A picture containing text, font, black, graphics

Description automatically generated

Manual de uso de funciones Hipersónico

**Definición y uso de objetos de la clase ShapeHipersonic - Matlab**

**Escola de Enxeñaría Aeronáutica e do Espazo**

A green lines on a black background

Description automatically generated with low confidence

Autor/a: Christian Nicolás La Banca Lotter

# Definición de objetos de la clase

El constructor de la clase es un constructor genérico sin argumentos, por lo que al llamar a ***shapeHipersonic()*** y asignarlo a una variable del Workspace, se crea un objeto vacío:

A screen shot of a computer

Description automatically generated

Ilustración .: Uso del constructor genérico

Si queremos conocer qué funciones (métodos) tiene asociada la clase de nuestro nuevo cuerpo, podemos usar el comando methods():

A blue screen with white text

Description automatically generated

Ilustración .: Funciones disponibles para el usuario

# Descripción del flujo de trabajo

El workflow se basa en definir la geometría y posteriormente obtener la distribución de CP y las fuerzas resultantes en base a esa geometría.

La geometría estará definida por dos superficies, una superficie inferior y otra superior, y está basada en la definición del plano XY como el plano base, teniendo la posibilidad de definir puntos para Z\_upper y Z\_lower asociados a cada uno de los puntos definidos en el plano XY.

Ambas superficies deben tener un formato meshgrid (ver documentación de Matlab para [meshgrid](https://es.mathworks.com/help/matlab/ref/meshgrid.html)). Por lo que para su generación se pueden utilizar la función meshgrid de Matlab y las funciones customizadas como triangle\_meshgrid o f\_meshgrid.

## Preset de geometría

La clase tiene un **preset** de geometría preconstruido, que es un con su eje de revolución situado a lo largo del eje X, con su punta en (0,0,0). Para acceder a este preset, se utilizará la función ***setPremadeShape(****settings****)***.

Ilustración .: Definición del cono preestablecido

Se nos muestra el siguiente mensaje que indica que la geometría se ha definido correctamente:



## Visualización de geometría

Podemos utilizar la función ***drawGeometry()*** para visualizar la geometría que hemos definido, el código de colores se basa en le valor de la variable Z.

A graph of a rainbow colored triangle

Description automatically generated

Ilustración .: Visualización de la geometría

## Obtención de CP

Podemos utilizar la función ***solveCP(***alpha,beta***)*** para calcular la distribución de CP en la geometría para un ángulo de ataque **alpha** y un ángulo de guiñada **beta**, ambos en radianes.

Calculamos CP para el cono a un Alpha de 15º y se nos enseña en consola el siguiente mensaje que indica que se ha calculado CP correctamente:



## Obtención de la fuerzas y momentos

Tenemos a nuestra disposición dos funciones, distintas.

**[CF]=** ***solveForces(***alpha,beta***)***

calcula los coeficientes de fuerzas, **en ejes cuerpo,** para un ángulo de ataque alpha y un ángulo de guiñada beta, ambos en radianes, mientras que

**[CF, CM] =** ***solveForcesAndMoments(***alpha,beta,punto***)***

calcula las fuerzas y los momentos, **en ejes cuerpo,** para un ángulo de ataque alpha y un ángulo de guiñada beta, ambos en radianes.

# Definición de una geometría customizada

Existen 4 propiedades asociadas a la geometría: X\_data, Y\_data, Z\_data\_upper, Z\_data\_lower que se pueden visualizar pero no se pueden definir. Para definirlos, hay que utilizar la función ***setGeometry(***X,Y,Z\_up,Z\_low***)***

Esta función acepta 4 matrices en formato **meshgrid**, por lo que pueden definir cualquier geometría.

Por ejemplo:

## Definición de una cuña

Para definir una cuña definiremos un rectángulo en el plano XY que tendrá asociado una Z\_up y una Z\_down para cada punto x,y. Para definir estas matrices, utilizaremos la función meshgrid de Matlab:

A computer code with white text

Description automatically generated

Ilustración .: Definición del plano XY mallado en Matlab

Conocido el plano XY, definiremos una cuña con dos superficies, una superior y otra inferior



Se define la superior con una pendiente de 0.2 y la inferior de -1.

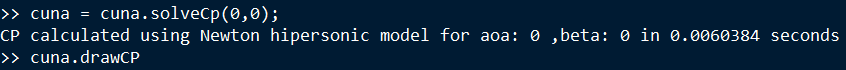
Ahora, debemos asociar estas matrices a nuestro nuevo objeto, para ello, generamos el objeto vacío y asociamos la geometría con ***setGeometry***.

A computer screen shot of a code

Description automatically generated

Ilustración .: Asociación de la geometría al objeto

Ahora, calcular y dibujar CP alrededor de la geometría es ahora simplemente utilizar las funciones de la clase:



Resultado:

A graph of a graph

Description automatically generated with medium confidence

Ilustración .: Distribución de CP para alpha = beta = 0 en la cuña que hemos definido

Para obtener las fuerzas utilizaremos la función ***solveForces***.

A blue screen with white text

Description automatically generated