Available online at www.sciencedirect.com

ScienceDirect

Publishing Services by Elsevier

International Journal of Naval Architecture and Ocean Engineering 8 (2016) 434–441

http://www.journals.elsevier.com/international-journal-of-naval-architecture-and-ocean-engineering/

Numerical discrepancy between serial and MPI parallel computations

Sang Bong Lee

Department of Naval Architecture and Offshore Engineering, Dong-A University, Busan, South Korea

Received 20 November 2015; revised 11 May 2016; accepted 12 May 2016
Available online 21 July 2016

Abstract

Numerical simulations of 1D Burgers equation and 2D sloshing problem were carried out to study numerical discrepancy between serial and parallel computations. The numerical domain was decomposed into 2 and 4 subdomains for parallel computations with message passing interface. The numerical solution of Burgers equation disclosed that fully explicit boundary conditions used on subdomains of parallel computation was responsible for the numerical discrepancy of transient solution between serial and parallel computations. Two dimensional sloshing problems in a rectangular domain were solved using OpenFOAM. After a lapse of initial transient time sloshing patterns of water were significantly different in serial and parallel computations although the same numerical conditions were given. Based on the histograms of pressure measured at two points near the wall the statistical characteristics of numerical solution was not affected by the number of subdomains as much as the transient solution was dependent on the number of subdomains.

Keywords: MPI; Parallel; Serial; OpenFOAM; Sloshing

1. Introduction

With the use of high performance computing becoming more prevalent, parallel computing is generally used in numerical simulations of naval hydrodynamics to increase computational efficiency. The parallel computation is divided into two categories: Shared Memory Programming (SMP) and Message Passing Programing (MPP). All central processing units of SMP have to be connected to a large shared memory physically or at least logically, which means that the parallel scalability of SMP is subordinate to hardware architecture. It is why SMP is not able to be popular despite its great advantages that a source code can be easily parallelized and numerical discrepancy between serial and parallel computations is negligible. Unlike SMP, the main attraction of MPP is the unlimited scalability for parallelization as long as a network communication is fast enough. Moreover a relatively lower price is another advantage of MPP machines. Nevertheless it is not easy to build up an efficient Message Passing Interface (MPI) of data between several machines.

As shown in Fig. 1, a numerical domain has to be decomposed into several subdomains to perform a parallel simulation using MPP. An inevitable problem of the domain decomposition is how to handle boundary conditions of each subdomain. Simply an explicit boundary condition can be applied to all of subdomains. For example in Fig. 1(a), the previous value of cell B_{11} is explicitly used as a boundary value for cell A_{1N} while cell B_{11} takes the previous value of cell A_{N1} for an explicit boundary condition. Because all of boundary values are definitely known on every subdomain it is possible to synchronize the parallel computation of subdomains and maximize the parallel efficiency by using the explicit boundary conditions as illustrated in Fig. 1(b). With the use of explicit boundary condition the parallel efficiency of MPP is theoretically dependent on the latency of network communication. A different way can be considered to improve the updating speed between subdomains. After cell B_{11} wait until the value of cell A_{1N} is determined, the new value of cell A_{1N} is able to be used as an updated boundary value for cell

E-mail address: sblee1977@dau.ac.kr.
Peer review under responsibility of Society of Naval Architects of Korea.

http://dx.doi.org/10.1016/j.ijnaoe.2016.05.011

Nomenclature

A	coefficient matrix
d	diffusion coefficient
L_i, D_i, U_i	discretized coefficients
Re	Reynolds number
S	source term
t	time
u	solution of Burgers equation
U	velocity of sloshing flows
x, y	coordinate system
Δt	time interval
Δu	difference of solution
ρ	density
ν	kinematic viscosity
ϕ	source matrix
air	air
water	water
(parallel)	parallel computation
(serial)	serial computation
(2 subdomains)	parallel computation using 2 subdomains
(*4 subdomains*)	parallel computation using 4 subdomains
n	time step

B_{11}. Similarly cell B_{1N} uses the previous value of cell C_{11} as an explicit boundary condition and then cell C_{11} can take the new value of cell B_{1N} as an updated boundary condition. In this way the speed of updating boundary values can be a little bit improved. However it is not a complete parallel computation but a partially sequential and partially parallel computation as displayed in Fig. 1(c). A processor being in charge of the second subdomain has to idly wait until the value of cell A_{1N} is determined and a processor taking charge of the third subdomain will be idle until the calculation of cell B_{1N} has been complete. As the number of subdomain increases, the part of sequential computation is expected to be dominant resulting in significant reduction of parallel efficiency. It is the reason why the first method corresponding to fully explicit boundary condition has been used in most of parallel simulations of Computational Fluid Dynamic (CFD) including OpenFOAM, Fluent and STAR-CCM+ (Lee, 2013; Choi et al., 2010; Park et al., 2013).

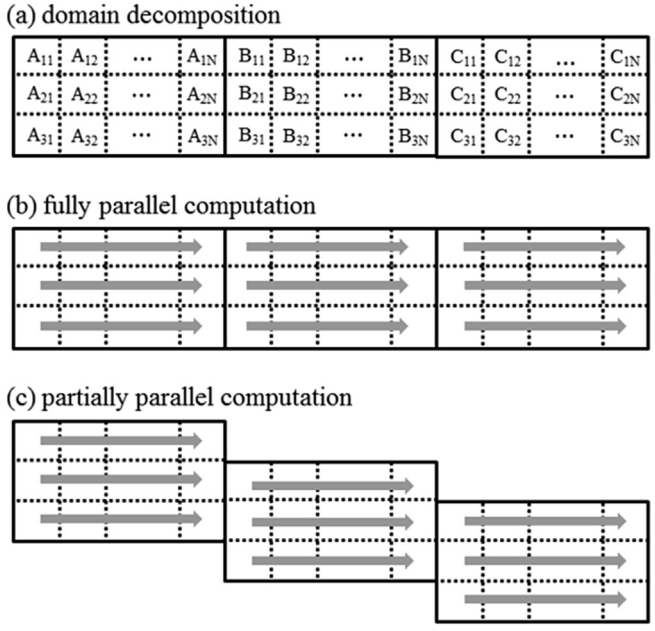

Fig. 1. Domain decomposition for parallel computation using message passing interface.

Now the only concern of MPP is the validation of explicit boundary conditions used on subdomains in the viewpoints of numerical stability and numerical consistency. When some cells with high aspect ratio are located on the boundary of subdomains the parallel computation can give rise to a numerical divergence due to mass imbalance as reported in Park et al. (2013). As every CFD user has experienced at least once, a solution obtained from parallel computation can be a little bit different from that of serial calculation although the boundary and initial conditions are the exactly same. However the numerical discrepancy by parallelization has been considered as one of numerical uncertainty in the process of verification and validation. The main goal of the present work is to investigate the influence of explicit boundary conditions used in subdomain on numerical solutions. To achieve the goal the numerical discrepancy by explicit boundary condition of MPP is introduced in one dimensional Burgers equation which has convective and diffusive terms similar to Navier–Stokes equations. Next two dimensional sloshing patterns will be shown to investigate the numerical discrepancy of Navier–-Stokes equations between serial and parallel computations. Transient and statistical characteristics of numerical solutions obtained from serial and parallel computations will be discussed.

2. Benchmark problems

2.1. One dimensional Burgers equation

Since Burger (1948) proposed an equation with three terms (unsteady, convective and diffusive terms) by removing the pressure gradient term from Navier–Stokes equation, one dimensional Burgers equation has received much attention to investigate the quality of numerical schemes as well as find stochastic phenomena similar to turbulence by adding a random source term (Kutluay et al., 2004; Wani and Thakar, 2013). The canonical Burgers equation consists of unsteady, convective and diffusive terms as follows

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = d\frac{\partial^2 u}{\partial x^2} \tag{1}$$

where d represents the diffusion coefficient which is corresponding to the inverse Reynolds number of Navier–Stokes equation ($d = \mathrm{Re}^{-1}$). The boundary conditions are imposed as $u(0,t) = 1$ and $u(1,t) = 0$ in the present study. 101 grid points are uniformly located between $x = 0$ and $x = 1$ and both