Stella Modeler’s Note

1. Program structure
   1. Program UI
   2. Simulation file
2. Program simulation

Python Stella developer’s note

1. Structure:
   1. Loading input configuration
   2. Loading excel data input
   3. Initial Stocks
   4. Simulation loop
      1. Getting current stock’s value
      2. Matrix calculation
         * Stella matrix calculation

Stella matrix calculation

Problems:

In Stella, it is very easy to process matrix calculation (add, sub, multiply, divide, sum). We only need it has a same dimension, we can shape the output. For example:

[[1, 2, 3], [4, 5, 6]] + [[1], [2]] = [[2, 3, 4], [6, 7, 8]]

We can easy do:

A(2, 3) + B(2, 1) = C(2, 3) or shape the output: A(2, 3) + B(2, 1) = C(3, 2)

A(3, 2) + B(2, 1) = C(2, 3) or shape the output: A(3, 2) + B(2, 1) = C(3, 2)

…

In Python, we can’t do like that:

We can’t shape the output:

np.add(A(2, 3), B(2, 3)) = C(2, 3), if we want to shape the output like C(3, 2), we can do:

np.add(A(2, 3), B(2, 3)).transpose(), the result will be C(2, 3)

The input is not automatically shaped to compatible:

A(2, 3) + B(1, 2) => error

np.add(A(2, 3), B(2, 1)) or np.add(A(2, 3), B(1, 2).transpose())

In Python numpy, note that a vector and a matrix is not the same:

B(2,) will be a vector

B(2, 1) will be a matrix

We can not process matrix calculation between a matrix and a vector. We should convert vector to matrix using np.array([V]). The result of that command will be a matrix with row = 1 and column = length of the vector.

SUM matrix

Sum(x[\*, column]) will sum all the rows of x, it equals np.sum(x, axis=0)

Sum(x[row, \*]) will sum all the columns of x, equal with np.sum(x, axis=1)

The result of sum(x[\*, column]) will be a vector ( we do not care about the vector and matrix in stella when processing calculation) with the length is the column

The result of np.sum(x, axis = 0) will be a vector ( in Python, we should care about vector and matrix), so we may need to covert it to a matrix if we want to do later calculation.

To make it convenient when processing with data in Python, we should follow the rules:

- The subcatchment will be treated as a vertical vector

- The matrix with subcatchment will act as rows