|  |
| --- |
| Centre_for_Water_Systems_rgb.jpg |
| CADDIES:  caflood application |
| USER GUIDE |
|  |
| **Michele Guidolin, Albert S. Chen, Nicola Pasquale** |
| **April 2015** |

|  |
| --- |
| Application Version 110 |

****

# Introduction

The **caflood** application performs a 2D pluvial flood inundation simulation using cellular automata (CA) techniques instead of solving the classic shallow water equations (SWEs). The CA technique offers a versatile method for modelling complex physical systems using simple operations (Wolfram 1984). This simplification dramatically reduces the computational load of a CA model in comparison to a physically based model.

CA model usually consists of five essential features: a discrete space, the distribution of the neighbour cells, the state of the cells, the discrete time step and the transition rules (Itami 1994). The transition rules are composed of simple operations that govern the evolution of each cell’s state. This makes use of the previous state of the cell itself and those in its neighbourhood. Since computing the new state of a cell does not depend on the state of any other cells at the same time step, CA algorithms are well suited to parallel computation.

The caflood application is part of the **CADDIES** framework/project which is final aim is to produce faster algorithms for handling dual drainage flood modelling, i.e. where the urban surface flow (major system) is combined with the sewer flow (minor system). This is achieved by using CA techniques together with modern parallel hardware.

|  |
| --- |
| ca-2d-1d.png  Figure 1. Example of interactions between 2D surface flow and 1D sewer flow via linking elements (manholes) |

The caflood application implements the Weighted Cellular Automata 2D model (**WCA2D**) with simplified transition rules to simulate inundation events instead of using complex physically based equations and mathematical operations. This model improves the methodology adopted in the CA2D model (Ghimire et al. 2013) using a weight-based system. The WCA2D is a diffusive-like model that ignores inertia terms and momentum conservation. The model has been designed to work with various general grids, (e.g., rectangular, hexagonal or triangular grid) with different neighbourhood types (e.g., the four cells of the von-Neumann (VN) neighbourhood or the eight cells of the Moore neighbourhood).

The major points of this model are:

* The ratio of water transferring from the central cell to the downstream neighbour cells (intercellular-volume) is calculated using a minimalistic and quick weight-based system;
* The volume of water transferring between the central cell and the neighbour cells is limited by a single equation, which is a composition of a simplified Manning’s formula and the critical flow condition.
* Both the adaptive time step and the velocity, as an average velocity, are evaluated within a larger update time step to speed up the simulations.

Additional information about the WCA2D model is available in (Guidolin et al. 2015)

The caflood application uses the **CADDIES Application Programming Interface** (API) to implement the 2D model (Guidolin et al. 2012). The CADDIES API defines a standard set of methods, data structures and variables that can be used to develop parallel CA algorithms. A developer needs to write the code of the CA model only once. After that, the CADDIES API gives the flexibility to produce the same CA model for any type of CA grid, square/hexagonal/triangular grid, and to use different high performance acceleration techniques without changing the code or with minimum effort.

In caflood, the WCA2D model has been implemented using only a square cell grid with von-Neumann neighbourhood. Furthermore, thanks to the CADDIES API, a simulation can be executed in a multi-core CPU using OpenMP library (Dagum and Menon 1998) and in a multi-core CPU and on a graphics card GPU using the OpenCL library (Munshi and others 2011).

# CAFLOOD EXECUTABLE

The caflood application is composed of a single executable, called ‘**caflood**’, which can be executed from a command line shell, like Windows PowerShell of UNIX Bash. As previously mentioned, caflood can run a simulation on a multi-core CPU or in a graphics card GPU. Furthermore, the simulation can be performed with either single precision or double precision floating point values. These different characteristics are identified at compile time to improve performance; thus, it is possible to have multiple versions of the caflood executable.

The features of the application are:

* Can run on multi-core CPU and on graphics card GPU;
* Can be easily controlled by simple CSV set-up files;
* Uses an adaptive time step;
* Can simulate three types of events (rain fall, inflow and water level rise/fall) in all the domain or in a specific area of the domain;
* Uses a single roughness value for all the domain;
* Can compiled to use single precision or double precision floating point values;
* Takes as input only ASCII/GRID DTM/DEM files with projected coordinate system;
* Can produce ASCII/GRID raster file of the water depth/level and velocity at specific time steps and for the maximum values;
* Can produce CSV files with the time plot of the water depth/level and velocity at specific points in the domain.

This caflood application was tested only on a Windows 7 64 bits operating system (OS) and on a Linux 64 bit OS.

# Execution and configuration file examples

In order to run a single simulation using the compiled executable, **caflood** requires three arguments, which must be given in the following order when the command is executed: 1) an input folder, 2) a setup-file, and 3) an output folder. The user needs to produce a comma separated value (CSV) setup-file which can have any name but must be located inside the input folder. The file is a list of command, files, instructions and parameters that the model need to run a simulation and produce one or more output. In the next paragraph, there is a synthetic example of a setup-file called Sim.csv.

**[Sim.csv]**

1. Simulation Name , Simple Sloping Domain Simulation
2. Short Name (for outputs) , simple
3. Pre-proc Name , simple
4. Version , 1,0,0
5. Model Type , WCA2Dv2
6. Time Start (seconds) , 0
7. Time End (seconds) , 43200
8. Max DT (seconds) , 60
9. Min DT (seconds) , 0.01
10. Update DT (seconds) , 60
11. Slope Tolerance (%) , 0.528
12. Max Iterations , 4000000
13. Roughness Global , 0.015
14. Ignore WD (meter) , 0.005
15. Tolerance (meter) , 0.0025
16. Boundary Ele (Hi/Closed-Lo/Open) , -9000
17. Elevation ASCII , simple.asc
18. Rain Event CSV , Rain40mm60min.csv
19. Water Level Event CSV ,
20. Inflow Event CSV ,
21. Time Plot CSV , WLpoints.csv, VELpoints.csv
22. Raster Grid CSV , WDraster.csv, VELraster.csv
23. Output Console , true
24. Output Period (s) , 600
25. Output Computation Time , true
26. Check Volumes , true
27. Remove Proc Data (No Pre-Proc) , true
28. Remove Pre-Proc Data , true
29. Raster VEL Vector Field , true
30. Raster WD Tolerance (meter) , 0.01
31. Update Peak Every DT , false
32. Expand Domain , false
33. Ignore Upstream , true
34. Upstream Reduction (meter) , 1.0

This file gives the input instruction for a single simulation. Instructions include the file name, the prefix name which will be used to save the data output files, some model parameters (roughness, time step, input and output setup files etc.). The input and output data needed and produced by the simulation are specified by multiple input-setup-files and output-setup-files (some examples will be presented later), respectively, that must be located in the input folder. The details of each instruction line are described as following:

1. **Simulation Name** is the full name of the simulation.
2. **Short Name** is the short cut of the Simulation Name. The string here defined will be used as a prefix to the names of data output files. The suffix of the data output files is composed of the name of the output-setup-file and one between the time when the data was generated and the name PEAK for the maximum values. In the example case all data output files (velocity, water depth, etc…) will be named automatically simple\_WDraster\_1800.asc, simple\_VELraster\_PEAK.asc, simple\_WLpoints.csv, etc…
3. **Pre-proc Name** in principle is the same as Short Name. This is used to name the preprocessing files created during the initial phase of the simulation. This parameter is useful when a catchment needs multiple different events to be simulated, so the pre-processed catchment data is loaded faster between simulations.
4. **Version** is used to indicate the version of the simulation executed. It can be used to identify the console output of same simulation when it is executed with different parameters. It does not have any extra impact on the simulation and it is safe to leave it as ‘1.0.0’.
5. **Model Type** is usedto indicate the flood model utilized during the simulation. The caflood application is designed to run multiple models. This version can run two models: WCA2Dv1 and WCA2Dv2. The former is the first version of the weight-based model and is going to be deprecated. The latter is the second version which is the one used in the last journal article (Guidolin et al 2015).
6. **Time Start (seconds)** is generally equal to zero, meaning the first time step corresponding to the beginning of the event and it can be used to start a simulation from a certain point in time. In future version of the software, this allows the CADDIES-2D to resume a simulation from a previously produced output.
7. **Time End (seconds)** is the timing to end of the simulation.
8. **Max DT (seconds)** sets the upper limit for the time step DT. The DT is defined as

Where is the length of a cell and is the maximum velocity in the domain ( is introduced later). When velocity is approaching zero, the time step tends to be infinite and thus the computation would stop. Therefore an upper limit of DT has to be defined. The value of *Δt* is automatically updated by the program every *update DT* seconds (see later). Generally an upper limit of 60 seconds is recommended by the software developer.

1. **Min DT (seconds)**: the CADDIES-2D uses adaptive time step to capture the movement of flow front. The time step reduces as the maximum velocity increases. In contrast to the Max DT, very small time step may occur when the velocity is high and it will affect the model performance significantly. Therefore, it is mandatory to set a minimum value. Generally a lower limit of 0.01 seconds is recommended by the software developer.
2. **Update DT (seconds):** *Δt* changes according to the velocity. Theoretically it should be update at every time step; however to compute *Δt* at each and every time step is a time consuming task. The user can set up an interval called update DT in which the same *Δt* t is used. At the end of each interval the new *Δt* is computed. In this version of the software the value cannot be over 60 and must be an integer factor of 60 such as 10, 20 or 30 seconds. Software developer recommends using 60 seconds to not affect the performance and to provide good accuracy.
3. **Slope Tolerance (%):** in the WCA2Dv2 model, the time step*Δt* that the software uses at each update interval (see. Pt. 9) is computed using the formula suggested by Hunter et al. (2005) :

This formula has the drawbacks that when the slope between two cells tends to be zero, the time step is driven to zero. Thus the slope tolerance is used to ignore all the cells where the water slope is less than the tollerance. A rule of thumb is to use an order of magnitude less than the average slope percent of the terrain used. For example, if the average slope is 5.28%, then the value of the slope tolerance should be 0.528.

1. **Max Iterations** It is necessary to set a number just to be sure that the simulation stops in case*Δt* becomes very small.
2. **Roughness Global:** this is the roughness value which is applied to all the cells of the domain.
3. **Ignore WD (meter):** this parameter is used to speed up the model by ignoring the flow dynamic for cells with very shallow water. During the simulation, if a cell at a given time step has water depth (WD) < 0.005 m, the model assumes there is NO OUTFLOW for the cell and skips the computing. However, there might always be inflow from the surrounding cells. In other words, in order to have outflow from a given cell the condition WD ≥ 0.005 m must be satisfied.
4. **Tolerance (meter):** This is another parameter used to speed up the simulation. During the simulation, if two neighbor cells have a water level difference that is less than the tolerance, then the caflood application assumes there is NO OUTFLOW between the pair and skips the computation.
5. **Boundary Ele (Hi/Closed-Lo/Open):** this parameter sets the boundary condition. If a catchment (or a spatial domain) has very low boundary elements, the flow will be free to leave the domain and the exiting volume is a volume lost. On the contrary, by setting a high value of the boundary cell it is like virtually closing the domain and, at the downstream of the catchment or the exit point of the domain, the water keeps accumulating. For our purposes it is better to set a very low and negative value (e.g., -9000).
6. **Elevation ASCII:** terrain input file. It must be a DTM ASCII grid file.
7. **Rain Event CSV:**  this is an input-setup-file(s) and it indicates a rain event which can be applied to the full domain or to a specific given area. The rain can vary in time (hyetograph) and it is indicated in mm/hr. This input is composed of a single CSV for each event. It is possible to have multiple events by passing multiple files.
8. **Water Level Event CSV:**  this is an input-setup-file(s) and it indicates a water level event which can be applied to the full domain or to a specific given area. This event is used to specify a specific water level (in meter) at specific time. This input is composed of a single CSV for each event. It is possible to have multiple events by passing multiple files.
9. **Inflow Event CSV:** this is an input-setup-file(s) and it indicates an inflow event which can be applied to the full domain or to a specific given area. The inflow can vary in time and it is indicated in m3/s. Attention, this differs from the rain event since the inflow value is calculated by interpolation. This input is composed of a single CSV for each event. It is possible to have multiple events by passing multiple files.
10. **Time Plot CSV:** this is an output-setup-file(s) and it indicates the points in the domain where the time varying values of a specific given physical variable (e.g. water depth, velocity, etc.) must be saved. Each time plot is identified by a single CSV file and it can save only one single physical variable over multiple points using a time period. The data output generated is a CSV file. It is possible to have multiple time plot outputs by passing multiple output-setup-files.
11. **Raster Grid CSV:**  this is an output-setup-file(s) and it indicates that all the values in the domain of a specific physical variable must be saved for every given time interval. It is also possible to save the maximum values and the final values at the end of the simulation. Each raster grid is identified by a single CSV file and it can save only one single variable. The data output generated is an ASCII-grid file. It is possible to have multiple raster grid outputs by passing multiple output-setup-files.
12. **Output Console:** true or false. If true, the progress of the simulation is displayed to the console (or to a specific console output file in the BATCH Mode) once every specified period (see pt. 28) in seconds. If false, nothing is displayed. The information displayed are: total number of iterations and the number of iterations from the last output, the simulation time, the minimum, maximum, average and last DT from the last output, the maximum velocity in the domain, highest point where water is still moving (if pt. 38 is true), the input and output volumes (if pt. 30 is true) and the run time in seconds from the beginning of the simulation (if pt. 29 is true).
13. **Output Period (s):** the output displayed on the screen is updated once every specified seconds from the start of the simulation (in this case 600s, i.e., once every ten minutes in the simulation time, no run-time). The number of outputs depends on the length of the simulation (Time End). For example a 2 hours simulation and an output period of 600 seconds produce twelve outputs to console.
14. **Output Computation Time:** true or false. If the parameter is set as “true” then the computation time is printed on the screen. In the opposite case not.
15. **Check Volumes:** true or false. Check that the mass conservation is respected or at least preserved within a certain error.
16. **Remove Proc Data (No Pre-Proc):** true or false. When this command is activated it removes all the temporary CADDIES-2D data files generated during the simulation. Note: the files generated during the preprocessing phaseare not involved by this command.
17. **Remove Pre-Proc Data:** this command complements pt.31. If the parameter is set on “true” then all files created during preprocessing are also deleted.
18. **Raster VEL Vector Field:** true or false. When this command is not activated (false) and there is a Raster Grid output-setup-file for the velocity, two output ASCII grid files are generated which show respectively the speed and the angle of the velocity for each cell. If this command is activated (true), only a single output CSV file is generated where each line contains the velocity information of one cell. This information are: the X and Y coordinates of the cell centre, the speed, the angle in radian, and the angle in degree. This command is used to plot the velocity of the cells as vectors in GIS software instead than a raster grid.
19. **Raster WD Tolerance (meter):** when caflood has run the simulation it takes the input of this instruction to set a threshold for wet and non-wet cells before saving the output. Referring to the example, all cells with less than 0.01 m of water depth will be treated as dry and therefore no water depth will be exported for those cells. All the raster output are influenced by this parameter, i.e. any eventual depth and velocity outputs. The user has to consider that the value of this parameter strongly depends on the vertical accuracy of digital elevation model, the type of terrain, the grid resolution and the quality of the precipitation and infiltration type of data.
20. **Update Peak Every DT:** true or false. When this command is activated, the maximum/peak values of a physical variable are saved at every time step. The default is false; the maximum values are saved only every update DT (usually 60 seconds). Empirical tests showed that this command, when activated, has an insignificant impact on the accuracy but a large impact on the run time of the simulation.
21. **Expand Domain:** true or false. When this command is not activated (false), the computational domain is the full domain, i.e. each cell with elevation data is processed. When this command is activated, the computational domain is set to be the same of the given events and it will expand when water reach the border of the computational domain. This command is useful to save run time when a simulation has a limited number of point sources of water like an inflow event from a small area.
22. **Ignore Upstream:** true or false. When this command is activated the model identifies the elevation height where the water is still, i.e. no outflow is generated during an update step. Once a new height is identified all cells with elevation above the height are not considered for computation any more. This can save some run-time depending on the terrain and on the type of events modeled. At the worst case scenario, it might add around 1% of extra run-time.
23. **Upstream Reduction (meter):** it is used by **Ignore Upstream** command to identify the amount of meter to reduce the possible elevation height at each update step.

The order of which the instruction lines appear in the setup CSV file is not significant, i.e. they can be of any order. If an instruction line has a default value, it can be omitted. If the model does not recognize an instruction line in the setup file, it will stop the computation and return an error. In the next paragraph, there is an example of output-setup-file for the Raster Grid CSV parameter called WDraster.csv:

**[WDraster.csv]**

1. Raster Grid Name , Simple Sloping Domain WD Raster Grid
2. Physical Variable , WD
3. Peak , true
4. Period (seconds) , 0

The details of each instruction line are described as following:

1. **Raster Grid Name:** is the full name used to identify the raster grid file.
2. **Physical Variable:** is the physical variable that is going to be saved in the ASCII raster grid. The options are: water depth (WD), water level (WL) and ‘angular’ velocity (VEL).
3. **Peak:** true of false. When this command is activated the simulation memorizes and produces a raster grid with the peak/maximum values of the physical variable chosen.
4. **Period:** a raster grid of the physical variable will be produced every period seconds from the start of the simulation. If, for instance, the user wants to have an output every 10 minutes, he/she have to use 600.

In the next paragraph, there is an example of output-setup-file for the Time Plot CSV parameter called WLpoints.csv:

**[WLpoints.csv]**

1. Time Plot Name , Simple Sloping Domain WL Points
2. Physical Variable , WL
3. Points Name , 1, 2
4. Points X Coo , 264682, 264538
5. Points Y Coo , 664581, 664665
6. Period (seconds) , 60

The details of each instruction line are described as following:

1. **Time Plot Name:** is the full name used to identify the time plot file.
2. **Physical Variable:** is the physical variable that is going to be saved in the data output CSV file. The options are: water depth (WD), water level (WL) and ‘angular’ velocity (VEL).
3. **Points Name:** a list of names which identifies the locations where the time varying values of the physical variable will be saved.
4. **Points X Coo:** the X coordinates of the points. The number of coordinates must be the same of the number of names in the ‘Points Name’ list.
5. **Points Y Coo:** the Y coordinates of the points. The number of coordinates must be the same of the number of names in the ‘Points Name’ list.
6. **Period:** the value of the various locations of the physical variable will be saved every period seconds from the start of the simulation. If, for instance, the user wants to have an output every 10 minutes, he/she have to use 600.

In the next paragraph, there is an example of output-setup-file for the Rain Event CSV parameter called Rain40mm60minl.csv:

**[Rain40mm60min.csv ]**

1. Event Name , Rain Intensity for An Hour
2. Rain Intensity (mm/hr) , 40, 0
3. Time Stop (seconds) , 3600, 7200
4. Zone (tlx tly w h) ,

The details of each instruction line are described as following:

1. **Event Name:** is the full name used to identify the rain fall event.
2. **Rain Intensity:** is a list of values that indicate the amount of rainfall to fall in mm/hr at specific times.
3. **Time Stop:** is a list of times in seconds when the corresponding values in ‘Rain intensity’ will be stop to be used in the simulation. This list must be the same length of the list ‘rain Intensity’.
4. **Zone:** is the area where the event will be applied. The area is identified by the top-left corner x and y coordinate and the width and height sizes. If this parameter is empty, the event will be applied in the entire domain.

An example of command line execution of the **caflood** program in widows and UNIX are respectively the following:

* caflood /sim C:\Indir Sim.csv C:\Outdir
* caflood -sim C:\Indir Sim.csv C:\Outdir

Other than the obligatory three arguments previously described (input folder, setup-file, output folder), the program can be launched with multiple optional arguments identified by the prefix ‘/’ in Windows OS and the prefix ‘-’ in UNIX OS. In this case, the ‘WCA2D’ argument is used to indicate that the model requested is the weighted cellular automata 2D model; without this argument no simulation will be executed. Using the argument ‘help’ is possible to have a list of optional arguments.

# References

Dagum, Leonardo, and Ramesh Menon. 1998. ‘OpenMP: An Industry Standard API for Shared-Memory Programming’. *Computational Science & Engineering, IEEE* 5 (1): 46–55.

Ghimire, Bidur, Albert S. Chen, Michele Guidolin, Edward C. Keedwell, Slobodan Djordjević, and Dragan A. Savić. 2013. ‘Formulation of a Fast 2D Urban Pluvial Flood Model Using a Cellular Automata Approach’. *Journal of Hydroinformatics* 15 (3): 676. doi:10.2166/hydro.2012.245.

Guidolin, Michele, Albert S. Chen, Bidur Ghimire, Ed Keedwell, Slobodan Djordjevic, and Dragan A. Savic. 2015‘A Weighted Cellular Automata 2D Inundation Model for Rapid Flood Analysis’. *Under Review*.

Guidolin, Michele, Andrew Duncan, Bidur Ghimire, Mike Gibson, Edward Keedwell, Albert S. Chen, Slobodan Djordjevic, and Dragan Savic. 2012. ‘CADDIES: A New Framework for Rapid Development of Parallel Cellular Automata Algorithms for Flood Simulation’. In Hamburg, Germany: IWA (International Water Association).

Itami, Robert M. 1994. ‘Simulating Spatial Dynamics: Cellular Automata Theory’. *Landscape and Urban Planning* 30 (1): 27–47.

Munshi, A., and others. 2011. ‘The OpenCL Specification Version 1.1’. *Khronos OpenCL Working Group*.

Wolfram, Stephen. 1984. ‘Cellular Automata as Models of Complexity’. *Nature* 311 (October): 419–24.