Hyper-resolution 1D-2D urban flood modelling using LiDAR data and hybrid parallelization

Seong Jin Noh[1], Junhak Lee[2], Seungsoo Lee[3*], Kenji Kawaike[4], Dong-Jun Seo[1]

[1]Department of Civil Engineering, University of Texas at Arlington, Texas, 76019, USA.

[2]Department of Earth & Environmental Sciences, University of Texas at Arlington, Texas, 76019, USA.

[3]Climate Change Research Team, APEC Climate Center, Centum 7-ro, Haeundae-gu, Busan 48058, Republic of Korea.

[4]Disaster Prevention Research Institute, Kyoto University, Yoko-oji, Fushimi, Kyoto 612-8235, Japan.

Corresponding author: Dr. Seungsoo Lee (seungsoo_lee@apcc21.org)

To be submitted to

Environmental Modelling and Software

Highlights

- A hybrid parallel code, Hybrid-1D-2D, or H12, is developed for coupled 1D-2D urban flood modelling.
- H12 enables street-resolving simulation based on LiDAR-based topography.
- Variable grid sizing is used for efficient and accurate simulation in areas of interest.
- Advantages of hybrid parallelization over OpenMP or MPI alone are analyzed.

Abstract

Coupled 1D-2D modelling is a widely used approach to predict water movement in complicated surface and subsurface drainage systems in urban or peri-urban areas. In this study, a hybrid parallel code, H12, is developed for 1D-2D coupled urban flood modelling. Hybrid-1D-2D, or H12, enables street-resolving hyper-resolution simulation over a large area by combining Open Multi-Processing (OpenMP) and Message Passing Interface (MPI) parallelization. Variable grid sizing is adopted for detailed geometric representation of urban surfaces as well as efficient computation. To assess the capability of H12, simulation experiments were carried for the Johnson Creek Catchment (~40 km^2) in Arlington, Texas. The LiDAR-derived detailed land surface at 1-m resolution is used to represent the terrain and urban features in flood modelling. Hybrid parallelization achieves up to a 79-fold reduction in simulation time compared to the serial run and is more efficient than either OpenMP or MPI alone especially in hyper-resolution simulations.

Keywords: Coupled 1D-2D modelling; Hybrid parallelization; Hyper-resolution; LiDAR

1. Introduction

Floods is one of the most destructive natural hazards (Michel-Kerjan and Kunreuther, 2011). With urbanization and climate change, the frequency and magnitude of floods are changing in many parts of the world (Hirabayashi et al., 2013; Karen R. Ryberg et al., 2014; Mallakpour and Villarini, 2015). In particular, urban flooding is becoming increasingly costly and difficult to manage due to a greater concentration of population and assets in urban centers. Urban flooding is a phenomenon affected by various factors such as rainfall, topography, and hydrologic and hydraulic processes on land surface and in subsurface. Dual drainage, i.e., the concurrent water flow not only in sewer pipes but also on land surface at the same location, is a unique and important aspect of urban hydrology and has been studied by many researchers (Bazin et al., 2014; Djordjević et al., 1999, 2014; Fraga et al., 2015; Leandro et al., 2009; Lee et al., 2015; Teng et al., 2017). Among the numerous modelling approaches, coupled 1D-2D modelling is one of the most widely used to simulate dual drainage through one-dimensional sewer flow, two-dimensional surface flow and the exchange between the 1D and 2D domains in which the above flows occur (Adeogun et al., 2015; Chen et al., 2015; Djordjević et al., 2014; Fraga et al., 2015; Leandro et al., 2009; Noh et al., 2016a).

Since surface flow is strongly influenced by topography, availability of accurate digital elevation models (DEM) at appropriate resolutions is central to accurately simulating flooding (Abily et al., 2016; Bierkens et al., 2015; Bates et al., 2003; Leitão et al., 2009). Mark et al. (2004) also pointed out that the use of high-resolution geomorphological data is critical to accounting for the effects of man-made structures such as buildings, roads and curbs on the urban surface. Recently, remote sensing technology has revolutionized high-resolution water modelling (Fewtrell et al., 2008; Liu et al., 2015). In particular, light detection and ranging (LiDAR) systems have been widely used for flood inundation modelling which can capture floodplain topography at 1-m or finer resolution with high horizontal and vertical accuracy (Gallegos et al., 2009; Marks and Bates, 2000; Mason et al., 2007, 2014; Neal et al., 2009a; Ozdemir et al., 2013; Schubert et al., 2008). However, as discussed by Dottori et al. (2013), overconfidence can be placed in such high-resolution data when in reality accuracy is not necessarily improved by higher resolution due to uncertainty sources. In addition, since the laser beam measures the distance to the first object on its path, the DEMs from LiDAR may be different from the actual surface boundary for water movement. For instance, if vegetation canopy and man-made structures such as bridge or elevated roads are not properly treated in the raw LiDAR data, the resulting flood simulation is not likely to be realistic. Although there remain various issues on the tradeoff among model resolution, model complexity, data and parameter uncertainty, and computational feasibility (Abily et al., 2016; Bates et al., 2003; Fewtrell et al., 2008; Mignot et al., 2006; Neal et al., 2012), hyper- or high-resolution data sets are radically changing computer models and their use, increasing their complexity and range of applications (Beven, 2007; Dottori et al., 2013). Although there is no formal distinction between hyper- and high-resolutions, we use the term "hyper-resolution" for grid sizes of 1-m or finer and "high-resolution" for grid sizes which are coarser than 1 m. In addition, within the manuscript, we use the terms "fine" and "coarse" to differentiate grid sizes in relative terms within high- and hyper-resolutions.

Hyper- or high-resolution 2D inundation modelling is, however, computationally expensive for many real world applications (Neal et al., 2010). For example, 1-m resolution modelling of a 40 km^2 area (40×10^6 grids) requires more computational grids than 500-m resolution modelling of the contiguous United States (CONUS) (32×10^6 grids). In addition to the dimensionality of the computational grid, a small time step of 0.05 s or less is required for convergence (e.g., Courant-Friedrichs-Lewy (CFL) condition) for dynamic or diffusive wave modelling which renders hyper-resolution modelling more challenging for practical implementation. As such, some form of parallelization is necessary for hyper-resolution flood modelling. Parallel computing methods include Message Passing Interface (MPI) in a distributed memory system(Gropp et al., 1996), Open Multi-Processing (OpenMP) in a shared memory (Chapman et al., 2007) and co-processor parallelism which utilizes graphics processing units (GPU) (Ament et al., 2011) or many-core processors such as Intel Xeon Phi (Intel, 2017).

There have been significant advances in parallel 2D inundation modelling (Gallegos et al., 2009; Neal et al., 2010, 2009b; Sampson et al., 2015; Sanders et al., 2010; Shen et al., 2015). Sanders et al. (2010) developed a parallel Godunov-type shallow water code, ParBrezo, using MPI and evaluated it for urban dam break flood and storm surge inundation cases. Neal et al. (2010) compared three parallelization methods based on OpenMP, MPI and accelerator cards, and found that MPI is slightly more efficient than OpenMP and shows high scalability with a large number of cores. Leandro et al. (2014a) developed a 2D parallel diffusive wave model for flood plain inundation with the Matlab parallel computing toolbox and Fortran OpenMP. Sampson et al. (2015) used OpenMP parallelized LISFLOOD-FP and reach decomposition to simulate inundation over the entire globe at 90-m resolution.

Each parallelization method has positives and negatives. Though straightforward to implement, OpenMP is limited in scalability by the size of the shared memory system. MPI is scalable for high-dimensional problems but at the expense of reduced computational efficiency due to increasing communications among the CPUs. Despite large potential, co-processor parallelism is highly device-dependent and improves performance only for specific types of calculations. In recent years, a number of studies have demonstrated that hybrid parallelization that combines the above may be an effective solution for drastically reducing computational time for high-dimensional problems in various research areas (Borges et al., 2014; Gorobets et al., 2013; Ibanez et al., 2016; Mininni et al., 2011; Satarić et al., 2016; Wan and Lin, 2013). However, to the best of the authors' knowledge, there has been no investigation into hybrid parallelization for 1D-2D flood modelling presumably due to the convoluted nature of dual drainage which requires extensive coupling.

In this study, a hybrid parallel code, Hybrid-1D-2D, or H12 for short, is developed for coupled 1D-2D urban flood modelling to enable street-resolving hyper-resolution simulation for a large area by combining OpenMP and MPI. The code developed uses variable grid sizing for detailed geometric representation of urban land surfaces and computational efficiency. In order to assess the capability of hyper-resolution 1D-2D modelling in a realistic setting, H12 is applied for the Johnson Creek Catchment (~40 km^2) in Arlington, Texas. We used LiDAR-derived DEM with 1-m resolution for topography. The LiDAR-derived DEM is post-processed to remove vegetation canopy and to capture flow paths below urban objects such as elevated roads, overpasses and bridges. The value of hyper-resolution 1D-2D modelling is discussed in comparison with low-resolution modelling results. In addition, performance of hybrid parallelization over OpenMP and MPI is analyzed.

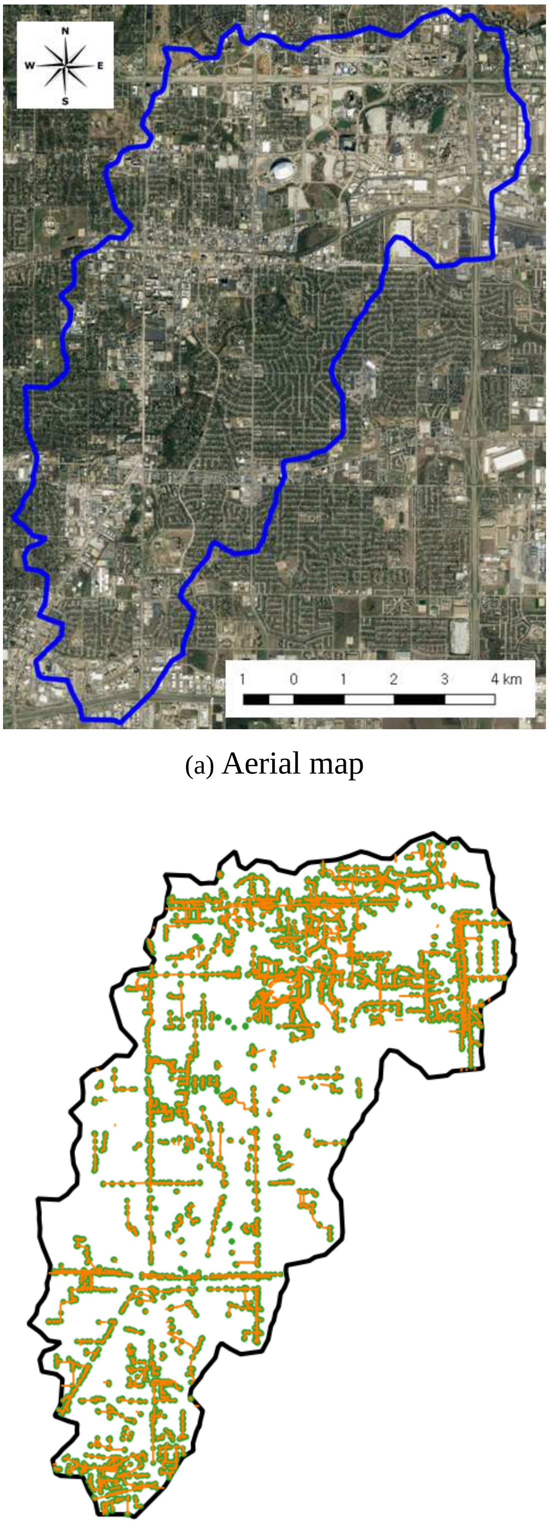
The rest of this paper is organized as follows: Section 2 describes the study area and data used. Section 3 describes the coupled 1D-2D urban flood model and development of the parallel code, H12, including the variable sized scheme, profiling of the serial code and structure of the hybrid parallel code. Section 4 presents simulation results at two different resolutions, compares run-times and efficiency with different parallelization methods, and discusses challenges of hyper-resolution urban flood modelling and parallel computing. Section 6 summarizes the main findings.

2. Study area and data used

In this section, we describe hydrologic and hydraulic characteristics of the study domain and the procedure of LiDAR data processing for hyper-resolution 1D-2D urban flood modelling.

2.1 Study area

The study domain is the Johnson Creek catchment (40.2 km^2) located in Arlington, Texas (Fig. 1). The Johnson Creek (length: 21 km) originates near Interstate 20 in eastern Tarrant County, TX, and flows northeasterly to drain into the Trinity River in Grand Prairie in Dallas County. The major land cover types include residential area, road, parking lot and lawn. There are 7,017 sewer pipes connected with 3,615 inlets, 3,259 storm fittings, 549 manholes, 286 culverts and 415 outfalls in the model domain. The total lengths of sewer pipes and culverts are 190.7 and 5.8 km, respectively. According to the previous study (Asquith and Roussel, 2004), 500-year return period rainfall for 1 and 2-hour durations is about 117 and 71 mm/hr, respectively, which was used for code evaluation and benchmarking.



(a) Aerial map

(b) Pipes/culverts (orange) and manholes/inlets (green)

Fig. 1. Study area (the Johnson Creek catchment, Arlington, Texas).

2.2 Hyper-resolution LiDAR-derived land surface

Hyper-resolution DEM was obtained from LiDAR point data. The LiDAR data used in this study was created in April 2009 with the Leica ALS-50 system. The system was flown at an average flying altitude of 1,400 m above ground with an average point spacing of 0.57 m. The resulting LiDAR data have a vertical accuracy of 0.07 m and a horizontal accuracy of 1 m at 95% confidence level. Given the density of the raw data, 1 m was considered the finest resolution retrievable for the study area. Initially, a 1-m bare-earth DEM was created from the ground-return LiDAR points using a natural neighbor interpolation method (Sibson, 1981) with ArcGIS 10.4 (www.esri.com). Then, solid urban features such as houses and buildings which obstruct flow of storm water were added to the bare-earth DEM for numerical flood modelling. We obtained the building footprint GIS layer from the City of Arlington and identified positions of buildings and houses in the DEM by increasing the elevation of the pixels inside of the building foot-print polygons. Due to the fact that certain urban features such as elevated roads and bridges could not be removed correctly by using the provided LiDAR classification codes, we adjusted the DEM manually to remove the erroneous elevated roads and bridges over water bodies which unrealistically obstruct the flow of water in the ground surface model (Abdullah et al., 2012; Meesuk et al., 2015). The 1-m resolution DEM generated through the above processes is shown in Fig. 2.

Mixed effects of natural and artificial urban components are found in the hyper-resolution DEM. Within the 40.2 km^2 area, the maximum difference of elevation within the DEM is about 70 m. Fig. 3 enlarges the DEM and land cover maps in the red circle in Fig. 2. It is also found that the hyper-resolution DEM represents variations of elevation at road and building scale. In this work, infiltration and evapotranspiration were not modeled. For runoff simulation, we used the city-provided GIS data to create the land cover type map. The six land cover types used in modelling include road, building, parking lot, vegetation, bare ground, and water body. Different runoff coefficients were assigned to different land cover types as shown in Table 1. Calibration of runoff coefficients was beyond the scope of this study.

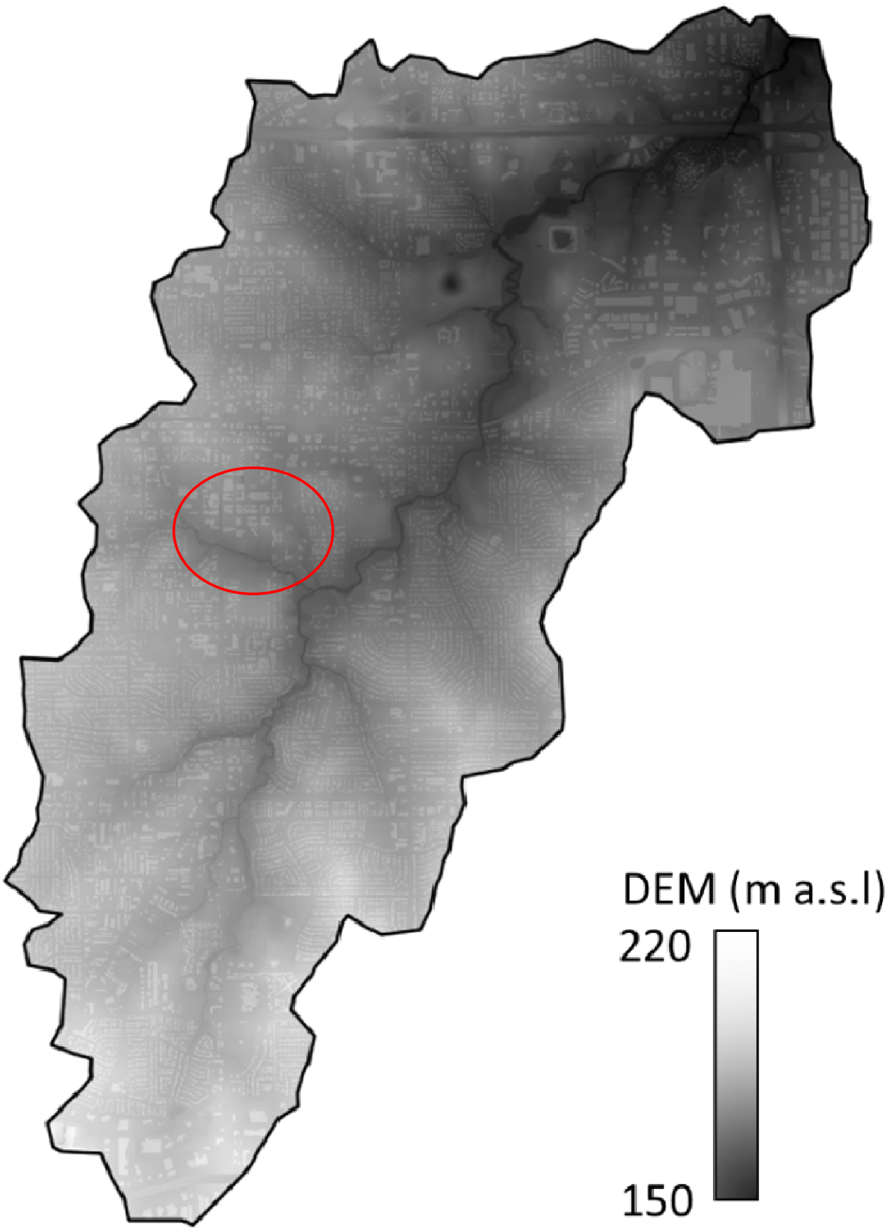


Fig. 2 1-m resolution DEM retrieved from LiDAR.

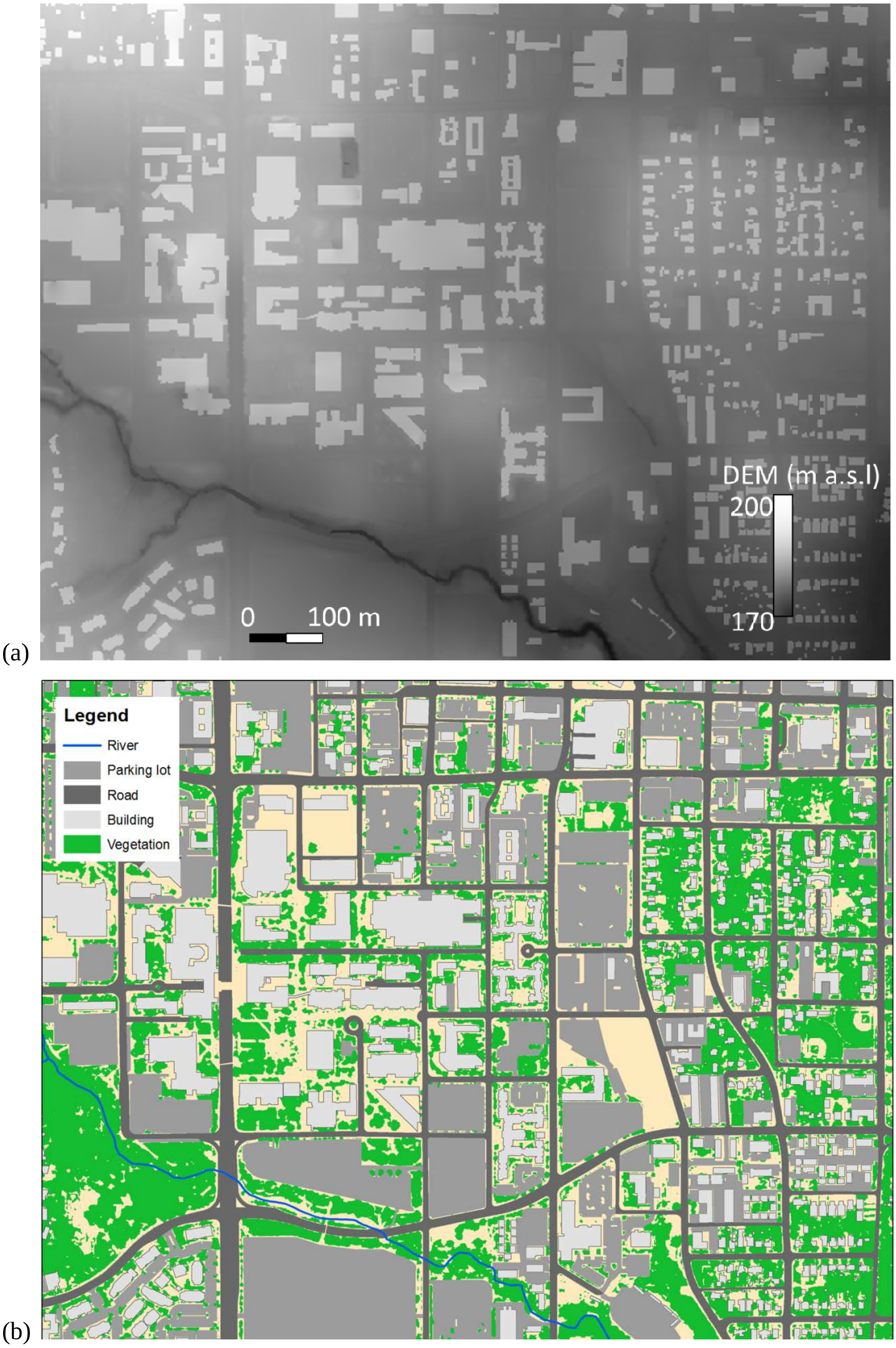


Fig. 3 Enlarged views of hyper-resolution input data. (a) DEM, (b) Land cover.

Table 1. The land cover types and runoff coefficients.

Land cover	Road	Building	Parking lot	Vegetation	Bare ground	Water body
Runoff coeff.	0.9	0.9	0.9	0.5	0.6	1.0

3. Numerical model

This section describes the coupled urban flood model used and development of the hybrid parallel code. The coupled 1D-2D urban flood model consists of a horizontal 2D inundation flow model (Kawaike et al., 2011), a 1D Preissmann slot model (Chaudhry, 1979) and an interaction model between the 1D and 2D domains. This model has been used in a number of previous studies to improve understanding of urban flooding (Lee et al., 2012; Lee, 2013; Lee et al., 2015; Noh et al., 2016a). Although the overall model structure is similar to other hydrodynamical models, one of the unique features of the proposed model is to incorporate physically-based representation of the exchange of flow not only between the land surface and sewer pipes but also between the sewer pipes and manholes (Noh et al., 2016a). In order to develop an efficient parallel code, the serial code is profiled in a hyper-resolution case in terms of computation run-times. Then, the hybrid parallel code which combines MPI and OpenMP, is described.

3.1 Coupled 1D-2D urban flood model

Two dimensional (2D) shallow water equations which consist of continuity (Eq. (1)) and momentum (Eqs. (2) and (3)) equations are used as governing equations for the 2D surface flow model as follows:

$$\frac{\partial h}{\partial t}+\frac{\partial M}{\partial x}+\frac{\partial N}{\partial y}=\alpha\cdot r_e-q_{ex\_s} \tag{1}$$

$$\frac{\partial M}{\partial t}+\frac{\partial (uM)}{\partial x}+\frac{\partial (vM)}{\partial y}=-gh\frac{\partial H}{\partial x}-\frac{gn^2M\sqrt{u^2+v^2}}{h^{4/3}}, \tag{2}$$

$$\frac{\partial N}{\partial t}+\frac{\partial (uN)}{\partial x}+\frac{\partial (vN)}{\partial y}=-gh\frac{\partial H}{\partial y}-\frac{gn^2N\sqrt{u^2+v^2}}{h^{4/3}}, \tag{3}$$

where h is the water depth, H is the water level, u and v are the x- and y-directional velocities, respectively, M (= uh) and N (= vh) are the x- and y-directional flow fluxes, respectively, q_{ex_s} (=Q_{ex}/A_{sur}) is the interacting discharge between the surface and the manhole (m/s), the interacting discharge, Q_{ex}, is calculated from the water depths in surface and manhole grids using weir and orifice equations (Noh et al., 2016a), A_{sur} is the area of surface grid, g is the gravitational acceleration, r_e is the rainfall intensity, α is the runoff coefficient for each land use, and n is Manning's roughness coefficient. In this study, we used n = 0.012.

The Preissmann slot model (Chaudhry, 1979) is used to simulate sewer pipe flow. This model assumes a hypothetically narrow slot on the pipe crown. The governing equations are as follows:

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=q_{man}\,, \tag{4}$$

$$\frac{\partial Q}{\partial t}+\frac{\partial (uQ)}{\partial x}=-gA\frac{\partial H_p}{\partial x}-gn^2\frac{Q|Q|}{R^{4/3}A}, \tag{5}$$

where A is the wetted area of the cross section, Q is the flow discharge, u is the velocity, R is the hydraulic radius, and n is the roughness coefficient. H_p is the piezometric head ($H_p = Z_p + h_p$), Z_p is the bottom elevation of the sewer pipe, q_{man} (Q_{man}/dx_{sw}) is the flow discharge from the manhole to the sewer pipe per unit pipe length, Q_{man} is interactive flow discharge from the manhole to the sewer pipe and h_p is the water depth. The water depth in the sewer pipe can be calculated from wetted area of pipe based on hydraulic characteristic curve (Lee et al., 2016). It is also worth noting that, in H12, a sewer pipe longer than a threshold grid length (10 m in this simulation) is divided into multiple subsections for numerical simulation (e.g. a 150-m long sewer pipe is divided into 15 subsections each with a length of 10 m). The lateral inflow/outflow between manhole and sewer pipes, q_{man} (m^2/s), in Eq. (4) occurs only at the subsections which are connected to a manhole and for other subsections in pipes, q_{man}, is zero. The water depth in a manhole, h_{man}, is updated based on the discharge interactions between the surface and the manhole, q_{ex_m}, as well as between the manhole and the sewer pipe (Q_{man}). The governing equation is as follows:

$$\frac{\partial h_{man}}{\partial t}=q_{ex\_m}-\frac{\sum_{i=1}^{N}Q_{man_i}}{A_{man}} \tag{6}$$

where q_{ex_m}(=Q_{ex}/A_{man}) is the interacting discharge between surface and manhole (m/s; positive value means water flow from surface to manhole and negative value means water flow from manhole to surface), A_{man} is the bottom area of manhole, h_{man} is the water depth in the manhole, N is the number of sewer pipes which are connected to the manhole.

The 2D surface flow model and 1D sewer network model are coupled by the interaction model to exchange flow under various hydraulic conditions. Modified orifice and weir equations are used to calculate flow between surface and manhole as well as between manhole and sewer pipe. In this study, flow exchange between the 2D surface and 1D sewer domains is assumed to occur at all inlet or manhole grids, while inlet is conceptualized as a type of manhole. The details of the interaction model can be found in (Noh et al., 2016a).

For mesh generation, we used variable grid sizing adopted and extended from the non-uniform but structured quadrilateral grid scheme (Liang, 2011). This technique is known to reduce computation time while maintaining accuracy where the topography is relatively steep and the flow pattern is complicated (Kesserwani and Liang, 2012). An example mesh from variable grid sizing is shown in Fig. 4. The figure assumes that a 4-way intersection is located in the middle of the hypothetical domain and that more detailed simulation is required along the road. With variable grid sizing, the grids along or in the vicinity of the road is selectively defined at hyper resolution while coarser resolution grids are used for other types of land cover. For numerical computation on coarse grids that abut fine grids, the values on the find grids are aggregated onto the coarse grids by averaging them, which results in a virtual ‘parent’ grid of the fine grids. For fine grids that abut coarse grids, the coarse grid is divided into multiple virtual ‘child’ grids that share the same value with the coarse grid. Although the original scheme allows multiple grid sizes that may vary in time according to computational conditions, only two grid sizes were used in this work for two different land uses (road: fine grid, other land covers: coarse grid) without temporal variation. A major benefit of the variable grid sizing scheme is reduced computation cost. As shown in Table 2, the number of the surface grids with the scheme using 1- and 8-m resolutions is 13,998,516 and 284,768, respectively, which means 35% and 45% reduction compared to that of uniformly-sized grids. The reduced dimension contributes to a reduction not only in run time but also in memory.

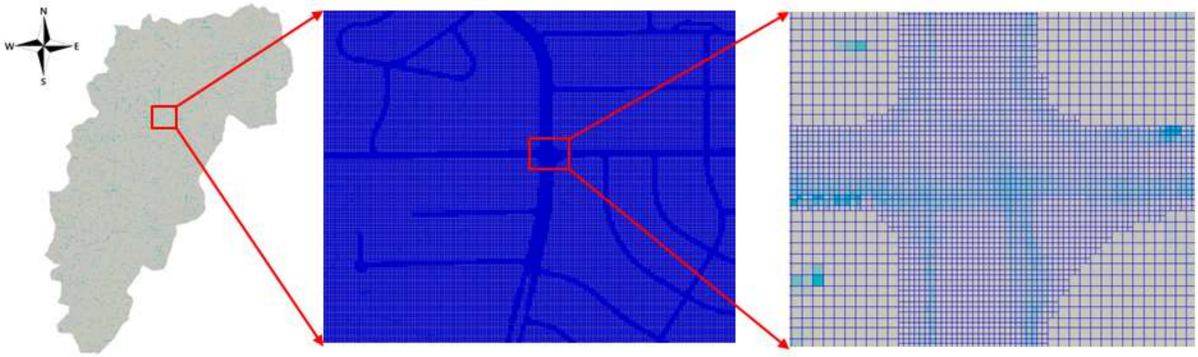


Fig. 4 An example of variable grid sizing.

Table 2. The number of grids/elements with variable grid sizing in different resolutions.

Resolution	Surface domain	Sewer domain		
		Pipe	Manhole	Inlet
1 m	13,998,516	7,017	549	3,615
2 m	3,667,558			
4 m	995,839			
8 m	284,768			

Fig. 5 shows the flowchart of coupled 1D-2D modelling. The module RDAT reads input data for topography, land use, sewer pipes, and manholes. A time loop is then iterated at every time step (dt) until the simulation is complete. Modules that deal with gridded rainfall and artificial pumping at multiple locations are available but not explicitly described in the flowchart for simplicity. At each time loop, the momentum and continuity equations for the 2D and 1D domains are solved alternately and then the exchange of flow between two domains is calculated by the module CONNECT-MH. In the right part of Fig. 5, the run time in sequential processing is profiled to develop a hybrid parallelization strategy. The relative run time, i.e., the run time of each module normalized by the total run time, was estimated in a serial 1 m-resolution run. The relative run time can vary according to different spatiotemporal forcings and configurations of the 1D-2D modelling domain. In Fig 5, the run time for input and output (I/O) operations was not included in order to examine the time requirement for each process-related module only. It was found that the solver for the momentum equations for the 2D domain (FLUX module) accounted for most run time (47.4 %), followed by the module that calculates values of each flux and water depth variables in Eqs. (1)-(3) for the next time step (MDVEL, 30.9%) and the module that solves the continuity equations for the 2D domain (SUISIN, 12.4%). Not surprisingly, the momentum and continuity equation solvers for 1D sewer flow (FLUX-SW, SUISIN-SW) accounted for less than 0.1 % of the run time. Due to the differences in the computational cost between surface and sewer domains, multiple approaches are required to parallelize the code and integrate two domains, which will be discussed in the next section.

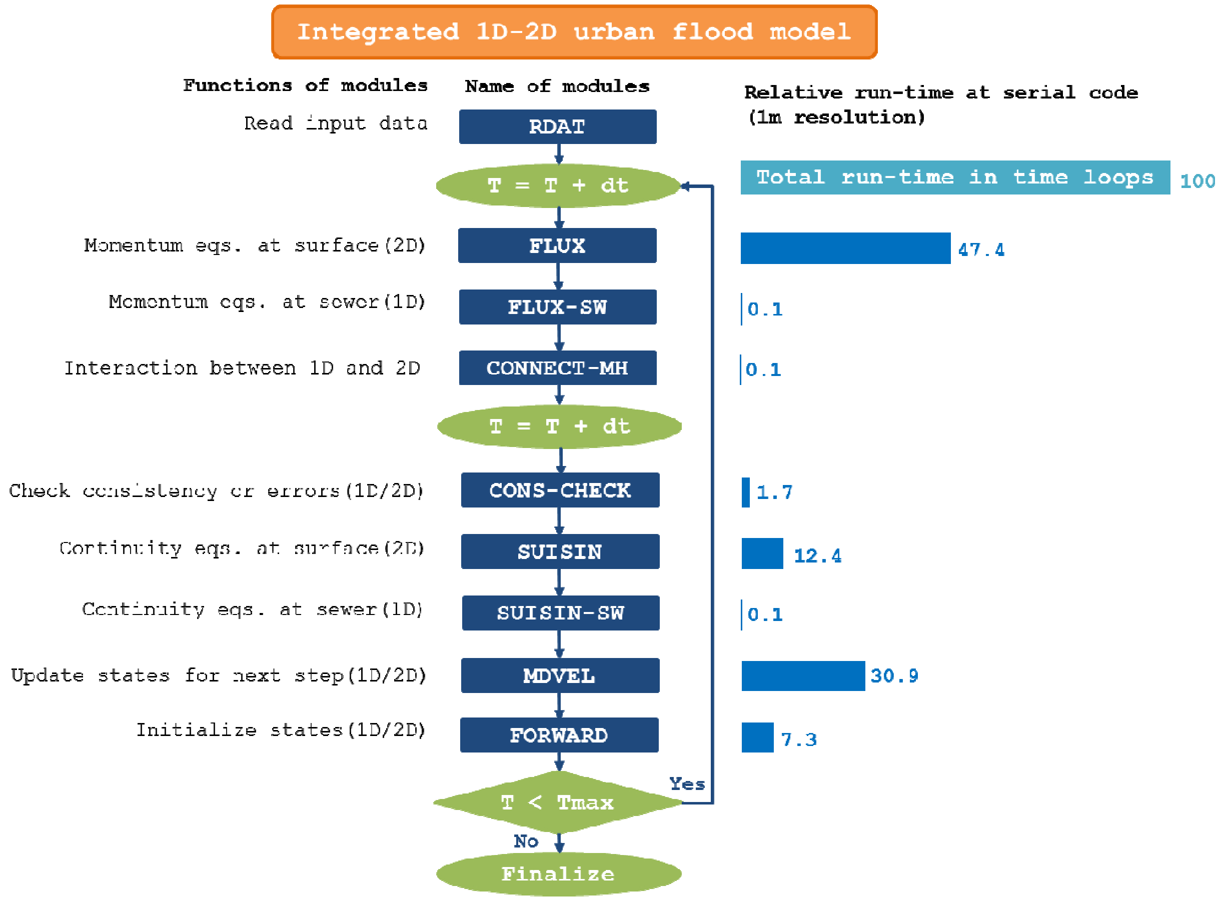

Integrated 1D-2D urban flood model
Functions of modules
Name of modules
Relative run-time at serial code (1m resolution)
Read input data
RDAT
T = T + dt
Total run-time in time loops
100
Momentum eqs. at surface(2D)
FLUX
47.4
Momentum eqs. at sewer(1D)
FLUX-SW
0.1
Interaction between 1D and 2D
CONNECT-MH
0.1
T = T + dt
Check consistency or errors(1D/2D)
CONS-CHECK
1.7
Continuity eqs. at surface(2D)
SUISIN
12.4
Continuity eqs. at sewer(1D)
SUISIN-SW
0.1
Update states for next step(1D/2D)
MDVEL
30.9
Initialize states(1D/2D)
FORWARD
7.3
Yes
T < Tmax
No
Finalize

Fig. 5. Flowchart of the coupled 1D-2D model and relative run-times per each module (1-m resolution).

3.2 Hybrid parallel code: Hybrid-1D-2D (H12)

The hybrid parallel code, H12, was developed for the coupled 1D-2D urban flood model described above by combining the strengths of two parallelization methods of OpenMP and MPI. OpenMP is an application programming interface (API) that supports multi-platform shared memory multiprocessing. Since OpenMP is implemented through compiler directives that instruct segments of code to be run as parallel threads without major changes in serial code, it is one of the least labor-intensive parallelism methods (Neal et al., 2010). Despite the computational overhead to create and merge multiple threads in the beginning and end of each OpenMP statement, communication among threads is not necessary since system memory is shared. The scalability of OpenMP, however, is explicitly limited by the number of cores in the shared system. For instance, in this study the maximum number of cores available for OpenMP was sixteen because each node includes two 8-core processors. MPI is a standardized and portable message-passing specification to run a parallel program across distributed memory systems and therefore is scalable for large-scale parallel applications if there are appropriate computing resources available such as supercomputers. The hybrid approach developed in this work maximizes computational efficiency through leveraging shared memory-based parallelism (OpenMP) in each MPI task in the distributed memory system.

Fig. 6 illustrates the workflow of H12. A hyper-resolution surface domain is decomposed into smaller multiple domains each of which is solved by an MPI task. The domain was partitioned unidirectionally to simplify the parallelization procedure. For balanced and efficient parallel computation, the same number of grids is assigned to each MPI task by dividing the total grids by the number of MPI tasks. The surface grids in the 2-D domain are re-ordered in a 1-D computation space according to the geometry of the model domain (e.g. from upper left to lower right), and the domain decomposition in the surface grids by MPI is also implemented in a 1-D space. Therefore, the irregular boundary of the catchment in the 2-D domain does not affect the efficiency of parallel computation by MPI. For each decomposed 2D domain, OpenMP directives were implemented for additional parallelization using multiple OpenMP threads in a shared memory system (i.e., a node in a PC or supercomputer). For forward calculation of shallow water equations in a decomposed domain, updated information (i.e., water depth) of connected grids which are being calculated at other MPI tasks, is necessary. In other words, each decomposed domain has their own new boundary conditions as whole domain was divided into several segments. Ghost and effective grids overlap each other between two adjacent decomposed domains. Non-blocking communication was used to exchange state variables between ghost and effective grids. For structured uniform grid, a complete solution can be obtained by exchanging three columns in the new virtual boundaries between the two MPI tasks. For unstructured variable grid, however, it is necessary to exchange additional grids during the MPI communications. In this study, 3% of the grids are set to be exchanged based on sensitivity and integrity analyses. However, the number of exchanging grids may be further reduced through investigating the exact range of grids affected by virtual boundaries, and also vary with respect to catchment geometry. It is also worth noting that the gains from variable grid sizing far outweigh the additional cost of increasing the number of grids for communication with virtual boundaries in terms of runtime. In addition, since computational cost of the sewer domain is far less than that of the surface domain (see Fig. 5), OpenMP parallelization was implemented for the sewer domain without domain decomposition. One potential bottleneck in parallelization in coupled 1D-2D modelling is that the state variables for surface and sewer domains need synchronization at every time step. To reduce the computational overhead for coupling the two domains, communication among MPI tasks was minimized through synchronizing the states at the inlet/manhole grids only, which account for less than 0.001 % of the total surface grids at 1-m resolution.

Fig. 7 shows a simplified example of hybrid parallelization applied to the momentum equation solver for the 2D domain (FLUX module). The red lines are associated with part of the MPI parallelization for each time step for a decomposed domain and communication between the ghost and effective grids. The blue lines are associated with part of the OpenMP directives for a loop for a decomposed domain. Except for the lines for parallelization, the other part of the code is the same as the serial code. It was verified that the simulation results by H12 are entirely consistent with those by the serial code in all simulation cases.

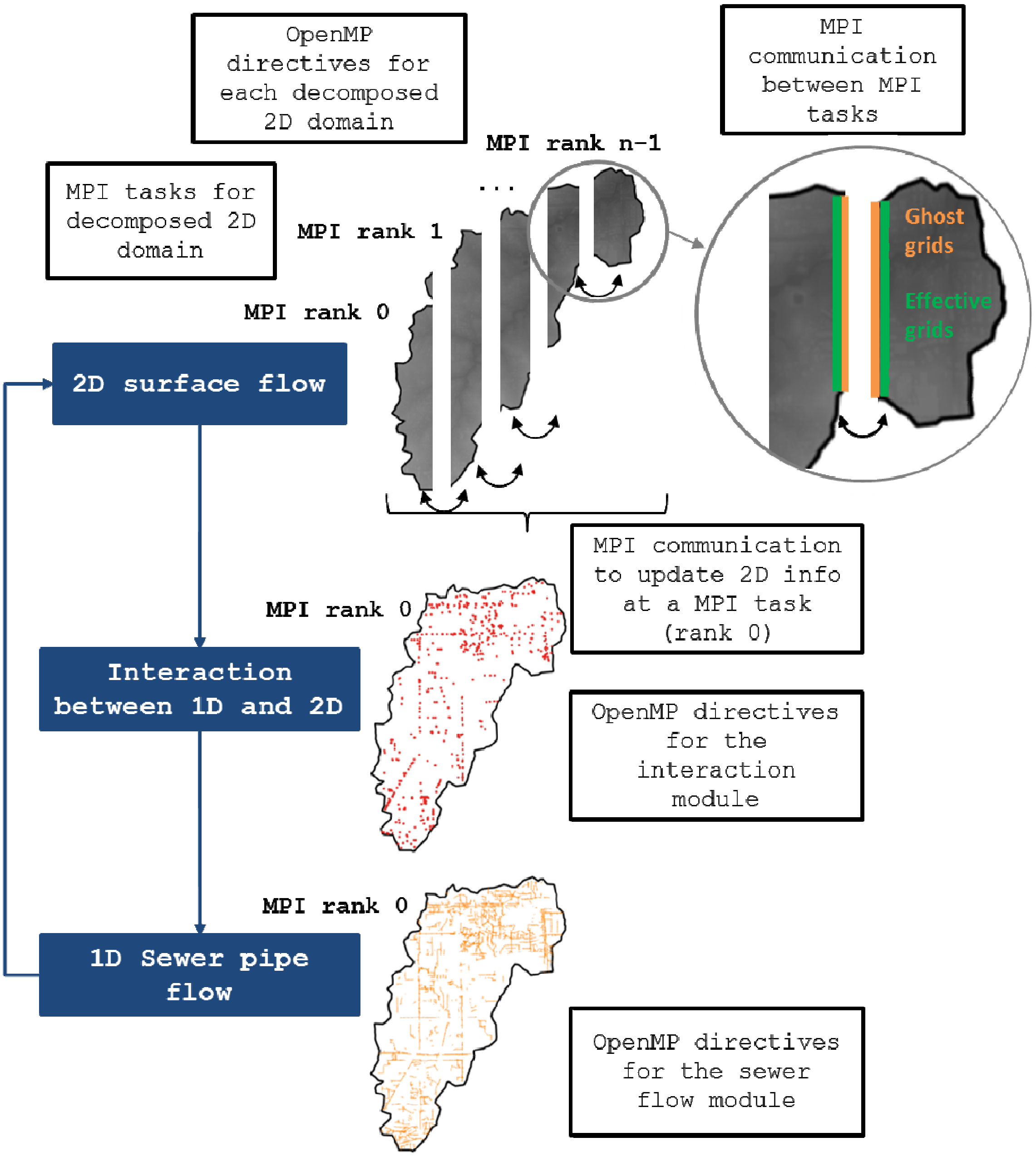
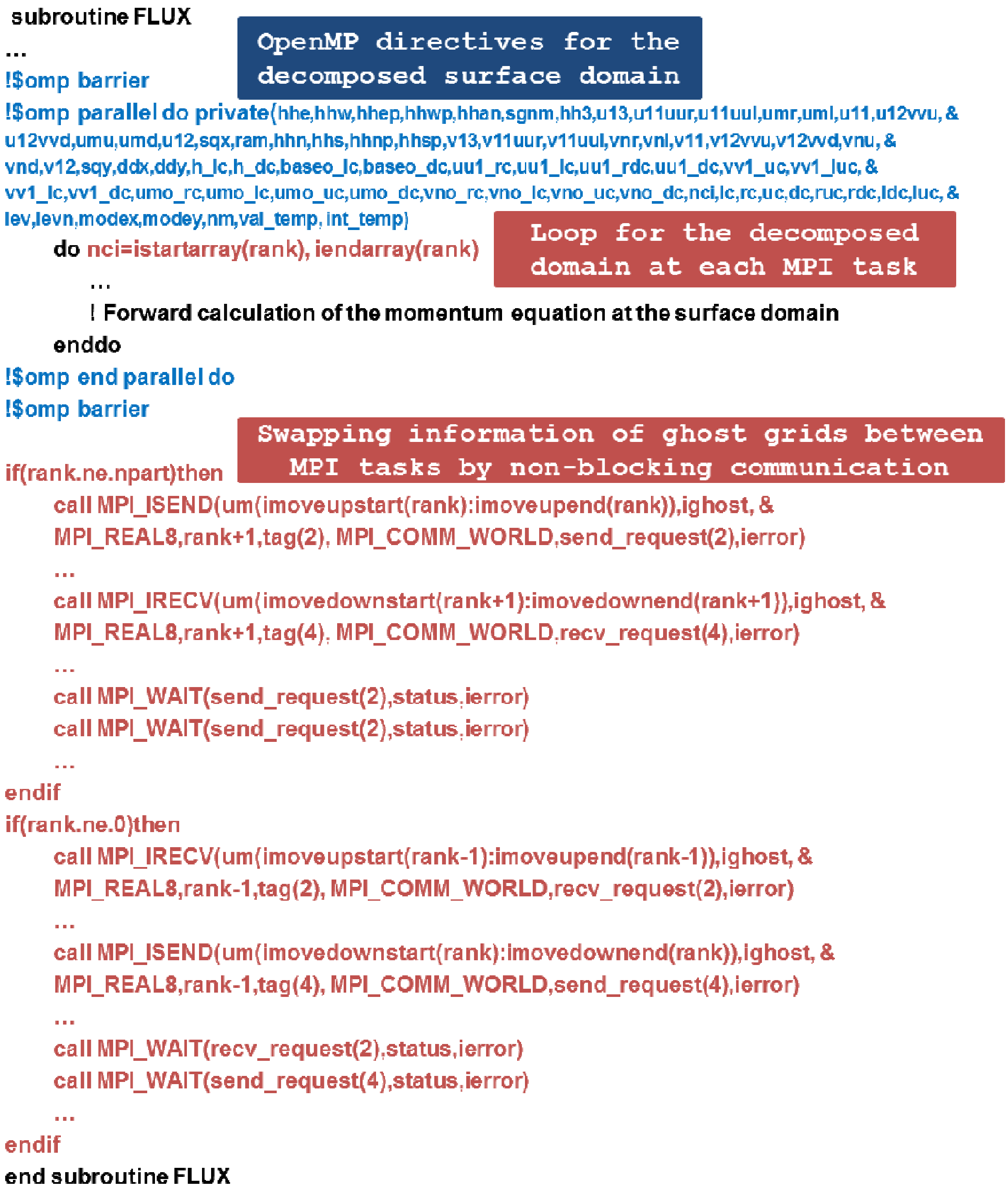

OpenMP directives for each decomposed 2D domain
MPI communication between MPI tasks
MPI rank n-1
MPI tasks for decomposed 2D domain
...
MPI rank 1
MPI rank 0
Ghost grids
Effective grids
2D surface flow
MPI communication to update 2D info at a MPI task (rank 0)
MPI rank 0
Interaction between 1D and 2D
OpenMP directives for the interaction module
MPI rank 0
1D Sewer pipe flow
OpenMP directives for the sewer flow module

Fig. 6. The parallel workflow of H12 for coupled 1D-2D modelling.



OpenMP directives for the decomposed surface domain

Loop for the decomposed domain at each MPI task

Swapping information of ghost grids between MPI tasks by non-blocking communication

```
 subroutine FLUX
...
!$omp barrier
!$omp parallel do private(hhe,hhw,hhep,hhwp,hhan,sgnm,hh3,u13,u11uur,u11uul,umr,uml,u11,u12vvu, &
u12vvd,umu,umd,u12,sqx,ram,hhn,hhs,hhnp,hhsp,v13,v11uur,v11uul,vnr,vnl,v11,v12vvu,v12vvd,vnu, &
vnd,v12,sqy,ddx,ddy,h_lc,h_dc,baseo_lc,baseo_dc,uu1_rc,uu1_lc,uu1_rdc,uu1_dc,vv1_uc,vv1_luc, &
vv1_lc,vv1_dc,umo_rc,umo_lc,umo_uc,umo_dc,vno_rc,vno_lc,vno_uc,vno_dc,nci,lc,rc,uc,dc,ruc,rdc,ldc,luc, &
lev,levn,modex,modey,nm,val_temp, int_temp)
      do nci=istartarray(rank), iendarray(rank)
          ...
          ! Forward calculation of the momentum equation at the surface domain
      enddo
!$omp end parallel do
!$omp barrier

if(rank.ne.npart)then
      call MPI_ISEND(um(imoveupstart(rank):imoveupend(rank)),ighost, &
      MPI_REAL8,rank+1,tag(2), MPI_COMM_WORLD,send_request(2),ierror)
      ...
      call MPI_IRECV(um(imovedownstart(rank+1):imovedownend(rank+1)),ighost, &
      MPI_REAL8,rank+1,tag(4), MPI_COMM_WORLD,recv_request(4),ierror)
      ...
      call MPI_WAIT(send_request(2),status,ierror)
      call MPI_WAIT(send_request(2),status,ierror)
      ...
endif
if(rank.ne.0)then
      call MPI_IRECV(um(imoveupstart(rank-1):imoveupend(rank-1)),ighost, &
      MPI_REAL8,rank-1,tag(2), MPI_COMM_WORLD,recv_request(2),ierror)
      ...
      call MPI_ISEND(um(imovedownstart(rank):imovedownend(rank)),ighost, &
      MPI_REAL8,rank-1,tag(4), MPI_COMM_WORLD,send_request(4),ierror)
      ...
      call MPI_WAIT(recv_request(2),status,ierror)
      call MPI_WAIT(send_request(4),status,ierror)
      ...
endif
end subroutine FLUX
```

Fig. 7. An example of hybrid parallelization in the subroutine FLUX of H12.

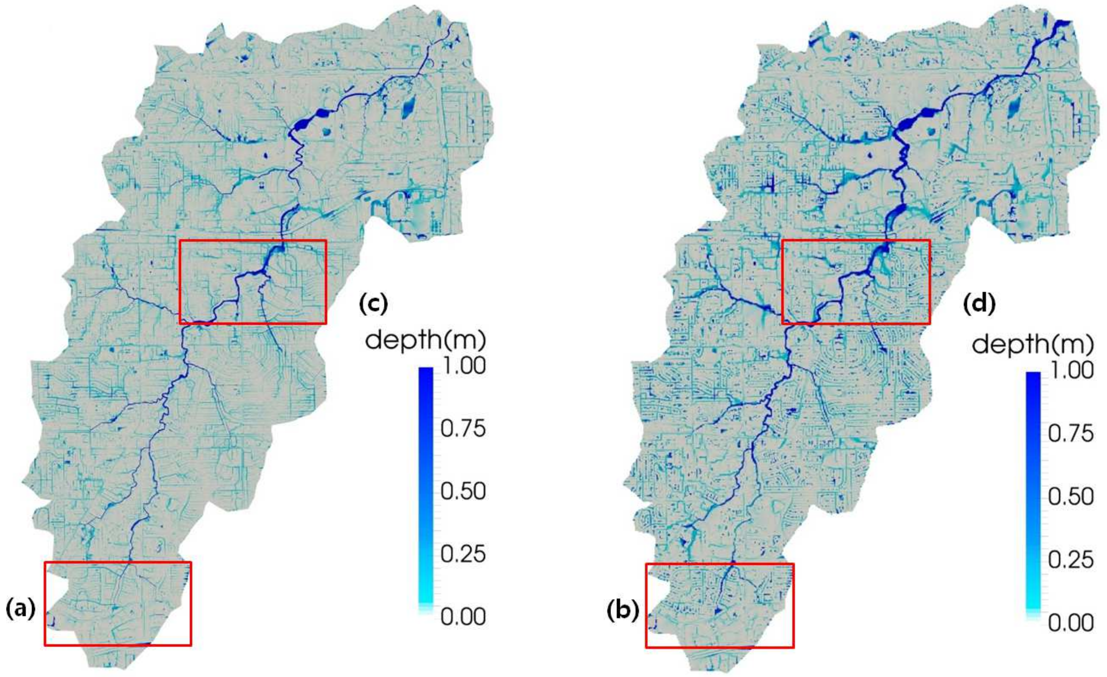
4. Results and discussion

4.1 Analysis of hyper-resolution 1D-2D modelling

In order to assess the realism of hyper-resolution 1D-2D modelling, simulations at two different resolutions were made with grid sizes of 1 and 8 m. Except for the grid size for the surface domain, the same conditions were applied for all other components such as the interaction model and the 1D sewer model. With variable grid sizing, the resolutions of the road grids were 1 and 8 m while those of all other land cover types were twice as large at 2 and 16 m, respectively. The total number of grids in the surface domain at varying resolutions is summarized in Table 2. Each case was simulated for 2 hours with uniform rainfall of 71 mm/hr which is equivalent to 500-year return period precipitation for 2-hour duration (Asquith and Roussel, 2004).

Fig. 8 shows the spatial distribution of water depth at the end of the 2-hour simulation. Overall, the simulated water depth at the two resolutions shows a similar pattern at the catchment scale. At smaller scales, however, considerable differences exists. Compared to the 1 m-resolution result (Fig. 8(a)), the simulated water depth at 8-m resolution shows more dispersion and numerous artificial puddles. Small shallow puddles may temporally form on the surface. Formation of deep puddles, which are frequently found in the 8-m resolution result (Fig. 8(b)), appears to be affected by the coarse resolution. Because the 8 m-resolution grid is obtained from averaging the 1 m-resolution grid (64 = 8 rows × 8 columns), the terrain depicted at the coarse resolution could be prone to distortion especially when the width of flow pass (e.g., road, street and stream) is smaller than the grid resolution.

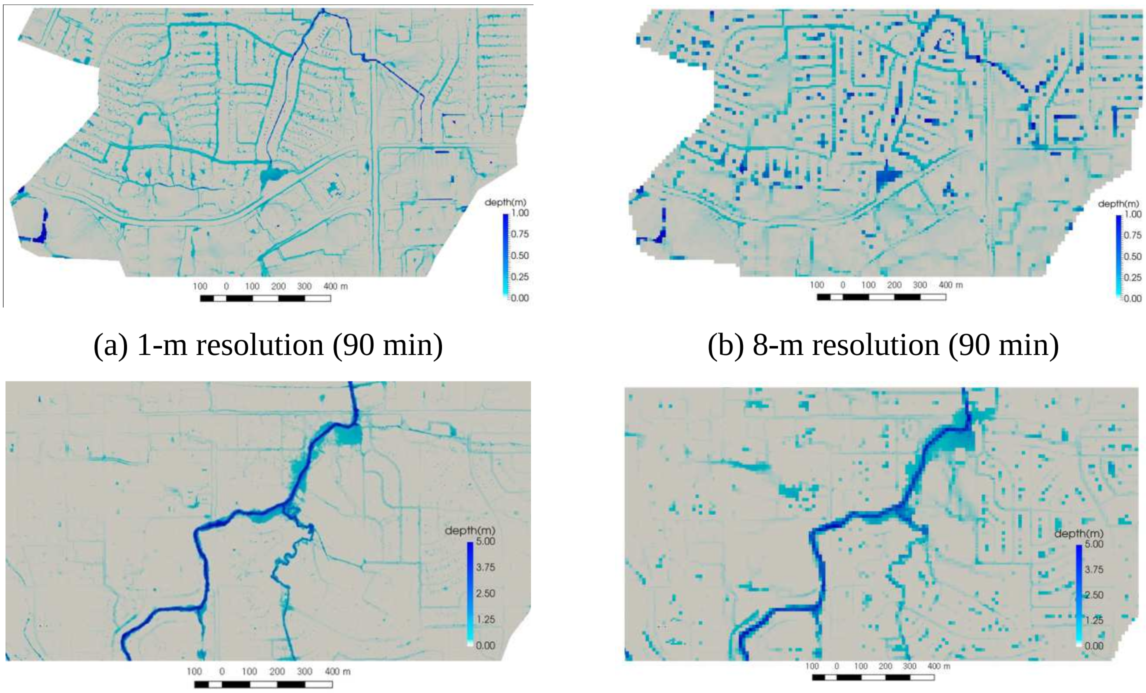
Differences in simulation results between the two resolutions are more clearly observed in Fig. 9 which shows enlarged views for the circled areas in Fig. 8. For larger contrast, results valid at different simulation times were selected for comparison in Fig. 9. Fig. 9 (a) and (b) show the water depth at the southern part of the catchment at 90 min into simulation. In the 1-m resolution result (Fig. 9 (a)), deep water shown in dark blue is consistently found along streets and streams. In the 8-m resolution map (Fig. 9 (b)), it is found that the flow along small streams and streets is disconnected and the storm water is locally trapped in multiple locations. Comparison of water depth in the main stem of Johnson Creek is shown in Fig. 9 (c) and (d). The simulated water depth in the 1-m resolution is deeper than that in the 8-m resolution which renders the higher-resolution simulation to convey more storm water along the creek. Accordingly, the inundation extent in the 1-m resolution is narrower than that in the 8-m resolution along the creek. Whereas ground validation is necessary to verify the above, it is readily seen from the visual examination that hyper-resolution modelling is able to depict storm water flow at small scales more realistically, and that coarse resolution modelling is likely to lose details of the terrain such as channel width and depth, and ground elevation and hence distort flow dynamics and patterns particularly in urban areas. The difference between 1-m and 8-m resolutions does not, however, directly indicate improvement in accuracy but illustrates potential benefits from high-resolution urban inundation modelling. As Teng et al. (2017) noted, there is no such thing as a “perfect model” and the selection of model complexity including resolution must be balanced against various factors such as computation demand and the requirements of the end user. A more rigorous analysis on the relation between model accuracy and grid resolution is left as a future endeavor. We also note here that the study area of the Johnson Creek Catchment is currently being developed into a hydroinformatics testbed (Noh et al., 2016b; Seo et al., 2015) to support validation of H12 and other research activities. The study area is under the CASAWX network of X-band radars from which real-time products of high-resolution quantitative precipitation estimates are also available (Chandrasekar et al., 2012).



(a) 1-m resolution

(b) 8-m resolution

Fig. 8. Simulated water depth using 1- and 8-m resolutions (120 min elapsed).



(a) 1-m resolution (90 min)

(b) 8-m resolution (90 min)

(c) 1-m resolution (120 min)

(d) 8-m resolution (120 min)

Fig. 9. Enlarged views of simulated water depth using 1- and 8-m resolutions.

4.2 Comparison of run time

This subsection assesses the computational performance of the hybrid parallel code, H12. Model grids in the 1D and 2D domains are identical to those in Subsection 4.1 except that run time was evaluated for 1-hour simulation forced by the uniform rainfall of 117 mm/hr or 500-year return period precipitation for 1-hour duration. All cases were run on the Stampede supercomputer at the Texas Advanced Computing Center (TACC, https://www.tacc.utexas.edu/). The majority of the 6400 nodes in Stampede at include two Intel Xeon E5 Sandy Bridge processors in each node and the nodes are interconnected by Mellanox FDR InfiniBand technology. These compute nodes are configured with 32GB of memory.

Fig. 10 compares run times for 1-hour simulation at 1-m resolution using the serial code and the two parallelization methods of OpenMP and hybrid. The run time of 48.8 hrs of the serial code is shown in the red bar. OpenMP parallelization is shown in blue bars with run times ranging from 24.9 to 7.3 hrs as the number of OpenMP threads increases from 2 to 16. The gain from OpenMP parallelization became smaller as the number of threads increased. Specifically, when the number of OpenMP threads doubled from 8 to 16, the run time was reduced only by 18 % from 9.0 to 7.3 hrs presumably due to the configuration of the computing nodes as explained below. Since an Intel Xeon E5 processor includes eight CPU cores, additional overhead may be required to implement OpenMP with 16 threads in two processors in a single node. In hybrid parallelization shown in green bars, run times were assessed with varying MPI tasks and a fixed number of OpenMP threads (8 threads). As the number of MPI tasks increased from 8 to 16, 32, and 48, run times of the hybrid cases were reduced to 1.3, 0.79, 0.67, and 0.61 hours, respectively, which demonstrates the efficacy of hybrid parallelization in hyper-resolution 1D-2D urban flood modelling.

Fig. 11 compares the impact of varying the number of OpenMP threads on run time in hybrid parallelization when the number of MPI tasks was fixed at 8. As expected, a reduction in run time was observed when the number of OpenMP threads increased from 2 to 8. Counterintuitively, however, the run-time increased by 26% as the number of OpenMP threads in the hybrid parallelization increased from 8 to 16. As with the reduced performance seen in OpenMP parallelization in Fig. 10, it appears that the hybrid parallelization remains efficient when the number of OpenMP threads does not exceed the total number of CPU cores in a processor. A similar run-time pattern was also observed when the impact of the number of OpenMP threads was assessed with a large number of MPI tasks (not shown).

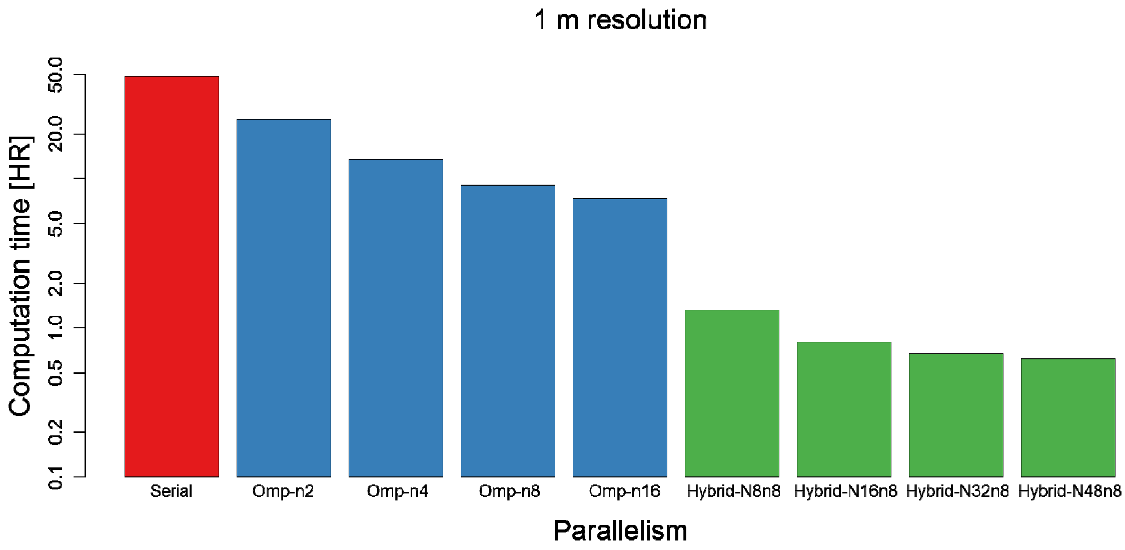


Fig. 10. Comparison of run-times (1-m resolution). Omp and Hybrid denote OpenMP and hybrid parallelization, respectively. ‘N’ and ‘n’ after Omp or Hybrid denote the number of MPI tasks and openMP threads used for computation.

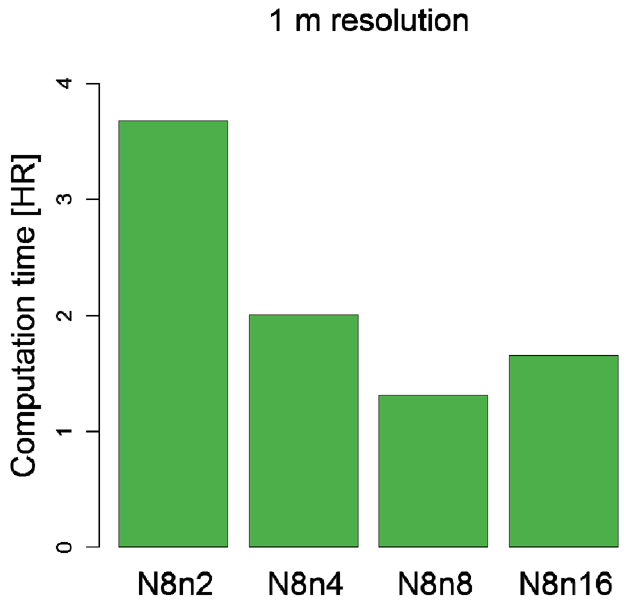


Fig. 11. Comparison of run-times with varying OpenMP threads in hybrid parallelization (1-m resolution). ‘N’ and ‘n’ denote the number of MPI tasks and openMP threads used for computation.

Performance of the parallel code was assessed through two metrics: speedup (Eq. (7)) and efficiency (Eq. (8)). ‘Speedup’ indicates increased computational speed achieved by parallelization over the serial run while ‘efficiency’ refers to the effectiveness of computational resources for speedup. A perfect efficiency equals 1. Efficiency is typically less than 1 due to the overhead associated with parallel computing such as communication and synchronization.

$$Speedup = \frac{Serial\ run\ time}{Parallel\ run\ time} \tag{7}$$

$$Efficiency = \frac{Speedup}{Number\ of\ CPU\ cores} \tag{8}$$

Fig. 12(a) shows the speedup result by the three different parallelization methods of OpenMP (blue line), MPI (grey line) and hybrid (green line). Note that MPI parallelization was implemented by assigning one OpenMP thread per MPI task (i.e., OMP_NUM_THREADS=1) in the hybrid parallel code. The maximum speedup obtained by hybrid parallelization (green line) was 79.0 using 48 MPI tasks and 8 OpenMP threads (384 cores in total). With hybrid parallelization (green line), speedup also increased from 37.2 to 61.2 as the number of total cores doubled from 64 to 128 (from 8 to 16 MPI tasks with 8 OpenMP threads). The rate of speedup in hybrid parallelization (green line) showed a declining pattern: as the number of total cores doubled from 192 to 384, the speedup increased 18 % from 67.2 to 79.0. Speedup of MPI parallelization (grey line) showed a maximum of 29.0 using 64 cores and a decrease in spite of increased cores from 128 to 256. The decrease in speedup in MPI parallelization with a larger number of cores may be understood in terms of computational granularity of the tasks, i.e., the amount of computational work assigned to each task. The larger the number of CPU cores is, the greater the potential for speedup is due to small granularity but at the expense of greater overhead for synchronization and communication among the MPI tasks. With the number of MPI tasks over 64, the amount of communication per MPI task (e.g., ghost grids in the surface domain) does not change but the total communication increases linearly, which negates the gain from computational granularity. In hybrid parallelization, on the other hand, speedup is improved by balancing MPI and OpenMP parallelization even with a large number of cores. From the perspective of the size of each MPI domain, granularity of hybrid parallelization is coarser than that of MPI parallelization. The overhead for OpenMP parallelization, however, does not increase linearly with increasing MPI tasks because in hybrid parallelization communication by OpenMP occurs only within each MPI task. Speedup by OpenMP parallelization ranged from 1.9 to 6.8 as the number of OpenMP threads increased from 2 to 16.

In Fig. 12(b), the parallelization methods were assessed in terms of efficiency. Overall, the most efficient method varied according to the number of cores used. When the number of cores was less than 4, efficiency of OpenMP parallelization (blue line) was higher than 0.9; between 8 and 16, MPI parallelization (grey line) outperformed OpenMP parallelization. When the number of cores was larger than 64, hybrid parallelization (green line) outperformed MPI parallelization (blue line). Speedup by OpenMP parallelization (blue line) increased but attenuated as the number of threads increased. Efficiency of MPI parallelization (grey line) deteriorated sharply as the number of cores exceeded 100. Efficiency of MPI parallelization with 128 and 256 tasks was 0.16 and 0.07, respectively. In hybrid parallelization (green line), on the other hand, efficiency still remained higher than 0.45 until the number of total cores reached 128.

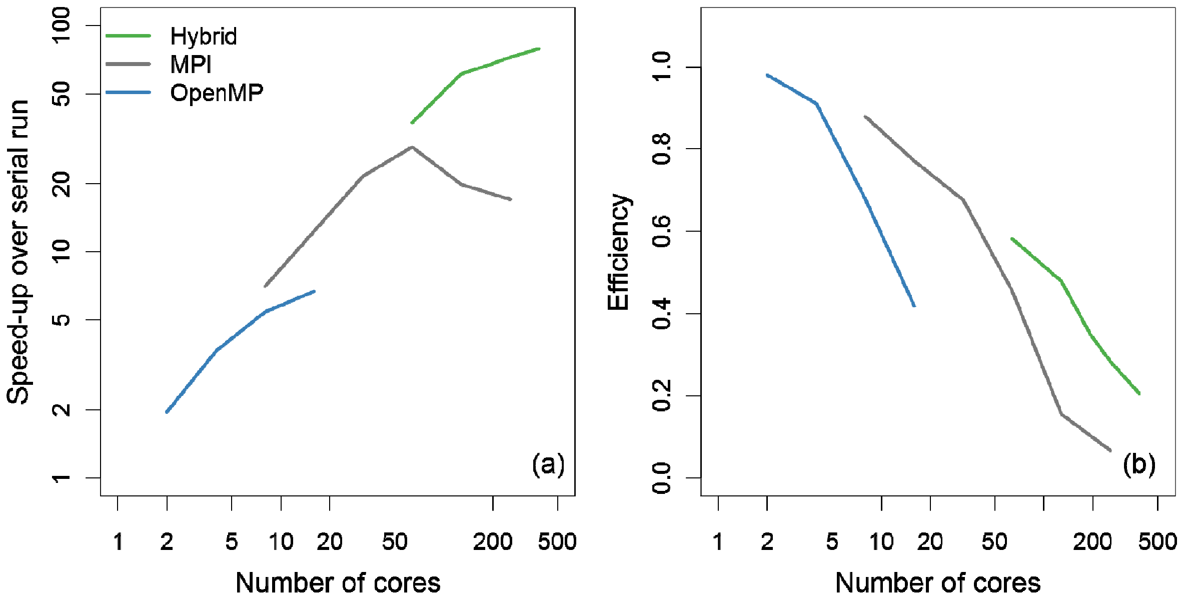


Fig. 12. Speedup and efficiency of parallelization methods (1-m resolution).

In Fig. 10 to Fig. 12, the performance of parallelization methods was analyzed at 1-m resolution. A question may arise whether hybrid parallelization is computationally efficient only in hyper-resolution modelling. Fig. 13 compares performance of hybrid parallelization in coarser but still high-resolution cases where 4- and 8-m resolution simulations were made using different combinations of MPI tasks and OpenMP threads. Except for the grid size, the simulation conditions were identical to those used in the 1-m resolution cases of 1-hour simulation forced by rainfall amount of 117 mm. Even though the same number of CPU cores was used, hybrid parallelization showed a reduction in run time over MPI parallelization in both 4- and 8-m resolution cases. In addition, among hybrid runs, the run time was reduced as more OpenMP threads were assigned. Although the difference in run time between MPI and the best hybrid (N8n8) runs decreased for coarser resolution (8 m), the normalized reduction ratio of run time between the MPI and hybrid parallelization was similar as 51% and 57% for 4 and 8 m resolutions, respectively. The above suggests that the hybrid parallelization developed in this work is applicable to real-world implementation of H12 at high-to-hyper resolutions.

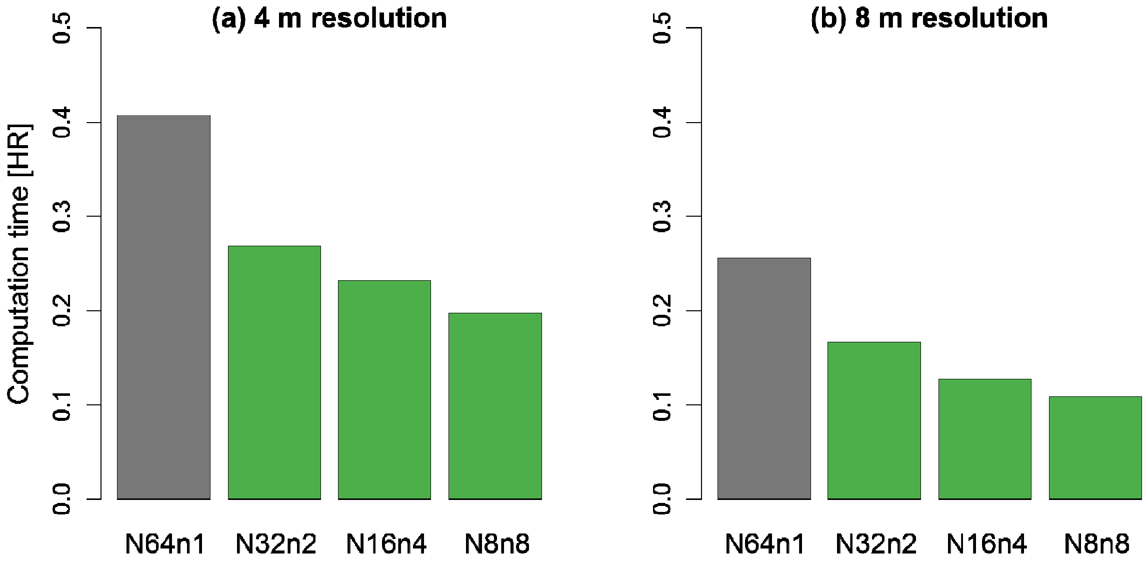


Fig. 13. Comparison of run-times with different MPI and OpenMP cores in MPI and hybrid parallelization (4-m and 8-m resolutions). ‘N’ and ‘n’ denote the number of MPI tasks and openMP threads used for computation. Note that the total number of cores is the same as 64 in all cases.

4.3 Discussion

In the era of hyper-resolution data, parallel computing plays central roles in the development of environmental modelling approaches including urban flood models. However, due to the rapid development of high performance computing (HPC) technologies, significant changes have been made in recent years including heterogeneous architectures, increasing core counts, greater vector widths and sophisticated memory hierarchies which may challenge the development of parallel code (McIntoshi-Smith, 2016). Especially, the use of hundreds of cores running at lower frequency, namely many-core, is an emerging trend in HPC. For instance, a considerable number of leadership class supercomputers rely on many-core processors (e.g., Intel Xeon Phi) or co-processors such as GPU (www.top500.org). It is worth noting that hybrid parallelization is flexible to speed up a wide range of high dimensional hydrologic and hydraulic modelling in various HPC environments without a significant change of the existing code. The major challenge is to find optimal combinations of MPI tasks and OpenMP threads in different many-core systems. H12, which is based on hybrid parallelization, is designed to be implemented in both multi-core and many-core computing systems. We are currently pursuing improving H12 for heterogeneous parallel computing environments and the results will be reported in the near future.

It is worth noting that additional performance improvements could be achieved by vectorization and in-line functions. The key idea of vectorization is to increase the number of computations by increasing the number of operations per cycle (i.e. applying the same operation to multiple items in parallel. Recently, some hardware supports registration of a 512-bit width vector (e.g., AVX-512) which means an 8-fold reduction of the run-time in double precision computation if the code is perfectly vectorized (TACC, 2017). The use of compiler flags is one of the widely selected options to vectorize the code automatically. However, complicated loops and frequent usage of functions and subroutines in the code are major obstacles to automatic vectorization. The use of in-lined functions and manual simplification of the code may further improve performance by vectorization.

One of the key challenges for developing hyper-resolution modelling is collecting in-situ reference data to validate the accuracy of the model. Traditional observations of street flooding from the existing flash flood survey systems are far too sparse to resolve the spatiotemporal variations that hyper-resolution models (such as H12) can represent. Toward that end, non-traditional observational approaches such as crowdsourcing (Habibi et al., 2017) and unmanned aerial vehicle sensing (Perks et al., 2016) as well as development of intensely instrumented testbeds are necessary. ~~We note here that the study area of the Johnson Creek Catchment is currently being developed into a hydroinformatics testbed (Noh et al., 2016b; Seo et al., 2015) to support validation of H12 and other research activities. The study area is under the CASAWX network of X-band radars from which real-time products of high-resolution quantitative precipitation estimates are also available (Chandrasekar et al., 2012).~~

5. Conclusions

In this study, a hybrid parallelization code, H12, was developed for 1D-2D coupled urban flood modelling. The code combines the Open Multi-Processing (OpenMP) and Message Passing Interface (MPI) parallelization methods to enable street-resolving hyper-resolution simulation of movement of water over a large urban area. H12 uses variable grid sizing for detailed geometric representation of urban land surfaces and computational efficiency. To assess H12, the code developed was applied for the Johnson Creek Catchment (~40 km^2) in Arlington, Texas. Model topography is based on 1-m LiDAR data. The major findings of this study are as follows:

- Hyper-resolution modelling depicts storm water flow in urban areas realistically whereas coarser resolution modelling lead to locally isolated and distorted water depth distribution due to lack of topographic details.
- Hybrid parallelization for H12 achieves a 79-fold reduction in computing time compared to the serial run of the same model at 1-m resolution for a 40 km^2 catchment.
- Variable grid sizing reduces the number of the surface grids by 35-45% compared to that by uniform grid sizing which greatly reduces computational cost while maintaining accuracy.
- Speedup by hybrid parallelization consistently increases up to 384 cores in usage whereas that by MPI parallelization decreases as the number of cores exceeds 128.
- Despite the linearly increase in speedup with a smaller number of cores, OpenMP parallelization is not effective in hyper-resolution modelling due to the limited number of cores in the shared-memory system.
- Improved performance of hybrid parallelization over MPI-only parallelization also holds at coarser resolutions of 4- and 8-m grids.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. CyberSEES-1442735 (Dong-Jun Seo, University of Texas at Arlington, PI). S. Lee acknowledges the support from the APEC Climate Center. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. URL: http://www.tacc.utexas.edu. The authors are grateful for constructive comments by two anonymous reviewers and Dr. Haksu Lee of NOAA, USA.

References

Abdullah, A.F., Vojinovic, Z., Price, R.K., Aziz, N. a. A., 2012. Improved methodology for processing raw LiDAR data to support urban flood modelling – accounting for elevated roads and bridges. J. Hydroinformatics 14, 253–269. doi:10.2166/hydro.2011.009

Abily, M., Bertrand, N., Delestre, O., Gourbesville, P., Duluc, C.-M., 2016. Spatial Global Sensitivity Analysis of High Resolution classified topographic data use in 2D urban flood modelling. Environ. Model. Softw. 77, 183–195. doi:10.1016/j.envsoft.2015.12.002

Adeogun, A.G., Daramola, M.O., Pathirana, A., 2015. Coupled 1D-2D hydrodynamic inundation model for sewer overflow: Influence of modeling parameters. Water Sci. 29, 146–155. doi:10.1016/j.wsj.2015.12.001

Ament, M., Frey, S., Sadlo, F., Ertl, T., Weiskopf, D., 2011. GPU-Based Two-Dimensional Flow Simulation Steering using Coherent Structures, in: Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering. Civil-Comp Press, Stirlingshire, UK. doi:10.4203/ccp.95.18

Asquith, W., Roussel, M., 2004. Depth-Duration Frequency of Precipitation for Texas (No. FHWA/TX-04/5-1301-01-1), Scientific Investigations Report. U.S. Geological Survey, Austin, TX.

Bates, P.D., Marks, K.J., Horritt, M.S., 2003. Optimal use of high-resolution topographic data in flood inundation models. Hydrol. Process. 17, 537–557. doi:10.1002/hyp.1113

Bazin, P.-H., Nakagawa, H., Kawaike, K., Paquier, A., Mignot, E., 2014. Modeling Flow Exchanges between a Street and an Underground Drainage Pipe during Urban Floods. J. Hydraul. Eng. 140, 04014051. doi:10.1061/(ASCE)HY.1943-7900.0000917

Beven, K., 2007. Towards integrated environmental models of everywhere: uncertainty, data and modelling as a learning process. Hydrol Earth Syst Sci 11, 460–467. doi:10.5194/hess-11-460-2007

Bierkens, M.F.P., Bell, V.A., Burek, P., Chaney, N., Condon, L.E., David, C.H., de Roo, A., Döll, P., Drost, N., Famiglietti, J.S., Flörke, M., Gochis, D.J., Houser, P., Hut, R., Keune, J., Kollet, S., Maxwell, R.M., Reager, J.T., Samaniego, L., Sudicky, E., Sutanudjaja, E.H., van de Giesen, N., Winsemius, H., Wood, E.F., 2015. Hyper-resolution global hydrological modelling: what is next? Hydrol. Process. 29, 310–320. doi:10.1002/hyp.10391

Borges, F., Gutierrez-Milla, A., Suppi, R., Luque, E., 2014. A Hybrid MPI+OpenMP Solution of the Distributed Cluster-based Fish Schooling Simulator. Procedia Comput. Sci., 2014 International Conference on Computational Science 29, 2111–2120. doi:10.1016/j.procs.2014.05.195

Chandrasekar, V., Wang, Y., Chen, H., 2012. The CASA quantitative precipitation estimation system: a five year validation study. Nat Hazards Earth Syst Sci 12, 2811–2820. doi:10.5194/nhess-12-2811-2012

Chapman, B., Jost, G., Pas, R. van der, 2007. Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press.

Chaudhry, M.H., 1979. Applied hydraulic transients. Van Nostrand Reinhold.

Chen, A.S., Leandro, J., Djordjević, S., 2015. Modelling sewer discharge via displacement of manhole covers during flood events using 1D/2D SIPSON/P-DWave dual drainage simulations. Urban Water J. 13, 1–11. doi:10.1080/1573062X.2015.1041991

Djordjević, S., Prodanović, D., Maksimović, Č., 1999. An approach to simulation of dual drainage. Water Sci. Technol. 39, 95–103. doi:10.1016/S0273-1223(99)00221-8

Djordjević, S., Vojinović, Z., Dawson, R., Savić, D.A., 2014. Uncertainties in Flood Modelling in Urban Areas, in: Applied Uncertainty Analysis for Flood Risk Management. IMPERIAL COLLEGE PRESS.

Dottori, F., Di Baldassarre, G., Todini, E., 2013. Detailed data is welcome, but with a pinch of salt: Accuracy, precision, and uncertainty in flood inundation modeling. Water Resour. Res. 49, 6079–6085. doi:10.1002/wrcr.20406

Fewtrell, T.J., Bates, P.D., Horritt, M., Hunter, N.M., 2008. Evaluating the effect of scale in flood inundation modelling in urban environments. Hydrol. Process. 22, 5107–5118. doi:10.1002/hyp.7148

Fraga, I., Cea, L., Puertas, J., 2015. Validation of a 1D-2D dual drainage model under unsteady part-full and surcharged sewer conditions. Urban Water J. 14, 1–11. doi:10.1080/1573062X.2015.1057180

Gallegos, H.A., Schubert, J.E., Sanders, B.F., 2009. Two-dimensional, high-resolution modeling of urban dam-break flooding: A case study of Baldwin Hills, California. Adv. Water Resour. 32, 1323–1335. doi:10.1016/j.advwatres.2009.05.008

Gorobets, A.V., Trias, F.X., Oliva, A., 2013. A parallel MPI + OpenMP + OpenCL algorithm for hybrid supercomputations of incompressible flows. Comput. Fluids 88, 764–772. doi:10.1016/j.compfluid.2013.05.021

Gropp, W., Lusk, E., Doss, N., Skjellum, A., 1996. A High-performance, Portable Implementation of the MPI Message Passing Interface Standard. Parallel Comput 22, 789–828. doi:10.1016/0167-8191(96)00024-5

Habibi, H., Nazari, B., Norouzi, A., Noh, S.J., Seo, D.-J., Sinha, S., Yu, X., Bartos, M., Kerkez, B., Lakshman, L., Zink, M., Lyons, E., Philips, B.J., Jangyodsuk, P., Gao, J., 2017. Integrated Sensing and Prediction of Flash Floods for the Dallas-Fort Worth Metroplex (DFW). Presented at the 97th American Meteorological Society Annual Meeting.

Hirabayashi, Y., Mahendran, R., Koirala, S., Konoshima, L., Yamazaki, D., Watanabe, S., Kim, H., Kanae, S., 2013. Global flood risk under climate change. Nat. Clim. Change 3, 816–821. doi:10.1038/nclimate1911

Ibanez, D., Dunn, I., Shephard, M.S., 2016. Hybrid MPI-thread parallelization of adaptive mesh operations. Parallel Comput. 52, 133–143. doi:10.1016/j.parco.2016.01.003

Intel, 2017. Intel® Xeon Phi$^{\mathrm{TM}}$ Coprocessor Developer’s Quick Start Guide | Intel® Software [WWW Document]. URL https://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-developers-quick-start-guide (accessed 7.20.17).

Karen R. Ryberg, Wei Lin, Aldo V. Vecchia, 2014. Impact of Climate Variability on Runoff in the North-Central United States. J. Hydrol. Eng. 19, 148–158. doi:10.1061/(ASCE)HE.1943-5584.0000775

Kawaike, K., Nakagawa, H., Baba, Y., Shimizu, A., 2011. Experimental Study on Validation of Stormwater Interaction Model between the Ground Surface and Sewerage System. Proc. 34th World Congr. Int. Assoc. Hydro-Environ. Res. Eng.

Kesserwani, G., Liang, Q., 2012. Dynamically adaptive grid based discontinuous Galerkin shallow water model. Adv. Water Resour. 37, 23–39. doi:10.1016/j.advwatres.2011.11.006

Leandro, J., Chen, A.S., Djordjević, S., Savić, D.A., 2009. Comparison of 1D/1D and 1D/2D Coupled (Sewer/Surface) Hydraulic Models for Urban Flood Simulation. J. Hydraul. Eng. 135, 495–504. doi:10.1061/(ASCE)HY.1943-7900.0000037

Leandro, J., Chen, A.S., Schumann, A., 2014. A 2D parallel diffusive wave model for floodplain inundation with variable time step (P-DWave). J. Hydrol. 517, 250–259. doi:10.1016/j.jhydrol.2014.05.020

Lee, S., 2013. Study on Experimental Validation of Interaction Model among Sewer Pipe, Manhole and Ground Surface for Development of Integrated Urban Inundation Model. Proc. 35th World Congr. Int. Assoc. Hydro-Environ. Res. Eng.

Lee, S., Nakagawa, H., Kawaike, K., Zhang, H., 2016. Urban inundation simulation considering road network and building configurations. J. Flood Risk Manag. 9, 224–233. doi:10.1111/jfr3.12165

Lee, S., Nakagawa, H., Kawaike, K., Zhang, H., 2015. Urban Inundation Simulation Considering Road Network and Building Configurations. J. Flood Risk Manag. in press. doi:10.1111/jfr3.12165

Lee, S., Nakagawa, H., Kawaike, K., Zhang, H., 2012. Study on Inlet Discharge Coefficient Through the Different Shapes of Storm Drains for Urban Inundation Analysis. J. Jpn. Soc. Civ. Eng. Ser B1 Hydraul. Eng. 68, I_31-I_36. doi:10.2208/jscejhe.68.I_31

Leitão, J.P., Boonya-Aroonnet, S., Prodanović, D., Maksimović, C., 2009. The influence of digital elevation model resolution on overland flow networks for modelling urban pluvial flooding. Water Sci. Technol. J. Int. Assoc. Water Pollut. Res. 60, 3137–3149. doi:10.2166/wst.2009.754

Liang, Q., 2011. A structured but non-uniform Cartesian grid-based model for the shallow water equations. Int. J. Numer. Methods Fluids 66, 537–554. doi:10.1002/fld.2266

Liu, L., Liu, Y., Wang, X., Yu, D., Liu, K., Huang, H., Hu, G., 2015. Developing an effective 2-D urban flood inundation model for city emergency management based on cellular automata. Nat Hazards Earth Syst Sci 15, 381–391. doi:10.5194/nhess-15-381-2015

Mallakpour, I., Villarini, G., 2015. The changing nature of flooding across the central United States. Nat. Clim. Change 5, 250–254. doi:10.1038/nclimate2516

Mark, O., Weesakul, S., Apirumanekul, C., Aroonnet, S.B., Djordjević, S., 2004. Potential and limitations of 1D modelling of urban flooding. J. Hydrol., Urban Hydrology 299, 284–299. doi:10.1016/j.jhydrol.2004.08.014

Marks, K., Bates, P., 2000. Integration of high-resolution topographic data with floodplain flow models. Hydrol. Process. 14, 2109–2122. doi:10.1002/1099-1085(20000815/30)14:11/12<2109::AID-HYP58>3.0.CO;2-1

Mason, D.C., Giustarini, L., Garcia-Pintado, J., Cloke, H.L., 2014. Detection of flooded urban areas in high resolution Synthetic Aperture Radar images using double scattering. Int. J. Appl. Earth Obs. Geoinformation 28, 150–159. doi:10.1016/j.jag.2013.12.002

Mason, D.C., Horritt, M.S., Hunter, N.M., Bates, P.D., 2007. Use of fused airborne scanning laser altimetry and digital map data for urban flood modelling. Hydrol. Process. 21, 1436–1447. doi:10.1002/hyp.6343

McIntoshi-Smith, S., 2016. HPC future trends from a science perspective, in: Proceedings of 4th ENES Workshop on High Performance Computing for Climate and Weather. Toulouse, France.

Meesuk, V., Vojinovic, Z., Mynett, A.E., Abdullah, A.F., 2015. Urban flood modelling combining top-view LiDAR data with ground-view SfM observations. Adv. Water Resour. 75, 105–117. doi:10.1016/j.advwatres.2014.11.008

Michel-Kerjan, E., Kunreuther, H., 2011. Redesigning Flood Insurance. Science 333, 408–409. doi:10.1126/science.1202616

Mignot, E., Paquier, A., Haider, S., 2006. Modeling floods in a dense urban area using 2D shallow water equations. J. Hydrol. 327, 186–199. doi:10.1016/j.jhydrol.2005.11.026

Mininni, P.D., Rosenberg, D., Reddy, R., Pouquet, A., 2011. A hybrid MPI–OpenMP scheme for scalable parallel pseudospectral computations for fluid turbulence. Parallel Comput. 37, 316–326. doi:10.1016/j.parco.2011.05.004

Neal, J., Bates, P.D., Fewtrell, T.J., Hunter, N.M., Wilson, M.D., Horritt, M.S., 2009a. Distributed whole city water level measurements from the Carlisle 2005 urban flood event and comparison with hydraulic model simulations. J. Hydrol. 368, 42–55. doi:10.1016/j.jhydrol.2009.01.026

Neal, J., Fewtrell, T., Trigg, M., 2009b. Parallelisation of storage cell flood models using OpenMP. Environ. Model. Softw. 24, 872–877. doi:10.1016/j.envsoft.2008.12.004

Neal, J., Fewtrell, T.J., Bates, P.D., Wright, N.G., 2010. A comparison of three parallelisation methods for 2D flood inundation models. Environ. Model. Softw. 25, 398–411. doi:10.1016/j.envsoft.2009.11.007

Neal, J., Villanueva, I., Wright, N., Willis, T., Fewtrell, T., Bates, P., 2012. How much physical complexity is needed to model flood inundation? Hydrol. Process. 26, 2264–2282. doi:10.1002/hyp.8339

Noh, S.J., Lee, S., An, H., Kawaike, K., Nakagawa, H., 2016a. Ensemble urban flood simulation in comparison with laboratory-scale experiments: Impact of interaction models for manhole, sewer pipe, and surface flow. Adv. Water Resour. 97, 25–37. doi:10.1016/j.advwatres.2016.08.015

Noh, S.J., Nazari, B., Hamideh, H., Seo, D.-J., Lee, J., 2016b. iSPUW: integrated sensing and prediction of urban water for sustainable cities. Presented at the 2016 AGU Fall Meeting, San Franscisco.

Ozdemir, H., Sampson, C.C., de Almeida, G.A.M., Bates, P.D., 2013. Evaluating scale and roughness effects in urban flood modelling using terrestrial LIDAR data. Hydrol Earth Syst Sci 17, 4015–4030. doi:10.5194/hess-17-4015-2013

Perks, M.T., Russell, A.J., Large, A.R.G., 2016. Technical Note: Advances in flash flood monitoring using unmanned aerial vehicles (UAVs). Hydrol Earth Syst Sci 20, 4005–4015. doi:10.5194/hess-20-4005-2016

Sampson, C.C., Smith, A.M., Bates, P.D., Neal, J.C., Alfieri, L., Freer, J.E., 2015. A high-resolution global flood hazard model. Water Resour. Res. 51, 7358–7381. doi:10.1002/2015WR016954

Sanders, B.F., Schubert, J.E., Detwiler, R.L., 2010. ParBreZo: A parallel, unstructured grid, Godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale. Adv. Water Resour. 33, 1456–1467. doi:10.1016/j.advwatres.2010.07.007

Satarić, B., Slavnić, V., Belić, A., Balaž, A., Muruganandam, P., Adhikari, S.K., 2016. Hybrid OpenMP/MPI programs for solving the time-dependent Gross–Pitaevskii equation in a fully anisotropic trap. Comput. Phys. Commun. 200, 411–417. doi:10.1016/j.cpc.2015.12.006

Schubert, J.E., Sanders, B.F., Smith, M.J., Wright, N.G., 2008. Unstructured mesh generation and landcover-based resistance for hydrodynamic modeling of urban flooding. Adv. Water Resour. 31, 1603–1621. doi:10.1016/j.advwatres.2008.07.012

Seo, D.-J., Kerkez, B., Zink, M., Fang, Z., Gao, J., Yu, X., 2015. iSPUW: A Vision for Integrated Sensing and Prediction of Urban Water for Sustainable Cities., in: Dynamic Data-Driven Environmental Systems Science.

Shen, D., Wang, J., Cheng, X., Rui, Y., Ye, S., 2015. Integration of 2-D hydraulic model and high-resolution lidar-derived DEM for floodplain flow modeling. Hydrol Earth Syst Sci 19, 3605–3616. doi:10.5194/hess-19-3605-2015

Sibson, R., 1981. A brief description of natural neighbor interpolation, in: Barnett, V. (Ed.), Interpreting Multivariate Data. Wiley, Chichester [West Sussex]; New York.

TACC, 2017. Introduction to Manycore Programming.

Teng, J., Jakeman, A.J., Vaze, J., Croke, B.F.W., Dutta, D., Kim, S., 2017. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. Environ. Model. Softw. 90, 201–216. doi:10.1016/j.envsoft.2017.01.006

Wan, X., Lin, G., 2013. Hybrid parallel computing of minimum action method. Parallel Comput. 39, 638–651. doi:10.1016/j.parco.2013.08.004