UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

FÍSICA COMPUTACIONAL

Producto Integrador de Aprendizaje:

“Distribución del campo de velocidades de un fluido newtoniano con flujo laminar”

EQUIPO 5

INTEGRANTES

Dylan Eduardo Soto Hernández 1863572
Kevin Alfredo Cansino Tortoledo 1941507
Francisco Iván Rugerio Salinas 1867280

Profesor:

Alfredo Tlahuice Flores

Fecha: 28/05/2021

Resumen:
Se desarrollaron las ecuaciones de Navier-Stokes para los siguientes casos: uno donde el campo de velocidades dependiera de una componente, y esta a su vez, dependiera de una sola variable. El otro caso fue: uno donde el campo de velocidades dependiera de una componente, y esta a su vez, dependiera de dos variables.

Esto para un fluido laminar, newtoniano y estado estacionario; el cual fluye por una cavidad rectangular. Es decir, un flujo de Poiseuille.

Luego se proponen las soluciones analíticas, para ambos tipos de E.D.P. Luego se procede a resolver ambas por el método de diferencias finitas, para, por último, comparar los resultados numéricos con los analíticos.

Caso 1:

Se quiere encontrar el campo de velocidades de un fluido: Newtoniano, de movimiento laminar y de estado estacionario en $\mathbb{R}^2$, que fluye por dos paredes paralelas. Donde dicho campo tiene una componente, que depende de una variable.

Desarrollo de expresión analítica para el caso 1, siendo $\boldsymbol{u(y)}$ una función de la velocidad

Para nuestro primer caso, desarrollaremos una expresión que nos dé la velocidad con respecto al eje "y".

Comenzamos con establecer la ecuación de Navier Stokes que expresa la segunda ley de Newton:

$$\rho\frac{Du}{Dt} = \rho g_x - \frac{\partial P}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + \frac{1}{3}\mu\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \text{ (1)}$$
$$\rho\frac{Dv}{Dt} = \rho g_y - \frac{\partial P}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) + \frac{1}{3}\mu\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \text{ (2)}$$
$$\rho\frac{Dv}{Dt} = \rho g_z - \frac{\partial P}{\partial z} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) + \frac{1}{3}\mu\frac{\partial}{\partial z}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \text{ (3)}$$

Siendo (1) la componente de la ecuación en el eje "x", (2) en el eje "y" y (3) en el eje de "z".

Es necesario establecer ciertas condiciones para lograr desarrollar una expresión que sea óptima:

A. **Flujo Newtoniano:** Este tendrá una viscosidad constante por lo que $\mu = Cte$.
B. **Flujo Laminar:** El fluido será ordenado y predecible, para hacer más ameno su estudio, se aplicará en 2D y de forma estacionaria, por lo que se podría analizar con una fotografía o un vídeo, ya que no variaría.
C. **Flujo Incompresible:** $\rho \approx cte \Leftrightarrow disV = 0$.

D. **Flujo Bidimensional en el plano xy:** Con esta representación podemos observar que el campo de velocidades no varía en la dirección del eje “z”, ni hay componentes de la velocidad en ese eje. Dándonos así que $w = 0$; $\frac{\partial}{\partial z} = 0$.
E. **Flujo Unidireccional:** Esto implica que todos los vectores de velocidad apuntan en la misma dirección, que en nuestro caso será en la dirección del eje x. Lo cual es $u \neq 0$; $v = w = 0$.
F. **Flujo Estacionario:** Que el campo de las velocidades y otras magnitudes que se obtengan en la solución no cambiará a lo largo del tiempo.
G. **Velocidad en las paredes:** La velocidad en los extremos de cada pared será igual a cero, siendo $u(0) = 0$ y $u(h) = 0$.

Tenemos en general que la ecuación de Navier Stokes está dada por $\frac{D\rho}{Dt} + \rho\underline{\nabla}.\underline{V} = 0$ pero de la condición C, podemos eliminar $\frac{D\rho}{Dt}$ quedando lo siguiente: $\rho\underline{\nabla}.\underline{V} = 0$. Vamos a centrarnos en la divergencia de la velocidad la cual se da como $\underline{\nabla}.\underline{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$.

Con la hipótesis D eliminamos $\frac{\partial v}{\partial y}$y con la E $\frac{\partial w}{\partial z}$. Esta es una manera más de representar la condición G que establecimos en un inicio.

También que junto a E podemos eliminar toda la componente del eje “z”, es decir, (3). Volvemos a organizar la ecuación de Navier Stokes con las modificaciones que tiene.

$\rho(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}) = \rho g_x - \frac{\partial P}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + \frac{1}{3}\mu\frac{\partial}{\partial x}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})$ (4)
$\rho(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}) = \rho g_y - \frac{\partial P}{\partial y} + \mu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) + \frac{1}{3}\mu\frac{\partial}{\partial y}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})$ (5)

Utilizando la hipótesis C, eliminamos $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ de (4), también $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ de (5).
Con la hipótesis E, eliminamos $\frac{\partial u}{\partial x}$, $\frac{\partial^2 u}{\partial x^2}$ de (4).

Volviendo a acomodar la ecuación de Navier Stokes:

$\rho g_x - \frac{\partial P}{\partial x} + \mu\frac{\partial^2 u}{\partial y^2} = 0$ (6) ecuación del modelo
$\rho g_y - \frac{\partial P}{\partial y} = 0$ (7)

Con la ecuación (6) vamos a considerar la componente del peso en dirección del eje “x” así como la variación de presión en la misma dirección mediante: $P_x = P - \rho g_x$, denominando a P_x como “desequilibrio hidrostático en x”.

En la ecuación (7) se describe se describe en la dirección del eje “y”. el peso del fluido en el equilibrio hidrostático con la componente del eje “y” del gradiente de presión.

La ecuación en dirección “x” puede escribirse como:
$\int_0^L \rho g_x - \frac{dP_x}{dx} + \mu\frac{\partial^2 u}{\partial y^2} dx = 0$ (8)

Dando como resultado:
$$-(P_L - P_0) + \mu \frac{\partial^2 u}{\partial y^2} L = 0 \text{ (9)}$$

Con (9) hacemos un despeje.
$$\frac{\partial^2 u}{\partial y^2} = \frac{(P_L - P_0)}{\mu L} = \frac{-\Delta P}{\mu L} \text{ (10)}$$

De (10) observamos que la delta es mayor debido a que la presión inicial sea mayor que la final para que el flujo se empuje en la dirección del eje x que deseamos. Con ello observamos que la componente u de la velocidad solo depende del eje de "y". Es necesario integrar por ambas partes la ecuación (10) con respecto a "y", lo cual da:
$$\frac{\partial u}{\partial y} = \frac{-\Delta P}{\mu L} y + C_1 \text{(11)}$$

Ahora es necesario que integremos (11) para obtener el valor de $u(y)$:

$$u(y) = \frac{-\Delta P}{\mu L} \frac{y^2}{2} + C_1 y + C_2 \text{(12)}$$

Ahora pasamos a establecer las condiciones de borde necesarias. Para ello es necesario la condición G.

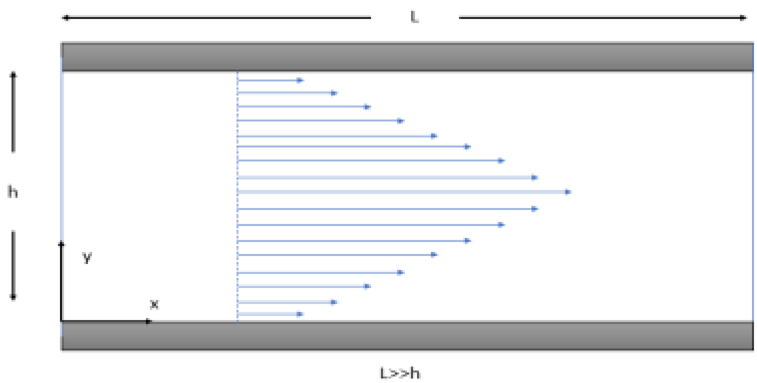
Trabajando con las constantes que se han obtenido:
Para C_2:
$$u(0) = 0 \Rightarrow C_2 = 0 \text{(13)}$$
Para C_1:
$$u(h) = 0 \Rightarrow \frac{-\Delta P}{\mu L} \frac{h^2}{2} + C_1 h \Rightarrow C_1 = \frac{\Delta P h}{2\mu L} \text{(14)}$$

Ahora es necesario que tomemos (13) y (14) las sustituyamos en (12), dando:

$$u(y) = \frac{\Delta P}{2\mu L} (\mathrm{h}y - y^2) \quad \text{(15)}$$ i.e. distribución de velocidad entre ambas placas.

Esta expresión dada en (15) es la solución analítica que se utilizará a lo largo del trabajo para comparar nuestros resultados con respecto a la solución numérica que se desarrolló y con ello saber la efectividad de nuestro trabajo.

Con u(y) podemos tener un esquema analítico, con el cual compararemos los resultados del método numérico, el cual es:



Desarrollo numérico del caso 1

Sean las ecuaciones para resolver numéricamente:

$\frac{\partial p}{\partial x} = \mu \frac{\partial^2 u}{\partial^2 y} + p g_x$ (8)

$\frac{\partial p}{\partial x} = 0$ (9)

Reordenando la ecuación 8 se puede obtener la siguiente forma:

$\frac{\partial^2 u}{\partial^2 y} = \frac{1}{\mu}(\frac{\partial p}{\partial x} - p g_x)$ (10)

Para este caso, se está considerando que las fuerzas gravitatorias que actúan en el fluido son muy pequeñas en magnitud en comparación con aquellas fuerzas debidas a las variaciones de presión en el fluido, por lo tanto, la ecuación 10 se reescribe de la siguiente manera:

$\frac{\partial^2 u}{\partial^2 y} = \frac{1}{\mu}\frac{\partial p}{\partial x}$ (11)

Esta ecuación se puede resolver mediante el método de *diferencias finitas*. La geometría del espacio en el que flujo se mueve dispone de una forma rectangular, por lo que, considerando que: la velocidad *u* varía únicamente en *y* y que se tienen elementos uniformes; entonces, con base al método, se obtiene que:

$\frac{\partial^2 u}{\partial^2 y} = u_{yy} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta y^2}$ (12)

Donde: Δy es la longitud de cada elemento de los *N* elementos que estamos considerando desde la placa inferior hasta la placa superior. Esto conduce a una discretización, de forma que:

$$\Delta y_i = \Delta y = \frac{h}{N}, \qquad \forall\ i \in [1, h]$$

Sustituyendo la ecuación (12) anterior en la ecuación (11), se obtiene:

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta y^2} = \frac{1}{\mu}\frac{dp}{dx}$$

$$u_{i+1} - 2u_i + u_{i-1} = \frac{1}{\mu}\frac{dp}{dx}\Delta y^2 \quad (11)$$

Para un flujo laminar que se mueve entre placas rígidas, la velocidad es igual a cero en aquellos puntos en que el fluido está muy cerca de la pared, por lo que podemos definir las siguientes condiciones tipo *Dirichlet*:

$$u(0) = 0$$

$$u(h) = 0$$

Estableciendo que $\lambda = \frac{1}{\mu}\frac{dp}{dx}\Delta y^2$ dar valores a *i* en la ecuación (12), obtenemos un sistema de ecuaciones:

$$u_3 - 2u_2 + u_1 = \lambda\Delta y^2$$
$$u_4 - 2u_3 + u_2 = \lambda\Delta y^2$$
$$u_5 - 2u_4 + u_3 = \lambda\Delta y^2$$

:

:

$$u_{i+1} - 2u_i + u_{i-1} = \lambda\Delta^2$$

Donde: $u_1 = 0$ y $u_h = u_N = 0$. Expresando en forma matricial el sistema anterior, se obtiene (para el caso en que $N = 6$):

$$\begin{bmatrix} u_1 & 0 & 0 & \dots & & 0 \\ u_1 & -2u_2 & u_3 & \dots & & 0 \\ 0 & u_2 & -2u_3 & u_4 & \dots & 0 \\ 0 & 0 & u_3 & -2u_4 & u_5 & 0 \\ 0 & 0 & 0 & u_4 & -2u_5 & u_6 \\ 0 & 0 & 0 & 0 & 0 & u_6 \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ 0 \end{bmatrix}$$

Luego, la matriz del miembro izquierdo se puede factorizar, de modo que se obtenga el producto entre la matriz de coeficientes y el vector u_i, el cual es nuestro vector solución:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & & 0 \\ 1 & -2 & 1 & \dots & & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ 0 \end{bmatrix}$$

Finalmente, si invertimos la matriz de coeficientes en el lado derecho, obtenemos una expresión matricial que nos permite calcular a cada uno de elementos del campo de velocidades (con $N = 6$):

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & & 0 \\ 1 & -2 & 1 & \dots & & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ \lambda\Delta y^2 \\ 0 \end{bmatrix}$$

Extendiéndose al caso general en que $N = h$, se obtiene:

$$
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ \vdots \\ u_h \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & \cdots & & & & 0 \\
1 & -2 & 1 & \cdots & & & & \\
0 & 1 & -2 & 1 & \cdots & & & 0 \\
\vdots & \cdots & 1 & -2 & 1 & 0 & \cdots & 0 \\
 & \cdots & & 1 & -2 & 1 & 0 & \cdots \\
 & \cdots & & & 0 & 1 & -2 & 1 \\
 & & & & & & 1 & -2 \\
0 & \cdots & & & & & & 1
\end{bmatrix}^{-1}
\begin{bmatrix} a \\ \lambda \Delta y^2 \\ \lambda \Delta y^2 \\ \lambda \Delta y^2 \\ \lambda \Delta y^2 \\ \lambda \Delta y^2 \\ \vdots \\ b \end{bmatrix}
$$

Explicación del código:

Comenzamos por definir las matrices y vectores utilizados: a es la matriz de los coeficientes y c es la matriz inversa de a (ambas con dimensiones nxn); b1 es el vector que contiene a las condiciones de borde y a los demás términos a la derecha del sistema de ecuaciones; d1 es el vector resultante o solución y d2 son los incrementos en y. Por último, cabe resaltar, que todos los vectores tienen dimensiones: nx1.

```
program PIAV5_E05
 !Este programa calcula la sol. num. de una e.d.p. (Uyy=Cte.)
 !Con u=u(y)

implicit none
real*8 dy, d1, h,la
integer*8 i, j, k, n
real*8, allocatable, dimension (:,:):: a,c !nxn
real*8, allocatable, dimension (:,:):: b1,d1,d2  !nx1

!incremento en y
dy=h/n
!fijacion de elementos
allocate(a(n,n))
allocate(b1(n,1))
allocate(c(n,n))
allocate (d1(n,1))
allocate (d2(n,1))
```

Hasta ahora solo se han definido los parámetros dentro del programa, por lo que se procede a calcular la matriz de coeficientes. Esto se hizo tomando en cuenta la diagonal principal de la matriz, pues para cualquier N, esta siempre es: -2. Por lo tanto, si los contadores i y j son iguales, se aplica la fórmula de diferencias finitas.

Esto anterior, solo para los elementos ij que ambos contadores sean mayores a 2 y menores a n. De lo contrario, los elementos ij serán: condiciones de borde o los ceros.

```
!**********************************************************************
      subroutine matriz1(a,n)
      integer*8 i, j, k, n
      double precision a(n,n)

      Do i=1, n     !Llenado de a
      Do j=1, n

      if (i.ge.2) then
      if (i.eq.j) then    !coeficientes
      a(i,j)= -2
      a(i-1,j) = 1
      if (i.lt.n) then
      a(i+1,j)= 1
      end if
      end if
      else
      a(i,j) = 0          !los demas elementos
      end if
      a(1,1)= 1       !condiciones de borde u(0)
      a(1,2)= 0
      a(2,1)= 1
      a(n,n)= 1       !condiciones de borde u(h)
      a(n,n-1)= 0
      end do
      end do

      end subroutine
!**********************************************************************
```

Lo siguiente es calcular la matriz inversa de a, esto se hizo con un método basado en la factorización LU. Esta parte del programa NO es de nuestra autoría y fue tomada de un programa encontrado en internet (citado en las referencias). Por lo tanto, solo nos limitaremos a utilizarlo sin explicar más.

Para determinar el vector 1, vector a la derecha del sistema, no hicimos más que definir cada elemento ij a un término constante, además de poner las condiciones de borde para los elementos 1j y n1.

```
!**********************************************************************
      subroutine matriz2(b1,n,la,dy,a1,a2)
      real*8 dy, dl, h,la,a1,a2
      integer*8 i, j, k, n
      double precision b1(n,1)

      Do i=1,n
      Do j=1,1

      if(i.eq.1) then !condicion de borde
      b1(i,j)= a1

      else
      b1(i,j)= la*(dy**2)
      end if

      b1(n,1)= a2    !condicion de borde
      end do
      end do
      end subroutine
!**********************************************************************
```

En el siguiente vector, definido como los incrementos en y, i.e. los puntos a los que corresponde el vector solución. Fue obtenido multiplicando la constante delta y, por el contador i, que llega hasta n.

```
!**********************************************************************
      subroutine matriz5 (d2,dy,n)
      integer*8 i, j, k, n
      real*8 dy
      double precision d2(n,1)

      Do i=1, n  !calculo de incrementos
      Do j=1, 1

      d2(i,j)=i*dy

      end do
      end do
      end subroutine
!**********************************************************************
```

Definimos el vector solución, es decir, el vector resultante de la multiplicación: matriz inversa de los coeficientes y el vector 1. Este fue encontrado utilizando la definición de la multiplicación de matrices, donde: d1 representa el vector solución.

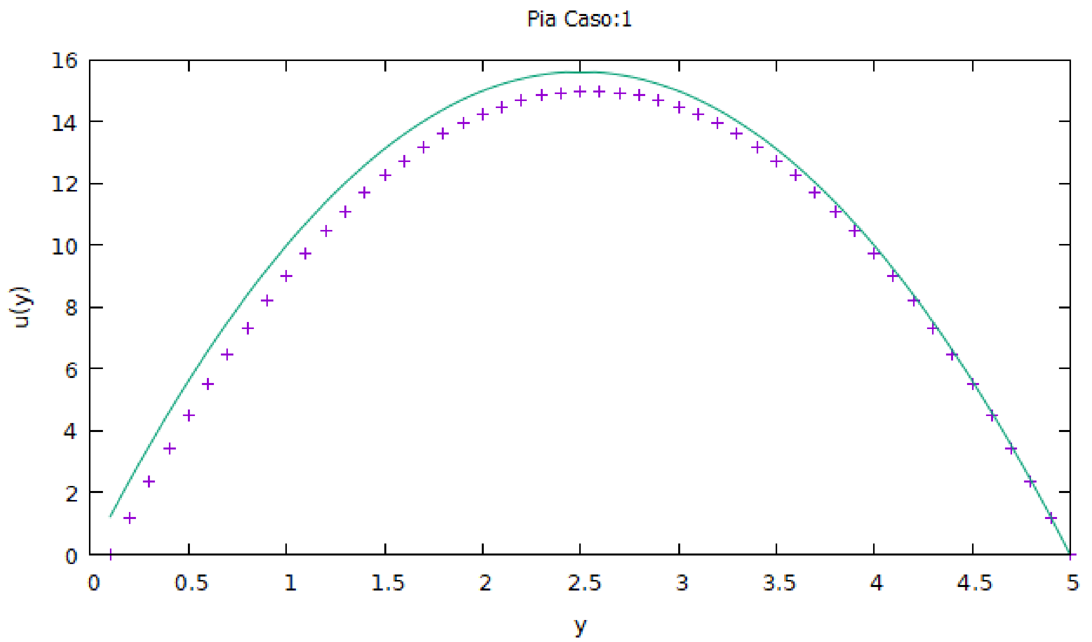
```
!**********************************************************************
      subroutine matriz4 (c,b1,d1,n)
      integer*8 i, j, k, n
      double precision c(n,n), b1(n,1), d1(n,1)

      !calculo del vector sol.  i.e. multiplicacion de c*b1
      Do i=1, n
      Do j=1, 1

      d1(i,j)= 0

      Do k=1, n
      d1(i,j)=  d1(i,j)+ ( c(i,k)*b1(k,j) )
      end do

      end do
      end do
      end subroutine
!**********************************************************************
```

Por último, concentramos todo lo necesario para graficar ambas soluciones en subrutinas de nombre: matriz1 (matriz a), matriz2 (vector 1), matriz3 (matriz inversa de a), matriz4 (vector solución) y matriz5 (vector de los incrementos en y). Dichas fueron llamadas al programa principal, para después graficar los vectores: d2 y d1 i.e. las velocidades del fluido a un cierto punto.

Resultados:

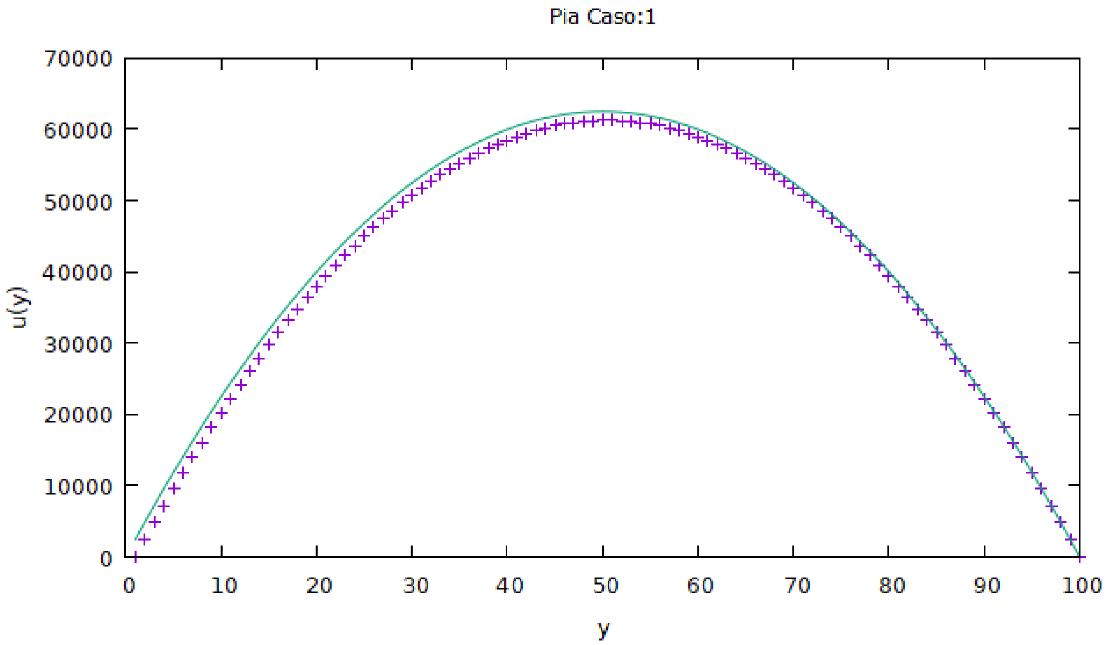
Como las soluciones más precisas requieren un mayor número de subintervalos, para este caso y el subsecuente, lo que hace inviable una tabla con todos los puntos. Nos limitaremos a mostrar los pantallazos: de la ejecución y las gráficas de ambas soluciones (con diferentes parámetros).

Con: N = 50, h = 5, lamda = -5.



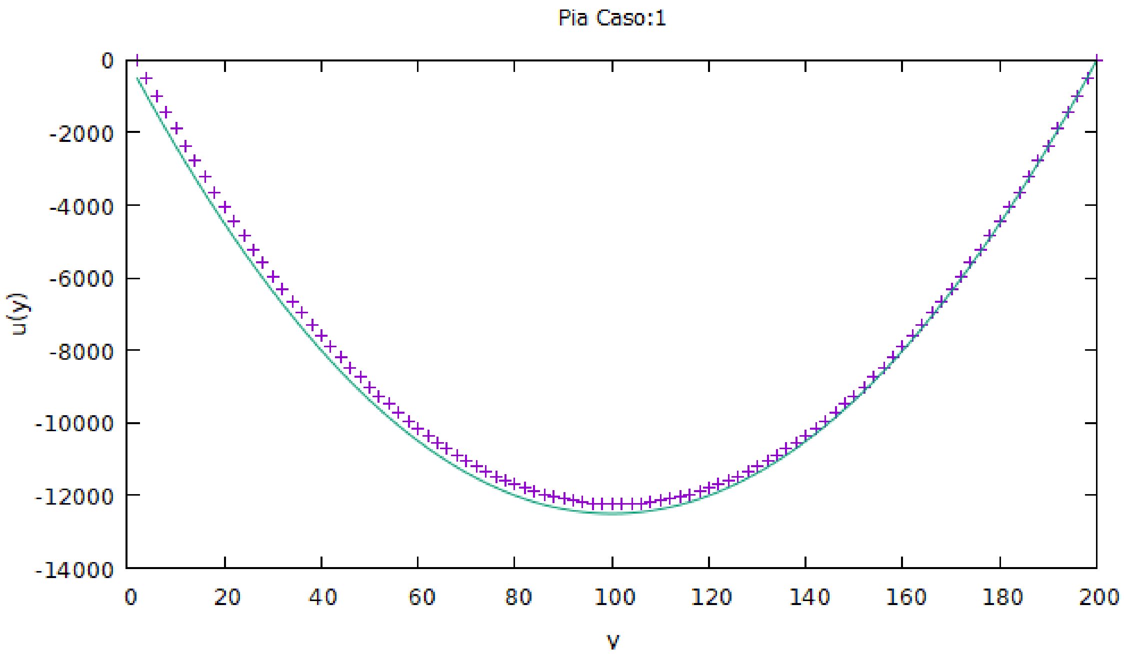
```
*******************************************************
 Fijacion de parametros:
 Este programa calcula la sol. num. de una e.d.p.       (Uyy=Cte.), con u=u(y)
 Donde la e.d.p proviene del planteamiento del caso: 1
*******************************************************
 Escriba el numero de subintervalos (N) entero:
50
 Escriba la longuitud en y (h)
5
 Escriba el termino cte. lamda (lamda<0):
-5
 Escriba las condiciones Dirichet:
 En u(0):
0
 En u(h):
0
*******************************************************
 Matriz A:
*******************************************************

 Inversa matriz A^{-1}:
*******************************************************
 Vector 1:
*******************************************************
 Inc. en y:
*******************************************************
 Vector sol.:
*******************************************************
 Aqui acaba el programa:)
PAUSE
To resume execution, type go.  Other input will terminate the job.
```

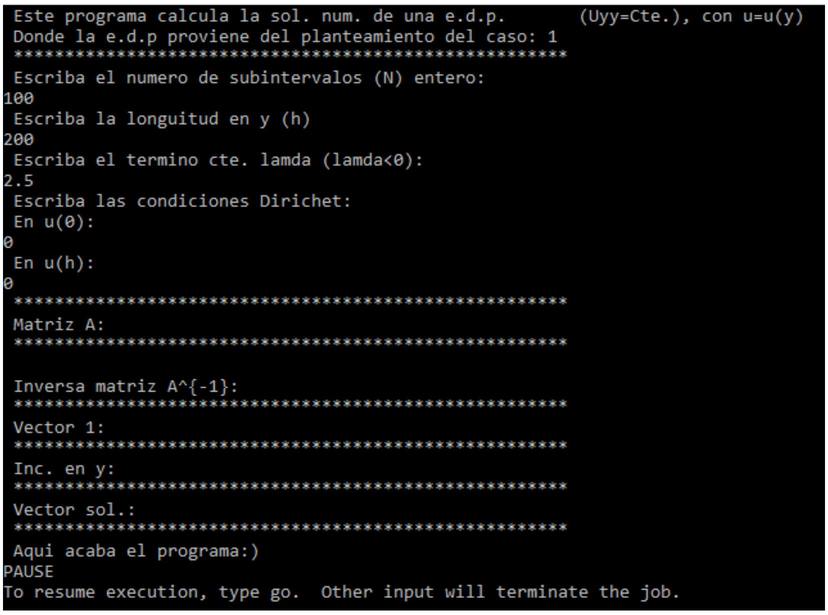
Con: N = 1000, h = 100, lamda = -50




```
 Este programa calcula la sol. num. de una e.d.p.       (Uyy=Cte.)
 Donde la e.d.p proviene del planteamiento del caso: 1
*******************************************************
 Escriba el numero de subintervalos (N) entero:
1000
 Escriba la longuitud en y (h)
100
 Escriba el termino cte. lamda (lamda<0):
-50
 Escriba las condiciones Dirichet:
 En u(0):
0
 En u(h):
0
*******************************************************
 Matriz A:
*******************************************************

 Inversa matriz A^{-1}:
*******************************************************
 Vector 1:
*******************************************************
 Inc. en y:
*******************************************************
 Vector sol.:
*******************************************************
 Aqui acaba el programa:)
PAUSE
To resume execution, type go.  Other input will terminate the job.
```

Con: N = 100, h = 200, lamda = 2.5





```
 Este programa calcula la sol. num. de una e.d.p.        (Uyy=Cte.), con u=u(y)
 Donde la e.d.p proviene del planteamiento del caso: 1
 ******************************************************
 Escriba el numero de subintervalos (N) entero:
100
 Escriba la longuitud en y (h)
200
 Escriba el termino cte. lamda (lamda<0):
2.5
 Escriba las condiciones Dirichet:
 En u(0):
0
 En u(h):
0
 ******************************************************
 Matriz A:
 ******************************************************

 Inversa matriz A^{-1}:
 ******************************************************
 Vector 1:
 ******************************************************
 Inc. en y:
 ******************************************************
 Vector sol.:
 ******************************************************
 Aqui acaba el programa:)
PAUSE
To resume execution, type go.  Other input will terminate the job.
```

Caso 2

Se quiere encontrar el campo vectorial de un fluido: Newtoniano, de movimiento laminar y de estado estacionario en $\mathbb{R}^2$, que fluye en una cavidad rectangular de dimensiones: l, h, z.

Donde dicho campo tiene una componente, que depende de dos variables. Además, se cumplen las mismas condiciones que en el caso 1, a excepción del apartado donde se menciona que u = u(y). Pues para este caso: u = u (x, y).

Expresión analítica, siendo $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{y})$ una función de la velocidad

La solución analítica para este caso es un tanto más complicado, por lo que nos limitaremos a mostrar la E.D.P. del modelo y su solución (encontrada por: Boussinesq (1868)). La cual se desarrolla en una cavidad rectangular de dimensiones: y ∈ [0, h], x ∈ [0, l].

Usando el mismo sistema de ecuaciones diferenciales Navier-Stokes del caso 1, y usando las mismas premisas excepto la relativa al campo de velocidades, tenemos que:

$$\rho\frac{Du}{Dt} = \rho g_x - \frac{\partial p}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}) + \frac{1}{3}\mu\frac{\partial}{\partial x}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})$$

Sea la ecuación de continuidad:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

$$\Rightarrow \rho\frac{Du}{Dt} = \rho g_x - \frac{\partial P}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

Como no consideramos efectos gravitacionales y se cumple que:

$$\frac{Du}{Dt} = g_x = 0$$

$$\frac{\partial p}{\partial x} = -P = cte.$$

Por lo tanto, la ecuación de nuestro modelo queda como:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{-P}{\mu}$$

La cual es una E.D.P elíptica, parecida a la ecuación de Laplace, que será resuelta por medio de diferencias finitas. Y que en particular tiene la siguiente solución analítica:

$$u(x, y) = \frac{Ph^2}{2\mu}\left[\frac{y}{h} - \left(\frac{y}{h}\right)^2\right] + f(x, y)$$

Donde f tienen la siguiente forma:

$$f(x,y) = \frac{-4Ph^2}{\mu\pi^3}\sum_{n=0}^{\infty}\frac{\left[\,sinh\frac{(2n+1)\pi x}{h} + \; sinh\frac{(2n+1)\pi(l-x)}{h}\right]}{(2n+1)^3\; sinh\frac{(2n+1)l}{h}}\; sin\frac{(2n+1)\pi y}{h}$$

Condiciones de borde:

Si tomamos a la función analítica como las velocidades en del fluido, y, tomamos en cuenta que este fluido está en una cavidad rectangular de las dimensiones ya dichas.

Tendríamos velocidades iguales a cero en los bordes de y (en (x, 0), (x, h)) por las condiciones de flujo laminar. Y en los bordes de x (en (0, y), (l, y)) tendríamos unas velocidades que dependieran de y. Pues f cumple que es igual a cero en: y = 0 o y = l.

$$En\ x:\ u(0,y) = -\frac{Ph^2}{2\mu}\left[\frac{y}{h} - \left(\frac{y}{h}\right)^2\right]$$

$$u(l,y) = -\frac{Ph^2}{2\mu}\left[\frac{y}{h} - \left(\frac{y}{h}\right)^2\right]$$

$$\begin{aligned} En\ y:\ u(x,0) &= 0 \\ u(x,h) &= 0 \end{aligned}$$

Desarrollo numérico del caso 2

Para resolver numéricamente la ecuación del modelo utilizaremos el método de diferencias finitas, donde las derivadas de orden n, serán calculadas por diferencias centradas. Para este caso, n = 2, lo que implica que general, las diferencias centradas tendrán la siguiente forma:

$$\frac{\partial^2\Phi}{\partial x^2} = \frac{\Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial\Phi}{\partial y} = \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y}$$

Los indices ij corresponden al mallado que se tenga con base a la discretización de la ecuacion del modelo. Ambos indices irán sumando o restando 1, a sus contadores, según sea la variable de la derivada parcial. Para derivadas mostradas, x equivale al subindice i, por lo que una derivada parcial en x, afectará al indice i. Esto no necesariamente se cumple para la siguiente discretización y mallado.

Discretización de la ecuación del modelo:

$$x \in [0,\ l]$$
$$y \in [0,\ h]$$

$$\Rightarrow \Delta y = \frac{h}{Ny} \qquad \Delta x = \frac{l}{N_x}$$

Ny: Número de nodos en y.
Nx: Número de nodos en x.

Para propósitos de nuestro mallado, asignamos: y como el subíndice j y a x como el subíndice i.

$$x \equiv j$$
$$y \equiv i$$

Teniendo en cuenta lo anterior, la ecuación del modelo, en su forma numérica queda como:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta y^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta x^2} = -\frac{P}{\mu}$$

Despejando el elemento u (i, j) …

$$\Delta x^2(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \Delta y^2(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = -\Delta x^2 \Delta y^2 \frac{P}{\mu}$$

$$Sea:\ \sigma = \Delta x^2 \Delta y^2 \frac{P}{\mu} \wedge \beta = 2\left(\Delta x^2 + \Delta y^2\right)$$

$$\Delta x^2(u_{i+1,j} + u_{i-1,j}) + \Delta y^2(u_{i,j+1} + u_{i,j-1}) + \sigma = 2u_{i,j}\left(\Delta x^2 + \Delta y^2\right)$$

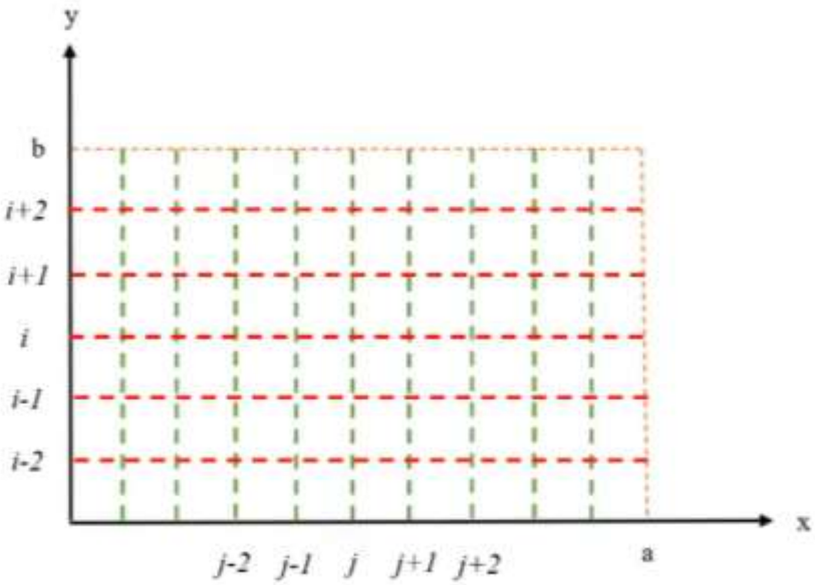
$$\therefore\ u_{i,j} = \frac{1}{\beta}\left[\Delta x^2(u_{i+1,j} + u_{i-1,j}) + \Delta y^2(u_{i,j+1} + u_{i,j-1}) + \sigma\right]$$

Esto significa que un elemento del mallado, correspondiente a la velocidad del fluido, esta supeditado a una combinación de cuatro elementos a su alrededor, todos ellos con diferente “peso” o “valor”.

La ponderación de ese “peso” viene dado por los coeficientes que acompañan a los elementos y para ese caso, se le suma un término extra, correspondiente a la igualación de la E.D.P. al valor constante P/μ.

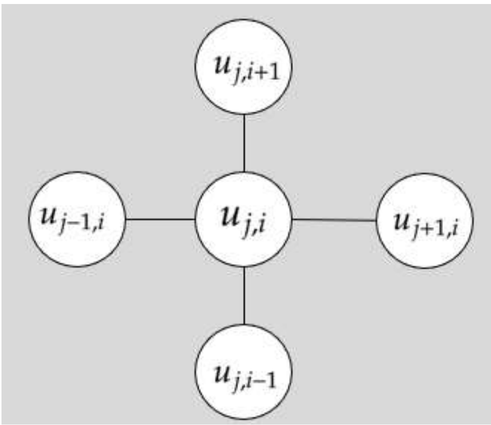
Mallado y stencil:

Como ya lo habíamos hablado, la solución numérica de la E.D.P. tendrá que ser calculada en términos de una malla con Nx(Ny) nodos en total. Cada nodo de la malla equivale a un punto de en el campo de velocidades (i.e. u (x, y) ≡ u (j, i). Como habíamos definido: i variable x y a j como variable y. El mallado para ambas dimensiones tendrá la siguiente forma:



Nota: para nuestros fines, b equivale a Ny, a equivale a Nx.

El stencil, definido grosso modo, es el arreglo geométrico en el que se relacionan los nodos conocidos con los que no. Dicho concepto varia de E.D.P. a otra. No obstante, las elípticas comparten una forma de stencil:



Para resolver a las ecuaciones con dicho tipo de stencil, se requiere conocer las condiciones de frontera (los nodos del borde). Después, se tiene que aplicar una suposición para los valores interiores de los nodos. Luego, iterando sobre los nodos interiores, y aplicando la formula del stencil, se podrá aproximar una solución para la E.D.P. elíptica.

Tomando en cuenta la forma del elemento u (i, j), este forma el stencil característico para una ecuación elíptica. Solo con la diferencia, de que cada punto a su alrededor lo afecta en mayor o menor medida dependiendo de su coeficiente y a esto se le suma un término constante. No obstante, estas diferencias son fácilmente resueltas en el código.

Además, se conocen las condiciones de frontera para la solución analítica. Por tanto, se conocen los nodos del borde, restaría aplicar la formula del elemento u (i, j) a los nodos interiores.

Sean los elementos de borde en u:

$$u_{i,1} = u_{i,nx+1} = -\frac{Ph^2}{2\mu}\left[\frac{y}{h} - \left(\frac{y}{h}\right)^2\right]$$

$$u_{1,j} = u_{ny+1,j} = 0$$

Esto implica que ya tenemos todos los elementos necesarios para desarrollar el método de diferencias finitas en fortran.

Explicación del código:

Al igual que en el código anterior definiremos las siguientes matrices: a como la matriz de la solución numérica de dimensiones Ny, Nx, a1 como la matriz de la solución analítica de las mismas dimensiones que a. Y definamos a las vectores: x1 y y1 como los incrementos en esas variables, de dimensiones Nx, 1 y Ny, 1 respectivamente.

```
program PIA_E05_C2


implicit none
real*8:: l,h,dx,dy,alfa,b,b1,b2,c1,c2,d1,d2,e1,e2,s,la,p,h1,h2
real*8:: h3,h4,h5,h6,h7,h8
real*8:: f, f1, f2, f3, f4, f5, f6, f7
integer*8:: i,j,ny,nx,k,k1
real*8, allocatable, dimension (:,:):: a, a1, a2, a3, a4
real*8, allocatable, dimension (:,:):: x1, y1

OPEN( 7, FILE= "MatrizNum.txt")
OPEN( 8, FILE= "MatrizAna.txt")
OPEN( 9, FILE= "Pia_Caso2.gpl")

write(9,*)"set xlabel ""x"" font ""Times-Roman, 10"""
write(9,*)"set ylabel ""y"" font ""Times-Roman, 10"""
write(9,*)"set zlabel ""u"" font ""Times-Roman, 10"""
write(9,*)"set title ""Pia caso 2"""
write(9,*)"set xtics font ""Times-Roman, 10"""
write(9,*)"set ytics font ""Times-Roman, 10"""
write(9,*)"set ztics font ""Times-Roman, 10"""
write(9,*)"splot ""MatrizNum.txt"", ""MatrizAna.txt"""
```

Después de fijar los parámetros para ambas soluciones, procedo al cálculo de los nodos en borde para la matriz a, así como definir un valor para los puntos interiores de a. Tenemos que los bordes en x, dependen de una función en y, por lo tanto hay que ir evaluando dicha función (en incrementos de y) para cada nodo de x.

En código esto es:

a (i, 1) = - f4(c1, la, h) a (i, nx+1) = - f4(c1, la, h)

Donde: f4 es la función que depende de y
la y h son parámetros.
C1 son los incrementos en y.

Nota: como en fortran no existen índices 0 en matrices, las matrices de ambas soluciones tendrán dimensiones: nx+1, ny+1. Así mismo para los vectores. Además, los nodos interiores, tendrán dimensiones: nx, ny.

```fortran
allocate( a(ny+1,nx+1) )
allocate( a1(ny+1,nx+1) )
allocate( x1(nx+1,1) )
allocate( y1(ny+1,1) )

!Llenado de las condiciones de frontera
Do i=1, ny+1
Do j=1, nx+1

c1= i*dy    !incremento con los que se evalua u(x,0)^ u(x,1)
!c2= (i)*dy
a(i,j)=0
a(i,1)= -1*f4(c1,la,h)   !condiciones en x
a(i,nx+1)= -1*f4(c1,la,h)

a(1,j)= d1    !condiciones en y
a(ny+1,j)=d2


end do
end do

!Llenado de vectores
Do i=1, ny+1
y1(i,1)=i*dy
End do

Do i=1, nx+1
x1(i,1)=i*dx
End do
```

Para encontrar los puntos interiores de a, se implementa la forma del elemento u (i, j), pero, adaptado al código. Aunado a esto, como la suposición inicial es muy baja, la manera de encontrar los puntos interiores tiene mucho error respecto de la solución analítica. Por ello, habrá que implementar un ciclo en que se vuelvan a calcular los puntos interiores k veces.

```fortran

!Matriz de los puntos interiores
Do k=1,k1
Do i=2,ny
Do j=2,nx
              !formula de DFM
a(i,j)=(b2/b)*(a(i+1,j)+a(i-1,j))+(b1/b)*(a(i,j+1)+a(i,j-1)) + s/b

end do
end do
end do

!Matriz de la sol. ana.
Do i=1, ny+1
Do j=1, nx+1

e1=j*dx
e2=i*dy
a1(i,j)= f(e1,e2,la,h,1)

 !incremento con los que se evalua u(x,0)^ u(x,1)

a1(i,1)= -1*f4(e2,la,h)   !condiciones en x
a1(i,nx+1)= -1*f4(e2,la,h)

a1(1,j)= d1    !condiciones en y
a1(ny+1,j)= d2

end do
end do

```

Asimismo, en la imagen anterior se llena la matriz a1, la cual es la matriz de los resultados analíticos. Donde, análogamente, se llenan las mismas condiciones de borde. Pero, cada elemento a1(j, i), deber ser igualado con f (i.e. la función definida como solución analítica en el programa).

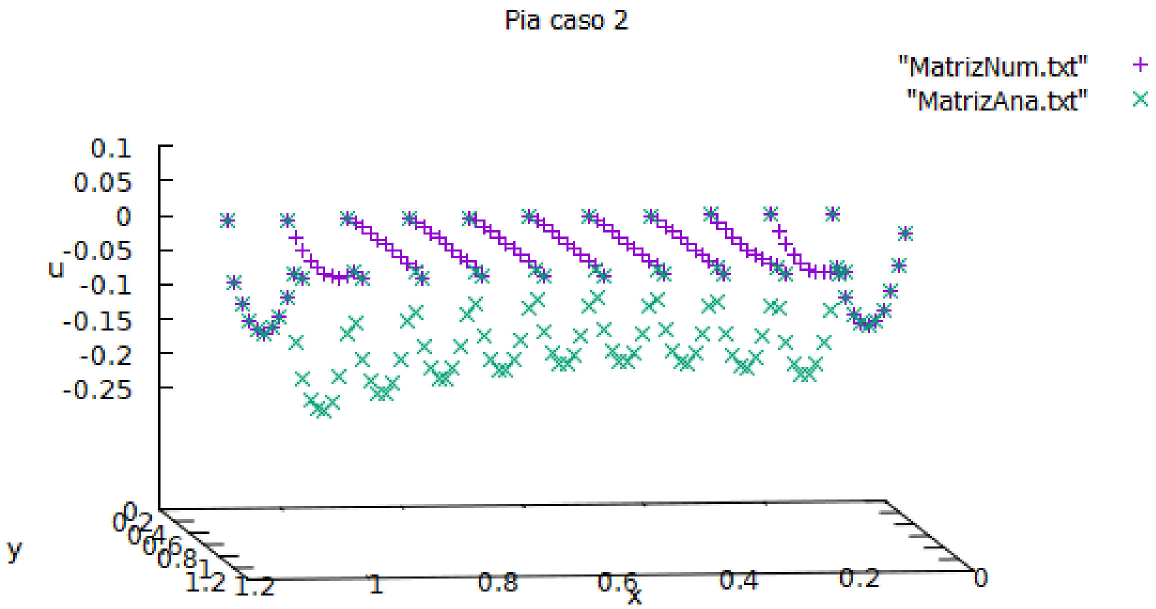
Por último, ambas matrices y sus respectivos incrementos son puestos en archivos .txt, para su posterior vista en gnuplot.

Nota: la solución analítica fue desarrollada, en el programa, hasta el termino n = 5 de la sumatoria.

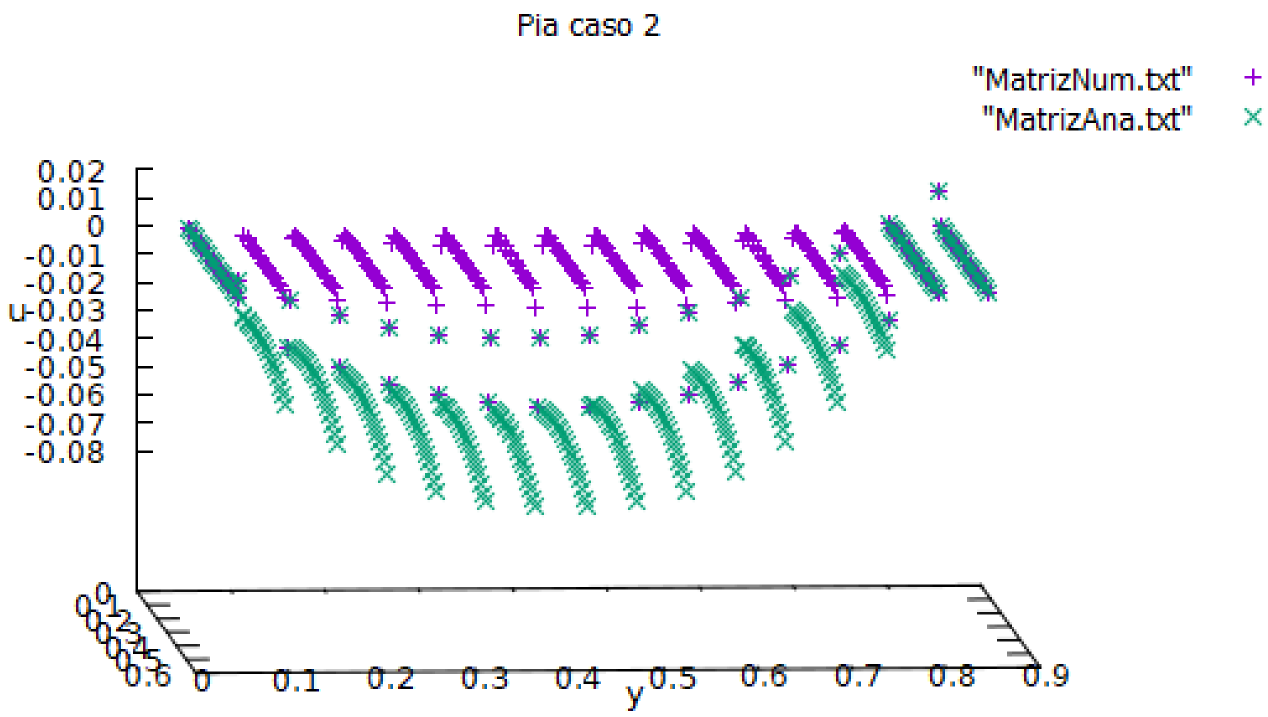
Resultados:

Al igual que el caso anterior, nos limitaremos a mostrar los pantallazos de las gráficas de ambas soluciones (con diferentes parámetros).

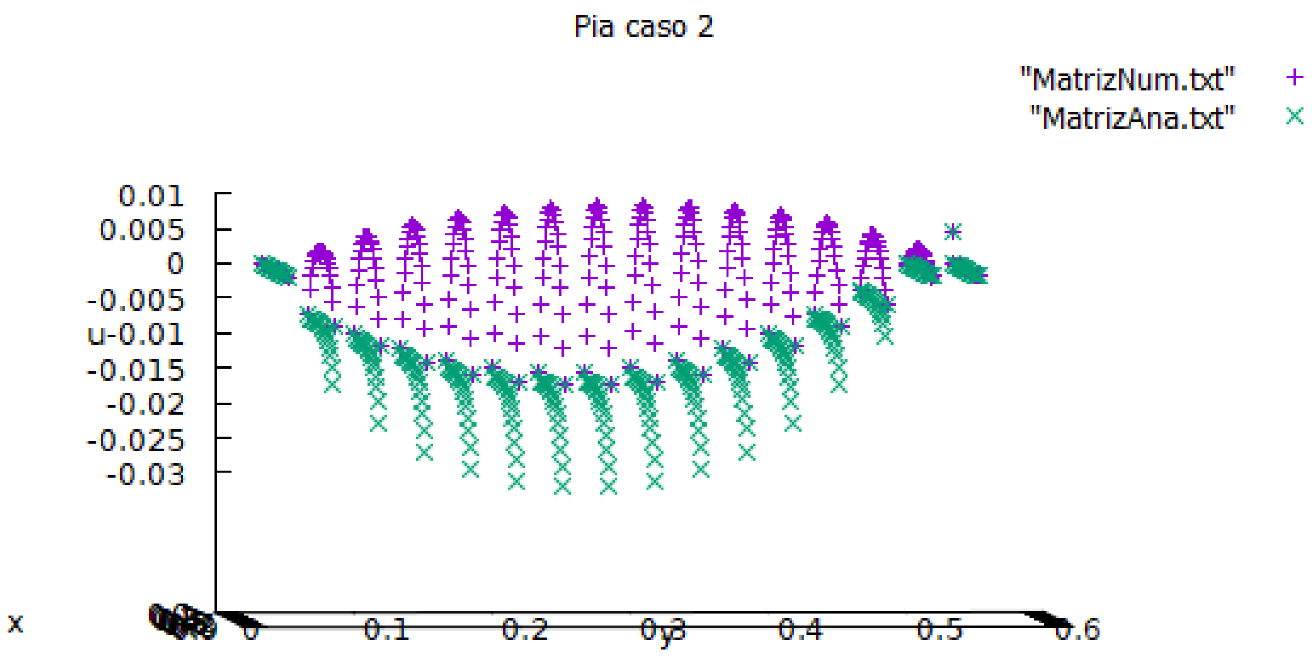
Con: Nx = Ny= 10, n = 0,1,…,5, h= l = 1, lamda = k = 1



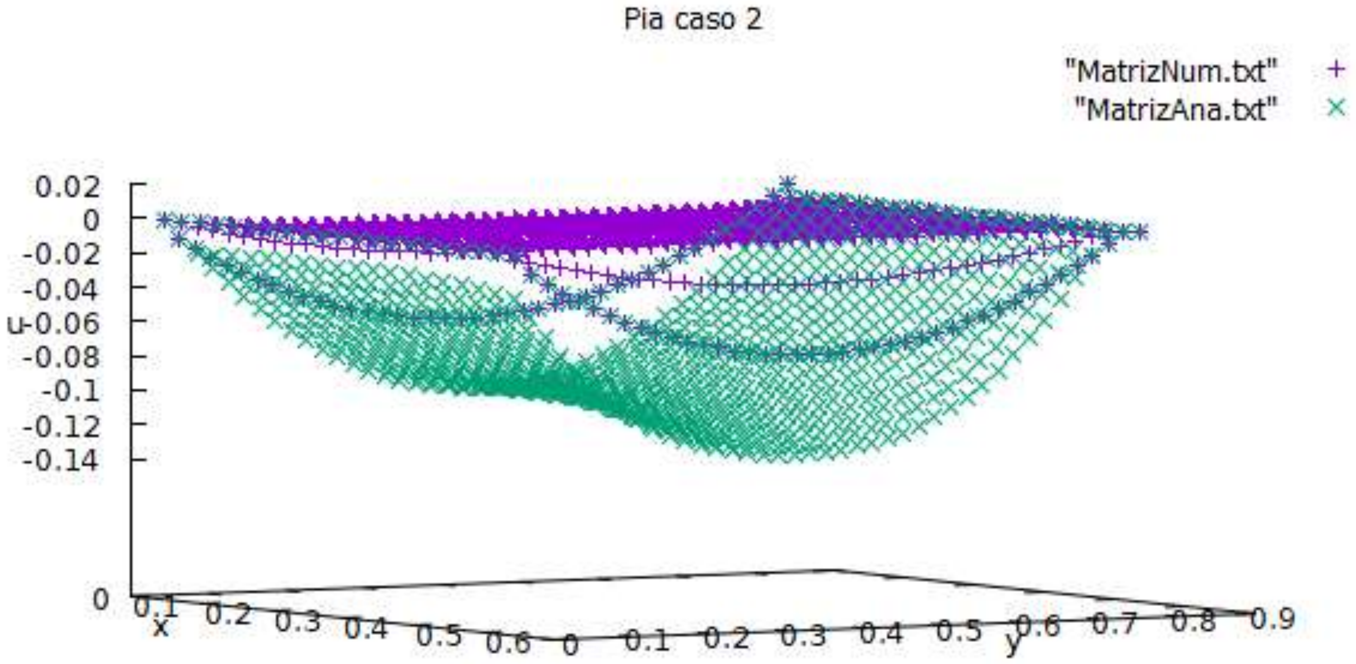
Con: Nx = 15, Ny = 15, n = 0,1,…,5 , h = 0.5, l = 0.8, lamda = 0.5, k = 1



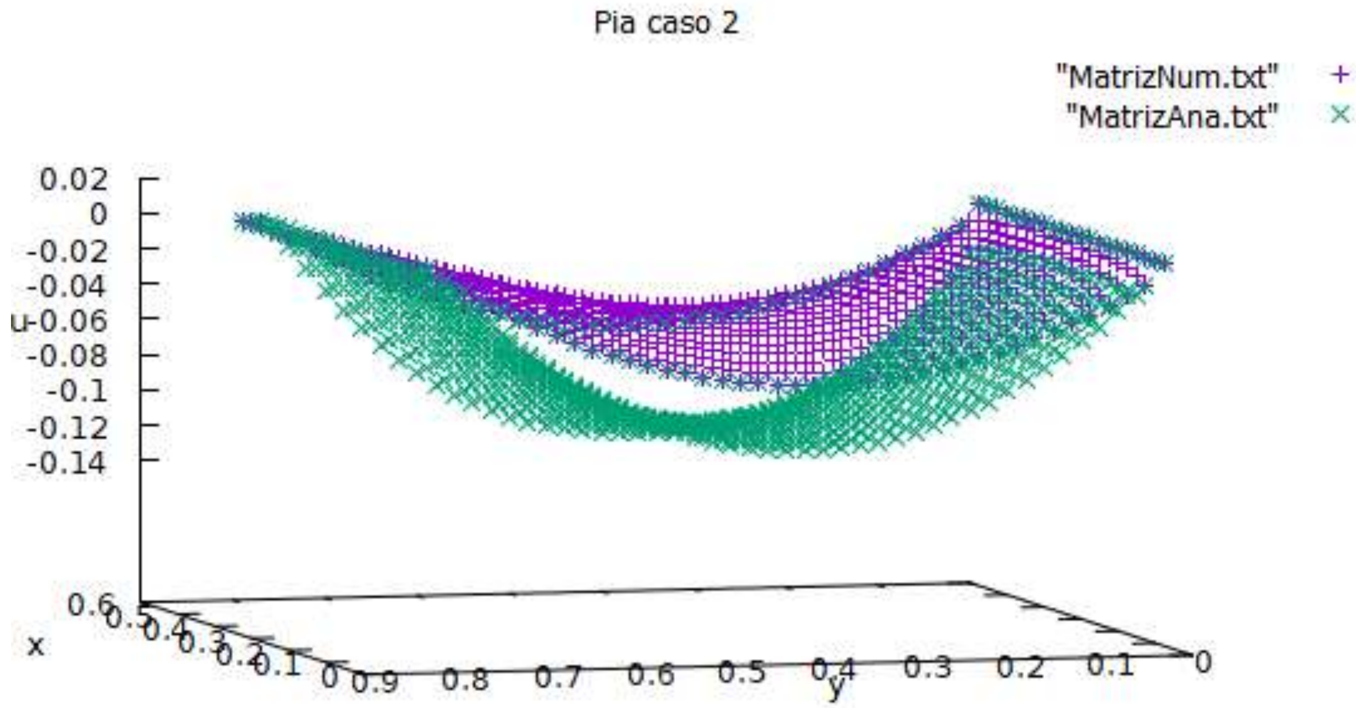
Con: Nx = 15, Ny = 15, n = 0,1,…,5 , h = 0.5, l = 0.8, lamda = 0.5, k = 5000



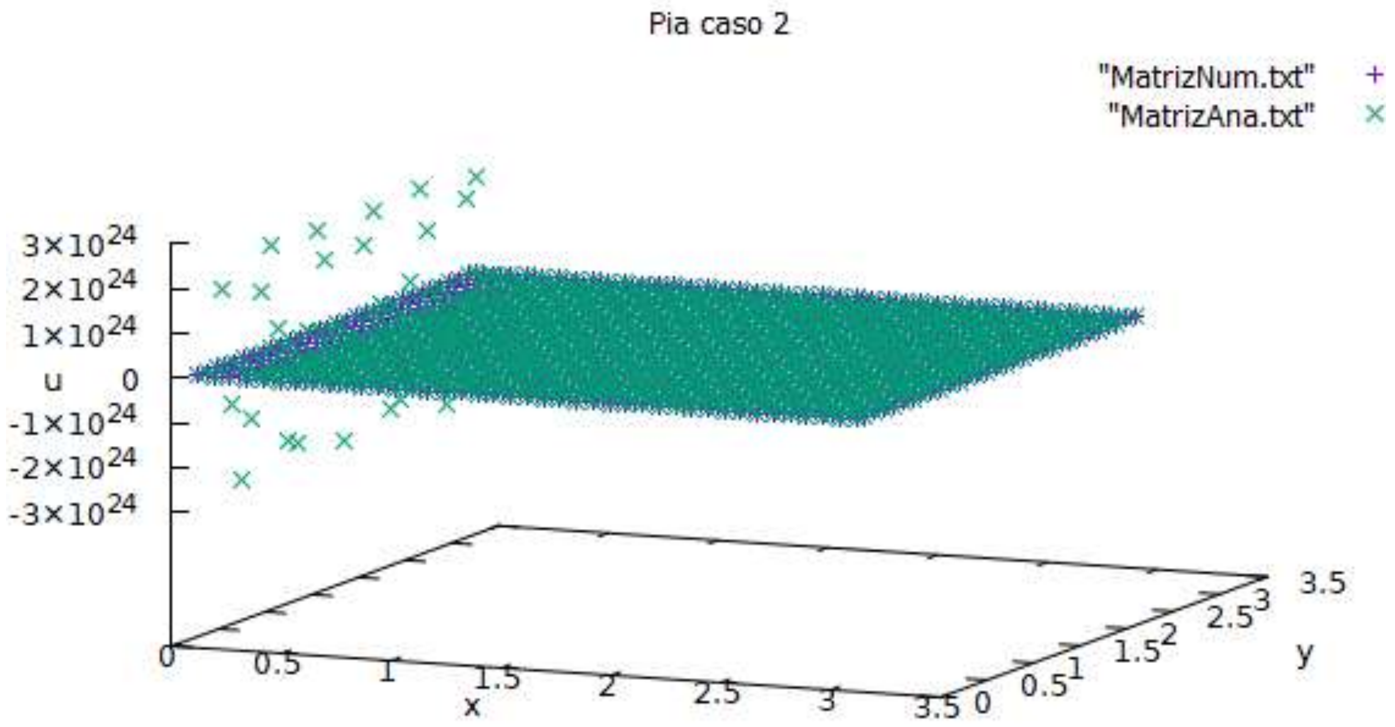
Con: Nx = 20, Ny = 40, n = 0,1,…,5 , h = 0.8, l = 0.5, lamda = 0.8, k = 1



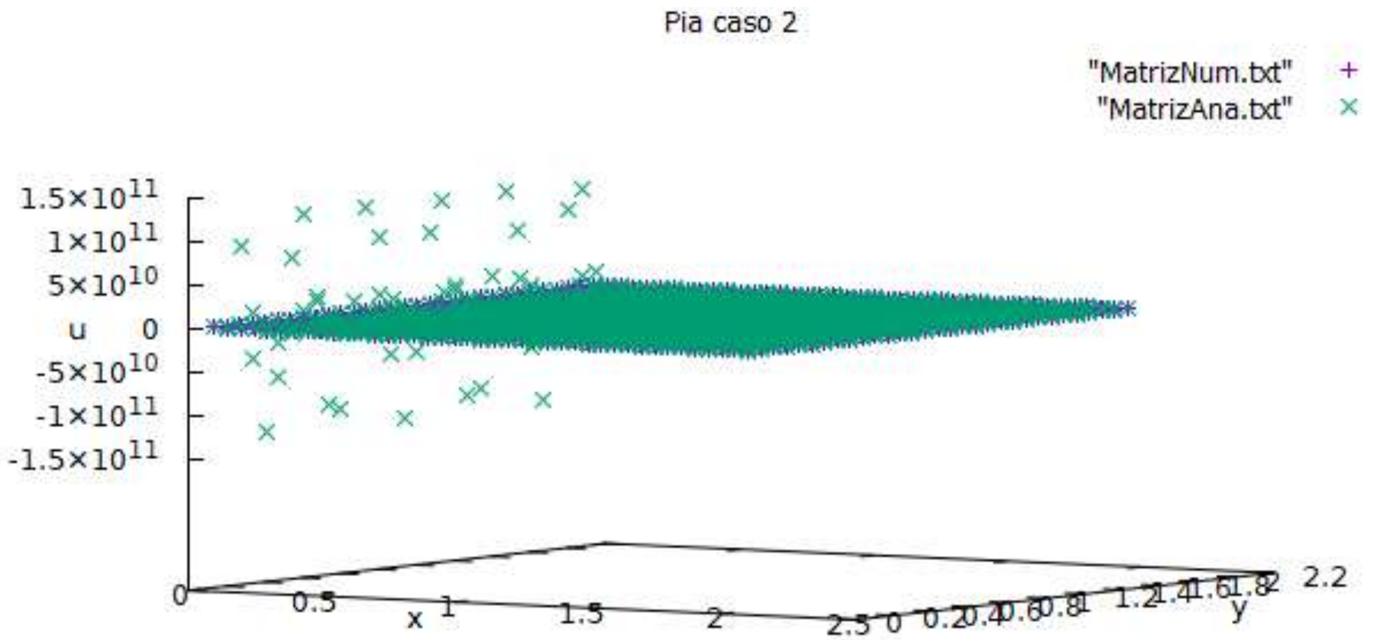
Con: Nx = 20, Ny = 40, n = 0,1,…,5 , h = 0.8, l = 0.5, lamda = 0.8, k = 5000



Con: Nx = 40, Ny = 30, n = 0,1,…,5 , h = l = 3, lamda = 0.8, k = 5000



Con: Nx = 40, Ny = 30, n = 0,1,…,5 , h = l = 2, lamda = 0.8, k = 5000



Conclusiones:

-Las simplificaciones en las ecuaciones de Navier-Stokes, entre más sean, los resultados se distanciarán más de la realidad del fenómeno. Es decir, el flujo de Poiseuille, es un flujo muy idealizado. Por lo que estos resultados están sujetos a dichas consideraciones.

-La solución analítica y numérica, en caso 1, tiene muy poco error y esto se puede reducir aumentando el número de subintervalos.

-La solución analítica y numérica, en caso 2, tiene mucho error. Y suponemos que es porque la solución analítica está truncada a una cierta n. Es decir, en tanto se tengan más términos en la sumatoria de la solución analítica, ambas soluciones tendrán un menor rango de error.

-Además, para el caso 2, el método aquí desarrollado es insuficiente, pues el perfil de velocidades aumenta exponencialmente. Por lo tanto, el método funcionará para medidas pequeñas de l y h menores a 1.5.

-Por último, concluimos que el método de diferencias finitas es un buen método para empezar, no obstante, hay mejores métodos para resolver Navier-Stokes en diferentes situaciones.

Referencias:

-White M. Frank, Mecánica de los Fluidos.

-T.Cebeci, J.RShao, F. Kafyeke, E. Laurendeau. Computational Fluid Dynamics for Engineers.

-Riley Norman y Drazin Philip.(2006). The Navier–Stokes equations: a classification of flows and exact solutions.

- Steven E. Koonin y Dawn C. Meredith. Computational physics Fortran versión.

Código sobre factorización LU:
- https://ww2.odu.edu/~agodunov/computing/programs/book2/Ch06/Inverse.f90