Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIII, págs. 2441-2441 (resumen)
Graciela Bertolino, Mariano Cantero, Mario Storti y Federico Teruel (Eds.)
San Carlos de Bariloche, 23-26 Setiembre 2014

IMPLEMENTACIÓN EN GPU DEL MÉTODO COMBINADO DE LATTICE BOLTZMANN Y FRONTERA INMERSA PARA LA SIMULACIÓN DE FLUIDOS

Pablo R. Rinaldi[a,b,c], **Javier Dottori**[a,c,d], **Diego Dalponte**[a,b,c] **and Gustavo Boroni**[a,c,d]

[a]*Universidad Nacional del Centro de la Provincia de Buenos Aires – UNCPBA, Pinto 399 Tandil, Argentina, prinaldi@exa.unicen.edu.ar, http://www.unicen.edu.ar*

[b]*Comisión de Investigaciones Científicas de la Pcia. De Buenos Aires - CICPBA, Calle 526 entre 10 y 11 La Plata, Argentina, http://www.cic.gba.gov.ar*

[c]*Instituto Pladema, Paraje Arroyo Seco S/N Tandil, Argentina, http://www.plademа.net*

[d]*Consejo Nacional de Investigaciones Científicas y Técnicas - CONICET, Av. Rivadavia 1917 Ciudad Autónoma de Buenos Aires, Argentina, http://www.conicet.gov.ar*

Abstract. Se presenta una implementación para GPU del método de redes de Boltzmann o Lattice Boltzmann (LBM) en combinación con el método de Frontera Inmersa (IB). LBM ha demostrado un gran potencial en la simulación de fluidos pese a su simplicidad. Al tratarse de un autómata celular con un esquema explícito que se ejecuta sobre grillas regulares logró beneficiarse en gran forma del potencial de las GPUs. Numerosos trabajos muestran implementaciones que logran una performance órdenes de magnitud superior a un código similar en CPU. Pero el inconveniente principal de LBM es la representación de condiciones de contorno, ya sea de los límites del dominio o de las paredes de los objetos inmersos en el fluido. Trabajos recientes acoplan a LBM el método de Frontera Inmersa que permite simular correctamente condiciones de contorno curvas o incluso flexibles sin requerir un excesivo refinamiento de la grilla. IB representa las paredes como sucesiones de puntos que reaccionan e interactúan con el fluido mediante fuerzas. Desafortunadamente la naturaleza implícita del algoritmo de IB junto con la distribución espacial irregular de los puntos de frontera plantea un desafío al querer lograr una implementación eficiente sobre GPU. Muchas técnicas utilizadas en códigos LBM puros implementados en GPU no son aplicables al combinarlo con IB o, si lo son, dejan de ser eficientes. Lo mismo pasa al intentar traducir a lenguaje de GPU las implementaciones para CPU de LBM-IB. En este trabajo se logró una implementación eficiente, que explota el potencial paralelo de las GPU de última generación para ejecutar alternativamente los dos métodos, utilizando configuraciones de ejecución y estructuras de datos específicamente diseñados para acelerar la ejecución de código.