Sistema trampolín / cuerpo rígido

Trabajo Práctico - Simulación de Sistemas

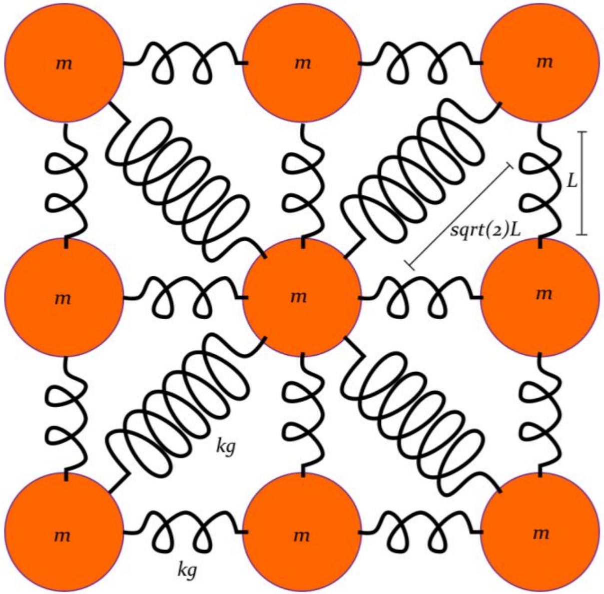
Equipo 9
Marcantonio - 56288 | Raies - 56099 | Saqués - 56047

Sistema físico y Modelización

Modelo del trampolín

- El trampolín ha sido modelizado como una grilla bidimensional de partículas de igual radio y masa, separadas por una distancia L establecida por el usuario, enlazadas a sus ocho vecinos con resortes de constante elástica k_{grid}.
- Las dimensiones de la grilla también son parametrizables.

Modelo del trampolín



Sistema físico del trampolín

- Las fuerzas aplicadas a cada partícula de pueden subdividir en elásticas y viscosas. Las fuerzas elásticas siguen la Ley de Hooke:

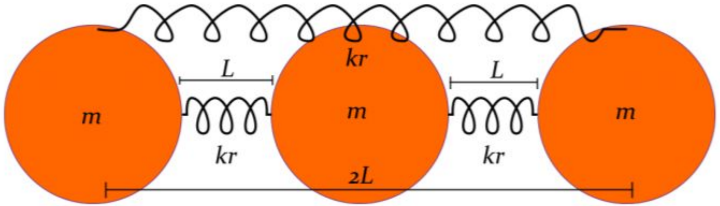
$$F_{el(i,j)} = -k_{grid} * (d_{(i,j)} - L) * \vec{r}_{i,j}$$

- Por otro lado, las fuerzas viscosas siguen una forma proporcional a la velocidad de la partícula, según la fórmula:

$$F_{vi(i)} = -b_{grid} * \vec{V}.$$

Modelo del cuerpo rígido

- El cuerpo rígido se ha modelado como una serie de masas de radio r, unidas por resortes donde $k_{rigid} \gg k_{grid}$.



Colisiones entre partículas del trampolín y del cuerpo rígido

- Las colisiones entre partículas pueden dividirse en normales y tangenciales. Las fuerzas normales siguen la forma:

$$F_{N(i,j)} = -k_n \, \xi_{i,j} \, \vec{n}_{i,j}$$
$$\xi_{i,j} = R_i + R_j - \|\vec{r}_i - \vec{r}_j\|$$

- Las fuerzas tangenciales se definen de la siguiente forma:

$$F_{T(i,j)} = -k_t \, \xi_{i,j} \left[(\vec{v}_i - \vec{v}_j) \, \vec{t}_{i,j} \right] \vec{t}_{i,j}$$

Implementación

Arquitectura

- Simulación : CUDA/C++ (GPGPU)
- Animación : OVITO
- Gráficos: Octave

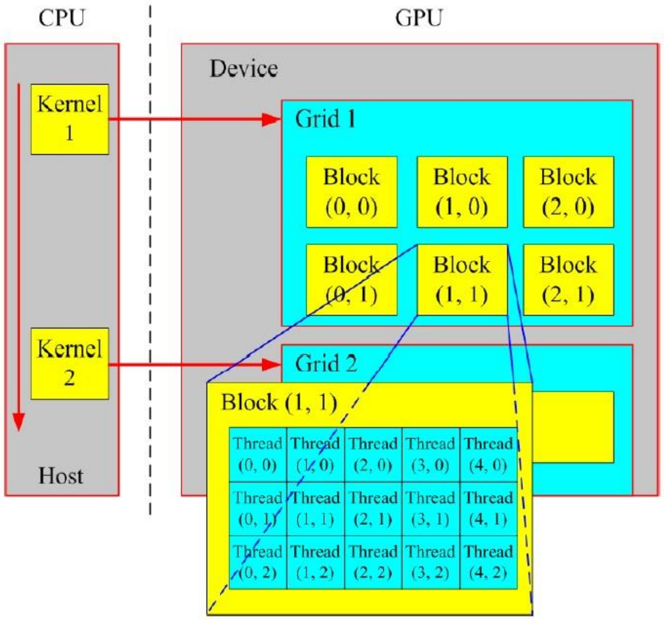
GPGPU

- El concepto GPGPU se basa en la implementación de funciones conocidas como kernels, nombre que surge de las matrices de convolución habitualmente utilizadas en aplicaciones gráficas.
- Estos kernels son unidades de código ejecutadas paralelamente por unidades de procesamiento de alto throughput como son las GPUs .

Invocación de *kernels*

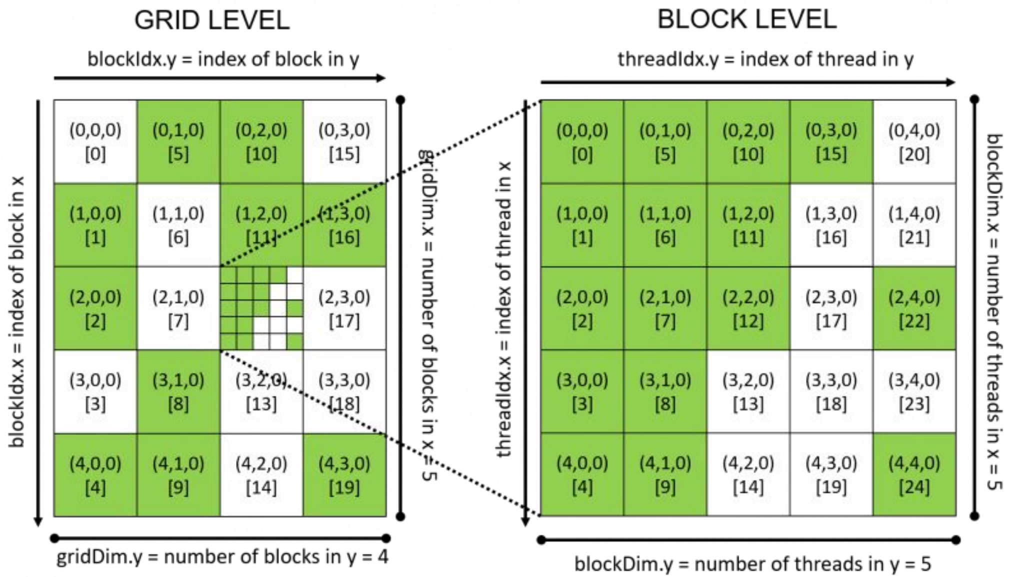
```
gridElasticForce<<<Dg,Db,Ns,S>>>(grid_dev,...);
```

- ***dim3 Dg:*** dimensiones de la grilla
- ***dim3 Db:*** dimensiones del bloque
- ***size_t Ns:*** tamaño de la memoria compartida
- ***cudaStrean_t S:*** el *stream* asociado



Contexto de cada *thread*

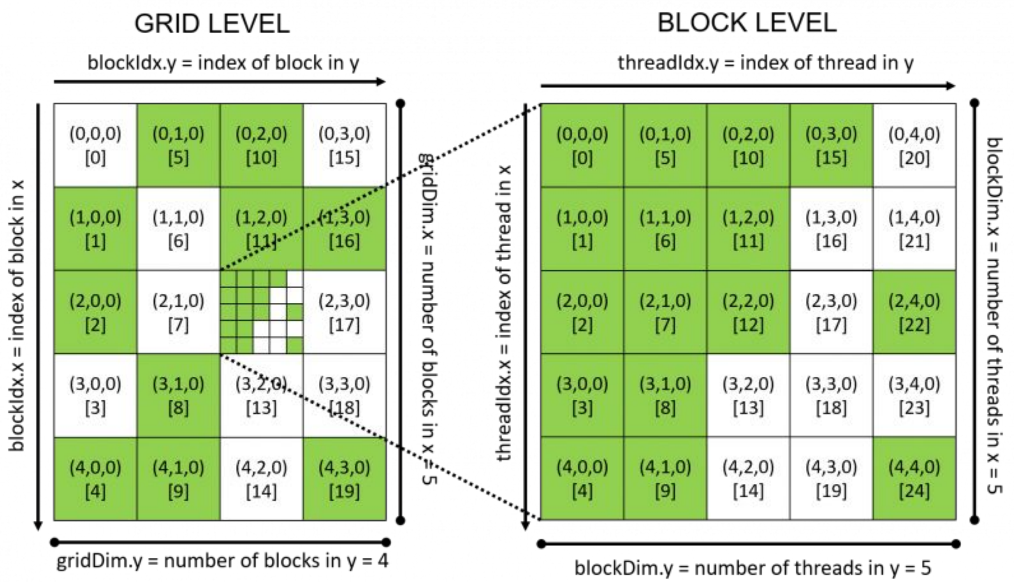
- ***dim3* blockDim*:*** dimensiones del ***block***
- ***dim3* blockIdx*:*** posición del ***block*** dentro de la ***grid***
- ***dim3* threadIdx*:*** posicion del ***thread*** dentro del ***block***



Contexto de cada *thread*

- ***dim3* blockDim*:*** dimensiones del ***block***
- ***dim3* blockIdx*:*** posición del ***block*** dentro de la ***grid***
- ***dim3* threadIdx*:*** posicion del ***thread*** dentro del ***block***

Índices*:* blockIdx *** blockDim *+* threadIdx

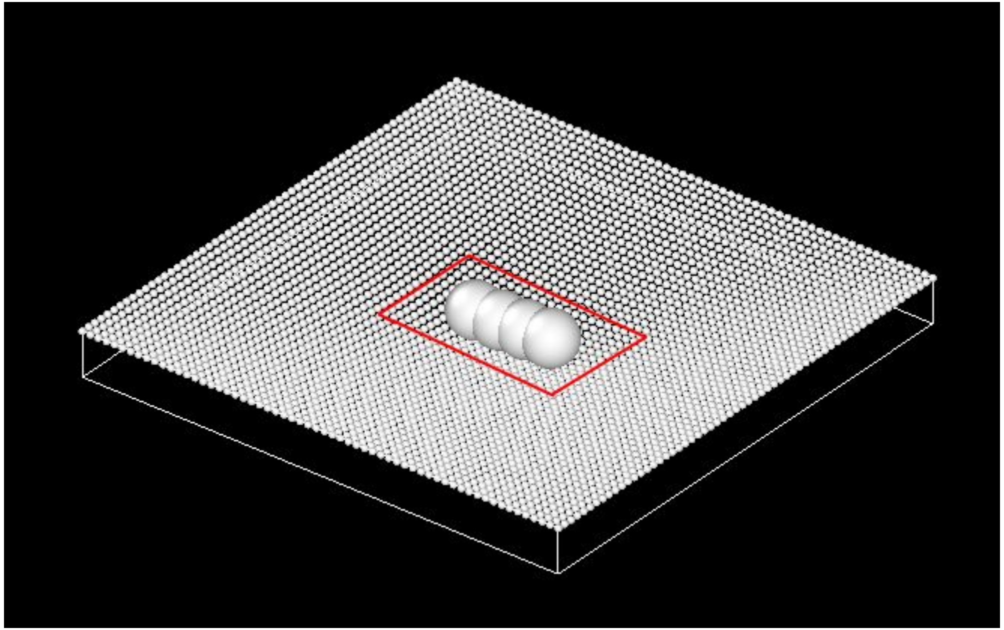


Integración: *Verlet*

$$.x(t + \Delta t) = x(t) + v(t)\ \Delta t + ½\ a(t)\ \Delta t\text{^}2$$
$$a(t + \Delta t\ ) = f(x(t + \Delta t\ ))$$
$$v(t + \Delta t) = v(t) + ½\ (a(t) + a(t + \Delta t\ ))\ \Delta t$$

- `verletPositions<< <dimGrid, dimBlock >> >(grid,delta_t):`
- `gridElasticForce<< <dimGrid, dimBlock >> >(grid, k, b, natural)`
- `computeBigMassForces(big, big_size, grid, sep, kn, kt, sep_big, kbig, bbig);`
- `updateEulerBigMass(big,big_size, grid, delta_t,sep_big, g_earth);`
- `verletVelocities << <dimGrid, dimBlock >> > (grid, delta_t);`

Colisiones entre los cuerpos y la grilla: área

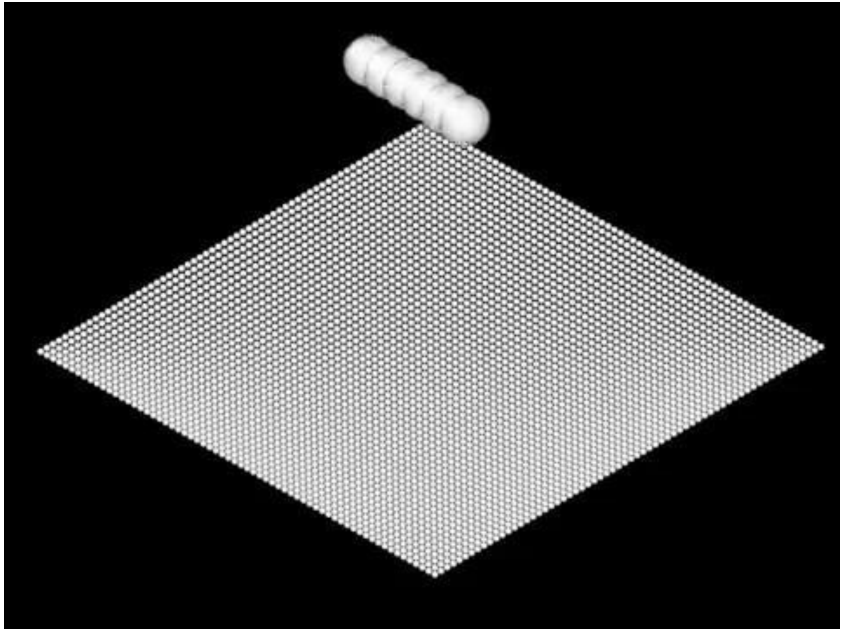


Resultados

Parámetros

- Método = verlet
- t sim = 5s
- Δt = 0,0001 s
- filas = 64
- columnas = 64
- separación horizontal en la grilla= 0,05 m
- masa de partículas de la grilla= 0,001 kg
- radio de partículas de la grilla= 0,02 m
- g= 9,81 m/s^2
- tamaño del cuerpo rígido= 6
- k rígido = 1 * 10 ^ 7 N/m
- k grilla= 2500 N/m
- b grilla= 0,5
- masa por partícula del cuerpo rígido= 0,5 kg
- radio por partícula del cuerpo rígido= 0,15 m
- kn = 1 ∗ 10^ 5 N/m
- kt = 1 * 10 ^ 3 N/m
- separación entre partículas del cuerpo rígido= 0,15 m
- h = 6 m y Θ = 45 °

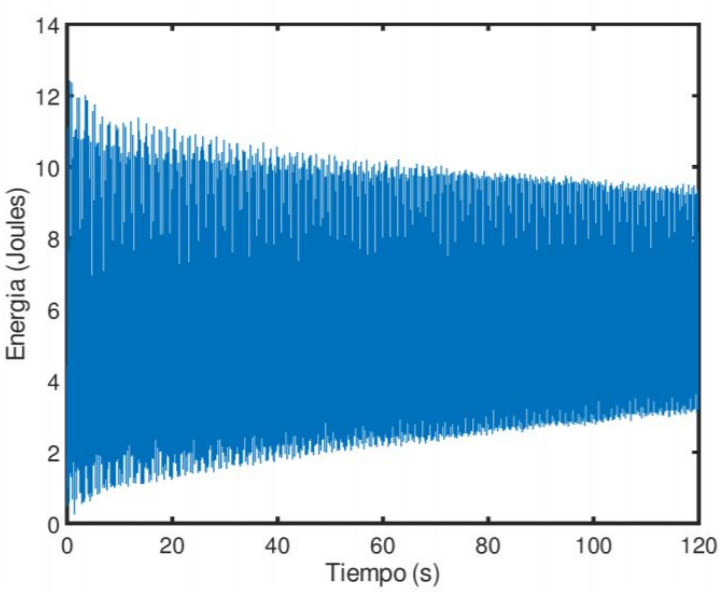
Output



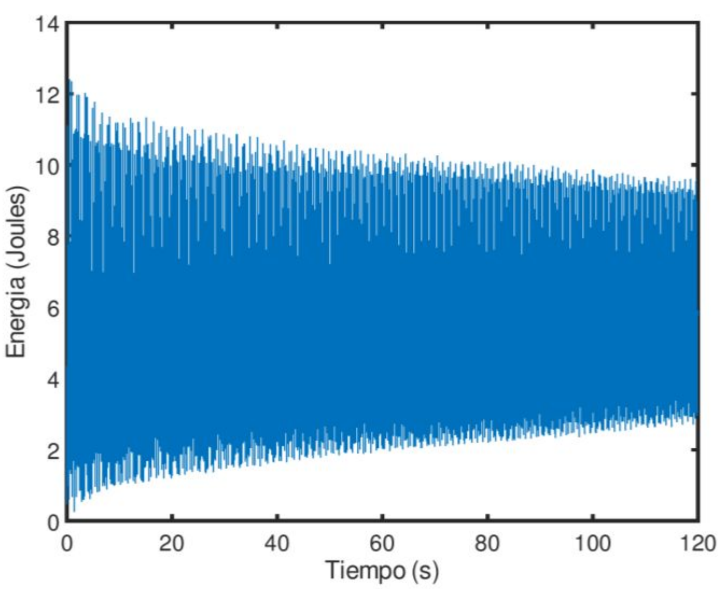
Métodos de integración y conservación de la energía

- t sim = 120s
- Δt = 0,001 s
- filas = 32
- columnas = 32
- separación horizontal en la grilla= 0,05 m
- masa de partículas de la grilla= 0,01 kg
- radio de partículas de la grilla= 0,02 m
- g= 9,81 m/s^2
- k grilla= 3000 N/m
- b grilla= 0 Kg/s

Métodos de integración y conservación de la energía

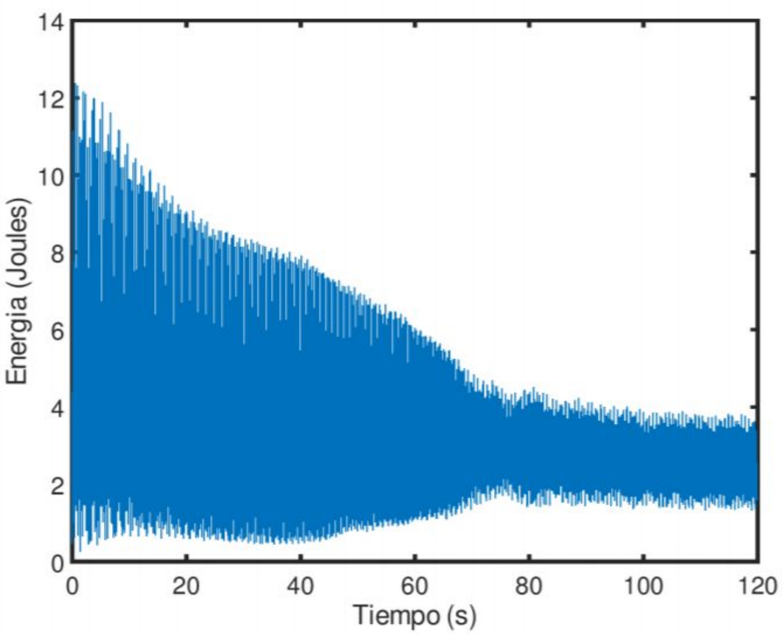


- Euler



- Verlet

Métodos de integración y conservación de la energía



- ***Runge-Kutta 4*** a diferencia de ***Euler*** y ***Verlet*** fue el método que perdió energía más rápidamente

Análisis de la deformación del cuerpo rígido

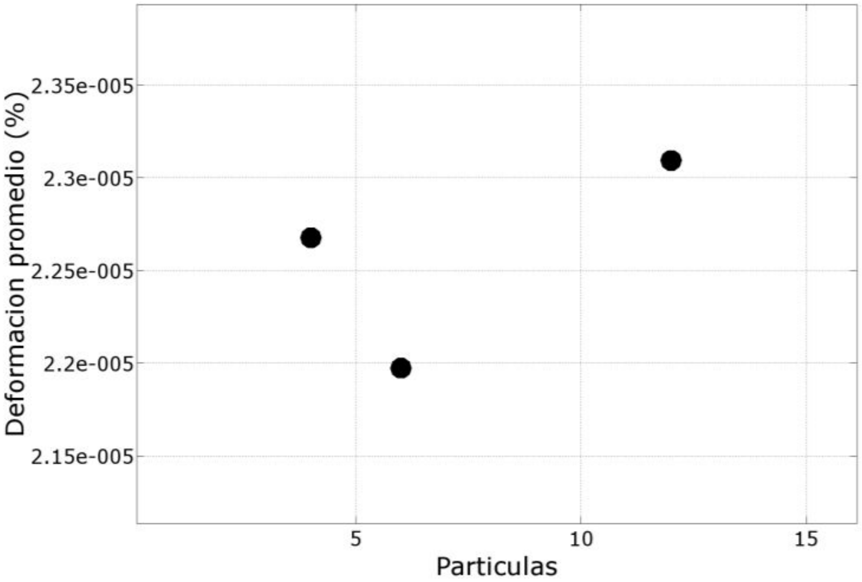
$$Deform = \frac{L(N-1)}{dst(\vec{r_0}, \vec{r_{N-1}})}$$

- Deform promedio
- Deform máxima

Análisis de la deformación del cuerpo rígido

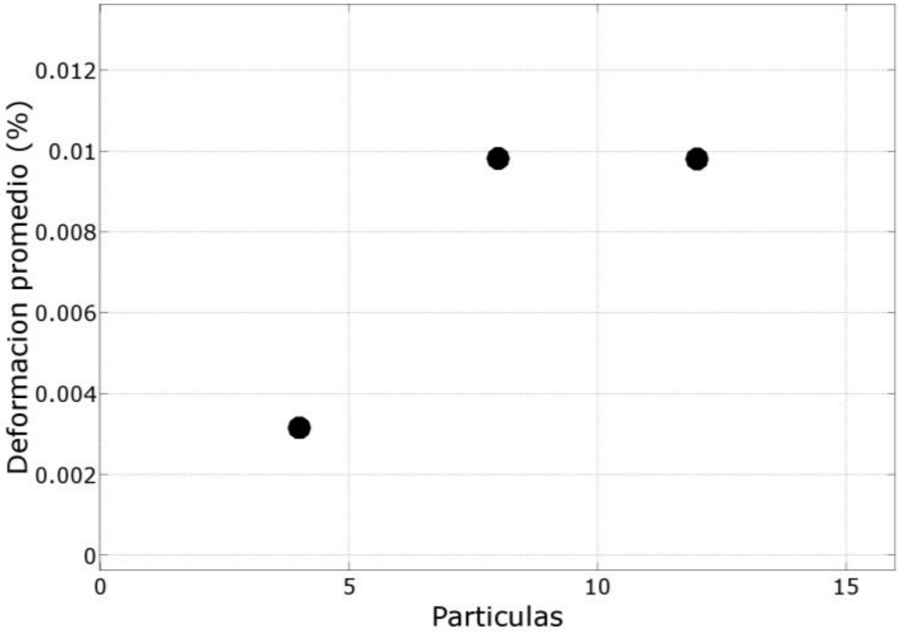
- Método = verlet
- t sim = 5s
- Δt = 0,0001 s
- filas = 60
- columnas = 60
- separación horizontal en la grilla= 0,05 m
- masa de partículas de la grilla= 0,001 kg
- radio de partículas de la grilla= 0,02 m
- g= 9,81 m/s^2
- k grilla= 3000 N/m
- b grilla= 0,5 Kg/s
- masa por partícula del cuerpo rígido= 0,5 kg
- radio por partícula del cuerpo rígido= 0,15 m
- kn = 1 ∗ 10^ 5 N/m
- kt = 1 * 10 ^ 3 N/m
- separación entre partículas del cuerpo rígido= 0,15 m
- h = 3m y Θ = 0 °

Análisis de la deformación del cuerpo rígido



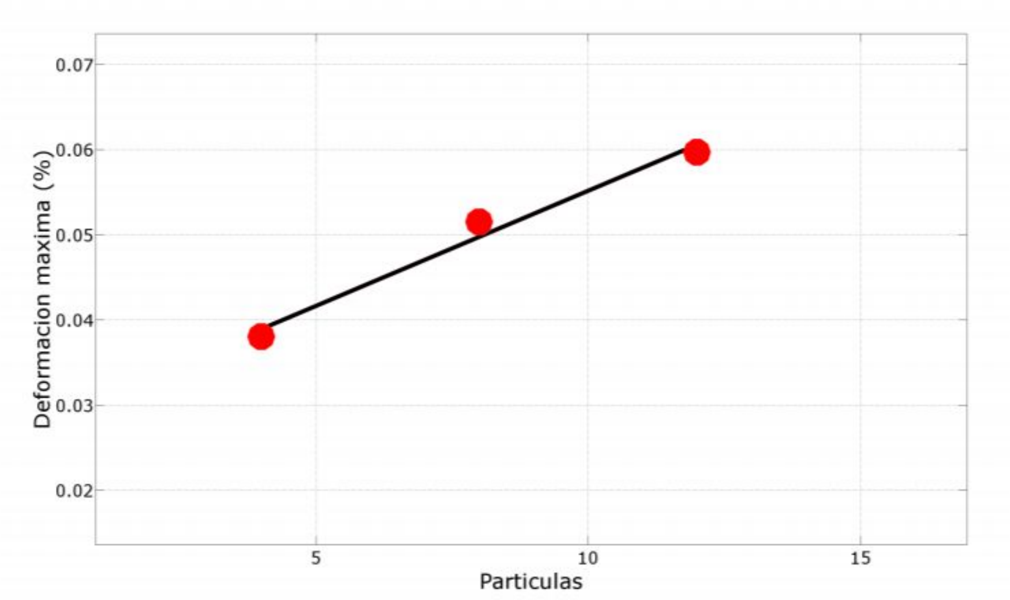
- k_{rigid} = 1∗ 10^ 7 N/m
- 4, 6 y 12 partículas
- Deformación insignificante

Análisis de la deformación del cuerpo rígido



- k_{rigid} = 1∗ 10^ 3 N/m
- 4, 8 y 12 partículas
- Deformación promedio entre 4 y 10%

Análisis de la deformación del cuerpo rígido

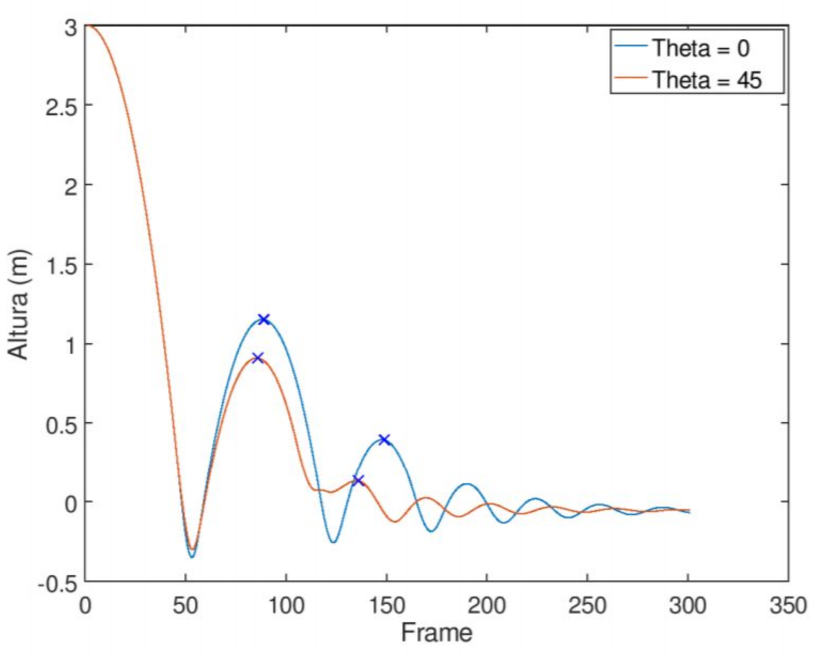


- Tendencia lineal
- $y = 0.002705\,x + 0.0281033$

Alturas alcanzadas variando la orientación del cuerpo rígido

- Método = verlet
- t sim = 5s
- Δt = 0,0001 s
- filas = 64
- columnas = 64
- separación horizontal en la grilla= 0,05 m
- masa de partículas de la grilla= 0,001 kg
- radio de partículas de la grilla= 0,02 m
- g= 9,81 m/s^2
- tamaño del cuerpo rígido= 6
- k rígido = 1 * 10 ^ 7 N/m
- k grilla= 2500 N/m
- b grilla= 0,5 Kg/s
- masa por partícula del cuerpo rígido= 0,5 kg
- radio por partícula del cuerpo rígido= 0,15 m
- kn = 1 ∗ 10^ 5 N/m
- kt = 1 * 10 ^ 3 N/m
- separación entre partículas del cuerpo rígido= 0,15 m

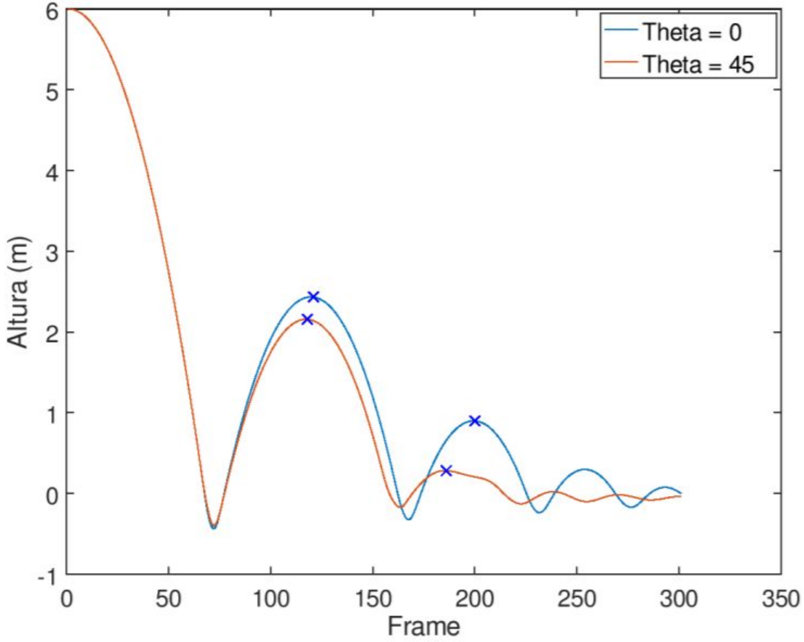
Alturas alcanzadas variando la orientación del cuerpo rígido



- h = 3m

$max(\Theta = 0)$	Idx	$max(\Theta = 45)$	Idx	Rel
1.152 m	89	0.909 m	86	1.267
0.396 m	149	0.133 m	136	2.986

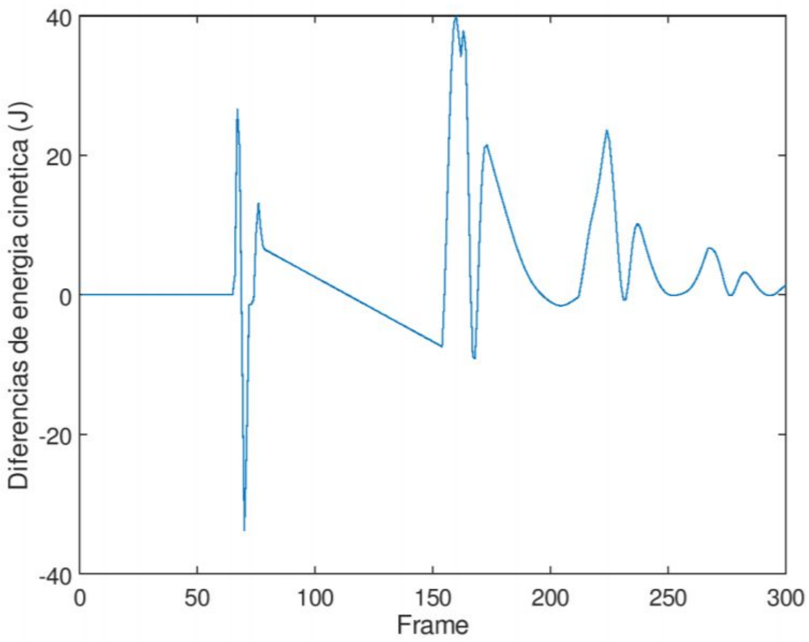
Alturas alcanzadas variando la orientación del cuerpo rígido



- h = 6m

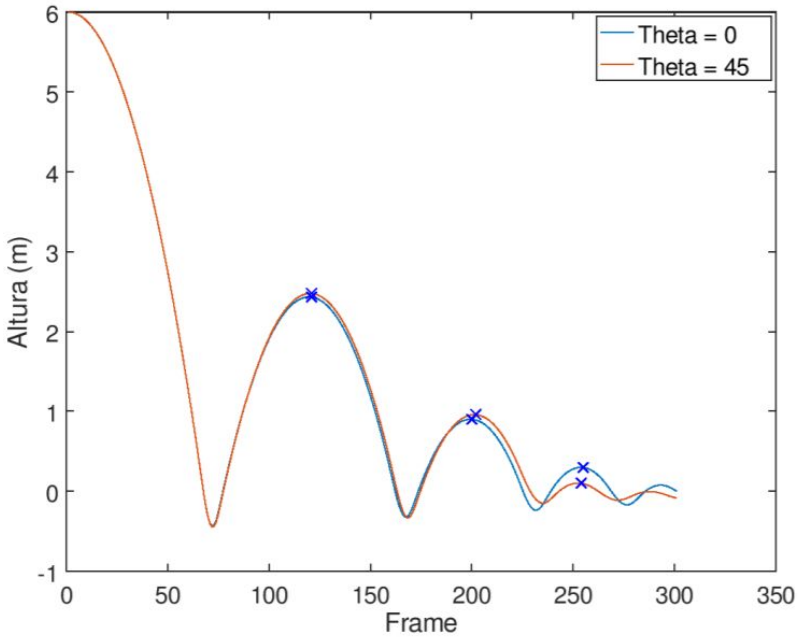
$max(\Theta = 0)$	Idx	$max(\Theta = 45)$	Idx	Rel
2.434 m	121	2.157 m	118	1.128
0.899 m	200	0.283 m	186	3.172

Alturas alcanzadas variando la orientación del cuerpo rígido



- Diferencia en la energía cinética entre el cuerpo no inclinado y el inclinado

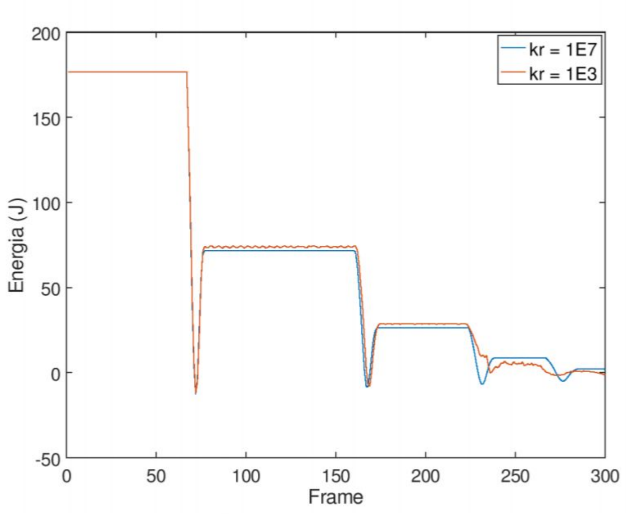
Alturas alcanzadas variando la constante elástica del cuerpo rígido

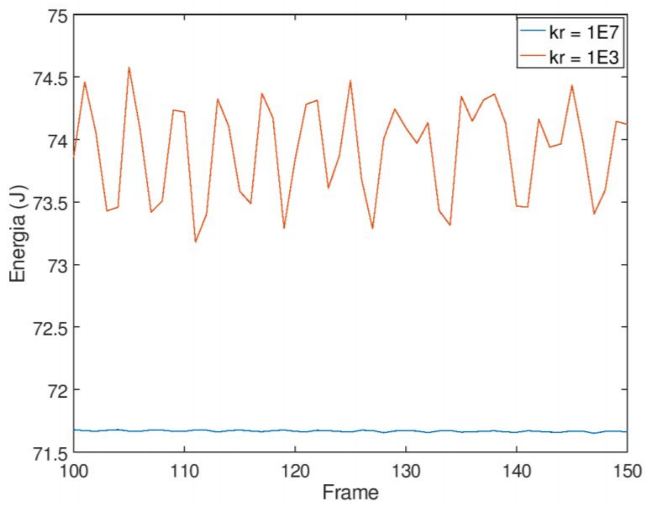


- k rígido = 1 * 10 ^ 3 N/m
- h =6m

$max(k_{1E7})$	Idx	$max(k_{1E3})$	Idx	Rel
2.434 m	121	2.478 m	121	0.982
0.899 m	200	0.958 m	202	0.939
0.297 m	255	0.101 m	254	2.943

Alturas alcanzadas variando la constante elástica del cuerpo rígido





Comparación de la energía total

Conclusiones

- Runge-Kutta 4 tiene mayor pérdida de energía que tanto Velocity Verlet como Euler.
- La deformación del cuerpo rígido, de la forma que se planteó en este trabajo, se ve más afectada por variaciones de la constante k_{rigid} si bien se ha observado una relación de tipo lineal entre esta deformación y dicha constante.
- Se notó que la deformación máxima se ve afectada por la cantidad de partículas de forma directa.
- Se ha comprobado que a iguales alturas iniciales pero diferentes orientaciones, los cuerpos rígidos alcanzan picos de altura diferentes