MCBtec

Mas información en
www.mcbtec.com

# INTRODUCCIÓN A LA SIMULACION POR ORDENADOR

## Indice:



## Objetivo de este texto.

Con este texto, solo pretendo hacer una introducción muy básica e intuitiva a la simulación por ordenador, utilizando las ecuaciones fundamentales de la dinámica. Para entender este texto se necesitan conocimientos básicos de física, trigonometría y programación de ordenadores.

## Simulación por ordenador.

La simulación por ordenador nos permite modelar sistemas complejos en el ordenador y poder experimentar con ellos para poder comprenderlos. Por ejemplo, podemos simular en el ordenador lo que pasaría si un viento de 150 Km/h chocara contra la estructura de un rascacielos sin necesidad de construirlo y esperar que llegue un huracán. Esta claro, que el ahorro en tiempo, dinero y sobre todo en vidas humanas ( en el caso de que el edificio se cayera ) justifica con mucho simular el sistema.

La utilización del ordenador y su capacidad de calculo, permite calcular las ecuaciones diferenciales ( que casi siempre intervienen en el comportamiento de los sistema complejos) sin necesidad de resolverlas, además de poder analizar y visualizar los datos e introducir perturbaciones en el sistema para poder analizar su comportamiento.

La simulación por ordenador se puede aplicar a casi cualquier rama de la ciencia; ingeniería, genética, historia, sociología, historia, etc..., de todas formas, en esta pequeña introducción nos vamos a centrar en el estudio de sistemas dinámicos que son más cotidianos y los mas sencillos matemáticamente hablando.

## Dinámica y simulación.

Básicamente para realizar cualquier problema de dinámica, tenemos las siguientes ecuaciones en forma de derivadas:

$$v = \frac{dx}{dt} \quad a = \frac{dv}{dt} \quad f = ma$$

Como ya hemos dicho la capacidad de calculo de los ordenadores nos permite hacer el calculo de los *dt* uno a uno, sin necesidad de calcular la derivada real de la ecuación, por lo que podemos poner las ecuaciones de arriba de la siguiente forma:

$$dx = v \cdot dt \quad dv = a \cdot dt$$

Esto quiere decir que el móvil que va a velocidad *v* m/s incrementara su posición en *dx* metros durante el tiempo *dt*, si en el bucle acumulamos todos estos *dx* en una variable ( $Px = \sum dx$ ) esta contendrá la posición *x* del móvil para un tiempo $t = \sum dt$.

La realización practica del programa la haríamos mediante el uso de un bucle en el que incrementaremos el tiempo *t* en la cantidad *dt* que nosotros creamos más conveniente ( por ejemplo 1 ms = $10^{-3}$ s) en cada iteración, de tal forma que si quisiéramos saber la posición y la velocidad de un móvil con una aceleración constante en el eje x de 4 m/s en $t_0$, el bucle sería el siguiente:

```
// Inicialización de variables
dt = 10^-3;
a_x = 4 m/s;
v_x= 0; p_x= 0; t = 0; // Velocidad inicial y posición inicial = 0
// Bucle de tiempo, cada iteración el tiempo pasa en dt segundos.
Repite
        v_x = v_x+a_x*dt;        // Acumula en Vx el incremento de velocidad
        p_x = p_x+v_x*dt;        // Acumula en px el incremento de posición
        t = t+dt;                // Incrementa el tiempo en dt segundos
Hasta t >= t_0;                  // Si t es mayor o igual que t_0 se acaba el bucle
Imprime(v_x, p_x );              // Imprime el valor de v_x y p_x en t_0
```

Cada iteración del bucle indica que el tiempo se ha incrementado en $10^{-3}$ s, es decir 1 milisegundos y las ecuaciones acumulan en $v_x$ y $p_x$ lo que ha aumentado la velocidad y la posición en ese *dt*.

Este ejemplo no tiene mucho sentido ya que el mismo resultado lo podríamos obtener mediante $v_x = a_x * t_0$ y $p_x = 1/2 * ( v_x * t_0)$, pero sería muy útil si la aceleración variara con el tiempo, ya que podríamos crear un array de datos o una ecuación que fuera suministrado los valores de $a_x$ en función de $t$ para cada $dt$, evitando así tener que calcular la derivada de la aceleración para calcular la velocidad, derivada que puede ser muy difícil o incluso imposible de encontrar. Esta es una de las razones por la que los programas de simulación por ordenador tienen tanta utilidad.

En este ejemplo tenemos un objeto que se mueve en una sola dirección del espacio, si queremos calcular el movimiento en dos o tres dimensiones solo necesitaremos obtener las correspondientes $v_x$, $v_y$, $v_z$, $p_x$, $p_y$ y $p_z$ de forma independiente mediante las aceleraciones $a_x$, $a_y$ y $a_z$. Si tenemos $a$ en forma de vector lo podremos descomponer en sus componentes $a_x$, $a_y$ y $a_z$ utilizando trigonometría ( ver ejemplo disparo de proyectil).

Si lo que tenemos es una fuerza, que también puede variar con el tiempo, podemos calcular la aceleración directamente mediante la ecuación $a(t)=f(t)/m$ y aplicarla al programa.

Para elegir $dt$, es importante saber que su valor determina el numero de iteraciones que hará el bucle, de tal modo que si queremos analizar el sistema durante un segundo y elegimos 1 μs para $dt$ tendremos que el bucle se ejecutará un millón de veces (1s=1.000.000 μs). Cuanto más pequeño sea dt mejor será en análisis del sistema pero la duración del proceso de simulación se alargará, en sistemas dinámicos podemos utilizar el rango de entre 1 μs y 1 ms.

Lo mejor es que estudiemos ejemplos prácticos.

# Ejemplo disparo de un proyectil.

En este caso vamos a simular el disparo de un proyectil en dos dimensiones. Este tiene una velocidad inicial $v_i$ y un ángulo de inclinación de $\alpha$.

Como el sistema funciona en 2 dimensión tenemos que plantear dos sistemas de ecuaciones independientes uno para el eje X y otro para el eje Y que solo compartirán una variable, el tiempo $t$. Elegimos $dt = 10^{-3}$ s, es decir, cada iteración del bucle indicará que el tiempo se ha incrementado en 1 ms.

Primero tendremos que descomponer $v_i$ , en sus componentes para cada eje $v_{xi}$ y $v_{yi}$, para ello utilizamos trigonometría y obtendremos que:

$$v_{xi} = v_i \cos(\alpha)$$
$$v_{yi} = v_i sen(\alpha)$$

Vamos a plantear como seria el programa que calculara la posición en el eje X e Y del proyectil en $t = t_0$, para ello vamos a suponer un ángulo inicial de 30º (π/6 rad) y una velocidad de modulo 40 m/s.

// Incialización de variables
$dt = 10^{-3}$ ;
$v_i = 40$ m/s ;
$\alpha = \pi/6$ rad ; // Para que el calculo sea correcto α tiene que estar en radianes
$v_{xi} = v_i \cos(\alpha)$ ; $v_{yi} = v_i \operatorname{sen}(\alpha)$ ;
$p_y = 0$; $p_x = 0$; $t = 0$;
// Bucle de tiempo, cada iteración el tiempo pasa en dt segundos.
Repite
    $v_{yi} = v_{yi} - g*dt$;  // Acción de la gravedad sobre el eje y
    $p_x = p_x + v_{xi}*dt$;
    $p_y = p_y + v_{yi}*dt$;
    $t = t + dt$;
Hasta $t >= t_0$;
Imprime($p_x$, $p_y$);  // Imprime el valor de $p_x$ y $p_y$ en $t_0$

En cada iteración del bucle obtenemos la posición del proyectil para cada incremento del tiempo, esto nos permitiría dibujar en la pantalla del ordenador la posición del proyectil en función del tiempo y obtener y procesar toda la información de la trayectoria.

En la pagina web www.mcbtec.com tienes el programa de este ejercicio.

## Ejemplo sistema con muelle.

En el caso anterior solo actuaba la fuerza de la gravedad (g) que es constante con una aceleración igual a 9,8 m/s, en este ejemplo vamos a estudiar un sistema con muelle en el que la fuerza ( y por tanto la aceleración del sistema a=f/m) no es constante, sino que depende de la longitud en la se ha comprimido o estirado el muelle.

En este ejemplo estudiaremos:

- Simulación en dos coordenadas espaciales ( X e Y ).
- Como descomponer una velocidad ( lo mismo se puede aplicar a una aceleración ) en sus componentes X eY.
- Bucle de simulación con aceleración constante.

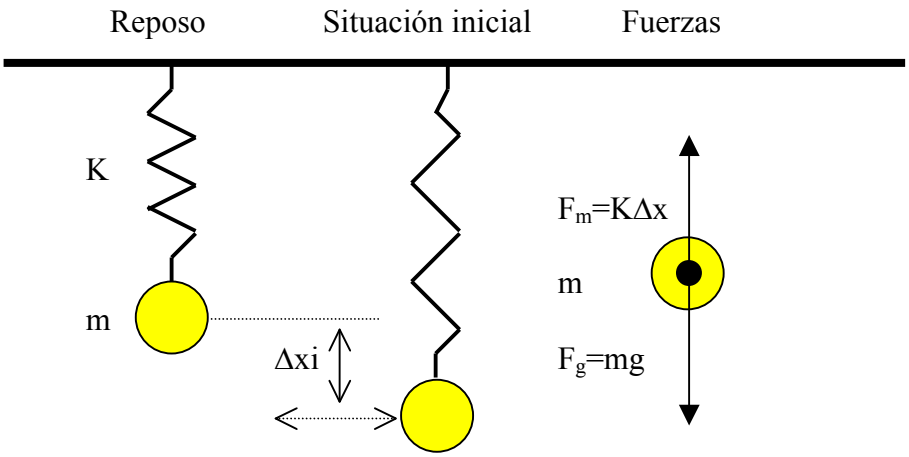
Los muelles son elementos mecánicos que generan una fuerza que se opone a la compresión/ estiramiento de este, según la ecuación:

$$f_m = K \cdot \Delta x$$

La constante K, es propia de cada muelle y nos indica la fuerza que ejerce el muelle por unidad de longitud comprimido o estirado. El valor de $\Delta x$ es la longitud comprimida o estirada ( según el signo) del muelle. Supongamos un sistema en el que tenemos un objeto de masa *m* suspendido de un muelle de constante K si desplazamos el objeto de su posición de reposo una distancia $\Delta xi$ y soltamos el objeto, este empezara a

oscila por efecto de la gravedad y la fuerza del muelle. Veamos como haríamos un programa de simulación de este sistema.

Un esquema del sistema sería el siguiente:




Como en casos anteriores elegiremos un *dt* que se adapte bien a nuestras necesidades, por ejemplo, $10^{-3}$s = 1ms. El programa que queremos hacer nos tiene que suministrar la velocidad y la posición del objeto en $t = t_0$. El programa seria el siguiente:

```
// Incialización de variables
dt = 10^-3 ;
K= 0.1;          // Valor de la constante del muelle
vm = 0 m/s ;     // Velocidad del móvil inicial 0
py= -Δxi;        // Posición inicial del móvil -Δxi
t = 0;           // Tiempo = 0
// Bucle de tiempo, cada iteración el tiempo pasa en dt segundos.
Repite
        am=( -K* py+mg)/m;  // La aceleración = (Σ fuerzas) / masa
        vm= vm+ am*dt;      // K* py es negativa porque se opone al movimiento
        py = py+vm*dt;
        t = t+dt;
Hasta t >= t0;
Imprime(vm, py);            // Imprime el valor de vm y py en t0
```

En la pagina web www.mcbtec.com tienes el programa de este ejercicio.

## Visualización en tiempo real.

Una vez que tenemos el programa para simular un determinado sistema físico, nos puede interesar representar gráficamente el sistema en movimiento en tiempo real, es decir, como si estuviéramos viéndolo físicamente. Para esto tenemos que conseguir que el ordenador represente en la pantalla las distintas posiciones del objeto en el tiempo que le corresponde.

Si suponemos que el ojo humano necesita como mínimo 24 imágenes por segundo para percibir correctamente una secuencia de video, podemos crear una función en el ordenador que nos genere un retraso de 1/24 s aproximadamente y hacer que el bucle del programa de simulación almacene en un array el valor de la posición del objeto que queramos representar cada 1/24s de tal forma que si hacemos un bucle del tipo:

```
// Incialización de variables
Array Posiciones [NP];          // Array con NP posiciones
n = 0;                          // Posición dentro del array inicialmente = 0
// Bucle de dibujo en pantalla.
Repite
        DibujaObjetoEnPosicion(Posiciones[n]);
        Espera1/24Segundos;
        n = n+1;
Hasta n >NP;
```

En el array de posiciones (*Posiciones[NP])* almacenaremos la posición del objeto a visualizar cada 1/24 s de tal forma que en Posiciones[0] tendremos la posición del objeto en t=1/24s en Posiciones[1] tendremos la posición en t=2*(1/24)s en Posiciones[3] tendremos la posición en t=3*(1/24)s y así hasta Posiciones[NP] en la que estará la posición en t=(NP+1)*(1/24)s.

El problema que tiene el programa anterior es que el windows es un sistema multitarea y por tanto es muy difícil establecer esperas de tiempo precisas, la mejor manera de realizar esto seria crear un programa que nos generara las imágenes de las distintas posiciones en ficheros que luego editaríamos con un programa grafico adecuado para generar un fichero tipo *avi* o similar.