Computers & Geosciences 64 (2014) 96–103

Contents lists available at ScienceDirect

Computers & Geosciences

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

# Parallelisation study of a three-dimensional environmental flow model

Fearghal O'Donncha, Emanuele Ragnoli *, Frank Suits

*IBM Research, Ireland*

ARTICLE INFO

*Article history:*
Received 1 October 2013
Received in revised form
29 November 2013
Accepted 5 December 2013
Available online 17 December 2013

*Keywords:*
MPI
Numerical modelling
Parallel computing
Ocean model

ABSTRACT

There are many simulation codes in the geosciences that are serial and cannot take advantage of the parallel computational resources commonly available today. One model important for our work in coastal ocean current modelling is EFDC, a Fortran 77 code configured for optimal deployment on vector computers. In order to take advantage of our cache-based, blade computing system we restructured EFDC from serial to parallel, thereby allowing us to run existing models more quickly, and to simulate larger and more detailed models that were previously impractical. Since the source code for EFDC is extensive and involves detailed computation, it is important to do such a port in a manner that limits changes to the files, while achieving the desired speedup. We describe a parallelisation strategy involving surgical changes to the source files to minimise error-prone alteration of the underlying computations, while allowing load-balanced domain decomposition for efficient execution on a commodity cluster. The use of conjugate gradient posed particular challenges due to implicit non-local communication posing a hindrance to standard domain partitioning schemes; a number of techniques are discussed to address this in a feasible, computationally efficient manner. The parallel implementation demonstrates good scalability in combination with a novel domain partitioning scheme that specifically handles mixed water/land regions commonly found in coastal simulations. The approach presented here represents a practical methodology to rejuvenate legacy code on a commodity blade cluster with reasonable effort; our solution has direct application to other similar codes in the geosciences.



## 1. Introduction

Numerical modelling has several advantages in the study of coastal ocean flow processes and events. Chief among these is the reduced cost and ease of deployment of a numerical model compared to field work or other methods of investigation. In addition, it is easier to configure a numerical model to investigate different flow conditions and scenarios. However, with the drive to model more realistic and detailed simulations, the computational demands of numerical solutions increase, due primarily to finer grid resolution and the simulation of a greater number of passive and active tracers. As a result, the practical ability of numerical models to solve real-world problems is constrained. Parallel computing allows faster execution and the ability to perform larger, more detailed simulations than is possible with serial code. The research reported here presents details on the porting of an existing coastal ocean model from serial code to parallel. This work is driven partly by a desire to model larger simulations in greater detail, but also to allow experimentation in more computationally demanding methods of data assimilation to improve the performance of real-time predictive modelling.

The model used for the study, Environmental Fluid Dynamics Code (EFDC), is a widely used, three-dimensional, finite difference, hydrodynamic model (Hamrick, 1992). The parallelisation adopts an efficient domain decomposition approach that theoretically permits deployment on a large cluster of machines; however the fundamental objective of our work centres on real-time simulation capabilities of a given model on a commodity blade system and not optimal scalability on an arbitrarily large system. We were therefore guided by the following requirements considered key to the success of the parallelisation effort and subsequent operation on similar cluster systems:

1. Limited changes to the large number of source files (approximately 50 000 lines of code), to avoid introducing computational errors.
2. Binary regression of the parallel model versus serial simulations, to ensure the simulation runs in parallel exactly as it ran serially. Even a small deviation could mask the presence of an error in the port.
3. Automation of the setup process for a parallel run to allow the originally setup serial models to run properly on the parallel code. This involves automatic generation of source code specific

* Corresponding author.
*E-mail address:* eragnoli@ie.ibm.com (E. Ragnoli).

0098-3004/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2013.12.006

to each parallel run of a model, to avoid manual effort and the introduction of errors.

Several parallel versions of numerical ocean models have already been described in the literature, and they have computational methods also used by other codes in the geosciences. Wang et al. (1997) present elements of the widely used Parallel Ocean Program (POP), while Beare and Stevens (1997) build on the parallelisation of the Modular Ocean Model (MOM). However, the fundamental structure of these models makes them more suitable for global, ocean-scale problems, and they are not as well-suited to the finer scale resolution of coastal water phenomena. A parallelisation study on the Princeton Ocean Model (POM) and the Regional Ocean Modelling System (ROMS) is discussed by Sannino et al. (2001) and Wang et al. (2005) respectively. A common feature of these models is the adoption of a split-explicit formulation of the equations governing vertically averaged transport. This representation permits easier parallelisation since global communication in the horizontal is eliminated. However the maximum computational timestep is constrained by the Courant–Friedrich–Levy restriction (Ezer et al., 2002), as opposed to the greater numerical flexibility provided by implicit approaches (Jin et al., 2000). De Marchis et al. (2012) presents details on a parallel code that adopts finite volume methods for the solution of the fundamental governing equations.

Among all branches of the geosciences, atmospheric modelling was one of the first to use parallel computers due to the intrinsic needs of both weather models that run in real-time, and climate models that operate in time scales of centuries. Coupling with ocean models similarly creates computational demands that benefit from parallel computation. Drake et al. (1993) present details on the parallel version of the NCAR Community Climate Model, CCM2. The parallelisation strategy decomposes the model domain into geographical patches with a message passing library conducting communication between segregated domains. Wolters and Cats (1993) describe the parallelisation strategy included in the HIRLAM model, a state-of-the-art system for weather forecasts up to 48 h, while Fournier et al. (2004) discuss aspects of deploying a spectral element atmospheric model in parallel. Michalakes et al. (1998) describe the parallelisation approach adopted for the widely used Weather Research and Forecast model.

In the following sections, the model is introduced along with a description of the computational schemes used to solve the governing equations. Section 3 discusses the parallelisation strategy adopted with particular emphasis on load balancing of the computation within an irregular coastal waterbody. Section 4 presents the parallel speedup and performance of the amended model; a case study analysis focuses on Galway Bay, on the West Coast of Ireland to enable a realistic assessment of practical gain. The conclusions and a discussion are found in section 5.

## 2. Model description

EFDC is a public domain, open source, modelling package for simulating three-dimensional flow, transport and biogeochemical processes in surface water systems. The model is specifically designed to simulate estuaries and subestuarine components (tributaries, marshes, wet and dry littoral margins), and has been applied to a wide range of environmental studies in the Chesapeake Bay region (Shen et al., 1999). It is presently being used by universities, research organisations, governmental agencies, and consulting firms (Ji, 2008).

The equations that form the basis for the EFDC hydrodynamic model are based on the continuity, Reynolds-averaged Navier–Stokes equations. Adopting the Boussinesq approximation for variable density fluid; which states that, density differences are sufficiently small to be neglected except where they appear in terms multiplied by g; the model governing equations can be expressed as

$$\frac{\partial\eta}{\partial t}+\frac{\partial Hu}{\partial x}+\frac{\partial Hv}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{1}$$

$$\frac{\partial Hu}{\partial t}+\frac{\partial(Huu)}{\partial x}+\frac{\partial(Hvu)}{\partial y}+\frac{\partial(wu)}{\partial z}-fHv = -H\frac{\partial(g\eta+p)}{\partial x}-\left(\frac{\partial h}{\partial x}-z\frac{\partial H}{\partial x}\right)\frac{\partial p}{\partial z}+\frac{\partial\left(\frac{A_v}{H}\frac{\partial u}{\partial z}\right)}{\partial z} \tag{2}$$

$$\frac{\partial Hv}{\partial t}+\frac{\partial(Huv)}{\partial x}+\frac{\partial(Hvv)}{\partial y}+\frac{\partial(wv)}{\partial z}+fHu = -H\frac{\partial(g\eta+p)}{\partial y}-\left(\frac{\partial h}{\partial y}-z\frac{\partial H}{\partial y}\right)\frac{\partial p}{\partial z}+\frac{\partial\left(\frac{A_v}{H}\frac{\partial v}{\partial z}\right)}{\partial z} \tag{3}$$

where, $\eta$ is water elevation above or below datum; $u$, $v$ are the velocity components in the curvilinear orthogonal coordinates $x$ and $y$ with $w$ representing the vertical component; $H$ is total water depth ($=h+\eta$, where $h$=water depth below datum); $f$ is Coriolis parameter; $p$ is excess water column hydrostatic pressure; and $\nu_v$ is vertical turbulent viscosity.

The equations governing the dynamics of coastal circulation contain propagation of fast moving external gravity waves and slow moving internal gravity waves. It is desirable in terms of computational economy to separate out vertically integrated equations (external mode) from the vertical structure equations (internal mode) (Blumberg and Mellor, 1987). The external mode associated with barotropic, depth-independent, horizontal long wave motion is solved using a semi-implicit three time level scheme; the external mode computes surface elevation and depth-averaged velocities, $\overline{u}$ and $\overline{v}$. The internal mode, associated with baroclinic, fully three-dimensional, velocity components is solved using a fractional step scheme combining an implicit step for the vertical shear terms with an explicit discretisation for all other terms; the depth-averaged velocities computed in the external mode equations serve as boundary conditions to the computation of the layer integrated velocities. This approach solves the two dimensional depth-averaged momentum equations implicitly in time, hence allowing for the model's barotropic time step to equal the baroclinic time step. The primary limitation of this semi-implicit method is the introduction of an elliptic solver (preconditioned conjugate gradient) to solve implicitly for the free surface elevation solution. This has traditionally posed a problem in the efficient projection of model codes onto parallel computers due to the inherent non-local conditions of the solver (Griffies et al., 2000); this feature is addressed in further detail in the next section.

## 3. Parallelisation

EFDC is a Fortran 77 code originally designed for deployment on vector computers as opposed to distributed systems. The code was configured to achieve a degree of parallelisation on shared memory processors by directives inserted in the source specific to vectorised architectures. However, the existing vectorisation code is not of benefit for parallelisation on distributed memory systems. For performance comparable to vector systems, scalable cache based processors achieve speedup through massively parallel partitioning of the problem among many processors working concurrently (Griffies et al., 2001). The origins of our parallelisation study lie in the rapidly increasing computational requirements of environmental flow solutions, along with the general availability of distributed clusters, particularly blade systems. To enable real-time modelling of large