ARTICLE IN PRESS

Computers and Mathematics with Applications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

High performance shallow water kernels for parallel overland flow simulations based on FullSWOF2D

Roland Wittmann *, Hans-Joachim Bungartz, Philipp Neumann

Technical University of Munich, Boltzmannstraße 3, 85748 Garching, Germany

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Vectorization
Overland flows
Shallow water equations
High performance kernels

ABSTRACT

We describe code optimization and parallelization procedures applied to the sequential overland flow solver FullSWOF2D. Major difficulties when simulating overland flows comprise dealing with high resolution datasets of large scale areas which either cannot be computed on a single node either due to limited amount of memory or due to too many (time step) iterations resulting from the CFL condition. We address these issues in terms of two major contributions. First, we demonstrate a generic step-by-step transformation of the second order finite volume scheme in FullSWOF2D towards MPI parallelization. Second, the computational kernels are optimized by the use of templates and a portable vectorization approach. We discuss the load imbalance of the flux computation due to dry and wet cells and propose a solution using an efficient cell counting approach. Finally, scalability results are shown for different test scenarios along with a flood simulation benchmark using the Shaheen II supercomputer.

1. Introduction

Over the last decade we observe an increase in the number and intensity of floods world-wide. These floods cause widespread destruction as well as losses of human lives and economic property. Efficient simulation of these events at larger scales (e.g. 10–100 km^2) and at high resolution is required to, e.g., design adequate countermeasures in the long term and to provide accurate forecasts in the case of immanent extreme weather events. Various efforts have been undertaken in this regard: In [1], Huang et al. use a quasi-2D modeling approach to reduce the computational work for simulating floods along the Elbe River in Germany, and compare the results to a 2D finite volume code. Sanders et al. [2] have developed a parallel unstructured grid code, called ParBreZo, to simulate flood events on a regional scale.

The simulation of overland flows poses major challenges with respect to numerical algorithms and their efficient parallel implementations. Different numerical algorithms are still being developed either using explicit or (semi-) implicit [3] time stepping schemes. While explicit methods allow for simple implementations and parallelization, they suffer from very small time steps due to the Courant–Friedrichs–Levy (CFL) condition when using high resolution datasets. Though implicit methods allow larger time steps for these kinds of scenarios, they require the solution of a sparse linear equation system in each time step. This becomes expensive and yields more challenges for efficient parallelization. Besides, one has to properly treat drying and wetting of areas in the computational domain. On the implementational side, this usually requires different implementations through branches to select the appropriate scheme for dry or wet areas. On the one hand, this may cause a severe load imbalance because the computation of numerical fluxes in dry areas is cheaper than the computation in

* Corresponding author.
E-mail addresses: roland.wittmann@mytum.de (R. Wittmann), bungartz@in.tum.de (H.-J. Bungartz), philipp.neumann@tum.de (P. Neumann).

http://dx.doi.org/10.1016/j.camwa.2017.01.005
0898-1221/

Please cite this article in press as: R. Wittmann, et al., High performance shallow water kernels for parallel overland flow simulations based on FullSWOF2D, Computers and Mathematics with Applications (2017), http://dx.doi.org/10.1016/j.camwa.2017.01.005

wet areas. On the other hand, the optimization of these potentially dynamic branches is still a big challenge for today's compilers. While compilers are able to efficiently optimize branches which simply switch between single values, branches such as in the drying/wetting case may require completely different computational approaches, effectively preventing autovectorization of large code parts. However, efficient vectorization is crucial to harness the capabilities of today's computer architectures with respect to computational speed and efficient use of memory bandwidth. Bader et al. studied for example the implementation of augmented Riemann solvers which benefit from compiler-based auto-vectorization [4].

In this paper, we demonstrate how we can benefit from the object-oriented design to *incorporate parallelism and highly efficient kernels using vectorization and templates into existing flooding simulations*. For this purpose, we make use of the solver package "**Full S**hallow **W**ater equations for **O**verland **F**low" (FullSWOF2D) [5]. This finite-volume solver was specifically designed for the simulation of flooding and respective drying/wetting scenarios. Its object-oriented C++-based design allows for modular extensions and modifications. FullSWOF2D operates on a regular Cartesian grid which is beneficial with regard to the application of optimization techniques. However, the code itself does not offer any form of parallelism yet although there has already been an attempt using SkelGIS [6] before, comparing it with a MPI implementation. Recently, Unterweger et al. [7] demonstrated how FullSWOF2D can be used in an adaptive mesh refinement context to simulate rain-induced overland flows through adaptive local time stepping using the software package PeanoClaw.

The remainder is structured as follows: in Section 2, we briefly summarize the hyperbolic system of balance equations based on the shallow-water model and the corresponding source terms along with the numerical methods. We continue with a step-by-step transformation of the original code towards a parallelized algorithm using MPI in Section 3. In Section 4, we optimize the performance bottlenecks using templates as a main building block. We further discuss issues for auto-vectorization which motivates our design for a portable vectorization concept to deal with cases where auto-vectorization fails. Using this concept we have developed a fully AVX-based version of the Harten–Lax–van Leer-Contact (HLLC) Riemann solver for the shallow water equations. The used optimization techniques are not limited to purely regular Cartesian grids, but they can also be used for block-structured grids and unstructured grids as well. In [8], a concept for block-structured grids is discussed which allows the dynamic fusion of grids. Hence, smaller grids from the leaf nodes are then replaced by bigger grids in refined nodes of an adaptive mesh refinement tree. This approach allows a more efficient application of auto-vectorization and manual vectorization techniques at the expense of additional memory copies while still being able to benefit from adaptive mesh refinement. In the case of unstructured grids, in [9] the authors present a GPU implementation of a shallow water solver on an unstructured triangular grid using CUDA. Their implementation uses three accumulator arrays to track the contributions along the three edges of each triangular volume and the corresponding neighboring volumes. The size of these arrays is determined by the number of volumes in the grid and they parallelize over all edges. Our presented vectorization technique may be applied to their concept by adjusting the intrinsics for re-combining vector contents with respect to the neighboring volumes along each edge. Nevertheless, depending on the traversal scheme at hand this may require an additional pre-sorting step on a subset of grid cells to ensure that the vector intrinsics always combine the right neighbors, independent of the current selection of neighbors. Further, we embedded a novel dry-cell handling mechanism which only requires a single branch for a set of cells instead of a branch for each cell to effectively deal with load imbalances. We demonstrate the effectiveness of our approach for two extreme case scenarios, cf. Section 5. In Section 6, we show the benefits of our transformations and optimization by applying the optimized FullSWOF software to the simulation of a flooding in Glasgow, United Kingdom. We close with a short summary in Section 7 and give an outlook to future work.

2. FullSWOF2D and the shallow water equations

FullSWOF2D solves the shallow water equations

$$\begin{aligned}
\partial_t h + \partial_x(hu) + \partial_y(hv) &= R - I \\
\partial_t(hu) + \partial_x\left(hu^2 + \frac{gh^2}{2}\right) + \partial_y(huv) &= gh(S_{0_x} - S_{f_x}) \\
\partial_t(hv) + \partial_x(huv) + \partial_y\left(hv^2 + \frac{gh^2}{2}\right) &= gh(S_{0_y} - S_{f_y}) \\
-\partial_x z_b(x,y) &= S_{0_x} \\
-\partial_y z_b(x,y) &= S_{0_y}
\end{aligned} \tag{1}$$

using a finite volume discretization. The variable h denotes the water height above ground, z_b the height of the topography, u and v denote the velocities in x and y direction, respectively. The rain intensity is determined by R, I specifies the infiltration rate. The gravity is given by parameter g. The spatial topography gradients $\partial_x z_b$ and $\partial_y z_b$ are defined by the negative slopes S_{0_x} and S_{0_y} in x and y direction, respectively. The shallow water equations (1) are solved numerically in 10 steps, cf. Fig. 1. In steps 1 and 5, the boundary conditions depending on the user's choice are set in the boundary layer of the grid. FullSWOF2D offers wall boundary conditions, Neumann conditions, imposed discharge height and imposed discharge momentum conditions as well as periodic boundaries. In steps 2 and 6, a reconstruction method is applied to achieve second order in space, either based on Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) or Essentially Non-Oscillatory (ENO) schemes. For the details of both schemes, we refer to [10]. The fluxes are computed in step 3 using the values from step 2

Please cite this article in press as: R. Wittmann, et al., High performance shallow water kernels for parallel overland flow simulations based on FullSWOF2D, Computers and Mathematics with Applications (2017), http://dx.doi.org/10.1016/j.camwa.2017.01.005