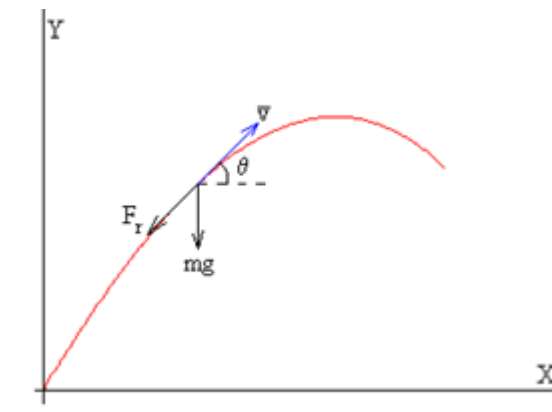


Se desea realizar un programa MATLAB que calcule el alcance máximo de un tiro parabólico, considerando el rozamiento del proyectil con el aire. Aunque normalmente el rozamiento es función del cuadrado de la velocidad, para valores pequeños se puede considerar proporcional a la velocidad.

En esas condiciones, en un instante dado, el proyectil sufre una aceleración vertical debida a la gravedad y una deceleración en la dirección del movimiento, proporcional a la velocidad que lleve el proyectil en ese instante. Si b es la constante de proporcionalidad del rozamiento con la velocidad, las ecuaciones del movimiento del proyectil son:



$$m \frac{dv_x}{dt} = -mbv_x$$

$$m \frac{dv_y}{dt} = -mg - mbv_y$$

Si integramos dos veces y tenemos en cuenta las condiciones iniciales ($t=0$, $x=0$, $y=0$, $v_x=v_{0x}$, $v_y=v_{0y}$) podemos obtener las expresiones de x e y en función del tiempo (ver anexo), que nos permitirían calcular el alcance máximo del proyectil, simplemente calculando el instante en el que el proyectil alcanza la cota $y=0$ y sustituyendo ese valor en la expresión de x . El problema que se nos presenta es que calcular ese instante no tiene solución analítica, por lo que debemos recurrir a métodos numéricos (ocurre lo mismo en el caso del rozamiento proporcional al cuadrado de la velocidad, aunque con expresiones un poco más complicadas)

El método numérico que vamos a utilizar consiste en considerar intervalos muy pequeños de tiempo, de tal manera que el movimiento real se pueda aproximar por un movimiento con aceleración prácticamente constante en cada intervalo. De esa forma, podemos calcular las posiciones y velocidades en un intervalo (k) a partir de los valores calculados previamente en el intervalo anterior ($k-1$) según las expresiones:

$$v_x^{(k)} = v_x^{(k-1)} - b \cdot v_x^{(k-1)} \cdot \Delta t$$

$$v_y^{(k)} = v_y^{(k-1)} - (g + b \cdot v_y^{(k-1)}) \cdot \Delta t$$

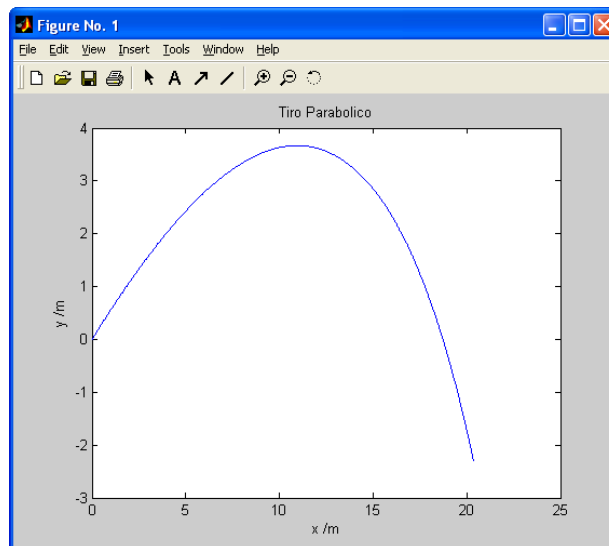
$$x^{(k)} = x^{(k-1)} + v_x^{(k-1)} \cdot \Delta t$$

$$y^{(k)} = y^{(k-1)} + v_y^{(k-1)} \cdot \Delta t$$

Donde Δt es la anchura fija de cada intervalo y $v_x^{(k)}$ es el valor de la velocidad v_x en el intervalo (k), calculado a partir de la velocidad $v_x^{(k-1)}$ en el intervalo anterior (lo mismo para las otras tres magnitudes). Los valores iniciales serían $x^{(1)}=0$, $y^{(1)}=0$, $v_x^{(1)}=v_{0x}$, $v_y^{(1)}=v_{0y}$.

a) Se pide hacer un programa en Matlab que haga lo siguiente:

- Pedir por teclado los valores de la velocidad inicial del proyectil (V_0), el ángulo de lanzamiento (α) y el coeficiente de proporcionalidad del rozamiento (b).
- Crear un vector t que contenga 1000 valores distribuidos uniformemente entre 0 y $2 \cdot V_0 \cdot \sin(\alpha) / g$, que corresponde al tiempo de vuelo en caso de rozamiento nulo.
- Crear cuatro vectores x , y , v_x y v_y con 1000 elementos de valor 0.
- Establecer los valores de posiciones y velocidades del primer intervalo (con índice 1) según las condiciones iniciales de posición y velocidad. Es decir, $x(1)=0$; $y(1)=0$; $v_x(1)=V_0 \cdot \cos(\alpha)$ y $v_y(1)=V_0 \cdot \sin(\alpha)$
- Hacer un bucle con índice k , desde 2 hasta el final del vector, calculando los valores de $x(k)$, $y(k)$, $v_x(k)$ y $v_y(k)$ a partir de los valores anteriores $x(k-1)$, $y(k-1)$, $v_x(k-1)$ y $v_y(k-1)$, según la ecuaciones descritas al final de la página anterior.
- Después de ese bucle ya tenemos calculados los valores numéricos aproximados de las posiciones y velocidades en cada intervalo. Dibujar la grafica x - y (La siguiente gráfica es un ejemplo con valores $V_0=20$ m/s, $\alpha=30^\circ$ y $b=0.6$ s $^{-1}$)



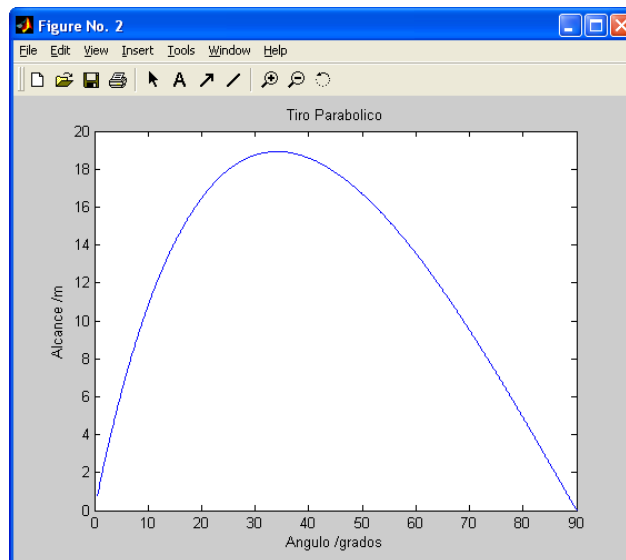
- A continuación, calcular el alcance máximo y mostrarlo en pantalla. Para ello, se debe encontrar el índice P de la primera componente del vector y en que el valor pasa a ser negativo. Una vez localizado ese índice P , el alcance máximo será precisamente la correspondiente componente en el vector x , es decir $x(P)$. (En el ejemplo, 18.7286m)

Se sugiere que, en este punto, se guarde el programa en un fichero que se entregará como parte A, y se copie en otro fichero que se modificará para hacer el apartado B.

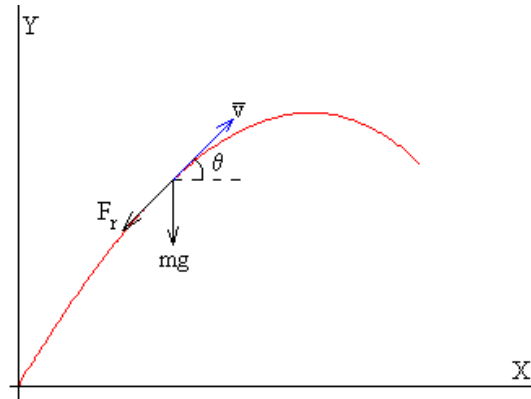
b) Modificar el programa del apartado A para que calcule el ángulo que consiga mayor alcance. En el caso de tiro parabólico sin rozamiento, ese ángulo corresponde a 45° , pero si hay rozamiento, ese ángulo es menor. El programa tiene que hacer lo siguiente:

- Pedir por teclado los valores de velocidad inicial del proyectil (V_0) y coeficiente de proporcionalidad del rozamiento (b)
- Crear un vector Ang, con 500 valores distribuidos entre 0.01° y 90°
- Para cada uno de los ángulos del vector Ang, calcular el alcance máximo correspondiente (como se calculó en la parte A del examen) y almacenarlo en la componente correspondiente del vector Alcance.
- Pintar en pantalla la grafica de Ang-Alcance
- Utilizando el comando “max” de Matlab, encontrar el mayor valor del vector Alcance y el índice de ese valor máximo. En la misma posición en el vector Ang tendremos el ángulo de mayor alcance. Escribir ese valor en pantalla.

(La siguiente gráfica muestra el resultado para el ejemplo con $V_0=20$ m/s y $b=0.6$ s $^{-1}$. El ángulo de alcance máximo es 34.0857°)



ANEXO:



Las ecuaciones del movimiento del proyectil son:

$$m \frac{dv_x}{dt} = -mbv_x$$

$$m \frac{dv_y}{dt} = -mg - mbv_y$$

donde b es la constante de proporcionalidad del rozamiento con la velocidad. Si integramos respecto a t , con las condiciones iniciales $t=0$, $v_x=v_{0x}$, $v_y=v_{0y}$, la solución es:

$$\frac{dx}{dt} = v_{0x} \exp(-bt)$$

$$\frac{dy}{dt} = \left(\frac{g}{b} + v_{0y} \right) \exp(-bt) - \frac{g}{b}$$

que, integrando de nuevo, con las condiciones iniciales $t=0$, $x=0$, $y=0$, nos lleva a:

$$x = \frac{v_{0x}}{b} (1 - \exp(-bt))$$

$$y = \frac{1}{b} \left(\frac{g}{b} + v_{0y} \right) (1 - \exp(-bt)) - \frac{g}{b} t$$

Si queremos calcular el alcance máximo, sólo tenemos que igualar $y=0$ y obtener el tiempo de vuelo, que sustituyendo en la expresión de x , nos permite calcular la distancia máxima a la que puede llegar el proyectil. El problema es que esa ecuación no tiene solución analítica, por lo que debemos recurrir a métodos numéricos.