

DESARROLLO DE HERRAMIENTAS DE CÁLCULO POR MÉTODO SPH PARA PROCESOS DE GRANDES DEFORMACIONES

L. Buglioni⁽¹⁾, H.G.Svoboda^(2,3)

buglioni@inti.gov.ar

⁽¹⁾ Dependencia del Autor - INTI,

⁽²⁾ Depto de Ingeniería Mecánica, Facultad de Ingeniería, UBA. Buenos Aires, Argentina.

⁽³⁾ CONICET. Buenos Aires, Argentina.

Palabras Clave: SPH; FEM; FSW; GPU

INTRODUCCIÓN

La mecánica computacional es una poderosa herramienta utilizada de manera extensa en todas las áreas de la ingeniería. Es utilizada en diferentes procesos para obtener valores de ciertas variables difíciles de medir. Los autores han trabajado en la simulación numérica de procesos de Soldadura por Fricción- Agitación (FSW) y su variante de punto (FSSW) [1] que involucran grandes deformaciones. El método *Smoothed Particle Hydrodynamics* es un método sin malla, recientemente utilizado con éxito para el proceso FSW [2], con la ventaja de no contar con los inconvenientes de mallado del método de elementos finitos (FEM) tradicional. Por otra parte, se hoy día se cuenta con el poder de cálculo de las placas gráficas (GPU), que resultan en tiempos de cálculo reportados de hasta 70 veces menores al cálculo convencional por CPU [2]. Finalmente, el desarrollo de programas gratuitos de fuente abierta (*Open Source* o *FOSS*) permite a los usuarios utilizar dichas herramientas y optimizarlas para soluciones particulares.

OBJETIVOS

El objetivo del presente trabajo es desarrollar dos herramientas libres y gratuitas para el cálculo de procesos que involucran grandes deformaciones plásticas de corta a media duración, acelerados por CPU / GPU.

DESARROLLO

El método SPH es un método sin malla utilizado para una gran cantidad de problemas dinámicos, que puede ser fácilmente formulado en un marco de referencia *Lagrangiano* permitiendo resolver problemas de grandes deformaciones. El método fue propuesto por dos grupos independientes en 1977 en el campo de la astrofísica. En 1982 fue

modificada la formulación para conservar la cantidad de movimiento lineal y angular [3]. En el método SPH, las ecuaciones de conservación son resueltas por interpolación a partir de un set de partículas j que están dentro de la influencia del dominio de la partícula i (Fig. 1).

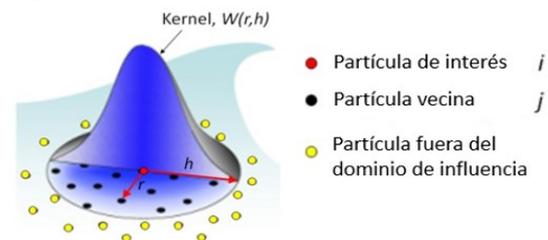


Figura 1: interpolación en el método SPH [2]

La herramienta desarrollada y puesta a prueba en este trabajo (*WeldForm*¹) está siendo escrita en lenguaje C++ tomando como base la librería existente *PersianSPH*², agregándose la resolución de problemas térmicos [4] y acoplamiento termomecánico (en desarrollo), contacto con fricción estática y dinámica (con generación de calor), materiales bilineales, de Hollomon y dependientes de la deformación, su velocidad y temperatura (*Johnson-Cook*) ecuaciones de flujo de calor, generación de calor por deformación plástica, cálculo de energía, condiciones de simetría, integración temporal secuencial, entre otros. Por otra parte, se está desarrollando desde cero otra herramienta *-WeldFormGPU*³, que trabaja por aceleración GPU (*Graphics Processor Unit*) en su propio lenguaje (CUDA).

El desarrollo de los diferentes códigos difiere respecto a cómo se paraleliza el cálculo. En el caso de la versión CPU, se ha hecho por par de partículas, mientras que en el caso GPU se paraleliza por partícula siguiendo a [2]. El primer enfoque tiene la ventaja de que el tiempo cálculo de fuerzas se reduce a la mitad (ya que la contribución de fuerza es igual y

Esta obra está bajo una Licencia Creative Commons Atribución-No Comercial-Sin Derivadas 4.0 Internacional (CC BY-NC-ND 4.0) <https://creativecommons.org/licenses/by-nc-nd/4.0/>

(1) <https://github.com/luchete80/WeldForm>

(2) <https://github.com/mghkorzani/persiansph>

(3) <https://github.com/luchete80/WeldFormGPU>

opuesta en cada par de partículas), teniendo a su vez la desventaja de ser algo más lento en el momento de escribir en memoria.

RESULTADOS

Para este programa se ha escrito alrededor de 5500 líneas de código fuente para la versión CPU del programa y más de 30k líneas de código en casos de testeo, mientras que se han escrito más de 9k líneas de código para la versión GPU hasta el momento. En la Figura 2 se muestran las deformaciones plásticas a los 5ms en un cilindro de aluminio sometido a compresión, tal como el de [2], pero agregando contacto con fricción 0.1 (estática y dinámica), cuyos resultados son equivalentes a aquellos obtenidos en ABAQUS®.

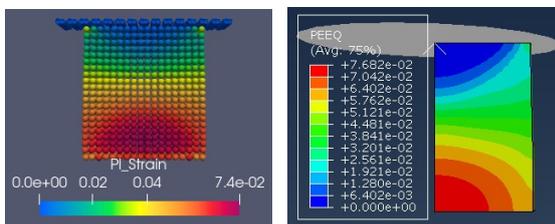


Figura 2: Temperatura en un cilindro sometido a compresión: ABAQUS® (izq) WeldForm (der)

En la Figura 3 se muestra la temperatura luego de 5 milisegundos para un cilindro de aluminio similar al anterior comprimido en ambos extremos [2]. Se verifica muy buena correlación.

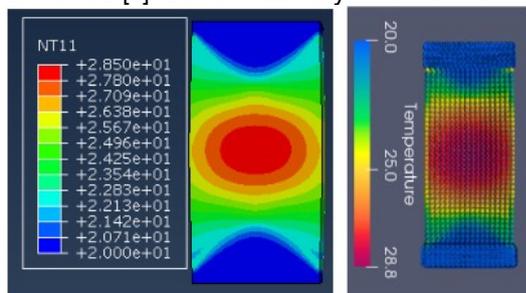


Figura 3: Temperatura en un cilindro sometido a compresión: ABAQUS® (izq) WeldForm (der)

En la Fig. 4 se muestra una probeta cilíndrica sometida a tracción [2]. Puede verse que los valores son similares como en los casos anteriores.

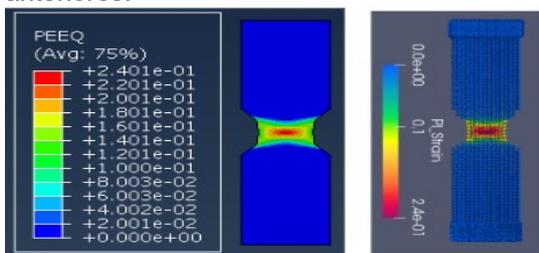


Figura 4: Deformación plástica equivalente en una probeta sometida a tracción: ABAQUS (izq) WeldForm® (der)

En la Fig. 5 se muestra la salida de un problema de compresión de 12k partículas, resuelto en cada una de las aproximaciones. Puede verificarse que, aún realizando mayor cantidad de cálculos, el tiempo por GPU es 5 veces mayor (219s vs 1100s, primeras líneas).

```

-----
Time 0.0101001, GPU time 219.287
Current time step: 7.19739e-07
Forces calc: 0
Stresses calc: 2.927
Max disp: 0.0147374, 0.0147436, 0.00209907
Max pl strain: 0.262643
Total Steps: 14833, time spent 219.293000
Program ended.
-----
Total CPU time: 1.1e+03
Calculation Times
Accel: 0.26%, Density: 0.19%, Stress: 0.31%,
Energy: 0.11%, Contact: 0%, Nb: 0.05%, Update: 0.0097%,
Output No. 102 at 0.01 has been generated
Current Time Step = 7.2e-07
Max plastic strain: 0.24in particle6619
Max Displacements: 3 [ 0.014 0.014 0.0019 ]
    
```

Fig. 5 Tiempo de cálculo en CPU (arriba) y CPU (abajo).

DISCUSIÓN Y CONCLUSIONES

Se encuentran en desarrollo dos herramientas de cálculo numérico por métodos complementarios. Se han obtenido resultados satisfactorios en los benchmarks resueltos por ambas, mientras que se cuenta con una interfaz intuitiva y de alta calidad que cumple con los objetivos propuestos. Por otra parte, más allá de las diferencias en la implementación, se encuentra una gran ventaja acelerando el cálculo por placa gráfica vs el cálculo por CPU tradicional, lo cual es de gran utilidad dada la complejidad de los modelos de FSW a desarrollar en el futuro.

AGRADECIMIENTOS

Se agradece a la Universidad de Buenos Aires por el equipamiento GPU y las licencias de ABAQUS®.

REFERENCIAS BIBLIOGRÁFICAS

[1] Buglioni L., Piccini J.M., Svoboda H.G. (2019) Desarrollo de modelos explícitos para soldadura de punto por fricción-agitación (FSSW): evaluación de la geometría de la herramienta. Proc. del XXIV Congreso sobre Métodos Numéricos y sus Aplicaciones.

[2] K. Fraser (2017). "Robust And Efficient Meshfree Solid Thermo Mechanics Simulation of Friction Stir Welding". Université du Québec à Chicoutimi, UQAC.

[3] Gingold RA, Monaghan JJ. Kernel estimates as a basis for general particle methods in hydrodynamics. Journal of Computational Physics. 1982;46: 429-53.

[4] Buglioni L., Miranda A., Tufaro L.N., Svoboda H.G. "Desarrollo de herramientas de cálculo por SPH orientadas a la resolución de problemas térmicos", CADI/CLADI CAEDI V Congreso Argentino de Ingeniería, Noviembre 2021.

- (1) <https://github.com/luchete80/WeldForm>
- (2) <https://github.com/mghkorzani/persiansph>
- (3) <https://github.com/luchete80/WeldFormGPU>