IO processing for HiPIMS in Python

# System requirements

* Python 3.6 or newer version installed
* Required python packages: **numpy**, **pandas, ﻿matplotlib, ﻿scipy, ﻿shutil**

# Functions for IO processing

***﻿HiPIMS\_setup*** from HiPIMS\_IO.py which is supported by python files: ArcGridDataProcessing.py, ﻿InputSetupFuncs.py, ﻿InputSetupFuncs\_MG.py

The format of calling ***HiPIMS\_setup:***

﻿ ﻿HiPIMS\_setup(folderName, demMat, demHead, numSection=1, boundList=[],

fileToCreate='all', h0=0, hU0=[0,0], manning=0.035,

rain\_mask=0, rain\_source=numpy.array([[0,0],[60,0]]),

sewer\_sink=0, cumulative\_depth=0, hydraulic\_conductivity=0,

capillary\_head=0, water\_content\_diff=0,

gauges\_pos=[], timesValue=[0,3600,600,3600,720000])

The paramters with default value can be omitted when calling ***HiPIMS\_setup,*** for example:

﻿ HiPIMS\_setup(folderName, demMat, demHead)

It will create input data in directory given by ‘folderName’ with DEM data given by ‘demMat’ and ‘demHead’. All the parameters ***HiPIMS\_setup*** in are introduced in Table 1.

Table . List of parameters in python function HiPIMS\_setup

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | | Default value | Optional values | Meanings |
| ﻿folderName | string | | None | Directory to store input data |
| ﻿demMat | Numpy array | | None | Array to store elevation values. It can be read from DEM file using arcgridread |
| ﻿demHead | dict | | None | Spatial reference information of DEM, it can be read from DEM file using arcgridread |
| ﻿numSection | Scalar (int) | | 1 | Number of GPUs |
| ﻿boundList | List of dict with keys: | polyPoints | [] (indicates outline boundary) | 2-col numpy array: x-y coordinates to define boundary frame |
| type | 'open' | 'rigid': To indicate boundary type |
| h | [ ] | 2-col numpy array: time series of water depth |
| hU | [ ] | 3-col numpy array: Time series of water velocity |
| ﻿fileToCreate | String | | ﻿'all' | ﻿'z', 'h', 'hU', 'manning', 'sewer\_sink',  'cumulative\_depth', 'hydraulic\_conductivity',  'capillary\_head', 'water\_content\_diff'  'precipitation\_mask', 'precipitation\_source',  'boundary\_condition', 'gauges\_pos': Files to generate |
| ﻿h0 | Scalar or Numpy array | | 0 | Initial water depth |
| ﻿hU0 | List of scalar or Numpy array | | [0,0] | Initial water velocity |
| ﻿manning | Scalar or Numpy array | | 0.035 | It is the manning coefficient. If it is a scalar, then all the cells in the domain have the same manning value. |
| ﻿rain\_mask | Scalar or Numpy array | | 0 | Serial number of rainfall sources starting from 0. Grids with the same serial number will have the same rainfall from the same source. |
| ﻿rain\_source | Numpy array | | ﻿[0,0;  60,0] | To give rainfall value for different region of the domain. The first column is time(s) and the second and right forward columns are the rainfall rate(m/s), and the output rainfall source file will be a single file named ‘precipitation\_source\_all.dat’. The number of the single array column should be in accordance with the number of rainfall source in rain\_mask. |
| ﻿sewer\_sink | Scalar or Numpy array | | 0 | It is sewer sink rate (m/s). If it is a scalar, then all the grids in the domain have the same sewer sink value. |
| ﻿cumulative\_depth | Scalar or Numpy array | | 0 | It is one of the infiltration parameters. If it is a scalar, then all the grids in the domain have the same cumulative depth value. |
| ﻿hydraulic\_conductivity | Scalar or Numpy array | | 0 | It is one of the infiltration parameters. If it is a scalar, then all the grids in the domain have the same hydraulic conductivity value. |
| ﻿capillary\_head | Scalar or Numpy array | | 0 | It is one of the infiltration parameters. If it is a scalar, then all the grids in the domain have the same capillary head value. |
| ﻿water\_content\_diff | Scalar or Numpy array | | 0 | It is one of the infiltration parameters. If it is a scalar, then all the grids in the domain have the same capillary head value. |
| ﻿gauges\_pos | Numpy array | | [ ] | 2-col numpy array: Coordinates of the gauge points inside domain |
| ﻿timesValue | Numpy array | | ﻿[0,3600,600,1800] | 4-element numpy array of model time: ﻿start, end, output interval, backup interval |

Structure of data folder for HiPIMS

The structure of data folder for single GPU and multi-GPU are different.

## Single GPU:

The structure of data folder for single GPU model is illustrated as follows. A bold name indicates a folder rather than a file in the table.

|  |  |  |  |
| --- | --- | --- | --- |
| Contents in 1st directory | Contents in 2nd directory | Contents in 3rd directory | Notes |
| input | **mesh** | DEM.txt | Topography data |
| **field** | >=15 files | Files listed in *Table 1* |
| output | Output files |  | Ascii and time series files |
| times\_setup.dat |  |  | To define model run time |
| device\_setup.dat |  |  | To choose a GPU device |

## Multiple GPUs:

The structure of data folder for *N* GPUs model is illustrated as follows. Each folder in the 1st directory has the same structure with folder ‘**0**’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Contents in 1st directory | Contents in 2nd directory | Contents in 3rd directory | Contents in 4th directory | Notes |
| 0 | **input** | **mesh** | DEM.txt | Topography data |
| **field** | >=15 files | Files listed in *Table 1* |
| **output** | output files |  | Ascii and time series files |
| 1 | **…** | … | … | The same structure with folder 0 |
| … |  |  |  |  |
| *N*-1 | **…** | … | … | N is the number of GPUs |
| times\_setup.dat |  |  |  | To define model run time |
| device\_setup.dat |  |  |  | To choose GPU devices |
| halo.dat |  |  |  | To define the shared boundary rows between sub domains |

List of model input files

Mesh file:

DEM.txt spatial reference of the DEM raster of the model domain in ArcGrid data format

Field files:

There are typically 15 field files if one boundary condition was given for depth and velocity. More field files would be created if there were more boundary conditions. The fundamental files are introduced in Table 1.

*Table 2. List of field files*

|  |  |  |
| --- | --- | --- |
| File name | File type | Notes |
| *z.dat* | Basic terrain | ID, type, and elevation values of each cell |
| *h.dat/eta.dat* | Initial conditions | initial water depth/elevation (m) of each cell |
| *hU.dat* | initial water velocities (m/s) of each cell |
| *precipitation.dat* | initial rainfall rate (m/s) of each cell |
| *manning.dat* | Parameters | friction coefficient of each cell |
| *hydraulic\_conductivity.dat* | hydraulic conductivity of each cell |
| *cumulative\_depth.dat* | cumulative depth of each cell |
| *capillary\_head.dat* | capillary head of each cell |
| *water\_content\_diff.dat* | water content difference of each cell |
| *sewer\_sink.dat* | sewer sink rate of each cell |
| *h\_BC\_0.dat* | Boundary conditions | water depth on the first boundary |
| *hU\_BC\_0.dat* | water velocities on the first boundary |
| *precipitation\_mask.dat* | Rainfall sources | rainfall source ID of each cell |
| *precipitation\_source\_all.dat* | time series of all rainfall sources |
| *gauges\_pos.dat* | Monitors | coordinates of monitoring points in the model |

Other files:

*times\_setup.dat* Four values respectively indicate model start time, total time, output interval, and backup interval in seconds

*device\_setup.dat* Values to define the ID of GPU devices that used to run model in each subdomain.