PRÁCTICA 4: SIMULACIÓN DE OBJETOS DEFORMABLES

Jordi Beltran y Alberto Játiva

SIMULACIÓN Universitat de València

CONTENIDO

INTRODUCCIÓN

Esta práctica consta de dos partes.

La primera, simulará una malla deformable formada por partículas unidas entre si mediante muelles, parecida a una portería que, al interactuar con una esfera en movimiento, trate la colisión con la malla mediante muelles entre la esfera y las partículas de la malla con las que ha colisionado (como se vio en la parte 2 de la práctica pasada).

La segunda parte trata de un mapa de alturas simulando el oleaje del mar mediante distintos tipos de ondas (direccionales, radiales y Gerstner) donde se desarrollará un fenómeno físico. Dicho fenómeno se trata de una pelota de playa flotando sobre la superficie.

PARTE 1. PORTERÍA

CONFIGURACIÓN BASE

Enlace al vídeo: P4 - Portería: Configuración Base

PARÁMETROS DE ENTRADA

Número de partículas	20x20
Distancia entre partículas	40x20
Constante elástica de la malla	100
Constante de amortiguación de la malla	5
Masa de los nodos	0.1
Radio de la pelota	50
Masa de la pelota	3
Velocidad inicial de la pelota	[0, 500, 0]
Distancia inicial	200
Constante elástica de colisión	500
Fuerza máxima de rotura	200
Distancia de activación del muelle de colisión	51
Elongación en reposo del muelle de colisión	51

DIFERENCIAS DE COMPORTAMIENTO

STRUCTURAL: Se unen las partículas en horizontal y vertical mediante muelles a la partícula más próxima (distancia 1). Su elongación en reposo, por tanto, será la distancia de separación entre partículas.

SHEAR: Se unen las partículas en diagonal con las partículas más cercanas (a distancia 1), lo que provoca una forma de “x”. Su elongación en reposo se obtiene mediante Pitágoras, hallando la hipotenusa entre la distancia a las partículas adyacentes.

BEND: Esta malla se forma uniendo las partículas a distancia 2. Por tanto, se creará un muelle cuya elongación en reposo será del doble de la distancia que las separa.

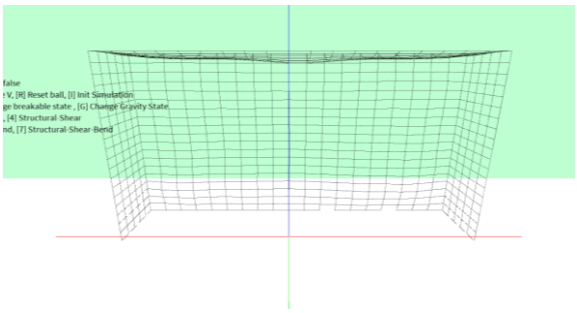
STRUCTURAL – SHEAR: Se combina la distribución ***Structural*** y ***Shear***.

STRUCTURAL – BEND: Se combina la distribución ***Structural*** y ***Bend***.

SHEAR – BEND: Se combina la distribución ***Shear*** y ***Bend***.

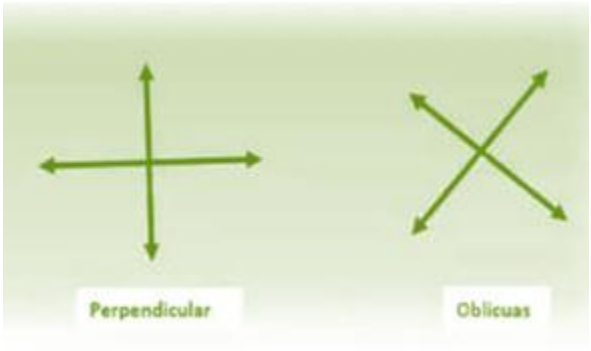
STRUCTURAL – SHEAR – BEND: Se combinan todas las distribuciones principales (***Structural***, ***Shear*** y ***Bend***).

En el caso de simular una portería, la distribución más adecuada, teniendo en cuenta la forma de la red que poseen las porterías convencionales, será la ***Structural***.



En cuanto al comportamiento de cada tipo de disposición, podemos observar cómo aumenta la resistencia a medida que se añaden muelles, siendo las combinadas más resistentes que las disposiciones básicas. Esto se debe a que actúan más fuerzas sobre las partículas, ya que presentan un mayor número de muelles y, por tanto, están influenciadas por el comportamiento de un mayor número de partículas.

Además, la forma de romperse cada estructura es distinta. En el caso de ***Structural*** y ***Bend***, la maya se romperá en perpendicular por los puntos donde la Fuerza elástica ejercida en los muelles sea mayor a una fuerza máxima dada, mientras que en la disposición ***Shear***, se romperán los enlaces de forma oblicua.



En cuanto a las disposiciones combinadas, se verán reflejados los distintos tipos de rotura en base a las disposiciones principales que las forman.

Cabe destacar que, cuanto mayor sea el número de partículas y menor la distancia entre estas, mayor será la influencia de estas sobre la pelota cuando se produzca la colisión, ya que colisionará con un mayor número de partículas, y más realista será la simulación. Este efecto se podrá observar en el segundo vídeo, al aumentar el número de partículas.

VARIANDO PARÁMETROS

Enlace al vídeo: P4 - Portería: Variaciones

VELOCIDAD DE LA PELOTA

Se puede observar cómo, al aumentar demasiado la velocidad de la pelota de la pelota, esta atraviesa la malla. Esto ocurre ya que, durante la colisión con la malla, el muelle que se activa entre las partículas que colisionan y la pelota, produce una fuerza elástica que no es suficiente para invertir la velocidad de la pelota y, por tanto, tras la colisión, la velocidad de la pelota mantiene su dirección sobre el eje y.

MASA DE LA PELOTA

Al aumentar la masa, aumenta la fuerza que se ejerce sobre los muelles de la malla y, por tanto, aumenta la deformación que estos sufren, produciendo una elongación mayor sobre estos y, por tanto, mayor será la fuerza elástica entre la malla y la pelota.

Si las partículas de la malla se distancian lo suficiente (como se observa en la disposición Bend con masa = 10), la pelota podrá cruzar la malla, ya que en ese momento no colisionará con ninguna partícula.

RADIO DE LA PELOTA

Al aumentar el radio, la pelota colisionará con un mayor número de partículas, por lo que aumentaremos el número de muelles de colisión partícula – pelota. Sin embargo, si utilizamos un radio menor a la mitad de la distancia entre las partículas, la pelota pasará por la malla sin colisionar con ninguna partícula, ya que la separación entre las partículas será mayor al tamaño de la pelota.

MASA DE LAS PARTÍCULAS DE LA MALLA

Para una misma fuerza aplicada, al variar la masa de las partículas que forman la malla, variaremos de forma inversa la aceleración que estas sufren debido a la Segunda Ley del movimiento de Newton (a = F/m). Eso quiere decir que, a mayor masa, menor aceleración y, por tanto, menor velocidad y, por tanto, menor diferencia de posiciones con respecto al instante anterior, lo que se traduce en una menor elongación de los muelles que forman la malla.

NÚMERO DE NODOS

Variando el número de nodos, pero respetando el tamaño de la malla (aumentando la distancia entre estos), obtenemos un resultado similar al obtenido debido a la variación del radio de la pelota.

Al haber menos nodos, habrá más distancia entre estos y, por tanto, menor será el número de partículas que colisionarán con la esfera. Si la distancia es mayor al doble del radio de la pelota, la pelota podría atravesar la malla sin colisionar con ninguna partícula.

Cabe destacar que, como al disminuir el radio de la esfera, si disminuimos el número de nodos, se puede plantear un escenario donde la fuerza elástica ejercida por la activación de los muelles pelota – partículas debido a la colisión no sea suficiente para invertir la velocidad de la esfera (simular rebote), permitiendo que la pelota pase a través de la malla.

Así mismo, cuanto mayor sea el número de nodos, menor será la distancia entre estos y, por tanto, al colisionar, mayor será el número de muelles de colisión activos y, por tanto, obtendremos un resultado más realista y solucionaremos el problema anteriormente mencionado.

CONSTANTE ELÁSTICA DE LA MALLA

Variar la constante elástica de la malla se traduce en variar la resistencia a la elongación y compresión de los muelles que la forman con respecto a sus longitudes de reposo.

Al aumentar la constante elástica de los muelles, aumentaremos la resistencia al cambio, obteniendo menores elongaciones.

Al disminuir la constante elástica de los muelles, disminuiremos la resistencia al cambio y, por tanto, los muelles se estirarán más (mayor será la elongación).

CONSTANTE DE AMORTIGUAMIENTO DE LA MALLA

La constante de amortiguamiento se utiliza para que, tras perturbar la malla, esta vuelva a su estado de reposo y no se quede vibrando de forma infinita. Esto se debe a que dicha constante se utiliza para calcular la fuerza de amortiguamiento, una fuerza cuya dirección se opone a la dirección de la fuerza elástica.

Para una Kd = 0, nunca se llegará al reposo ya que la fuerza de amortiguamiento será 0.

Al aumentar la Kd, disminuiremos el tiempo necesario para que la malla vuelva a su estado de reposo, ya que aumentaremos la fuerza de amortiguamiento y, por tanto, habrá una oposición mayor sobre la fuerza elástica.

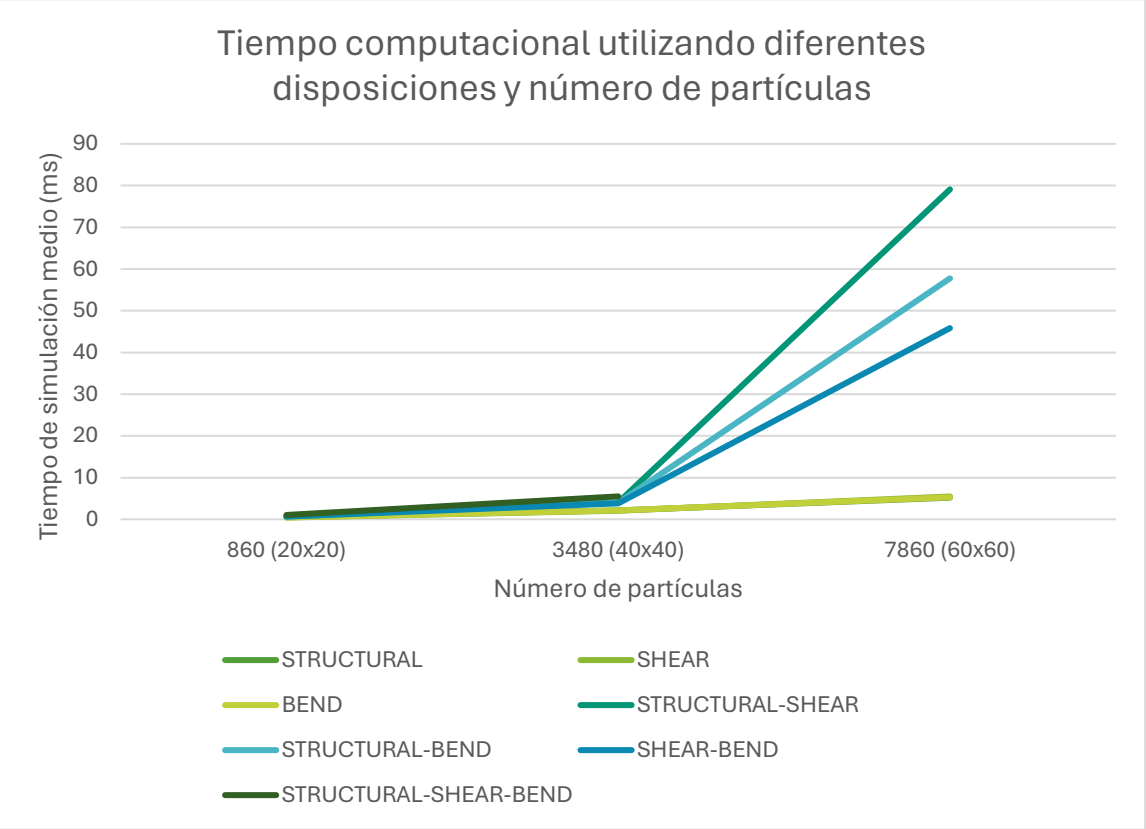
Con Kd muy alta, pueden surgir comportamientos indeseados debido a aproximaciones internas del sistema en el cálculo de variables, así como errores causados por el método de integración utilizado.

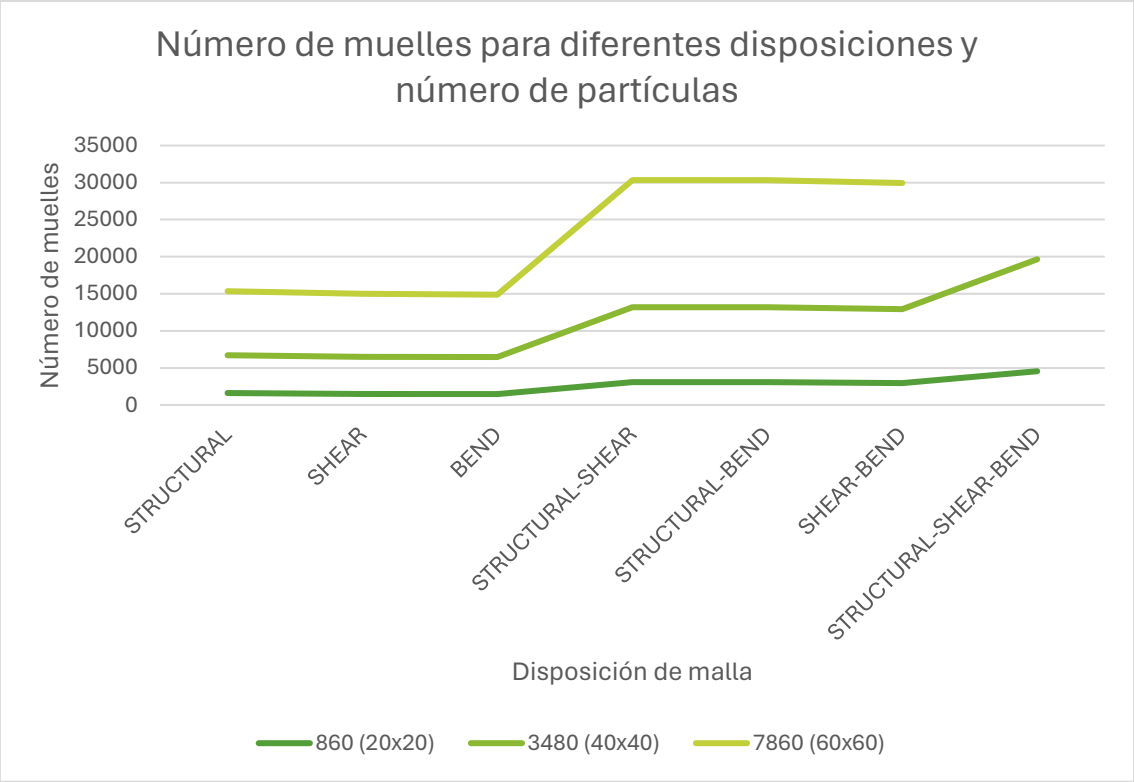
COSTE COMPUTACIONAL DE LAS DIFERENTES DISPOSICIONES

El coste computacional de las diferentes disposiciones viene determinado por el número de muelles que forman la malla.

TIEMPO COMPUTACIONAL			
	Número de partículas		
	860	3480	7860
STRUCTURAL	0,62628035	2,10945155	5,19696183
SHEAR	0,43056113	2,08350862	5,47542135
BEND	0,46146242	2,10411542	5,2934735
STRUCTURAL-SHEAR	0,76304267	4,08222115	79,090465
STRUCTURAL-BEND	0,69336718	3,96077457	57,7663443
SHEAR-BEND	0,6550887	3,82030541	45,8365383
STRUCTURAL-SHEAR-BEND	1,04373362	5,49336451	

NÚMERO DE MUELLES			
	Número de partículas		
	860	3480	7860
STRUCTURAL	1597	6713	15340
SHEAR	1482	6474	14986
BEND	1474	6466	14878
STRUCTURAL-SHEAR	3079	13187	30335
STRUCTURAL-BEND	3071	13179	30327
SHEAR-BEND	2956	12940	29964
STRUCTURAL-SHEAR-BEND	4553	19653	





Como se puede observar en los datos obtenidos de las diferentes simulaciones plasmados en los gráficos, los muelles creados varían según la disposición, pero podemos destacar que:

- **Para disposiciones básicas (*Structural*, *Shear* y *Bend*),** se crea alrededor de 2*número de partículas. Esto se debe a que, de cada partícula menos de los extremos se generan dos muelles. (Órden de 2N, siendo N el número de nodos)
- **Para disposiciones cuyo origen se basa en la unión de dos disposiciones básicas**, se crea la suma de ambas disposiciones y, por tanto, por cada partícula se crean alrededor de 4 muelles (4 * número de partículas), excepto en los extremos (como se ha mencionado anteriormente). (Órden de 4N, siendo N el número de nodos)
- **Para la disposición *Structural – Shear – Bend***, se genera un total aproximado de 6 muelles por partícula (la suma de todas las disposiciones básicas). (Órden de 6N, siendo N el número de nodos)

Por tanto, podemos concluir que, a medida que se suman disposiciones y se incrementa el número de nodos, se aumenta de forma proporcional el coste computacional que supone comprobar y actualizar el estado de cada uno de los muelles debido a que se debe recorrer todo el array, array que puede ser el doble, cuádruple o séxtuple que el número de nodos de la malla según la disposición utilizada, siendo las más “baratas” computacionalmente ***Structural*, *Shear* y *Bend*,** con un coste muy similar.

PARTE 2. OLAS

En la segunda parte de esta práctica, hemos llevado a cabo la simulación cinemática de olas utilizando un mapa de alturas, también conocido como "*Height Map*". Este mapa es una estructura de datos que consta de un conjunto de puntos distribuidos en un plano, donde cada punto tiene coordenadas (x, y, z).

TIPOS DE OLAS

Olas direccionales: En este caso, contamos con todos los parámetros de la onda y el vector D, que indica la dirección deseada de la onda. Solo generamos variación de movimiento en el eje Y (vertical). Para estas olas, en la ecuación de la onda, multiplicamos el producto escalar entre la posición de cada uno de los vértices del mapa de alturas y el vector D por el parámetro W (frecuencia angular).

```
class WaveDirectional extends Wave
{
  public WaveDirectional(float _a,PVector _srcDir, float _L, float _C)
  {
    super(_a, _srcDir, _L, _C);
  }

  public PVector getVariation(float x, float y, float z, float time)
  {
    // Solo existe variación en el eje Y (vertical)
    tmp.x = 0;
    tmp.z = 0;

    PVector dir = D.copy();
    dir.normalize();

    float p_Escalar = PVector.dot(new PVector(x,y,z),dir);

    tmp.y = -A * sin(p_Escalar*W + phi * time);

    return tmp;
  }
}
```

Olas radiales: Son casi idénticas a las direccionales, pero en este caso el vector D no indica la dirección deseada de la onda, sino que señala el epicentro de la misma. Para estas olas, en la ecuación de la onda, multiplicamos la distancia entre los vértices del mapa de alturas y el vector D por el parámetro W (frecuencia angular).

```
class WaveRadial extends Wave
{
  public WaveRadial(float _a,PVector _srcDir, float _L, float _C)
  {
    super(_a, _srcDir, _L, _C);
    _srcDir.normalize();
  }

  public PVector getVariation(float x, float y, float z, float time)
  {
    // Solo existe variación en el eje Y (vertical)
    float d_ep = dist(x, z, D.x, D.z);

    tmp.x = 0;
    tmp.y = -A * sin(W * d_ep + phi * time);
    tmp.z = 0;
```

```
17.     return tmp;
18.   }
19. }
```

Olas Gerstner: Los modelos anteriores son demasiado perfectos y regulares. Aquí es donde surge este tipo de onda, que se deriva de las ondas direccionales, pero en este caso hay movimiento en todos los ejes X, Y, Z. Introducimos el parámetro Q, que controla la inclinación de la cresta de la onda. En nuestro caso, hemos limitado Q entre 0 y 1.

```
1. class WaveGerstner extends Wave
2. {
3.   public WaveGerstner(float _a,PVector _srcDir, float _L, float _C)
4.   {
5.     super(_a, _srcDir, _L, _C);
6.   }
7.
8.   public PVector getVariation(float x, float y, float z, float time)
9.   {
10.    // Recordar que Y es el eje vertical
11.    PVector dir = D.copy();
12.    dir.normalize();
13.
14.    tmp.x = Q * A * dir.x * cos(W*(dir.x*x + dir.z*z) + time*phi);
15.    tmp.z = Q * A * dir.z * cos(W*(dir.x*x + dir.z*z) + time*phi);
16
17.    tmp.y = -A * sin(W*(dir.x*x + dir.z*z) + time*phi);
18.
19.    return tmp;
20.  }
21. }
```

SIMULACIÓN REALISTA DE UN FÉNOMENO NATURAL MEDIANTE ONDAS

Para la simulación de olas, utilizamos el mapa de alturas que contiene un *ArrayList* de objetos Wave. Añadimos olas de cualquiera de los tipos anteriores a este *ArrayList* y observamos cómo se introduce la variación punto a punto del mapa de alturas. Esto se realiza mediante el método *update* de *HeightMap*.

Para lograr una simulación realista de algún fenómeno natural mediante ondas, hemos añadido a nuestra simulación una clase Ball. Con ella, creamos una bola en la superficie del *HeightMap*. A partir de un método que hemos añadido a la clase *HeightMap*, obtenemos la altura según la posición de la pelota en el plano XZ y se la aplicamos a la pelota, además de hacerla rotar sobre sí misma, para un efecto más realista.

Además, se han adaptado los datos para hacer dos simulaciones de mar. La primera, simula un mar en calma mientras que, la segunda, simula un mar con oleaje. Para ello, se ha reducido y aumentado tanto la amplitud como la longitud de onda y su velocidad.

	MAR EN CALMA	**MAR CON OLEAJE**
AMPLITUD (A)	random(2f) + 50f	random(2f) + 50f
LONGITUD DE ONDA (λ)	A * (80 + random(10f))	A * (10 + random(10f))
VELOCIDAD (V)	λ / (1 + random(2f))	λ / (1 + random(3f))
ENLACE AL VÍDEO	https://youtu.be/5KltfujzeY0	https://youtu.be/SBsK4HTYDlc