Journal of Rock Mechanics and Geotechnical Engineering 6 (2014) 535–545

Contents lists available at ScienceDirect

Journal of Rock Mechanics and Geotechnical Engineering

journal homepage: www.rockgeotech.org

Full length article

Numerical modelling of flow and transport in rough fractures

Scott Briggs*, Bryan W. Karney, Brent E. Sleep

University of Toronto, Toronto M5S 1A4, Canada

ARTICLE INFO

Article history:
Received 30 June 2014
Received in revised form
5 October 2014
Accepted 8 October 2014
Available online 7 November 2014

Keywords:
Hydrogeology
Fracture flow
Solute transport
Computational fluid dynamics
Lattice Boltzmann method (LBM)
Random walk (RW)

ABSTRACT

Simulation of flow and transport through rough walled rock fractures is investigated using the lattice Boltzmann method (LBM) and random walk (RW), respectively. The numerical implementation is developed and validated on general purpose graphic processing units (GPGPUs). Both the LBM and RW method are well suited to parallel implementation on GPGPUs because they require only next-neighbour communication and thus can reduce expenses. The LBM model is an order of magnitude faster on GPGPUs than published results for LBM simulations run on modern CPUs. The fluid model is verified for parallel plate flow, backward facing step and single fracture flow; and the RW model is verified for point-source diffusion, Taylor-Aris dispersion and breakthrough behaviour in a single fracture. Both algorithms place limitations on the discrete displacement of fluid or particle transport per time step to minimise the numerical error that must be considered during implementation.

© 2014 Institute of Rock and Soil Mechanics, Chinese Academy of Sciences. Production and hosting by Elsevier B.V. All rights reserved.

1. Introduction

As computer modelling has evolved, so too have two diametrically opposed requirements of those models in modelling. Certainly, it is desirable to improve a model's computational speed and, yet, it is also desirable to improve the model's accuracy and capability, and attributes that usually require an increase in computational time. Improvements can take the form of enhanced or additional physics or an increase in model resolution to improve accuracy. Often it is a combination of both factors, for example, a new physics model may require higher resolution, or finer meshes to simulate new phenomena. Therefore, one must balance model accuracy and computational time.

To address these computational limitations over the past few decades, more computational power was used, often simply taking advantage of new processor technology. However, single processors, or CPUs, are reaching a performance limit due to manufacturing constraints. Therefore, to continue improving performance, the CPU industry has moved toward using multiple CPUs in parallel. The challenge with this approach then becomes the implementation of conventional numerical algorithms and methods on parallel architectures, including clusters of CPUs and graphics processing units (GPUs). GPUs have evolved over time with more complex computing capabilities, similar to a conventional CPU, and are referred to as general purpose GPUs (GPGPUs).

The lattice Boltzmann method (LBM) is increasingly used for the simulation of fluid flows in complex geometries (Stockman et al., 1998; Eker and Akin, 2006; Yan and Koplik, 2008; Dou et al., 2013). However, its engineering applications have been limited by the required computing power. The local nature of LBM, where only next-neighbour node communication is required, is suitable for parallelization. Previous work has shown that an increase of an order of magnitude in performance can be expected when implementing LBM on a GPGPU (Bailey et al., 2009; Tölke, 2010). However, such work did not show the applicability and validation for flows in rock fractures that are of interest to hydrogeology.

Other computational fluid dynamic (CFD) methods begin with the continuum Navier–Stokes equations governing the macroscopic movement of fluids, and then discretize these equations with a suitable numerical method (Eker and Akin, 2006). In the LBM model, the microscopic interaction of particles on a grid and the averaging of those interactions emerge into the macroscopic continuum of a fluid. These interactions include two main steps: streaming and collision. The streaming step is a translation of particles from one node on the grid to the next. The collision step conserves momentum by redirecting particles that 'collide' or occupy the same node.

This study demonstrates a verified GPGPU code for simulating two-dimensional (2D) laminar flow through rock fractures using a D2Q9 LBM.

Effective understanding of solute transport in fractures is underpinned by the need for accuracy in the simulation of fluid flow. To account for the effects of tortuosity (Tsang, 1984) and Reynolds number above unity, a CFD approach is used. A CFD

* Corresponding author. Tel.: +1 416 978 5969.
E-mail address: scott.briggs@mail.utoronto.ca (S. Briggs).

Peer review under responsibility of Institute of Rock and Soil Mechanics, Chinese Academy of Sciences.
1674-7755 © 2014 Institute of Rock and Soil Mechanics, Chinese Academy of Sciences. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jrmge.2014.10.004

approach, like the LBM, provides local velocities throughout the model domain which are used to move particles through the process of advection. A diffusive process is also included using a random walk (RW) algorithm which is shown to accurately capture the complex geometries associated with single fractures.

In this study, high performance GPGPU numerical methods are developed, validated and shown to be capable of modelling flow and transport in synthetic and real fractures. It is increasingly important for projects of all scales to conduct modelling studies which may require meshes with millions of nodes or large parametric searching, for example, variations in Reynolds number, boundary conditions and fracture geometries. The large meshes, or grids, become computationally and financially expensive on conventional CPUs or clusters. However, a single GPGPU can bridge the gap to high performance computing if the required algorithms are well implemented on the GPGPU with sufficient performance advantages.

2. Model implementation

2.1. Lattice Boltzmann method (LBM)

The LBMs have been used in a variety of engineering applications, including the field of porous and fractured media flow (Sukop et al., 2013). Additional development of LBM theory can be found in the literature (Succi, 2001; Sukop and Thorne, 2006; Latt, 2007). For the purpose of modelling flow in fractures, a 2D LBM code was developed using nine velocity directions e_i, also known as D2Q9. The LBM can be summarised in the following form:

$$f_i(x + e_i\Delta t,\ t + \Delta t) - f_i(x,\ t) = -\frac{1}{\tau}\left[f_i\left(x,\ t\right) - f_i^{\mathrm{eq}}\left(x,\ t\right)\right] \tag{1}$$

where the left hand side of the equation represents the streaming step and the right hand side represents the collision step, and τ is the relaxation parameter that governs the rate at which the fluid tends towards equilibrium. For the LBM model presented, τ takes the following form:

$$\tau = 3\upsilon_{\mathrm{L}} + 1/2 \tag{2}$$

where υ_{L} is the numerical viscosity defined by the discretization of the system into lattice units.

The model runs on a GPGPU using CUDA, a proprietary programming model developed by NVIDIA. One of the drawbacks of GPGPU implementations is the discrepancy between 32-bit and 64-bit floating point precision as current hardwares have limited support for 64-bit, or double precision calculations. Typical GPGPU implementations offer double precision performance that is approximately one third or one quarter that of single precision performance, depending on the model and manufacturer. Without double precision calculations, the numerical error, or the code complexity required to compensate for error, increases.

Another type of numerical error in CFD models, conventionally referred to as numerical dissipation, describes the artificial dissipation of momentum in fluid. Since the LBM is essentially a finite difference approximation to the Boltzmann equation, it is subjected to the same numerical truncations as other finite difference methods. The numerical error can cause dissipation of the advection term which by definition should be free of dissipation (Zhu et al., 2006).

To minimise the potential for numerical instabilities in the LBM and maintain the second order accuracy of the LBM, the model parameters are defined using the method presented by Latt and Krause (2008) as part of the OpenLB User Guide. The process involves selecting physical units then converting to lattice units to finally obtain the relaxation parameter τ. The relaxation parameter plays an important role in the collision term of the LBM. It controls the tendency of the system to move towards local equilibrium. In the literature, the relaxation parameter has been found to cause numerical instabilities when it approaches 0.5 (τ must be greater than 0.5 for physical viscosities). Stable values of τ close to unity are preferred for numerical accuracy of the LBM and can be found using the method outlined below (Sukop and Thorne, 2006; Sukop et al., 2013).

In this study, water is the physical fluid being simulated with a kinematic viscosity, υ, in a fracture of aperture, $2a$, and with physical velocity, u. This leads to an expression for the Reynolds number:

$$Re = \frac{2au}{\upsilon} \tag{3}$$

The dimensionless expression for Reynolds number is then used to convert from the physical units of the system to lattice units. The fracture width is discretized into lattice nodes of length δ_x with discrete time δ_t. In order to minimise the slightly compressible nature of the LBM and maintain the second order accuracy, the following constraints are used respectively when determining system discretization:

$$\delta_t < \frac{\delta_x}{\sqrt{3}} \tag{4}$$

$$\delta_t \sim \delta_x^2 \tag{5}$$

Practically, to ensure stability and numerical accuracy, these constraints are addressed by limiting the numerical velocity to a maximum of 0.1 lattice units per time step, which minimises the partial compressibility of the LBM (Sukop and Thorne, 2006). The lattice viscosity (υ_{L}) is calculated based on the discretization of the system and the dimensionless Reynolds number. Finally, the relaxation parameter is calculated according to Eq. (2) and is kept as close to unity as possible by adjusting the mesh size and maximum lattice velocity.

2.2. Boundary conditions

One of the distinct advantages of the LBM comes from its discrete nature. It is efficient for modelling complex geometries (Chen et al., 1994; Eker and Akin, 2006; Lammers et al., 2006; Brewster, 2007) that arise in the analysis of rock fractures. Within the modelling domain, each node may represent a rock mass or fluid node. At the solid boundaries, a no-slip condition is used to create a zero velocity boundary along the surface. A different set of collision equations are used at the solid boundary and are referred to as mid-plane bounce-back boundary conditions (Succi, 2001). The name arises from the applied boundary rules where particles entering a boundary at time t are sent back out with equal velocity magnitude and opposite direction at time $t + \Delta t$. This effectively puts the boundary at a distance midway between a fluid and solid node.

Constant flux, pressure and gravity-driven boundary conditions can be used to drive the fluid through the fracture. Solid and no-slip boundaries are used along the fracture surfaces while periodic boundary conditions are used at entry and exit of the fracture where fluid and solutes leaving the fracture are re-injected with equivalent velocity and direction. Periodic boundary conditions simulate an infinite domain with periodically repeated geometry. A periodic, or wrapped boundary, in combination with applied gravity boundary conditions, removes entry or exit effects which would otherwise arise under conventional constant flux boundaries.

Download English Version:

https://daneshyari.com/en/article/286707

Download Persian Version:

https://daneshyari.com/article/286707

Daneshyari.com