

Métodos Numéricos para la Ciencia e Ingeniería: Informe 2

Álvaro Céspedes

September 2015

1 Problema de choque inelástico

1.1 Introducción

Para una partícula de masa m que choca inelásticamente contra una superficie regida por un movimiento sinusoidal de amplitud máxima A y frecuencia angular ω se busca encontrar un algoritmo que logre determinar la posición y la velocidad de la partícula en el instante posterior a un rebote con el suelo, en función de la posición y velocidad de la partícula en el momento justo antes del bote.

1.2 Procedimiento

La idea detrás del algoritmo es crear una función que, a partir de un vector de condiciones iniciales (V_0, H_0) , vaya creando las nuevas condiciones iniciales para cada bote posterior. En detalle, para esto hay que considerar que, primero, el suelo sigue un movimiento del tipo

$$Y_s = A \sin(\omega t)$$

y con velocidad

$$V_s = A \omega \cos(\omega t)$$

Mientras que la partícula se rige por las ecuaciones

$$Y_p = H_0 + V_0 t - \frac{1}{2} g t^2$$

y con velocidad

$$V_p = V_0 - g t$$

Donde g es la aceleración de gravedad.

Esto conduce a que, como queremos la posición y velocidad en un instante t^* correspondiente a un choque, primero debemos calcular dicho tiempo, y para ello buscamos las raíces de la función

$$Y_p - Y_s$$

Esto se hace usando el método "Brentq" de la librería *optimize* de Python (usada adecuadamente, es decir, encontrando intervalos donde la función cambie de signo).

Finalmente, con t^* calculado se obtienen la posición y velocidad del instante del golpe, y con la siguiente ecuación se calcula la velocidad en el instante posterior al golpe:

$$V_{pp}(t^*) = (1 + \eta)V_s(t^*) - \eta V_p(t^*)$$

Donde η corresponde al coeficiente de restitución (η entre 0, y 1; $\eta = 1$ corresponde al caso elástico).

Para este algoritmo en particular se eligen $A = 1$, $g = 1$ y $m = 1$, con el fin de simplificar los cálculos y los procedimientos.

Con el procedimiento ya instaurado, lo restante es iterar n veces (Número de botes) y el programa se encargará de ir agregando al vector de condiciones iniciales las posiciones, velocidades y tiempos de choque para cada bote, o almenos esa es la idea...

1.3 Resultados

El algoritmo no funcionó al momento de agregar la iteración (Cuando se probó el programa sin modificar el código para agregar iteraciones arrojaba valores coherentes) y se desconoce el problema detalladamente, pero se cree que se están reescribiendo los valores iniciales de los vectores, por lo que los vectores de tiempo y posición de choque se llenan con ceros. (Condiciones iniciales dadas, el primer 'choque' se considera en el tiempo 0 producto que la altura inicial es cuando la distancia entre la partícula y la superficie es nula).

2 Conclusiones

A pesar de que se tenía una idea concreta de como abordar el problema, el algoritmo falló por no tener conocimiento suficiente de trabajar iteraciones sobre funciones (Problemas al nombrar las variables de cada función), sin embargo, se espera que para próximas experiencias problemas como este no vuelvan a suceder y el problema de la programación pase a segundo plano.