



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 N° 29 – ABRIL DE 2010

“ALGORITMOS MATEMÁTICOS EN FÍSICA”

AUTORÍA YAMIL BUMEDIEN FARIS
TEMÁTICA NUEVAS TECNOLOGÍAS EN EL AULA
ETAPA BACHILLERATO

Resumen

Este artículo se centra en la “traducción” de las ecuaciones de la cinemática en algoritmos matemáticos y, posteriormente, en la implementación de éstos en cualquier lenguaje de programación. En este caso implementamos los algoritmos en lenguaje PHP.

Por lo tanto se trata de hacer uso de los conocimientos de programación adquiridos en la asignatura TIC de 1º Bachillerato y aplicarlos para resolver ecuaciones de la física.

Palabras clave

Algoritmo.

Cinemática.

Pseudocódigo.

Lenguaje de programación.

1. ¿QUÉ ES UN ALGORITMO?

Empezaremos este artículo definiendo la palabra **algoritmo**¹, un algoritmo, proveniente del latín *dixit algorithmus* y éste a su vez del matemático persa al-Jwarizmi, es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución. De esta forma los algoritmos se comportan como funciones matemáticas, es decir reciben una entrada y la transforman en una salida, sin embargo, no todo lo que transforma una entrada en una salida lo podemos considerar un algoritmo; para que sea considerado como tal, las instrucciones que lo componen han de ser ordenadas, definidas y finitas.

¹ Definición obtenida de Wikipedia



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 29 – ABRIL DE 2010

Hay que tener especial cuidado con *no confundir* un algoritmo con un programa, mientras que un algoritmo es una especificación de pasos a seguir orientados a la obtención de una solución, un programa es un conjunto de pasos especificados en un determinado lenguaje de programación y para una determinada computadora.

Pero entonces, ¿cómo podemos expresar un algoritmo?, un algoritmo puede ser expresado de muchas maneras, podemos utilizar el propio lenguaje natural, pseudocódigo, diagramas de flujo y lenguajes de programación entre otros. Las descripciones en lenguaje natural tienden a ser ambiguas y extensas. El usar pseudocódigo y diagramas de flujo evita muchas ambigüedades del lenguaje natural. Dichas expresiones son formas más estructuradas para representar algoritmos; no obstante, se mantienen independientes de un lenguaje de programación específico.

La descripción de un algoritmo usualmente se hace en tres niveles:

1. **Descripción de alto nivel.** Se establece el problema, se selecciona un modelo matemático y se explica el algoritmo de manera verbal, posiblemente con ilustraciones y omitiendo detalles.
2. **Descripción formal.** Se usa pseudocódigo para describir la secuencia de pasos que encuentran la solución.
3. **Implementación.** Se muestra el algoritmo expresado en un lenguaje de programación específico o algún objeto capaz de llevar a cabo instrucciones.

En nuestro caso, para la descripción de algunos problemas físicos, utilizaremos una descripción formal (pseudocódigo) apoyándonos en el lenguaje natural y lo implementaremos mediante el lenguaje *PHP*, es obvio que cualquiera de los algoritmos que expondremos se puede implementar en cualquier lenguaje de programación, solo que, a modo de ejemplo, lo implementamos mediante scripts *PHP*.

¿Por qué utilizar pseudocódigo?, muy fácil, porque el pseudocódigo se asemeja a un lenguaje de programación pero con algunas convenciones del lenguaje natural. Tiene varias ventajas con respecto a los diagramas de flujo, entre las que destaca el poco espacio que se requiere para representar instrucciones complejas y además el pseudocódigo no está regido por ningún estándar. Pseudo viene de falso y por ende es un código al que aunque es entendible no se aplica al proceso que debe realizar la maquina.

2. UN POCO DE CINEMÁTICA...

Antes de todo, hay que preguntarse ¿qué es la cinemática?, pues bien, la cinemática es la parte de la mecánica que estudia el movimiento **sin tener en cuenta las causas que lo producen**; pero ¿y qué es la mecánica?, bueno, la mecánica se podría definir como la parte de física que se dedica al estudio del movimiento.

Vayamos acotando, entonces, los problemas que van a ser objeto de nuestro estudio; en nuestro caso, vamos a estudiar el **movimiento rectilíneo y uniforme**, a partir de ahora haremos mención a este tipo



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 N° 29 – ABRIL DE 2010

de movimiento de la siguiente manera *MRU*, y el **movimiento rectilíneo uniformemente variado**, a partir de ahora *MRUV*.

El MRU se caracteriza por ser la trayectoria rectilínea y la velocidad constante por lo que sus ecuaciones del movimiento son las siguientes (supongamos el movimiento en el eje X):

$$v = k$$
$$x = x_0 + v \cdot t$$

donde v es la velocidad, k es el valor de la velocidad que será constante en el tiempo, x la posición en cada instante y x_0 es la posición inicial.

El MRUV se caracteriza por ser la trayectoria rectilínea y por ser las variaciones de la velocidad uniformes, es decir, que la aceleración es constante. Este tipo de movimiento puede ser, a su vez, de dos tipos en función de que la velocidad vaya aumentando o disminuyendo, al primero se le conoce como movimiento rectilíneo uniformemente acelerado (MRUA) y al segundo de los casos se le conoce como movimiento rectilíneo uniformemente retardado (MRUR) o, simplemente, movimiento de frenada.

Las ecuaciones del movimiento son idénticas en ambos casos, sólo difieren en un signo que es el que indica si es acelerado o de frenada, veámoslas:

$$v = v_0 \pm a \cdot t$$
$$x = x_0 + v_0 \cdot t \pm \frac{1}{2} \cdot a \cdot t^2$$

donde v es la velocidad en cada instante, v_0 la velocidad inicial, x la posición en cada instante, x_0 la posición inicial y a es la aceleración que permanece constante en el tiempo. El signo $+$ o $-$ delante de la aceleración indica si estamos ante MRUA (signo $+$) o ante el MRUR (signo $-$).

3. ...Y AHORA LOS ALGORITMOS

Para el MRU, el algoritmo escrito de manera formal, es decir, en pseudocódigo tiene el siguiente aspecto:

Algoritmo: Descripción del movimiento rectilíneo y uniforme
<pre> función MRU() para (t = 0 hasta t = cualquier_valor) hacer /*v es la velocidad que es constante y x_inicial es la posición inicial*/ x(t) = x_inicial + v·t // Δ t son incrementos de tiempo. t = t+ Δ t fin de hacer devolver x(t) </pre>

Discutamos un poco este algoritmo; en primer lugar tenemos una función dentro de la cual está definido un bucle desde un tiempo $t = 0$ hasta un tiempo indicado por nosotros, $t = \text{cualquier_valor}$, en cada paso suma un incremento Δt y recoge esos valores en un array, es decir, es como si tuviéramos un vector donde cada coordenada representa el valor de la posición en cada instante t . Estos datos luego pueden ser utilizados para representarse gráficamente o simplemente en una tabla con dos columnas (posición, tiempo).

Una **implementación** de este algoritmo en PHP puede ser la siguiente:

```

<?
//Consideramos que la velocidad es 50 m/s y
//la posición inicial es 100 m del origen

$velocidad=50;
$x_inicial=100;

//Hallamos los valores de la posición desde t = 0s hasta t = 10s

    for($t=0; $t<=10; $t++){
        $x[$t]=$x_inicial + ($velocidad*$t);
    }

//Lee el array $x y representa en una tabla los valores de tiempo y posición
//En este caso el array tiene 11 elementos

echo "<table border=2 align= center><tr><td><B>Tiempo</B></td><td><B>Posicion</B></td></tr>";
    
```



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 29 – ABRIL DE 2010

```
foreach ($x as $tiempo=>$posicion){
echo "<tr><td>".$tiempo."</td><td>".$posicion."</td></tr>";
}

echo "</table>";
?>
```

Os animo a que instaléis el módulo PHP y Apache y probéis estos scripts, ¡funcionan!, (en Internet podéis encontrar todas las instrucciones para instalar estos módulos), además sería interesante que los datos que recojáis los representéis gráficamente en una hoja de cálculo, por ejemplo Excel, y veáis los resultados. Si alguien se atreve puede realizar un script o modificar el ejemplo de arriba para que el resultado sea una gráfica con los puntos obtenidos.

Para el MRUV, podemos representar dos algoritmos, uno para el MRUA y otro para el MRUR, pero como son idénticos sólo representaremos el algoritmo para el MRUA e indicaremos en qué se diferencia del de MRUR, este algoritmo en pseudocódigo quedaría de la siguiente forma:

```
Algoritmo: Descripción del movimiento rectilíneo uniformemente acelerado
función MRUA()

  para (t = 0 hasta t = cualquier_valor) hacer

    /*a es la aceleración que es constante, v_inicial es la velocidad
    inicial y x_inicial es la posición inicial.*/

    /*si ponemos el signo - antes de la aceleración (a), estaríamos
    ante el MRUR.*/

    v(t) = v_inicial + a·t
    x(t) = x_inicial + v_inicial·t + ½·a·t²

    // Δt son incrementos de tiempo.

    t = t+ Δt

  fin de hacer

devolver x(t) y v(t)
```



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 29 – ABRIL DE 2010

Para este caso, nuestro algoritmo tiene dos salidas $x(t)$ y $v(t)$ que serán dos arrays o vectores donde cada coordenada de estos vectores serán la posición y la velocidad en cada instante t respectivamente. La **implementación** de este algoritmo en PHP es la siguiente:

```
<?
//Consideramos que la velocidad inicial es 50 m/s
//la posición inicial es 100 m del origen
//la aceleración es 2 m/s2

$x_inicial=100;
$v_inicial=50;
$aceleracion=2;

echo          "<table          border          =2          align          =
center><tr><B><td><B>Tiempo</B></td><td><B>Posición</B></td><td><B> Velocidad</B></td></tr>";

//Calcula los valores de la velocidad y la posición para cada instante de tiempo t desde
// t = 0 hasta t = 10, variando el tiempo de segundo en segundo, y luego los recoge en
// en una tabla.

    for($t=0; $t<=10; $t++){

        $v[$t]=$v_inicial + ($aceleracion*$t);
        $x[$t]=$x_inicial + ($v_inicial*$t) + (1/2*($aceleracion*(pow($t,2))));

        echo "<tr><td align = center>".$t."</td><td align =
center>".$x[$t]."</td><td align = center>".$v[$t]."</td></tr>";
    }

echo "</table>";

?>
```

En este caso, sería interesante, como ejercicio, representar los valores de la velocidad frente al tiempo en una gráfica y los valores de la posición frente al tiempo en otra gráfica. ¿Cómo son esas gráficas?...

Un caso particular de MRUV es la *caída libre*, en este caso, el algoritmo es el mismo salvo que el valor de la aceleración habría que sustituirlo por 9.8 m/s^2 que, como ya sabéis, es la aceleración con la que la Tierra atrae a los cuerpos cerca de su superficie, y adaptar las condiciones de posición y velocidad iniciales a nuestro problema.

Hasta ahora sólo hemos hablado de movimientos en un eje, es decir, en una dimensión, pero, ¿qué pasa con los movimientos en dos dimensiones?. En este caso, hay que recordar “el principio de independencia de los movimientos” de Galileo que dice lo siguiente:

“Si sobre un cuerpo actúan varias causas de movimiento, dichas causas actúan de forma independiente”.

Veamos un ejemplo para clarificar esto:

Imaginemos un avión que sobrevuela en línea recta una región y luego suelta un proyectil, en este caso, un observador desde el suelo vería que este proyectil realiza una trayectoria parabólica de la siguiente forma:

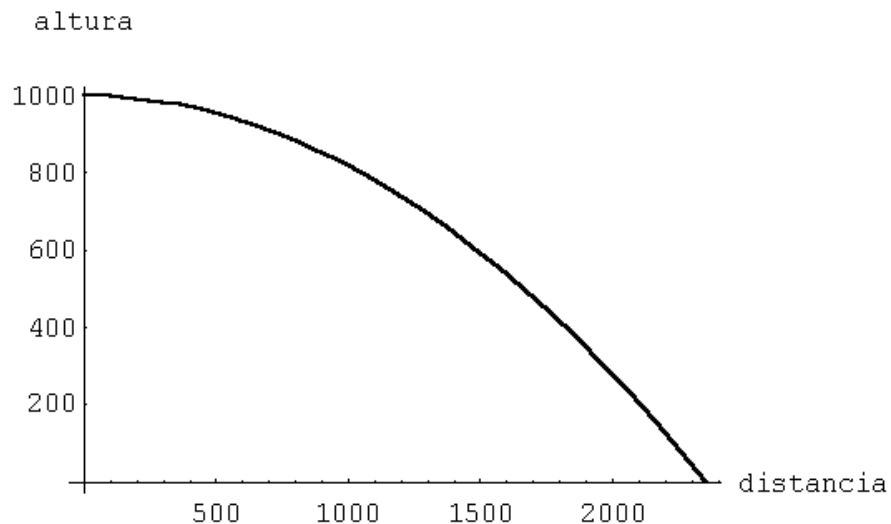


Figura 1.-Caída de un proyectil desde un avión que vuela a 1000m de altura con una velocidad de 600 Km/h.

Fuente: <http://www.astronomia.net/cosmologia/lec106.htm>

El movimiento en este caso es en dos ejes, llamemos al eje de la altura, eje Y, y al eje de la distancia, eje X; pues bien, el movimiento en el eje X sería un MRU con velocidad constante igual a 600 Km/h, y el movimiento en el eje Y es un movimiento de caída libre, es decir, un caso particular de MRUV, donde la aceleración vale 9.8 m/s^2 .

Con lo cual el algoritmo que representa este movimiento en dos dimensiones es el siguiente:

```

Algoritmo: Descripción de un movimiento en 2D: lanzamiento de un proyectil desde una
                altura h

función MOV_2D()

    para (t = 0 hasta t =  $\sqrt{\frac{2 \cdot h}{g}}$ ) hacer
        /*g es la aceleración de la gravedad que es constante, voy es la
        velocidad inicial en el eje Y, h es la altura desde la que se deja caer el
        proyectil, vox es la velocidad inicial en el eje X.*/

        y(t) = h - voy·t - ½·g·t2
        x(t) = x_inicial + vox·t

        // Δ t son incrementos de tiempo

        t = t + Δ t

    fin de hacer

devolver x(t) e y(t)
    
```

Veamos, ahora, su **implementación** en PHP:

```

<?
//Siguiendo el ejemplo de la figura 1.

$gravedad = 9.8;

//Ponemos la velocidad en unidades del S.I, es decir, en m/s
//Para una velocidad inicial de 600 Km/h, tenemos que son 166.67 m/s

$v_ejex=166.67;

//La altura es de 1000 m en el momento que se suelta el proyectil.

$h=1000;
    
```



```

//Recogemos estos datos en una tabla.

echo "<table border =2 align=center><tr><B><td><B>Tiempo</B></td><td><B>X(m)</B></td><td><B>Y(m)
</B></td></tr>";

$t=0;
while($t<=pow((2*$h/$gravedad),0.5)){

    $y[$t]=$h-(1/2*($gravedad*(pow($t,2))));
    $x[$t]= $v_ejex*$t;

    echo "<tr><td align = center>".$t."</td><td align=center>".$x[$t]."</td><td align =
center>".$y[$t]."</td></tr>";

//En cada paso del bucle sumamos 0.05 s al tiempo anterior

    $t=$t+0.05;
}

echo "</table>";

?>

```

Si representamos los datos obtenidos para x e y en una gráfica utilizando una hoja de cálculo, en este caso, Excel, obtenemos la siguiente gráfica:

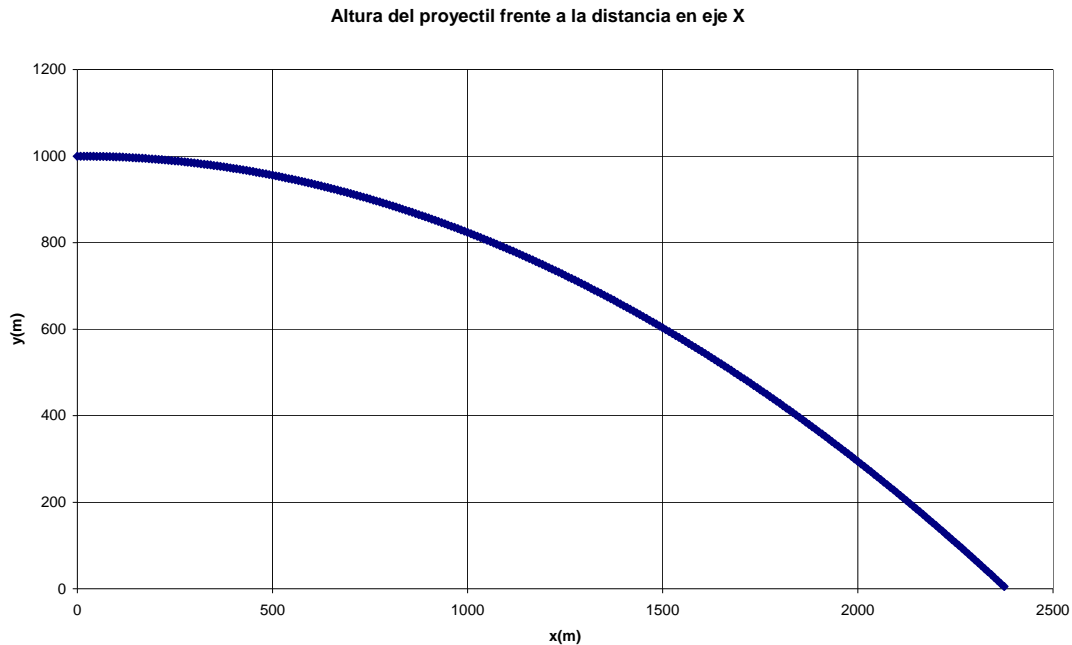


Figura 2.-Representación gráfica de los puntos obtenidos por el algoritmo implementado en PHP.

Si observamos esta gráfica, vemos que es idéntica a la de la figura 1, sólo que esta vez la hemos obtenido por méritos propios.

4. CONCLUYENDO...

Si os habéis dado cuenta, los algoritmos, ya sea en el campo de la física, o en cualquier otro campo, constituyen una herramienta muy útil a la hora de resolver problemas, haciendo uso, claro está, de los cada vez más potentes ordenadores.

A veces no podemos obtener una solución analítica para la ecuaciones físicas, por lo que, en este caso, se recurre a la resolución mediante métodos numéricos.

Aunque no lo haya mencionado antes, un buen algoritmo ha de hacer un buen uso del tiempo empleado y de los recursos que se consuman, sobre todo de memoria, por lo que, con unos conocimientos informáticos básicos y un buen dominio de algún lenguaje de programación, podemos traducir cualquier algoritmo y resolver nuestro problema de la manera más eficaz.



ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 29 – ABRIL DE 2010

5. BIBLIOGRAFÍA

- Ballester, M. y Barrio, J.. (2008). Física y Química. Editorial Oxford University Press,
- López Quijado, J. (2008). Domine PHP 5. Editorial Ra-ma,

Autoría

- Nombre y Apellidos: Yamil Bumedién Faris
- Centro, localidad, provincia: IES Enrique Nieto, Melilla, Melilla.
- E-mail: ybumedienfa@cofis.es