Entrega 2: Implementación de elementos finitos

Walter Andrés Castañeda Díaz

Universidad Nacional de Colombia

RESUMEN

Todos los gases son compresibles en cierta medida, el helio, debido a su baja masa molar, tiene una alta compresibilidad en comparación con gases más pesados. Esto significa que su densidad puede cambiar significativamente con variaciones en la presión. En condiciones normales de temperatura y presión, el helio puede aproximarse como un gas incompresible para muchos propósitos prácticos, especialmente cuando las variaciones de presión no son significativas. El método de elementos finitos es un método numérico para la aproximación de ecuaciones diferenciales que son complejas de solucionar de manera analítica. El propósito de este trabajo es de realizar una descripción de un problema en ecuaciones diferenciales parciales encontrado en el artículo [1]. Luego, mostrar la implementación del método de elementos finitos a uno de los problemas planteados en dicho artículo mediante el sofware Python.

DESCRIPCIÓN Y OBJETIVOS

El modelo se basa en la consideración de un tubo cilíndrico de 40 cm de longitud y 3 cm de radio, el cual no está expuesto a temperaturas externas, a excepción de la temperatura ambiente. Se introduce helio en el tubo a través de uno de los orificios y se permite que fluya hacia fuera por el otro extremo.

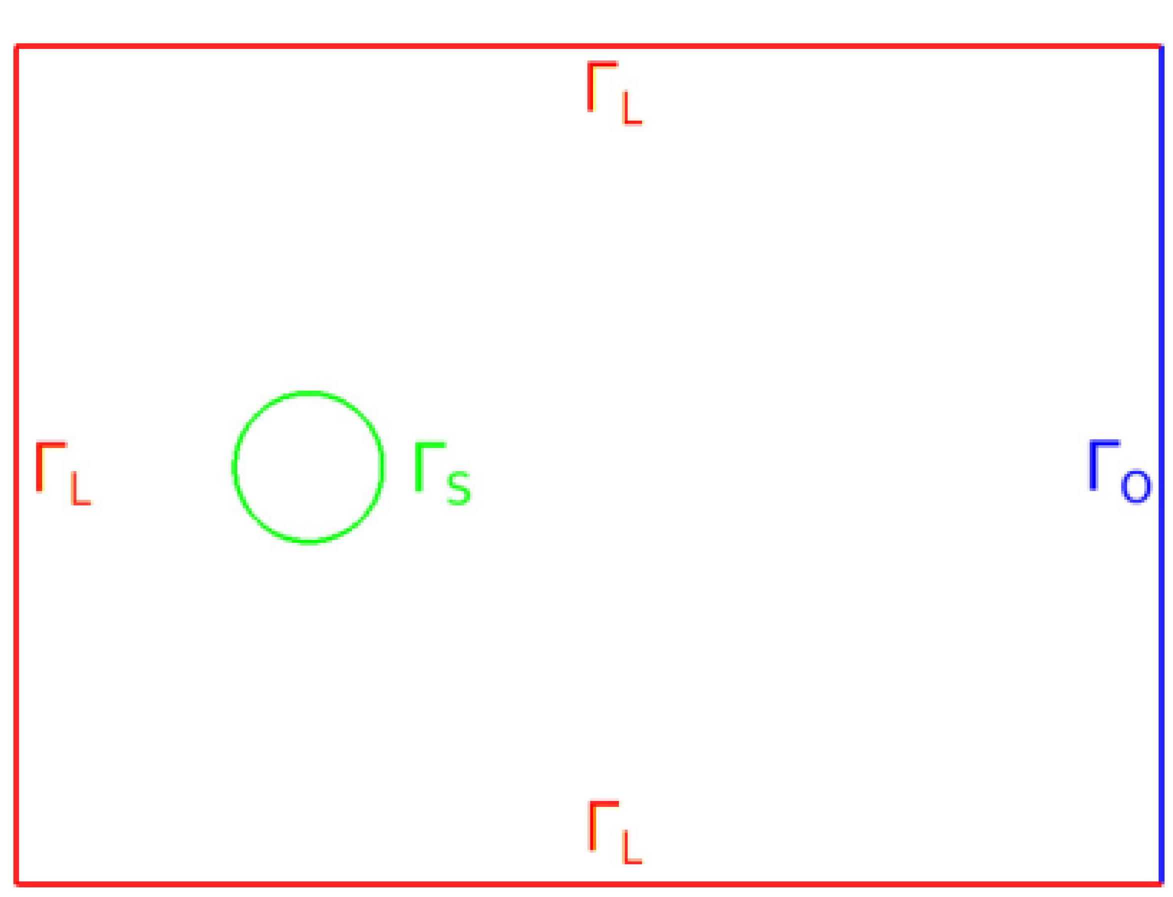
Se asume que el helio es incompresible, isentrópico y sigue las leyes de los gases ideales. Estas suposiciones son válidas bajo condiciones normales de temperatura y presión. Se realizan estas consideraciones con el propósito de modelar el comportamiento de la velocidad y la presión del gas en este entorno. Finalmente, también se supondrá que el helio sigue las leyes de los gases ideales para facilitar la implementación del modelo.

HIPÓTESIS Y NIVEL DE DETALLE

Según fenómenos conocidos, el gas se comportará de forma turbulenta, formando vórtices. Además las presiones y velocidades no serán altas debido a el medio en que se encuentra.

DEFINICIÓN DE SUBSISTEMAS Y PARTICIONES

Dado que el proceso se lleva a cabo exclusivamente a través de un tubo, existe un único sistema compuesto por las paredes de dicho tubo. Para simplificar, el modelo se realiza en dos dimensiones. Para ello, se utiliza una región rectangular con un hueco en ella.



Para ello se divide el diagrama en 4 regiones:

Γ_L : Representan las fronteras físicas dentro del cilindro.

Γ_S : Representa la frontera fisica por donde ingresa el fluido.

Γ_O : Representa una frontera ficticia, por donde sale el fluido.

Ω : Representa la parte interior comprendida por estas fronteras.

PRINCIPIOS, TEORIAS O LEYES

Sea un dominio $\Omega \subset \mathbb{R}^2$ con frontera acotada $\Gamma = \partial\Omega$, las ecuaciones compresibles isentrópicas están dadas por:

$$\rho\frac{\partial u}{\partial t} + \rho(u\cdot\nabla)u - \mu\nabla^2 u - \frac{1}{3}\mu\nabla(\nabla\cdot u) + \nabla p = 0; \text{ en } \Omega$$
$$\frac{\partial \rho}{\partial t} + u\cdot\nabla\rho + \rho\nabla\cdot u = 0; \text{ en } \Omega. \tag{1}$$

Mediante las siguientes igualdades:

$$\frac{p_0}{p} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{2}$$
$$\frac{\rho_0}{\rho} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{1}{\gamma-1}}, \tag{3}$$

despejando p de (2) y derivando respecto a t, la derivada temporal final de la ecuación para un flujo de gas de baja velocidad puede aproximarse como:

$$\frac{\partial p}{\partial t} \approx c_0^2\frac{\partial \rho}{\partial t}.$$

De otra forma la velocidad de sonido c, puede ser calculada como

$$c^2 = c_0^2\left(1 + \frac{\gamma-1}{2}M^2\right)^{-1},$$

Y así obtener la siguiente igualdad:

$$\frac{\partial p}{\partial t} = c^2\frac{\partial \rho}{\partial t}. \tag{4}$$

De forma análoga se puede deducir

$$\nabla p = c^2\nabla\rho. \tag{5}$$

Despejando $\frac{\partial\rho}{\partial t}$ de (4), $\nabla\rho$ de (5) y remplazando en (1) se obtiene

$$\rho\frac{\partial u}{\partial t}+\rho(u\cdot\nabla)u-\mu\nabla^2u-\frac{1}{3}\mu\nabla(\nabla\cdot u)+\nabla p=0;\text{ en }\Omega$$
$$\frac{1}{c^2}\frac{\partial p}{\partial t}+\frac{1}{c^2}u\cdot\nabla p+\rho\nabla\cdot u=0;\text{ en }\Omega.$$

Dado que el helio se puede comportar como una gas isentrópico debido a las condiciones normales de temperatura, entonces $\frac{\partial\rho}{\partial t}=0$. Así, $\frac{\partial p}{\partial t}=0$.Además, suponiendo que la densidad es constante

$$\rho\frac{\partial u}{\partial t}+\rho(u\cdot\nabla)u-\mu\nabla^2u-\frac{1}{3}\mu\nabla(\nabla\cdot u)+\nabla p=0;\text{ en }\Omega$$
$$\rho\nabla\cdot u=0;\text{ en }\Omega.$$

RECONOCIMIENTO DE LA ESTRUCTURA

Los términos asociados la estructura del modelo son:

- ρ: densidad del helio el que se supone constante.
- u: velocidad del fluido que se compone de 2 componentes, es decir $u=[u_1,u_2]$.
- p: presión
- μ: viscosidad dinámica del fluido.
- c: la velocidad del sonido.

Una breve descripción de los términos en el sistema:

- $(u\cdot\nabla)u$. Es el término convectivo del sistema.
- ∇^2u. Representa como se propaga la velocidad debido a la viscosidad.
- Comprende el termino no convectivo de el sistema.
- Los demás términos ya son conocidos.

ECUACIONES CONSTITUTIVAS Y PARÁMETROS

Las ecuaciones constitutivas para el modelo son:

- Relación de la densidad total con la densidad de estancamiento en función del número de Mach.
$$\frac{\rho_0}{\rho}=\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{1}{\gamma-1}}.$$
- Relación de la presión total con la presión de estancamiento en función del número de Mach.
$$\frac{p_0}{p}=\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}}.$$
- Relación de la velocidad del sonido con la velocidad del sonido en un gas ideal en función del número de Mach
$$c^2=c_0^2\left(1+\frac{\gamma-1}{2}M^2\right)^{-1}.$$
- Velocidad del sonido de un gas ideal
$$c_0=\sqrt{\frac{\gamma RT_0}{\mathcal{M}}}$$

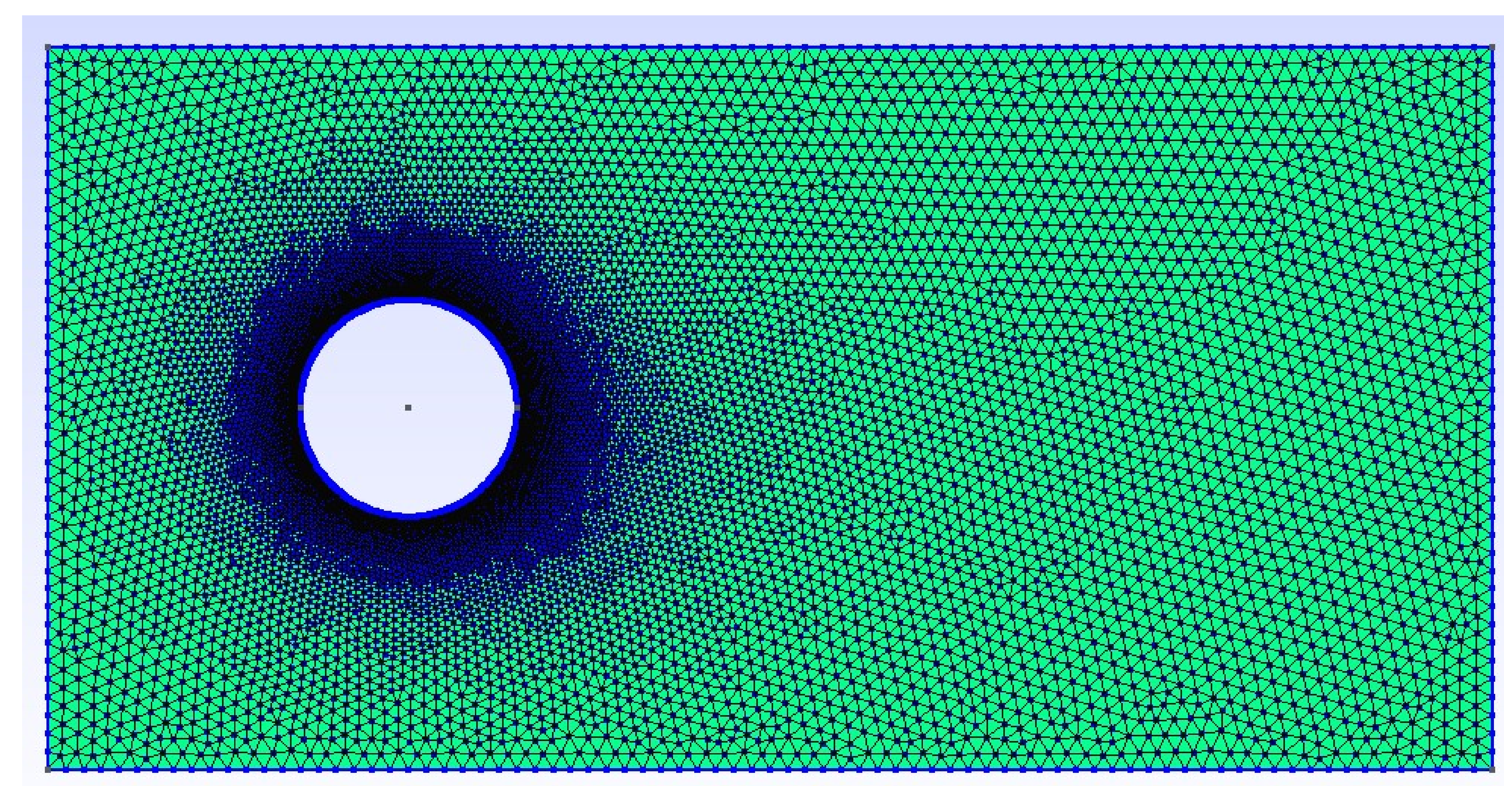
Las parámetros funcionales o hiper-parámetros esta dados por:

- p_0. Presión de estancamiento.
- ρ_0. Densidad de estancamiento.
- c_0. Velocidad del sonido en un gas ideal.
- R. La constante de gas universal.
- T_0. Temperatura de el campo en estancamiento.
- $\mathcal{M}$. Masa molar del gas.
- M. Número de Mach.

GRADOS DE LIBERTAD Y CONSTRUCCIÓN DEL MODELO COMPUTACIONAL

Dado que lo que se pretende hallar ... así el modelo es resoluble.

Para la construcción de el modelo, inicialmente se tuvo que crear una malla asociada al problema. Inicialmente se creo un archivo con extensión .geo de donde se especifica las dimensiones de la malla, y el tamaño de los elementos, cerca al hueco de la superficie y lejos de este. La representación gráfica de la malla llamada ”malla_entrega.geo”se puede observar mediante el software Gmsh.



Esta implementación se realizado con la ayuda de el software Python, mediante la librería sfepy.

Primero se extrae la malla a Python y almacenamos nuestra malla mediante el código

```
filename_mesh = data_dir + ’/meshes/2d/malla_entrega.mesh
mesh=Mesh.from_file(filename_mesh)
```

se importan las funciones Mesh, FEDomain y Field, que nos ayudará a crear la malla, los dominios y campos que queremos hallar, que en este caso, son la velocidad y la presión.

```
from sfepy.discrete.fem import Mesh, FEDomain, Field
```

A continuación se crea un dominio llamado ’domain’, para elementos finitos, mediante el código

```
domain=FEDomain("domain",mesh)
```

Se crean las regiones de el sistema a las cuales se les pondra condiciones

```
omega=domain.create_region("Omega","all")
bottom = domain.create_region(’Bottom’,’vertices␣in␣(y␣<␣0.001)’,
   ’facet’)
left=domain.create_region(’Left’, ’vertices␣in␣(x␣<␣0.001)’, ’
  facet’)
top=domain.create_region("Top",’vertices␣in␣(y␣>␣19.99)’,’facet’)
```

luego se crean los campos de velocidades y presión mediante la función Field

```
field=Field.from_args("fu",np.float64,"vector",omega,approx_order
 =2)
presion=Field.from_args("fu2",np.float64,"scalar",omega,
 approx_order=1)
```

Para solucionar un problema de elementos finitos es necesario definir variables del modelo y propiedades del fluido. Además, como la solución de elementos finitos involucra integrales, se deben determinar el orden de las integrales númericas a realizar y definir ecuaciones que se emplean en el modelo. Se importan esta funciones mediante el codigo

```
from sfepy.discrete import (FieldVariable, Material,Integral,
 Function,Equation,Equations,Problem)
```

Ahora se definen las variables tanto de entrenamiento como de prueba las cuales las llamaremos u, v, q, p

```
u=FieldVariable("u","unknown",field)
v=FieldVariable("v","test",field,primary_var_name="u")
p=FieldVariable('p','unknown',presion)
q=FieldVariable('q' ,'test',presion,primary_var_name="p")
```

Mediante la función Material, se definen la propiedades del fluido, en este caso del helio.Suponiendo que la densidad del helio es de $0.164\frac{kg}{m^3}$ y l viscosidas es de 0.0018. Además, se hallan otras constantes para el modelo

```
h=2
densidad= 0.164
velocidad_gas=343
viscosidad=0.001
c2=viscosidad/3
c3=h*densidad
c4=1/(velocidad_gas**2)
```

Ahora, se define el orden de la integral que se usará para desarrollar el problema(entre mayor sea el orden mayor será la aproximación

```
integral3=Integral("i3",order=3)
integral4=Integral("i4",order=4)
integral5=Integral("i5",order=5)
integral6=Integral("i6",order=6)
```

Para definir las ecuaciones, primero se debe definir cada uno de los términos de la ecuación de lo forma débil del sistema de ecuaciones. Dado que se supone, que no hay cambio de velocidades en el tiempo y la densidad es constante. Además suponiendo que no existen fuerzas internas que actúan entonces la forma débil del problema queda dado por

$$\rho h < v, u > + < \rho v, (u \cdot \nabla)u > + \mu < \nabla v, \nabla u > + \frac{1}{3}\mu < \nabla \cdot v, \nabla \cdot u >$$
$$- < \nabla \cdot v, p > + \frac{h}{c^2} < q, p > + \frac{1}{c^2} < q, u \cdot \nabla p > + < \rho q, \nabla \cdot u >= 0$$

A continuación se define cada uno de los teminos, que se puede realizar mediante la función Term y que se pueden encontrar en la documentación de sfepy .Documentación

```
from sfepy.terms import Term

t1=Term.new("dw_dot(v,u)",integral,omega,v=v, u=u)
t2=Term.new("dw_convect(v,u)",integral,omega,v=v, u=u)
t3=Term.new("dw_div_grad(v,u)",integral,omega,v=v, u=u)
t4=Term.new("dw_st_grad_div(v,u)",integral,omega,v=v, u=u)
t5=Term.new("de_stokes(v,p)",integral,omega,v=v, p=p)
t6=Term.new("dw_dot(q,p)",integral,omega, q=q,p=p)
t7=Term.new("dw_convect_v_grad_s(q,u,p)",integral,omega,q=q,u=u,
 p=p)
t8=Term.new("dw_stokes(u,q)",integral5,omega,q=q, u=u)
```

Ahora se definen la ecuación, que representa el sistema

```
eq1=Equation("balance",t1*c3+t2*densidad+t3*viscosidad-t5+t6+t7*
 c4)

eqs=Equations([eq1])
```

Para definir las condiciones de frontera se deben importar la función EsencialBC

```
from sfepy.discrete.conditions import Conditions, EssentialBC
```

suponiendo que las condiciones de frontera son algunos valores constantes

```
fix1_u=EssentialBC("fix1_u",bottom,{"u.all":0.7})
fix2_u=EssentialBC("fix2_u",left,{"u.all":6})
fix3_u=EssentialBC("fix3_u",top,{"u.0":3,"u.1":-3.4})
```

por último para solucionar las ecuaciones no lineales y lineales se usan las siguientes funciones

```
from sfepy.base.base import IndexedStruct
from sfepy.solvers.ls import ScipyDirect
from sfepy.solvers.nls import Newton
```

Para resolver este problema, se activan dichas funciones mediante el comonado

```
ls=ScipyDirect({})
nls_status=IndexedStruct()
nls=Newton({},lin_solver=ls,status=nls_status)
```

por ultimo se compacta el problema mediante los codigos

```
pb=Problem("entrega",equations=eqs)

pb.save_regions_as_groups("regions")

pb.set_bcs(ebcs=Conditions([fix1_u,fix2_u,fix3_u]))
```

mediante en siguiente codigo, se resuelve el problema

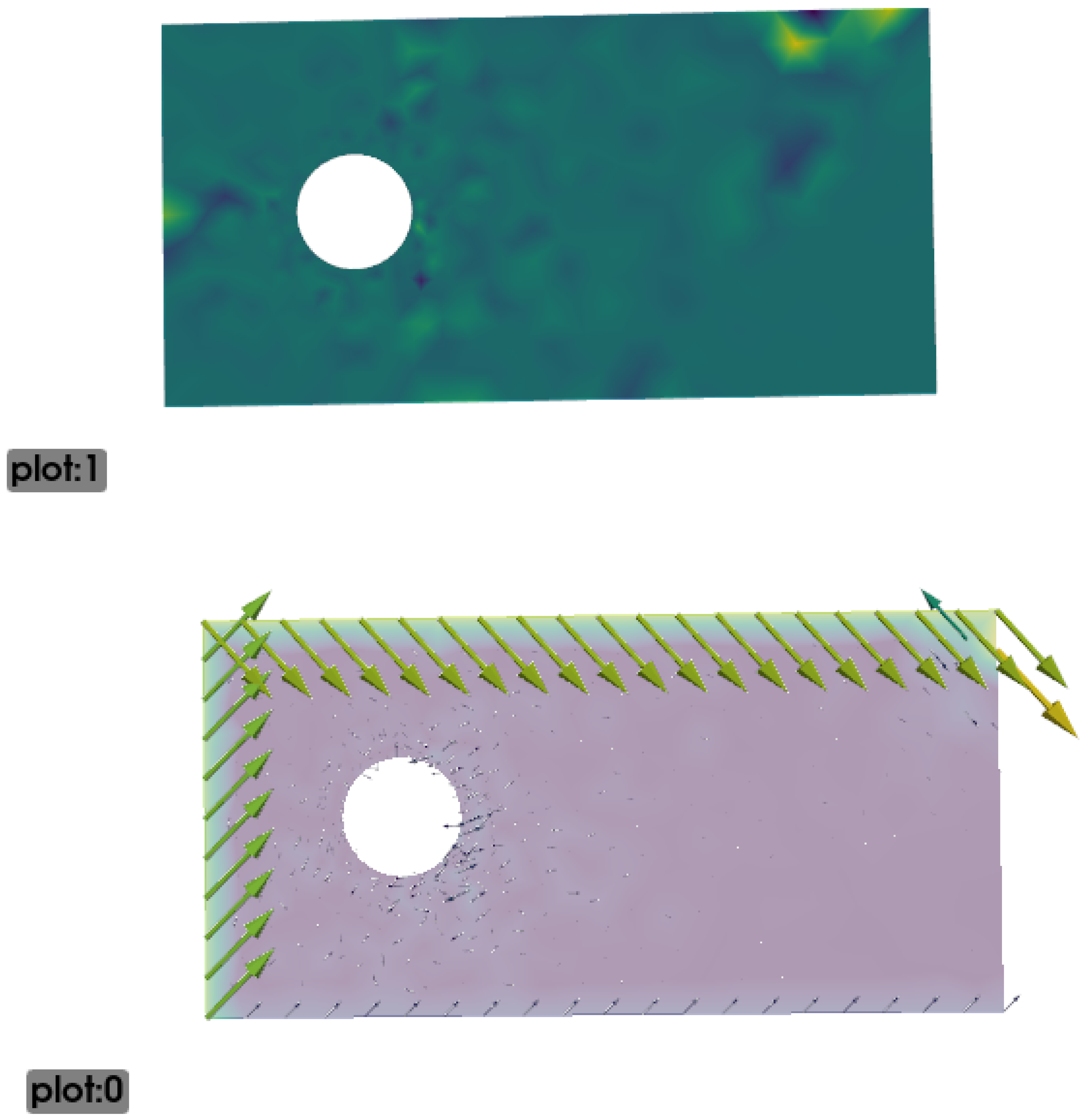
```
pb.set_solver(nls)
status=IndexedStruct()
vec=pb.solve(status=status)
```

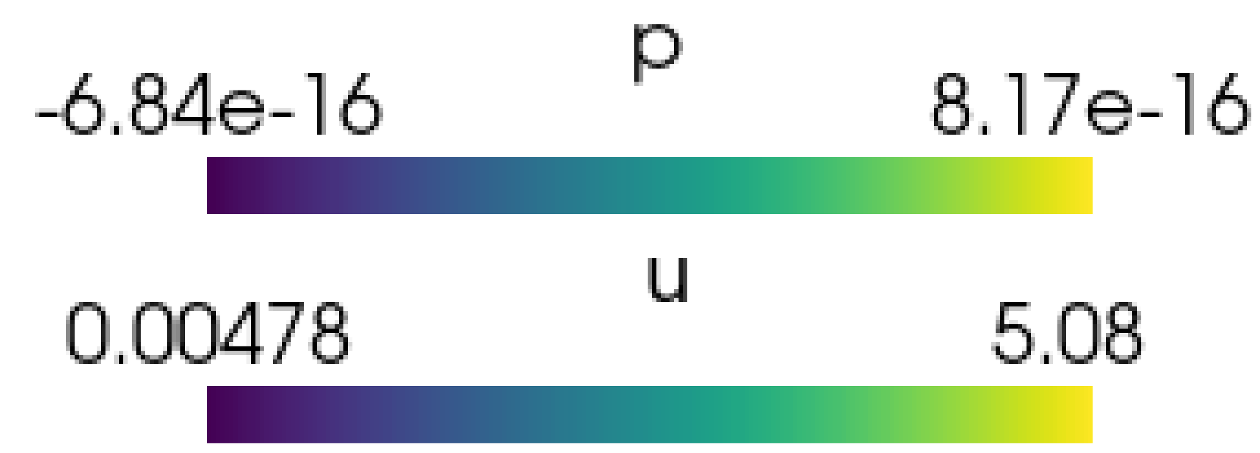
y se guarda el resultado en un archivo .vtk mediante el código

```
pb.save_state("EntregaWalterC.vtk",vec)
```

CONCLUSIONES

Se verifica que las presiones son muy bajas las cuela es encuentran entre $-6.84x10^{-16}$ y $8.17x10^{-16}$. Las velocidades son distintas en todas las regiones y fluctúan entre 0.00478 y 5.08. Además como se preveía, el gas se iba a comportar de una forma en el cual se fromarian vortices.





Referências

[1] Arnau Pont, Ramon Codina, Joan Baiges, and Oriol Guasch. Unified solver for fluid dynamics and aeroacoustics in isentropic gas flows. *Journal of Computational Physics*, 363:11–29, 2018.