

PRUEBAS, RESULTADOS y CONCLUSIONES

"When you have eliminated all which is impossible, then whatever remains, however improbable, must be the truth."

"The Adventure of the Blanched Soldier."

Sir **Arthur Conan Doyle** (1859–1930)

I. PRUEBAS:

Una vez concluida la etapa de desarrollo del código, se ha de comprobar la eficiencia de la tecnología CUDA aplicada al problema de partida. Recordemos que la motivación de nuestro proyecto era analizar las posibilidades de la tecnología CUDA aplicada al cálculo numérico para los problemas de mecánica computacional. En esta etapa final se ha de comprobar cuán útil podía ser esta tecnología como nueva herramienta a integrar junto a las demás disponibles.

Para proceder de manera sistemática se definieron las condiciones para la realización de las pruebas.

METODOLOGÍA:

a. EQUIPO:

Se procedió a realizar las pruebas en el equipo del laboratorio descrito anteriormente. El equipo tenía instalado el Sistema Operativo "Windows 7" por defecto. Sin embargo para las necesidades del proyecto se decidió instalar Linux.

La primera opción fue instalar "Ubuntu" en sus versiones 9.04, 9.10 o bien 10.04, pero este sistema operativo dio problemas para reconocer y deshabilitar el sistema de discos RAID. Sin embargo con el sistema "OpenSuse11.2", se pudo instalar el sistema en un disco, quedando el segundo disponible para futuras aplicaciones.

Tras la instalación del Sistema Operativo y de los paquetes básicos, se procedió a instalar las herramientas propias para programación en CUDA y el programa MATLAB.

b. Datos de entrada:

Mediante la herramienta de mallado de MATLAB se generaron 4 modelos con aumento gradual del número de elementos.

Este modelo de mallado se somete al programa "TrataMalla" para generar los archivos correspondientes:

1. coordenadas.dat.
2. Triangulos.dat
3. dirichlet.dat y
4. neumann.dat

Estos 4 archivos constituyen los datos de entrada a los programas de prueba, mediante los cuales queda definido enteramente el modelo de mallado. En la tabla 1 se resumen las características fundamentales de los 4 modelos.

MALLADO	Nodos	Triángulos	Magnitud aproximada de la diferencia entre las coordenadas de 2 puntos contiguos: $\Delta x = \frac{1}{\text{Nodos}}$	Número de Puntos aproximado por eje.
A	10145	19968	0,009928279	100
B	40257	79872	0,004984014	200
C	160385	319488	0,002496998	400
D	640257	1277952	0,001249749	800

Tabla 1: Mallados y sus características fundamentales.

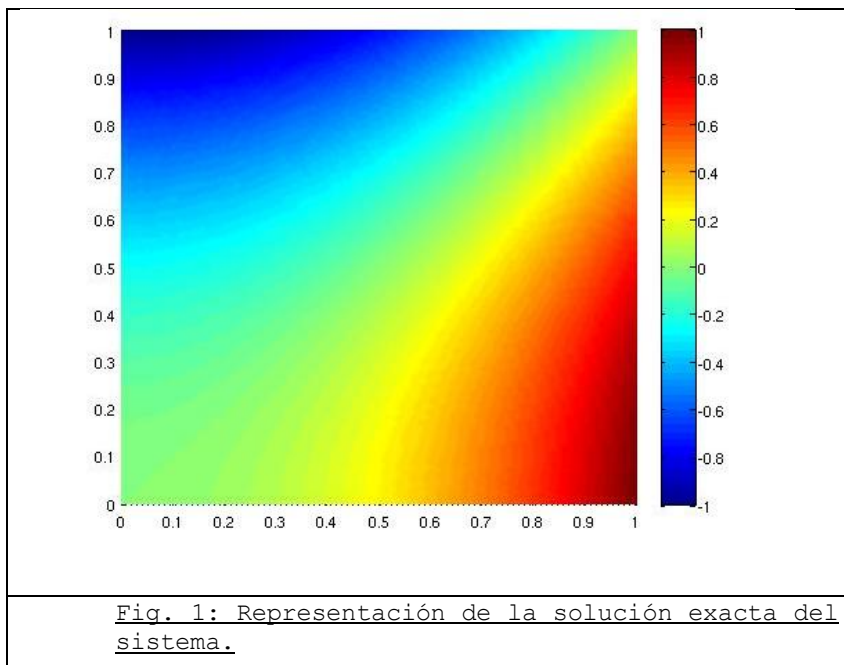
La tabla 1 también nos ofrece una idea de la magnitud mínima aproximada de la solución que se espera en cada mallado. Recordemos que la magnitud de la solución es aproximadamente la misma que la magnitud de la diferencia entre coordenadas. Un error aceptable debería ser, al menos, inferior en un orden de magnitud.

c. SISTEMA DE REFERENCIA.

La solución exacta del sistema es conocida de antemano, debido a la sencillez del sistema.

Mediante MATLAB se generó para cada mallado un archivo que contenía el vector solución exacta llamado "exacta.dat". Este archivo será usado para comprobar la exactitud de las soluciones ofrecidas tanto por el propio MATLAB como por los códigos desarrollados y permitirá la medición del error absoluto de las soluciones ofrecidas. La solución exacta cuyos valores están en el intervalo [-1; 1] es usada para la representación del error absoluto de las aplicaciones comparadas.

En la Fig. 1 se representa la solución exacta del sistema.



Por otro lado el sistema de referencia con respecto al cual se van a comparar las soluciones basadas en CUDA es el generado por el código MATLAB inicial. Esta aplicación realiza las siguientes tareas de manera secuencial:

- Lee los archivos de entrada.
- Genera la matriz de rigidez.
- Genera el vector de términos independientes.
- Resuelve el sistema $A \cdot X = B$ mediante el método directo usando la operación `"x[Lib]=A[Lib]\b[Lib]"`, para los nodos No Dirichlet.
- Ofrece en forma de archivo de texto la solución.

Cabe destacar que el operador `"\"` resuelve el sistema mediante un método directo. Sin embargo su sustitución por métodos iterativos `"bicgstab"` y `"cg"` de la biblioteca de MATLAB no modificaron la exactitud ni el tiempo de ejecución de manera apreciable.

Este código se modificó para medir el tiempo de ejecución total mediante la inserción de las instrucciones `"tic"` y `"toc"`. Pero también se usaron estas instrucciones para medir el tiempo de ejecución de la instrucción `"x[Lib]=A[Lib]\b[Lib]"` para compararlo con el tiempo invertido en las iteraciones del Solver CUDA.

De nuevo en MATLAB se usan los archivos de salida y se representan mediante un programa sencillo donde se comparan la solución, la solución exacta y los errores de manera gráfica.

d. Sistemas DE PRUEBA:

Las pruebas básicas consisten en medir el rendimiento de la aplicación global, que al igual que el código MATLAB realiza los pasos conducentes a la obtención del vector solución partiendo de los archivos de mallado. En concreto la secuencia de tareas es la siguiente:

- Lee los archivos de entrada.
- Genera la matriz de rigidez bruta.
- Elimina de la matriz las redundancias y los valores nulos obteniendo A.
- Genera el vector de términos independientes B.
- Reordena la Matriz de rigidez A para resolver en los nodos No-Dirichlet.
- Resuelve el sistema $A \cdot X = B$ mediante un método iterativo del subespacio de Krylov.
- Reordena el vector solución mediante la incorporación de las condiciones de Dirichlet.
- Ofrece en forma de archivo de texto la solución.

Los sistemas de prueba usan precisión "float" para todas las tareas excepto para el bloque de resolución, donde se utiliza precisión "double".

Los sistemas de pruebas se distinguen en cuanto al tipo de método iterativo que implementa el Solver. No se incluyó el Solver CUSP porque no ofrecía suficiente flexibilidad para integrarlo fácilmente en el conjunto de la aplicación. Por tanto los sistemas de pruebas serían:

- Solver CUBLAS basado en el método del Gradiente Conjugado (CG) con precisión "double" en el método iterativo.
- Solver CUBLAS basado en el método del bi-Gradiente Conjugado Estabilizado (bi-cgstab) con precisión "double" en el método iterativo.

e. MAGNITUDES MEDIDAS:

Nuestro objetivo es medir el rendimiento y eficiencia de las soluciones basadas en CUDA con respecto a una aplicación comercial por un lado y por otro compararlas entre sí. Para ello se compararán las siguientes características:

VELOCIDAD: Este es el objetivo principal al evaluar la Tecnología CUDA. Se pretende conocer hasta qué punto la paralelización y la ejecución en la GPU es capaz de acelerar las aplicaciones desarrolladas comparadas con un Software comercial como MATLAB. La velocidad se mide mediante el tiempo de ejecución de la aplicación completa; es decir desde la carga de los archivos generados por trataMalla hasta la generación del archivo que contiene el vector solución.

CONSUMO DE MEMORIA: Una limitación importante de las aplicaciones es el tamaño de las matrices que son capaces de mantener en memoria y de manejar. En este punto no debemos ignorar que hay siempre memoria que se usa para operaciones auxiliares o adyuvantes y que disminuye la memoria disponible.

PRECISIÓN: Partiendo de la limitación de la GPU en cuanto a que maneja precisión de tipo "float" de 32 bits y "double" de 64 bits, así como los cambios de formato debidos a los volcados y lecturas de archivos de texto, se decidió estudiar este aspecto.

Las características de la representación de los valores en las máquinas en función de la precisión según el estándar de la IEEE para aritmética en coma flotante (IEEE 754-2008) se muestran en la tabla 2.

Modelo	Signo	Mantisa	exponente	Total en bits	Bits de precisión
16 bits	1	10	5	16	11
32 bits(float)	1	23	8	32	24
64 bits(double)	1	52	11	64	53
80 bits(extended)	1	64	15	80	64
128 bits(Quadruple)	1	112	12	128	113

Tabla 2: Representación en función de la precisión según (IEEE 754-2008).

Para los formatos "float" y "double" esto implica que el error mínimo debido a la precisión de la máquina es de 2^{-24} ^a (aprox. 10^{-7}) y de 2^{-53} ^b (aprox. 10^{-16}) respectivamente. Estas serían las cotas de error absoluto teórico para las operaciones simples.

Matlab soporta el formato "long double" de 80 bits. Este formato es soportado por lenguaje C convencional, sin embargo no por las tarjetas gráficas CUDA de prueba. En este caso el error de máquina sería 2^{-64} ^c (aprox. 10^{-19})

En nuestro modelo para los mallados analizados, el error mínimo en las operaciones del Solver iterativo dependerá además del tamaño del vector. Este error sería la cota mínima de la norma del residuo según

la fórmula: $\mathcal{E} = \sqrt{\sum_{i=1}^{i=Nodos} \mathcal{E}_{\min}^2}$, obteniendo así los valores incluidos en la tabla 3.

^a $2^{-24} = 5,9604644775390625e-8$.

^b $2^{-53} = 1,1102230246251565404236316680908e-16$.

^c $2^{-80} = 5,4210108624275221700372640043497e-20$.

Mallado	Num. Nodos	Cota aproximada de error en float	Cota aproximada de error en double	Cota aproximada de error en extended
A	10145	1e-5	1e-14	1e-17
B	40257	2e-5	2e-14	2e-17
C	160385	4e-5	4e-14	4e-17
D	640257	8e-5	8e-14	8e-17

Tabla 3: Cota del residuo debido a la precisión de la máquina según el estándar de la IEEE para aritmética en coma flotante (IEEE 754-2008).

Estas serían las cotas aproximadas de los residuos y se debe tener en cuenta su magnitud a la hora de definir los parámetros de parada de las iteraciones del solver iterativo. Recordemos que estos interrumpen sus iteraciones en 3 circunstancias:

- Cuando se alcanza una norma de residuo igual o inferior a la cota de precisión exigida.
- Cuando se supera un máximo número de iteraciones.
- Cuando la norma del residuo se hace demasiado grande.

ERROR: Es una medida del ajuste del resultado con respecto al valor exacto. Cuando expresamos la exactitud de un resultado se expresa mediante el error absoluto que es la diferencia entre el valor obtenido y el valor exacto. Sin embargo para representar la magnitud de este error con respecto al valor exacto usamos el error relativo. En nuestro caso no se puede representar gráficamente el error relativo, por que la solución exacta se anula en varias regiones del dominio de resolución.

El error en nuestro sistema será debido a:

- Error del método de resolución que se basa en aproximaciones.
- Error de los cálculos que se realizan con una precisión limitada.

Para su medida representaremos el error absoluto de los sistemas de prueba y del de referencia.