Computers and Geosciences 110 (2018) 23–31

Contents lists available at ScienceDirect

Computers and Geosciences

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

Research paper

In situ visualization and data analysis for turbidity currents simulation

Jose J. Camata [a,c], Vítor Silva [b,c], Patrick Valduriez [d], Marta Mattoso [b,c], Alvaro L.G.A. Coutinho [a,c,*]

[a] High-Performance Computing Center and Department of Civil Engineering, Brazil
[b] Department of Computer Science, Brazil
[c] COPPE, Federal University of Rio de Janeiro, Brazil
[d] Inria and LIRMM, University of Montpellier, France

ARTICLE INFO

Keywords:
Turbidity currents
In situ visualization
In transit data analysis
Adaptive mesh refinement and coarsening
Parallel computing

ABSTRACT

Turbidity currents are underflows responsible for sediment deposits that generate geological formations of interest for the oil and gas industry. LibMesh-sedimentation is an application built upon the libMesh library to simulate turbidity currents. In this work, we present the integration of libMesh-sedimentation with in situ visualization and in transit data analysis tools. DfAnalyzer is a solution based on provenance data to extract and relate strategic simulation data in transit from multiple data for online queries. We integrate libMesh-sedimentation and ParaView Catalyst to perform in situ data analysis and visualization. We present a parallel performance analysis for two turbidity currents simulations showing that the overhead for both in situ visualization and in transit data analysis is negligible. We show that our tools enable monitoring the sediments appearance at runtime and steer the simulation based on the solver convergence and visual information on the sediment deposits, thus enhancing the analytical power of turbidity currents simulations.

1. Introduction

Turbidity currents are particle-laden underflows where the main driver is turbulence. According to Meiburg and Kneller (2010), turbidity currents Reynolds number in nature is of $\mathcal{O}(10^9)$. Thus particles can be carried for long distances and eventually they will settle, being responsible for sediment deposits that generate geological formations of considerable interest for the oil and gas industry. Sedimentation and erosion promoted by such particle-laden flows can mold the seabed, producing different geological structures like canyons, dunes, and ripples.

Meiburg and Radhakrishnan (Meiburg et al., 2015) review models and computational approaches for modeling gravity and turbidity currents. They vary from simple conceptual models, depth-averaged models, like shallow-water approximations, to more realistic depth-resolved models, based on the three-dimensional Navier-Stokes equations. Possible computational approaches, in this case, involve direct numerical simulation (DNS), large-eddy simulations (LES) and Reynolds averaged Navier-Stokes simulations (RANS). We use an LES finite element approach based on the residual-based variational multiscale (RBVMS) method as described in Guerra et al. (2013). However, to improve the front resolution, we extend the parallel adaptive mesh refinement/coarsening strategy used by Rossa and Coutinho (2013) for simulating three-dimensional lock-exchange configurations to the RBVMS method. Three-dimensional adaptive mesh refinement and coarsening (AMR/C) poses several challenges regarding parallel performance, but according to Burstedde et al. (2010), AMR/C is optimal for tackling large-scale problems governed by partial differential equations. Other software using AMR/C for similar problems are Fluidity-ICOM (Londonothers, 2017) and the Gerris solver (Popinet, 2017).

The standard turbidity current simulation workflow involves the following steps: (i) preprocessing and mesh generation; (ii) time stepping, saving data on disk when required, that is, velocity, pressure, sediment concentrations; and (iii) post-processing, typically visualizing the data generated by the simulation and extracting relevant information on the quantities of interest. When AMR/C is used, mesh data are also saved in step (ii). For large-scale problems, this workflow involves saving a huge amount of raw data in persistent storage.

* Corresponding author. High-Performance Computing Center and Department of Civil Engineering, Federal University of Rio de Janeiro, Brazil.
E-mail addresses: camata@nacad.ufrj.br (J.J. Camata), silva@cos.ufrj.br (V. Silva), patrick.valduriez@inria.fr (P. Valduriez), marta@cos.ufrj.br (M. Mattoso), alvaro@nacad.ufrj.br (A.L.G.A. Coutinho).

https://doi.org/10.1016/j.cageo.2017.09.013
Received 3 May 2017; Received in revised form 14 August 2017; Accepted 25 September 2017
Available online 28 September 2017
0098-3004/

In situ visualization techniques circumvent the storage bottleneck by removing the necessity of first storing data to persistent storage before processing. A recent review discusses the advantages of these techniques (Bauer et al., 2016). Besides savings in persistent storage, we can indeed generate more detailed visualizations, since in situ techniques directly access the memory allocated by the simulation codes. Then, we can, in principle, produce pictures for every time step, which is practically impossible in the standard workflow. We can extend these ideas, using in situ visualization techniques to provide information to help control the simulations. Often, by only observing a deposition pattern, an experienced interpreter can infer that something is not going well in the simulation, deciding to stop it or change parameters, preferably at runtime, resuming the simulation. However, to do that, the visualization should be complemented with information regarding *the evolution* of quantities of interest, such as residual norms, number of linear and nonlinear iterations, often within a specific time window, not just *the current* values. To obtain this complementary information, even the experienced interpreter has difficulty in identifying the files related to the time window, opening and parsing them to obtain specific values and tracking their evolution. The present paper discusses how to integrate in situ visualization with in transit data analysis techniques in large-scale parallel three-dimensional turbidity current simulations.

The rest of this work is organized as follows. Section 2 introduces the governing equations, and the numerical formulation used to simulate turbidity currents using libMesh (Kirk et al., 2006). Section 3 describes our in transit data analysis approach using DfAnalyzer tool (Silva et al., 2017) and in situ data extraction and visualization using ParaView Catalyst (Ayachit et al., 2015). We also discuss in this section how we integrate libMesh with the data analysis tools. Section 4 provides numerical results and a parallel performance evaluation of our solution solving two turbidity current scenarios. The results show that the overhead of in situ visualization and in transit data analysis is negligible, while the added analytical power enables monitoring deposition patterns and steering simulations. The paper ends with a summary of our conclusions.

2. Turbidity currents simulation in libMesh

2.1. Governing equations

This section establishes the mathematical setting for the numerical simulation of turbidity currents within a Eulerian–Eulerian framework. The flows of interest here are mainly driven by small density differences promoted by the heterogeneous presence of sediment particles within the fluid. The double mention to Eulerian is to emphasize that suspended particles, assumed to be present in a dilute proportion in the mixing with a clear fluid, are modeled as a continuum, which motion is governed by an advection dominated transport equation. Here we adopt the simplest three-dimensional depth-resolved model, considering just one sediment granulometry. More complex models can be found in (Necker et al., 2002, 2005; Nasr-Azadani and Meiburg, 2011; Camata et al., 2012; Guerra et al., 2016).

The suspension flow is governed by the incompressible Navier–-Stokes equations considering the Boussinesq approximation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \frac{1}{\sqrt{Gr}}\nabla^2\mathbf{u} + \mathbf{e}^g c \quad \text{in} \quad \Omega\times[0,t_f] \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in} \quad \Omega\times[0,t_f] \tag{2}$$

coupled with the equation for sediment transport

$$\frac{\partial c}{\partial t} + (\mathbf{u} + u_s\mathbf{e}^g)\cdot\nabla c = \frac{1}{Sc\sqrt{Gr}}\nabla^2 c \quad \text{in} \quad \Omega\times[0,t_f] \tag{3}$$

where $\mathbf{u}$, p, c, t, are respectively, non-dimensional velocity, pressure, sediment concentration and time. The dimensionless velocity u_s quantifies the settling velocity of the particles and $\mathbf{e^g}$ is the direction of gravity. *Gr* is the Grashof number, expressing the ratio between buoyancy and viscous effects given by

$$Gr = \left(\frac{u_b H}{\nu}\right)^2 \tag{4}$$

with ν the fluid kinematic viscosity, H a characteristic length of the flow, and u_b the buoyancy velocity. A second dimensionless number is the Schmidt number (*Sc*) that gives the ratio between diffusion and viscous effects,

$$Sc = \frac{\nu}{\kappa} \tag{5}$$

where κ is the diffusivity coefficient.

Essential and natural conditions for Eq. (1) are $\mathbf{u} = \mathbf{g}$ on Γ_g and $\mathbf{n}\cdot\left(-p\mathbf{I} + \frac{1}{\sqrt{Gr}}\nabla\mathbf{u}\right) = \mathbf{h}$ on Γ_h, where $\mathbf{g}$ and $\mathbf{h}$ are given functions, $\mathbf{n}$ is the unit outward normal vector of Γ_h. Γ_g and Γ_h are subsets of the domain boundary Γ. Initial conditions for velocity are chosen to respect the divergence free condition. For Eq. (3), essential conditions may be applied as $c = \overline{c}$ on Γ_{in}. We also apply no-flux boundary conditions at boundary Γ_T by imposing

$$u_s c - \mathbf{n}\cdot\left(\frac{1}{Sc\sqrt{Gr}}\nabla c\right) = 0 \quad \text{on} \quad \Gamma_T \tag{6}$$

This condition ensures that no particle is transported across this boundary. We also assume that particles leave the flow due to sedimentation. This is accomplished by imposing a convective boundary condition at Γ_b (typically the bottom wall).

$$\frac{\partial c}{\partial t} = \mathbf{n}\cdot(u_s\nabla c) \quad \text{on} \quad \Gamma_b \tag{7}$$

with $\Gamma = \Gamma_{in}\cup\Gamma_b\cup\Gamma_T$. No explicit particle resuspension mechanism, allowing particles going back to the flow after hitting the bottom, like erosion, is included. In fact, no significant amount of resuspension is expected for the flow conditions analyzed here (Necker et al., 2005). We compute the deposited particle layer thickness by integrating in time the particle flux through the bottom, that is,

$$D(\mathbf{x},t) = \int_0^t u_s\ c(\mathbf{x},t)d\tau \tag{8}$$

2.2. Weak form of the governing equations

Assuming that the test and weight functions belong to the standard discrete finite element spaces, the weak form for the incompressible Navier–Stokes, based on the Residual-Based Variational Multiscale method (RBVMS) reads

$$\begin{aligned}
&\left(\mathbf{w}^h, \frac{\partial \mathbf{u}^h}{\partial t}\right)_{\Omega^h} + \left(\mathbf{w}^h, \mathbf{u}^h - \tau_M\mathbf{r}_M, \nabla\mathbf{u}^h\right)_{\Omega^h} + \left(q^h, \nabla\cdot\mathbf{u}^h\right)_{\Omega^h} \\
&-\left(\nabla\cdot\mathbf{w}^h, p^h\right)_{\Omega^h} + \left(\nabla^s\mathbf{w}^h, \frac{1}{\sqrt{Gr}}\nabla^s\mathbf{u}^h\right)_{\Omega^h} - \left(\mathbf{w}^h, \mathbf{e}^g c^h\right)_{\Omega^h} \\
&+\left(\mathbf{u}^h\cdot\nabla\mathbf{w}^h\right)_{\Omega^h} \\
&+\left(\nabla q^h, \tau_M\mathbf{r}_M\right)_{\Omega^h} \\
&+\left(\nabla\mathbf{w}^h, \tau_C\nabla\cdot\mathbf{u}^h\right)_{\Omega^h} \\
&-\left(\nabla\mathbf{w}^h, \tau_M\mathbf{r}_M\otimes\tau_M\mathbf{r}_M\right)_{\Omega^h} = 0
\end{aligned} \tag{9}$$

while for the sediment transport equation we have,