High Performance Computing of Coupling 2D and 3D Numerical Modelling of Flood Propagation and its High Performance Interface and Visualisation

Name: Bobby Minola Ginting
E-Mail: bobbyminola.ginting@tum.de
Supervisor: Prof. Dr. rer. nat. Ernst Rank
PD Dr. rer. nat. habil. Ralf-Peter Mundani
Chair for Computation in Engineering
Started: Oct/2015
☑ ongoing ☐ finished

Numerical modelling of flood propagation problems has become an interesting topic since the last two decades. This topic plays a very significant role in simulating hydrodynamics in many areas, e.g. in rivers, lakes, flood plains and coastal regions. Using a numerical model, the characteristics of flow such as depth and velocity, can be investigated. In some countries, numerical models have even been extensively used as tools for a flood early warning system. Cheaper, faster and accurate are the reasons why people would rather do a numerical model than do a laboratory one. Having an accurate numerical model for those purposes remains an outstanding research challenge.

Solving the 2D shallow water equations (SWEs) is a well-established approach in order to simulate flood flows. The 2D SWEs, which are derived from the integration of the 3D Navier-Stokes equations (NSEs) in the vertical direction with the assumptions of uniformly distributed vertical velocity, negligible hydrostatic pressure distribution, free surface, small bed inclination and incompressibility, are solved using a numerical model to obtain the solutions of fluid flows even for the most complicated flow phenomena, such as transcritical flows, shock flows and wet-dry interfaces. The cell-centred finite volume (CCFV) method used in [1] is one of many numerical schemes that can be used to solve the 2D SWEs accurately. This scheme was very robust, relatively cheap, and highly flexible for domains with complex shapes. This is obviously beneficial for practical purposes in simulating flood flows as well as for related stakeholders to do fast responses concerning the flood early warning system. In Figs. 1 - 2, some results of our 2D model are presented.
Unfortunately, for some cases in which vertical profiles are of main interest and hydrostatic pressure assumption fails e.g. vertically circulating flows, the 2D model does not apply and the 3D model is thus required. The model presented in [2] was a 3D non-hydrostatic model and capable of producing non-oscillatory and highly accurate results. The sigma coordinate system in the vertical direction of this model enabled the simulations of topography changes. Despite being more accurate than 2D models, 3D models are still quite rarely used for simulating flood flows. The main reason is that the computation of a full 3D model may lead to a huge computational overhead; this can even become worse for large domains.

The main idea of this research emerged from the above problem that the 2D SWEs model will be coupled with the 3D NSEs model, thus saving more computational time. For areas in which wet-dry interfaces are dominant or vertical profiles are not of main interest, the 2D SWEs model is employed to preserve the well-balanced property or C-property. The 3D NSEs model is then employed for areas in which non-hydrostatic pressure condition applies. This coupled model will be solved using a finite volume method which is very flexible for general applications. The most challenging part of this coupling problem is the treatment of boundary condition at the interfaces between the 2D and 3D models. We hypothesise that at those interfaces, the water elevation can be specified using the 2D solvers. Also, the velocity can be computed using such solvers; however, a depth averaged condition should be used.

With regard to high performance computing, a hybrid programming which combines the OpenMP and MPI parallel programming is employed. Our code will be performed using the OpenMP for shared memory parallelisation inside of each node and communicated with other nodes using the MPI which is developed for distributed memory parallelisation. Using this hybrid model, some advantages, such as eliminating domain decomposition at node and lower memory latency and data movement within node, can be achieved.

References
