**A Methodology for Linking 2D Overland Flow Models with the Sewer Network Model SWMM 5.1 Based on Dynamic Link Libraries.**

Jorge Leandro, Ricardo Martins

**Supplementary Material**

int DLLEXPORT Swmm\_Link(double\* NodeID,int\* NodeID\_j, double\* S\_Node\_Crest, double\* S\_Node\_Bottom,double\* TotalDuration2, double\* SWMM\_DT)

//

// Input: NodeID = Node ID to add discharge flow

// NodeID\_j = Node ID j index to add discharge flow

// TotalDuration2 = total SWMM duration

// SWMM\_DT = RouteStep SWMM

// Output: returns error code

// Purpose: identify the j index of each node, total duration and route step.

//

{

 int j;

 double xid;

 for (j = 0; j < Nobjects[NODE]; j++)

 {

 xid= strtod(Node[j].ID, NULL);

 if ((Node[j].type!=1) && xid==\*NodeID)

 {

 \*NodeID\_j=j;

 \*S\_Node\_Crest=(Node[j].invertElev+Node[j].fullDepth)\* UCF(LENGTH);

 \*S\_Node\_Bottom=(Node[j].invertElev)\* UCF(LENGTH);

 }

 }

 \*TotalDuration2=TotalDuration/1000.00; //(sec)

 \*SWMM\_DT=RouteStep; //(sec)

 return ErrorCode;

}

int DLLEXPORT Swmm\_to\_2D( int\* NodeID\_j, double\* S\_Node\_Watlevel)

//

// Input: NodeID\_j = Node ID j index to exchange discharge flow

// S\_Node\_Watlevel = water level elevation

// Output: returns error code

// Purpose: obtain the water level in SWMM Node.

//

{

 int j;

 double factor;

 j=\*NodeID\_j;

 \*S\_Node\_Watlevel=(Node[j].invertElev+Node[j].newDepth)\* UCF(LENGTH);

 return ErrorCode;

}

int DLLEXPORT 2D\_to\_Swmm(int\* NodeID\_j, double\* Node\_Q)

//

// Input: NodeID = Node ID to exchange discharge flow

// Node\_Q = discharge value to exchange

// Output: returns error code

// Purpose: exchange discharge from 2D to SWMM.

//

{

 int j;

 double factor;

 j=\*NodeID\_j;

 Node[j].extInflow->baseline = \*Node\_Q;

 return ErrorCode;

}