Aceleración de la Simulación de Dinámicas Poblacionales en el Neolítico Europeo

Tamara Carrillo [1], Pascual Pérez [2], Ignacio García-Fernández [3], Miguel Lozano [4], Juan Manuel Orduña [5]

Resumen— **El periodo neolítico está caracterizado por una expansión gradual de la agricultura en Europa. Aunque las causas precisas de esta migración no están claras actualmente, diversos autores han propuesto modelos dinámicos que permiten estudiar la dispersión de la población humana de la era del Neolítico por medio de simulación por computador.**

Este trabajo presenta un estudio comparativo de diferentes implementaciones para simulación por computador de un modelo de migración de la población humana del Neolítico. Este modelo de reacción-difusión se basa en ecuaciones en derivadas parciales y tiene en cuenta el efecto de las particularidades geográficas de cada región para modular el proceso. En el estudio se han considerado la implementación secuencial clásica basada en CPUs actuales, implementaciones basadas en un programa comercial de cálculo numérico (usando librerías específicas para GPU) y una implementación ad-hoc básica no optimizada basada en el uso de GPUs. Los resultados demuestran que la implementación ad-hoc realizada sobre GPUs obtiene una ganancia significativa de prestaciones sobre el resto de implementaciones paralelas para tamaños de mediana escala del problema, aunque la versión basada en el programa comercial obtiene resultados ligeramente mejores a gran escala (todo el continente europeo). Estos resultados muestran que cualquier implementación ad-hoc sobre GPUs puede acelerar significativamente la simulación de estas dinámicas poblacionales.

Palabras clave— **Simulación numérica, Aceleración de aplicaciones, Computación sobre GPUs.**

I. Introducción

El origen del Neolítico y el mecanismo de expansión de la agricultura en Europa continúan siendo temas de actualidad en la arqueología [1]. Con independencia de cuáles fueran las causas que la provocaran, dicha transición (Mesolítico- Neolítico) puede modelarse mediante modelos de difusión que incorporen los factores fundamentales que rigen este tipo de migraciones. La combinación de estos modelos matemáticos junto con las actuales capacidades de cálculo permiten realizar simulaciones para validar diferentes hipótesis frente a los datos arqueológicos disponibles [2], [3], [4]. La validación de las hipótesis que fundamentan cada modelo es posible gracias a la comparación del frente de onda en la simulación, la región en la que la población aparece por primera vez en un momento determinado, y la información contenida en las bases de datos que albergan muestras de radiocarbono (^{14}C) datadas en las épocas objeto de estudio. Gracias a esta información se pueden calcular tasas de error del modelo y determinar así la verosimilitud del mismo.

Entre los modelos de migración que podemos encontrar en la bibliografía aparece el propuesto por Davison et al [5]. Asumiendo la hipótesis de que la expansión de la agricultura en el neolítico se debe a un proceso de expansion démica, el modelo entiende esta migración como un proceso de difusión aplicado al valor de densidad de población. De este modo, dado un punto inicial de partida con densidad no nula, se asume que la población se dispersa con una variación en el tiempo que depende de las derivadas de segundo orden de la densidad de población respecto a la posición geográfica. Este modelo numérico goza actualmente de gran reconocimiento [6], [7], debido a la introducción de factores que influyen en la propagación del frente de onda, como la altura del terreno o la advección proporcionada por los ríos más significativos y la linea de costa, al actuar como impedimentos o transmisores de la migración. La combinación de estos factores da lugar a un comportamiento anisotrópico, ya que la tendencia a moverse en una dirección se ve alterada por el entorno o la facilidad de viajar en una dirección determinada. En la Sección III se describe en detalle el modelo matemático empleado.

Para poder emplear este tipo de simulaciones como herramienta básica de validación de hipótesis y estudiar el modelo en profundidad, se hace necesario disponer de un buen número de simulaciones con diferentes valores de los parámetros. Sin embargo, el tiempo de simulación necesario para obtener los datos de validación puede llegar a ser demasiado grande si la zona de interés es extensa, como suele ser habitual.

El presente trabajo tiene como objetivo la aceleración del problema de simulación planteado por Davison, donde la zona de interés se encuentra entre los 75º N y los 25º N de latitud, y entre los 60º E y los 15º W de longitud. Para esta región se ha extraído el correspondiente mapa de alturas a una resolución aproximada de 1 dato de altitud por cada km lineal, similar a la empleada por la propuesta original [5]. Con esta resolución, el total de puntos a simular supera los 60 millones y en todos ellos debe resolverse numéricamente el modelo matemático propuesto.

En este modelo de simulación la cantidad de ci-

[1]Dpt. Informática Univ. Valencia, e-mail: tacama2@alumni.uv.es
[2]Dpt. Informática Univ. Valencia, e-mail: paspeca@alumni.uv.es
[3]Dpt. Informática Univ. Valencia, e-mail: ignacio.garcia@uv.es
[4]Dpt. Informática Univ. Valencia, e-mail: miguel.lozano@uv.es
[5]Dpt. Informática Univ. Valencia, e-mail: juan.orduna@uv.es

clos necesarios hasta la obtención de una solución numérica depende directamente del intervalo de tiempo que se desea simular y del paso de integración temporal empleado en el método de integración numérica elegido. El periodo de tiempo contemplado en [5] está comprendido entre el año 7000AC y el 3000AC. Con el fin de garantizar la estabilidad numérica del método empleado, este paso de integración debe ser pequeño (con una cota que dependerá de los parámetros del modelo), lo que supone que será necesario un mayor número de ciclos a computar y por tanto un mayor coste computacional para cada simulación que se ejecute. Este escenario motiva la integración de técnicas de computación paralela que permitan acelerar la simulación del modelo. El trabajo que se presenta aquí se centra en el análisis de la mejora de prestaciones que se puede obtener mediante la ejecución del modelo de simulación en una arquitectura paralela basada en tarjetas gráficas (GPU). Para ello, y con el objetivo de estudiar el peor de los casos, hemos desarrollado una implementación ad-hoc básica no optimizada (sin preprocesamiento de datos, ni optimización en el algoritmo utilizado para la multiplicación de matrices, etc.) sobre tarjeta gráfica. Los resultados demuestran que la implementación ad-hoc básica sobre tarjeta gráfica obtiene una ganancia significativa de prestaciones sobre el resto de implementaciones paralelas para instancias pequeñas y medianas del problema, aunque para el mayor tamaño del problema (todo el continente europeo) la versión para tarjetas gráficas basada en programa comercial obtiene prestaciones ligeramente superiores. Estos resultados muestran que cualquier implementación ad-hoc sobre GPUs puede acelerar significativamente la simulación estas dinámicas poblacionales.

El resto del articulo se ha organizado de la siguiente manera: la sección II revisa la literatura de las distintas vertientes asociadas al problema planteado. Posteriormente, la sección III revisa las características principales del modelo implementado. Las secciones IV y V muestran la metodología empleada y la evaluación de prestaciones llevada a cabo de forma experimental. Finalmente, en la sección VI se exponen las principales conclusiones derivadas del presente trabajo.

II. Estado de la tecnología

El estudio de un proceso complejo como la expansion de la agricultura en el neolítico requiere de conocimientos provenientes de campos científicos distantes. En el caso que nos ocupa la investigación se centra, por una parte, en la selección de muestras de radiocarobono y, por otra parte, en el modelado del proceso y el cálculo de los errores del modelo diseñado.

Para la selección de muestras es frecuente recurrir a tablas de datos recopilados por diversos autores. En nuestro caso, usamos los datos empleados por Davison et al. [5], quienes a su vez los recopilaron a partir de varios trabajos centrados en diversas regiones: Europa central, oeste y sur [8], [9], [10], Europa del este [11] [12]. La selección de las muestras que van a emplearse para la validación depende de distintos factores como su localización geográfica, pero sin duda, un factor clave es el error mínimo de calibración de las mismas, σ, que estima la precisión en la datación de la muestras de radiocarbono, y que viene expresado en años (p.ej. $\sigma = 160$ años). De acuerdo con [12], parece interesante homegeneizar las muestras en torno a este parámetro y, en este sentido, se buscan grupos o *clusters* de muestras que tengan una desviación pequeña en σ.

Los modelos empleados en el estudio de la difusión del neolítico en Europa plantean un problema todavía en discusión. Por un lado, la hipótesis principal consiste en una migración directa de la población agrícola, dando lugar a la idea de expansión démica y a los modelos basados en la velocidad de avance del frente de onda [13]. Por otro lado, existen hipótesis alternativas [14] donde se propone la neolitización como una adopción de las técnicas agrícolas por las comunidades indígenas de cazadores-recolectores y la transmisión de cultura.

Por otra parte, los modelos físicos de advección-reacción-difusión empleados, han sido ampliamente utilizados en distintas disciplinas científicas para el modelado de procesos de transmisión de naturaleza expansiva. Por ejemplo, existen estudios atmosféricos [15], médicos [16], químicos [17].

Más recientemente han aparecido un conjunto de trabajos centrados en la aceleración del proceso de simulación del modelo de difusión correspondiente. La computación basada en GPUs [18] ha sufrido un espectacular aumento de interés debido a la buena relación entre el coste y la capacidad de paralelización que ofrece para un gran número de aplicaciones, entre las que se encuentra la simulación. Por ejemplo, en [19] se presenta un marco de trabajo general para la exploración de soluciones numéricas usando GPUs. La simulación de fluidos y partículas son problemas clásicos abordados recientemente con GPUs [20]. Por último, la simulación de muchedumbres también utiliza modelos de movimiento equivalentes que han sido acelerados convenientemente con GPUs para reducir el tiempo de simulación [21].

Por todo ello, parece conveniente intentar la aplicación de la computación basada en tarjetas gráficas para intentar acelerar la simulación por computador del modelo de difusión de la población del Neolítico propuesto por Davison.

III. Modelo de difusión

Esta sección presenta la ecuación en derivadas parciales que describe el modelo propuesto por Davidson et al. [22], discutiendo la influencia de cada término en el comportamiento del sistema.

La función que se quiere predecir es la densidad de población, N, expresada en habitantes por km^2 en una región de la superficie terrestre correspondiente a Europa y Oriente Medio a lo largo de un periodo de tiempo de unos 4000 años. Esta función depende

de tres variables:

1. θ es la co-latitud, en grados. Es decir, $\theta = 90^o - \delta$, donde δ es la latitud geográfica del lugar.
2. ϕ es la longitud geográfica, en grados.
3. t es el tiempo, medido en años

De acuerdo con el modelo propuesto por Davidson et al. [22], [5], la ecuación que determina la evolución de la densidad de población a lo largo del tiempo viene dada por

$$\frac{\partial N}{\partial t} + (V \cdot \nabla)N = \gamma N \left(1 - \frac{N}{K}\right) + \nabla \cdot (\nu \nabla N). \quad (1)$$

donde N es la densidad de población, V es la velocidad de advección, γ es la tasa de crecimiento de la población y ν es la difusividad. A continuación se describe con mas detalle el significado cada uno de estos parámetros.

A. Descripción del modelo

Para facilitar la discusión de los diferentes términos de la ecuación despejamos, en primer lugar, el término de la derivada parcial de $N(\theta, \phi, t)$ respecto al tiempo y agrupamos el resto de términos a la derecha de la ecuación, obteniendo:

$$\frac{\partial N}{\partial t} = \gamma N \left(1 - \frac{N}{K}\right) + \nabla \cdot (\nu \nabla N) - (V \cdot \nabla)N. \quad (2)$$

La ecuación (2) indica que la tasa neta de crecimiento de la población en cada punto de la región estudiada viene determinada por el valor de tres funciones que modelan la influencia de factores como el crecimiento demográfico local, la dispersión de la población o la facilidad que ofrece la orografía para desplazarse en determinadas direcciones. Los parámetros K, V, ν y γ son conocidos de antemano y no evolucionan con el tiempo, aunque pueden tener distintos valores en regiones diferentes del dominio para modelar las propiedades locales de cada punto. Una vez fijados estos parámetros, para determinar el valor de la tasa de crecimiento únicamente es necesario conocer el valor de la propia densidad de población, así como el de sus derivadas respecto a las variables espaciales.

El primer término de la parte derecha en (2) es el modelo conocido como *ecuación logística*, y representa la tasa de crecimiento demográfico local sin tener en cuenta las migraciones. La *capacidad de carga*, K, dada en habitantes por km^2, determina la densidad de población maxima de acuerdo con los recursos existentes en el entorno, y la tasa de crecimiento intrínseco, γ (años^{-1}), representa la mortalidad y fecundidad observada en una población. Los valores de γ y de K variarán en función de la localización, modelando variaciones en la capacidad del hábitat para soportar la población.

El segundo término en (2) especifica el proceso de migración. Se trata del modelo de difusión clásico, basado en el operador laplaciano. Según este modelo la población se dispersa de forma que la densidad tiende a ser homogénea. viene dado por:

$$\nabla \cdot (\nu \nabla N) = \nu \frac{\partial^2 N}{\partial \theta^2} + \nu \frac{\partial^2 N}{\partial \phi^2} \quad (3)$$

donde ν es la difusividad en km^2/año.

Davidson et al. [22] proponen la introducción de un término adicional con el fin de reproducir una mayor velocidad de migración en determinadas regiones geográficas, observada en los datos arqueológicos:

$$(V \cdot \nabla)N = V \cdot \left(\frac{\partial N}{\partial \theta}, \frac{\partial N}{\partial \phi}\right) = v_\theta \frac{\partial N}{\delta \theta} + v_\phi \frac{\partial N}{\partial \theta}. \quad (4)$$

La velocidad de advección, $V = (v_\theta, v_\phi)$, representa la facilidad en la que se produce la migración en una determinada dirección en cada localización geográfica, y viene dada en km/año.

Durante las simulaciones, los valores empleados para los parámetros de los diferentes términos del modelo han sido $\gamma = 0.02$ años^{-1}, $\nu = 12.5$km^2/año, $K = 3.5$hab/km^2 y V un campo vectorial con un modulo maximo de 5km/año, según los valores propuestos por Davidson et al. [5].

B. Discretización del modelo

A partir de la ecuación (2), y conociendo el valor de la densidad de población $N(\theta, \phi, t)$, es posible obtener la tasa de variación en cada localización. Por tanto, la realización de las simulaciones consistirá en la resolución de un problema con condiciones iniciales que puede ser integrado numéricamente.

Para la discretización del modelo se ha empleado un esquema de diferencias finitas explícito de orden dos en el espacio y orden uno en el tiempo sobre una malla regular. La malla tiene un paso de discretización espacial de 0.9km y, teniendo en cuenta los valores de los parámetros del modelo y el criterio de estabilidad de Courant-Friedrichs-Lewy, se ha empleado un paso de integración temporal de 0.02 años [23].

IV. Implementación del modelo

Para la realización del estudio de prestaciones se ha decidido partir de una implementación utilizando el sistema de cálculo numérico Matlab [24], por su facilidad de utilización y de procesamiento de los datos de salida. En segundo lugar, se ha realizado una implementación del modelo en C++ con ejecución sin paralelismo. Por último se ha desarrollado una versión del código en C++ orientada al procesamiento paralelo con GPUs utilizando CUDA [18]

Con el fin de disponer además de un modelo paralelo de referencia con el que comparar los resultados de la simulación con C++ en GPU, se han incluido también pruebas utilizando las bibliotecas de Matlab que permiten la ejecución paralela explotando las arquitecturas basadas en GPUs para cálculos matriciales (Matlab-GPU). La diferencia principal respecto a la versión serie consiste en declarar las matrices del modelo en la memoria de la GPU, que posteriormente realizará los cálculos necesarios. Se realizarán, por tanto, simulaciones con un modelo serie

en C++, un modelo paralelo con Matlab ejecutado sobre GPU y un modelo paralelo implementado en CUDA y ejecutado sobre GPU.

Por otro lado, la versión CUDA del modelo se basa en una división de los datos de acuerdo a criterios espaciales (malla 2D). En nuestro caso, cada nodo de la malla necesita acceder a datos pertenecientes a sus vecinos más próximos en ambas direcciones (4 vecinos). Esta división de datos en la malla facilita que cada *CUDA-Thread* se encargue de un conjunto de puntos de la malla, con lo que se maximiza la paralelización de los cálculos del modelo. En la figura 1 podemos ver este reparto, donde la malla (en nuestro caso el mapa de Europa) se divide en bloques horizontales, y cada bloque a su vez se divide en subbloques, que asignaremos a los *CUDA-Threads*. Cada uno de estos Threads puede calcular 1 o varios puntos en función del tamaño del mapa, y sobre cada punto se realiza el calculo del modelo.

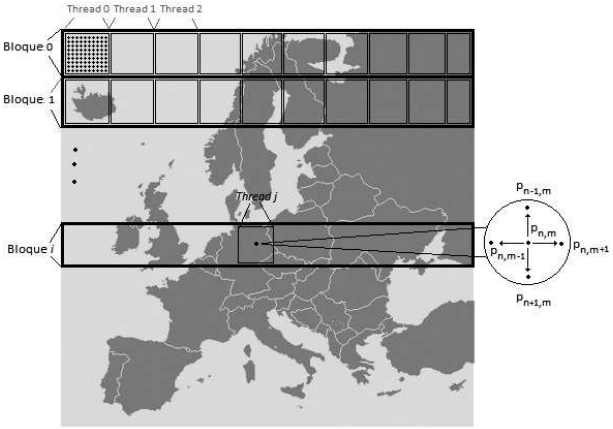

Fig. 1. Reparto en CUDA Threads.

Cada simulación del modelo se puede dividir en dos partes. En primer lugar, una parte dedicada al preproceso, donde se carga el mapa de altitudes para los nodos de la malla y se calculan los valores de los parámetros ν y K. En segundo lugar, la simulación propiamente, con la ejecución el bucle principal en el cual se calculan las discretizaciones de las derivadas de la función $N(\theta, \phi, t)$ y se actualiza el valor de la densidad de población de cada punto.

Las medidas de tiempo incluidas en el presente estudio de aceleración se refieren al tiempo requerido para la ejecución de un ciclo, que corresponde a una ejecución del bucle principal de simulación. Cada uno de los tiempos indicados ha sido promediado con los valores de 100 ejecuciones del bucle. Para la ejecución de las pruebas se han contemplado tres tamaños distintos de mapa, un mapa pequeño con 2400x2400 puntos, donde cada punto representa una medida de altura, uno mediano con 4800x4800 puntos y, por último, un mapa completo de la zona de interés de 10801x6000 puntos.

V. Evaluación de prestaciones

En esta sección se muestra la evaluación de prestaciones de todas las implementaciones del modelo descritas en la sección anterior, en términos de tiempo de ejecución requerido para simular la evolución temporal del modelo durante un periodo determinado. La plataforma utilizada para la realización de las pruebas ha sido un servidor bullX R425 con tarjeta gráfica Nvidia Tesla C2070.

El primer paso en las pruebas ha sido la calibración del modelo implementado en CUDA. Para ello se han realizado varias pruebas, midiendo las prestaciones obtenidas al variar tanto el número de bloques como el de hilos de la ejecución. Una vez vistos los resultados, se ha decidido escoger como tamaño de bloque dos valores, 1024 y 4096, que son aquellos que obtienen mejores resultados en nuestras pruebas. Respecto al número de hilos, se ha seleccionado un valor de 1024 por la misma razón.

En la tabla I resumimos los resultados obtenidos en el primero de los mapas (2400x2400 puntos). En ella mostramos los valores de tiempo medio de ciclo medido en segundos, y la desviación del mismo. Estos valores los mostramos para las implementaciones serie, Matlab-GPU y CUDA (esta última con 2 tamaños distintos de bloque).

TABLA I
Tiempo medio con mapa de 2400x2400 puntos.

	Serie	Matlab GPU	CUDA 1024B	CUDA 4096B
Media (s)	0,5149	0,0726	0,0112	0,0127
Desv.	0,0190	0,0003	0,0050	0,0005

En este primer caso podemos ver como las diferencias en los tiempos medios de ciclo, existe una diferencia de un orden de magnitud entre las versiones serie y paralelo (Matlab-GPU y CUDA). Además, podemos ver como la versión CUDA es 7 veces más rápida que la versión paralela de Matlab. Por otro lado, vemos como los valores de desviaciones de todas las pruebas son pequeños, lo que indica que hay muy poca variabilidad entre las ejecuciones del mismo tipo, mostrándonos un comportamiento muy estable. Este hecho se reproduce en las demás pruebas.

En la tabla II podemos ver los resultados en el caso del mapa mediano. En ella vemos como la diferencia entre las ejecuciones se mantiene respecto al caso anterior. Sin embargo la mejora de la versión CUDA respecto a la versión paralela de Matlab se ha reducido a 2. El estudio de dicha reducción está explicado más adelante.

TABLA II
Tiempo medio con mapa de 4800x4800 puntos.

	Serie	Matlab GPU	CUDA 1024B	CUDA 4096B
Media (s)	1,9300	0,2846	0,1229	0,1030
Desv.	0,0052	0,0029	0,0045	0,0050

En la tabla III tenemos los valores al usar el mapa grande, y en ella podemos ver como se mantiene

la diferencia entre las ejecuciones serie y paralelas. Sin embargo, los tiempos de ejecución entre las dos implementaciones de Matlab y CUDA obtienen valores temporales muy similares e, incluso, la implementaciónen MATLAB obtienen valores ligeramente inferiores a la implementación en CUDA. Estos resultados muestran que la implementación paralela de Matlab escala mejor con el tamaño del mapa que la implementación ad-hoc sobre CUDA.

TABLA III
TIEMPO MEDIO CON MAPA DE 10801X6000 PUNTOS.

	Serie	Matlab GPU	CUDA 1024B	CUDA 4096B
Media (s)	5,4308	0,8971	1,0175	0,9631
Desv.	0,0066	0,0523	0,0046	0,0046

Para ver mejor la evolución de las aceleraciones obtenidas con cada implementación en función del tamaño del mapa utilizado, en la tabla IV mostramos los valores de aceleración obtenidos por la versiones de Matlab y CUDA con respecto a la versión serie.

TABLA IV
ACELERACIÓN OBTENIDA EN FUNCIÓN DEL TAMAÑO DEL MAPA.

	Matlab GPU	CUDA 1024B	CUDA 4096B
2400x2400	7,09	45,67	40,44
4800x4800	6,78	15,69	18,73
10801x6000	6,05	5,34	5,64

La tabla IV muestra que los valores de aceleración obtenidos por Matlab son prácticamente constantes (entre 6 y 7) en los diferentes tamaños de mapa. Es decir, la implementación paralela de Matlab escala con el tamaño del mapa. Sin embargo, en los valores de aceleración obtenidos con la implementación ad-hoc sobre CUDA se aprecia un descenso de la aceleración conforme aumentamos el tamaño del mapa. Pero en cualquier caso, el uso de computación paralela mediante GPUs proporciona una aceleración de más de 5x para el tamaño más grande del problema considerado.

Para estudiar el motivo de la falta escalabilidad en la implementación ad-hoc sobre CUDA, hemos estudiado tanto la utilización de la GPU como de la memoria de la GPU que hacen ambas implementaciones. La figura 2 muestra el porcentaje de utilización de la GPU. Esta figura muestra que la implementación ad-hoc basada en CUDA utiliza casi al 100% la GPU, mientras que la implementación basada en Matlab no pasa de una utilización media del 87%, para el caso del mapa de 4800x4800 puntos. Por tanto, parece haber una mayor (mejor) utilización de la GPU en la implementación ad-hoc, y por tanto esta gráfica no explica el comportamiento no escalable con el tamaño del mapa que presenta dicha implementación.

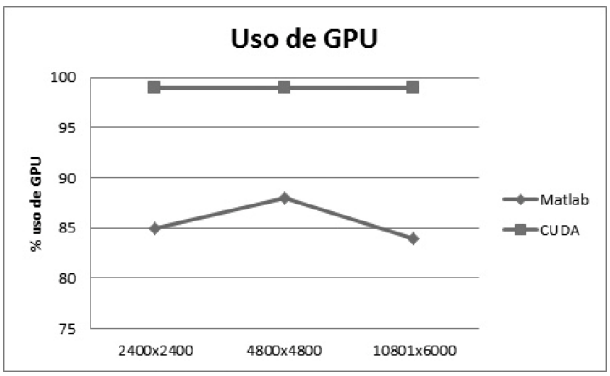

Fig. 2. Porcentaje de utilización de la GPU en los distintos tamaños de mapa utilizados.

La figura 3 muestra la utilización que cada implementación hace de la memoria presente en la GPU. Esta figura, sin embargo, muestra cómo el uso que la implementación basada en Matlab utiliza porcentualmente más memoria de GPU que la implementación ad-hoc basada en CUDA, y cómo la diferencia de la utilización de memoria se va ampliando conforme aumenta el tamaño de mapa considerado. Esta diferencia explica la mejor escalabilidad de la primera implementación frente a la segunda. La razón para este comportamiento es que al aplicar el modelo sobre tamaños de mapa de mayor escala, el porcentaje de puntos del mapa que coinciden sobre el mar aumenta. La versión ad-hoc procesa estos puntos aunque su valor de densidad sea 0 (no puede haber población asentada en el mar, obviamente), mientras que la versión basada en Matlab hace preprocesamiento de estos puntos, triangularizando las matrices y procesando sólo aquellas zonas que no coinciden con zona marina. Por eso sus prestaciones son lineales con la escala del mapa, mientras que las de la versión ad-hoc no lo son.

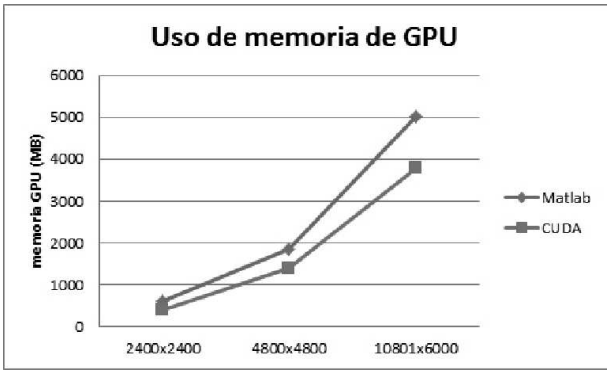

Fig. 3. Uso de memoria de GPU en los distintos tamaños de mapa utilizados.

VI. CONCLUSIONES

En este trabajo hemos presentado diferentes implementaciones sobre tarjetas gráficas de un modelo de migración de la población humana del Neolítico. Este modelo de reacción-difusión se basa en ecuaciones en derivadas parciales y tiene en cuenta el

efecto de las particularidades geográficas de cada región para modular el proceso. Con el objetivo de realizar un estudio comparativo, hemos considerado la implementación secuencial clásica basada en CPUs actuales, implementaciones basadas en un programa comercial de cálculo numérico y una ad-hoc básica no optimizada (sin preprocesamiento de datos, ni optimización en el algoritmo utilizado para la multiplicación de matrices, etc.) sobre tarjeta gráfica (GPU).

Los resultados de simulación muestran que la implementación paralela de Matlab escala con el tamaño del mapa, ya que los valores de aceleración obtenidos no varían con el tamaño. Sin embargo, en los valores de aceleración obtenidos con la implementación ad-hoc básica sobre CUDA se aprecia un descenso de la aceleración conforme aumentamos el tamaño del mapa. Esta variación muestra que el uso la memoria de la GPU está claramente mejorado en las librerías comerciales con respecto a la implementación ad-hoc básica considerada. Por otro lado, los resultados muestran claramente que el uso de procesamiento en GPUs, ya sea utilizando las librerías comerciales Matlab-GPU, o incluso mediante implementaciones ad-hoc básicas, puede acelerar significativamente la simulación de estos modelos de migraciones humanas utilizados en arqueología, consiguiendo en cualquier caso una aceleración 5 veces superior con respecto a la implementación secuencial.

Agradecimientos

Este trabajo ha sido financiado conjuntamente por los fondos FEDER de la Unión Europea, por el anterior Ministerio Español de Ciencia e Investigación (MICINN), actualmente Ministerio de Economía y Competitividad (MINECO) bajo los proyectos Consolider-Ingenio CSD2006-00046, TIN2009-14475-C04-04, y TIN2010-12011-E y por la Generalitat Valenciana, bajo el proyecto para Grupos Emergentes GV/2012/007.

Referencias

[1] stephen shennan, "evolutionary demography and the population history of the european early neolithic.," *hum biol*, vol. 81, no. 2-3, pp. 339–55, 2009.

[2] A. J. Ammerman and L. L. Cavalli-Sforza, *The Neolithic Transition and the Genetics of Populations in Europe*, Princeton University Press, Princton :, 1984.

[3] Carsten Lemmen, Detlef Gronenborn, and Kai W Wirtz, "A simulation of the neolithic transition in western eurasia," *Journal of Archaeological Science*, vol. 38, no. 12, pp. 3459–3470, 2011.

[4] ron pinhasi, joaquim fort, and albert j ammerman, "tracing the origin and spread of agriculture in europe.," *plos biol*, vol. 3, no. 12, pp. e410, 2005.

[5] K. Davison, P. M. Dolukhanov, G. R. Sarson, A. Shukurov, and G. I. Zaitseva, "Multiple sources of the european neolithic: Mathematical modelling constrained by radiocarbon dates," *Quaternary International*, vol. 203, no. 1-2, pp. 10 – 18, 2008.

[6] R.A. Bentley, P. Ormerod, and M. Batty, "Evolving social influence in large populations.," *Behavioral ecology and sociobiology.*, vol. 65, no. 3, pp. 537–546, March 2011.

[7] Jean-Pierre Bocquet-Appel, Stephan Naji, Marc Vander Linden, and Janusz K. Kozlowski, "Detection of diffusion and contact zones of early farming in europe from the space-time distribution of 14c dates," *Journal of Archaeological Science*, vol. 36, pp. 807–820, 2009.

[8] S. Shennan and J. Steele, "Chronological patterns in the neolithisation of europe," 2000, http://ads.ahds.ac.uk/catalogue/resources.html?c14meso.

[9] Marina Gkiasta, Thembi Russell, Stephen Shennan, and James Steele, "Neolithic transition in europe: the radiocarbon record revisited," *Antiquity*, vol. 77, no. 295, pp. pp. 45–62, 2003.

[10] L. Thissen, A. Reingruber, and D. Bischoff, "14c databases and chronocharts (2005-2006) updates," 2006, http://www.canew.org/data.html.

[11] Timofeev V.I., Zaitseva G.I., Dolukhanov P.M., and Shukurov A.M., "Radiocarbon chronology of the neolithic in northern eurasia.," *Teza*, 2004.

[12] Pavel Dolukhanov, Anvar Shukurov, Detlef Gronenborn, Dmitry Sokoloff, Vladimir Timofeev, and Ganna Zaitseva, "The chronology of neolithic dispersal in central and eastern europe," *Journal of Archaeological Science*, vol. 32, no. 10, pp. 1441 – 1458, 2005.

[13] A J Ammerman and L L Cavalli-Sforza, *A population model for the diffusion of early farming in Europe*, pp. 343 – 357, Duckworth, 1973.

[14] Christopher Tilley, *A Phenomenology of Landscape: Places, Paths and Monuments*, vol. 69, Berg, 1994.

[15] Fort J., "Population expansion in the western pacific (austronesia): A wave of advance model.," *Antiquity*, vol. 77, pp. 520 – 530, 2004.

[16] Cavalli-Sforza LL Fort J, Pujol T, "Paleolithic populations and waves of advance," *Cambridge Archaeological Journal*, vol. 14, pp. 53 – 61, 2004.

[17] A G Fix, "Gene frequency clines produced by kin-structured founder effects.," *Human Biololgy*, vol. 69, no. 5, pp. 663–73, 1997.

[18] NVIDIA, *NVIDIA CUDA Programming Guide 3.2*, 2010.

[19] Allen R Sanderson, Miriah D Meyer, Robert M Kirby, and Chris R Johnson, "A framework for exploring numerical solutions of advection - reaction - diffusion equations using a gpu-based approach," *Computing and Visualization in Science*, vol. 12, pp. 155 – 170, 2008.

[20] B. Rocker T. Hahn, V. Heuveline, "Fluid and particulate flows with cuda: Gpu accelerated scientific computing," *PARS-Mitteilungen*, vol. 26, 2009.

[21] Guillermo Vigueras, Juan M. Orduña, and Miguel Lozano, "A gpu-based multi-agent system for real-time simulations," in *PAAMS*, 2010, pp. 15–24.

[22] K. Davison, P. Dolukhanov, G.R. Sarson, and A. Shukurov, "The role of waterways in the spread of the neolithic," *Journal of Archaeological Science*, vol. 33, no. 5, pp. 641 – 652, 2006.

[23] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, SIAM: Society for Industrial and Applied Mathematics, 2004.

[24] MATLAB, *version 7.10.0 (R2010a)*, The MathWorks Inc., Natick, Massachusetts, 2010.