Implementation code for
“A leaky-conduit model of flow in karstic aquifers”

Eric Chicken
Department of Statistics
Florida State University, Tallahassee FL

David E. Loper
Department of Civil and Environmental Engineering
FAMU/FSU College of Engineering, Tallahassee FL

The code used to implement the procedures in the manuscript “A leaky-conduit model of flow in karstic aquifers” is presented and described.

1. Introduction  This manuscript is a companion to Loper and Chicken (2011) and the related manuscript Loper and Chicken (2009). Those two papers propose a method for modeling the flow in a specified portion of a karstic aquifer under specific conditions. This manuscript provides information about the code used to implement the model. The inputs necessary to replicate the results in Loper and Chicken (2011) are given and described. The outputs to the code are also detailed.

The software used is R, R Development Core Team (2008). R is a freely available, open source software much used in the academic and scientific communities. This code is archived at Loper and Chicken (2010).

This manuscript is organized as follows. Section 2 provides instructions on obtaining R and the R code for the model. Section 3 gives a description of the routines needed to implement the model. Section 4 describes the inputs to the code and outputs from running the code. Section 5 reviews the examples from Loper and Chicken (2011) from the software point-of-view.

2. Obtaining the code  To obtain R, visit web page The R Project for Statistical Computing at http://www.r-project.org. This web page provides links for downloading R and help in installing it. The code to implement the proposed model requires an additional package be installed with R. This is the package “geometry”. The rest of this manuscript assumes that R and the geometry package are properly installed.

The code to implement the model, referred to from now on as KFM, is available from the archival web page http://stat.fsu.edu/~chicken/KFM. The only necessary download from this page is the file KFM.RData. R will run under Linux, Windows and MacOS X operating systems at this time. This manuscript assumes that the user is running the Windows operating system. However, the file KFM.RData is an R image that will work under any of the three operating systems.

Once this file is downloaded, it should be placed in a new, empty folder. Double-clicking the file will start R and load the KFM R image. All the code necessary to implement the model as
given in Loper and Chicken (2011), as well as the input matrices needed to replicate the examples in that manuscript, are now available.

3. Routines The main software routine is $KFM$. It calls subroutines, one of which calls an additional subroutine. The hierarchy of the routines is as follows,

$$KFM \text{ calls subroutines } map, \text{ plot.map and } td.flux$$

$$td.flux \text{ calls subsubroutine } h.lin$$

Using indentations to reflect subroutines, this is represented as

$$KFM$$
- $map$
- $plot.map$
- $td.flux$
  -- $h.lin$

Brief descriptions of the purpose of these routines are

$KFM$. The main routine. Sends user inputs to appropriate subroutines and gathers outputs from the subroutines for user interpretation.

$map$. Creates a system of nodes, elements and connections. Stores information about the system in a series of matrices.

$plot.map$. Provides a map of the connections, elements and nodes. An optional subroutine.

$td.flux$. Calculates the time-dependent flow within the system map.

$h.lin$. An iterative subroutine to estimate the node heads as they change over time.

4. Inputs and outputs There are multiple user inputs to the model. They are used to specify the structure of the system map and provide changing system conditions over time. All the inputs are provided to the routine $KFM$ as arguments.

$doplot, n.c, n.e, n.n, a$. This will plot a basin map. $n.c, n.e, n.n$ display connection, element, node numbers on map. $a$ is the offset of these numbers on the graph. $a$ is numeric, the others are Boolean ($T, F$). For example, $KFM(doplot=T, n.n=T, a=350)$ will give the output shown in Figure 1. This uses the nodes as defined in L&C10.

$K$. The number of time periods. If a recharge matrix is specified, $K$ is the number of columns in the matrix – 2.
The length of a time step in seconds. The default value is one day, or 24*60*60 seconds.

**recharge.** The recharge (due to rainfall) to the elements for $K$ time periods. It is a matrix with a row for each element, and $(K + 2)$ columns. Column 1 is element identifier, column 2 is steady state recharge, and columns 3 through $(K + 2)$ contain the recharge for the next $K$ time periods. If no variation of recharge is specified (recharge=NULL) then the recharge is set to be steady state for the entire period. In this case, the recharge is a set value $R$ for each time period and element.

**spring.** A user-specified head at the spring. It is a vector of length $K$, a value for each of the $K$ time periods. The default value of NULL calculates the spring head based on the other inputs.
$b.nodes$, $i.nodes$. The set of nodes is given by these two matrices, each with two columns specifying the coordinates of the node locations. $b.nodes$ contains coordinates of the nodes for the boundary and the spring. $i.nodes$ contains coordinates of the interior nodes.

$\phi$. The matrix permeability; this is a constant set equal to 0.3 in the examples.

$tol$. A small positive number used in testing the convergence of the iteration algorithm for the time-dependent estimation of the node heads.

$Tr$. This is $T$ in the text. The transmissivity.

$f$. The friction factor.

$h.s, h.1, h.2$. User-specified values describing the water-table elevations that are used to determine the steady state node heads.

$A$. Area of the springshed.

$Q.s$. Steady state flow at the spring.

$R$. Recharge rate. By default, this is $Q.s/A$.

To run the main routine, a command is entered into the R console, for example:

\[ KFM(recharge=recharge.1) \]

$recharge$ is an argument for the function $KFM$, while $recharge.1$ is a specific value for that argument. This command will send the output to the R console screen. To save the output for later use within R, use such a command as

\[ KFM.out = KFM(recharge=recharge.1) \]

where $KFM.out$ is an arbitrary variable name.

The output from $KFM$ is two sets of matrices and/or vectors. The first set is the time dependent outputs, while the second set is map data and diagnostic matrices. The first set can be accessed by $KFM.out$\$Time.Dep$, the second by $KFM.out$\$Map$. Within each of the sets, secondary objects are accessed by $KFM.out$\$Time.Dep$\$Node.Head$ and $KFM.out$\$Map$\$Node.Table$, for example.

The first set consists of the following objects, in order:

Node.Head. A matrix with $(K + 2)$ columns and $N$ rows. The first column is the node identifying number. The second column is the steady state node head. The remaining $K$
columns are the node heads for the \( K \) time periods. This matrix can be accessed by 
\( KFM.out$Time.Dep$Node.Head \).

*Element.Head.* A matrix with \((K + 2)\) columns and \( E \) rows. The first column is the 
element identifying number. The second column is the steady state element head. The 
remaining \( K \) columns are the element heads for the \( K \) time periods. This matrix can be 
accessed by \( KFM.out$Time.Dep$Element.Head \).

*Flux.Add.* A matrix with \((K + 2)\) columns and \( C \) rows. The first column is the connection 
identifying number. The second column is the steady state flux added to the connections. 
The remaining \( K \) columns are the added fluxes to the connections for the \( K \) time periods. 
This matrix can be accessed by \( KFM.out$Time.Dep$Flux.Add \).

*Flux.Mean.* A matrix with \((K + 2)\) columns and \( C \) rows. The first column is the connection 
identifying number. The second column is the steady state mean flux in the connections. 
The remaining \( K \) columns are the mean fluxes in the connections for the \( K \) time periods. 
This matrix can be accessed by \( KFM.out$Time.Dep$Flux.Mean \).

*\( Q \).* A vector of length \((K + 1)\) giving the flux at the spring. The first component is the 
steady state flux, the rest are time dependent. This vector can be accessed by 
\( KFM.out$Time.Dep$Q \).

Within the second set, in order,

*Node.Table.* A matrix with \( N \) rows and five columns. The first column is the node 
identifier, the second contains a 1 if the node is a boundary node or 0 if it is an interior 
ode. Columns 3 and 4 give the coordinates of the node. Column 5 is the distance of the 
ode to the spring. (Additional columns contain steady state node heads and steady state 
flow at a node.) See Appendix C.1.1.1 of L&C10. This matrix can be accessed by 
\( KFM.out$Map$Node.Table \).

*Element.Table.* A matrix with \( E \) rows and 10 columns. The first column is the element 
identifier. The second, third and fourth contain the nodes adjoining the element. The 
fifth, sixth and seventh contain the connections adjoining the element. The eighth is the area of the element, \( A_e \). Column 9 is \( S_e \), the incenter distance of an element. Column 10 contains the distance of the element’s in-center point to the spring. (Additional columns contain element response time, total element side length, steady state element heads, and coordinates of the element’s center.) See Appendix C.1.1.3 of L&C10. This matrix can be accessed by \( KFM.out$Map$Element.Table \).

*Connections.* A matrix with \( C \) rows and eight columns. The first column is the connection 
identifier, the second and third are the distal and proximal nodes adjoining the connection. 
Column 4 is the length of the connection. The entry in column five is 1 if the connection lies on the springshed boundary and 0 if not. The entry in column six is 1 if the connection is activated as a conduit and 0 if not. Column seven contains the cosine of angle the connection makes with the spring. Column eight contains the distance from
the center of the connection to the spring. See Appendix C.1.1.2 of L&C10. This matrix can be accessed by KFM.out$Map$Connections.

*Steady.Conn.* A matrix with \(C\) rows and five columns. The first column contains the connection identifier. Column two contains the steady state hydraulic gradient in the connection. Column three contains the steady state flux added to the connection. Column four contains the mean flux in the connection. Column five contains the estimated cross-sectional area of the connection. See Appendix C.3 of L&C10. This matrix can be accessed by KFM.out$Map$Steady.Conn.

### 5. Examples

Four examples are now provided. These inputs are provided in the KFM.RData R image. These inputs are:

#### N.b.
A set of ten boundary nodes and the spring matching those used in the paper. This is a value passed to the argument *b.nodes*.

#### N.i.
A set of sixteen interior nodes matching those used in the paper. This is a value passed to the argument *i.nodes*.

#### recharge.1.
The drought case, example 1. A \(42 \times 367\) matrix of rainfall on the 42 elements for 365 days. Recall that column 1 is the element identifier and column 2 is steady state value. The first ten days contain the steady state rainfall. The remaining days have no rain. Every element experiences the same recharge rate. This is a value passed to the argument *recharge*.

#### recharge.2.
A \(42 \times 367\) matrix of rainfall for example 2: uniform 3-day rain. Days 11 – 13 receive 30 times the steady value. All other days receive steady state rainfall. Every element experiences the same recharge rate.

#### recharge.3.
A \(42 \times 367\) matrix of rainfall for example 3: rain on element 18. Element 18 receives 100 times the steady value on days 11 – 13. All other elements receive the steady value.

#### spring.1.
A vector of length 367, specifying the spring head for example 4. This is a value passed to the argument *spring*.

A wrapper function is included in the R image to provide the appropriate inputs to KFM and to fashion the output into the graphs from the paper. This function is KFM.ex. To replicate the results in Loper and Chicken (2011), use the commands

\[
KFM.ex(p=1)
\]

The argument \(p\) may take on values 1 through 4 to generate output for examples 6.1 to 6.4 in Loper and Chicken (2010) discussed below. Optional arguments to KFM.ex are phi, tol, Tr and alpha. The default values used in the examples are phi=0.3, tol=0.1, Tr=0.04 and alpha=0.
The function \texttt{KFM.ex} is more complex than just including a call of \texttt{KFM}. The additional lines in the code are used to create visually pleasing graphs, as well as to make the code work for all four examples.

\textbf{Example 1}

The first example, \texttt{KFM.ex(p=1)}, calls \texttt{KFM} with \texttt{recharge=recharge.1}. There is steady rain $2 \times 10^{-8}$ on days 1 through 10, then no rain. The remaining arguments remain at their default values. Let \texttt{KFM.out} be the output of \texttt{KFM}. Figure 2 shows a plot of the head at node 4 (\texttt{KFM.out$Time.Dep$Node.Table[4,]}) and a plot of the spring discharge (\texttt{KFM.out$Time.Dep$Q}).

\textbf{Example 2}

The second example, \texttt{KFM.ex(p=2)}, is similar to the first except for \texttt{recharge=recharge.2}. Steady rain on all days, but 30 times steady rain on days 11, 12 and 13. Figure 3 plots the same two outputs used in Figure 2.

\textbf{Example 3}

The third example, \texttt{KFM.ex(p=3)}, calls \texttt{KFM} with \texttt{recharge=recharge.3}. Steady rain on all days, with 100 times steady rain on days 11, 12 and 13 on element 18 only. The remaining arguments remain at their default values. Figure 4 shows a plot of the head at element 18 (\texttt{KFM.out$Time.Dep$Element.Table[18,]}) and a plot of the spring discharge (\texttt{KFM.out$Time.Dep$Q}).

\textbf{Example 4}

The final example calls \texttt{KFM} with \texttt{spring=spring.1}. The remaining arguments remain at their default values. Figure 5 shows a plot of the head at the spring (\texttt{spring.1}) and a plot of the spring discharge (\texttt{KFM.out$Time.Dep$Q}). The output needed for Figure 5 is generated by a call to \texttt{KFM}: \texttt{KFM(spring=spring.1)}.
Figure 2.
Figure 4.
Figure 5.

Example 4

- Spring Discharge (m$^3$/s)
- Spring Head (m)
References


