \subset P$, define the *interior boundary* $\partial_I Z$ for Z in P to be the set of pixels $p \in Z$ such that $\{N(p) \cap \{P \setminus Z\}\} \neq \varnothing$. I.e., pixels in Z that share an edge or vertex with a pixel in $P \setminus Z$ comprise the interior boundary for Z.
- For subset $Z \subset P$, define the *exterior boundary* $\partial_E Z$ for Z in P to be the set of pixels $p \in P \setminus Z$ such that $\{N(p) \cap Z\} \neq \varnothing$. I.e., pixels in $P \setminus Z$ that share an edge or vertex with a pixel in Z comprise the exterior boundary for Z.

3.5.1. Existing Hydrologic Pixel-Level Parameters

In Jenson & Domingue (1988), the authors describe methods for gradient estimation and stream network identification based solely on pixel-level calculations using a DEM. First, they introduce the concept of a *filled*, or *depressionless*, DEM, whereby all depressions (sinks) of the DEM are identified, and all of the pixel elevation values from each sink are replaced with the elevation value of the lowest pixel immediately bordering (sharing an edge or vertex with) the sink (i.e., sinks are

filled to the elevation of their *spill point*). See Figure 3.14 for an example showing a DEM before filling and after filling.

Sink filling is a deterministic operation that eliminates all strict local minima from the DEM with the least manipulation of DEM values. This operation simplifies the analysis of global surface flow. In surface flow analyses, these sinks serve as static storage areas. Until they overtop, such features typically do not factor into the spatial dynamics of traditional flow propagation models such as HEC-RAS. Thus the fill operation is assumed to have little impact on global surface flow analysis. Unless specifically noted, all future references to DEMs and DEM values used in this paper refer to the filled DEM. The elevation for a pixel p in a filled DEM will be denoted E(p).

Using the filled DEM, the following two pixel-level parameters are then introduced in Jenson & Domingue (1988): *flow direction* and *flow accumulation*. The *flow direction* for a pixel p in a DEM provides an estimate for the topographic gradient direction at p, if we associate the gradient with the direction of maximum descent. The *flow accumulation* for a pixel p in a DEM provides an estimate for the size of the catchment for p contained within the DEM. The *catchment* (or *watershed*) for p is the area upstream from p from which surface waters drain through p. More detailed descriptions for flow direction and flow accumulation are warranted.

FLOW DIRECTION—This is the direction of the minimum local directional derivative among the eight simple local directional derivatives that can be estimated using a pixel's eight neighboring cells (four adjacent, four diagonal) and the

difference quotient $\Delta E/\Delta x$. Suppose that p_0 is the pixel of interest, and pixel p is one of its neighbors. Then $\Delta E = E(p) - E(p_0)$, and $\Delta x = 1$ if p_0 and p are adjacent, or $\Delta x = \sqrt{2}$ if they are diagonal. Logical decision rules apply in the event of a tie or if a flat region (such as a lake surface) is encountered, so that each pixel is assigned a single flow direction. In a flat area, all flow is routed toward the spill point(s) of the flat area. The resulting *flow direction map*, or FDR, is a raster dataset that provides a discrete approximation to the gradient direction field, if we associate the gradient with the direction of maximum descent. Figure 3.15 shows an example DEM, and Figure 3.16 shows the FDR for this DEM. The Mud Creek study area depicted in this series of graphics is a subset from a larger area that was actually processed, so that edge effects of the DEM processing can be ignored throughout the discussion.

Using pixel-to-pixel movements indicated by the FDR, every pixel obtains a unique, non-increasing (in elevation) *exit path*, or *trajectory*, out of the study area. Exit paths will necessarily be non-increasing (in elevation) as a consequence of using the filled DEM, which ensures that no uphill flow directions need be specified because all strict local minima have been removed from the original DEM.

FLOW ACCUMULATION—Based on the FDR, a pixel's flow accumulation value is the number of pixels with exit paths that pass through the pixel. The flow accumulation value is thus proportional to the size of a pixel's upstream watershed (catchment) within the study area. Pixels with large flow accumulation values occur at drainage channel bottoms of the DEM, generally coinciding with actual in-stream locations on the Earth's surface. Thus simple thresholding of the resulting raster *flow*

accumulation map, or FAC, can produce a reasonably accurate delineation of the study area drainage channel network. To "threshold", one identifies the set of pixels with FAC values greater than some threshold value specified by the user. The set of pixels identified through FAC thresholding is called the *synthetic stream network* (or just *stream network*).

Figure 3.17 shows the FAC derived using the FDR shown in Figure 3.16. Figure 3.18 shows the stream network obtained from the FAC, using a threshold of 10^5 pixels. To facilitate display, the stream network is shown as a thickened polyline (i.e., a set of connected straight line segments) approximating the actual pixelated stream network. Also indicated in Figure 3.18 is the Mud Creek segment (shown in blue) from the stream network, which will be used in later examples. For an examination of the effects of DEM resolution and accuracy on drainage area and runoff volume estimation, see Kenward *et al.* (2000). DEM filling and calculation of the FDR and FAC can all be achieved using existing software. In particular, the author used the ArcHydro extension[8] (Maidment 2002) for ESRI ArcMap 9.2. All additional data processing required for the proposed algorithms was coded by the author using MATLAB, because no existing software has these capabilities in-built.

With the introduction of the FDR and the concept of an "exit path", two additional definitions can now be described:

- The *trajectory* $T(p_0)$ for pixel $p_0 \in P$ is the ordered set of pixels in P along the exit path for p_0, which is uniquely determined using movements dictated by the FDR.

[8] http://support.esri.com/index.cfm?fa=downloads.dataModels.filteredGateway&dmid=15

A subspan of a trajectory is called a *segment*, or *stream segment* if the trajectory is part of a stream network.

- The *depth h backfill watershed* $B(p_0,h)$ for pixel p_0 is the set of pixels $p \in P$ such that $p_0 \in T(p)$ and p is such that elevation $E(p) \leq E(p_0) + h$, for some $h > 0$.

  Denote this set by $B(p_0,h)$. Because all paths are non-increasing in elevation, $B(p_0,h)$ will be connected. $B(p_0,h)$ values are used in the backfill flood algorithm (BFA).

3.5.2. Proposed Hydrologic Pixel-Level Parameters

Building from the work of Jenson & Domingue (1988), the author proposes two new pixel-level parameters related to floodplain mapping and analysis. The first parameter is *depth to flood* (DTF), which specifies the minimum flood depth required to inundate a floodplain pixel with floodwater originating from the stream segment (or stream network) in question. The second parameter is *flood source pixel* (FSP), which specifies the stream pixel from which floodwaters can originate that inundate the floodplain pixel at that minimum flood depth. Pixel-level DTF and FSP values are determined computationally using the FLDPLN model, and thus are described only vaguely here.

The DTF map essentially provides a replica of floodplain topography, but with the stream surface slope removed using hydrologic surface flow information. The DTF map is distinguished from the local statistical models described in Section

3.1.1 in that DTF values are dictated by hydrologic connectivity (as expressed in the DEM and FDR) rather than by using a purely statistical spatial extrapolation of stream elevation for detrending. Because of this attribute, FSP and DTF information allows for the simulation of flood wave propagation by using progression down an ordered set of FSPs as a proxy for time progression. This concept, used in conjunction with release volume, underlies the author's dam breach inundation model that was described in Section 3.3.2.

To compute DTF and FSP values, we introduce the *floodplain algorithm* as the basis for the FLDPLN model. To implement the algorithm, the only requirements are a DEM, a list of stream pixels for which the floodplain is sought, a maximum flood depth value (h), and one free parameter (dh). Modularity of the algorithm is demonstrated in the validation study, where three different h values are used to seamlessly flood three different (but connected) stream segments to different depths. As noted above, the FDR is also required, but this is determined from the DEM. The FAC (determined from the FDR) is required only for identifying stream network pixels. The maximum flood depth value h is provided by the user or can be determined using observed water surface elevation data (such as recorded during or predicted for a flood event).

As will be described, the free parameter dh is necessary to allow for warranted ridge violations by floodwaters. Ridges in the DEM can present false discontinuities in the DTF map if not properly addressed. The magnitude of dh, which satisfies $0<dh\leq h$, affects the model output. If dh is too large, this can result in more and larger

erroneous discontinuities in the DTF map (i.e., discontinuities in excess of DEM discontinuity). If *dh* is too small, then this increases the chance that too much spillover flooding will occur downstream from the stream pixels under study, which may not accurately reflect downstream flooding characteristics. This can undermine the real-time use of the model's output for flood extent estimation.

3.6. The Backfill Flood Algorithm (BFA)

The first algorithm is the *backfill flood algorithm* (BFA). The BFA requires the filled DEM, the FDR, and a set of flood source pixels $X = \{x_j\}$ to backfill flood. X can be a stream segment or network, or it can be any set of pixels. Each x_j is assigned a maximum flood height h_j, stored in $H = \{h_j\}$. By design, upon algorithm completion, we will have $\text{FSP}(p) \in X$ for all pixels p in the identified backfill floodplain Z. For all $p_{jk} \in Z$ such that $\text{FSP}(p_{jk}) = x_j$, we will have $0 \leq \text{DTF}(p_{jk}) \leq h_j$. Here is the precise specification of the BFA:

1) Determine the backfill watershed $B_j = B(x_j,h_j)$ for each pixel $x_j \in X$.

2) For each pixel $p_{jk} \in B_j$, for all j, define $\text{DTF}_j(p_{jk}) := \text{E}(p_{jk}) - \text{E}(x_j)$ and $\text{FSP}_j(p_{jk}) := x_j$.

3) Define $Z(X,H) := \cup B_j$. For each pixel $p \in Z$, define $DTF_Z(p) := \min_k \{DTF_k(p) \mid p \in B_k\}$. Also define $FSP_Z(p) := \{x_j \in X \mid DTF_j(p) = \min_k \{DTF_k(p) \mid p \in B_k\}\}$. Because B_k's determined for x_k's that do not occur on a single trajectory will necessarily be disjoint, there can be more than one such x_j satisfying the condition in the definition for $FSP_Z(p)$ only if those x_j's occur along a single trajectory. In this case, define $FSP_Z(p)$ to be the x_j that is most upstream in the trajectory.

$Z(X,H)$ is the *backfill floodplain* for pixel set X using corresponding maximum flood depths from H. Each pixel in Z has a DTF and FSP value, determined in Step 3 of the BFA. If a uniform flood height h is used for all pixels in X, then denote the backfill floodplain by $Z(X,h)$.

The BFA is a deterministic computational model that provides the engine for the FLDPLN model. In the FLDPLN model, X will consist of either a trajectory or a trajectory subspan, or X will be the interior boundary for some intermediate floodplain determined in the course of determining the final floodplain.

3.6.1. Violating the Gradient: The Need for Spillover Flooding

Using a maximum flood depth of 10 m, the BFA was applied to all pixels along the Mud Creek stream segment in the larger Mud Creek study. The 10-m backfill floodplain DTF map is depicted in Figure 3.19, for the same study area subset used in Figures 3.15-3.18. Several erroneous DTF discontinuities (i.e., discontinuity

in excess of underlying DEM discontinuity) are visible. Two of the most severe DTF discontinuities are highlighted, the top one also resulting in a notable floodplain underestimation. Paralleling of tributary-specific watersheds with the main flow channel is generally the source of this problem. To illustrate this point, trajectories to Mud Creek are shown in Figure 3.19 for sample floodplain pixels on each side of the two featured flow divides. The reason for the problem is diagramed in Figure 3.7. If both backfill and spillover flooding are used, point Q_1 apparently will be inundated by floodwaters originating from the upper flood source point at a lower flood depth than if just backfill flooding is used to inundate Q_1 with floodwaters originating from the lower flood source point.

DTF discontinuity problems arise due to flow divides in the FDR, which generally correspond to actual ridgelines in the DEM. Ridgelines, which need not be greatly pronounced with respect to landscape relief, correspond to watershed boundaries at some watershed scale. Flow divides present flow barriers when backfill flooding using the FDR. Fortunately, a simple (though computationally intensive) solution exists to largely fix this problem. Specifically, we allow for possible *spillover flooding* to occur on the portions of the backfill boundary where pixel DTF values are less than the maximum flood depth, and apply this procedure along with the BFA in an iterative algorithm. Using 0.5-m BFA iterations and accounting for spillover flooding on the floodplain boundary between iterations, we obtain the steady-state floodplain DTF map shown in Figure 3.20. This map was created using the FLDPLN model.

3.7. The Floodplain Algorithm (the FLDPLN Model)

The *floodplain algorithm* requires the DEM, the FDR, a set of stream pixels R to serve as possible FSPs, and a maximum flood depth h. As previously noted, the BFA provides the engine for the FLDPLN model. The critical additional feature of FLDPLN is the use of spillover flooding, whereby floodwaters are allowed to breach ridges in the FDR whenever the flood depth necessary for this to occur is exceeded. The algorithm is made iterative so that spillover flooding effects are properly modeled at increasing flood depths. Thus successive, nested floodplain extents are estimated using increasing flood depth values until the maximum flood depth h is reached.

Due to its iterative design, FLDPLN requires a user-specified flood depth increment value $dh \leq h$, which is the only free parameter required by the algorithm. With this iterative strategy, the magnitude of erroneous discontinuities in the final floodplain DTF map cannot exceed dh. Parameter h is also user-determined and thus "free" in some sense, but variations in the output from the algorithm essentially can be caused only by the choice for dh. For example, the floodplain determined by FLDPLN using maximum flood depth = h will be the identical to the floodplain determined using maximum flood depth = $2h$ after discarding all points with flood depth > h, if a single dh value commensurable with h (i.e., $h = c \cdot dh$, for some $c \in \mathbb{Z}^+$) is used for both model implementations.

Initialize starting flood depth $h_0 = dh$, and initialize total floodplain $F = R$ for some stream segment (trajectory or trajectory subspan) R. Also, initialize floodplain interior boundary $X = \partial_I F = R$. Let $T_D(R) = T(R) \backslash R$ denote the set of pixels in the study area comprising the downstream trajectory of R, but excluding pixels in R. $T_D(R)$ represents the "main channel" downstream from R, and is used to reasonably constrain spillover flooding if R does not extend to the study area boundary. Every pixel in F has two attributes, DTF and FSP. To start, define DTF(p) := 0 and FSP(p) := p for all $p \in F$ (= R). DTF provides an estimate of the minimum local flood depth required to inundate a particular floodplain pixel, and FSP indicates the stream pixel in R from which floodwaters can originate capable of inundating that floodplain pixel at the minimum flood depth. Subscripts on DTF and FSP indicate values associated with particular intermediate floodplain evaluations, some of which must be temporarily defined prior to assimilation into the total floodplain $F = F(R,h,dh)$. DTF and FSP values with no subscript are associated with the total floodplain F. By design, upon algorithm completion, we will have $0 \leq \text{DTF}(p) \leq h$ and $\text{FSP}(p) \in R$ for all pixels $p \in F$.

The following is a precise, step-by-step description of the floodplain algorithm. With the exception of h, h_0, and dh, lower case variables refer to individual pixels. Also, FSP(·) refers to individual pixels. Upper case variables refer to sets of pixels. In addition to DTF(·), Greek letters refer to computed scalar values.

1) **[*Apply the BFA to interior boundary pixels X of F*]** Determine the temporary backfill floodplain $Z = Z(X,H)$, where $H = h_0 - \text{DTF}(X) = \{h_0 - \text{DTF}(x_1), h_0 - \text{DTF}(x_2),\ldots\}$. Redefine $Z := Z \setminus \{Z \cap F\}$.

NOTES: Due to algorithm design, backfill floodplain pixels determined in this step that already occur in F will produce assimilated DTF values (determined in Step 2) not less than the DTF value that is already assigned to them in F. Consequently, we can immediately exclude pixels in $\{Z \cap F\}$ from consideration at this point.

2) **[*Assimilate backfill floodplain Z into F*]** For all pixels $p \in Z$, define DTF and FSP values such that $\text{DTF}(p) := \text{DTF}_Z(p) + \text{DTF}(\text{FSP}_Z(p))$ and $\text{FSP}(p) := \text{FSP}(\text{FSP}_Z(p))$. Redefine $F := F \cup Z$.

NOTES: DTF(p) is the sum of the backfill flood depth for p determined in Step 1 plus the previously determined flood depth for the interior boundary point of F that provided the source of floodwaters that inundated p using the BFA. FSP(p) is the pixel in R from which floodwaters originated that inundated the interior boundary point of F that was identified as $\text{FSP}_Z(p)$ in Step 1.

3) **[*Identify interior and exterior boundary pixels for F*]** Identify the current floodplain interior boundary pixels $x_j \in X = \partial_I F$ and exterior boundary pixels $y_k \in Y = \partial_E F \setminus T_D(R)$.

NOTES: $T_D(R)$ is excluded from the exterior boundary set Y so that spillover flooding cannot apply directly to points in the main channel downstream from R. This restriction provides a reasonable, practical constraint for spillover flooding.

4) **[*Identify exterior boundary pixels in Y where spillover flooding will occur*]** For each $y_k \in Y$, determine if there are neighboring interior boundary pixels $w_{ki} \in X \cap N(y_k)$ such that $E(y_k) \leq E(w_{ki}) + h_0 - DTF(w_{ki})$. Redefine $Y := \{y_k \mid \{w_{ki}\} \neq \varnothing\}$. Define scalar values $v_{ki} := E(w_{ki}) + h_0 - DTF(w_{ki}) - E(y_k)$.

NOTES: The maximum v_{ki} value will be identified in Step 5 to determine which of the w_{ki} provides the best option for spillover flooding. The expression for v_{ki} will favor w_{ki} that provide the maximum spillover depth. If more than one w_{ki} have spillover depth equal to the maximum available spillover depth $h_0 - DTF(w_{ki})$ (which can occur whenever $E(y_k) \leq E(w_{ki})$, characterizing downhill

spillover; see Figure 3.5(a)), then the expression for υ_{ki} will favor the w_{ki} with the largest elevation value, and thus the largest elevation drop between w_{ki} and y_k.

5) **[*Determine the spillover flood depth for each pixel in Y*]** For each $y_k \in Y$, define $w_k := \{w_{kg} : \upsilon_{kg} = \max_i \{\upsilon_{ki}\}\}_1$ (NOTE: the subscript "1" indicates to take the (random) first element of this set, in case multiple w_{kg} meet this criterion). Define $\varphi_k := h_0 - \text{DTF}(w_k) - \max\{0, \text{E}(y_k) - \text{E}(w_k)\}$. φ_k is the *spillover flood depth* for y_k.

NOTES: If $\text{E}(y_k) \leq \text{E}(w_k)$, then the spillover flood depth φ_k equals its maximum possible value $h_0 - \text{DTF}(w_k)$ ("downhill spillover"; e.g., see Figure 3.5(a)). Otherwise, if $\text{E}(y_k) > \text{E}(w_k)$, then the difference $\text{E}(y_k) - \text{E}(w_k)$ is subtracted from $h_0 - \text{DTF}(w_k)$ to obtain the spillover flood depth because an additional flood depth of $\text{E}(y_k) - \text{E}(w_k)$ is required for floodwaters from w_k to reach y_k ("uphill spillover"; e.g., see Figure 3.5(b)).

6) **[*Sort Y in a suitable manner to facilitate spillover flooding*]** Sort the y_k so that they are decreasing in spillover flood depth φ_k. If necessary, perform a secondary sort so that y_k's that have identical φ_k values are decreasing in elevation. With this operation, spillover flooding will occur first for pixels in Y with the largest spillover depth, and which are higher up on the landscape for pixels in Y that have the same spillover depth. This assures that spillover flooding of pixels in Y will occur in a logical progression that will help limit redundant processing.

For each $y_k \in Y$:

a. **[*Determine the trajectory for y_k*]** Determine $T(y_k)$. Halt the growth of $T(y_k)$ if a pixel $p \in T(y_k)$ is encountered such that either (1) $p \in F$ and $\text{DTF}(p) \leq h_0 - \varphi_k = \text{DTF}(w_k) + \max\{0, \text{E}(y_k) - \text{E}(w_k)\}$ (i.e., a floodplain pixel is encountered that has a DTF value less than or equal to the DTF value that would be assigned during this spillover flooding operation), or (2) $p \in T_D(R)$ (i.e., a pixel is encountered that is in the main channel downstream from R). If criterion (1) is first satisfied, then let the final pixel included in $T(y_k)$ be the pixel just upstream from p. If criterion (2) is first satisfied, then let p be the final pixel included in $T(y_k)$. If neither criteria is met, then use the complete trajectory for $T(y_k)$.

   NOTES: Criterion (1) is imposed to limit redundant processing. Criterion (2) occurs when the new floodwater route returns to the main channel downstream from R.

b. **[*Apply the BFA to $T(y_k)$*]** Determine the temporary backfill floodplain $Z = Z(T(y_k),\varphi_k)$.

c. **[*Prepare Z for assimilation into F*]** For all pixels $p \in Z$, redefine $DTF_Z(p) := DTF_Z(p) + h_0 - \varphi_k$ and $FSP_Z(p) := FSP(w_k)$.

NOTES: $DTF_Z(p)$ is now the sum of the backfill flood depth for p determined in Step 6(b) plus the flood depth required to produce spillover at y_k. $FSP_Z(p)$ is now the pixel in R from which floodwaters originated to produce spillover at y_k.

d. **[*Overwrite existing floodplain pixels in F as necessary*]** For all pixels $p \in Z \cap F$, if $DTF_Z(p) < DTF(p)$, then redefine $DTF(p) := DTF_Z(p)$ and $FSP(p) := FSP_Z(p)$.

e. **[*Assimilate new floodplain pixels into F*]** Redefine $F := F \cup \{Z \setminus \{Z \cap F\}\}$. For all new floodplain pixels $p \in \{Z \setminus \{Z \cap F\}\}$, define $DTF(p) := DTF_Z(p)$ and $FSP(p) := FSP_Z(p)$.

7) **[OPTIONAL STEP: *Perform spillover flooding until F converges*]** Repeat Steps 3-6 until no new pixels are added to F.

NOTES: Convergence of F is assured because either (*i*) the presence of downstream tributaries flowing into the main channel will inhibit additional spillover flooding (i.e., the current flood depth will be insufficient to spill over the ridge on the downstream side of some downstream tributary channel), or (*ii*) the study area boundary will be reached. Because of this convergence, when Step 7 is used in the FLDPLN model, the output will be referred to as the *steady state floodplain*.

Step 7 is optional for coarse resolution ($\geq$ 30 m pixel size, say) DEM data. This is because once floodwaters reach a pixel, the pixel is presumed to be 100% inundated. For higher resolution DEM data ($\leq$ 10 m pixel size, say), this assumption is increasingly likely to hold. In addition to representing smaller areas, variations in sub-pixel elevation values generally will become small in magnitude as pixel size gets smaller. Thus smaller pixel size reduces the likelihood that individual pixels will contain hidden ridgelines that inhibit spillover. However, for larger pixels, this may not be the case, and the risk of excess, erroneous spillover flooding will increase with pixel size if Steps 3-6 are repeated until F converges.

The possible negative effects of not using Step 7 are that (*i*) the floodplain extent may be underestimated, and (*ii*) the resulting floodplain DTF map may exhibit erroneous discontinuities in excess of dh. However, these factors may be offset

by the risk of overestimating the floodplain extent when using Step 7 with coarse resolution DEM data. Consequently, the user must consider the limitations of their data when deciding whether or not to apply Step 7.

8) **[*Identify interior boundary pixels for F*]** If $h_0 < h$, redefine $h_0 := h_0 + \min\{dh, h - h_0\}$, identify the floodplain interior boundary $X = \partial_I F$, and go back to Step 1.

Otherwise, if $h_0 = h$, then exit the algorithm.

When computed using the same *dh* value, steady-state floodplains determined using FLDPLN have the logical property of being nested. This is described in the following theorem.

THEOREM 3.1: [*Steady state floodplains produced by FLDPLN are nested*] Let R be a segment, and fix $c \in \mathbb{Z}^+$ and $dh \in \mathbb{R}^+$. Compute steady state floodplains $F1 = F(R, cdh, dh)$ and $F2 = F(R, (c+1)dh, dh)$ using the FLDPLN model. Define floodplain subset $G2 := \{p \in F2 \mid \mathrm{DTF}_{F2}(p) \leq cdh\}$. Then $G2 = F1$, and $\mathrm{DTF}_{G2}(p) = \mathrm{DTF}_{F1}(p)$ and $\mathrm{FSP}_{G2}(p) = \mathrm{FSP}_{F1}(p)$.

PROOF: *F1* is the output from the first iteration of FLDPLN when determining *F2*. Clearly we will have $F1 \subset G2$, because assigned DTF values never increase in future

iterations of algorithm. We need to establish two properties: (*i*) $G2 \subset F1$ (which will establish that $G2 = F1$); and (*ii*) $\mathrm{DTF}_{G2}(p) = \mathrm{DTF}_{F1}(p)$ for all pixels $p \in G2$.

(i) Due to the "steady state" design of the algorithm, all pixels $q \in \partial_e F1$ will necessarily require a flood depth > *cdh* to be flooded using backfill or spillover flooding from pixels in $\partial_i F1$.

Define $F21 := F2 \backslash F1$, which is the subset of *F2* not included in *F1*. Suppose $G2 \not\subset F1$, so that there exists a pixel $p_0 \in F21$ (i.e., $p_0 \in F2$ and $p_0 \notin F1$) such that $\mathrm{DTF}_{F2}(p_0) = c_0 \leq cdh$. Let $q_0 \in \partial_e F1$ be the pixel on the exterior boundary of *F1* that served as the intermediate flood source pixel for p_0 during the calculation of *F2*. Backfill and spillover flooding processes used in FLDPLN can never assign to a floodplain pixel a DTF value lower than the DTF value from the input pixel acting as the floodwater source. Consequently, we must have $\mathrm{DTF}_{F2}(q_0) \leq c_0 \leq cdh$. This contradicts the above observation that all pixels in $\partial_e F1$ require a flood depth > *cdh*. Therefore no such p_0 can exist, and we have $G2 \subseteq F1$.

(ii) From property (*i*), it follows that $\mathrm{DTF}_{F2}(p) > cdh$ for all $p \in F21$. Because neither backfill nor spillover flooding of pixels in *F21* can result in DTF

values < cdh, it follows that no pixels in *F1* (which all have DTF ≤ cdh) will be overwritten during the iteration with maximum flood depth (c+1)dh that determines *F2*. Therefore $DTF_{G2}(p) = DTF_{F1}(p)$ and $FSP_{G2}(p) = FSP_{F1}(p)$ for all pixels $p \in G2$. QED

3.7.1. Sensitivity of the FLDPLN Model to Free Parameter 'dh'

Here we show the effects of using different *dh* values in the FLDPLN model with two examples. We first look at the effect on the floodplain estimated for a single FSP, and then on an actual flood extent estimate using a set of stream segments. The first example uses units in meters, and the second example uses meters and feet (conversion: 1 ft = 0.3048 m).

Figure 3.21(a)-(b) show, respectively, the floodplains determined using FLDPLN with (h,dh) = (5,5) and (h,dh) = (5,1), applied to a single FSP from the Mud Creek stream segment. The floodplains are depicted as DTF maps. Imagine dividing these floodplains into backfill and spillover areas using the cross section line through the FSP. Note the similarity between the backfill areas to the north of the cross section, and the difference between the spillover areas to the south.

Note the resemblance between this actual example and the conceptual PFE's shown in Figure 3.13(a)-(b). This example illustrates how the choice for *dh* can greatly affect the spillover area while exerting almost no influence on the backfill area. Of course, using $dh = h$ (as in Figure 3.21(a)) is an extreme case (recall that $0<dh\leq h$). In practice, *dh* typically would be made reasonably small to limit spatial

discontinuity in DTF values. The next example compares flood extent estimates generated using FLDPLN with two relatively small values for *dh*.

In late June-early July 2007, a major flood event occurred along the Verdigris River in southeast Kansas. The portion of the Verdigris River that we will examine runs through Montgomery County, KS, south into Oklahoma. In particular, consider the river reach between Independence, KS, and Coffeyville, KS, shown in Figure 3.22. To add context, an oil refinery on the northeast edge of Coffeyville was the site of a major oil spill that occurred during this flood event. DEM data with 10-m spatial resolution were acquired from the National Elevation Dataset (NED, http://ned.usgs.gov/; Gesch *et al.* 2002) and used for the analysis.

Prior to examining flood extent estimates, we call attention to the span of the Verdigris River on which the National Weather Service (NWS) stream gage is positioned (the circled area in Figure 3.22). Note that the DEM-derived channel does not follow the Verdigris River in this area, but instead flows east into Big Hill Creek, following an actual pre-confluence fluvial connection between the Verdigris River and this tributary. This connection is along the upper right edge of the circled area. The DEM-derived channel then rejoins with the Verdigris just north of Coffeyville, at the confluence between Big Hill Creek and the Verdigris River (lower right quadrant of the circled area). This large channel placement discrepancy is not a processing error, but rather is the result of a single flow direction value redirecting the Verdigris River flow into the pre-confluence connection with Big Hill Creek. Fortunately, the FLDPLN model is robust to such mistakes. Because the FLDPLN model utilizes

hydrologic connectivity information, a complete floodwater route was identified through the actual Verdigris channel using a flood depth of only 1.01 ft (0.308-m). To illustrate this, the 2-ft floodplain generated using the DEM-derived channel is shown in Figures 3.22 and 3.23, though it is obscured largely by the DEM-derived channel where the two coincide.

Figure 3.23 shows the study area as it appeared in an ASTER satellite image acquired on July 7, 2007, which was five days after flood crest in the area. Though floodwaters had substantially receded by this date, most of the peak floodwater "footprint" is still visible in this image. The image is shown as a false-color composite, using a combination of color and infrared bands that maximize the contrast between wet and dry areas[9]. The peak floodwater surface elevation recorded during this flood event is shown (as a DTF value) for each of the three regularly monitored stream gages in the study area. Note that the peak flood DTF value declines more than 3 m between the USGS gage at Independence and the NWS gage at Coffeyville, indicating the presence of different flow regimes between these gaging stations.

During the stream network delineation phase of the DEM pre-processing, the Verdigris River reach between Independence and Coffeyville was partitioned into 10 stream segments. For each of these 10 segments, two peak flood extent estimates were generated using the steady-state FLDPLN model, one using dh = 1 m and one

[9] In the ASTER image, red areas correspond with healthy vegetation. Flood-damaged vegetation has a general blue-gray appearance. Most of the healthy vegetation visible within the apparent flood extent corresponds with unaffected tree canopies.

using dh = 0.3048 m (1 ft). The maximum flood depth for each segment was determined using the peak gage height values recorded during the flood. Each segment harboring a gaging station was flooded to that station's peak DTF value shown in Figure 3.23. These peak DTF values were linearly interpolated to determine maximum DTF values for segments in between gaging stations. Two area-wide flood extent estimates (corresponding to dh = 1 m and dh = 1 ft) were then generated by taking the union of the segment-specific flood extent estimates. The boundaries for these two area-wide estimates are shown in Figure 3.24.

The two flood extent estimates are, for all practical purposes, nearly indistinguishable. This outcome demonstrates the robustness (stability) of the FLDPLN model to moderately-sized variations in dh. The largest discrepancy in the study area occurs within the city limits of Coffeyville (Figure 3.25). A USACE floodwater extent estimate[10] is shown as a reference, illustrating that either of the two FLDPLN flood extent estimates is reasonable in this area. This example for the Verdigris River illustrates how output from the FLDPLN model can be used with stream gage data for real-time flood extent estimation, the need for which was described in Section 3.3.1.

[10] The USACE (USACE 2007, p.4-1) estimate was released in October 2007, months after the event. This estimate was created using records of high water marks and spatial interpolation. Interestingly, no extent estimate has been produced for this catastrophic flood event using hydrodynamic models.

3.7.2. Examples Using the FLDPLN Model

It is helpful to see some examples demonstrating the capability of the FLDPLN model for historic floodplain identification. Two examples are provided in which higher resolution DEM data are used to determine the steady-state floodplain. Two other examples are provided in which lower resolution DEM data are used to determine the floodplain, but without employing optional Step 7. To the author's knowledge, no one has ever attempted such comprehensive, detailed floodplain identification and mapping for rivers of large magnitude like those considered in the last three examples.

In the first example, FLDPLN was used to determine the 10-m, steady-state floodplain for approximately 10 km of the Mud Creek stream reach between Lake Dabinawa in Jefferson County, KS, almost to the point where Mud Creek enters the Kansas River Valley. 2-m resolution LIDAR DEM data (obtained from http://www.kansasgis.org/) were used to estimate the floodplain. The FLDPLN model was applied using $(h,dh) = (10,0.5)$. The result is shown in Figure 3.26.

In the second example, FLDPLN was used to determine the 10-m, steady-state floodplain for approximately 100 km of continuous stream reach beginning with the Big Blue River below Tuttle Creek Lake in northeast Kansas. The Big Blue River empties into the Kansas River, which comprises the remainder of the examined stream reach. 10-m resolution DEM data from the NED were used to estimate the floodplain. The FLDPLN model was applied using $(h,dh) = (10,1)$. The result is shown in Figure 3.27.

In the third example, FLDPLN was used to determine the 16-m floodplain for approximately 500 km of the Missouri River in the central U.S. 30-m resolution DEM data from the NED were used to estimate the floodplain. The FLDPLN model was applied using (h,dh) = (16,2), without using optional Step 7. Only the boundary of the resulting floodplain is shown in Figure 3.28, to more clearly demonstrate the ability of the FLDPLN model for historic floodplain (river valley) identification.

In the fourth example, FLDPLN was used to determine the 25-m floodplain for approximately 1700 km of the Amazon River in Brazil. 90-m resolution DEM data from the NED were used to estimate the floodplain. The FLDPLN model was applied using (h,dh) = (25,5), without using optional Step 7. The resulting floodplain is shown in Figure 3.29. In addition to being the largest capacity river in the world, the Amazon River also has one of the most complex floodplains due to the extremely low grade of the stream course and the regular occurrence of major, annual flood events.

3.8. Validation Study

The previous examples clearly demonstrate that the FLDPLN model can be used for identification of historic floodplains, examining a variety of rivers using different resolution elevation datasets. The question remains regarding whether or not output from the model can be used for actual flood extent estimation, i.e., whether or not DTF values produced by the model are accurate. The Verdigris River example presented earlier provided some qualitative evidence in support of this assertion, but a

quantitative study would be more convincing. Toward this end, we examine a recent flood event that occurred in early July 2007 in extreme eastern Kansas and western Missouri. This flood produced record or near-record flood depths in the study area, with near-crest values persisting for 2-4 days. The flood extent (flood footprint) was visible in a Landsat-5 scene (which has 30-m resolution) from 7/7/07, even though this was 3-6 days after crest. Computations were performed for a region larger than the study area, so that the study area extraction did not suffer edge effects. 30-m DEM data were used in the analysis, obtained on August 15, 2007, from the NED. Considering the 30-m resolution of the DEM, the FLDPLN model was implemented without using Step 7.

3.8.1. Study Area

Ignoring small-scale meanders, the study area is spanned by a 50 km segment of the Marais des Cygnes River, a 20 km segment of the Little Osage River, and a 60 km segment of the Osage River. The Osage River begins at the confluence of the Marais des Cygnes and the Little Osage. The upstream boundary of the study area is determined by gaging station locations on the Marais des Cygnes and the Little Osage, and the downstream boundary is approximately 1 km west of the entry of the Osage River into Harry S. Truman Reservoir (see Figure 3.30). Background imagery used in Figure 3.30 reflects non-flood conditions. These data, which are shown in true color and have 1-m spatial resolution, are from the 2006 National Agricultural

Imagery Program (NAIP), and were obtained from http://www.kansasgis.org/ (for Kansas imagery) and http://www.msdis.missouri.edu/ (for Missouri imagery).

Processing a region larger than the study area and using a flow accumulation value of 200,000 pixels, points on the Marais des Cygnes and Little Osage Rivers upstream from the study area were identified, and then propagated through the study area to provide the network of stream pixels that served as FSPs. Again, see Figure 3.30.

3.8.2. Gage #1 (Marais des Cygnes River)

Gaging station #1 (USGS 06916600, Trading Post, KS) is located on the Marais des Cygnes River at (37.2225 N, 94.6678 W). Gage information was obtained from http://waterdata.usgs.gov/mo/nwis/uv?site_no=06916600. Dr. Don Huggins obtained provisional gage height data from the USGS via personal correspondence in September 2007.

The datum (reference elevation) for gage #1 is 230.75 m above sea level. Floodwater crest with respect to the mean daily gage height was 12.12 m on 7/2/07, indicating a water surface elevation of 242.87 m.

The value of the 30-m filled DEM at the pixel where the gaging station is located is 237.69 m. This pixel is adjacent to a stream pixel, which has elevation 233.29 m in the filled DEM. Thus the expected optimal floodplain determined for this location should have a maximum DTF value of 242.87 – 233.29 = 9.58 m.

3.8.3. Gage #2 (Little Osage River)

Gaging station #2 (USGS 06917060, Horton, MO) is located on the Little Osage River at (37.9948 N, 94.3693 W). Gage information and provisional gage height data were retrieved in September 2007 from the USGS's website at http://waterdata.usgs.gov/mo/nwis/uv?site_no=06917060.

The datum for gage #2 is 213.36 m above sea level. Floodwater crest with respect to the mean daily gage height was 16.36 m on 7/1/07, indicating a water surface elevation 229.72 m.

The value of the 30-m filled DEM at the pixel where the gaging station is located is 230.71 m. This pixel is two pixels from the nearest stream pixel, which has elevation 224.67 m in the filled DEM. Thus the expected optimal floodplain determined for this location should have a maximum DTF value of 229.72 – 224.67 = 5.05 m.

3.8.4. Gage #3 (Osage River)

Gaging station #3 (USGS 06918070; Schnell City, MO) is located on the Osage River at (38.0559 N, 94.1454 W). Gage information and provisional gage height data were retrieved in September 2007 from the USGS's website at http://waterdata.usgs.gov/mo/nwis/uv?site_no=06918070.

The datum for gage #3 is 213.36 m above sea level. Floodwater crest with respect to the mean daily gage height was 14.90 m on 7/4/07, indicating a water surface elevation of 228.26 m.

The value of the 30-m filled DEM at the pixel where the gaging station is located is 219.24 m. This pixel is adjacent to a stream pixel that has the same elevation. Thus the expected optimal floodplain determined for this location should have a maximum DTF value of 228.26 – 219.24 = 9.02 m.

3.8.5. Results

Using the respective maximum flood depths described above, the FLDPLN model was applied to each of the three stream segments comprising the study area, using dh = 1 m. The extents from these applications are shown in Figure 3.31(a) as a red-green-blue (RGB) three color composite image, so that areas of overlap can be seen in addition to the combined extent. The union of the predicted, segment-specific flood extents provided an estimate for the total flood extent in the study area. Ignoring interior holes (potential islands in the floodwater expanse), the exterior flood extent boundary was identified. This predicted flood extent is shown in Figure 3.31(b), overlaid on Landsat-5 image collected 2-5 days after regional flood crest.

Using the Landsat-5 image, Kevin Dobbs of KARS manually digitized the exterior flood extent boundary. Mr. Dobbs has considerable experience creating manual digitizations of this sort, and was not shown the estimated extent from the FLDPLN model. Consequently, the delineated area produced by Mr. Dobbs can be considered to be both reasonably accurate and objectively determined. The manually delineated extent for a subset of the study area is shown in Figure 3.32(a). The modeled extent for the same subset is presented in Figure 3.32(b).

Considering the 30-m resolution of both the Landsat-5 image and the DEM used in the FLDPLN model, the modeled and manually delineated extents were sampled to the same 30-m grid. The resulting raster layers were used for accuracy assessment. The manually delineated extent indicated that 659,170 pixels (593.25 km^2) were inundated in the study area. Call this set MAN. The modeled extent indicated that 658,737 pixels (592.86 km^2) were inundated in the study area. Call this set MOD. With respect to total estimated flood area, the *percent bias* was computed for the modeled extent, given by

$$\text{percent bias} = 100 \cdot \frac{|\#\text{MOD} - \#\text{MAN}|}{\#\text{MAN}}. \qquad (3.1)$$

In (3.1), $|\cdot|$ denotes absolute value and # is set cardinality. A value of zero for percent bias occurs when the sizes of the modeled and manually delineated extents are the same, which suggests that the model does not demonstrate a tendency to either underestimate or overestimate total inundation area. Using (3.1), the percent bias for the modeled extent was 0.066%, indicating that the model produced a largely unbiased prediction for total inundation extent in the validation study area.

To assess the accuracy of the model prediction, the author used the equation for *percent accuracy* given by

$$\text{percent accuracy} = 100 \cdot \frac{\#\{\text{MAN} \cap \text{MOD}\}}{\#\{\text{MAN} \cup \text{MOD}\}}. \qquad (3.2)$$

Equation (3.2), which is also used in Bates & De Roo (2000), provides a simple, rigorous measure of model accuracy. If there is total agreement between the model and the manual delineation, the percent accuracy would be 100%. If there is no agreement at all (i.e., no overlap) between the model and the manual delineation, the percent accuracy would be 0%. Using (3.2), the percent accuracy for the modeled extent was determined to be 87.2%. For comparison, the largest percent accuracy achieved with any of the 1-D and 2-D hydrodynamic models tested in Bates & De Roo (2000) was 81.6%, in a validation study examining a major flood event occurring along 35 km of stream reach of the River Meuse in the Netherlands and Belgium.

Few other studies provide a similar validation study comparing modeled and observed flood extents (Bates & De Roo 2000). Another such study appears in Bradbrook *et al.* (2004), where the authors compare multiple 1-D and 2-D hydrodynamic model variants to estimate inundation area for a flood occurring on 4 km of river reach for the River Thames in England. Using an alternative, less robust accuracy statistic[11] that includes consideration for pixels "predicted" to not be flooded (i.e., dry pixels; this clearly will depend on the total size of the study area), the authors consistently achieved accuracies ranging from 81% to 84%. Using this same measure, the FLDPLN model demonstrated an accuracy of 93.2% in the validation study area shown in Figures 3.30-3.31.

[11] The authors themselves acknowledge that "This statistic is not necessarily a good means of model validation," and are using it simply to "maintain comparability" with Horritt and Bates (2001).

Finally, it should be noted using photo or image interpretation for manual delineation of actual flood extent is an error-prone procedure. This observation was also highlighted in Bates & De Roo (2000). In the present validation study, inspection of the manually delineated floodwater perimeter indicated that 10%-20% of this boundary abutted or passed through heavily wooded areas, which will occlude the actual floodwater boundary visible from above and possibly result in delineation errors (e.g., see the circled areas in Figure 3.32). Consequently, even if a model's output was 100% correct, percent accuracy values in the 80-90% range may be near the upper limit on achievable accuracy using this means of validation in many cases.

3.9. Conclusions and Future Directions

In this chapter, a new method (the FLDPLN model) for depth-dependent floodplain delineation was described. Compared to traditional hydrodynamic modeling methods, FLDPLN has several important advantages:

(i) The model has the utmost parametric economy, requiring DEM data and just two user-specified parameters (namely, *h* and *dh*) for general implementation.

(ii) The model is automated, and the output (namely, DTF and FSP values) is deterministic up to the choice of free parameter *dh*.

(iii) DTF values generated by the model can be used to provide a range of depth-dependent floodplain or flood extent estimates up to maximum flood depth *h*. Thus, with suitably chosen *h* and *dh* values, output from the

model can be used to develop a database that can be used for rapid estimation of local flood extent, given at least one coordinate from the floodwater surface or shoreline. Also, DTF values provide a meaningful index of hydrologic connectivity between floodplain locations and the main flow channel.

The FLDPLN model was developed to address some broad user needs for which current methods are generally incapable or prohibitive to implement. Examples of such needs are historic floodplain identification and rapid flood extent estimation. In other applications such as floodplain mapping for property zoning purposes (e.g., demarcating 100-year floodplains, i.e., areas where property development is constrained due to flood risk, or where the purchase of flood insurance is required by homeowners), FLDPLN may provide a low-cost alternative to hydrodynamic models if the local relationship between flood depth and flood frequency can be adequately estimated. Alternatively, FLDPLN may provide complementary information that can aid hydrodynamic model evaluation. For example, output from FLDPLN may facilitate more accurate spatial interpolation of the hydrodynamic model solution between cross sections. However, FLDPLN will never fully replace hydrodynamic models, which can provide additional outputs useful for studies in areas such as geomorphology and landform evolution. For example, output from hydrodynamic models also can include estimates for flow

velocity, discharge, and other physical parameters that are useful for simulating the processes of erosion and sedimentation (USACE HEC 2002, p.1-3).

The greatest practical shortcoming of the FLDPLN model is that it is computationally intensive, especially when implemented in steady-state form (i.e., when optional Step 7 from the FLDPLN algorithm is used). For example, 27 Mud Creek stream segments were processed to estimate the Mud Creek floodplain shown in Figure 3.20. This effort required approximately 200 CPU hours running MATLAB® on a 3.0 GHz desktop computer.[12] Though the author has already invested several hundred hours in FLDPLN code development, likely there are ways to increase algorithm efficiency that have not yet been considered. Future developments toward this end are desirable and potentially necessary, if the FLDPLN model gains acceptance and the author continues to obtain funding for large area floodplain mapping projects. However, due to ever-increasing desktop computing speeds, as well as the increasing availability of high-speed computing clusters, this problem is viewed as both temporary and ultimately inconsequential. Further, CPU time is essentially a one-time implementation cost, because once an appropriate database is constructed for the user's purposes (even if the purpose is rapid flood extent estimation for unknown future flood events), then no more model calculations need be performed.

[12] This time would have been reduced by 50%-75% had the entire Mud Creek stream reach been processed all at once. However, the author was using this dataset to test a recently developed version of the code that can be generally applied to large study areas without concern for memory limitations.

Several examples were presented to firmly establish the use of FLDPLN for historic floodplain identification and mapping, examining a variety of DEM resolutions and river sizes. Also, two examples examining the sensitivity of the FLDPLN model output to different choices for *dh* were described, helping to illustrate the behavior of the model and establishing the robustness of the model as it might be used for real-time flood extent estimation.

A validation study was presented to test the capability of the model for actual flood event estimation, which also provided assurance that the algorithm indeed functions as intended. For this study, inundation area was examined from a major flood event along 130 km of stream reach in a forked stream system situated mostly in eastern Missouri. Using just three water surface elevation values and 30-m DEM data, the FLDPLN model was able to predict inundation extent covering nearly 600 km^2 with 87.2% accuracy.

Compared to the few other studies that have also attempted model validation in this manner, this accuracy is outstanding. This result is even more impressive considering the large scope of the study area, the examination of a forked stream reach with many inflowing tributaries of various sizes, the general spatial complexity of the floodwater boundary, and the fact that the automated FLDPLN model prediction was based on just three floodwater surface elevation values. That said, the event examined was certainly "extreme"; calculations by scientists at the USGS estimated the revisit time for a flood of this magnitude to exceed 1000 years at some locations in the study area. Consequently, it is likely that this was a "valley full", or

"wall-to-wall" flood, which can be easier to model.[13] Also, the shallow landscape gradient across the study area (there is only a 22 m change in elevation between the DEM stream points at the western and eastern edges of the study area) generally makes this study site amenable to the uniform flow assumption underlying the segment-specific FLDPLN model evaluations. All things considered, however, the validation study provides strong support for the validity and utility of the FLDPLN model.

Similar validation studies should be undertaken to more thoroughly assess the validity and expose the limitations of the FLDPLN model for event-specific estimation of inundated area. Such studies would be enhanced if hydrodynamic model predictions could be compared side-by-side with predictions from FLDPLN. One concern the author has regards the accuracy of FLDPLN model predictions for more frequent (lower depth) flood events, where topographic considerations alone may not be sufficient for accurately estimating floodwater spread (e.g., where friction and inertial forces are more likely to have a substantial influence). If the FLDPLN model can be validated for a variety of flood magnitudes at a variety of locations, this would reduce concerns regarding the general "correctness" of the computational model itself, facilitating acceptance by hydrologists and adoption by private and public agencies concerned with floodplain mapping and flood extent estimation.

[13] On the other hand, ground survey data in the U.S. are rarely collected beyond expanses expected for 100- or 500-year flood events, so that necessary data are not generally readily available for hydrodynamic model estimation of extreme (i.e., > 500 year) flood events. This exposes a need for a tool generally capable of extreme flood modeling, such as FLDPLN.

Appendix: Asymptotic Consistency of Backfill Flooding with Planar Flooding

In Section 3.2.3, it was noted that floodplain cross sections used in hydrodynamic models are typically drawn orthogonal to elevation contours, and thus roughly follow topographic gradient lines. It was also noted (in a footnote) that hydrodynamic model solutions assume a constant water surface elevation along each cross section. Consequently, elevation contours from a 3-dimensional floodwater surface estimated using a hydrodynamic model will roughly coincide (in the horizontal planar projection) with cross-sectional topographic gradient lines. The same can be said for the "potential flood extent (PFE) for a stream segment" concept described in Section 3.4, which helped illustrate the objective of the FLDPLN model.

If the distance between cross sections (which correspond to hypothetical dam faces in the PFE setup) is allowed to become arbitrarily small, then the limiting floodwater surface is comprised entirely by elevation contours that follow cross-sectional topographic gradient lines. Because of this, the water surface profile along cross sections *orthogonal* to a pitched flow channel like that shown in Figure 3.11(a) will increase in elevation toward the middle of the channel. In other words, an observer on the shore looking straight across the floodwater surface would see water "mounding up" near the center of the channel, resulting in a ridge (or crest) of water running down the length of the floodwater surface. The author is unsure whether or not this "crested floodwater surface" provides an accurate representation of actual pitched channel flow, and needs to further investigate this matter.

As an alternative, one can obtain a "planar floodwater surface" by specifying floodplain cross sections orthogonal to the flow channel instead of orthogonal to topographic contours. In this case, an observer on the shore looking straight across the floodwater surface would be looking along a water surface contour, and thus would see a horizontal (uniform elevation) water surface. This representation may be more realistic than the "crested floodwater surface" obtained using gradient-based cross sections. Regardless, it seems likely that actual stream behavior generally will reflect one of these two characterizations, or something in between. Consequently, it is beneficial to understand the mathematical relationship between these two specifications.

Because the spillover portion of the PFE cannot be precisely defined, for simplicity we consider only the backfill portion of the PFE, which will be referred to as the *backfill PFE*. Using the idealized pitched channel depicted in Figure 3.11(a), the geometry for the alternative backfill PFE representation (which uses a hypothetical dam face that is orthogonal to the channel) is shown in Figure A3.1, overlaid on the original backfill PFE geometry depicted in Figure 3.12. Note the difference in cross section width.

For the pitched channel, the "planar water surface" is the simplest water surface to mathematically characterize. Define the *simple flood extent at flood depth* h (SFE(h)) for the pitched channel S using the plane $L_h = L + h$, where L is the landscape plane shown in Figure 3.11(c). The portion of L_h that lies above S defines the extent of SFE(h). Figure A3.2(a) shows SFE(h) for pitched channel surface S.

Figure A3.2(b) shows the SFE and the PFE (with the latter evaluated for a sequence of points along the channel bottom) in the horizontal planar projection, so that the two inundation extent estimates can be directly compared.

Define the *surface pitch* to be $P = |\partial z/\partial y|/|\partial z/\partial x|$, so that $P = 1/2$ for pitched channel S used in Figures 3.1-3.13 and Figures A3.1-A3.2. We will show that the backfill PFE constructed using a sequence of points along the bottom of the pitched channel converges to the SFE as $P \to 0$.

Consider the geometry of the backfill PFE in the pitched channel shown Figure A3.3. Since the only quantity of interest is areal flood extent, all quantities to be discussed refer to projections in the horizontal plane. Let D be the length of the channel segment under study. Assume the channel bottom points constituting stream segment R are uniformly spaced, and let d be the distance between consecutive points. Since the geometry is symmetric about the channel bottom, derivations are needed only from one side of the pitched channel. WLOG, assume d is small enough so that consecutive, point-specific backfill PFEs overlap. As channel pitch $P \to 0$ and the point-specific backfill PFEs elongate in the y direction, this is assured to be the case regardless of d, so this assumption imposes no constraint on the upcoming asymptotic analysis.

WLOG, assume that the denominator of P (i.e., $|\partial z/\partial x|$) remains constant, and that variations in P are attributed to changes in the numerator (i.e., $|\partial z/\partial y|$, the absolute slope of landscape plane L (see Figure 3.2(c)). With this setup, the areal extent of the SFE for a given flood depth will be constant as $P \to 0$. Consequently,

we can normalize the problem geometry so that the constant half-width of the SFE is equal to one (see Figure A3.3). For convenience, denote by SFE the areal extent of the SFE.

Denote the areal extent of the backfill PFE by PFE(d,P), since the backfill PFE is dependent on both point spacing and channel pitch. Define the error

$$Err(d,P) := \text{SFE} - \text{PFE}(d,P). \tag{A3.1}$$

Due to its construction in the pitched channel, the PFE necessarily will be contained in the SFE (see Figure A3.1). Thus we will have $Err(d,P) \geq 0$.

Due to the d-periodic errors between the SFE and PFE(d,P), from the geometry in Figure A3.3 we can construct the following inequality:

$$Err(d,P) \leq 2 \lceil D/d \rceil (A_1(d,P)+A_2(d,P)). \tag{A3.2}$$

The sum of A_1 and A_2 comprises the fundamental unit of error between PFE(d,P) and SFE. The $\lceil D/d \rceil$ coefficient provides an upper bound on the number of fundamental error units occurring on one side of the length-D channel, and the "2" coefficient doubles the error bound for the two sides of the channel.

Consider the acute angle θ between the y-axis (parallel to <0,-1>) and a contour. WLOG, suppose the right-hand half-plane comprising the pitched channel is

given by $z = nx + my$, where $n > 0$ and $m \geq 0$. Then contours on that half of the pitched channel are parallel to $\langle m,-n \rangle$, and it follows that $\tan(\theta) = m/n$. Since channel pitch $P = |\partial z/\partial y|/|\partial z/\partial x| = m/n$, we have the relationship $\theta = \theta(P) = \tan^{-1}(P)$.

From the geometry of the problem shown in Figure A3.3, we can determine the equation for A_1. In particular, we have $s_1 = d\sin\theta$ and $s_2 = d\cos\theta$. Thus we have

$$A_1(d,P) = s_1 s_2/2 = (d^2/2)\sin(\theta(P))\cos(\theta(P)). \qquad \text{(A3.3)}$$

Also, we can determine the equation for A_2. Using trigonometric relationships of similar triangles (see the dashed line triangles below A_1 and A_2 in Figure A3.3), it is easy to show that $w = \sin^2\theta$. Then the equation for A_2 is given by

$$A_2(d,P) = dw = d\sin^2(\theta(P)). \qquad \text{(A3.4)}$$

From equations (A3.3) and (A3.4), we obtain the following properties:

(a) $A_1 \rightarrow 0$ as $P \rightarrow 0$

(b) $A_2 \rightarrow 0$ as $P \rightarrow 0$

THEOREM A3.1: $Err \rightarrow 0$ as $P \rightarrow 0$.

PROOF: This follows from properties (a) and (b), the error bound from (A3.2), and the non-negativity of *Err*. QED

Theorem A3.1 establishes that for the pitched channel, PFE(d,P) is a consistent estimator for SFE as $P \rightarrow 0$.

Now suppose P is fixed, but we are able to arbitrarily decrease channel bottom point spacing. Then we have the following theorem:

THEOREM A3.2: $Err \rightarrow 2D\sin^2\theta$ as $d \rightarrow 0$.

PROOF: We can specify slightly different upper and lower bounding inequalities for Err:

$$2(D/d-1)(A_1(d,P) + A_2(d,P)) \leq Err(d,P) \leq 2(D/d+1)(A_1(d,P) + A_2(d,P))$$

$$\Leftrightarrow 2(D-d)(A_1(d,P)/d + A_2(d,P)/d) \leq Err(d,P) \leq 2(D+d)(A_1(d,P)/d + A_2(d,P)/d)$$

Substituting equations (A3.3) and (A3.4) for A_1 and A_2, respectively, we get

$$(D-d)(d\sin\theta\cos\theta + 2\sin^2\theta) \leq Err(d,P) \leq (D+d)(d\sin\theta\cos\theta + 2\sin^2\theta)$$

Letting $d \rightarrow 0$, this inequality becomes $2D\sin^2(\theta(P)) \leq Err(P) \leq 2D\sin^2(\theta(P))$, establishing the theorem. QED

With channel bottom points spaced finely enough that the cumulative A_1 error is negligible, by Theorem A3.2, the underestimation error w between the PFE width and

the SFE width will be $Err/D \approx 2\sin^2\theta = 2\sin^2(\tan^{-1}P)$. Thus for $P<<1$, we have $Err/D \approx 2(\tan^{-1}P)^2 \approx 2P^2$, indicating that the underestimation error between the PFE and the SFE will be small in such cases.

Tables and Figures

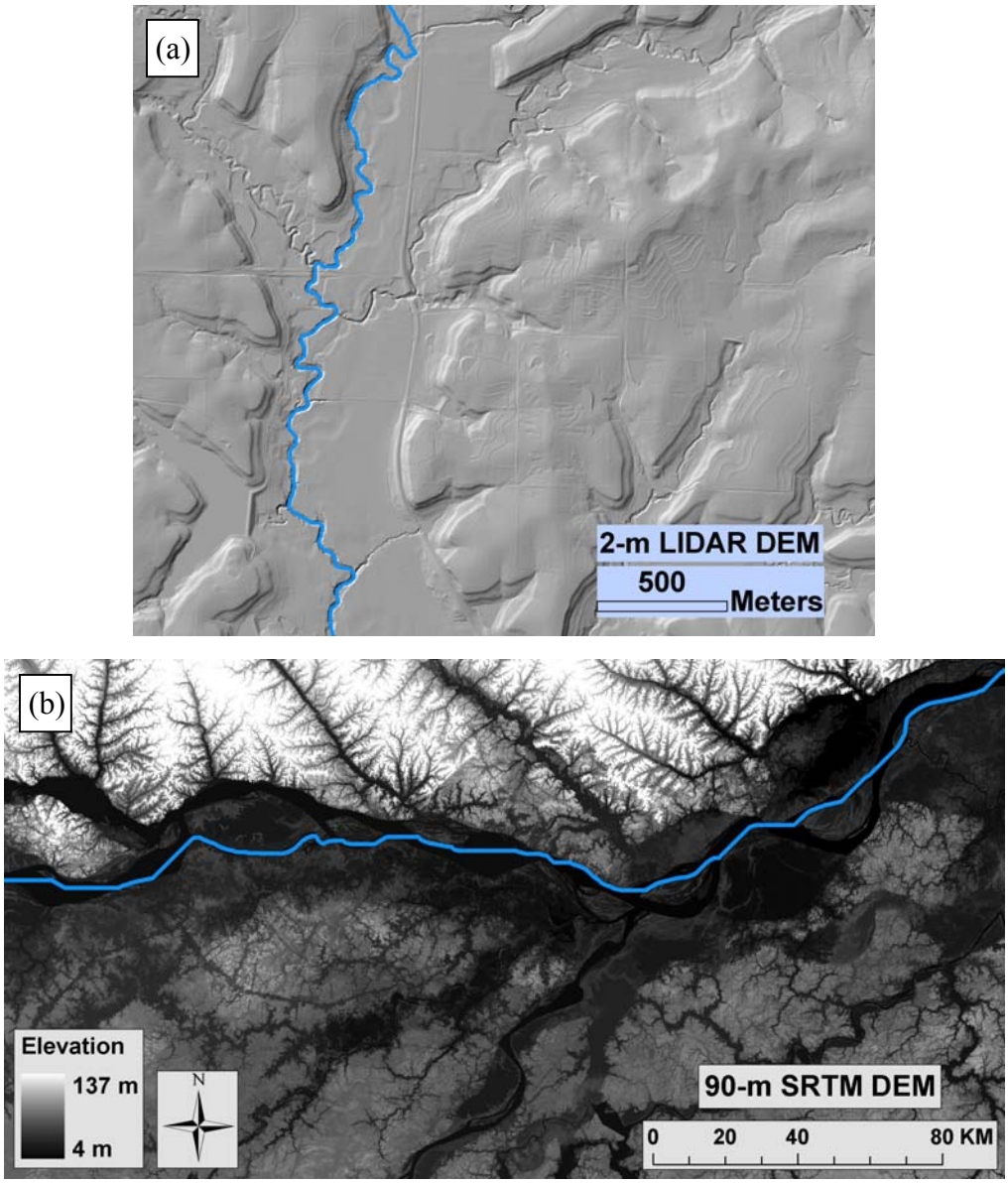


Figure 3.1. Digital elevation model (DEM) data are shown for two different stream segments. (a) High resolution (2-m) DEM data derived from aerial measurements using a LIght Detection And Ranging (LIDAR) instrument are shown in hillshade relief format. These data cover a portion of Mud Creek below Lake Dabinawa, located approximately nine miles north of Lawrence, KS. (b) Low resolution (90-m) DEM data derived from data collected during the Shuttle Radar Topography Mission (SRTM) are shown for a portion of the Amazon River in Brazil. The DEM-derived main flow channel is indicated in each subplot using a blue line. Though topographically distinct river valleys (i.e., historic floodplains) are generally visible in both plots, precise manual delineation of the floodplain boundary would be difficult in either case.

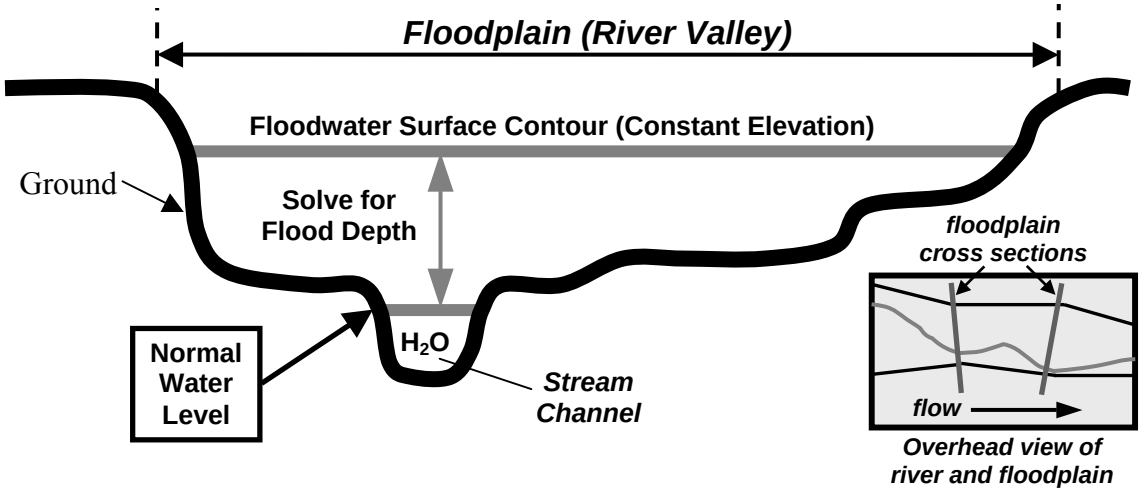


Figure 3.2. Simple diagram of a floodplain cross section used in hydrodynamic modeling.

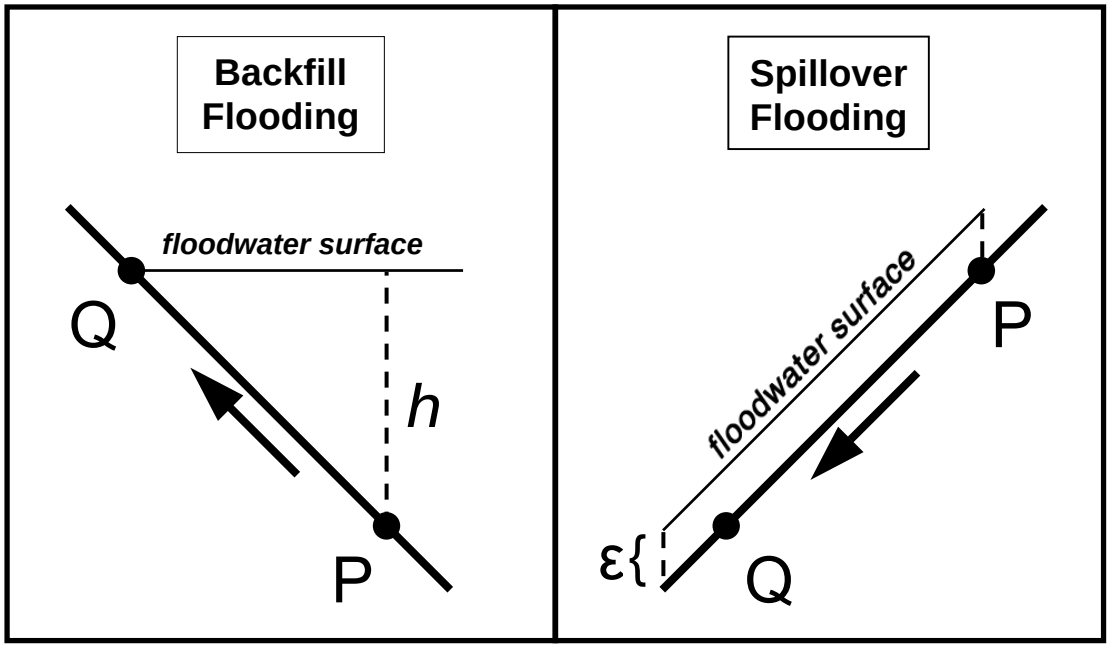


Figure 3.3. The proposed FLDPLN model is based on the assumption that there are two distinct ways for floodwaters originating from point P to inundate point Q, dependent on the position of P relative to Q. Backfill flooding (Q uphill from P) describes swelling processes, and spillover flooding (Q downhill from P) describes overland flow processes.

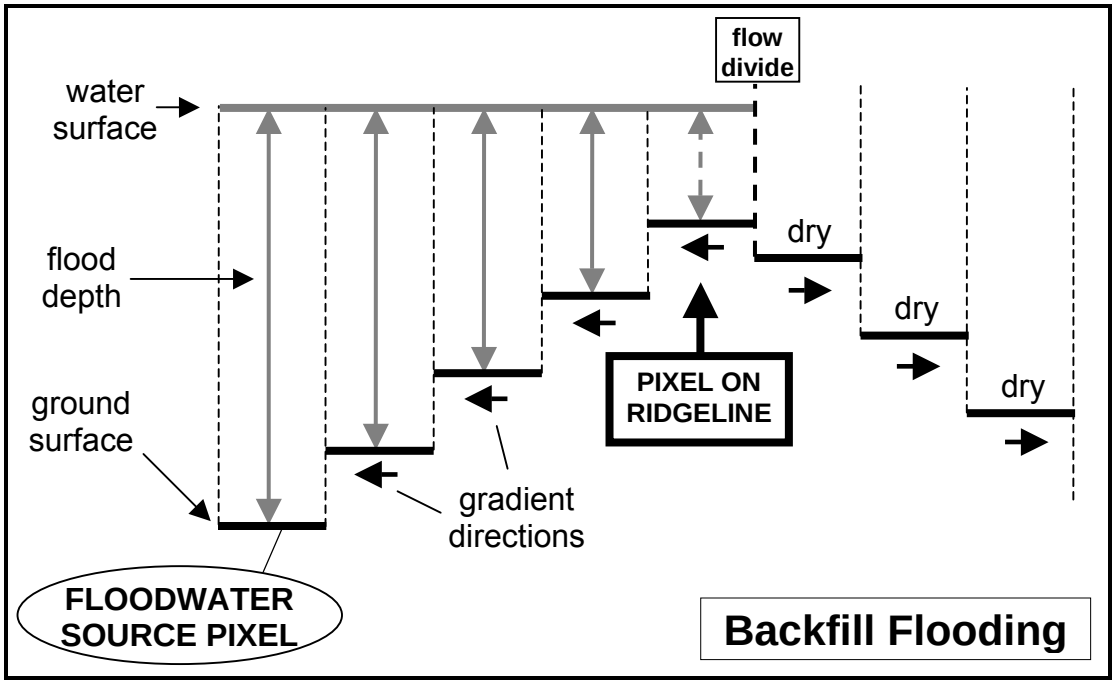


Figure 3.4. Backfill flooding models floodwater swelling effects. A shortcoming of backfill flooding is that it hangs up whenever a flow divide is encountered in the flow direction map (gradient direction field).

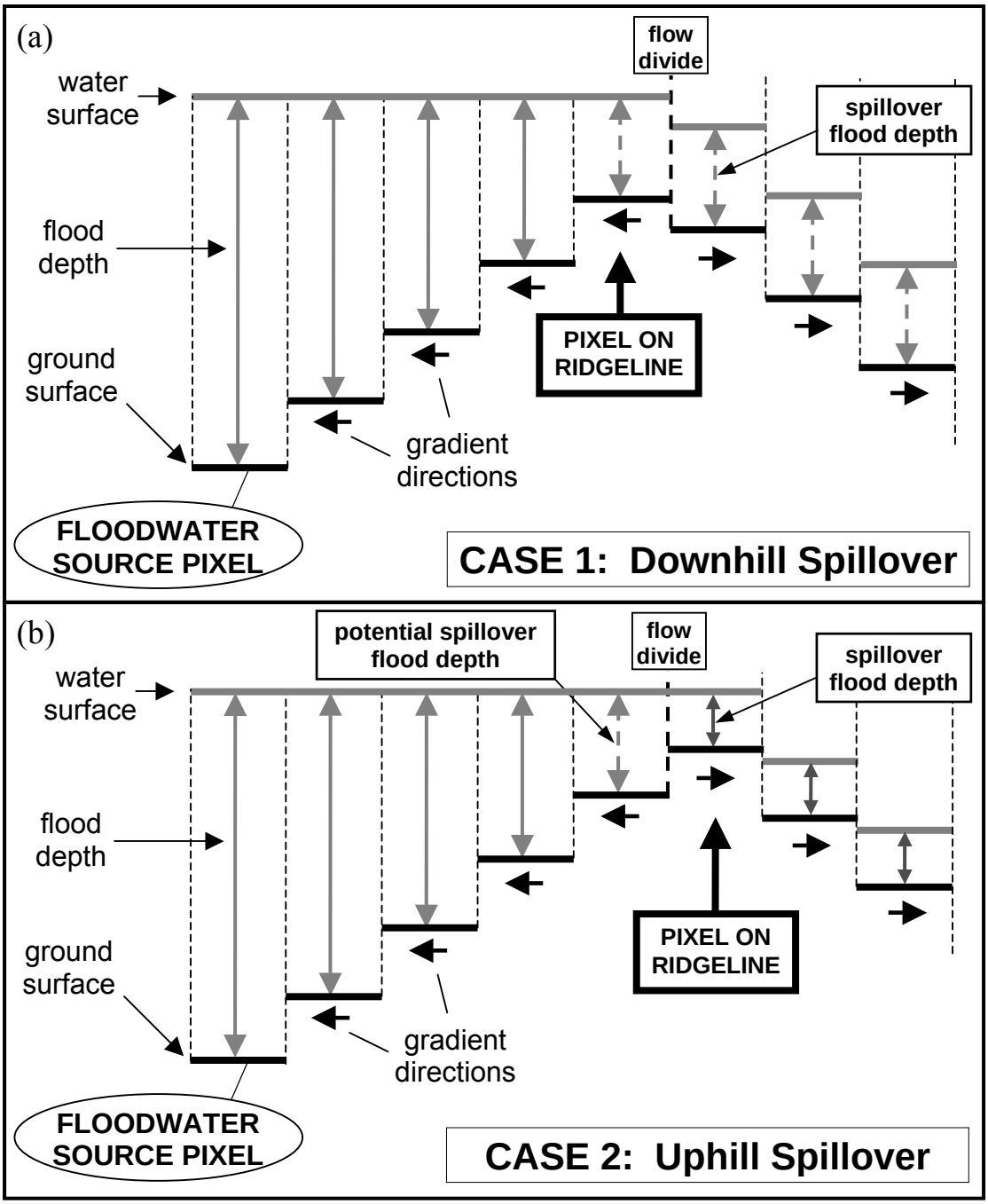


Figure 3.5. Spillover flooding resolves the ridgeline problem of backfill flooding by introducing new floodwater routes. Two types of spillover can occur, (a) downhill spillover and (b) uphill spillover. Depending on the break of ridge top pixels, part of the potential floodwater spillover depth sometimes must be subtracted to allow for uphill spillover.

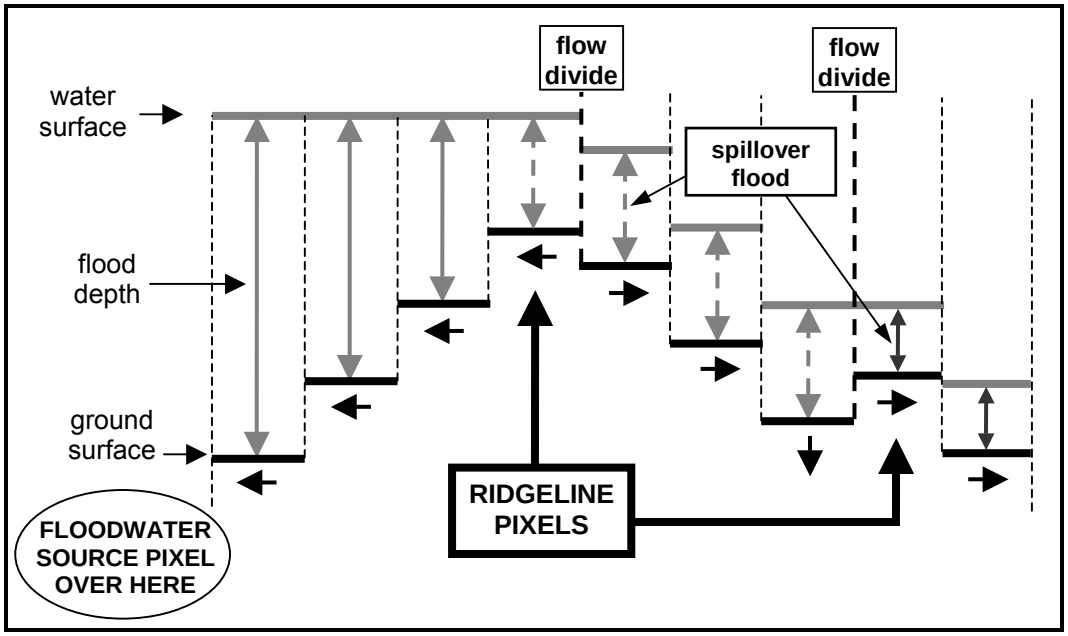


Figure 3.6. Sometimes multiple spillover steps are necessary.

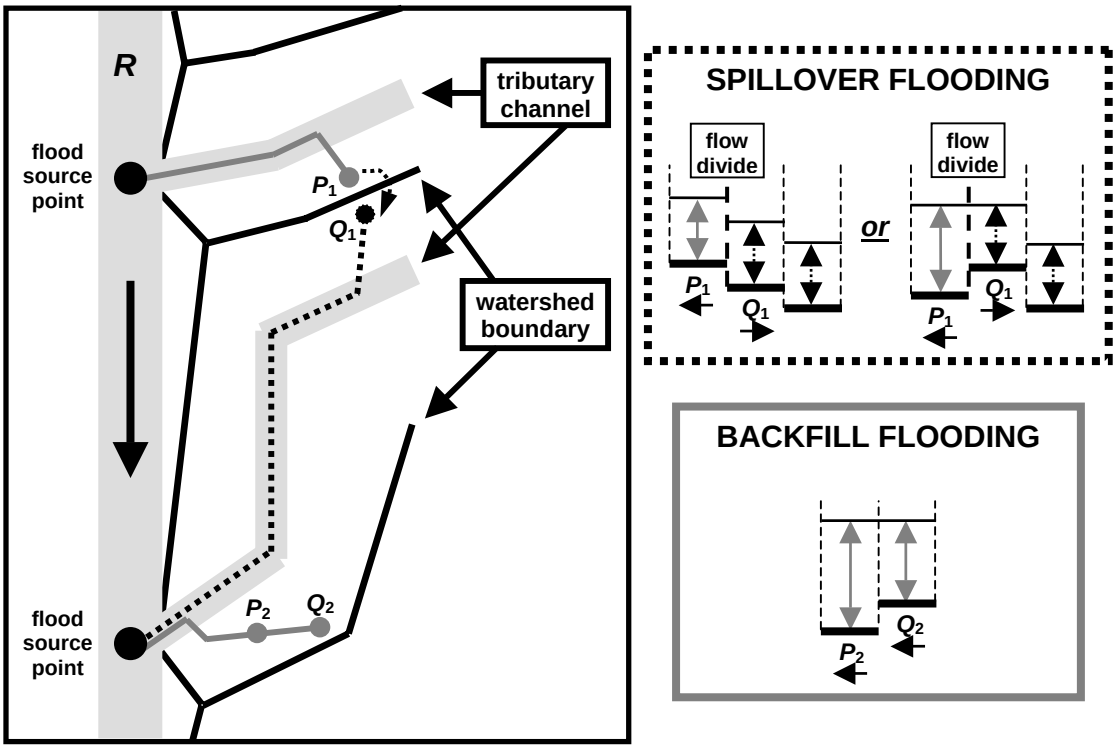


Figure 3.7. Plan view of spillover and backfill flooding. *R* denotes some stream segment for which the floodplain is being determined.

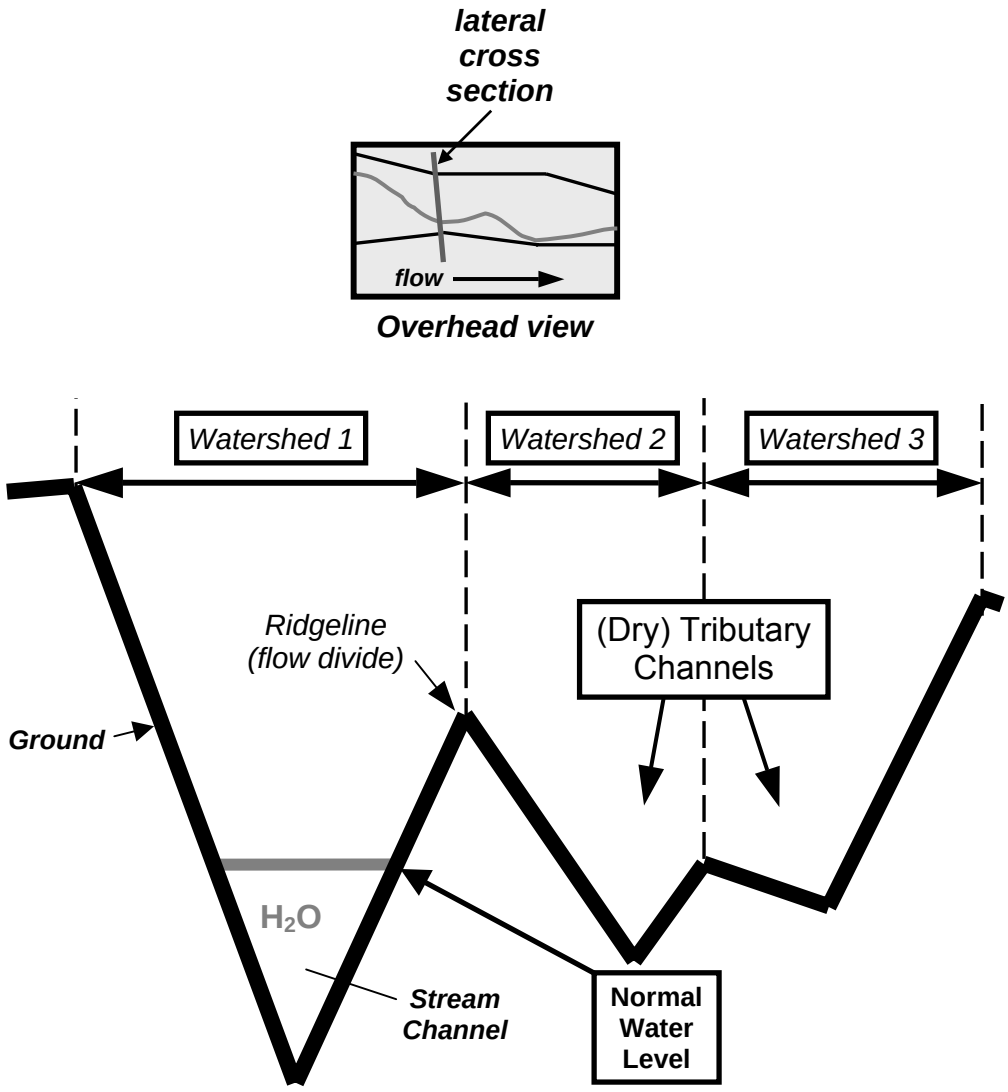


Figure 3.8. Diagram of a lateral floodplain cross section.

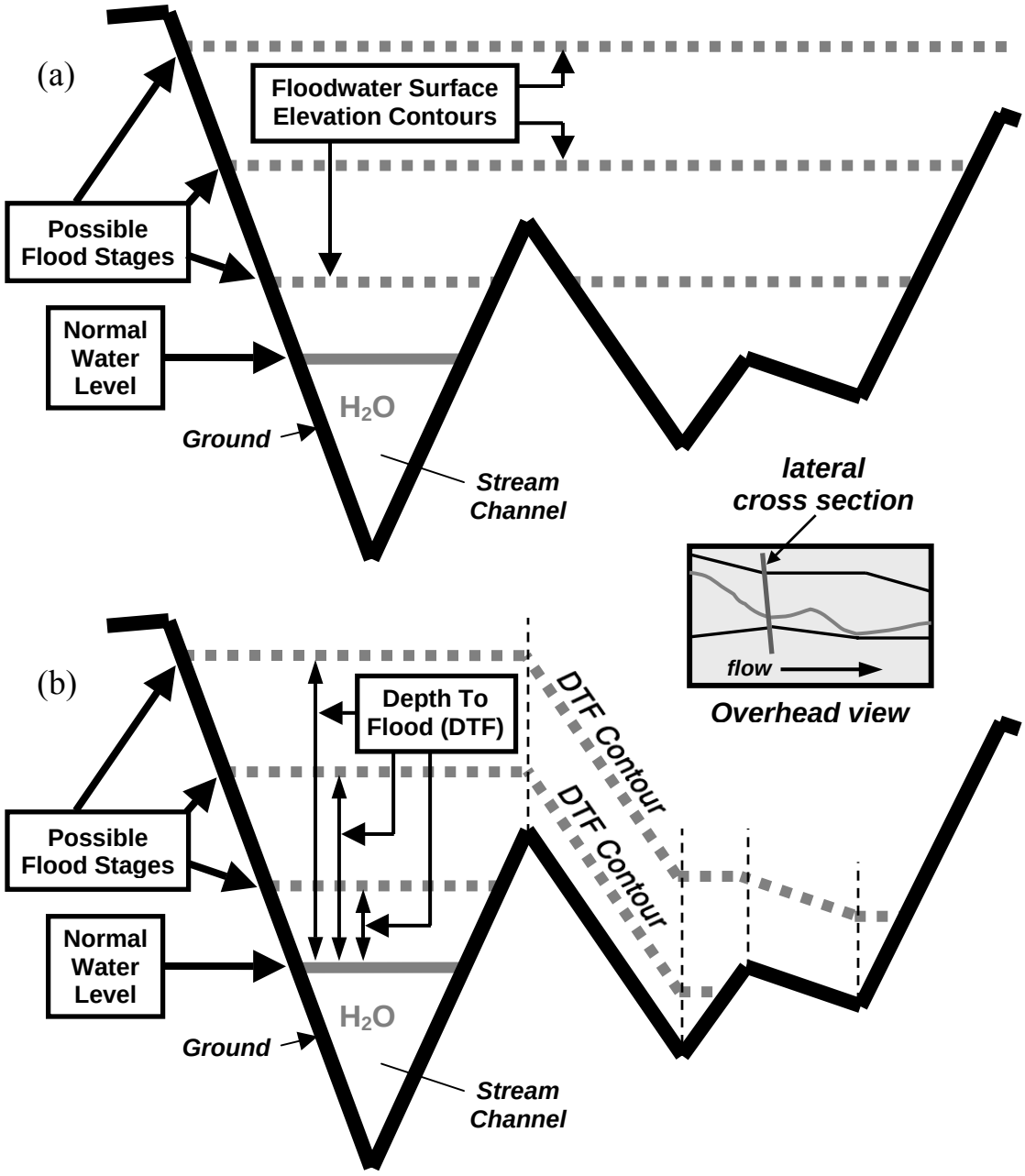


Figure 3.9. (a) Possible floodwater surface solutions obtained using a traditional hydrodynamic modeling approach. (b) Possible floodwater surface solutions obtained using the FLDPLN model.

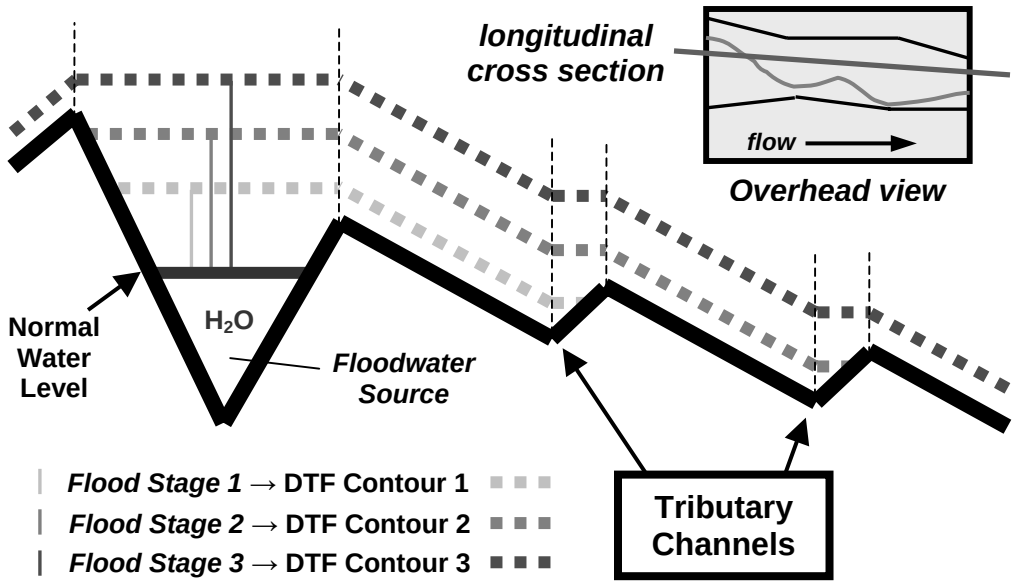


Figure 3.10. The FLDPLN model solution exhibits the same general behavior along any floodplain cross section. Each time floodwaters reach a tributary channel, the available spillover flood depth decays as floodwaters have to rise to breach the next downstream ridgeline.

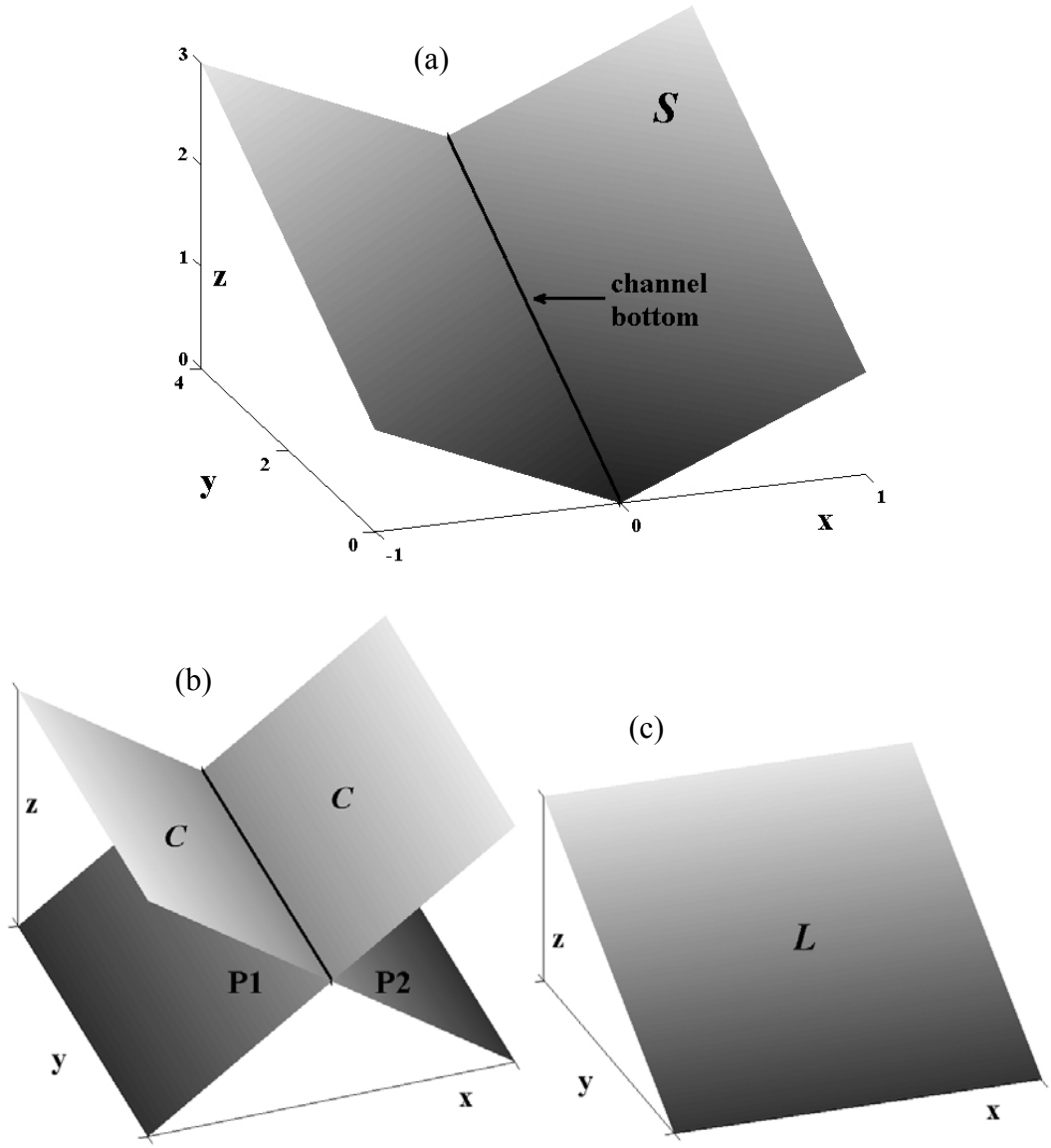


Figure 3.11. (a) Grayscale contour map of a pitched channel (surface *S*) with $|\partial z/\partial x| = 1$ and $|\partial z/\partial y| = 1/2$. Along the *x*-axis is the local topography gradient (detail scale), and along the *y*-axis is the landscape gradient (trend scale). Planar components are used to construct $S = C + L$. Horizontal channel *C* (constructed from planes P1 and P2) is shown in (b), and landscape plane *L* is shown in (c).

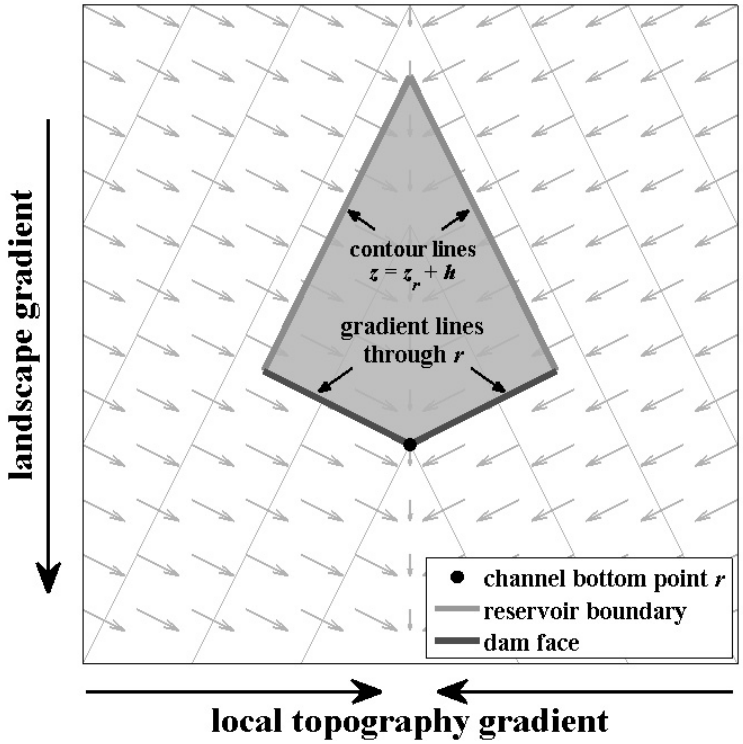


Figure 3.12. Hypothetical reservoir depicting the potential backfill flood area realized with a flood with depth h at channel bottom point r. A hypothetical dam is constructed along the sidewall gradient trajectories through r. The gradient field and elevation contours for pitched channel S are shown in the background. All gradient arrows indicate downhill directions.

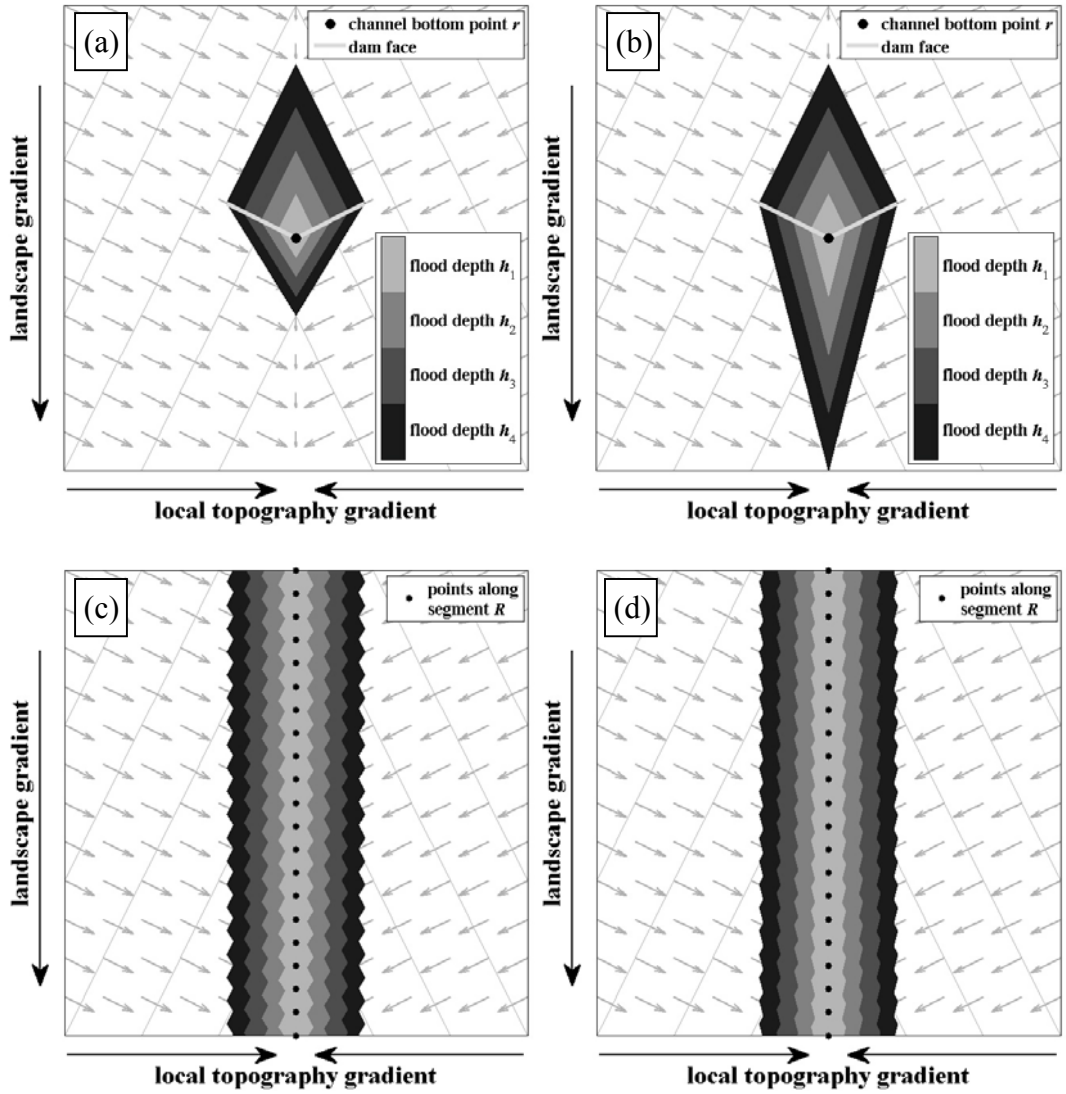


Figure 3.13. (a) Possible extents for $PFE_r(h)$, where r is a single point along the channel bottom of pitched channel S. Extents are shown for four flood depths $h_1 < h_2 < h_3 < h_4$. Subplot (b) shows alternative $PFE_r(h)$ extents, differing from (a) with respect to the spillover flood extent following dam removal. Note that the backfill (reservoir) portion of PFE_r is determined entirely by contour lines and gradient lines through r, and is thus identical in (a) and (b). Suppose stream segment R is comprised by the channel bottom points shown in (c) and (d). Then (c) shows possible extents for $PFE_R(h)$, obtained by taking the minimum flood depth union of point-specific PFEs of the form shown in (a) for each point in R. Subplot (d) is identical to (c), except that point-specific PFEs of the form shown in (b) were used. Comparing (c) and (d), the process of combining multiple point-specific PFEs overcomes much of the difference between (a) and (b).

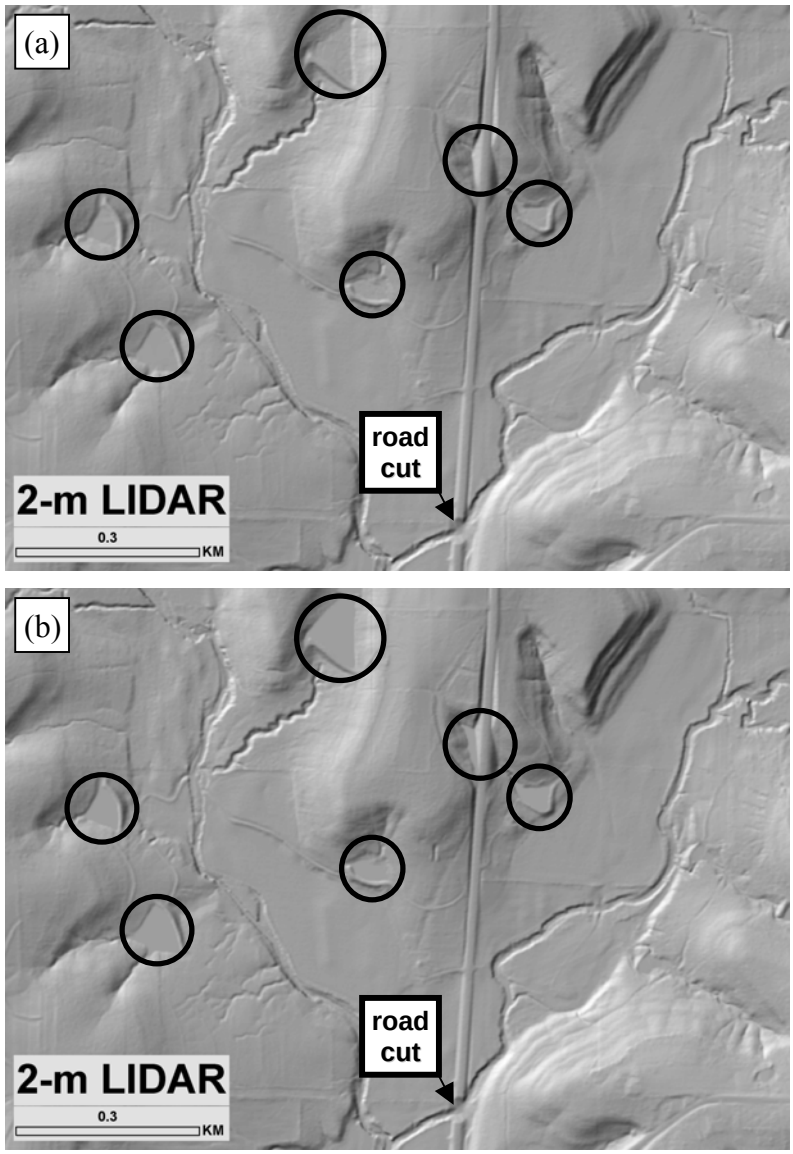


Figure 3.14. A DEM subset is shown for a portion of Mud Creek in Jefferson County, KS. Subplot (a) shows the original DEM acquired from http://www.kansasgis.org/. The filled DEM is shown in (b). Six depressions (sinks) are circled, five of which correspond to small, impounded ponds, and one from the high side of a ravine road overpass. Comparing circled areas, the effects of sink filling can be seen. A road cut is also highlighted, an intended consequence of the data processing methods used during DEM production. Removing obstructions over waterways in the DEM facilitates hydrologic studies and reduces the need for sink filling.

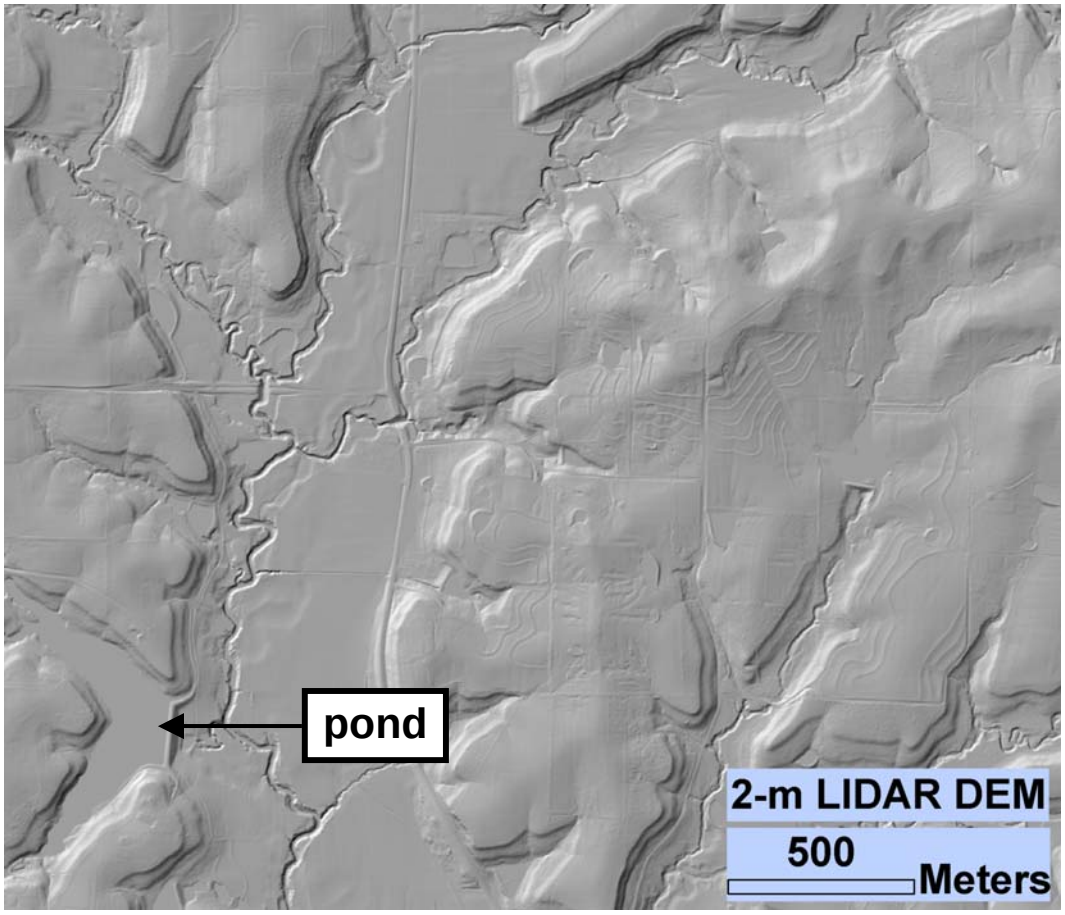


Figure 3.15. The filled DEM is shown in hillshade relief format for a different Mud Creek study area. The depicted extent is centered roughly at (39.092 N, 95.255 W). These DEM data, which have 2-m pixel size, are considered "high resolution". The pond required filling to reach its spill point on the upper side of the dam, and thus its water surface has constant elevation. This pond will be referenced in Figure 3.16.

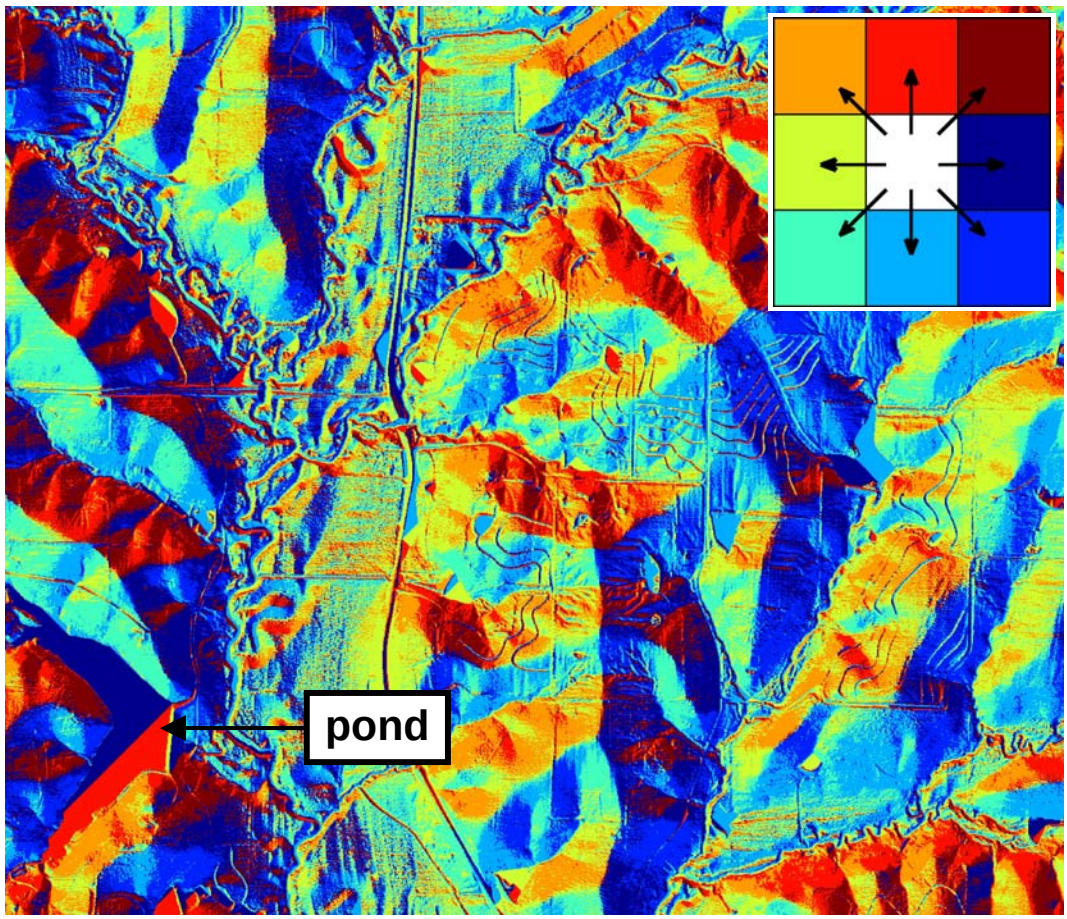


Figure 3.16. The flow direction map (FDR) is shown for the Mud Creek study area used in Figure 3.15. Each pixel is colored according to its flow direction (see the legend graphic in the upper right). The FDR provides a discrete approximation for the gradient direction field. Pixel-level flow directions are somewhat noisy, especially in the flat areas of the floodplain. On the uplands where there is more relief, it is fairly easy to identify features such as ridges and ravines by inspecting the general color patterns and matching these with the legend graphic. The semi-regular linear features visible on the right side of the image correspond with agricultural terraces. Flow through the pond is generally routed toward a single diagonal flow trajectory in the middle of the pond, which occurs at the interface between the blue and red areas but cannot be seen at this zoom level. The diagonal trajectory empties at the spill point on the upper side of the dam.

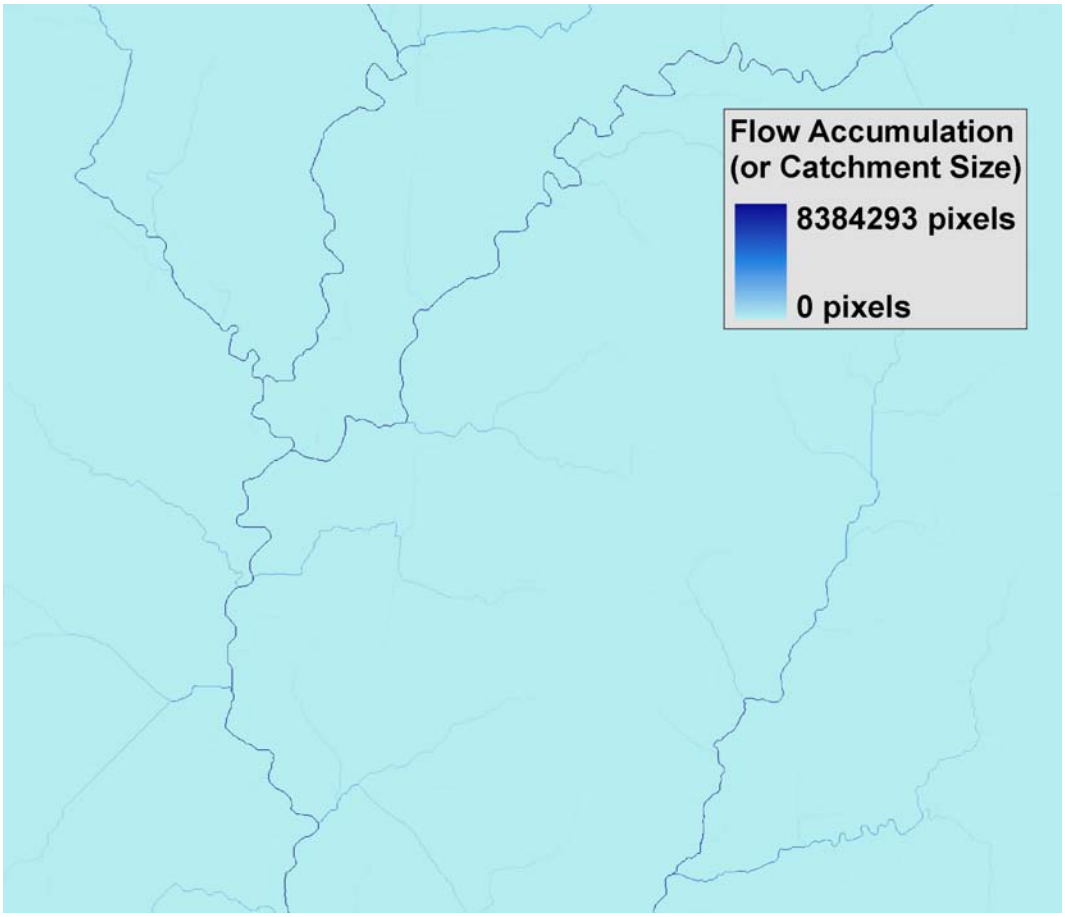


Figure 3.17. The flow accumulation map (FAC) is shown for the Mud Creek study area used in Figure 3.15. Pixel-level flow accumulation values are indicated using light to dark shades of blue. Note that the only pixels in the scene with distinctly large flow accumulation values occur along the bottoms of the drainage channels visible in the DEM shown in Figure 3.15.

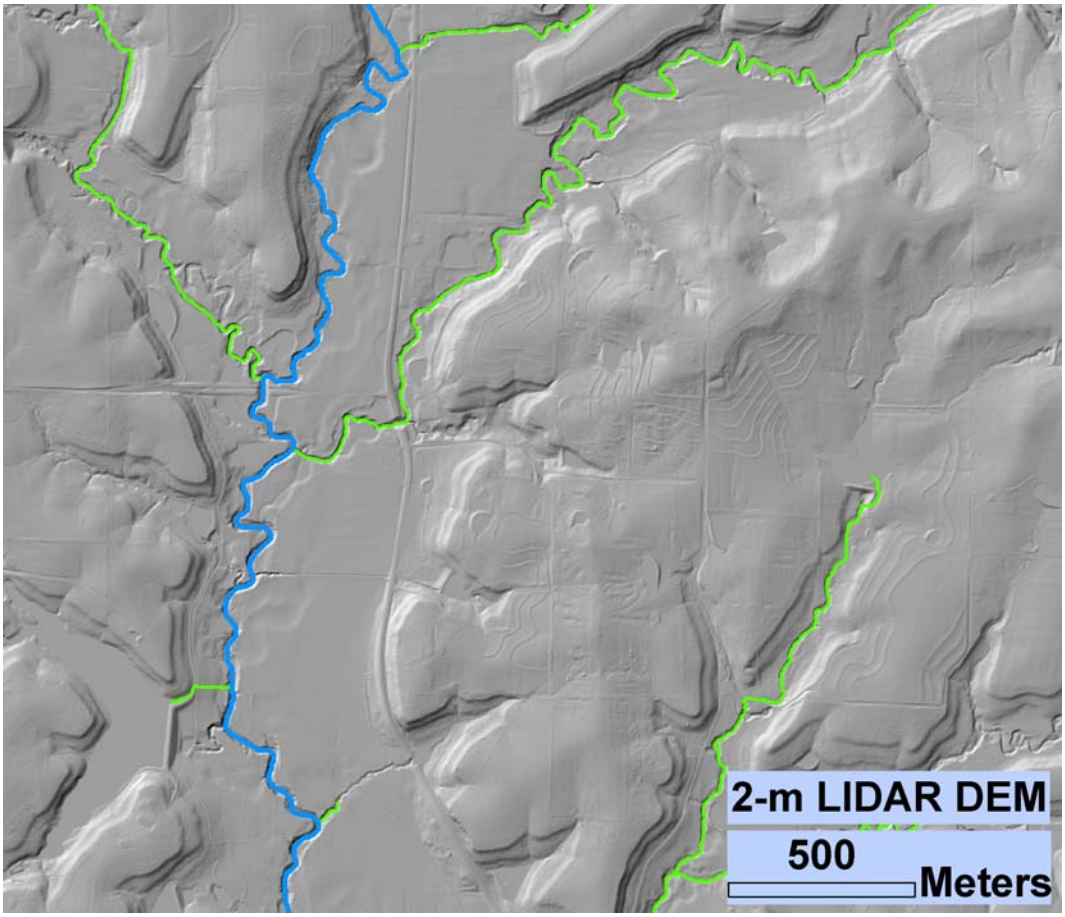


Figure 3.18. Pixels with catchments (FAC values) larger than 10^5 pixels were retained to define a synthetic stream network. The extent of this stream network is shown here, indicated by green and blue lines overlaid on the DEM. The blue line corresponds with the Mud Creek stream segment, which is specifically highlighted because it will be used for demonstration of the proposed floodplain mapping algorithms. Note the generally good correspondence between the stream network and the actual main drainage channels visible in the DEM.

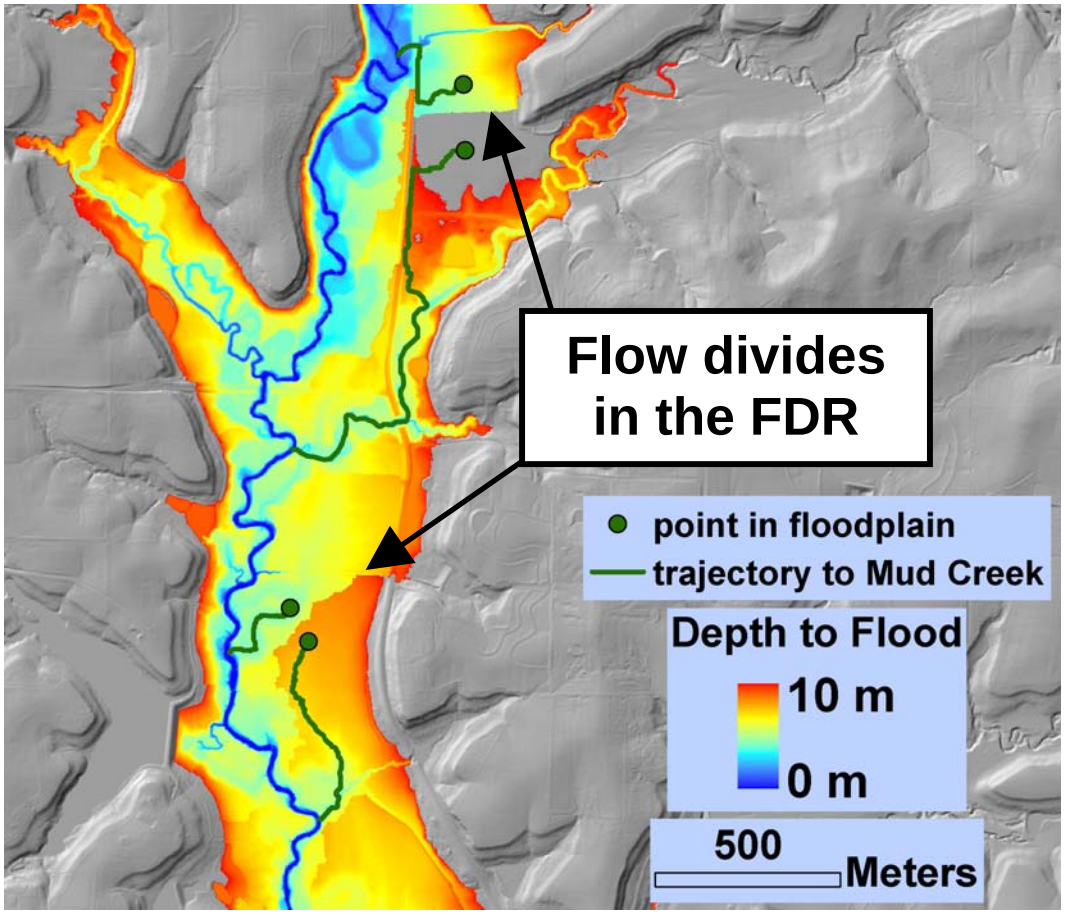


Figure 3.19. The 10-m backfill floodplain DTF map is shown for the Mud Creek study area used in Figures 3.15-3.18. This floodplain was determined using the Mud Creek segment shown in Figure 3.18. The BFA is unable to simulate the breaching of ridgelines (which correspond to flow divides in the FDR) by floodwaters. This shortcoming can result in (potentially large) erroneous discontinuities in DTF values and underestimation of the floodplain. Two severe discontinuities are identified above. Example trajectories are shown for floodplain pixels on both sides of the highlighted flow divides. As can be seen from these examples, discontinuity problems are a regular occurrence when tributary-specific watersheds run parallel with the main flow channel and only backfill flooding is used.

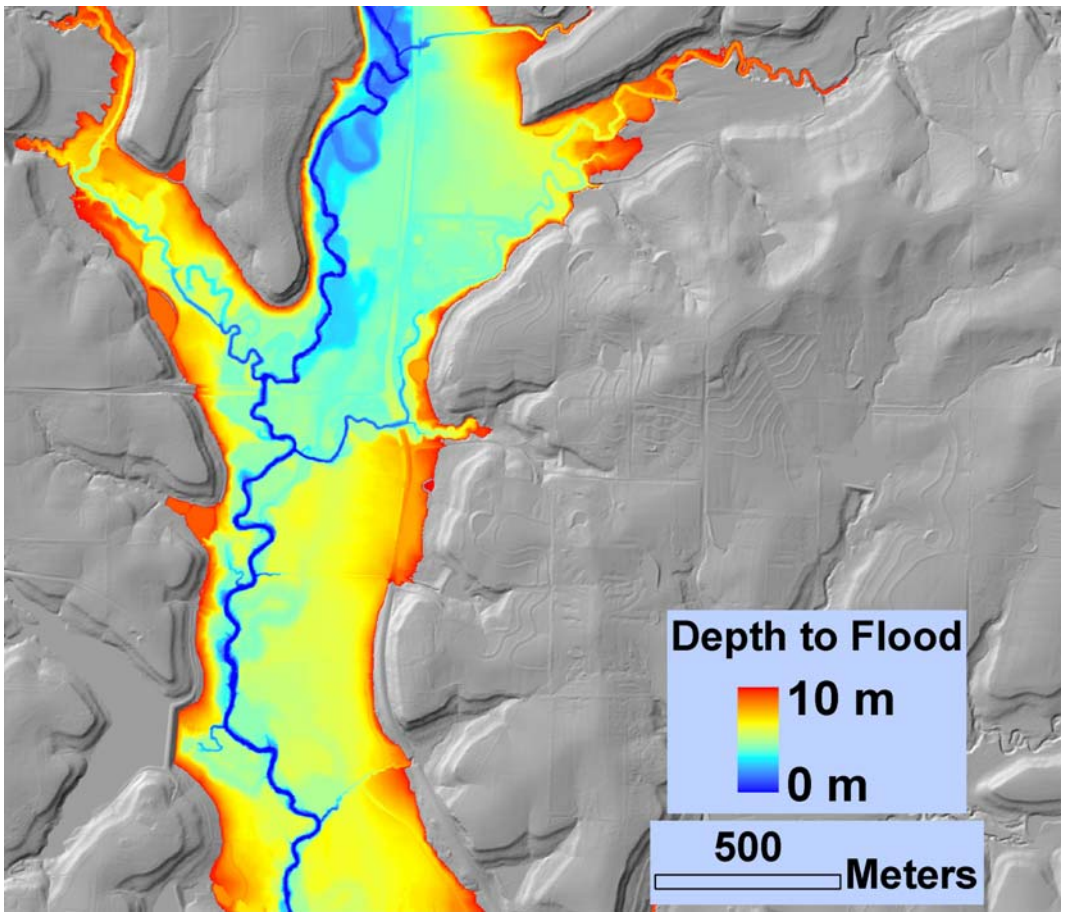


Figure 3.20. The 10-m steady-state floodplain DTF map is shown for the Mud Creek study area used in Figure 3.15-3.19. This floodplain was determined using the FLDPLN model with $dh = 0.5$. Compare this floodplain estimate to the BFA floodplain shown in Figure 3.19. Note that the DTF discontinuities visible in the BFA floodplain are no longer apparent in this map.

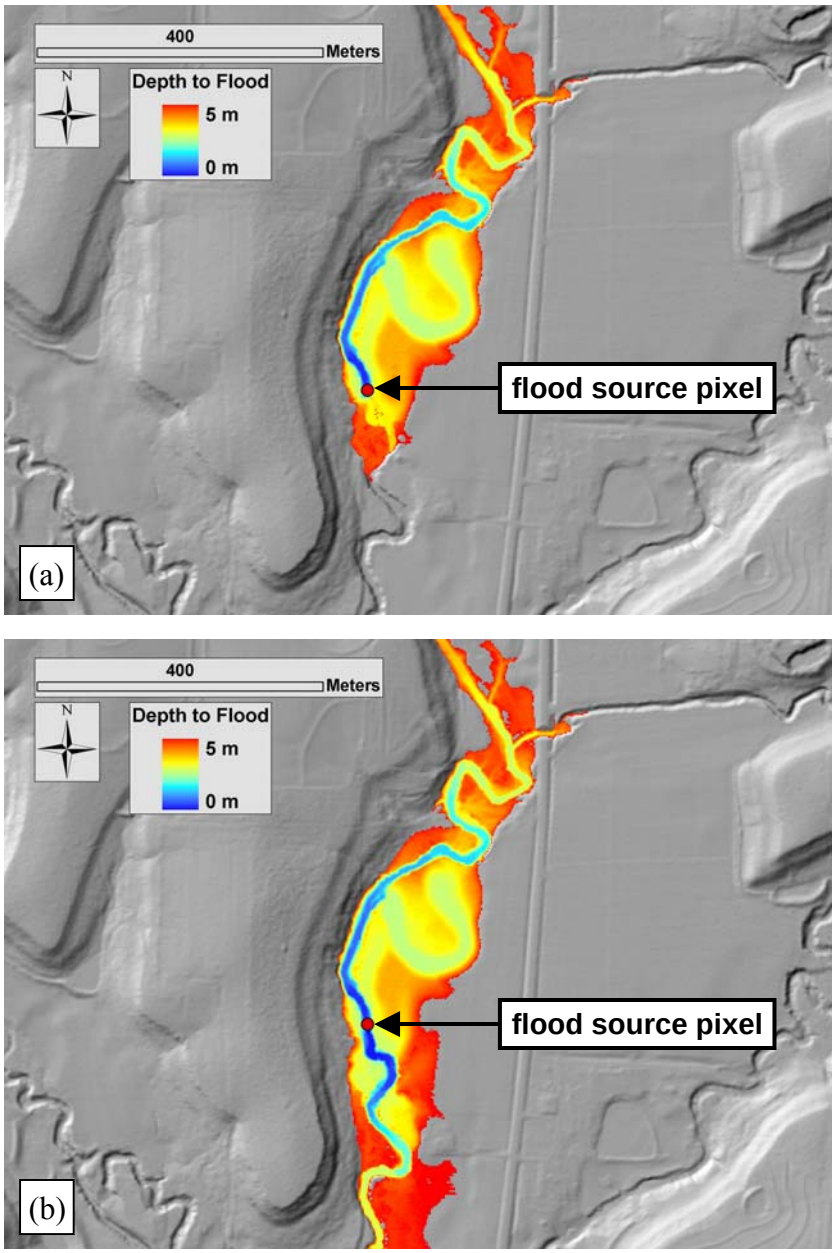


Figure 3.21. The steady state floodplain is shown, computed for a single stream pixel using the FLDPLN model with (a) (*h*,*dh*) = (5,5) and (b) (*h*,*dh*) = (5,1). Imagine bisecting these floodplains using the cross-section line through the FSP, so that the backfill area is to the north and the spillover area is to the south of this line. Note the difference between the spillover areas of the floodplain (i.e., downstream from the FSP). The spillover area gets larger as *dh* gets smaller. The backfill areas (upstream from the FSP) are almost identical. This example bears resemblance to the two conceptual *PFE*'s shown in Figure 3.13(a)-(b).

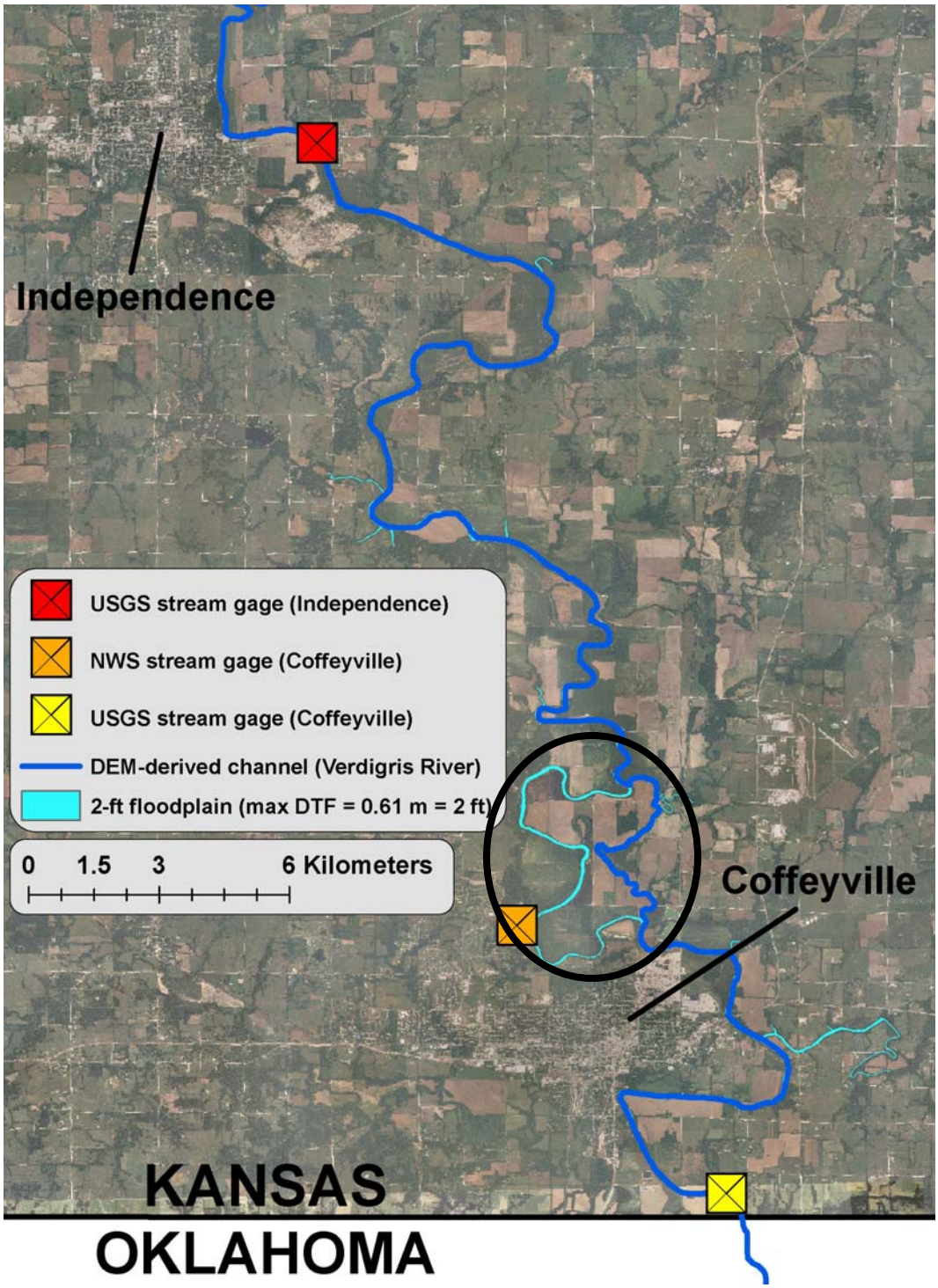


Figure 3.22. Montgomery County, KS, study area. The towns of Independence and Coffeyville are labeled. The backdrop is aerial imagery from the 2005 National Agricultural Imagery Program (NAIP). Three stream gages on the Verdigris River are shown. The circled area is described in the text.

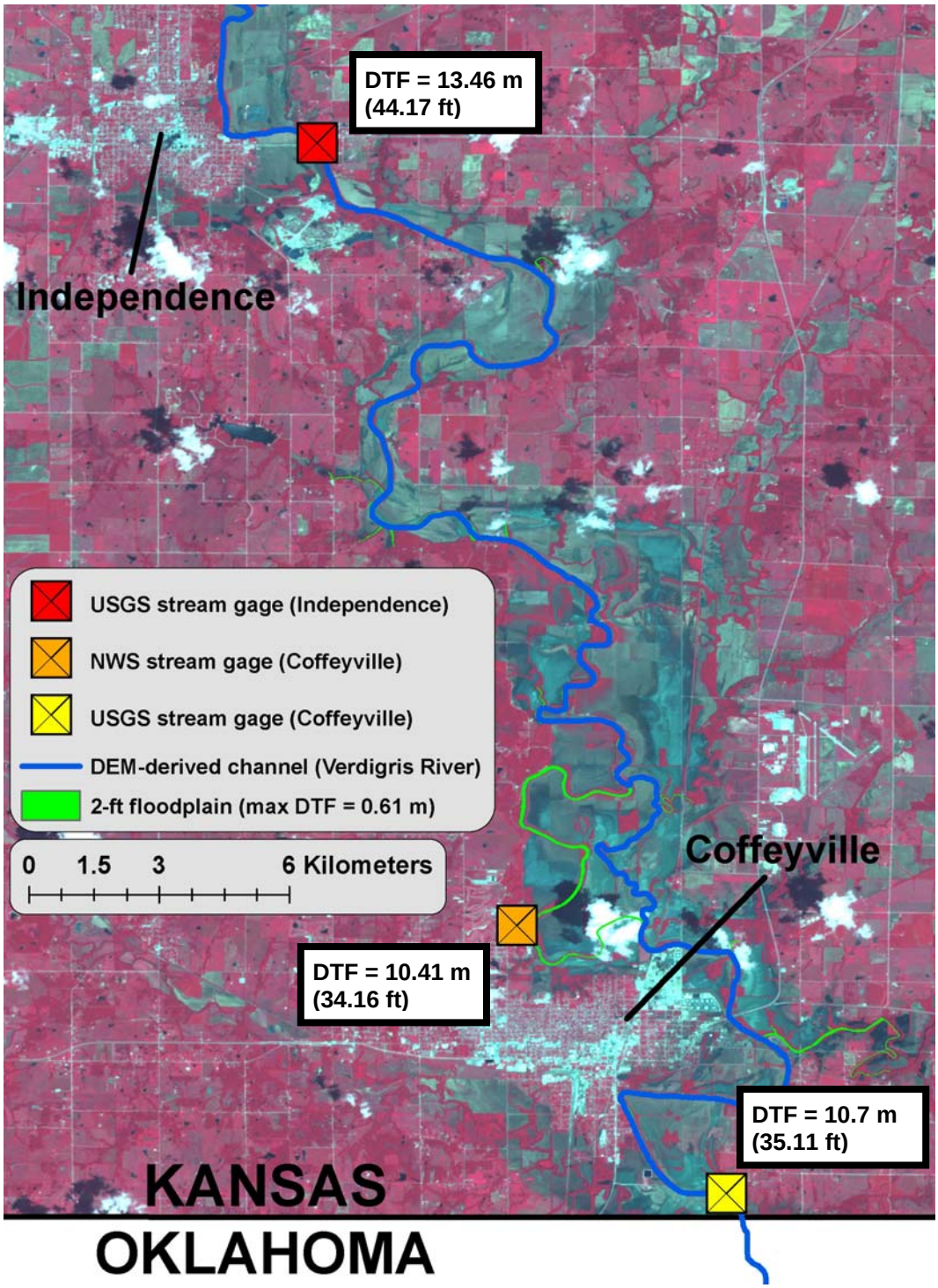


Figure 3.23. Same as Figure 3.22, but the backdrop is now a color-infrared ASTER satellite image captured on July 7, 2007, five days after flood crest. The peak flood "footprint" (floodwater extent) is generally visible in this image. The peak flood height for each gage is indicated, transformed to a DTF value.

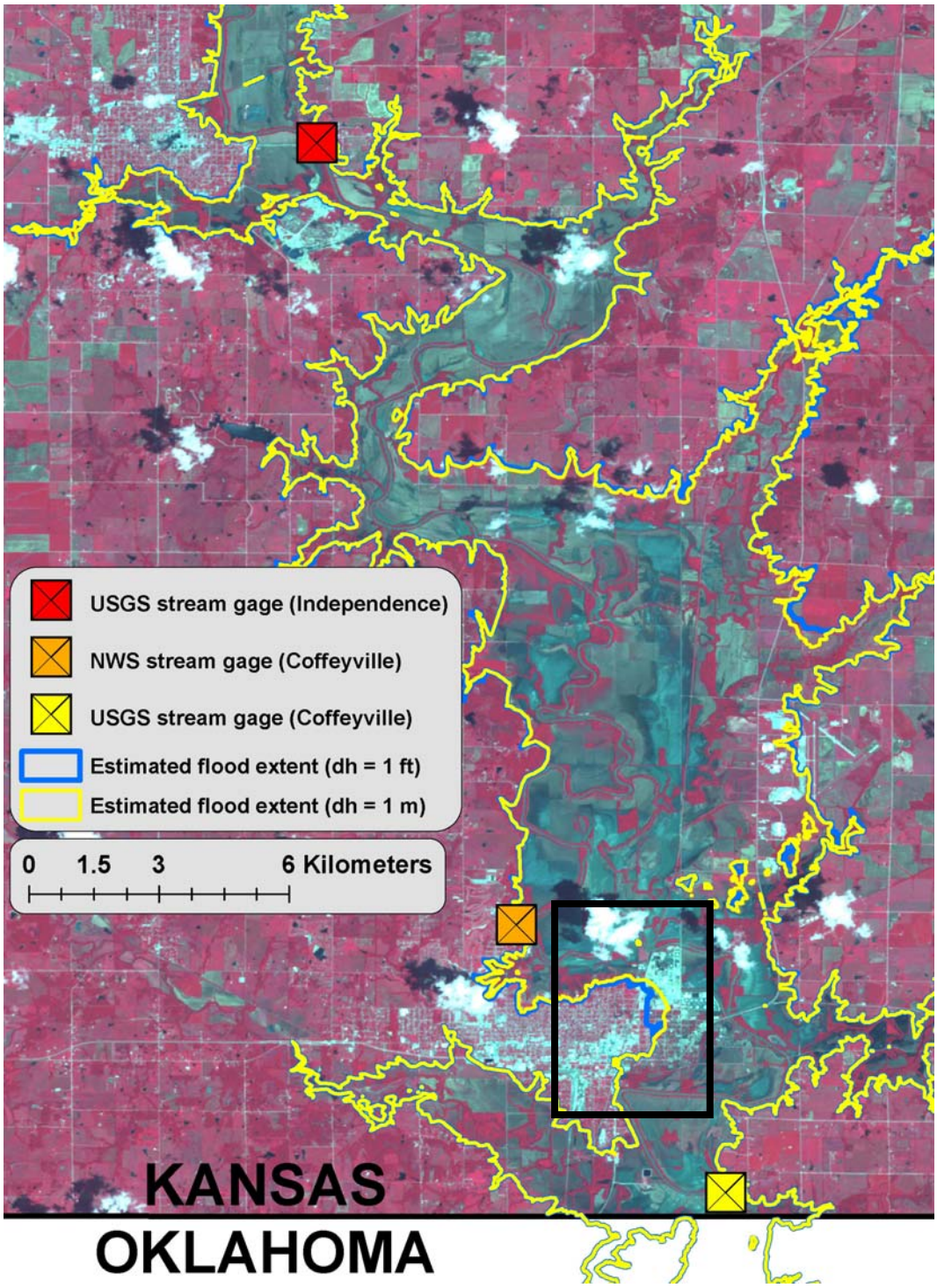


Figure 3.24. Using peak gage height data from three gaging stations, two flood extents estimates were generated, one using FLDPLN with *dh* = 0.3048 m (1 ft) and one with *dh* = 1 m. The boxed area contains one of the largest discrepancies between the two estimates, and is shown in Figure 3.25.

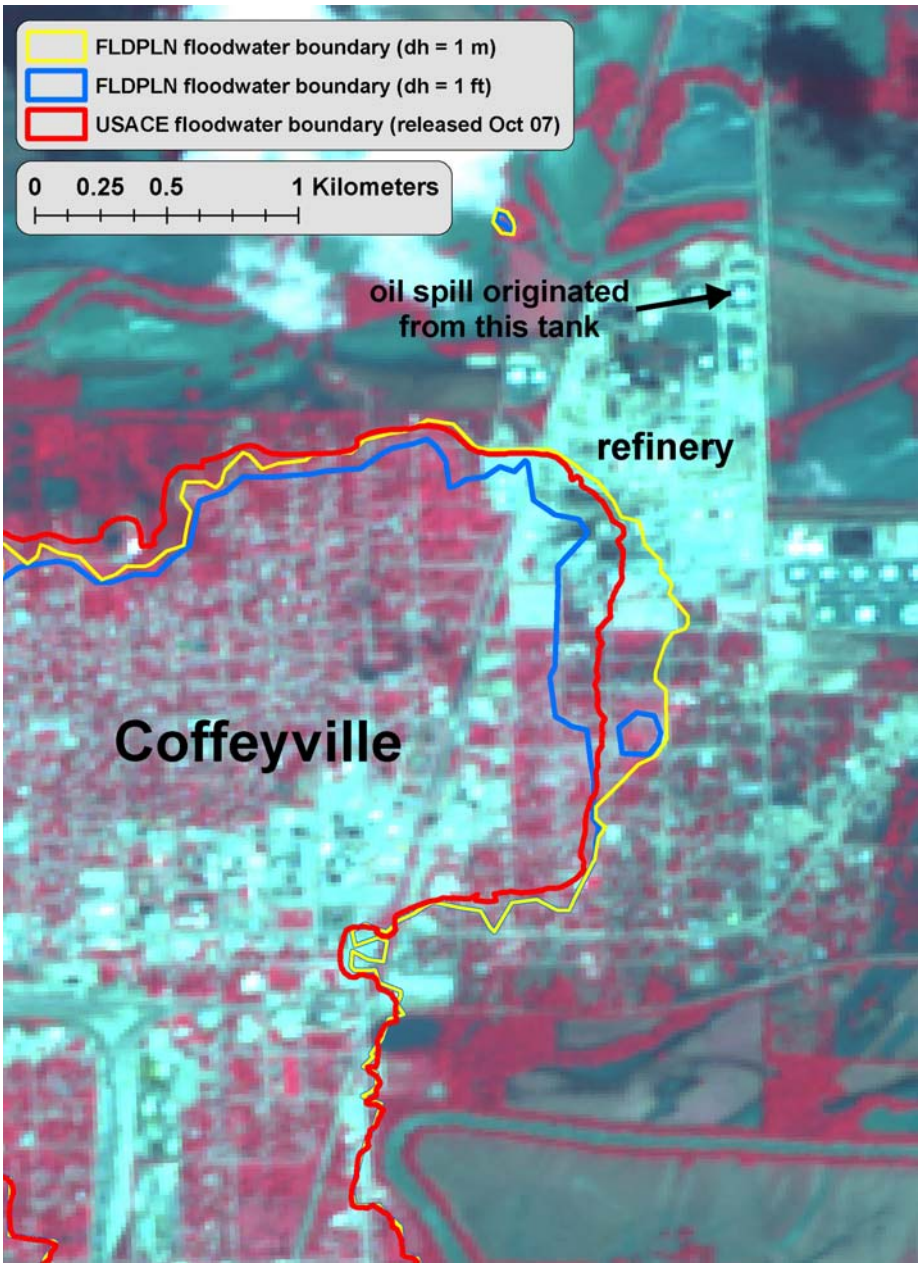


Figure 3.25. Three peak flood extent estimates for the Coffeyville area are shown, along with some contextual information regarding the oil spill that occurred on June 30, 2007, during this flood event. The FLDPLN (*dh* = 1 ft) estimate indicates that more of Coffeyville was inundated by floodwaters than the FLDPLN (*dh* = 1 m) estimate. Shown merely as a reference, the USACE estimate (which was developed using a sparse sampling of high water marks recorded during and after the flood event) splits the difference between the two FLDPLN estimates in the area where the two FLDPLN estimates differ the most.

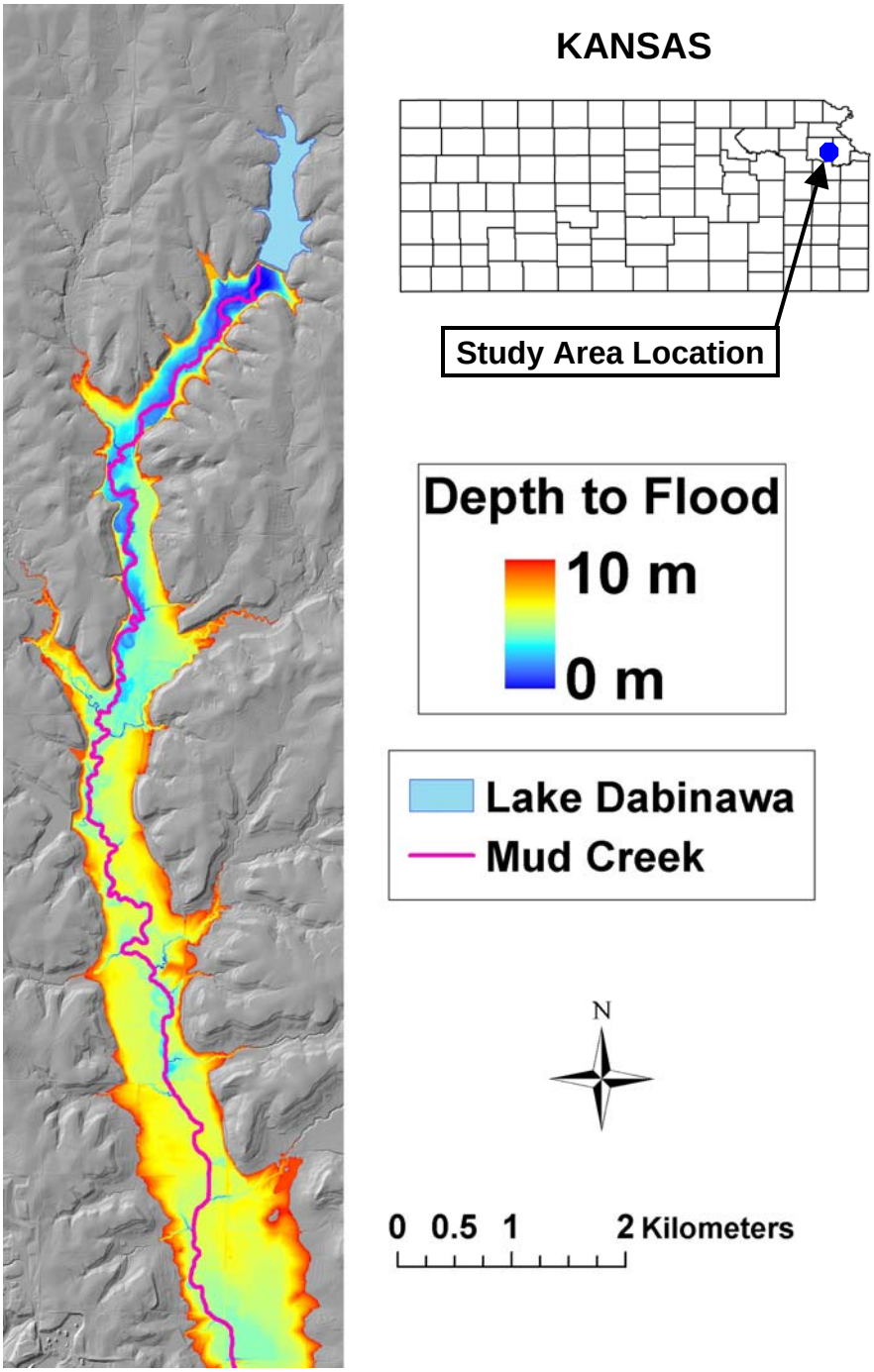


Figure 3.26. The 10-m steady-state floodplain is shown for roughly 10 km of the Mud Creek stream reach. This floodplain was created using the FLDPLN model with (h,dh) = (10,0.5) applied to 2-m resolution LIDAR DEM data.

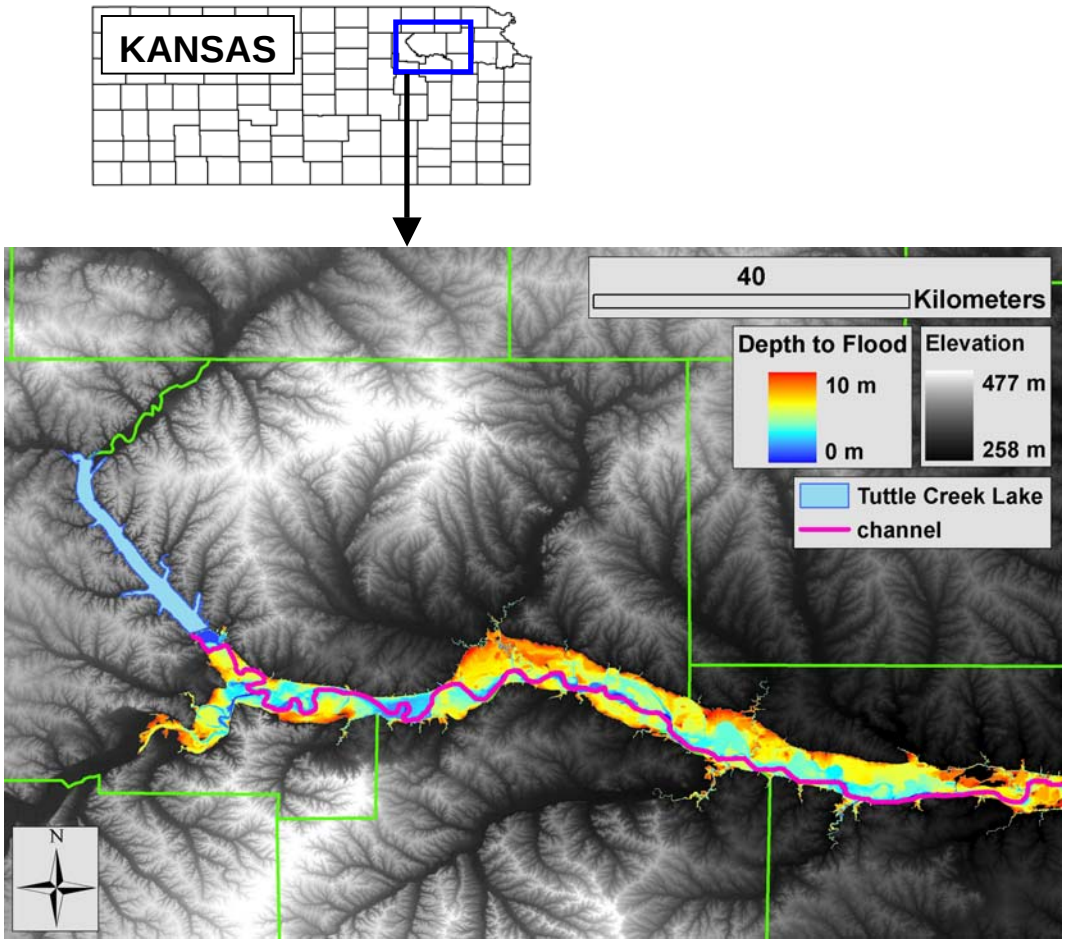


Figure 3.27. The 10-m steady-state floodplain is shown for approximately 100 km of continuous stream reach of the Big Blue River and the Kansas River below Tuttle Creek Lake. This floodplain was created using the FLDPLN model with (h,dh) = (10,1) applied to 10-m resolution DEM data from the NED. Green lines denote Kansas county boundaries. The hole in the floodplain near the right edge of the study area shows the effects of the levy in Topeka as a flood deterrent.

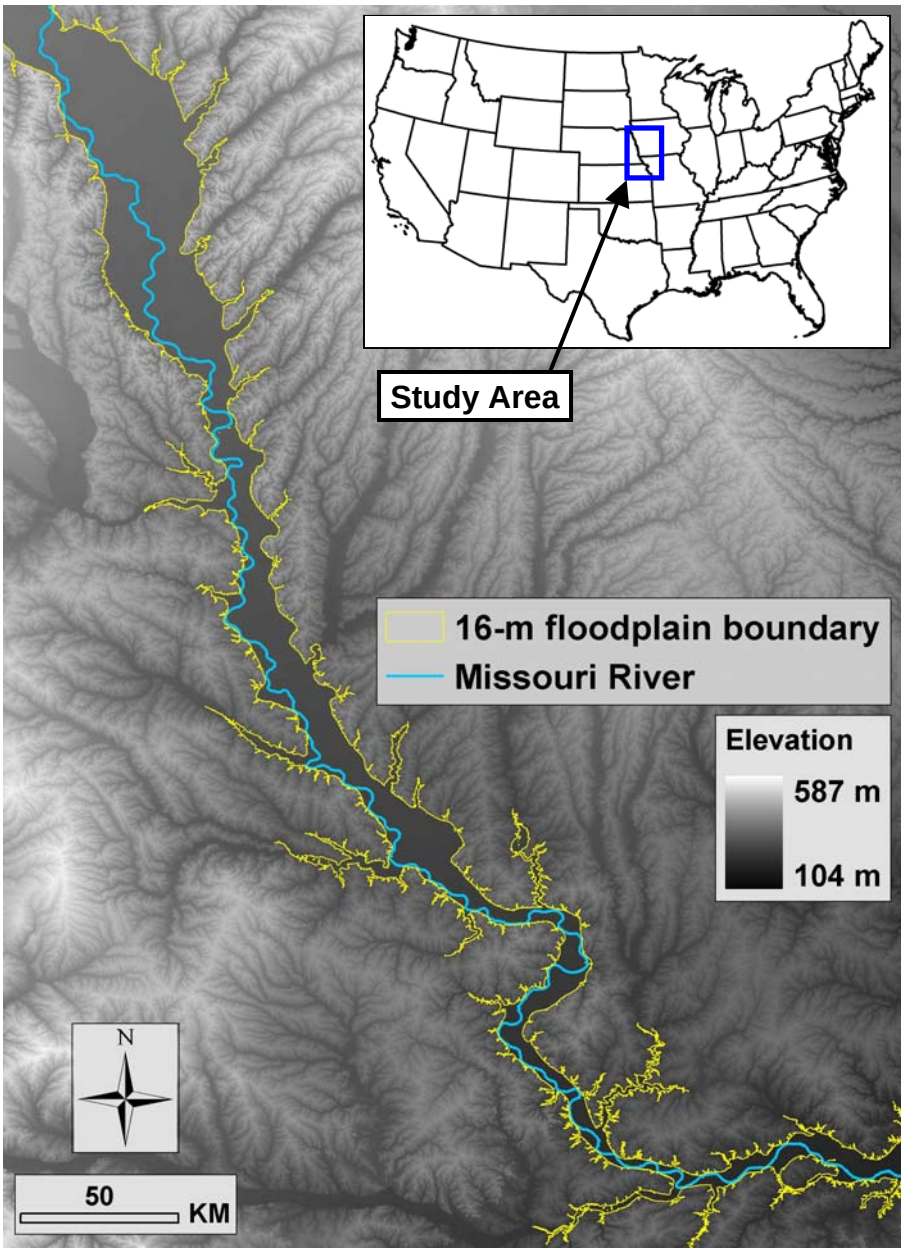


Figure 3.28. The 16-m floodplain boundary is shown for approximately 500 km of the Missouri River in the central U.S. This floodplain was created using the FLDPLN model with (h,dh) = (16,2) applied to 30-m resolution DEM data from the NED. Due to the relatively coarse resolution of the DEM data, Step 7 was not used in this application of FLDPLN. Only the floodplain boundary is shown in this example to illustrate the utility of FLDPLN for historic floodplain identification. The river valley for this part of the Missouri River is the visibly distinct, dark band enveloping the river course, and appears to be well delineated by the 16-m floodplain.

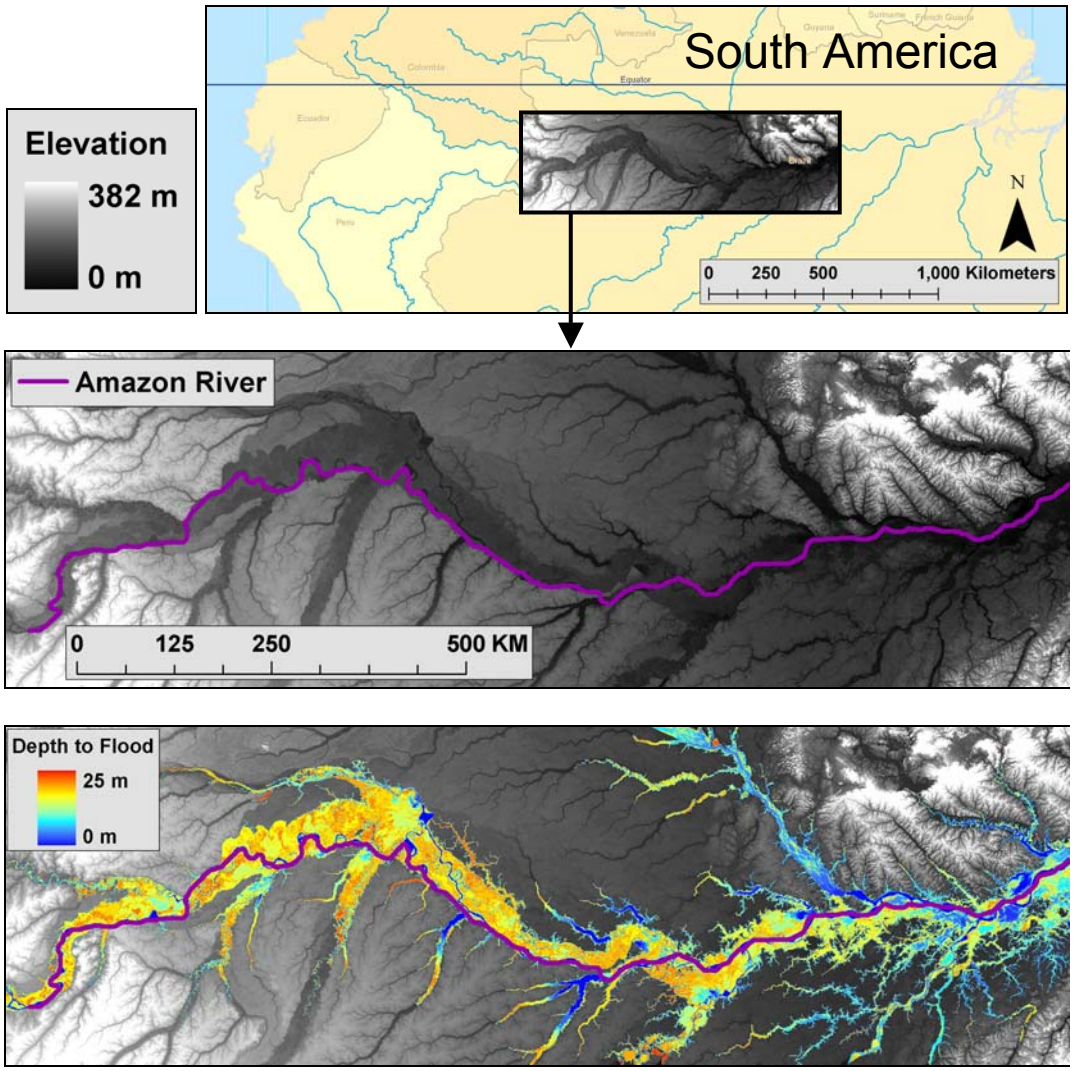


Figure 3.29. The 25-m floodplain boundary is shown for approximately 1700 km of the Amazon River in the Brazil. This floodplain was created using the FLDPLN model with (h,dh) = (25,5) applied to 90-m resolution DEM data from the NED. Due to the coarse resolution of the DEM data, Step 7 was not used in this application of FLDPLN. According to the DEM data, there is only a 17-m drop in elevation of the river surface from the western edge to the eastern edge of the study area. Due to this low relief, the floodplain is quite complex, especially on the west side of the study area.

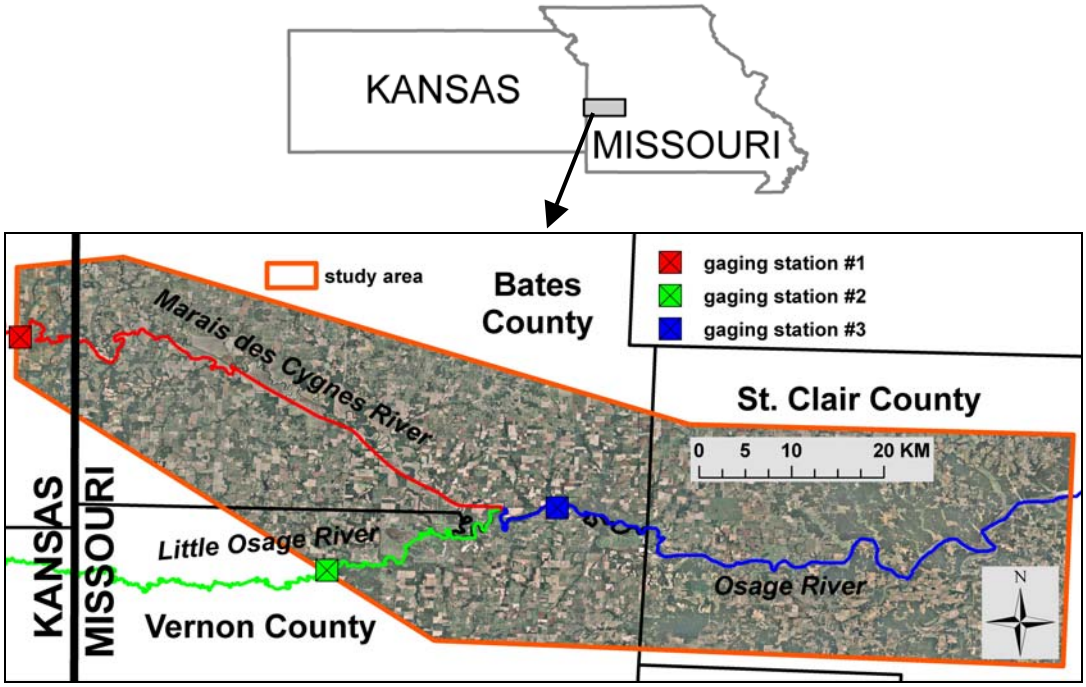


Figure 3.30. The validation study area is shown. Background imagery (shown in true color) is from the 2006 NAIP image archive.

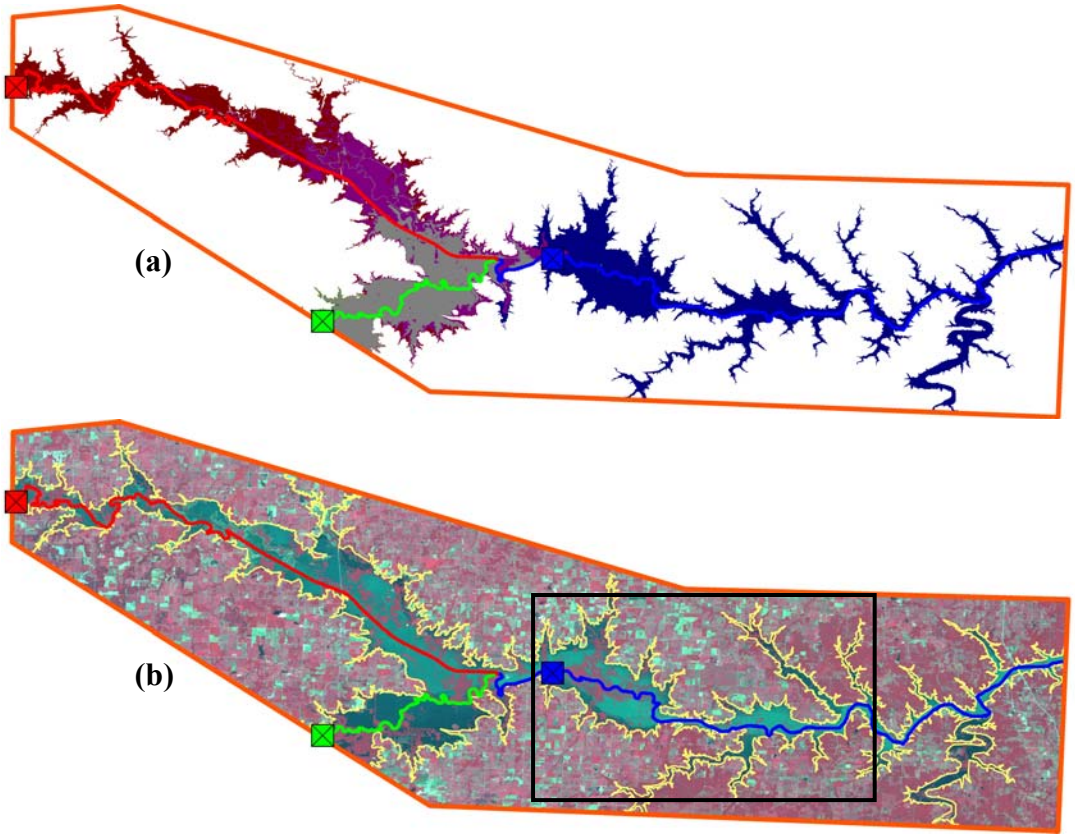


Figure 3.31. (a) Composite image of the three, segment-specific flood zone extents (color bands coincide with RGB stream segment colors). Each extent was generated using the FLDPLN model with $dh = 1$, using the crest mean daily gage height measured at each respective gaging station to set the segment-specific maximum flood depth values (h values). 30-m resolution DEM data from the NED provided the topographic data for the analysis, and consequently Step 7 from the FLDPLN algorithm was not used. Subplot (b) shows a post-flood, color infrared (false color RGB using Landsat-5 spectral bands 4-3-2) image collected by Landsat-5, which also has 30-m resolution. The exterior perimeter of the merged flood zone extent is shown in yellow. The modeled flood extent had 87.2% accuracy when compared to a manual delineation of the extent using the Landsat-5 image. The boxed area is used in Figure 3.32.

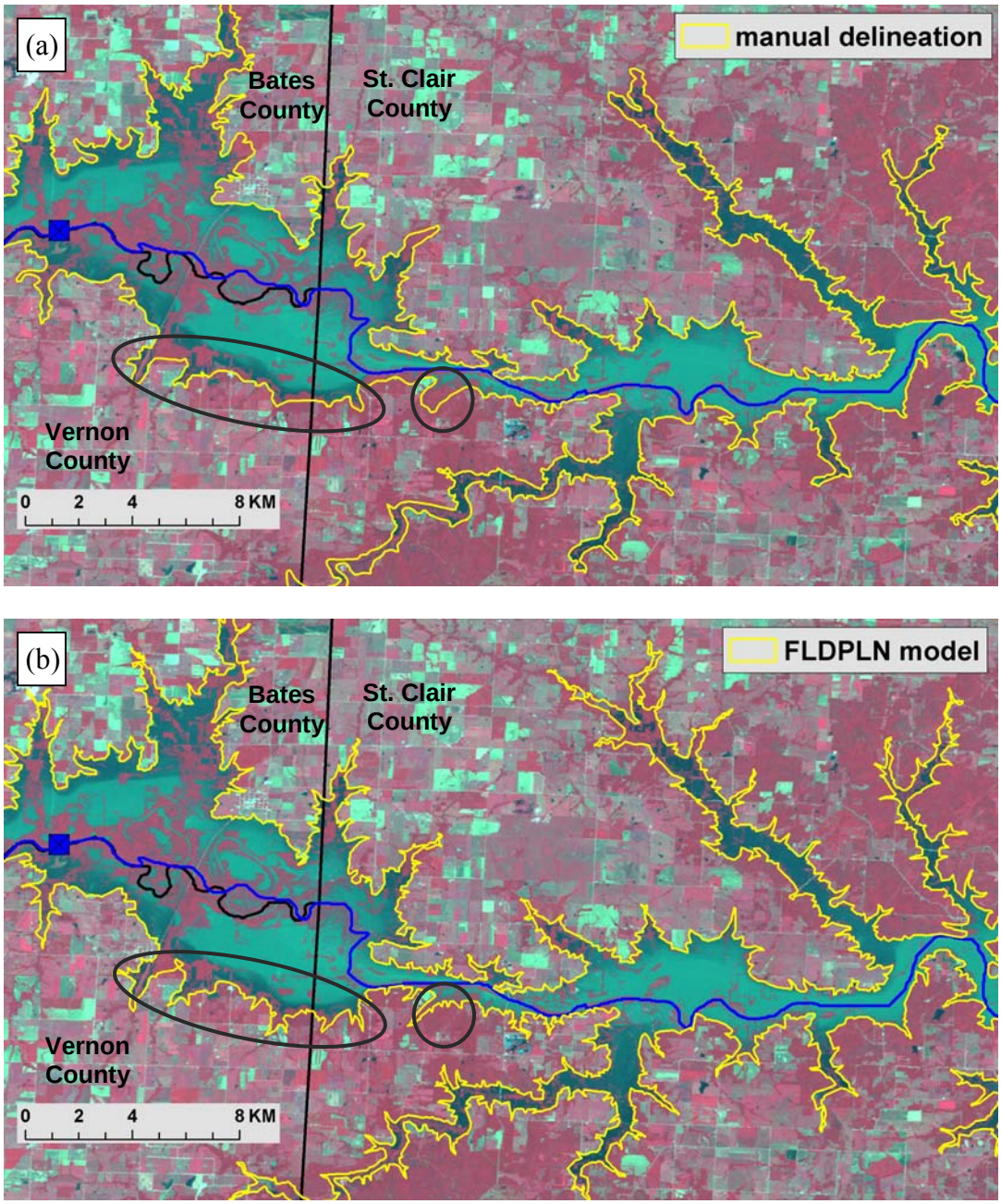


Figure 3.32. A subset of the flooded area from the validation study is shown. The manually delineated flood boundary is depicted in (a), and the boundary predicted using the FLDPLN model is depicted in (b). Dark red regions in the Landsat-5 color image correspond to forested areas, which can inhibit visual identification of floodwater boundaries (e.g., see the two circled areas). Consequently, the modeled extent could more accurately represent the floodwater boundary in these areas than the manually delineated extent.

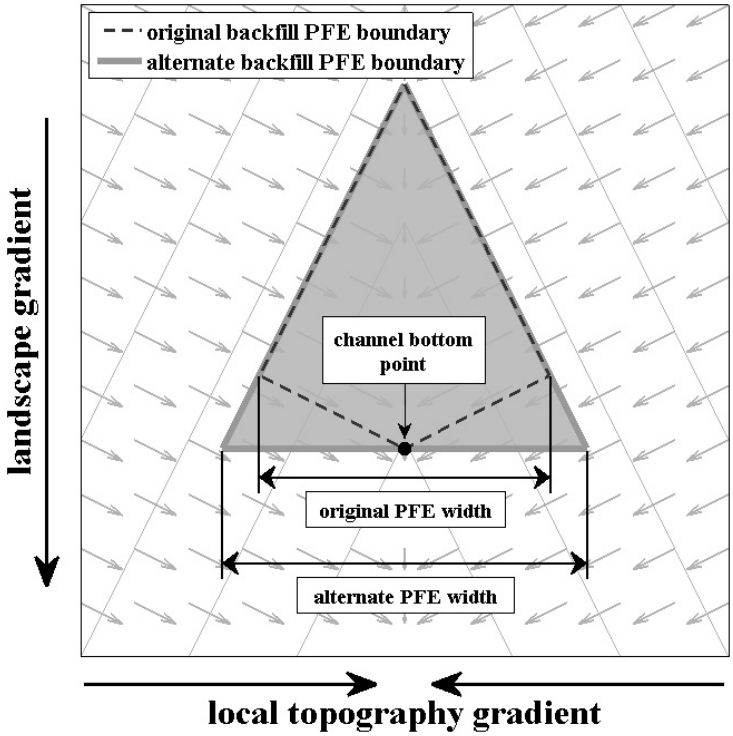


Figure A3.1. Two backfill PFEs are shown for a single point at the bottom of pitched channel *S*. The "original backfill PFE" reflects the original specification from Section 3.4, with the downstream reservoir boundary (i.e., the hypothetical dam face) following topographic gradient lines. The dam face for the "alternate backfill PFE" is instead orthogonal to the flow channel. Note that the alternate specification results in a wider PFE.

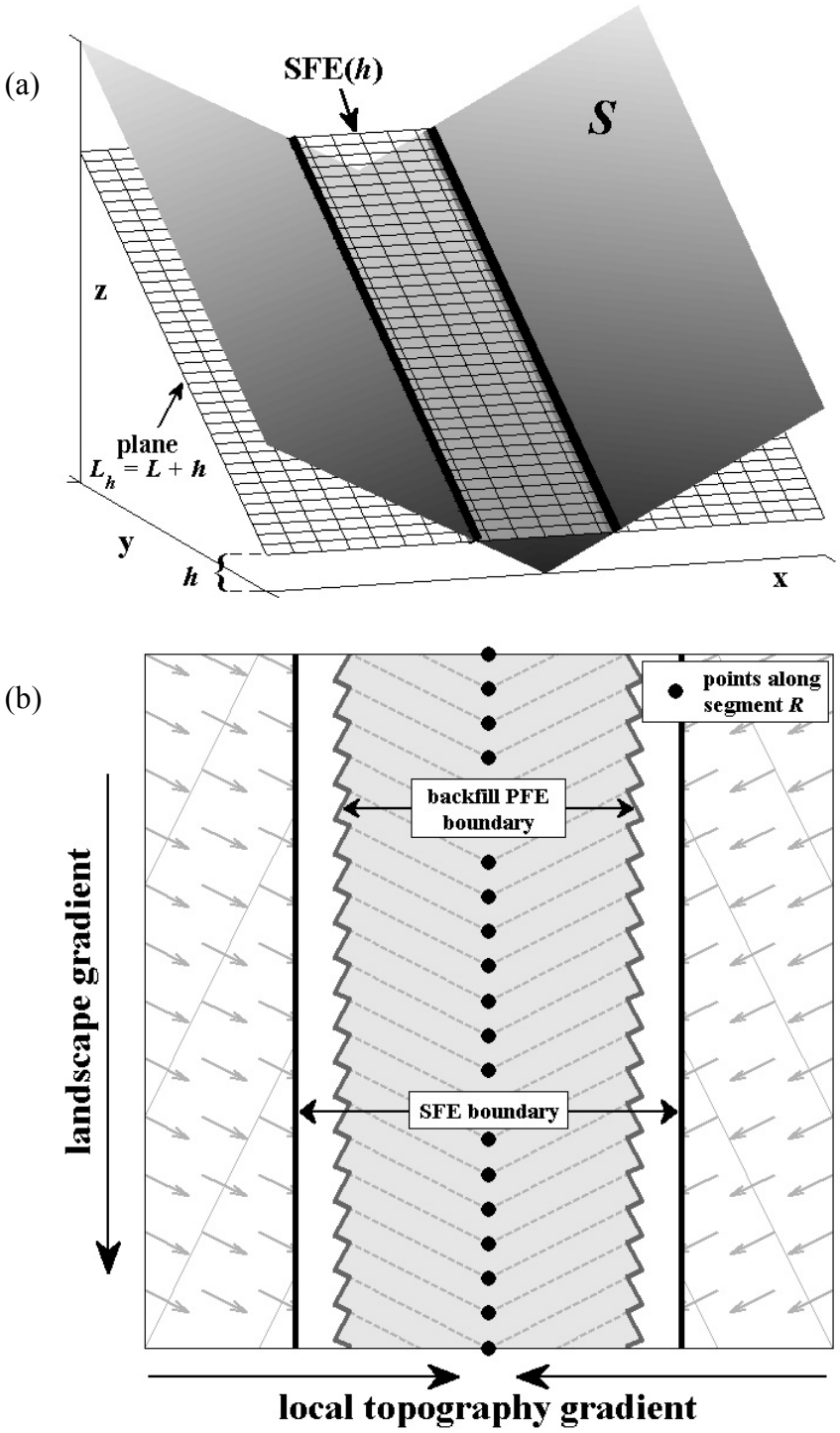


Figure A3.2. The simple flood extent at flood depth h (SFE(h)) for pitched channel S is shown in (a). Subplot (b) shows the areal extent comparison between the SFE and the backfill PFE at the same flood depth.

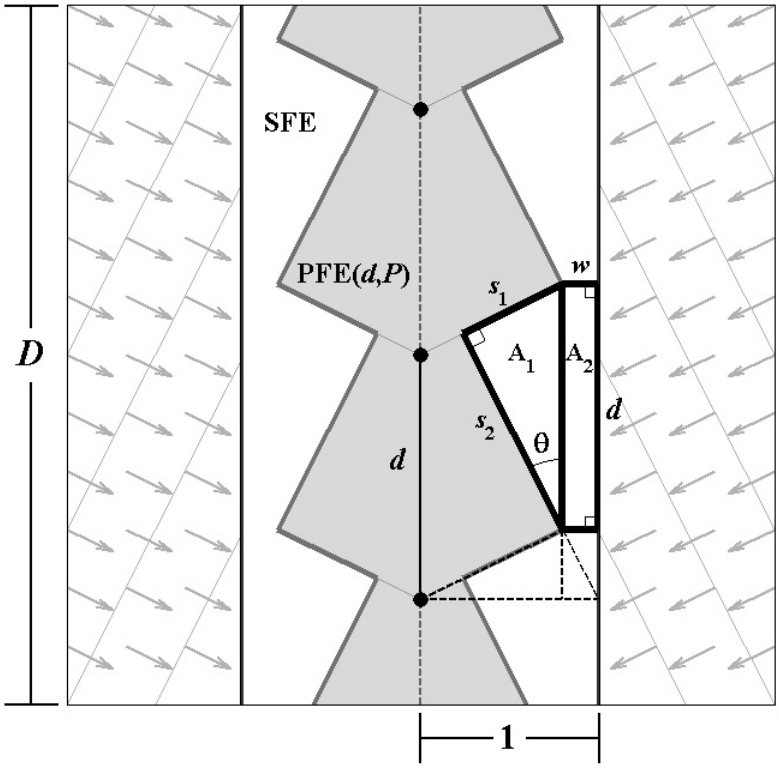


Figure A3.3. The normalized SFE for the pitched channel is shown, with corresponding PFE(*d*,*P*) overlaid. The error between the SFE and PFE(*d*,*P*) is periodic with the evenly spaced points along the channel bottom, and the error is symmetric about the channel bottom. Consequently, the error analysis can be reduced to an examination of areas A_1 and A_2.

References

Akaike, H. (1970). Statistical Predictor Identification. *Annals of the Institute of Statistical Mathematics*, 22: 203-217.

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in *2nd International Symposium on Information Theory*, Ed. B.N. Petrov and F. Csaki, pp.267-281. Akademia Kiado, Budapest.

Allen, D.M. (1974). The Relationship Between Variable Selection and Data Augmentation and a Method for Prediction. *Technometrics*, 16(1): 125-127.

Bates, P.D., and A.P.J. De Roo (2000). A simple raster-based model for flood inundation simulation. *Journal of Hydrology*, 236: 54-77.

Bayley, P.B. (1995). Understanding Large River-Floodplain Ecosystems. *BioScience*, 45(3): 153-158.

Bendel, R.B. (1973). Stopping rules in forward stepwise-regression. Ph.D. thesis, Biostatistics Dept., Univ. of California at Los Angeles.

Bradbrook, K. (2006). JFLOW: a multiscale two-dimensional dynamic flood model. *Water and Environment Journal*, 20: 79-86.

Bradbrook, K., S.N. Lane, S.G. Waller, and P.D. Bates (2004). Two dimensional diffusion wave modeling of flood inundation using a simplified channel representation. *International Journal of River Basin Management*, 2(3): 211-223.

Bradbrook, K., S. Waller, and D. Morris (2005). National Floodplain Mapping: Datasets and Methods—160,000 km in 12 months, *Natural Hazards*, 26: 103-123.

Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone (1984). *Classification and Regression Trees*. Belmot, CA: Wadsworth.

Buijse, A.D., H. Coops, M. Staras, L.H. Jans, G.J. Van Geest, R.E. Grifts, B.W. Ibelings, W. Oosterberg, and F.C.J.M. Roozen (2002). Restoration strategies for river floodplains along large lowland rivers in Europe. *Freshwater Biology*, 47: 889-907.

Carter, J.R. (1988). Digital Representations of Topographic Surfaces. *Photogrammetric Engineering & Remote Sensing*, 54(11): 1577-1580.

Cook, R.D., and S. Weisberg (1982). *Residuals and Influence in Regression*. New York: Chapman & Hall.

Davison, A.C., and D.V. Hinkley (1997). *Bootstrap Methods and their Application*. New York: Cambridge University Press.

Draper, N.R., and H. Smith (1998). *Applied Regression Analysis, Third Edition*. New York: Wiley-Interscience.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1): 1-26.

Efron, B. (1982). *The Jackknife, the Bootstrap, and Other Resampling Plans*. Philadelphia: Society for Industrial and Applied Mathematics.

Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382): 316-331.

Efron, B., and R.J. Tibshirani (1993). *An Introduction to the Bootstrap*. New York: Chapman & Hall/CRC.

Efron, B., and R. Tibshirani (1997). Improvements on Cross-Validation: The .632+ Bootstrap Method. *Journal of the American Statistical Association*, 92(438): 548-560.

Feller, W. (1957). *An Introduction to Probability Theory and Its Applications: Volume I, Second Edition*. New York: John Wiley & Sons, Inc.

Fread, D.L. (1993). Chapter 10: Flow Routing, in *Handbook of Hydrology*, Edited by Maidment, D.R. New York: McGraw-Hill (pp.10.1-10.36).

Fread, D.L., and J.M. Lewis (1988). "FLDWAV: A Generalized Flood Routing Model." Hydraulic Engineering, Proceedings of 1988 Conference, HY Div, American Society of Civil Engineers, Colorado Springs, CO, pp.668-673.

Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. *Journal of the American Statistical Association*, 70(350): 320-328.

Gesch, D. M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, and D. Tyler (2002). The National Elevation Dataset. *Photogrammetric Engineering & Remote Sensing*, 68(1): 5-11.

Gray, H.L., and W.R. Schucany (1972). *The Generalized Jackknife Statistic*. Marcel Dekker, Inc., New York.

Henderson, F.M. (1966). *Open Channel Flow*. Upper Saddle River, NJ: Prentice Hall.

Hervouet, J. (2007). *Hydrodynamics of Free Surface Flows: Modeling with the finite element method*. West Sussex, England: John Wiley & Sons, Ltd.

Hjorth, J.S.U. (1994). *Computer Intensive Statistical Methods*. New York: Chapman & Hall/CRC.

Horritt, M.S., and P.D. Bates (2001). Effects of spatial resolution on a raster based model of flood flow. *Journal of Hydrology*, 253: 239-249.

Jenson, S.K., and J.O. Domingue (1988). Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis. *Photogrammetric Engineering & Remote Sensing*, 54(11): 1593-1600.

Kastens, J.H., T.L. Kastens, D.L.A. Kastens, K.P. Price, E.A. Martinko, and R. Lee (2005). Image masking for crop yield forecasting using time series AVHRR NDVI imagery. *Remote Sensing of Environment*, 99(3): 341-356.

Kenward, T., D.P. Lettenmaier, E.F. Wood, and E. Fielding (2000). Effects of Digital Elevation Model Accuracy on Hydrologic Predictions, *Remote Sensing of Environment*, 74: 432-444.

Lea, R.D., and D.D. Diamond (2006). Missouri River Floodplain Modeling & Ownership Database Development: Final Report. Missouri Audubon. July, 2006. Available online at http://www.cerc.usgs.gov/morap/projects/audubon/KC_STJOE MO River Audubon Final Report.pdf.

Lear, J., S. Zheng, and B. Dunnigan (2000). "Flood-prone area delineation using DEMs and DOQs." 2000 ESRI User Conference Proceedings: 20th Annual ESRI International User Conference, July 26-30, 2000, San Diego, CA. URL: http://gis.esri.com/library/userconf/proc00/professional/papers/PAP492/p492.htm

Maidment, D.R., ed. (2002). *Arc Hydro: GIS for water resources*. Redlands, CA: ESRI Press.

Mallows, C.L. (1973). Some Comments on C_p. *Technometrics*, 15(4): 661-675.

McQuarrie, A.D.R., and C. Tsai (1998). *Regression and Time Series Model Selection*. River Edge, NJ: World Scientific Publishing Co. Pte. Ltd.

Miller, Alan (2002). *Subset Selection in Regression, Second Edition*. New York: Chapman & Hall/CRC.

Poole, G.C., J.A. Stanford, S.W. Running, and C.A. Frissel (2000). "A Linked GIS/Modeling Approach to Assessing the Influence of Flood-plain Structure on Surface- and Ground-water Routing in Rivers", 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs. Banff, Alberta, Canada, September 2-8, 2000.

Quenouille, M.H. (1949). Approximate Tests of Correlation in Time Series. *Journal of the Royal Statistical Society, Series B (Methodological)*, 11(1): 68-84.

Quenouille, M.H. (1956). Notes on Bias in Estimation. *Biometrika*, 43(3/4): 353-360.

Schwarz, G. (1978). Estimating the Dimension of a Model. *Annals of Statistics*, 6: 461-464.

Seber, G.A.F., and A.J. Lee (2003). *Linear Regression Analysis, Second Edition*. Hoboken, NJ: John Wiley & Sons, Inc.

Shao, J. (1993). Linear Model Selection by Cross-Validation. *Journal of the American Statistical Association*, 88(422): 486-494.

Shao, J. (1996). Bootstrap Model Selection. *Journal of the American Statistical Association*, 91(434): 655-665.

Shao, J. (1997). An Asymptotic Theory for Linear Model Selection. *Statistica Sinica*, 7: 221-264.

Shao, J., and D. Tu (1995). *The Jackknife and Bootstrap*. New York: Springer-Verlag, Inc.

Shao, J., and C.F.J. Wu (1989). A General Theory for Jackknife Variance Estimation. *The Annals of Statistics*, 17(3): 1176-1197.

Shibata, R. (1984). Approximate Efficiency of a Selection Procedure for the Number of Regression Variables, *Biometrika*, 71(1): 43-49.

Sparks, R.E., P.B. Bayley, S.L. Kohler, and L.L. Osborne (1990). Disturbance and Recovery of Large Floodplain Rivers. *Environmental Management*, 14(5): 699-709.

Tockner, K., J.V. Ward, and J.A. Standford (2002). Riverine floodplains: present state and future trends. *Environmental Conservation*, 29(3): 308-330.

Stein, C. (1960). *Multiple Regression*, Contributions to Probability and Statistics (Olkin, I., *et al.*, ed.). Stanford, CA: Stanford University Press.

Stone, M. (1974). Cross-validatory Choice and Assessment of Statistical Prediction (with discussion). *Journal of the Royal Statistical Society, Series B*, 36(2): 111-147.

Tukey, J.W. (1958). Bias and Confidence in Not-quite Large Samples (Abstract), *The Annals of Mathematical Statistics*, 29(2): 614.

Turner, M.G., S.E. Gergel, M.D. Dixon, and J.R. Miller (2004). Distribution and abundance of trees in floodplain forests of the Wisconsin River: Environmental influences at different scales. *Journal of Vegetation Science*, 15: 729-738.

USACE (2007). *Data Recovery of the June-July 2007 Flood in Region VII*. Report generated by URS Group, Inc., Gaithersburg, MD. October 2007. 416 pp. Contract No. W912BV-04-D1008 TO 18.

USACE HEC (2000). *HEC-RAS Hydrologic Modeling System: Technical Reference Manual*. March 2000. Available online at http://www.hec.usace.army.mil/software/hec-hms/documentation/CPD-74B_2000Mar.pdf.

USACE HEC (2002). *HEC-RAS River Analysis System: Hydraulic Reference Manual*, Version 3.1. November 2002. Available online at http://www.hec.usace.army.mil/software/hec-ras/documents/hydref/index.html.

Watts, T., J. Atwood, K. Price, and J. Kastens (2005). "The Big Picture—Satellite Remote Sensing Applications in Rangeland Assessment and Crop Insurance." USDA Agricultural Outlook Forum 2005, February 24-25, 2005, Arlington, VA. Speech Booklet 2, 26 pp. (Invited paper)

Wei, C.Z. (1992). On Predictive Least Squares Principles. *The Annals of Statistics*, 20: 1-42.

Wu, C.F.J. (1986). Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis (with discussion). *The Annals of Statistics*, 14(4): 1261-1295.

Wu, C.F.J. (1990). On the Asymptotic Properties of the Jackknife Histogram. *The Annals of Statistics*, 18(3): 1438-1452.

Zhang, P. (1993). Model Selection Via Multifold Cross Validation. *The Annals of Statistics*, 21(1): 299-313.