MODSIM 8.1: River Basin Management Decision Support System

Technical Appendices

John W. Labadie
Department of Civil and Environmental Engineering
Colorado State University
Ft. Collins, CO 80523-1372
email: labadie@engr.colostate.edu

2010

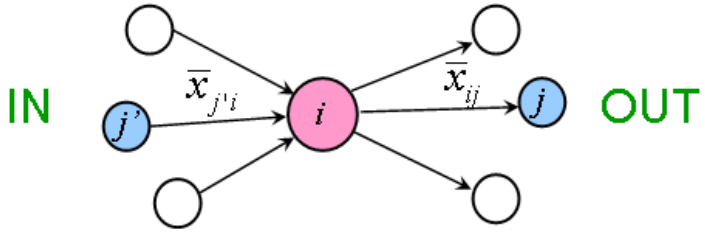
TABLE OF CONTENTS

APPENDIX A
Lagrangian Relaxation Algorithm for Solving Minimum Cost Flow Networks

A.1 Problem Formulation

The minimum cost network flow algorithm employed in MODSIM solves a dual problem that is specialized to take advantage of the linear network structure of the problem. In this formulation, link or arc (*i*,*j*) in MODSIM is designated by the node pair (*i*,*j*) representing the beginning and ending nodes of the link, respectively. This notation implies one unique node pair for each link, and is used for notational convenience only in the following development. The algorithm is actually capable of considering multiple links for the same node pair. The objective function is:

$$\min_{\bar{\mathbf{x}}} \sum_{(i,j)\in A} c_{ij}\bar{x}_{ij}$$



subject to:

$$\bar{l}_{ij} \le \bar{x}_{ij} \le \bar{u}_{ij}$$

$$\underbrace{\sum_{\{j\,|\,(i,j)\in A\}} \bar{x}_{ij}}_{\textbf{OUT}} - \underbrace{\sum_{\{j\,|\,(j,i)\in A\}} \bar{x}_{ji}}_{\textbf{IN}} = 0 \quad , \quad i = 1,\ldots,N$$

where $\bar{x}_{ij}$ represents the flow rate in arc (*i*,*j*) with link parameters $\left[\bar{l}_{ij}, \bar{u}_{ij}, c_{ij}\right]$, c_{ij} is the cost per unit flow for arc (*i*,*j*), *A* is the set of all links or arcs in the network, *N* is the total number of nodes, and $\bar{l}_{ij}$ and $\bar{u}_{ij}$ are the link flow lower and upper bounds, respectively. A transformation can be performed to remove the lower bounds from this problem. Let

$$x_{ij} = \bar{x}_{ij} - \bar{l}_{ij} \quad \text{or} \quad \bar{x}_{ij} = x_{ij} + \bar{l}_{ij}$$
$$u_{ij} = \bar{u}_{ij} - \bar{l}_{ij} \quad \forall (i,j) \in A$$

The transformed objective function is now formulated as:

$$\min_{\mathbf{x}} \sum_{(i,j)\in A} c_{ij}\left[x_{ij} + \bar{l}_{ij}\right]$$

Since the constant term can be removed, the objective is $\min_{\mathbf{x}} \sum_{(i,j)\in A} c_{ij}x_{ij}$

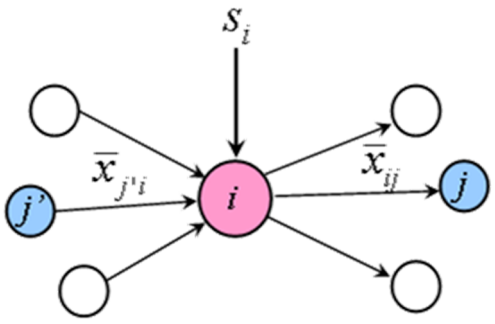
subject to:

$$0 \le x_{ij} \le u_{ij} = \bar{u}_{ij} - \bar{l}_{ij} \quad \forall (i,j) \in A$$

$$\sum_{\{j \mid (i,j) \in A\}} \left[x_{ij} + \bar{l}_{ij} \right] - \sum_{\{j \mid (j,i) \in A\}} \left[x_{ji} + \bar{l}_{ji} \right] = 0 \quad i = 1, \ldots, N$$

or

$$\sum_{\{j \mid (i,j) \in A\}} x_{ij} - \sum_{\{j \mid (j,i) \in A\}} x_{ji} = s_i \quad (i = 1, \ldots, N)$$



where

$$s_i = \sum_{\{j \mid (j,i) \in A\}} \bar{l}_{ji} - \sum_{\{j \mid (i,j) \in A\}} \bar{l}_{ij} \quad (i = 1, \ldots, N)$$

In this formulation, all link parameter data [u_{ij}, c_{ij}] and s_i are assumed to be integer values.

A.2 Lagrangian Relaxation Algorithm

The solution to this problem is based on a Lagrangian relaxation algorithm developed by Bertsekas (1991). Introducing generalized Lagrange multipliers or dual prices p_i, the Lagrangian function is defined as:

$$L(\mathbf{x}, \mathbf{p}) = \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{i=1}^{N} p_i \left[s_i - \sum_{\{j \mid (i,j) \in A\}} x_{ij} + \sum_{\{j \mid (j,i) \in A\}} x_{ji} \right]$$

Note that:

$$\sum_{(j,i)} p_i x_{ji} = \sum_{(i,j)} p_j x_{ij}$$

Therefore:

$$L(\mathbf{x}, \mathbf{p}) = \sum_{(i,j) \in A} \left[c_{ij} - (p_i - p_j) \right] x_{ij} + \sum_{i=1}^{N} s_i p_i$$

The optimality conditions specify that:

$$L(\mathbf{x}^*, \mathbf{p}) = \min_{\mathbf{0} \le \mathbf{x} \le \mathbf{u}} \left\{ \sum_{(i,j) \in A} \left[c_{ij} - (p_i - p_j) \right] x_{ij} + \sum_{i=1}^{N} s_i p_i \right\}$$

$$= \sum_{i=1}^{N} s_i p_i + \min_{\mathbf{0} \le \mathbf{x} \le \mathbf{u}} \left\{ \sum_{(i,j) \in A} \left[c_{ij} - (p_i - p_j) \right] x_{ij} \right\}$$

Instead of attempting to directly solve the original minimum cost network flow problem, the goal is to successively obtain updated dual price vectors **p** that solve the following *dual problem*:

$$\max_{\mathbf{p}} \; h(\mathbf{p})$$

where the dual function can be decomposed into separable optimization problems over each arc:

$$h(\mathbf{p}) = \sum_{i=1}^{N} s_i p_i + \sum_{(i,j)\in A} h_{ij}(p_i - p_j)$$

with

$$h_{ij}(p_i - p_j) = \min_{0 \le x_{ij} \le u_{ij}} \left(c_{ij} - (p_i - p_j)\right) x_{ij}$$

$$= \begin{cases} \left(c_{ij} - (p_i - p_j)\right) u_{ij} & \text{if } p_i - p_j > c_{ij} \\ 0 & \text{if } p_i - p_j \le c_{ij} \end{cases}$$

Solution of the dual problem results in solution of the original minimum cost network flow problem. Notice that in the dual problem, the node mass balance constraints are temporarily *relaxed* since they are placed in the objective function via the Lagrangian function; hence, the term *relaxation algorithm*. The link capacity constraints remain explicitly accounted for. The objective is to find the optimal dual price vector **p** that will result in a solution that fully satisfies the node mass balance constraints. The advantage of this approach is that the inner minimization problem as defined by $h_{ij}(p_i - p_j)$ is extremely easy to solve, as seen above.

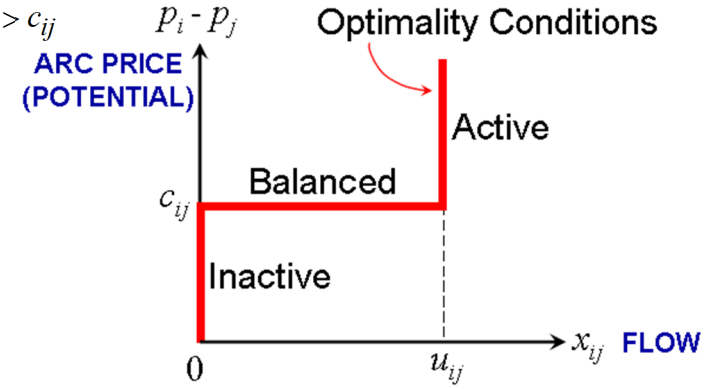
Solution of the above separable optimization problems results in the following general arc optimality conditions associated with flow in arc (*i*,*j*) for a given dual price vector **p**:

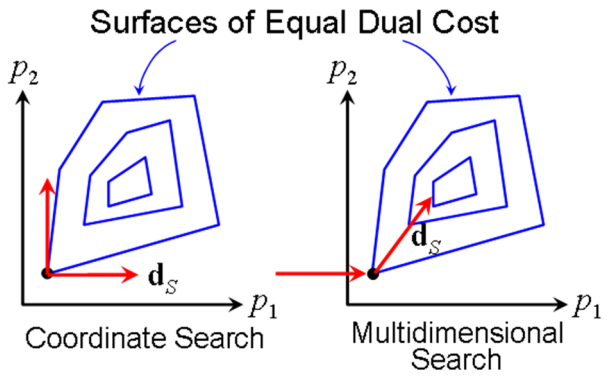
Inactive arc: $[x_{ij} = 0]$ if: $p_i - p_j < c_{ij}$

Balanced arc: $[0 \le x_{ij} \le u_{ij}]$ if: $p_i - p_j = c_{ij}$

Active arc: $[x_{ij} = u_{ij}]$ if: $p_i - p_j > c_{ij}$

Graphical representation of these arc optimality conditions is shown as the heavy line in this diagram. Optimal solution of the dual problem is found using a coordinate-wise dual ascent algorithm (although, at times, a multidimensional search is needed).





Changes in the dual prices **p** are made along the directional derivative, where:

$$\frac{\partial h(\mathbf{p})}{\partial p_i} = \sum_{\{j \mid (j,i) \in A\}} x^*_{ji} - \sum_{\{j \mid (i,j) \in A\}} x^*_{ij} + s_i \ , \quad i = 1,...,N$$

and the directional derivative is:

$$y'(\mathbf{p};\mathbf{d}_S) = \nabla h(\mathbf{p}) \cdot \mathbf{d}_S = \frac{\partial h(\mathbf{p})}{\partial p_1} \cdot \frac{\partial p_1}{\partial d_1} + \mathrm{L} \ + \frac{\partial h(\mathbf{p})}{\partial p_N} \cdot \frac{\partial p_N}{\partial d_N}$$

Evaluation of the directional derivative implies that the current flows are optimal since the arc optimality conditions specify that optimal flows are positive only for arcs that are active or balanced. Once a set of nodes $i \in S$ are identified for possible dual price changes, the directional derivative is calculated as:

$$y'(\mathbf{p};\mathbf{d}_S) = \sum_{\substack{(j,i)\text{: active,} \\ j \notin S, i \in S}} u_{ji} - \sum_{\substack{(i,j)\text{: active or balanced,} \\ i \in S, j \notin S}} u_{ij} + \sum_{i \in S} s_i$$

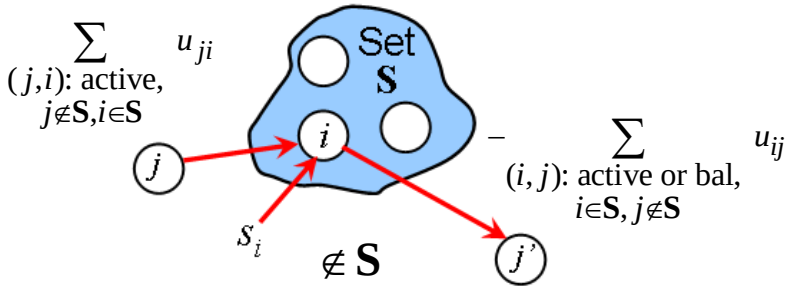
where $\mathbf{d}_S = (d_1,...,d_N)$, with

$$d_i = \begin{cases} 1 \text{ if } i \in S \\ 0 \text{ if } i \notin S \end{cases}$$

Notice in this formula that the directional derivative is not exactly evaluated, but rather is an approximation. However, it is a conservative approximation in that the directional derivative will be positive if a true ascent direction has been found. The first term only includes active arcs, where flows must be at the upper limits u_{ji} in order for the arc optimality conditions to be satisfied. The second term includes both active and balanced arcs, with the assumption made that flows in the balanced arcs are also at the upper bounds u_{ji}. This latter assumption may not be correct, but notice that since these bounds u_{ji} are subtracted in the directional derivative calculation, then overestimation of these

flows would not indicate an ascent direction that was really a descent direction. That is, if $y'(\mathbf{p};\mathbf{d}_S) > 0$, then direction $\mathbf{d}_S$ must be an ascent direction.

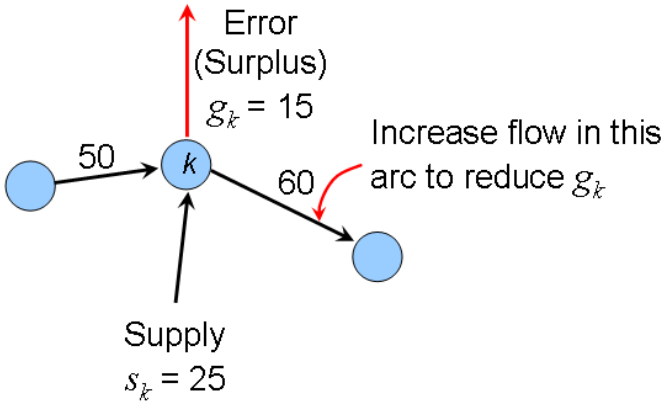
The difference between inflow and outflow across node set **S** when flows of active arcs are set at their upper bounds, and the flow of each balanced arc is set to upper bound depending if arc is outgoing from **S.**



Define the surplus g_i of node i as the difference between the total inflow into node i, less the total outflow from node i:

$$g_i = \sum_{\{j \mid (j,i)\in A\}} x_{ji} - \sum_{\{j \mid (i,j)\in A\}} x_{ij} + s_i$$

In essence, the node surplus represents *mass balance error* in the calculations. The dual ascent procedure attempts to reduce the mass balance error to zero.



At the start of an iteration, an integer flow-node price pair (**x**,**p**) is assumed to be available which satisfy the arc optimality conditions, but not flow mass balance. The current iteration will indicate if: (i) the primal problem is infeasible (i.e., cannot find a node surplus $g_i > 0$ for some i); (ii) (**x**, **p**) is optimal (i.e., $g_i = 0$ for all i, implying that **x** is feasible and, since the arc optimality conditions are satisfied, is also optimal); or (iii) a new pair can be found that improves the dual objective function (i.e., $g_i > 0$ for at least one node i). For the latter case, the iteration begins by selecting node k such that $g_k > 0$. The iteration maintains the two sets: S and L ; where $S \subset L$. At the initial iteration, set

$S = \{\emptyset\}$ and $L = \{k\}$. A label is also maintained for all nodes *i*, *L* which is an incoming arc to *i*.

The goal is to maximize the dual objective function, which will result in solution of the original minimum cost network flow problem. A **dual ascent** direction is defined for node prices for those nodes contained in set **S**. Since set **S** "*usually*" contains a single node, the search procedure generally proceeds in *one* coordinate direction *at a time* of the node price vector **p**. A dual ascent direction is defined for node prices for those nodes contained in set *S*. Since set *S* usually contains a single node, the search procedure generally proceeds in one coordinate direction at a time of the node price vector **p**. Dual prices are changed in the dual ascent direction so as to increase the dual objective function. Since the goal is to eventually achieve a solution where all $g_i = 0$, a flow augmentation step occurs in the algorithm where a path through the network is defined from a node *k* where $g_k > 0$ to a node *j*, where $g_j < 0$. This means that flow can be increased along that path, resulting in improved node surplus conditions for both nodes.

First, convert the network to an upper bound-only flow network (i.e., remove lower bounds $\bar{l}_{ij}$). Initially start with $x_{ij} = 0 \ \forall (i,j) \in A$ and $p_i = 0$, $i = 1,...,N$, although the initial flows may need to be adjusted to make sure that the arc optimality conditions (i.e., complementary slackness) conditions are initially satisfied for each arc.

A.3 Typical Relaxation Iteration

0. Initialization

Initially start with flows $x_{ij} = 0 \ \forall (i,j) \in A$ and $p_i = 0$, $i = 1,...,N$. Select a node k with node surplus $g_k > 0$ [if no such node can be found, then the solution is optimal or infeasible]

$$g_k = \sum_{\{j \mid (j,k) \in A\}} x_{jk} - \sum_{\{j \mid (k,j) \in A\}} x_{kj} + s_k$$

- Let the set of labels $L = \{k\}$
- Let the direction vector set $S = \{\emptyset\}$

1. Choose a Node to Scan

If: $S = L$ (i.e., an ascent direction is assured);
GOTO Step 4 and perform price change

Else: Select node *i* contained in the current set of labels, but not in the current direction vector set; i.e., select $i \in L - S$
$S := S \cup \{i\}$; GOTO Step 2

2. Label Neighboring Nodes of *i*

Check the directional derivative of the dual objective:

$$y'(\mathbf{p};\mathbf{d}_S) = \sum_{\substack{(j,i)\text{: active,}\\ j\notin S, i\in S}} u_{ji} - \sum_{\substack{(i,j)\text{: active or bal,}\\ i\in S, j\notin S}} u_{ij} + \sum_{i\in S} s_i$$

where direction vector $\mathbf{d}_S = (d_1,...,d_N)$, with

$$d_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

- **If:** $y' > 0$, then current direction $\mathbf{d}_S$ is an ascent direction; GOTO Price Change [Step 4]
- **Else:** add to labeled set of neighboring nodes that can eventually result in identification of a flow augmentation path from node k to node j: $L := L + \{j\}$

  for all nodes j such that:
  - arc (j, i) is balanced and $x_{ji} > 0$ [assign label (j, i)], or
  - arc (i, j) is balanced and $x_{ij} < u_{ij}$ [assign label (i, j)]

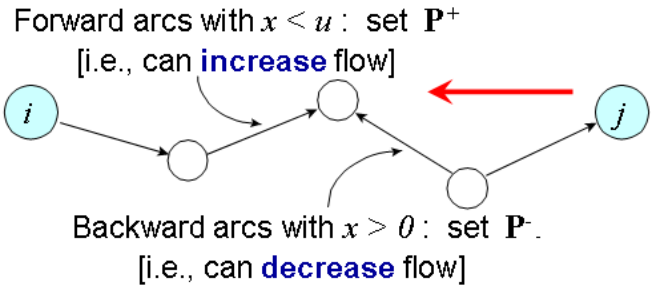
  If: for every node j added to L, $g_j > 0$, then a flow augmentation path is not yet found: RETURN to Step 1

  Else: select one of the nodes j with: $g_j < 0$; GOTO Step 3

3. Flow Augmentation

A flow augmentation path P has been determined to exist starting at node i and ending at node j as found in Step 2. Since $g_i > 0$ and $g_j < 0$, flow can be increased along the path such that g_i decreases towards zero and g_j increases towards zero, subject to limitations.

Path P is constructed by tracing labels backward starting from j, where P^+ is the set of forward arcs and P^- the set of backward arcs:



Calculate:

- For all links in P^+, ADD δ to the current flows.
- For all links in P^-, SUBTRACT δ from the current flows

$$\delta = \min \begin{cases} g_k \\ -g_j \\ (u_{mn} - x_{mn}) \ \forall (m,n) \in P^+ \\ x_{mn} \ \forall (m,n) \in P^- \end{cases}$$

- GOTO next iteration

3. Price Change

Set

$$x_{ij} = u_{ij} \quad \forall \text{ balanced links } (i, j) \text{ with } i \in S,\ j \notin S$$

$$x_{ji} = 0 \quad \forall \text{ balanced links } (j, i) \text{ with } i \in S,\ j \notin S$$

Let

$$\gamma = \min \begin{cases} \left[c_{ij} - (p_i - p_j) \right] \Big| x_{ij} < u_{ij}, i \in S,\ j \notin S & \textbf{INACTIVE} \\ \left[-c_{ji} + (p_i - p_j) \right] \Big| x_{ji} > 0, i \in S,\ j \notin S & \textbf{ACTIVE} \end{cases}$$

Set

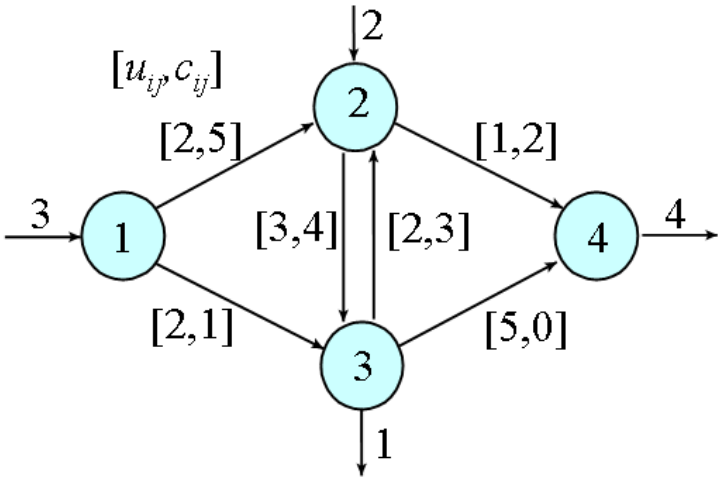
$$p_i = \begin{cases} p_i + \gamma & \text{if } i \in S \\ p_i & \text{otherwise} \end{cases}$$

GOTO Next Iteration

A.4 Example Problem

Consider the example network below, where exogenous flows are shown as supply and demand entering and leaving (respectively) each node. The link parameters are shown on each link, with all lower bounds set to zero. The objective is to find the minimum cost flow through the network that satisfies mass balance and all link flow upper bounds.

We begin with an initial solution for the integer flow vector, dual price vector pair as (**x**, **p**) = (**0**, **0**). Notice that this solution satisfies the arc optimality conditions, but violates feasibility since node surpluses $g_i \neq 0$.



[1] Bertsekas, D., *Linear and Network Optimization*, The MIT Press, Cambridge, Massachusetts, 1991.

ITERATION #1

Arc	x_{ij}	u_{ij}	s_i	s_j	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	0	2	3	2	5	0	0	3	2	INACT
(1,3)	0	2	3	-1	1	0	0	3	-1	INACT
(2,3)	0	3	2	-1	4	0	0	2	-1	INACT
(2,4)	0	1	2	-4	2	0	0	2	-4	INACT
(3,2)	0	2	-1	2	3	0	0	-1	2	INACT
(3,4)	0	5	-1	-4	0	0	0	-1	-4	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in\mathbf{A}} h_{ij}(p_i - p_j) = 0 + 0 =$ **0**

Primal Objective = $\sum_{(i,j)\in\mathbf{A}} c_{ij} x_{ij} =$ **0**

Step

0. $L = \{1\};\ S = \{\emptyset\}$

1. Select $i \in L - S;\ S := S \cup \{i\};$ so $i = 1$ and $S = \{1\}$

2. $y' = \sum_{\text{act}} u_{ji} - \sum_{\text{act or bal}} u_{ij} + \sum_{i\in S} s_i$

   $= 0 - 0 + 3 > 0$ [indicates that p_1 can be increased]

4. No x_{ij} adjustment is made at this iteration, since this is only done for *balanced* arcs; calculate $p_j - p_i + c_{ij}$

   $\gamma = \min \begin{cases} \cdot\ 0 - 0 + 5 \\ \cdot\ 0 - 0 + 1 \end{cases} = 1$ [for arc(1,3)] [this assures price does not go "too far"]

ITERATION #2

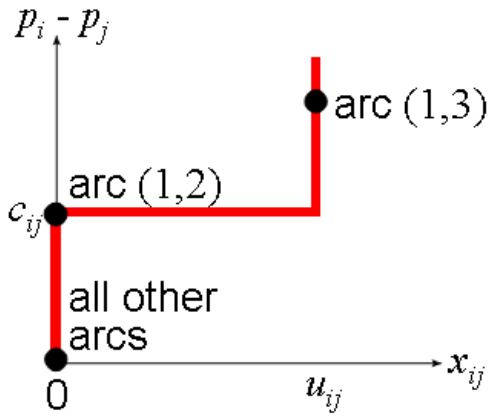
Arc	x_{ij}	u_{ij}	s_i	s_j	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	0	2	3	2	5	1	0	3	2	INACT
(1,3)	0	2	3	-1	1	1	0	3	-1	BAL
(2,3)	0	3	2	-1	4	0	0	2	-1	INACT
(2,4)	0	1	2	-4	2	0	0	2	-4	INACT
(3,2)	0	2	-1	2	3	0	0	-1	2	INACT
(3,4)	0	5	-1	-4	0	0	0	-1	-4	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in\mathbf{A}} h_{ij}(p_i - p_j) = 3 + 0 = \mathbf{3}$

Primal Objective = $\sum_{(i,j)\in\mathbf{A}} c_{ij} x_{ij} = \mathbf{0}$

Step

0. $L = \{1\}$; $S = \{\emptyset\}$
1. Select $i \in L - S$; $S = \{1\}$
2. $y' = -2 + 3 = 1 > 0$
4. Arc (1,3) is balanced--set $x_{13} = 2$
   (= 0 - 1 + 5 = 4 (for arc (1,2));
   therefore, $p_1 = 1 + 4 = 5$



ITERATION #3

Arc	x_{ij}	u_{ij}	s_i	s_j	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	0	2	3	2	5	5	0	1	2	BAL
(1,3)	2	2	3	-1	1	5	0	1	1	ACT
(2,3)	0	3	2	-1	4	0	0	2	1	INACT
(2,4)	0	1	2	-4	2	0	0	2	-4	INACT
(3,2)	0	2	-1	2	3	0	0	1	2	INACT
(3,4)	0	5	-1	-4	0	0	0	1	-4	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in \mathbf{A}} h_{ij}(p_i - p_j) = 3\cdot 5 - 4\cdot 2 = \mathbf{7}$

Primal Objective = $\sum_{(i,j)\in \mathbf{A}} c_{ij} x_{ij} = 1\cdot 2 = \mathbf{2}$

Step

0. $L = \{1\}$; $g_1 > 0$; so $S = \{\emptyset\}$

   [Note: node 1 is still selected, even though g_2 is a greater--arbitrary]

1. $S = \{1\}$

2. $y' = -\sum_{\substack{\text{act or bal,}\\ i\in S,\, j\notin S}} u_{ij} + \sum_{i\in S} s_i = -4 + 3 < 0$ [no improvement by increasing p_1]

   $L := L + \{j\}$

   $L = \{1,2\}$: outflow link *and* balanced *and* $x_{ij} < u_{ij}$

   Check if $g_2 > 0$ [yes!] [have not yet found flow augmentation path]

   RETURN TO Step 1:

1. $S = \{1\}$; $L = \{1,2\}$

   Select $i \in L - S$; $i = 2$; so $S = \{1,2\}$

2. $y' = \sum_{i\in S} s_i = -2 + 5 > 0$ [i.e., p_2 can be increased]

4. γ = min {[$p_j + c_{ij} - p_i$] for arcs (2,3), (2,4)}

   = min {4, 2} = 2

   Therefore, $p_1 = 5 + \mathbf{2} = 7$; $p_2 = 0 + \mathbf{2} = 2$ [for all nodes $i \in S = \{1,2\}$]

ITERATION #4

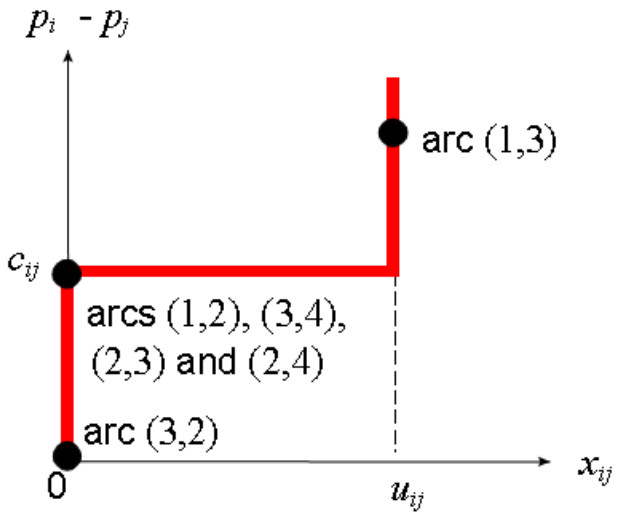
Arc	x_{ij}	u_{ij}	s_i	s_j	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	0	2	3	2	5	7	2	1	2	BAL
(1,3)	2	2	3	-1	1	7	0	1	1	ACT
(2,3)	0	3	2	-1	4	2	0	2	1	INACT
(2,4)	0	1	2	-4	2	2	0	2	-4	BAL
(3,2)	0	2	-1	2	3	0	2	1	2	INACT
(3,4)	0	5	-1	-4	0	0	0	1	-4	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in\mathbf{A}} h_{ij}(p_i - p_j) = 3\cdot 7 + 2\cdot 2 - 6\cdot 2 = \mathbf{13}$

Primal Objective = $\sum_{(i,j)\in\mathbf{A}} c_{ij}x_{ij} = 1\cdot 2 = \mathbf{2}$

Step

0. L ={1}; keep selecting node 1 since $g_1 > 0$
1. S ={1}
2. y' = (-2 -2) + 3 = -1 < 0
   L: = L + {j} ; L = {1,2}
   Check if $g_2 > 0$ [Yes!]
   RETURN to Step 1
1. S = {1,2}; L = {1,2}
2. y' = -2 - 1 + 5 = 2 > 0
4. Does $x_{ij} = u_{ij}$ for all balanced arcs OUT? Yes!--arc (2,4)
   Therefore, set x_{24} = 1
   $\gamma = \min\left\{\left[p_j + c_{ij} - p_i\right] \text{ for arc } (2,3)\right\} = 2$
   Therefore p_1 = 7 + **2** = 9; p_2 = 2 + **2** = 4



ITERATION #5

Arc	x_{ij}	u_{ij}	s_i	s_j	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	0	2	3	2	5	9	4	1	1	BAL
(1,3)	2	2	3	-1	1	9	0	1	1	ACT
(2,3)	0	3	2	-1	4	4	0	1	1	BAL
(2,4)	1	1	2	-4	2	4	0	1	-3	ACT
(3,2)	0	2	-1	2	3	0	4	1	1	INACT
(3,4)	0	5	-1	-4	0	0	0	1	-3	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in \mathbf{A}} h_{ij}(p_j - p_i) = 3\cdot 9 + 2\cdot 4 - 8\cdot 2 - 2\cdot 1 = \mathbf{17}$

Primal Objective = $\sum_{(i,j)\in \mathbf{A}} c_{ij} x_{ij} = 1\cdot 2 + 2\cdot 1 = \mathbf{4}$

Step

0. L={1}; $g_1 > 0$
1. S={1}
2. $y' = -\sum_{\text{act or bal}} u_{ij} + \sum_{i\in S} s_i = (-2-2)+3 < 0$

   L: = L + {j} with label (1,2); L = {1,2}

   Check if $g_2 > 0$; Yes! RETURN to Step 1
1. S = {1,2}; L = {1,2}
2. $y' = -\sum_{\text{act/bal OUT}} u_{ij} + \sum_{i\in S} s_i = (-2-3-1)+5 < 0$

   $L = L + \{j\}$ with label (2,3) ; L = {1, 2, 3}

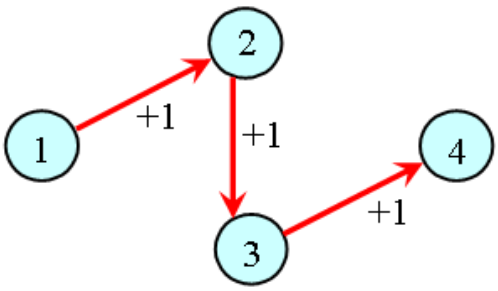
   Check if $g_3 > 0$; Yes! RETURN to Step 1
1. S = {1,2}; L = {1, 2, 3}; Select $i \in L - S$ = 3
2. $y' = -\sum_{\text{act/bal OUT}} u_{ij} + \sum_{i\in S} s_i = (-1-5)+(3+2-1) < 0$

   L: = L + {j} with label (3,4) ; L = {1,2,3,4}

   Check if g_4 < 0 ; Yes! = -3; GOTO Step 3: Flow Augmentation
3. Path of flow augmentation P is 1-2-3-4 [all forward arcs]; so $P^+ = P$

$$\gamma = \min \begin{cases} g1\ [g_1] \\ g\ -(-4)\ [-g_4] \\ g2\ [u_{12} - x_{12}] \\ g3\ [u_{23} - x_{23}] \\ g5\ [u_{34} - x_{34}] \end{cases} = \mathbf{1}$$

Flow Increase Along Path



ITERATION #6

Arc	x_{ij}	u_{ij}	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	1	2	5	9	4	0	1	BAL
(1,3)	2	2	1	9	0	0	1	ACT
(2,3)	1	3	4	4	0	1	1	BAL
(2,4)	1	1	2	4	0	1	-2	ACT
(3,2)	0	2	3	0	4	1	1	INACT
(3,4)	1	5	0	0	0	1	-2	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in \mathbf{A}} h_{ij}(p_j - p_i) = 3\cdot 9 + 2\cdot 4 - 8\cdot 2 - 2\cdot 1 =$ **17**

Primal Objective = $\sum_{(i,j)\in \mathbf{A}} c_{ij} x_{ij} = 5\cdot 1 + 1\cdot 2 + 4\cdot 1 + 2\cdot 1 =$ **13**

Step

0. $L = \{2\}$; $S = \{\emptyset\}$; node $k = 2$
1. $S = \{2\}$; $i = 2$
2. $y' = -\sum_{\text{act/bal OUT}} u_{ij} + \sum_{i\in S} s_i = (-3-1) + 2 < 0$

   $L = L + \{j\}$; add node 1 [label (1,2)] and node 3 [label (2,3)]

   $L = \{1,2,3\}$; check $g_1 = 0$ and $g_3 = 1$ [both > 0]

   RETURN to Step 1
1. set $L - S = \{1,3\}$

   Select node $i = 3$

   Therefore: $S = \{2,3\}$
2. $y' = \sum_{\text{act IN}} u_{ji} - \sum_{\text{act/bal OUT}} u_{ij} + \sum_{i\in \mathbf{S}} s_i = 2 - 5 + (2-1) = -2 <$ **0**

   $L = L + \{j\}$; add node 4

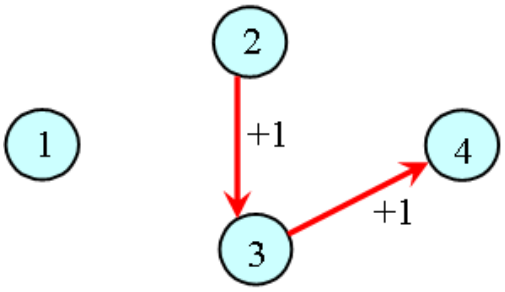
   Check $g_4 = -2 < 0$; GOTO Step 3: Flow Augmentation
3. $\delta = \min \begin{cases} \cdot 1\,[g_2] \\ \cdot 2\,[-g_4] \\ \cdot 4\,[u_{34} - x_{34}] \\ \cdot 2\,[u_{23} - x_{23}] \end{cases} = 1$

   Path P: 2-3-4

   All forward arcs--

   Therefore, $P^+ = P$

Flow Increase Along Path



ITERATION #7

Arc	x_{ij}	u_{ij}	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	1	2	5	9	4	0	0	BAL
(1,3)	2	2	1	9	0	0	1	ACT
(2,3)	2	3	4	4	0	0	1	BAL
(2,4)	1	1	2	4	0	0	-1	ACT
(3,2)	0	2	3	0	4	1	0	INACT
(3,4)	2	5	0	0	0	1	-1	BAL

Dual Objective = $h(\mathbf{p}) = \sum_i s_i p_i + \sum_{(i,j)\in \mathbf{A}} h_{ij}(p_j - p_i) = 3\cdot 9 + 2\cdot 4 - 8\cdot 2 - 2\cdot 1 = \mathbf{17}$

Primal Objective = $\sum_{(i,j)\in \mathbf{A}} c_{ij} x_{ij} = 5\cdot 1 + 1\cdot 2 + 4\cdot 2 + 2\cdot 1 = \mathbf{17}$

Step

0. $L = \{3\}$; $S = \{\emptyset\}$; node $k = 3$
1. $S = \{3\}$; node $i = 3$
2. $y' = \sum_{\text{act IN}} u_{ji} - \sum_{\text{act/bal OUT}} u_{ij} + \sum_{i\in S} s_i = +2 - 5 - 1 = -4 < 0$
   $L: = L + \{j\}$; add node 2 [label (2,3)] and node 4 [label (3,4)] ; $L = (2,3,4\}$
   Check $g_2 = 0$; $g_4 = -1$ [both < 0]
4. Path $P^+ = 3 - 4$;
   node $k = 3$; node $j = 4$
   $\delta = \min\{1\ [g_3],\ 1\ [-g_4],\ 3\ [u_{34} - x_{34}]\} = 1$

FINAL SOLUTION

Arc	x_{ij}	u_{ij}	c_{ij}	p_i	p_j	g_i	g_j	State
(1,2)	**1**	2	5	**9**	4	0	0	BAL
(1,3)	**2**	2	1	9	0	0	0	ACT
(2,3)	**2**	3	4	**4**	0	0	0	BAL
(2,4)	**1**	1	2	4	0	0	0	ACT
(3,2)	**0**	2	3	**0**	4	0	0	INACT
(3,4)	**3**	5	0	0	**0**	0	0	BAL

Notice that $g_i = 0$ for all nodes. Therefore, *dual objective* = *primal objective* and all complementary slackness conditions are satisfied.

APPENDIX B
Stream-Aquifer Modeling in MODSIM

B.1 Analytical Equations

The mathematical flow equation for general two dimensional flow in an unconfined groundwater aquifer can be derived from Darcy's Law and the principle of mass continuity. The resultant equation is a nonlinear, second-order partial differential equation known as the Boussinesq equation (Willis and Yeh, 1987):

$$\frac{\partial}{\partial x}\left(K_x b \frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_y b \frac{\partial h}{\partial y}\right)+Q=S\frac{\partial h}{\partial t} \tag{B.1}$$

where K_x, K_y are hydraulic conductivities along the x,y axes, respectively (Lt^{-1}); h is potentiometric head (L); Q is net groundwater withdrawal per unit area (Lt^{-1}); S is storage coefficient; and t is time (t).

Where variation in saturated thickness is small and the specific yield/storage coefficient is assumed constant, the governing groundwater equation can be written as a linear form of the Boussinesq equation:

$$T\left(\frac{\partial^2 h}{\partial x^2}+\frac{\partial^2 h}{\partial y^2}\right)+Q=S\frac{\partial h}{\partial t} \tag{B.2}$$

where T is transmissivity = Kb (L^2t^{-1}), K is hydraulic conductivity (Lt^{-1}), and b is saturated thickness (L).

Maddock (1974) showed that if the ratio of drawdown to saturated thickness is less than 20 percent, then for a nonlinear free-surface model (i.e., the Boussinesq equation), the linear contribution is between 75 to 100 percent of drawdown due to pumping. Accuracy of the linear model increases as the drawdown to saturated thickness ratio decreases. If the ratios are large, the Dupuit assumptions and the nonlinear flow equations are invalid.

Since the governing groundwater equation is linear and time invariant, linear system theory can be applied via the *principle of superposition* (Bear, 1979). This principle states that the presence of one boundary condition does not affect the response produced by the presence of other boundary conditions and that there are no interactions among the responses produced by the various boundary conditions. It is then possible to analyze the effect of individual events and then linearly combine the results.

Glover and Balmer (1954) and Glover (1968) presented an analytical procedure for determining depletion of flow in a nearby stream caused by pumping a well. Depletion flows were calculated using the distance of the well from the river, the properties of the

aquifer (i.e., storage coefficient and transmissivity), time of pumping and time from start of pumping. The following assumptions apply:

1. Aquifer is unconfined, homogeneous, isotropic, and of infinite extent
2. River is straight, fully penetrates the aquifer and is a constant head source.
3. Water table is initially horizontal and water is released instantaneously from storage.
4. Well fully penetrates the aquifer.
5. Pumping is steady and drawdown is small compared to aquifer thickness.
6. Residual effects of previous pumping are negligible.

According to Glover (1968), the ratio of stream depletion rate to the rate of well discharge is:

$$\frac{Q_s}{Q_w} = 1 - erf\left(\frac{a}{\sqrt{4tT/S}}\right) \tag{B.3}$$

where Q_s is rate of stream depletion (L^3t^{-1}); Q_w is rate of well discharge (L^3t^{-1}); a is perpendicular distance from well (L); t is pumping time (t); T is transmissivity (L^2t^{-1}); S is specific yield; and $erf(z)$ is the *error function.*

Glover (1977) extended the analytical approach to include bank storage, line source, return flows from irrigation, and intermittent well operation. Willis and Yeh (1987) presented a list of fifteen analytical response equations. Warner et al (1989) reviewed various analytical solutions to the artificial recharge problem, including Glover (1960), Hantush (1967), Rao and Sarma (1981), and Hunt (1971). The Hantush and Glover solutions were shown to be identical and were highly recommended for rectangular basins. It was also suggested that solutions for circular basins may be replaced by solutions for square basins with equivalent area. Madsen (1988) concluded that analytical models are not ideal for verifying the influence of existing wells on stream depletion, but are suitable as a tool for estimating impacts of new wells on streamflow depletion. Madsen (1988) also showed that analytical methods often overestimate stream depletion by failing to account for resistance near the stream.

The major disadvantage of the analytical method is that nonpoint sources of flow are often approximated as point sources (Warner et al., 1986). Other limitations of analytical methods such as Glover's method include (Morel-Seytoux and Zhang, 1990):

- Method of averaging transmissivities over a heterogeneous aquifer is arbitrary
- Procedure for calculating depletion from a certain reach (not the entire river) is inconvenient, involving numerical integration, or inaccurate because of steady state assumptions;
- In most cases, the river is not straight

Qazi and Danielson (1974) used a computer program based on the Glover equations to evaluate augmentation plans for wells, recharge lines, and pit operations in an alluvial aquifer. Contributory effects of only those pumped wells or recharge sources requiring evaluation are determined, and are independent of other interactions already in process such as: effects of precipitation, surface water application, evapotranspiration, or other wells, reservoirs, and ditches. Labadie, et al. (1983) used analytical solutions embedded in a conjunctive use model to consider groundwater pumping (Glover, 1977), reservoir seepage (Glover, 1977), canal seepage (McWhorter and Sunada, 1977), irrigation recharge (Maasland, 1959) and bank storage (Glover, 1977). Hantush and Marino (1989) developed a chance constrained stream-aquifer management model based on the Hantush (1959) analytical solution. Male and Mueller (1992) used the equations of Jenkins (1968) to develop a groundwater management model for prescribing groundwater use permits in Massachusetts.

B.2 Discrete Kernel/Response Functions

Most groundwater management scenarios require information only on select events in an aquifer. Extraneous information on drawdown and flow rates at noncritical locations is not only unnecessary but computationally prohibitive. Applying linear system theory to the groundwater equation allows the use of Green's function to solve the resulting non-homogeneous boundary value problem (Maddock, 1972). Response of the groundwater system due to external excitations such as pumping, recharge, or infiltration at any point in space and time can be expressed as a set of unit coefficients independent of the magnitude of the excitation. Integrated with a finite difference groundwater model, resultant flows can be superimposed to determine net effects at a single location due to a series of excitations or at a series of locations due to a single excitation.

It is convenient to express the Boussinesq equation in terms of water table drawdown:

$$\frac{\partial}{\partial x}\left(T\frac{\partial s}{\partial x}\right)+\frac{\partial}{\partial y}\left(T\frac{\partial s}{\partial y}\right)+Q_p=S\frac{\partial s}{\partial t} \tag{B.4}$$

where T aquifer transmissivity (L^2t^{-1}); s is water table drawdown (L); Q_p is groundwater withdrawal rate per unit area at well p (Lt^{-1}) ; S is storage coefficient; t is time (t); and x,y are horizontal coordinates (L).

This equation can be solved using Green's function (Maddock, 1972):

$$s_w(t)=\int_0^t \delta_{wp}(t-\tau)Q_p(\tau)d\tau \tag{B.5}$$

where $s_w(t)$ is drawdown at aquifer point w due to a single well pumping Q_p at point p (L); and δ_{wp} is the kernel function (Green's function) of aquifer drawdown at point w due to a unit impulse excitation at p. The discrete form of the convolution equation for a

heterogeneous aquifer with finite boundaries is (Maddock, 1972; Morel-Seytoux and Daly, 1975):

$$s_w(t) = \sum_{p=1}^{P} \sum_{\tau=1}^{t} \delta_{wp}(t-\tau+1) Q_p(\tau) \tag{B.6}$$

where t are now discrete time periods; $s_w(t)$ is drawdown from an initially horizontal (or initially steady) water table at any aquifer point w at the end of the period t (L); $Q_p(\tau)$ is the mean pumping rate per unit area from well p during the period τ (pumped volume for the period) (Lt^{-1}); P is the total number of excitation points or wells; and $\delta_{wp}(t)$ is the discrete response or kernel coefficient representing the drawdown at the end of period t if a unit volume of water is withdrawn during the initial period from well p, with well pumping terminated indefinitely thereafter.

Maddock (1972) first introduced the concept of a response function for a groundwater system, with drawdown in response to pumping stress modeled by a two-dimensional linear partial differential equation. This allowed an explicit coupling of a groundwater simulation model with a quadratic programming management model to optimize an economic objective of minimizing pumping costs subject to satisfying specified demands. Maddock (1974) used Green's function to extend this approach to the case of stream-aquifer interactions.

Again, based on linear system theory and the Green's function, Morel-Seytoux and Daly (1975) developed a finite difference model to generate any aquifer response as an explicit function of pumping rates, which they referred to as a *discrete kernel generator*. The discrete kernel method has been utilized extensively as a tool for solving complex groundwater management problems (Morel-Seytoux, et al., 1981; Illangasekare and Morel-Seytoux, 1982; Illangasekare and Brannon, 1987; and Illangasekare and Morel-Seytoux, 1986).

B.3 Parallel Drain Analogy for Stream-Aquifer Systems

The interaction of a water table aquifer receiving recharge from irrigation and precipitation, and an interconnected stream, can be modeled utilizing the method developed by Maasland (1959). This method was developed for a parallel drain system and can be applied to a stream-aquifer system as well. The idealized parallel drain system is shown in Fig. B.1.

The nonlinear partial differential equation for one-dimensional groundwater flow is

$$K \frac{\partial}{\partial x}\left(d+h\right)\frac{\partial h}{\partial x} = S\frac{\partial h}{\partial t} \tag{B.7}$$

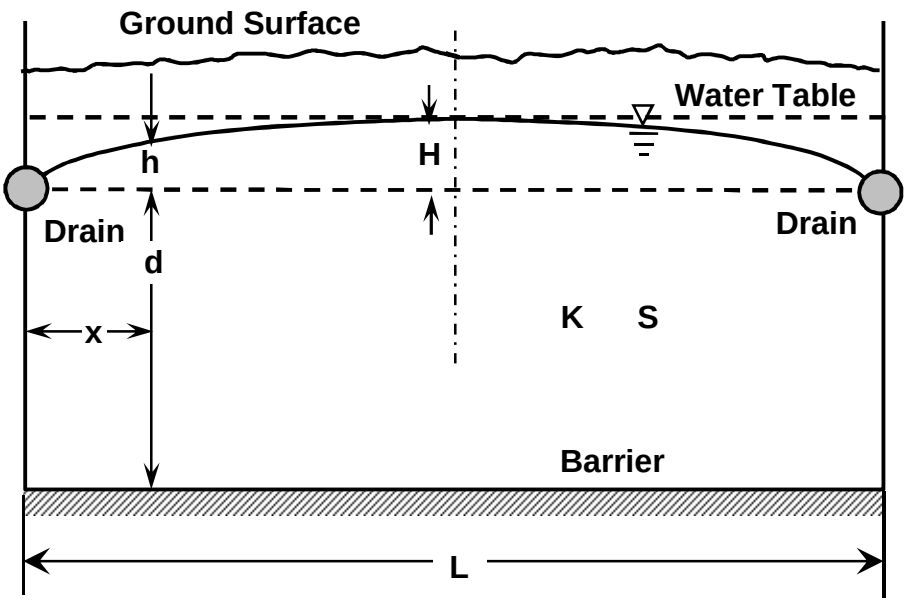


Fig. B.1 Parallel drain analogy for stream-aquifer systems.

where K is permeability of the aquifer (Lt^{-1}); d is original saturated thickness (L); S is specific yield; h is height of the water table measured from the assumed original stable water table level (L); d is depth measured from the assumed original stable water table level to the impermeable boundary (L); x is distance measured along the path of flow (L); and t is time.

By assuming h is small compared to d, the linearized form of Eq. B.7 is:

$$\alpha \frac{\partial^2 h}{\partial x^2} = \frac{\partial h}{\partial t} \tag{B.8}$$

where $\alpha = \frac{T}{S}$; T is transmissivity, which equals $K \cdot d$; and the boundary conditions are:

$h = 0$ when $x = 0$ for $t > 0$
$h = 0$ when $x = L$ for $t > 0$
$h = H$ when $t = 0$ for $0 < x < L$

Maasland (1959) obtained the solution as:

$$h = \frac{4H}{\pi} \sum_{n=1,3,5\ldots}^{\infty} \frac{1}{n} \exp\left(\frac{-n^2\pi^2\alpha t}{L^2}\right) \sin\left(\frac{n\pi x}{L}\right) \tag{B.9}$$

where H is initial uniform height of recharge water from an initial event and L is spacing of the parallel drains.

The volume per unit length of water remaining to be drained (L^2) is

$$V_d = S\int_0^L h\,dx \tag{B.10}$$

and the fraction remaining to be drained is

$$F = \frac{V_d}{V} \tag{B.11}$$

where initial drainable volume is

$$V = S \cdot H \cdot L \tag{B.12}$$

Therefore

$$F = \frac{S\int_0^L h\,dx}{S \cdot H \cdot L} \tag{B.13}$$

Assuming H = 1 represents a unit initial recharge event, substitution of h from Eq. B.9, and integration results in:

$$F = \frac{8}{\pi^2}\sum_{n=1,3,5...}^{\infty}\frac{1}{n^2}\exp\left(-n^2\pi^2\frac{\alpha t}{L^2}\right) \tag{B.14}$$

representing the fraction of the total initially drainable volume due to the unit recharge event that is in the aquifer at the end of time t and available for flow to the drains. For any time t from the beginning of the recharge event, F can be predetermined. The difference of successive F values over two adjacent discrete time periods represents the flow fraction to the drains during that time interval.

$$\begin{aligned} F_k - F_{k-1} = {} & \frac{8}{\pi^2}\sum_{n=1,3,5...}^{\infty}\frac{1}{n^2}\exp\left(-n^2\pi^2\frac{\alpha k\Delta t}{L^2}\right) \\ & - \frac{8}{\pi^2}\sum_{n=1,3,5...}^{\infty}\frac{1}{n^2}\exp\left(-n^2\pi^2\frac{\alpha (k-1)\Delta t}{L^2}\right) \end{aligned} \tag{B.15}$$

Define

$$\delta_k = F_k - F_{k-1} \tag{B.16}$$

where δ_k is a unit *response coefficient* or *discrete kernel* for a unit recharge rate at initial time interval k = 1. Note that the recharge rate is idealized as being uniformly distributed over the width of spacing between drains.

B.4 Return Flow Calculations

Consider the idealized stream-aquifer system as shown in Fig. B.2. The river is assumed to be located at the center of the valley. The solution described above can be applied directly with L equal to the valley width. The analogy is applicable since the middle section of the parallel drains is a no-flow boundary and is analogous to either the left boundary or the right boundary of the stream-aquifer system. If the parallel drain system is divided in half at the no flow boundary and rearranged to bring the drains into coincidence, the direct analogy with the stream-aquifer system is evident. The drains are replaced by the river and the flow to the drains represents return flow to the river.

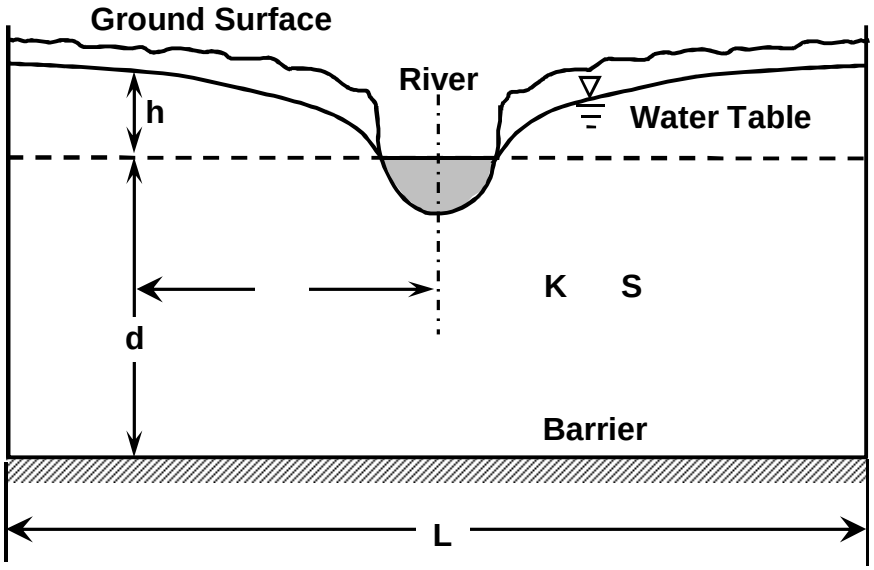


Fig. B.2. Idealized stream-aquifer system (Glover, 1977).

For cases where the river is not located at the center of the valley, the above solution (Eq. B.14) is still applicable with L equal to twice the width W of either side of the valley (i.e., $L^2 = 4W^2$. Fraction F can be determined for each side of the valley and return flows computed separately. Again, it is idealized that recharge events are uniformly distributed over each side of the valley.

Let N be the total number of time intervals of length Δt and I_k the recharge rate during the *k-th* time interval, where $k < N$, as shown in Fig. B.3. For any demand node i and current time period k, the total return flow IRF_{ik} from the current and previous time periods due to groundwater recharge is calculated using linear superposition:

$$IRF_{ik} = \sum_{\tau=1}^{k} I_{i\tau} \cdot \delta_{i,k-\tau+1} \; ; \; \delta_{i,k-\tau+1} = 0 \text{ for } k-\tau+1 > N \qquad \text{(B.17)}$$

where IRF_{ik} is the infiltration rate at node i, period k, and $\delta_{i,k-\tau+1}$ is the response or discrete kernel coefficient defined for node i, period k - τ+1.

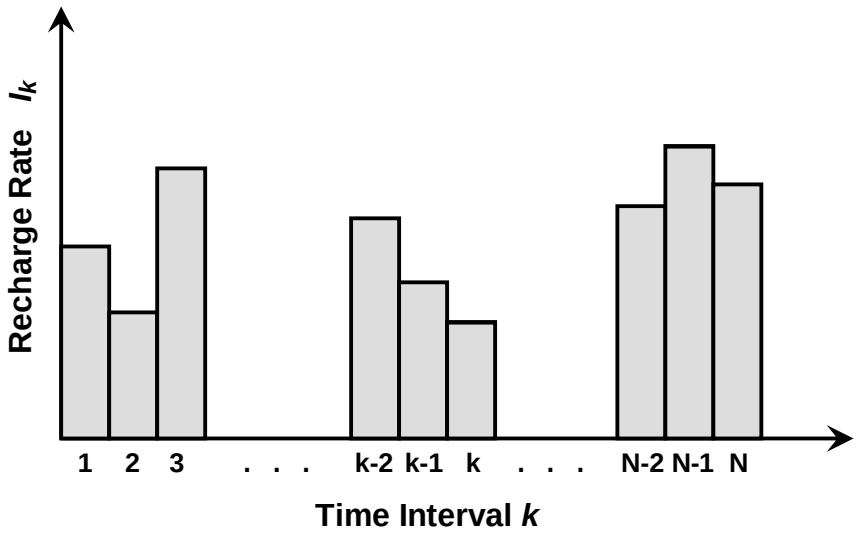


Fig. B.3 Series of recharge events

In MODSIM, upper bounds on return flow links (Fig. 4) are adjusted iteratively as follows: (1) all upper bounds are first set equal to the return flows computed from previous activities; (2) MODSIM is next run for the current period using these bounds, with return flows from all sources recomputed using available link flows obtained from this solution; (3) if current return flows agree with previous estimates, the process terminates; otherwise, return to step 2 and repeat until convergence is achieved.

B.5 Stream Depletion from Pumping

The same approach used for calculating return flows is also applied to calculation of stream depletion due to pumping PSD_{ik}, where

$$PSD_{ik} = \sum_{\tau=1}^{k} P_{i\tau} \cdot \alpha_{i,k-\tau+1} \;;\; \alpha_{i,k-\tau+1} = 0 \text{ for } k-\tau+1 > N \qquad \text{(B.18)}$$

In the case of groundwater withdrawal $P_{i\tau}$, the same principles described above are applicable to determining response coefficient kernels $\alpha_{i,k-\tau+1}$, but for river depletion rather than return flows. Again, it is idealized that pumping withdrawals are uniformly distributed over each side of the valley, rather than attempting to model individual wells. Since the computation is sequentially carried out period by period in MODSIM, the current period stream-aquifer interactions are contingent upon stresses during previous periods. Therefore, it is recommended to run MODSIM for an initial *N* periods for start-up or initialization purposes, such that after *N* periods, the model output can be trusted to properly account for past history. Specification of *N* is left to the user.

B.6 Canal Seepage

Seepage from a canal or a stream is assumed to correspond to a line source of recharge water. For a one-dimensional line source in an infinite aquifer, as shown in Figure B.4, the governing flow equation is (McWhorter and Sunada, 1977):

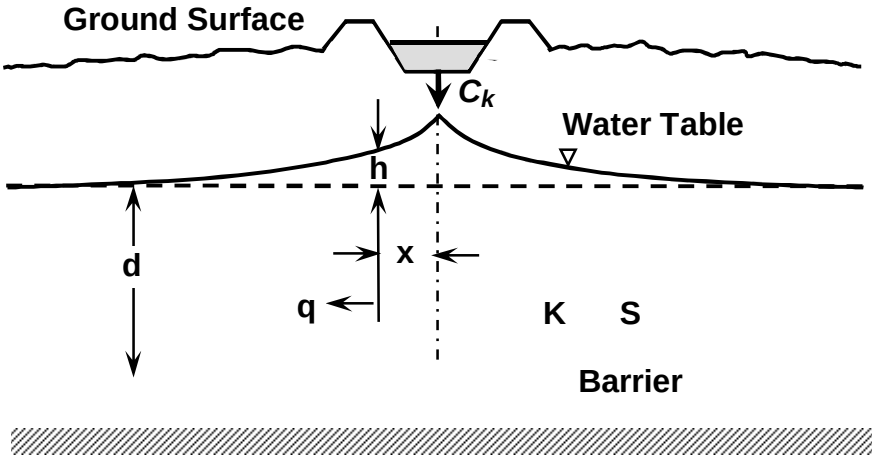


Fig. B.4. Illustration of line source for canal seepage.

$$\alpha \frac{\partial^2 q}{\partial x^2} = \frac{\partial q}{\partial t} \tag{B.19}$$

where x is the Cartesian coordinate in the horizontal plane (L) and q is the flow rate or Darcy velocity (L^2t^{-1}), calculated as:

$$q = -K \frac{\partial h}{\partial t} \tag{B.20}$$

The solution is (McWhorter and Sunada, 1977):

$$q = \frac{I}{2} \operatorname{erfc}\left(\frac{\mathrm{x}}{\sqrt{4\alpha \mathrm{t}}} \right) \tag{B.21}$$

where I is the one dimensional magnitude of the source (Lt^{-1}), with *erfc*(z) representing the complementary error function:

$$\operatorname{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du \tag{B.22}$$

assuming the following boundary and initial conditions:

$$\begin{aligned} &q = \frac{I}{2} \quad \text{at } x = 0 \\ &q = 0 \quad \text{as } x \to \infty \\ &q = 0 \quad \text{as } t = 0; \text{ for all } x \end{aligned} \tag{B.23}$$

Define $q_0 = \frac{I}{2}$ as the applied line source flow rate in the aquifer at the line source location. Note that the denominator of two is necessary since q flows in two horizontal directions. Integrating Eq. B.20 from zero to t results in the ratio of the volume of return flow to the stream and the line source volume of seepage applied up to time t:

$$\frac{v}{q_0 \cdot t} = \left(\frac{x^2}{2\alpha t} + 1 \right) \text{erfc}\left(\frac{x}{\sqrt{4\alpha t}} \right) - \left[\frac{x}{\sqrt{4\alpha t}} \frac{2}{\sqrt{\pi}} \exp\left(-\frac{x^2}{4\alpha t} \right) \right] \tag{B.24}$$

This solution is for a continuous application of a line source. After termination of the source, residual effects still contributes flow to the stream. The residual is accounted for by assuming an imaginary pumping source at the same location and initiating pumpage at the same rate as the recharge source from the time recharge terminates. The volume ratio at any time after recharge ceases is the difference between the volume ratio obtained if recharge had continued and the volume ratio obtained from pumping of the imaginary pumping source. For a discrete time interval, if the applied line source volume equals one, the volume ratio is in essence the unit response of line source or canal seepage.

Let φ represent the unit response of canal seepage. Then for canal link ℓ, the total return flow $CRF_{\ell k}$ from canal seepage $C_{\ell 1},,...,C_{\ell k}$ during each time interval k is:

$$CRF_{1k} = \sum_{\tau=1}^{k} C_{1\tau} \cdot \phi_{1,k-\tau+1} \; ; \; \phi_{1,k-\tau+1} = 0 \text{ for } k - \tau + 1 > N \tag{B.25}$$

B.7 Point Source Water Application

Reservoir seepage RS_{ik} is defined as a point source application for storage node i , time period k. The effect on the stream corresponds to the effect of a recharge well, which in turn has the same absolute flow magnitude as a pumping well, with the flow direction reversed. This solution turns out to be exactly the same as that for the line source solution (Glover, 1977). Therefore, $C_{\ell\tau}$ is replaced with $RS_{i\tau}$ in Eq. B.24, with the resulting return flow defined as RRF_{ik}. Again, there is little error in assuming reservoir seepage as a point source, as long as the reservoir surface area is small in comparison with the area of the subsystem containing it.

For reservoir i during time period k, the total return flow RRF_{ik} from reservoir seepage, based on current and previous period seepage, is

$$RRF_{ik} = \sum_{\tau=1}^{k} RS_{i\tau} \cdot \phi_{i,k-\tau+1} \; ; \; \phi_{i,k-\tau+1} = 0 \text{ for } k - \tau + 1 > N \tag{B.26}$$

B.8 Stream Depletion Factor Method (*sdf*)

Jenkins (1968) solved the Glover equation graphically by developing dimensionless curves and tables to compute the rate and volume of stream depletion by wells. The *stream depletion factor* (*sdf*) was arbitrarily chosen as time in days where the volume of stream depletion is 28% of the volume pumped during time *t*, and can be expressed as:

$$sdf = \frac{a^2 S}{T} \tag{B.27}$$

where a is perpendicular distance from the pumped well to the stream (L); S is specific yield of the aquifer (dimensionless); and T is transmissivity ($L^2 t^{-1}$).

In a complex system, the value of *sdf* at any location depends on the integrated effects of irregular impermeable boundaries, stream meanders, aquifer properties, areal variation, distance from the stream, and hydraulic connection between stream and aquifer. The basic assumptions are similar to those associated with the Glover equation:

Moulder and Jenkins (1969) introduced the *sdf* concept to a digital model and the USGS used it to generate groundwater response coefficients for developing regional models (Taylor and Luckey, 1972; Hurr, 1974; Hurr and Burns, 1980; and Warner et al., 1986) and groundwater *sdf* contour maps (Hurr, et al., 1972).

B.9 Response Functions from Finite Difference Groundwater Models

The Boussinesq partial differential equation for groundwater flow in a heterogeneous and anisotropic medium (Eq. B.1) can be solved using finite difference or finite element numerical approximation methods (Willis and Yeh, 1987). Finite difference methods require that a regular Cartesian grid be defined over the domain of the region to be modeled. The groundwater basin is spatially discretized into a grid structure represented by a finite number of rectangular cells. Differential terms in the partial differential equation are replaced with numerical approximations calculated from differences in potentiometric head at the cell locations. All aquifer parameters, heads and hydrologic data are assumed to be constant and homogeneously distributed within a grid cell. This numerical approximation results in replacement of the original partial differential equation with a system of simultaneous linear difference equations. The most popular finite difference groundwater modeling system is MODFLOW, developed by the U.S. Geological Survey (Harbaugh, et al., 2000). The finite element method is another approach to numerical modeling of groundwater basins, but is less popular and generally more computationally time consuming than the finite difference method, although it offers the advantage of more accurate representation of irregular aquifer boundaries.

Application of numerical models such as MODFLOW allow relaxation of most of the idealized assumptions associated with analytical modeling approaches such as the Glover

method. Hartwell (1987) compared results from a model based on the Glover solution, the *sdf* method, and a finite difference model for a recharge site along the along the South Platte River, Colorado. Use of the finite difference model was recommended in this study since it provided more accurate return flow calculations than the other methods. Sophocleous et al. (1995) also compared the Glover analytical solution with MODFLOW, concluding that the latter was preferred for more accurately treating irregular boundary conditions, streambed aquifer hydraulic conductivity, partial stream penetration into the aquifer and heterogeneity of the aquifer porous media.

Numerical groundwater flow models such as MODFLOW provide more accurate modeling of stream-aquifer systems, but also require extensive field data collection and observation well records for model calibration and verification. Once calibrated, attempting to link MODFLOW with MODSIM would be computationally intractable. However, use of the linear form of the Boussinesq equation (Eq. B.2) allows application of Green's function to solve the resulting nonhomogeneous boundary value problem (Maddock, 1972). Response of the finite difference groundwater modeling system due to external excitations such as pumping, recharge, or infiltration at any point in space and time can be expressed as a set of unit response coefficients independent of the magnitude of the excitation. These response coefficients are similar in concept with those calculated via the Glover model in Eq. B.16, but incorporate more realistic assumptions associated with the MODFLOW model. Although the linear form of the Boussinesq equation is only valid for confined aquifers where transmissivity does not vary with head, it can be applied to unconfined aquifers with reasonable accuracy as long as $\Delta h/\Delta H < 0.10$, where Δh is the change in groundwater elevation and H is saturated thickness of the aquifer.

Although a calibrated MODFLOW model can be applied to generating response functions for stream-aquifer interactions, Maddock and Lacher (1991) developed MODRSP as a variation of MODFLOW designed specifically for calculating response or kernel functions for stream-aquifer interactions. Transient, spatially distributed stream-aquifer response coefficients are automatically generated using MODRSP for allocating groundwater return/depletion flows to multiple return/depletion flow grid cell locations. MODRSP calculates responses for one well or recharge site at a time over the total simulation period, assuming a unit stress has been applied during the first period and discontinued for the remainder of the simulation. Response functions calculated in this way incorporate all of the complex characteristics of the stream-aquifer system into a unique cause-effect relationship.

Since MODRSP is a modification of the USGS MODFLOW finite difference groundwater model, it uses many of the same input data and file structures as MODFLOW. This means that MODFLOW can initially be applied and calibrated for the study area, allowing application of powerful packages such as GMS (The Department of Defense Groundwater Modeling System) (EMRL, 2005) that provide powerful geographic information system (GIS) tools for preparation of MODFLOW data sets. The MODFLOW input data sets developed from application of GMS can, for the most part, be directly utilized in MODRSP, although MODFLOW must be applied in conjunction with the constant transmissivity assumption associated with the linear form

of the Boussinesq equation. Additional assumptions that must be adhered to in applying MODFLOW for application of MODRSP include:

- The MODRSP river package does not require data on river stage height and the head at the bottom of the streambed. For return flow/depletion responses in river reaches, MODRSP assumes uniform water levels over each stream reach that are constant over each stress period. Flow conditions in the stream are assumed to vary insignificantly during stress periods. If streams go dry or overflow their banks during a stress period, it is assumed such events are of short duration and have negligible effect on long-term stream-aquifer interaction.
- Since all starting heads are set to zero in MODRSP, a starting head input file is not required.
- Since MODRSP is a linear model, transmissivity and storage coefficients are considered constant and must be entered as input data.
- It is unnecessary to prepare a well package since pumping data are not read into MODRSP.
- Response coefficient output data generated by MODRSP can be formatted to include well/recharge grid location, response grid location, stress period, and calculated response coefficient for that period. Typical database structure for response coefficient output data is presented in Table 2.

The following modifications were made to MODRSP by Fredericks and Labadie (1995):

- The program was modified to allow dynamic array dimensioning up to available RAM memory.
- The modules RRIV.FOR and RPGM.FOR source code were modified to reduce unnecessary output to a river response file. In line 1 of the RRIV input file, field 41 to 50, a decimal value for the variable, RDROP, can be input. Response coefficients lower than this value will not be printed to the river response output file, thereby reducing the size of the river response output file by eliminating zero value response functions.
- The modules RRIV.FOR and RPGM.FOR were modified to terminate a processing loop for a specific well when the calculated response coefficient values fall below a specified lower limit.
- The modules RRIV.FOR and RPGM.FOR were modified to read in a river reach file that assigns a specific river reach value to each river reach grid cell and then sums the response coefficients by river reach.
- The module RPGM.FOR was modified to read in a recharge site file that assigns a recharge site number to each well grid cell number.

An example finite difference grid for modeling a portion of the South Platte River basin in Colorado is shown Fig. B.5 (Fredericks, et al., 1998). For this case study, two different sets of response coefficients were generated: numerical coefficients calculated using the MODRSP finite difference groundwater model and analytical coefficients calculated with the Glover equation using predefined *sdf* values. The response functions

for a particular river reach in the study area are compared in Fig. B.6. It was found that use of the analytically-based *sdf* coefficients produces significantly lower net river return flow values when compared with coefficients derived from the numerically based finite difference model.

RIVER CAPTURE RESPONSE FUNCTIONS

RIVER REACH #	K	I	J	PUMP WELL #	K	I	J	TIME PER	RF [0]
1	1	2	2	1	1	5	10	1	.1644417E-02
2	1	2	3	1	1	5	10	1	.2860665E-02
3	1	2	4	1	1	5	10	1	.3791862E-02
4	1	3	4	1	1	5	10	1	.4539182E-02
5	1	4	4	1	1	5	10	1	.5317831E-02
6	1	4	5	1	1	5	10	1	.6792709E-02
7	1	4	6	1	1	5	10	1	.8866128E-02
8	1	4	7	1	1	5	10	1	.1095925E-01
9	1	4	8	1	1	5	10	1	.1415163E-01
10	1	4	9	1	1	5	10	1	.1891528E-01
11	1	4	10	1	1	5	10	1	.2429373E-01
12	1	4	11	1	1	5	10	1	.1800508E-01
13	1	3	11	1	1	5	10	1	.1408879E-01
14	1	3	12	1	1	5	10	1	.1173625E-01
15	1	3	13	1	1	5	10	1	.9830805E-02

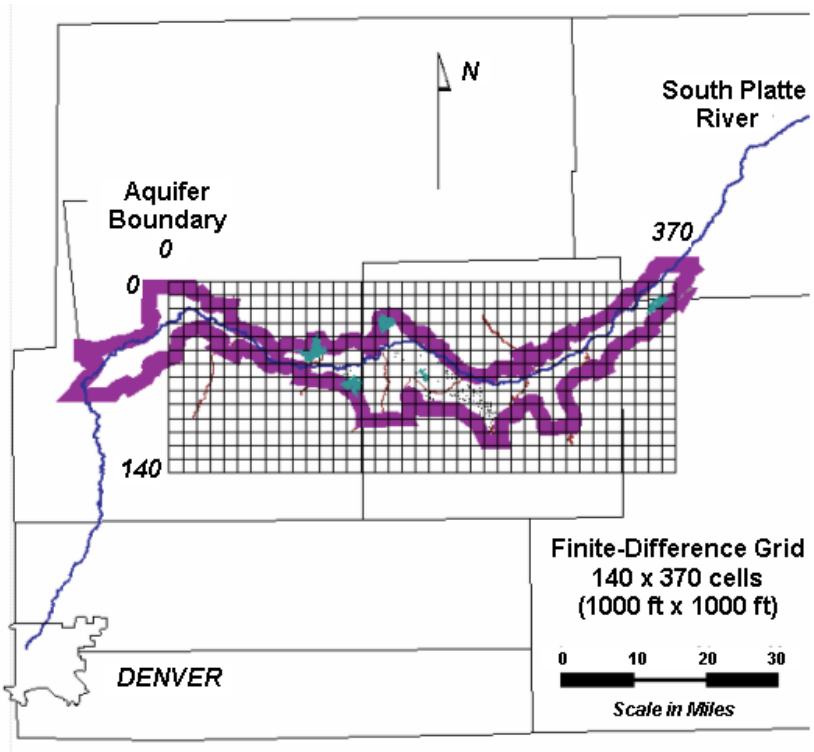


Fig. B.5 MODRSP finite difference grid for numerical groundwater model.

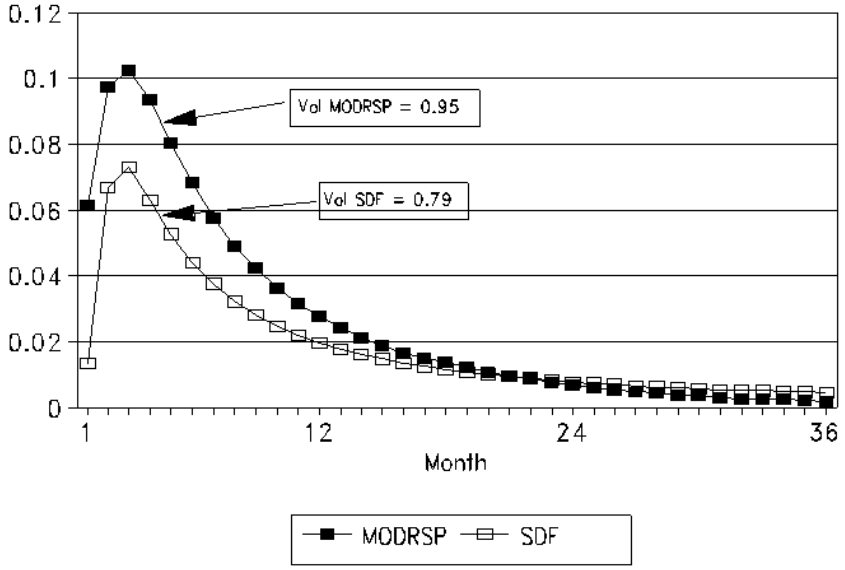


Fig. B.6 Comparison of MODSRP and *sdf* computed response functions.

References

Bear, J. (1979). *Hydraulics of Groundwater*, McGraw Hill, NewYork.

Environmental Modeling Research Laboratory. (2005). GMS Version 6.0. User Manual, Brigham Young University, Provo, UT.

Fredericks, J. and J. Labadie (1995). Decision support system for conjunctive stream-aquifer management. Open File Report No. 10, Colorado Water Resources Research Institute, Colorado State University, Fort Collins, CO.

Fredericks, J., J. Labadie, and J. Altenhofen (1998). Decision support system for conjunctive stream-aquifer management. *Journal of Water Resources Planning and Management*, ASCE, *124*(2), 1998.

Glover, R. (1960). Mathematical derivations as pertaining to groundwater recharge. Agricultural Research Service, USDA, Fort Collins, CO.

Glover, R. (1968). The pumped well. Technical Bulletin No. 100, Colorado State University Experiment Station, Fort. Collins, CO.

Glover, R. (1977). *Transient Groundwater Hydraulics*, Water Resources Publications, Highland Ranch, CO.

Glover, R. and G. Balmer (1954). River depletion resulting from pumping a well near a river. *Transactions of the American Geophysical Union*, *35*(3), 468-470.

Hantush, M. (1959). Non-steady flow to flowing wells in leaky aquifers. *Transactions of the American Geophysical Union*, *37*, 702-714.

Hantush, M. (1967). Growth and decay of groundwater-mounds in response to uniform percolation. *Water Resources Research*, *3*(1), 227-234.

Hantush, M. and M. Mariño. (1989). Chance-constrained model for management of stream-aquifer system. *Journal of Water Resources Planning and Management*, ASCE, *115*(3), 259-277.

Harbaugh, A., E. Banta, M. Hill, and M. McDonald. (2000). MODFLOW-2000, The U.S. Geological Survey Modular Ground-water Model. U.S. Geological Survey Open-File Report 00-92, Washington DC..

Hartwell, A. (1987). Artificial recharge in the South Platte River basin, Colorado: A comparison of four models. M.S. Thesis, Department of Civil Engineering, Colorado State University, Fort Collins, CO.

Hunt, B. (1971). Vertical recharge of unconfined aquifers. *Journal of the Hydraulics Division*, ASCE, *97*(HY7), 1017-1030.

Hurr, R. and A. Burns. (1980). Stream-aquifer management model of the South Platte River Valley, Northeastern Colorado. Preliminary Report, U.S. Geological Survey, Lakewood, CO

Illangasekare, T. and J. Brannon. (1987). Microcomputer based interactive simulator. *Journal of Hydraulic Engineering*, ASCE, *113*(5), 573-582.

Illangasekare, T. and H. Morel-Seytoux. (1982). Stream-aquifer influence coefficients as tools for simulation and management. *Water Resources Research*, *18*(1), 168-176.

Illangasekare, T. and H. Morel-Seytoux. (1986). Algorithm for surface/ground-water allocation under prior appropriation doctrine. *Ground Water*, *24*(2), 199-206.

Jenkins, C. (1968). Computation of rate and volume of stream depletion by wells: Hydrologic analysis and interpretation. in *Techniques of Water-Resources Investigations of the U.S. Geological Survey*, Book 4, Chapter D1, U.S. Printing Office, Washington DC.

Labadie, J., S. Phamwon, and R. Lazaro. (1983). Model for conjunctive use of surface and groundwater: Program CONSIM. Completion Report No. 125. Colorado Water Resources Research Institute, Colorado State University, Fort Collins, CO.

Maasland, M. (1959). Water table fluctuations induced by intermittent recharge. *Journal of Geophysical Research*, *64*(5), 549-559.

Maddock III, T. (1972). Algebraic technological functions from a simulation model. *Water Resources Research*, *8*(1), 129-134.

Maddock III, T. (1974). The operation of stream-aquifer system under stochastic demands. *Water Resources Research*, *10*(1), 1-10.

Maddock III, T. and L. Lacher (1991). MODRSP: A program to calculate drawdown, velocity, storage and capture response functions for multi-aquifer systems. HWR Report No. 91-020, Department of Hydrology and Water Resources, University of Arizona, Tucson, AZ.

Madsen, H. (1988). Irrigation, groundwater abstraction and stream flow depletion. *Agricultural Water Management*, *14*, 345-354.

Male, J. and F. Mueller (1992). Model for prescribing ground-water use permits. *Journal of Water Resources Planning and Management*, ASCE, *118*(5), 543-561.

McWhorter, D. and D. Sunada (1977). *Ground-Water Hydrology and Hydraulics*, Water Resources Publications, Inc., Highland Ranch, CO.

Morel-Seytoux, H. and C. Daly (1975). A discrete kernel generator for stream-aquifer studies. *Water Resources Research*, *11*(2), 253-260.

Morel-Seytoux, H., C. Daly, T. Illangasekare, and A. Bazaraa. (1981). Design and merit of a river-aquifer model for optimal use of agricultural water. *Journal of Hydrology*, *51*(1), 17-27..

Morel-Seytoux, H. and C. Zhang (1990). Anticipation of augmentation needs for allocation operations. *Proceedings Groundwater Engineering and Management Conference*, Colorado Water Resources Research Institute, Colorado State University, Fort Collins, CO, February 28-March 1.

Qazi, R. and J. Danielson (1974). Analytical model for management of alluvial aquifers. *Journal of the Irrigation and Drainage Division*, ASCE, *100*(IR2), 143-152.

Moulder, J. and C. Jenkins (1969). Analog-digital models of stream aquifer systems. *Ground Water*, *7*(5), 19-24.

Rao, N. and P. Sarma (1981). Groundwater recharge from rectangular areas, *Ground Water*, *19*(3), 271-274.

Sophocleous, M., A. Koussis, J. Martin, and S. Perkins (1995). Evaluation of simplified stream-aquifer depletion models for water rights administration. *Ground Water*, *33*(4), 579-588.

Taylor, O. and R. Luckey (1972). A new technique for estimating recharge using a digital computer. *Ground Water*, *10*(6), 22-26.

Warner, J., D. Sunada, and A. Hartwell (1986). Recharge as augmentation in the South Platte River Basin. Technical Report No. 13, Colorado Water Resources Research Institute, Fort Collins, CO.

Warner, J., D. Moulden, M. Chehata, and D. Sunada (1989). Mathematical analysis of artificial recharge from basins. *Water Resources Bulletin*, *25*(2), 401-411.

Willis, R. and W. Yeh (1987). *Groundwater Systems Planning and Management*, Prentice-Hall, Englewood Cliffs, NJ.

APPENDIX C
Backrouting in Daily River Basin Operations

C.1 Overview

As discussed in Section VI.D, the channel routing procedure in MODSIM temporarily disconnects the network at each routing link due to the iterative solution procedure that is employed. Because of this, a lower priority water use upstream of a routing link may receive flows in lieu of deliveries to higher priority downstream uses. To overcome this problem, an innovative *backrouting* procedure is incorporated in MODSIM to insure that water deliveries occur in the correct priority. A *look-ahead* approach is adopted whereby water delivery decisions for time step *t* are based on knowledge about future water system requirements using several network runs over future concurrent time steps. Downstream demand time series are *backrouted* upstream to represent flows in the current time step *t* required to pass through the routing link to satisfy future downstream water requirements without unnecessary shortages and spills.

C.2 General Procedure

MODSIM implements network optimization using the Lagrangian relaxation solver, with each time step solved as a separate problem, since network optimization is utilized in MODSIM to *simulate* the allocation of available water resources according to prespecified priorities and rules. The network optimization progresses from the starting simulation date to the ending simulation date without any knowledge about future events or demands in the system. As described in Section VI.D, channel routing in MODSIM over daily time steps is accomplished by defining a routing link with appropriate Muskingum routing coefficients or user-defined lag coefficients. The flow routing is accomplished by removing a portion of the flow from the upstream node defining the routing link, which is then returned to the downstream node of the routing link in future time steps according to the routing coefficients. Iterations in the current time step are repeated until successive network solutions match according to prespecified convergence criteria.

The normal streamflow routing procedure employed in MODSIM will produce correct solutions as long as there is sufficient water to satisfy all demands, whether they are of low or high priority. Difficulties arise when there is insufficient water available to meet all demands, and priorities exist on allocation of water. Under water shortage conditions and priorities on demands, routing time steps longer than one day can cause downstream demands to *pull* water from upstream reservoirs, although they do not receive this water immediately. This causes unnecessary releases of additional water from upstream reservoirs that are in excess of downstream demands. A *backrouting* methodology has been implemented in MODSIM to overcome the problem of excessive reservoir drawdown associated with longer routing periods.

In this approach, water delivery decisions for time step *t* are based on knowledge about future water system requirements that are calculated using several network runs over future concurrent times. The concept of concurrent time networks is based on the fact that in time step *t*, only a fraction of the water available at any location in the network will actually reach the farthest downstream region of the network during the same time step. For example, in Fig. C.1, none of the water available in link **A** will reach the farthest downstream node during period *t*, and only 50% of the water flowing in link **B** will reach the farthest downstream node during the current time step, with the remaining 50% of the water arriving at the most downstream point in time *t+1*. A concurrent time network that represents only the water available at the same time in the farthest downstream region is called a *downstream time network*. The downstream time network is constructed by transforming demands, inflows, groundwater return flows and reservoir storage targets to the corresponding portions of the flow available.

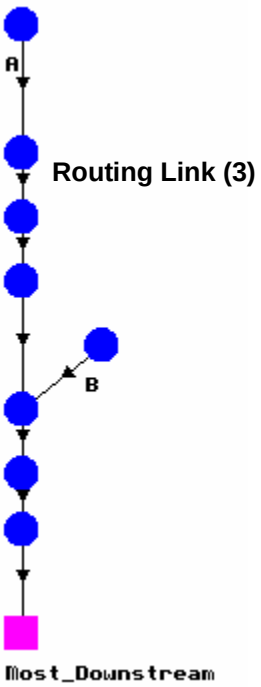


Fig. C.1. Simple real-time MODSIM routing network.

Downstream time networks for current and future time $t+\tau$ ($\tau = 0,\ldots,n$), where n is the maximum number of time lags, combine portions of previous time flows that reach the most downstream region in period $t+\tau$, and the portion of the water available in the system at time $t+\tau$ able to reach the farthest downstream node. Mathematical calculation of the downstream time network flows is accomplished by defining *regional routing coefficients* resulting combination of the multiple routing effects. Routing links in the MODSIM network essentially divide the system into routing *regions* (Fig. C.2) such that flows in each region are considered to be concurrent in time. The combination of routing link coefficients from the farthest downstream region that are *backrouted* to each upstream region allow calculation of regional routing coefficients $\mathbf{r}_j$ that transform the flow time series in region j to an equivalent time series in the farthest downstream region time frame. In this example, Region 1, the farthest downstream region, has a regional routing coefficient vector composed of the value 1 in the first element, followed by 0 values in the remaining elements, since all the water in Region 1 will flow to the farthest downstream region (i.e., itself) in the current time step.

$$\mathbf{c}_1 = \begin{bmatrix} c_{10} \\ c_{11} \\ \vdots \\ c_{1n} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{r}_1 = \begin{bmatrix} r_{10} \\ r_{12} \\ \vdots \\ r_{1n} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and routing coefficients for the next upstream region (Region 2) are represented as:

$$\mathbf{c}_2 = \begin{bmatrix} c_{20} \\ c_{21} \\ \vdots \\ c_{2n} \end{bmatrix}$$

where n represents the position of the first zero valued element in the vector $\mathbf{c}_2$ after the lexicographically last positive-valued element (i.e., $c_{2,n-1} > 0$; $c_{2n} = 0$). The regional routing coefficients for Region 2 are then calculated as a convolution process:

$$\mathbf{r}_2 = \mathbf{R}_1 \cdot \mathbf{c}_2 = \begin{bmatrix} r_{10} & 0 & 0 & 0 \\ r_{11} & r_{10} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ r_{1n} & r_{1,n-1} & \cdots & r_{10} \end{bmatrix} \cdot \begin{bmatrix} c_{20} \\ c_{21} \\ \vdots \\ c_{2n} \end{bmatrix} = \begin{bmatrix} r_{20} \\ r_{21} \\ \vdots \\ r_{2n} \end{bmatrix}$$

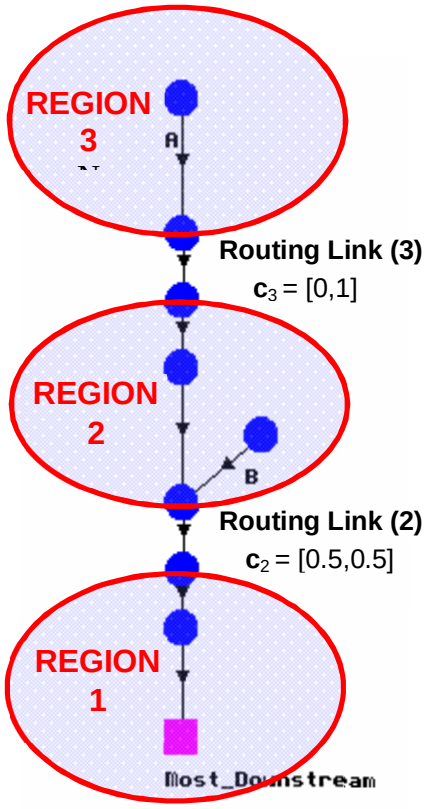


Fig. C.2.. Routing regions in MODSIM Network.

For the example shown in Fig. C.2, $\mathbf{r}_2$ would be calculated as:

$$\mathbf{r}_2 = \mathbf{R}_1 \cdot \mathbf{c}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

This convolution process continues upstream for Region j:

$$\mathbf{r}_j = \mathbf{R}_{j-1} \cdot \mathbf{c}_j = \begin{bmatrix} r_{j-1,0} & 0 & 0 & 0 \\ r_{j-1,1} & r_{j-1,0} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ r_{j-1,n} & r_{j-1,n-1} & \cdots & r_{j-1,0} \end{bmatrix} \cdot \begin{bmatrix} c_{j0} \\ c_{j1} \\ \vdots \\ c_{jn} \end{bmatrix} = \begin{bmatrix} r_{j0} \\ r_{j1} \\ \vdots \\ r_{jn} \end{bmatrix}$$
$$(\text{for } j = 2,...,N)$$

where N is the total number of the regions and n represents the position of the first zero valued element in the vector $\mathbf{c}_j$ after the lexicographically last positive-valued element (i.e., $c_{j,n-1} > 0$; $c_{jn} = 0$), or the position of the first zero valued element in the regional routing vector $\mathbf{r}_{j-1}$ (i.e., $r_{j-1,n-1} > 0$; $r_{j-1,n} = 0$), whichever is larger. For the simple example of Fig. C.2:

$$\mathbf{r}_3 = \mathbf{R}_2 \cdot \mathbf{c}_3 = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

Let q_{jt} represent the time series of inflows, demands or groundwater return flows in Region *j*. These time series are routed to the farthest downstream region time using the computed regional routing matrices $\mathbf{R}_j$ as follows:

$$\mathbf{q}'_j = \mathbf{R}_j \cdot \mathbf{q}_j = \begin{bmatrix} r_{j0} & 0 & 0 & 0 \\ r_{j1} & r_{j-1,0} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ r_{j-1,m} & r_{j-1,m-1} & \cdots & r_{j-1,0} \end{bmatrix} \cdot \begin{bmatrix} q_{jt} \\ q_{j,t+1} \\ \vdots \\ q_{j,t+m} \end{bmatrix} = \begin{bmatrix} q'_{jt} \\ q'_{j,t+1} \\ \vdots \\ q'_{j,t+m} \end{bmatrix}$$

where *m* represents the position number of the first nonzero element in the regional routing vector associated with the farthest upstream region *N*. That is, *m* is defined such that for $\left(r_{N0},...,r_{N,m-1},r_{Nm},r_{N,m+1},...,r_{Nm}\right)$, elements $r_{N0} = \mathrm{L} = r_{N,m-1} = 0$, $r_{Nm} > 0$. The farthest upstream region is used for defining *m* since the multiplied effects of downstream routing will be maximized in the farthest upstream region, which determines the number of future time networks that will need to be solved.

The downstream time series represent concurrent flows in the downstream time $t + \tau$, where τ is the index of future time steps. Fig. C.3 shows successive downstream network for two time steps. Notice that there are no routing links in the downstream networks and all water users and reservoirs compete for concurrent water in a continuous network. At every sequential real time step *t*, a series of *m*+1 downstream time networks over the current and future time steps are run to calculate current and future water requirements for the routing links in the real network. The flows $q'_{j,t+\tau}$ $\left(\tau = 0,1,...,m\right)$ calculated in these network solutions represent flows in routing link *j*, but translated to downstream region time $t+\tau$. These flows are now *backrouted* to time *t* using the inverse of the regional routing coefficient matrix $\mathbf{R}_j$ as follows:

$$\hat{\mathbf{q}}''_j = \left(\mathbf{R}_j\right)^{-1} \cdot \mathbf{q}'_j = \begin{bmatrix} r_{j\ell} & 0 & 0 & 0 \\ r_{j,\ell+1} & r_{j\ell} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ r_{j,\ell+m} & r_{j,\ell+m-1} & \cdots & r_{j\ell} \end{bmatrix}^{-1} \cdot \begin{bmatrix} q'_{j\ell} \\ q'_{j,\ell+1} \\ \vdots \\ q'_{j,\ell+m} \end{bmatrix} = \begin{bmatrix} \hat{q}''_{j\ell} \\ \hat{q}''_{j,\ell+1} \\ \vdots \\ \hat{q}''_{j,\ell+m} \end{bmatrix}$$

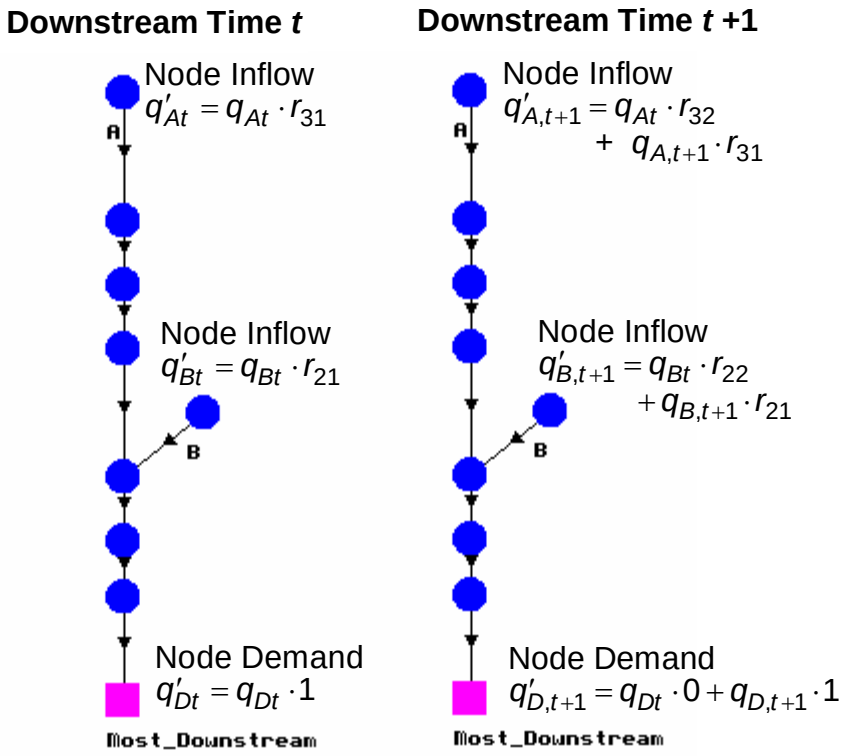


Fig. C.3. Downstream Time Networks Solved at Each Real-Time Step *t*

where ℓ represents the lexicographically first nonzero element in the regional routing vector $\mathbf{r}_j$. Defining the index ℓ truncates the regional routing matrix by removing 0 valued columns and rows in the matrix. This prevents the routing matrix from being singular and therefore allows calculation of the matrix inverse. Since there is no routing occurring prior to time step $t+\ell$ (i.e., the regional routing coefficients = 0 for those time steps), then $q''_{j\tau} = q'_{j\tau}$, for $\tau = 0,...,\ell-1$. Therefore, the final backrouted flows translated to current time *t* for routing link *j* are:

$$\mathbf{q}''_j = \begin{bmatrix} q'_{j0} \\ \vdots \\ q'_{j,\ell-1} \\ \hat{q}''_{j\ell} \\ \vdots \\ \hat{q}''_{j,\ell+m} \end{bmatrix} = \begin{bmatrix} q''_{j0} \\ \vdots \\ q''_{j,\ell-1} \\ q''_{j\ell} \\ \vdots \\ q''_{j,\ell+m} \end{bmatrix}$$

In effect, the backrouted flows q''_{j0} represent flows in the current real-time step *t* required in the routing link downstream of Region *j* to meet future water requirements without unnecessary shortages and spills. Note that even though the entire backrouted vector $\mathbf{q}''_j$ is calculated, only the first element is actually used in time step *t* since future backrouted flows will be recalculated when the simulation moves to the next time step *t*+1.

Invoking backrouting only requires selection of *Backrouting* under *Extensions* in the MODSIM main menu item (Fig. C.4). The same information as required for channel routing is necessary for the backrouting procedure.

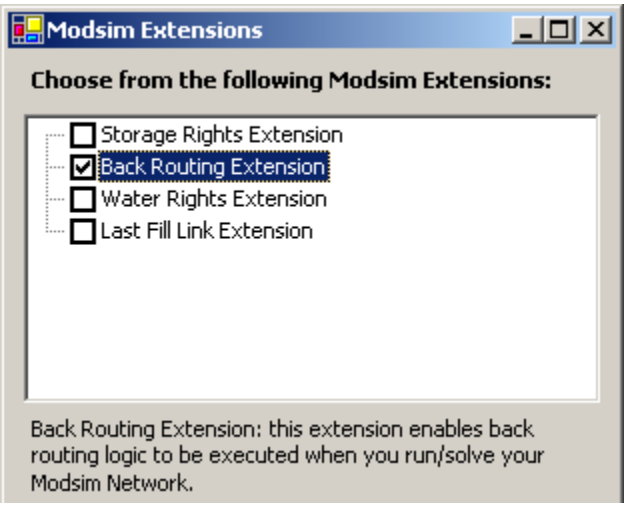


Fig. C.4. Selection of backrouting as a MODSIM Extension.

C.3 Backrouting Example

This example attempts to illustrate the advantages of using the backrouting procedure to insure that flow routing does not interfere with the priorities associated with water allocation in the network, particularly during drought or low-flow conditions. Fig. C.5 shows a simple network with a low priority reservoir node as a limited source of water, and two demands separated by a routing link. The node *dem2* has a higher priority over the node *dem1* in the water delivery, but both have the same water demand of 100 units.

This example represents a case with time lags of more than one day, where the routing coefficients are zero in the first time step, 0.20 in the second, and 0.80 time step. In order to *precondition* the network for previous flow conditions due to flow routing (i.e., prior to the starting time step), 80 units of flow are returned to node ms2 in the first time step, since the real system will rarely be completely dry in the downstream section. The lack of returning routed flows in the first time step would cause excessively large reservoir releases, thereby moving the solution farther from the real operation.

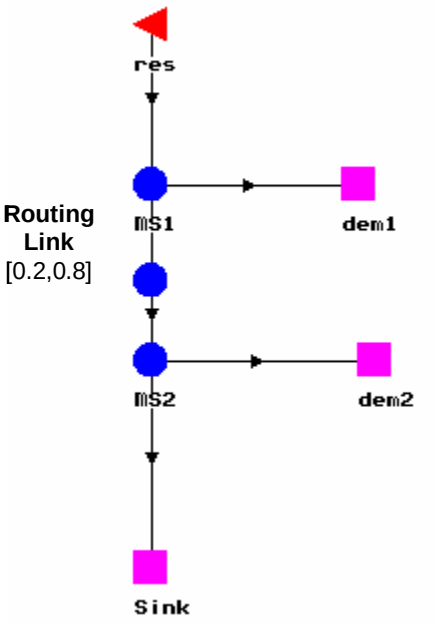


Fig. C.5. Backrouting example.

The results in this example demonstrate the *look-ahead* capabilities of back routing algorithm in the way that MODSIM solves the network. The physical channel routing solution (i.e., without backrouting) releases all water from the reservoir in an attempt to

satisfy the senior demands downstream of the routing link. The flow in the link going out the reservoir is shown in Fig. C.6.

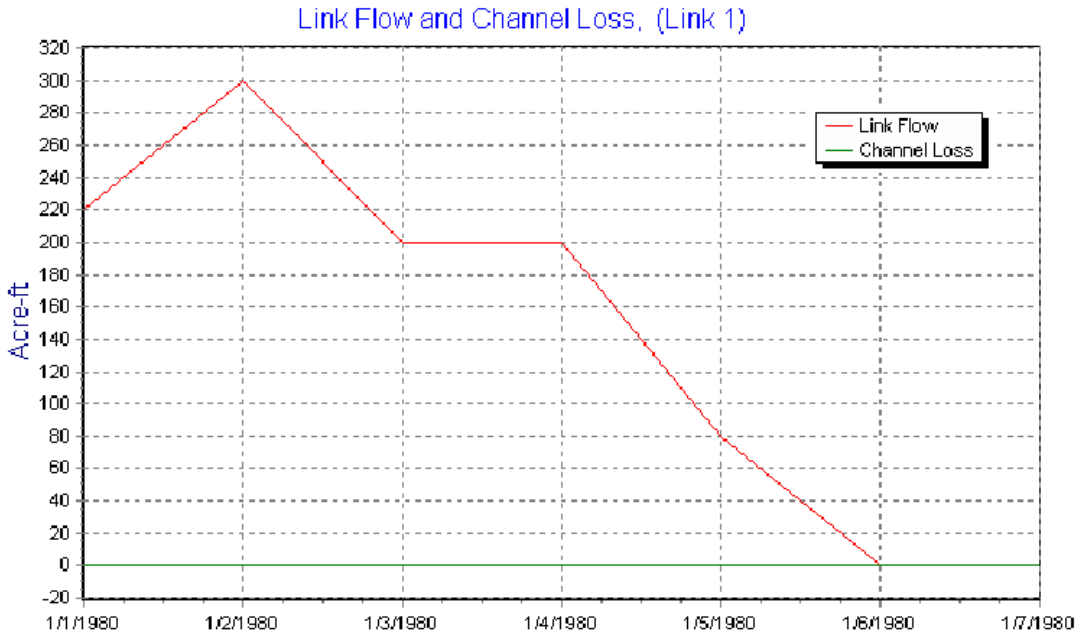


Fig. C.6. MODSIM Results for Streamflow Routing Case Showing Large Reservoir Releases

From Fig. C.7, it can be seen that the senior demand receives water mainly from flows added to precondition the network for routed flows prior to the current simulation period. Large shortages are observed for the case where the physical channel routing option is used.

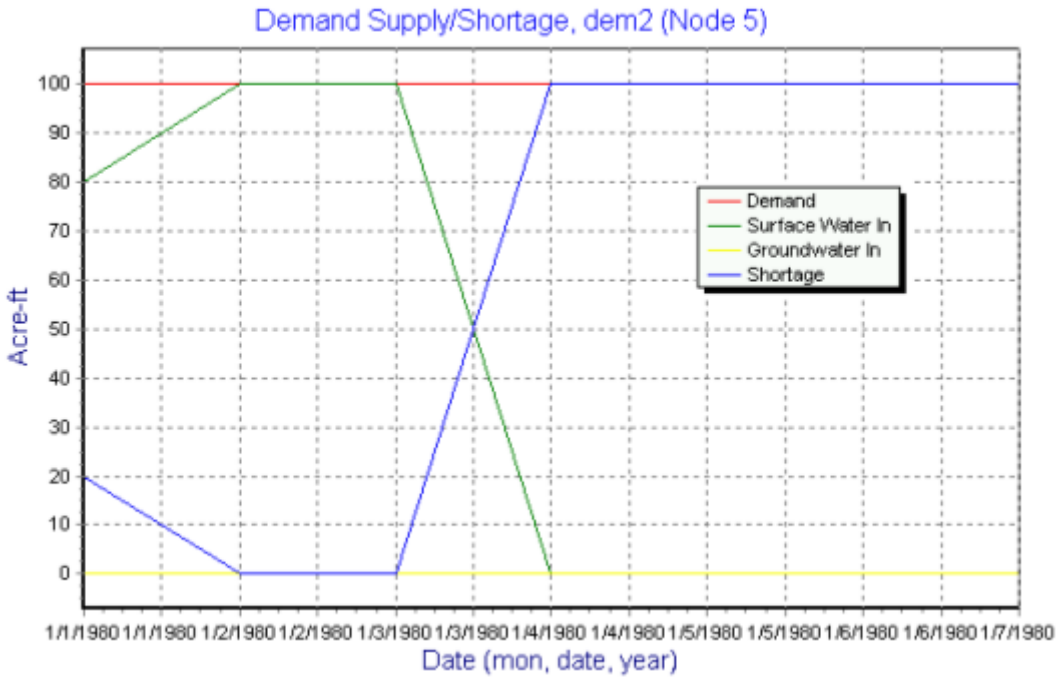


Fig. C.7. Shortages to senior demand under the physical channel routing solution.

In this solution, the junior demand receives no water, and most of the water is spilled at the downstream sink node, as seen in Fig. C.8.

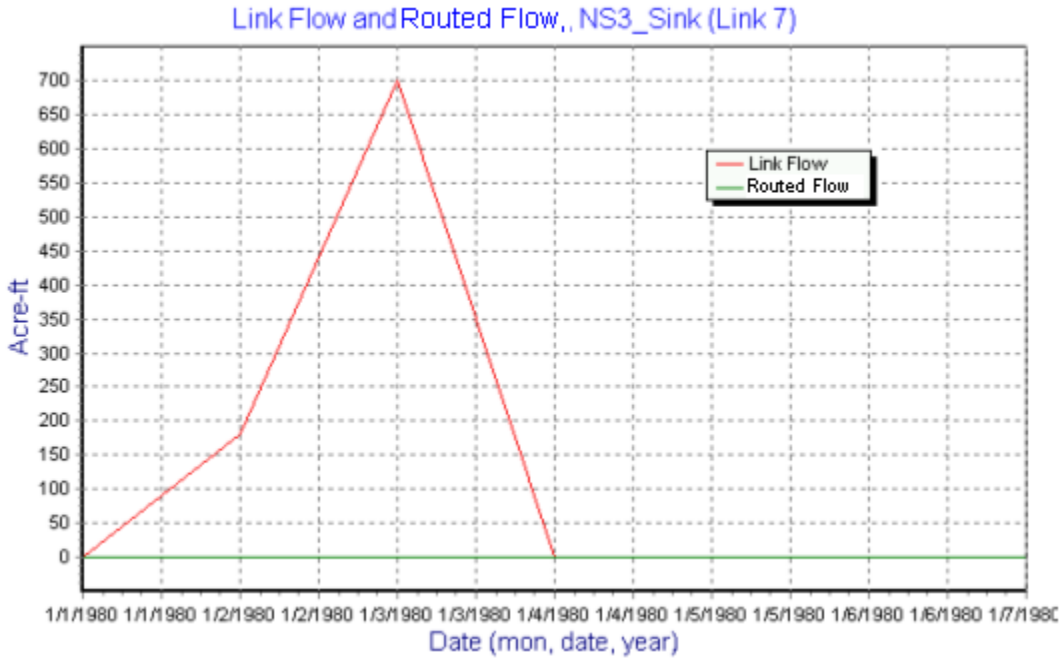


Fig. C.8. Large downstream spills in the physical channel routing solution.

Fig. C.9 shows that under the backrouting solution, the model produces the exact reservoir releases needed to meet demands over the future time steps. In contrast with the solution under physical channel routing, the junior demand (dem1) is able to receive sufficient flows to satisfy the demands through the first four time periods, as shown in Fig. C.10.

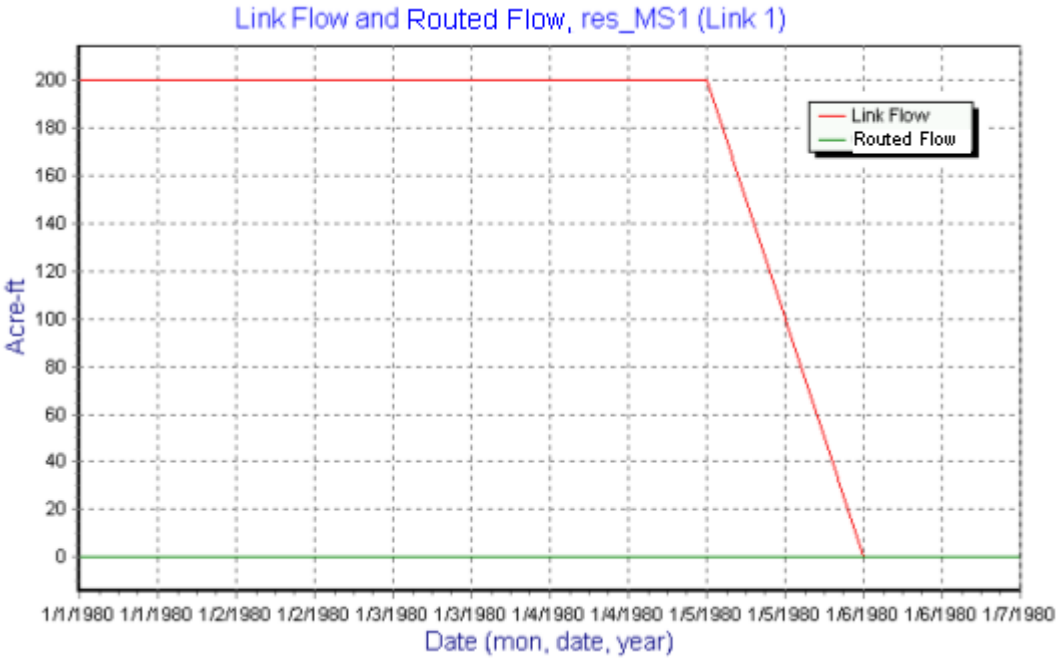


Fig. C.9. Correct reservoir releases under the backrouting solution.

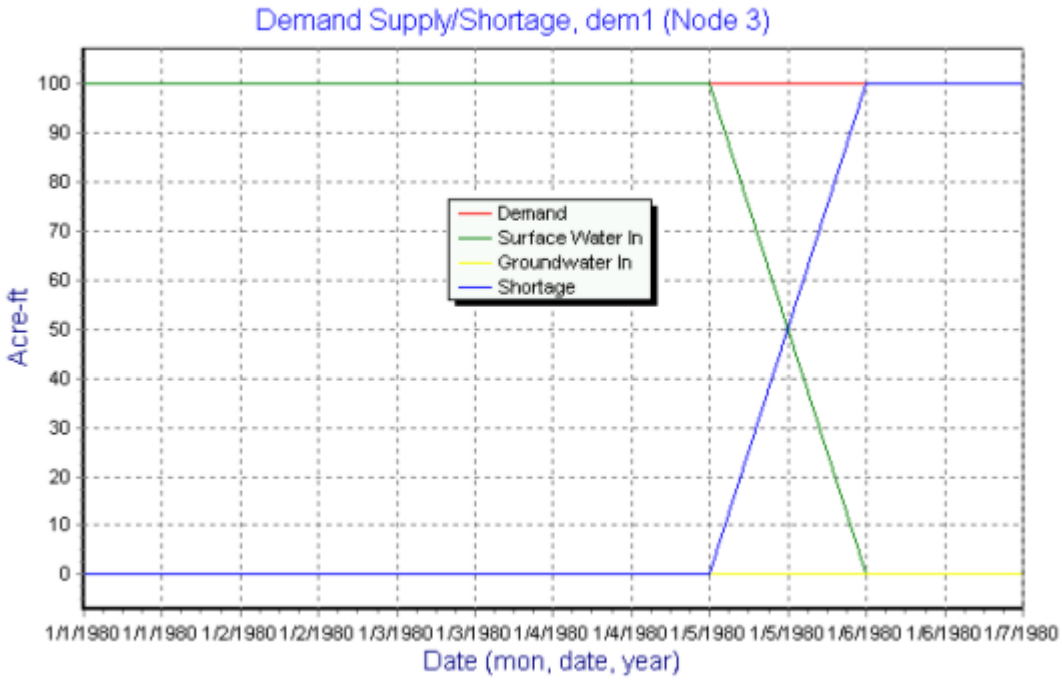


Fig. C.10. Demand satisfaction for low priority demand with backrouting.

Demands for the high priority demand node (dem2) are generally satisfied under backrouting, although some shortage occurs in the first period since it is physically impossible for the senior demand to receive water from the reservoir until the second time step due to the time lags (Fig. C.11). Flow through the routing link under the backrouting procedure is shown in Fig. C.12. The benefits of backrouting are demonstrated by the fact that no spills occur with this solution.

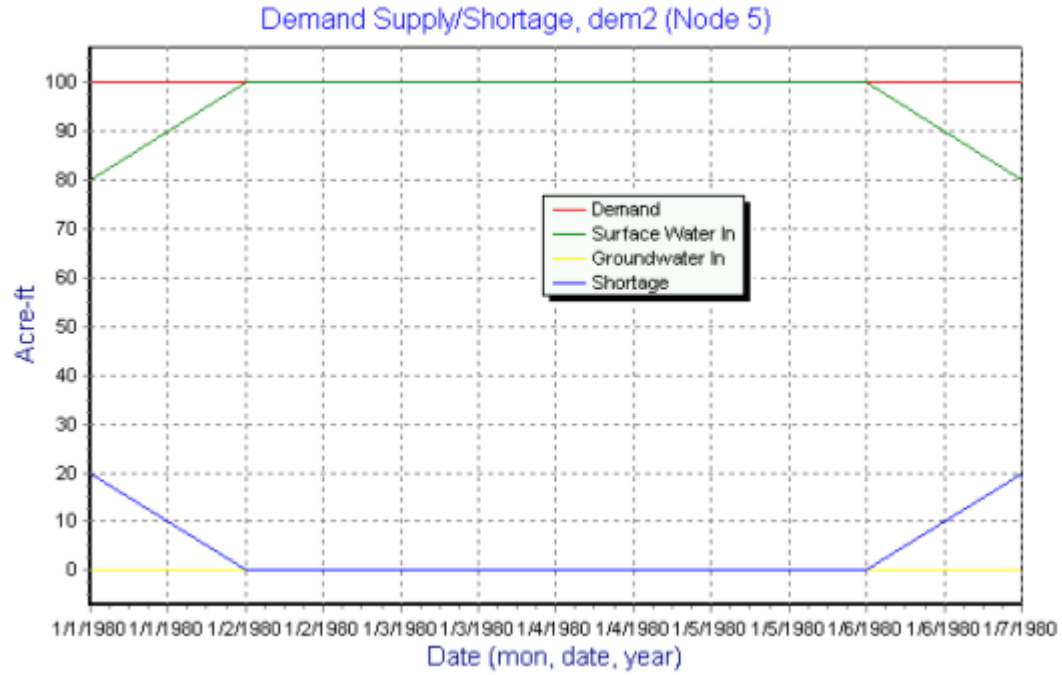


Fig. C.11. Demand satisfaction for the senior demand (dem2) under the backrouting solution.

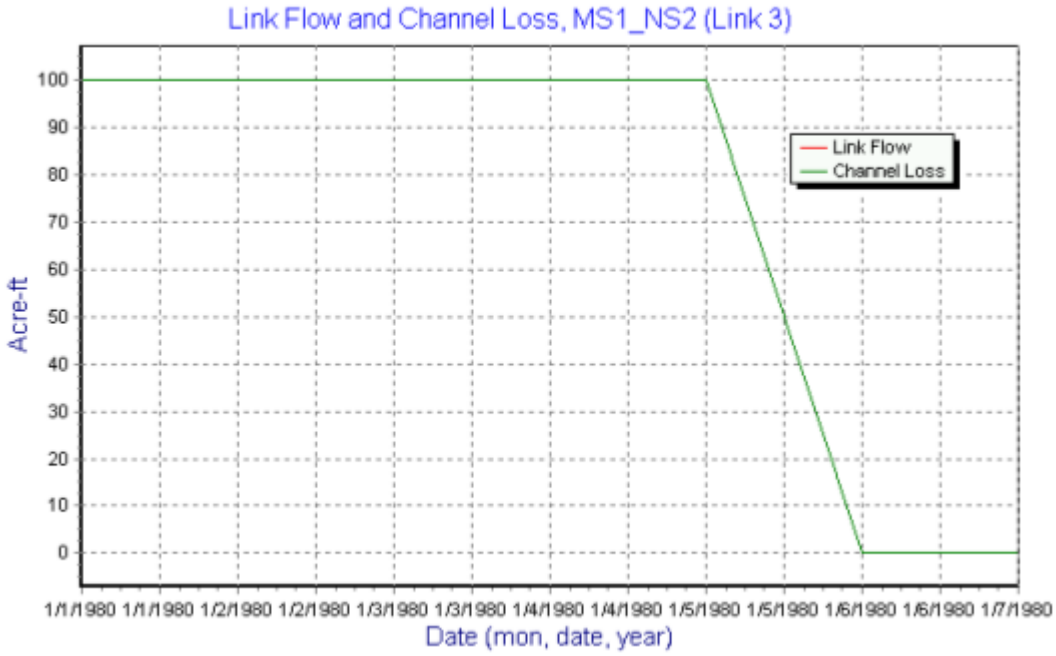


Fig. C.12. Flow through the routing link for the backrouting solution.