DOI: https://doi.org/10.33414/ajea.1061.2022

Modelos Numéricos en GPGPU para el tratamiento de fondos móviles erosionables

Numerical Models in GPGPU for the treatment of erodible beds

Presentación: 4 y 5 de Octubre de 2022

Doctorando:
Lucas C. Bessone Martínez
Facultad Regional Concordia, Universidad Tecnológica Nacional - Argentina
lcbessone@gmail.com

Director:
Mario Storti

Codirector:
Pablo Gamazo

Resumen

En este trabajo se emplean algoritmos y simulaciones numéricas de flujos fluidos tridimensionales acoplados con el transporte de sedimentos para reproducir los procesos de erosión y sedimentación. Se implementa un resolvedor para las ecuaciones de NS (Navier Stokes) en dominios con fondos móviles. Se utiliza el algoritmo FS (Fractional Step) para resolver el fluido incompresible. El cambio en la morfología del lecho de sedimentos se modela usando la ecuación de Exner, la cual se discretiza y acopla con el modelo discreto del fluido. Para el transporte de sedimentos se considera solamente la modalidad de gasto sólido de fondo. La implementación se realiza usando técnicas de cálculo en paralelo en unidades de procesamiento gráfico para propósitos generales (GPGPU).

Palabras clave: GPGPU, Flujos incompresibles, Método de los volúmenes finitos, Navier Stokes, HPC, Exner.

Abstract

In this work, algorithms and numerical simulations of three-dimensional fluid flows coupled with sediment transport are used to reproduce the processes of erosion and sedimentation. A solver for the NS (Navier Stokes) equations in domains with moving bed is implemented. The FS (Fractional Step) algorithm is used to solve the incompressible fluid. The change in sediment bed morphology is modeled using the Exner equation, which is discretized and coupled with the discrete fluid model. For sediment, only bed load transport is considered. The implementation is done using parallel computation techniques in general purpose graphics processing units (GPGPU).

Keywords: GPGPU, Incompressible fluid flows, Finite volume method, Navier-Stokes equations, HPC, Exner.

Introducción

La descripción matemática del modo en cómo se transportan las partículas sólidas en una corriente líquida es sumamente compleja. El transporte sólido en los ríos es de difícil determinación porque se tiene gran variabilidad en los fenómenos tanto en el espacio como en el tiempo, un elevado número de variables intervinientes y dificultad de comprobar en la naturaleza los resultados que se obtienen. Entre los problemas vinculados al transporte de sedimentos se encuentra la sedimentación y erosión localizada alrededor de pilas y estribos de puentes. Los daños vinculados a estos casos, se pueden reducir en la medida que se pueda predecir el transporte sólido. Debido al complejo carácter tridimensional del fenómeno de erosión localizada y las formas en el fondo del lecho del río, es necesario el uso de modelos de flujo que sean tridimensionales.

Materiales y Métodos

Con el paso de los años los cambios en hardware han permitido el desarrollo de modelos con mayor resolución y complejidad. En la última década las GPUs han surgido como una alternativa para la computación de propósito general y su uso para el cálculo científico ha ido aumentando (Che et al., 2008; Ujaldón, 2016). Para obtener un código GPU eficiente, los algoritmos deben diseñarse de acuerdo a las características específicas del hardware. El modelo de ejecución adoptado por las GPUs para el cómputo en paralelo es el de Simple Instrucción Múltiples Datos (SIMD). Las arquitecturas GPU se caracterizan por la abundante capacidad de cómputo en relación al ancho de banda de memoria (Loppi et al., 2018). Esto las hace muy buenas para resolver discretizaciones temporalmente explícitas y espacialmente compactas. Por otro lado, se debe tener en cuenta los aspectos del modelo de programación CUDA (Computed Unified Device Architecture) (Lindholm et al., 2008) en los métodos utilizados para resolver las ecuaciones discretizadas.

A. Estudio de solvers para flujos incompresibles

Comenzando con la ecuación general de transporte, se implementó un resolvedor basado completamente en GPU del tipo temporal explícito y otro implícito, usando el Método de los Volúmenes Finitos (FVM) en grillas cartesianas uniformes, guardando las variables en forma colocada. Para el término difusivo se utilizan diferencias centradas, mientras que para el término advectivo se dispone tanto de esquemas de bajo orden (upwind) como de alto orden (HO) y de alta resolución (HR), por ejemplo QUICK, MINMOD, MUSCL entre otros. Para resolver las ecuaciones de flujo incompresible se implementó el método de Pasos Fraccionados (FS). Para la solución de los sistemas lineales que surgen de los esquemas implícitos, se utilizan los métodos de Gradientes Conjugados (CG) para los sistemas simétricos combinado con precondicionador Multigrilla y Gradientes Biconjugados Estabilizado (BICGSTAB) para los sistemas no simétricos.

B. Tratamiento de obstáculos sólidos usando enfoques del tipo IBM

Para mantenerse alineado con el modelo de programación CUDA se deben usar grillas eulerianas estructuradas con el fin de evitar las lecturas de una tabla de conectividades. Por tal motivo, se implementó un Método de Fronteras Embebidas (IBM) utilizando celdas fantasmas (GC), siguiendo un enfoque de forzamiento directo, que permite lograr una precisión de segundo orden en un esquema espacial relativamente compacto (Fadlun et al., 2000; Mittal y Iaccarino, 2005; Costarelli et al., 2016). La superficie del lecho y los obstáculos sólidos se representan mediante técnicas del tipo Level Set (LS).

C. Tratamiento de la evolución de la superficie del lecho, métodos tipo level set combinados con IBM, solución de la ecuación de Exner

La evolución de la frontera inferior del dominio se determina a partir de la ecuación de Exner (Exner, 1925). La estrategia que se usa es la de mantener la malla fija del dominio total, la función de altura del lecho $\xi(x, y)$ identifica los puntos (x, y, ξ), lo que permite calcular una función curva de nivel (LS) como la distancia entre los puntos del dominio y los ubicados en la superficie del lecho. Se busca implementar una paralelización eficiente para lo cual se aproxima la función LS a partir de una estimación inicial dada por ξ y utilizando métodos de reinicialización. Luego se puede aplicar correctamente la condición de borde en el fondo usando el enfoque tipo IBM.

El conjunto de ecuaciones se resuelven usando una estrategia de acople particionado, es decir, en cada paso temporal, se resuelve primero el campo de velocidades para un dominio espacial fijo, se determinan los esfuerzos de corte, se evalúan las descargas de sedimento, luego se resuelve la ecuación de Exner y finalmente se actualiza el dominio físico para avanzar al siguiente paso de tiempo. A su vez, al modelo de flujo incompresible se incorporan modelos de turbulencia del tipo LES explícito o el modelo tipo RANS, $k - \omega\ SST$ (Menter, 1994).

D. Detalles de la implementación

Los algoritmos implementados se ejecutaron en un equipo Del PowerdEdge R740 con procesador Intel Xeon Gold CPU 6138 con 128GB DDR4 RAM provisto de una tarjeta de video NVIDIA Tesla V100 GPU con 32GB de memoria HBM2. El sistema operativo es CentOS Linux v7, el compilador GCC 4.8.5 y la versión 9.1 para el compilador NVCC de CUDA. Todas las implementaciones se realizaron en doble precisión (DP) en un código propio basado completamente en GPU. Todas las operaciones de álgebra lineal fueron implementadas con ayuda de la librería CUBLAS.

Resultados

En la Figura 1 se presenta a modo de ejemplo, los resultados obtenidos para el flujo alrededor de una pila circular en régimen turbulento, en las mismas, pueden apreciarse los campos de energía cinética y la tasa de disipación turbulenta. En la Figura 2 se muestran los remolinos mediante dos técnicas de visualización de estructuras de vórtices.

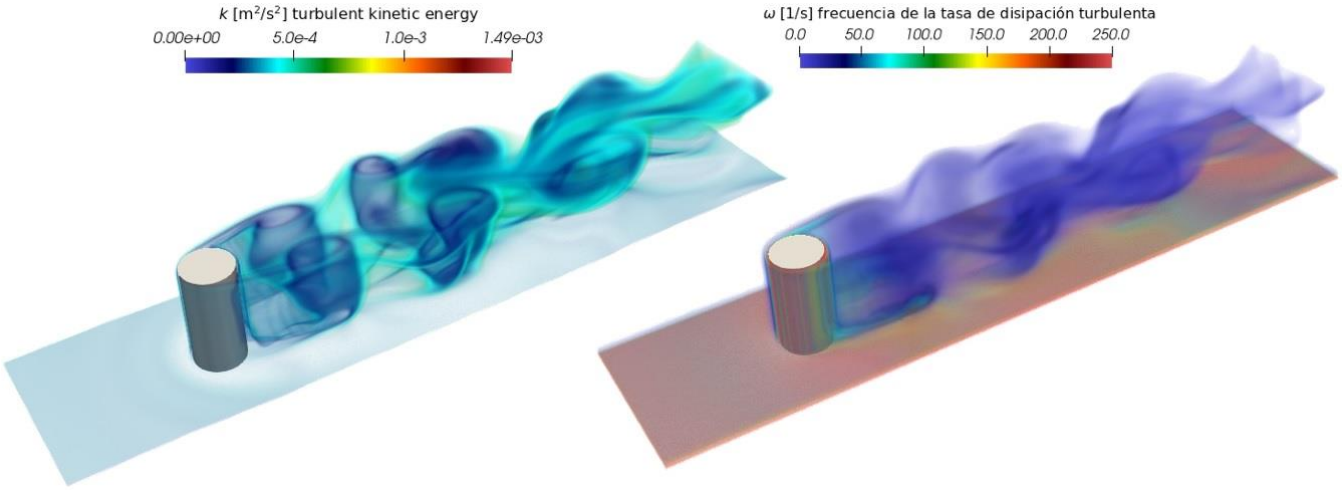

Figura 1: Visualización de los dos escalares transportados en el modelo $k-\omega\ SST$ en un instante de simulación. Energía cinética k (izquierda) y razón de disipación turbulenta ω (derecha).

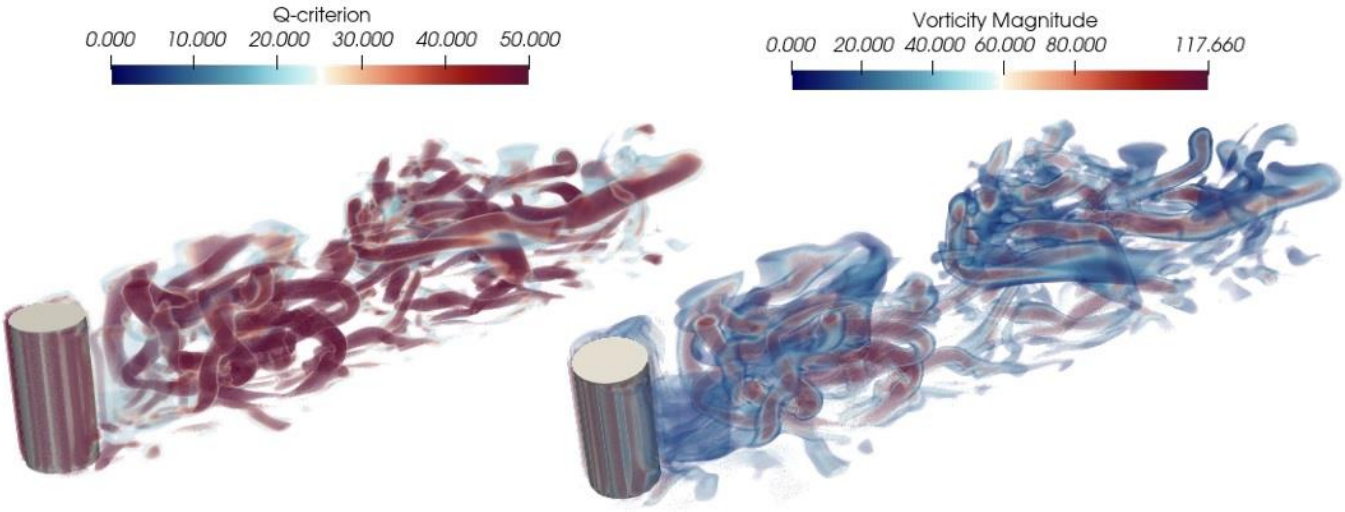

Figura 2: Visualización de las estructuras de vórtices utilizando el Criterio Q y el módulo del campo de vorticidad.

Se validó el código para el caso de erosión alrededor de un obstáculo rectangular a $Re = 1000$. En la figura 3 puede observarse que se logra obtener con buena aproximación el patrón de erosión – sedimentación alrededor del obstáculo, obtenidas del experimento físico de Burkow (2010). En la Figura 4 se puede apreciar cualitativamente las complejas estructuras de vórtices teóricas que se espera obtener para este caso.

Finalmente en la Figura 5 se compara la profundidad de erosión delante del objeto (curva azul) cuyos valores resultan más importantes para aplicaciones de ingeniería fluvial. Puede observarse que los resultados concuerdan con las simulaciones de Burkow y Griebel (2016). Por otra parte, la altura de sedimentación (curva roja) difiere levemente, respecto a esto, debe tenerse en cuenta que ambas simulaciones comparadas, se realizaron con diferentes condiciones de contorno, diferentes metodologías y algoritmos con los que el resultado obtenido se considera suficientemente aceptable.

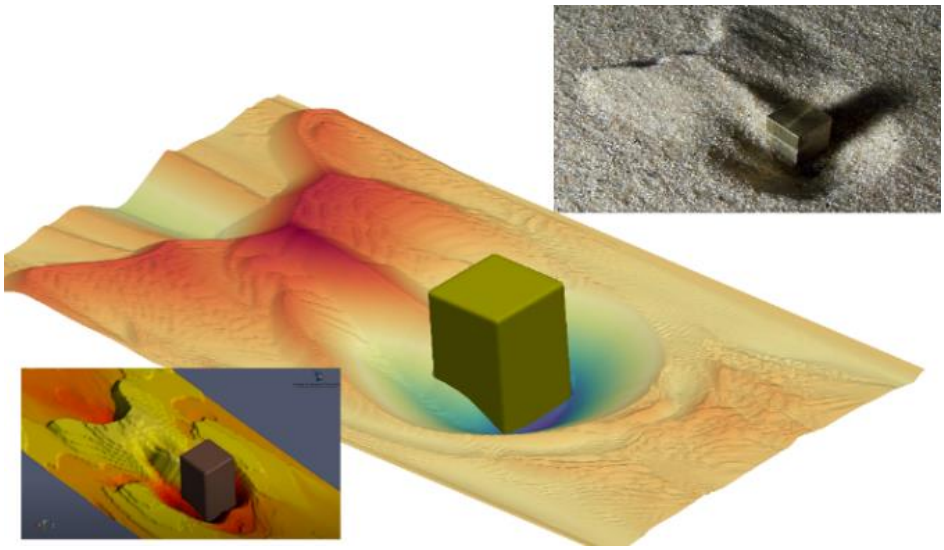

Figura 3: Comparación de la erosión alrededor de un obstáculo rectangular a $Re = 1000$, luego de 500 segundos. Resultados numéricos extraídos de Burkow y Gribel (2016) (izquierda-abajo), resultados de este trabajo (centro), resultados experimento físico extraído de Burkow (2010) (derecha-arriba).

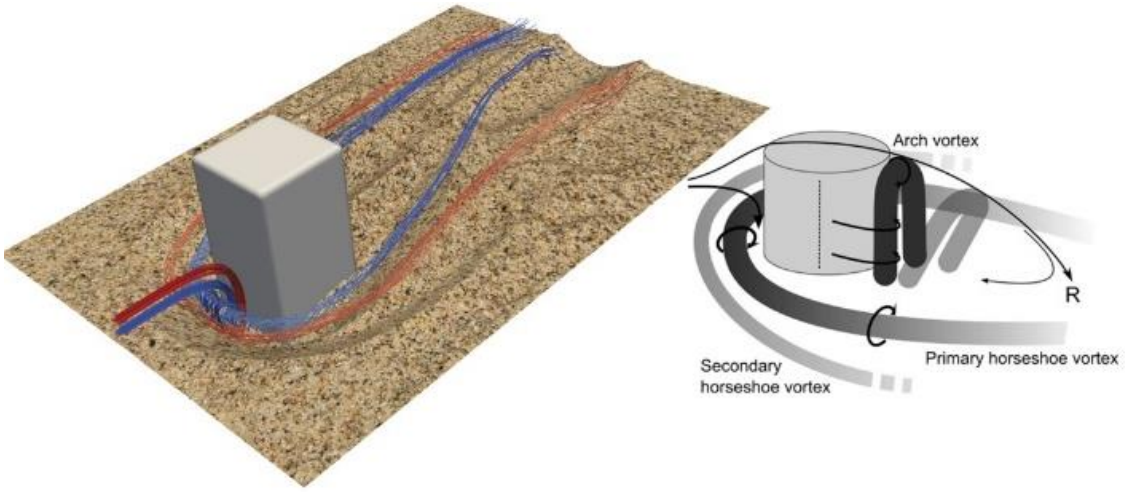

Figura 4: Captura de los vórtices de herradura (primario y secundario) obtenidos en la simulación numérica de este trabajo (izquierda). Patrones de flujo tridimensional idealizados alrededor del objeto, figura extraída de Euler y Herget (2012) (derecha).

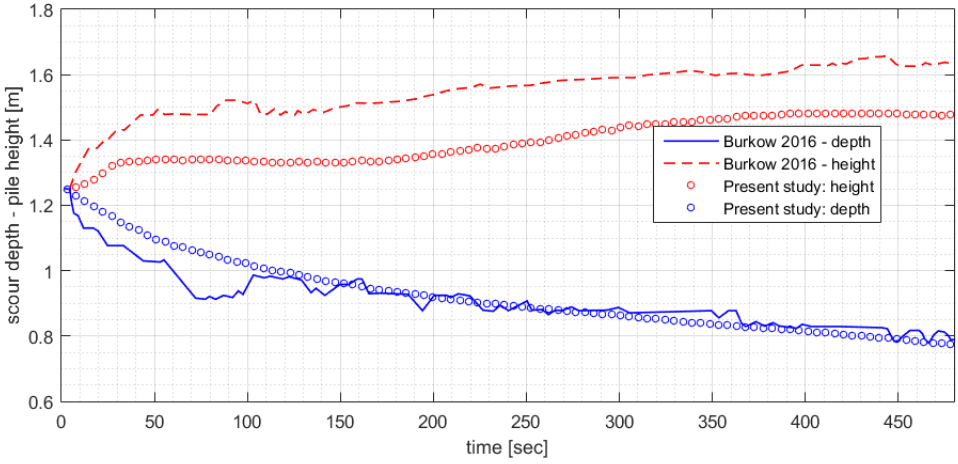

Figura 5: Validación de resultados: evolución temporal de la profundidad de socavación delante del obstáculo y la altura máxima de la duna depositaba tras el obstáculo. Comparación con el trabajo presentado en Burkow y Griebel (2016).

Los tiempos de cómputo para la simulación de erosión durante 500s físicos, fue del orden de 10h, utilizando una sola tarjeta de video, mientras que el tiempo reportado en Burkow y Griebel (2016), utilizando el software NaSt3D paralelizado en CPU, fue de 180h de cómputo, utilizando 2 nodos de un clúster con un total de 64 núcleos. Esto muestra que se requieren cerca de 36 nodos de similares características, para igualar el tiempo de cómputo de una tarjeta.

Conclusiones

Se implementaros algoritmos eficientes en GPU de acuerdo a las características específicas del hardware, para resolver el flujo y el transporte de sedimentos, que permiten predecir la socavación local alrededor de obstáculos inmersos en una corriente líquida.

Los resultados obtenidos fueron validados con los trabajos de otros autores. Se muestra entonces que la GPU surge como una alternativa para la computación de alto desempeño, capaz de realizar simulaciones numéricas aplicadas a la hidráulica fluvial con tiempos de cómputo reducidos en comparación al uso de un clúster de CPUs.

Agradecimientos

Universidad Tecnológica Nacional, Facultad Regional Concordia ("Becas de formación de doctores para fortalecer las áreas de I+D+i", Res. 1460/15).

Referencias

Burkow, M. (2010). Numerische simulation stromungsbedingten sedimenttransports und der entstehenden gerinnebettformen. Master's thesis, Mathematisch-Naturwissenschaftlichen Fakultat der Rheinischen Friedrich-Wilhelms-Universitat Bonn, Bonn, Germany.

Burkow, M., & Griebel, M. (2016). A full three dimensional numerical simulation of sediment transport and the scouring at a rectangular obstacle. Computers & Fluids, 125, 1-10. https://doi.org/10.1016/j.compfluid.2015.10.014

Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J. W., & Skadron, K. (2008). A performance study of general-purpose applications on graphics processors using CUDA. *Journal of parallel and distributed computing*, *68*(10), 1370-1380. https://doi.org/10.1016/j.jpdc.2008.05.014

Costarelli, S. D., Garelli, L., Cruchaga, M. A., Storti, M. A., Ausensi, R., & Idelsohn, S. R. (2016). An embedded strategy for the analysis of fluid structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, *300*, 106-128. https://doi.org/10.1016/j.cma.2015.11.001

Euler, T., & Herget, J. (2012). Controls on local scour and deposition induced by obstacles in fluvial environments. *Catena*, *91*, 35-46. https://doi.org/10.1016/j.catena.2010.11.002

Exner F. M. (1925). Über die wechselwirkung zwischen wasser und geschiebe in flüssen. *Sitzungsberichte. Abt. 2a Mathematik Astronomie Physik Und Meteorologie (Akademie Der Wissenschaften in Wien Mathematisch-Naturwissenschaftliche Klasse) 134. Bd. (jahr. 1925)*.

Fadlun, E. A., Verzicco, R., Orlandi, P., & Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of computational physics*, *161*(1), 35-60. https://doi.org/10.1006/jcph.2000.6484

Lindholm, E., Nickolls, J., Oberman, S., & Montrym, J. (2008). NVIDIA Tesla: A unified graphics and computing architecture. *IEEE micro*, *28*(2), 39-55. https://doi.org/10.1109/MM.2008.31

Loppi, N. A., Witherden, F. D., Jameson, A., & Vincent, P. E. (2018). A high-order cross-platform incompressible Navier–Stokes solver via artificial compressibility with application to a turbulent jet. *Computer Physics Communications*, *233*, 193-205. https://doi.org/10.1016/j.cpc.2018.06.016

Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, *32*(8), 1598-1605. https://doi.org/10.2514/3.12149

Mittal, R., & Iaccarino, G. (2005). Immersed boundary methods. *Annu. Rev. Fluid Mech.*, *37*, 239-261. https://doi.org/10.1146/annurev.fluid.37.061903.175743

Ujaldón, M. (2016, July). CUDA achievements and GPU challenges ahead. In *International Conference on Articulated Motion and Deformable Objects* (pp. 207-217). Springer, Cham. https://doi.org/10.1007/978-3-319-41778-3_20