

2.2 Movimiento parabólico: expresiones

Suponemos un lanzamiento de un proyectil determinado por dos parámetros:

1. **Velocidad inicial** de lanzamiento. Un número v_0 expresado en metros por segundo.
2. **Ángulo** de lanzamiento. Un número θ expresado en grados.

En la figura 2.1 puede ver un esquema del lanzamiento de un proyectil.

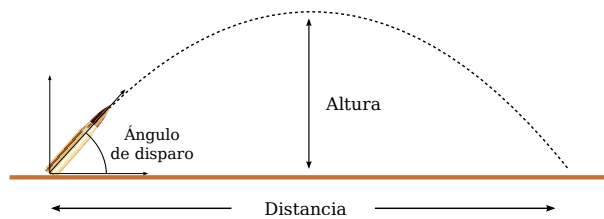


Figura 2.1
Trayectoria de un proyectil.

El primer problema que vamos a resolver es determinar el tiempo de vuelo, la altura máxima y la distancia recorrida por el proyectil, teniendo en cuenta que la única fuerza que consideramos es la de la gravedad (g). Para resolver el problema no hay más que proyectar el vector de la velocidad en la horizontal —con el coseno del ángulo— y en la vertical —con el seno del ángulo—. Si aplicamos las ecuaciones básicas de física y operamos podemos obtener las siguientes expresiones:

- Tiempo de vuelo:

$$t_v = \frac{2v_y}{g} = \frac{2v_0 \sin(\theta)}{g}$$

- Altura máxima:

$$y_{max} = \frac{v_y^2}{2g} = \frac{(v_0 \sin(\theta))^2}{2g}$$

- Alcance del proyectil:

$$x_{max} = v_x t_v = v_0 \cos(\theta) \frac{2v_0 \sin(\theta)}{g} = \frac{v_0^2 \sin(2\theta)}{g}$$

Ejercicio 2.1 Escriba un programa `parabola.cpp` que incluirá la función `main` con las líneas de código necesarias para resolver el problema. *Nota: La solución se limitará a la lectura, cálculo y escritura sin incluir ninguna sentencia condicional `if/else`.* Un ejemplo de su ejecución es:

```

Consola
Introduzca velocidad de lanzamiento(m/s): 100
Introduzca ángulo de elevación en grados (0-90): 45
Tiempo de vuelo: 14.421 segundos
Altura máxima: 254.929 metros
Alcance: 1019.72 metros
  
```

Para comprobar los resultados, puede usar las siguientes constantes:

$$G_{terrestre} = 9.80665 \quad \pi = 3.14159265358979323846$$

2.3 Movimiento parabólico en 3D: condicional

El ejercicio de la sección anterior incluía el cálculo de la distancia alcanzada por un proyectil que se disparaba con una velocidad inicial y un ángulo de disparo. En esta sección, vamos a plantear una solución más general en la que el proyectil se va a lanzar desde un punto del plano en un espacio 3D.

Puede considerar que vamos a simular el lanzamiento de un proyectil desde un punto terrestre (un punto en el plano), y deseamos saber en qué otro punto va a impactar. Para determinar el lanzamiento es necesario:

1. **Velocidad** de lanzamiento. Es un número v_0 expresado en metros por segundo.
2. **Elevación** de disparo. En la sección anterior lo hemos denominado “Ángulo de disparo” y se refiere a la elevación del cañón. Es un número θ expresado en grados.

3. **Rotación del cañón.** Se refiere a la rotación que aplicamos al cañón para determinar la dirección hacia la que se va a realizar el disparo. Podemos suponer que el ángulo cero se refiere al norte, y que éste está alineado con el eje Y del plano. Es un número ρ expresado en grados.

En la figura 2.2 se puede observar un esquema que representa el lanzamiento del proyectil.

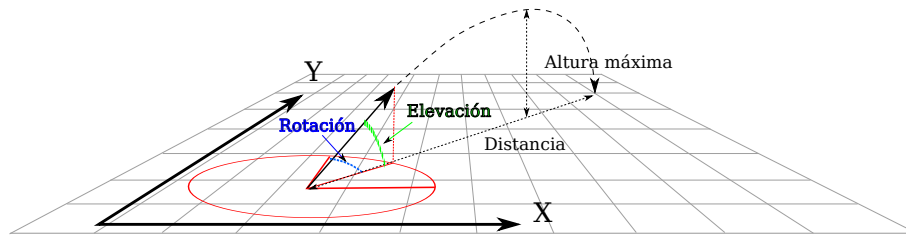


Figura 2.2
Trayectoria de un proyectil.

El conjunto de ecuaciones que se presentaron para el estudio de un movimiento parabólico en 2D siguen siendo válidas, es decir, podemos determinar a partir del ángulo de elevación y de la velocidad de lanzamiento los mismos valores. En particular, se puede determinar la distancia recorrida por el proyectil:

- Alcance del proyectil:

$$d_{\max} = v_{xy} t_v = v_0 \cos(\theta) \frac{2v_0 \sin(\theta)}{g} = \frac{v_0^2 \sin(2\theta)}{g}$$

En esta sección se propone resolver el problema de localizar el punto de impacto (x_f, y_f) a partir de:

1. **Posición del cañón:** dos parámetros (x_0, y_0) que determinan la localización del cañón en el plano XY.
2. **Parámetros de lanzamiento:** compuestos por los valores v_0, θ, ρ descritos anteriormente.

Es muy sencillo deducir que el punto de impacto vendrá dado por la siguiente ecuación:

$$(x_f, y_f) = (x_0 + d_{\max} \sin(\rho), y_0 + d_{\max} \cos(\rho))$$

Ejercicio 2.2 Escriba un programa `lanzamiento.cpp` que resuelva el problema de calcular el punto de impacto de un lanzamiento. El programa debe leer la posición del cañón, los ángulos de elevación y rotación y la velocidad inicial. La salida debe incluir el tiempo de vuelo, la altura máxima, el alcance y el punto de impacto. Un ejemplo de ejecución puede ser:

```

Consola
Introduzca dos valores con la posición del cañón: 0 0
Introduzca velocidad de lanzamiento(m/s): 100
Introduzca ángulo de elevación en grados (0-90): 45
Introduzca rotación: 60
----- Resultado del disparo -----
Tiempo de vuelo: 14.421 segundos
Altura máxima: 254.929 metros
Alcance: 1019.72 metros
Impacto: 883.1,509.858
  
```

Para resolver el problema, el programa debe incluir las siguientes restricciones:

- El parámetro de elevación tiene que estar en el rango $(0^\circ - 90^\circ)$ (sin incluir los extremos). En caso de no ser así, el programa debe terminar indicando que no es posible realizar el lanzamiento.
- La velocidad de lanzamiento debe ser positiva. En caso de no ser así, el programa debe terminar indicando que la velocidad no es correcta.
- Los parámetros de rotación deben estar normalizados en el rango $[-180^\circ, 180^\circ]$. En caso de no ser así, el programa debe recalcularlos para convertirlos en su valor equivalente dentro de este rango. Tenga en cuenta que podemos sumar o restar 360° tantas veces como sea necesario obteniendo ángulos equivalentes.

Por ejemplo, si introducimos una velocidad incorrecta:

```

Consola
Introduzca dos valores con la posición del cañón: 0 0
Introduzca velocidad de lanzamiento(m/s): -5
Error: Vinicial debe ser positiva
  
```

o si introducimos un ángulo erróneo:

```

Consola
Introduzca dos valores con la posición del cañón: 0 0
Introduzca velocidad de lanzamiento(m/s): 100
Introduzca ángulo de elevación en grados (0-90): 90
Error: elevación debe estar entre 0 y 90

```

Por otro lado, si damos un ángulo que se debe rectificar podemos incluir un mensaje informativo de la rectificación:

```

Consola
Introduzca dos valores con la posición del cañón: 100 100
Introduzca velocidad de lanzamiento(m/s): 500
Introduzca ángulo de elevación en grados (0-90): 30
Introduzca rotación: -270
Ángulo de rotación rectificado: 90
...

```

antes de presentar el resultado del lanzamiento.

2.3.1 Éxito del disparo

Se puede implementar un sistema para medir si el proyectil ha impactado en un objetivo o no. Si fijamos un objetivo en el punto (x_t, y_t) podemos calcular la distancia del punto de impacto como:

$$d = \sqrt{(x_f - x_t)^2 + (y_f - y_t)^2}$$

donde, recuerde, (x_f, y_f) es el punto de impacto que hemos calculado en la sección anterior.

Lógicamente, si la distancia es cero es que hemos alcanzado el objetivo de forma precisa. En la práctica, será muy difícil conseguirlo, ya que incluso por los errores de aproximación es muy complicado que se obtengan valores idénticos. Además, es más realista suponer que se ha conseguido un impacto aun siendo esa distancia algo mayor que cero. Así, catalogaremos el nivel de éxito de los disparos en una de estas categorías:

1. *Impacto directo.* Si el impacto se ha producido a menos de d_0 metros del objetivo, es decir, si la distancia d que hemos calculado es mejor que este valor.
2. *Impacto parcial.* Si el impacto se ha producido entre d_0 y d_1 metros del objetivo.
3. *No hay impacto.* Si el impacto ha sido a más de d_1 metros del objetivo.

Ejercicio 2.3 Amplíe el programa del ejercicio 2.2 para que, tras presentar el resultado del lanzamiento, compruebe si se ha impactado en un objetivo. En concreto, debe preguntar por un punto objetivo (x_t, y_t) , así como dos valores d_0 y d_1 , para informar sobre el impacto. Un ejemplo de ejecución es:

```

Consola
Introduzca dos valores con la posición del cañón: 100 100
Introduzca velocidad de lanzamiento(m/s): 500
Introduzca ángulo de elevación en grados (0-90): 30
Introduzca rotación: 90
----- Resultado del disparo -----
Tiempo de vuelo: 50.9858 segundos
Altura máxima: 3186.61 metros
Alcance: 22077.5 metros
Impacto: 22177.5,100
----- Comprobación de objetivo -----
Introduzca localización del objetivo: 22000 100
Introduzca distancias de impacto: 200 10
Distancia al objetivo: 177.504 -> impacto parcial.

```

donde puede observar que las distancias de impacto se han introducido incorrectamente (primero la mayor). Su programa deberá funcionar también en este caso. Además, si las distancias son negativas, deberá responder con un error y finalizar. Por ejemplo:

```
Consola
Introduzca dos valores con la posición del cañón: 100 100
Introduzca velocidad de lanzamiento(m/s): 300
Introduzca ángulo de elevación en grados (0-90): 40
Introduzca rotación: 85
----- Resultado del disparo -----
Tiempo de vuelo: 39.3277 segundos
Altura máxima: 1895.95 metros
Alcance: 9038.02 metros
Impacto: 9103.63,887.715
----- Comprobación de objetivo -----
Introduzca localización del objetivo: 9000 900
Introduzca distancias de impacto: -10 100
Error: distancias de impacto deben ser no negativas
```

Si volver a introducir los mismos valores pero con distancias correctas, el programa podría haber terminado como sigue:

```
Consola
...
...
Impacto: 9103.63,887.715
----- Comprobación de objetivo -----
Introduzca localización del objetivo: 9100 900
Introduzca distancias de impacto: 20 100
Distancia al objetivo: 12.809 -> impacto directo.
```

En donde se ha introducido los datos en el orden correcto (la menor primero). En este caso, el programa ha determinado un impacto directo pues la distancia es menor que el primero.