SIMULACIÓN DEL FLUJO BIDIMENSIONAL ALREDEDOR DE UNA VIGA MEDIANTE EL MÉTODO DE VORTICIDAD-FUNCIÓN DE CORRIENTE

Julián Aros-Laura Oliveros-Andrés Gómez
Programa Académico de Física
Universidad Distrital Francisco José de Caldas

PROBLEMA A SIMULAR

In an inertial frame for several days,
a physicist begins to long for absolute rest

Problema a Simular

Objetivo

Estudiar el comportamiento de un fluido viscoso e incompresible fluyendo alrededor de una viga sumergida en régimen estacionario.

Hipótesis

- Incompresibilidad: $\nabla \cdot \vec{v} = 0$
- Régimen estacionario: $\partial / \partial t = 0$
- Fluido viscoso con viscosidad cinemática ν
- Flujo bidimensional (x, y)

Problema a Simular

Ecuaciones de
Navier-Stokes 2D

$$\nabla \cdot \vec{v} = 0 \quad \text{(Continuidad)}$$

$$(\vec{v} \cdot \nabla)\vec{v} = -\frac{1}{\rho}\nabla P + \nu \nabla^2 \vec{v} \quad \text{(Momentum)}$$

Problema

Sistema acoplado no lineal con presión como incógnita adicional.

Problema a Simular

Reformulación: Función de Corriente y Vorticidad

Función de Corriente $u(x, y)$

$$v_x = \frac{\partial u}{\partial y},$$

$$v_y = -\frac{\partial u}{\partial x}$$

Vorticidad $w(x, y)$

$$w = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}$$

Problema a Simular

Sistema Reformulado

$$w = -\nabla^2 u$$

$$\nu \nabla^2 w = \frac{\partial u}{\partial y}\frac{\partial w}{\partial x} - \frac{\partial u}{\partial x}\frac{\partial w}{\partial y}$$

Ventaja

Se elimina la presión y se reduce a 2 ecuaciones en 2 incógnitas.

Problema a Simular

Discretización Numérica

Malla uniforme: $x_i = ih,\ y_j = jh$

Poisson:

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} + h^2 w_{i,j})$$

Vorticidad:

$$w_{i,j} = \frac{1}{4}(w_{i+1,j} + w_{i-1,j} + w_{i,j+1} + w_{i,j-1}) - \frac{R}{16}[(u_{i,j+1} - u_{i,j-1})(w_{i+1,j} - w_{i-1,j}) - (u_{i+1,j} - u_{i-1,j})(w_{i,j+1} - w_{i,j-1})]$$

Problema a Simular

Número de Reynolds y Algoritmo SOR

Número de Reynolds de malla:

$$R = \frac{V_0 h}{\nu}$$

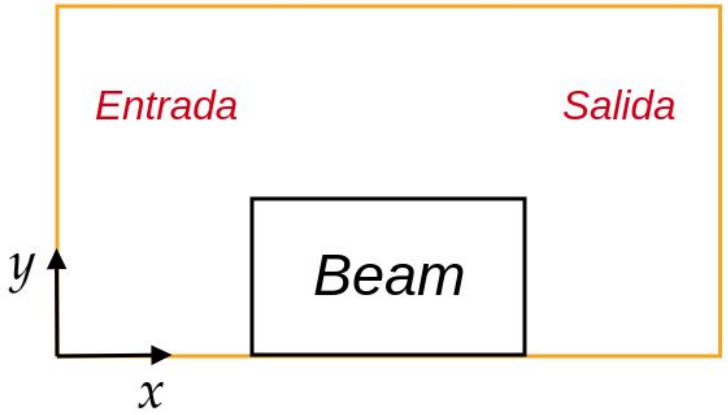
Actualización SOR:

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} + \omega(\mathrm{RHS}_{u_{i,j}} - u_{i,j}^{(n)})$$

- $0 < \omega < 2$
- Acelera la convergencia

Problema a Simular

Condiciones de Frontera



- **Entrada:** flujo uniforme $u_1, j = u_0, j,\ \omega = 0$
- **Salida:** derivada nula en x $u_{Nx,j} = u_{Nx-1,j}$
- **Superficie libre:** $u_{i,Ny} = u_{i,Ny-1} + V_0 h,\ \omega = 0$
- **Simetría (línea central):** $u = 0,\ \omega = 0$
- **En la viga:** $u = 0,\ \omega = -2(u \text{ vecino})/h^2$

PARTE SERIAL

Parte Serial

Parámetros del Problema

- Velocidad característica: $V_0 = 1{,}0$
- Espaciado de malla: $h = 1{,}0$
- Tamaño de malla: $N_x = 160, N_y = 30$
- Tamaño de la viga: $L = 8h, H = 8h$

Rangos del número de Reynolds:

- $Re < 5$: flujo adherido
- $5 < Re < 40$: vórtices estacionarios
- $Re > 40$: calle de von Kármán

Parte Serial

Ejecución

```
$ make run
Ejecutando el binario ./beam...
bash -c "time ./beam"
=== Solver de Navier-Stokes Multi-Reynolds (ESQUINAS CORREGIDAS) ===
Parámetros del dominio:
  Malla: 160 x 30
  Viga: posición x=[10,18], altura=8
  V0 = 1, h = 1
  Tolerancia = 1e-08
Directorio 'Datos' creado exitosamente.

============================================================
INICIANDO SIMULACIÓN PARA Re = 0.5
============================================================
Configuración para Re = 0.5:
  nu = 2
  omega = 0.1
Número de Reynolds de malla calculado R = 0.5
Objetivo: Re = 0.5
```

Parte Serial

Tiempo de Ejecución

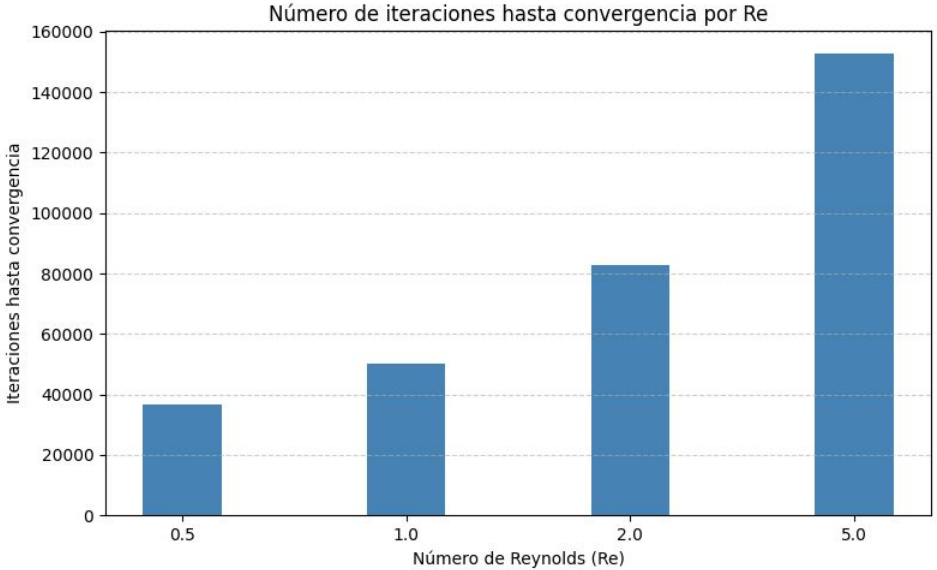
```
real    2m52.849s
user    2m24.630s
sys     0m1.945s
```

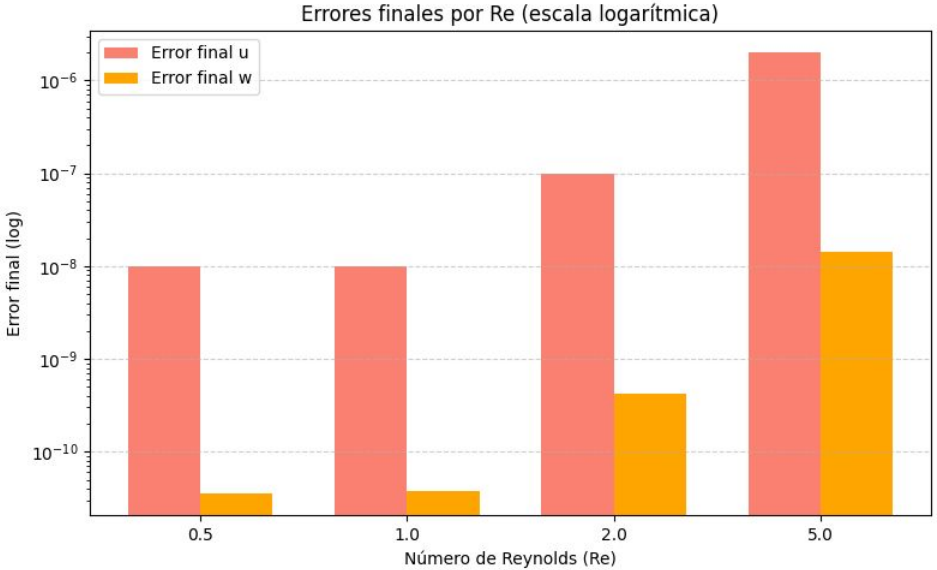
Errores Finales Hasta alcanzar convergencia

Re	Iteraciones	Error u final	Error ω final
0.50	36,533	9.999640×10^{-9}	3.567400×10^{-11}
1.00	49,990	9.998630×10^{-9}	3.787940×10^{-11}
2.00	82,967	9.999050×10^{-8}	4.169170×10^{-10}
5.00	152,746	2.000000×10^{-6}	1.418880×10^{-8}

Parte Serial

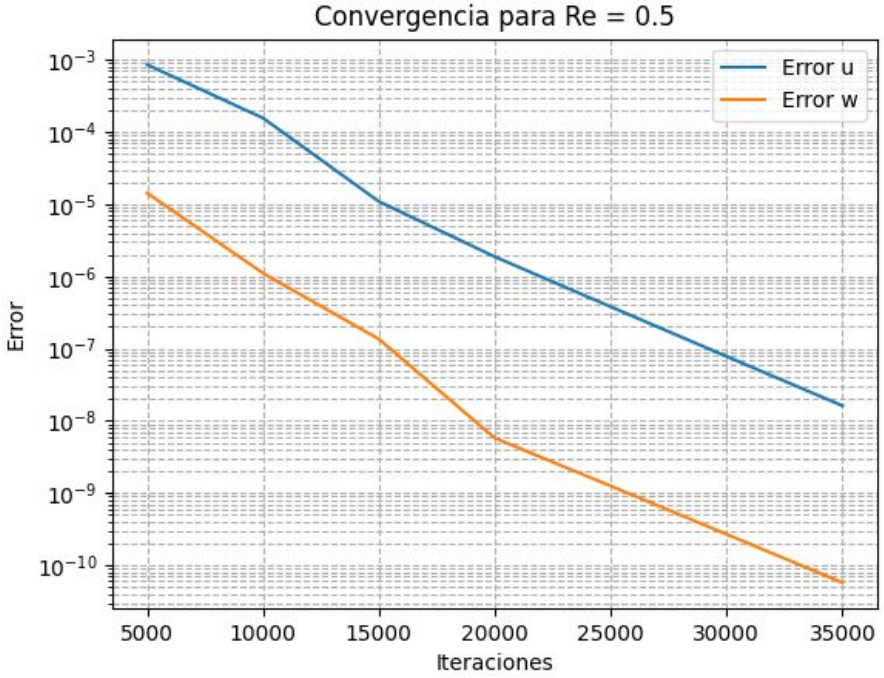
Errores Finales Hasta alcanzar convergencia

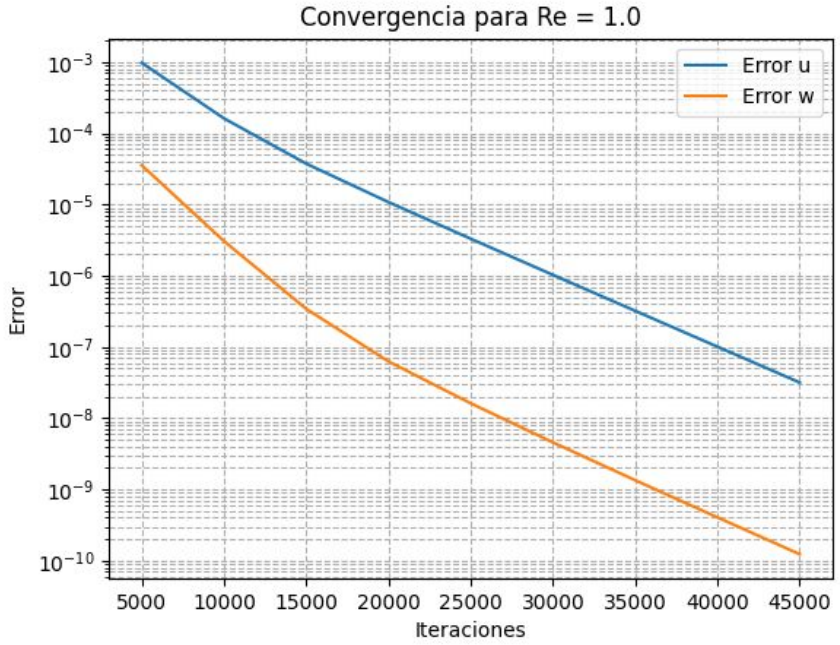




Parte Serial

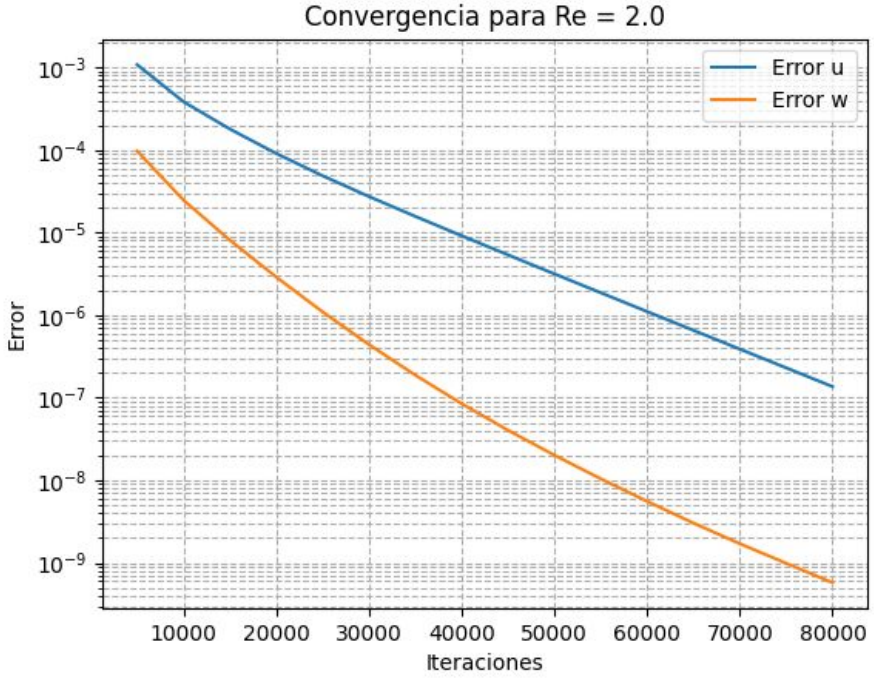
Tendencia de los Errores Hasta alcanzar convergencia

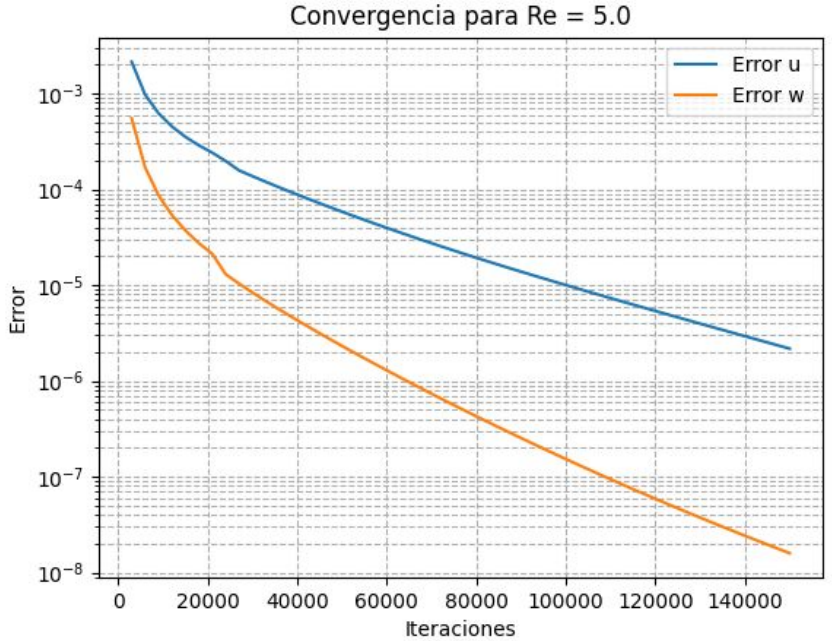




Parte Serial

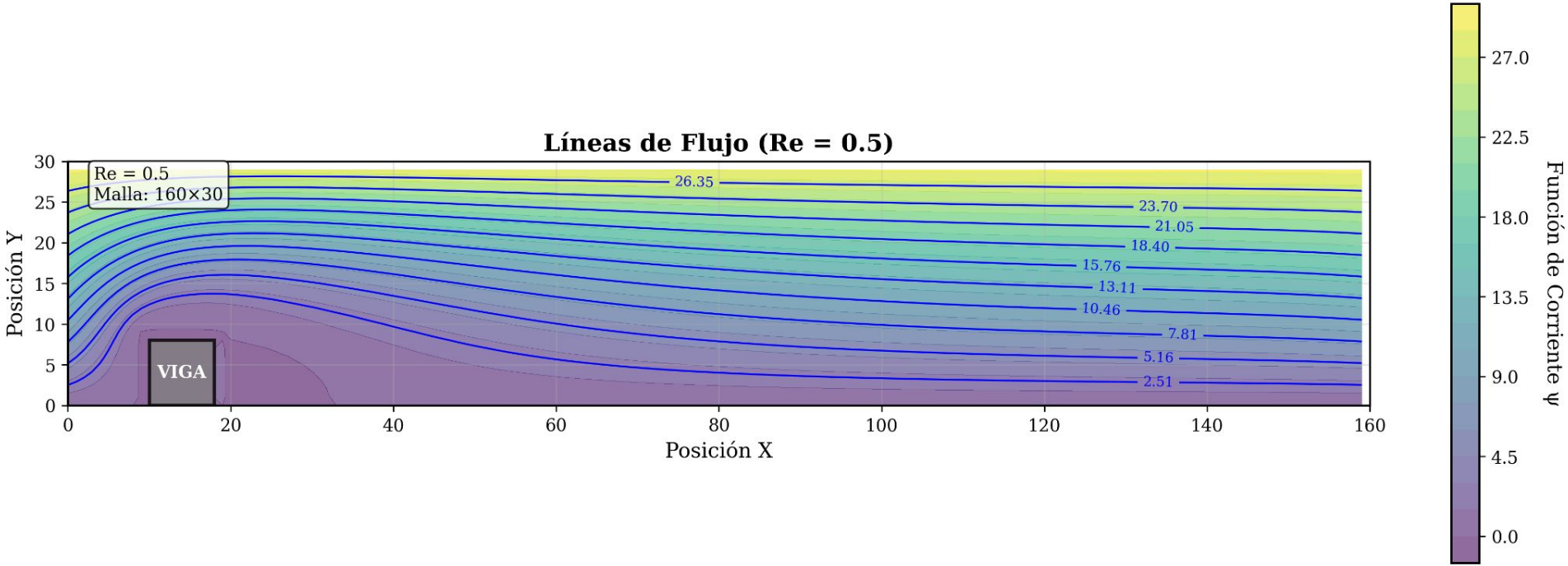
Tendencia de los Errores Hasta alcanzar convergencia





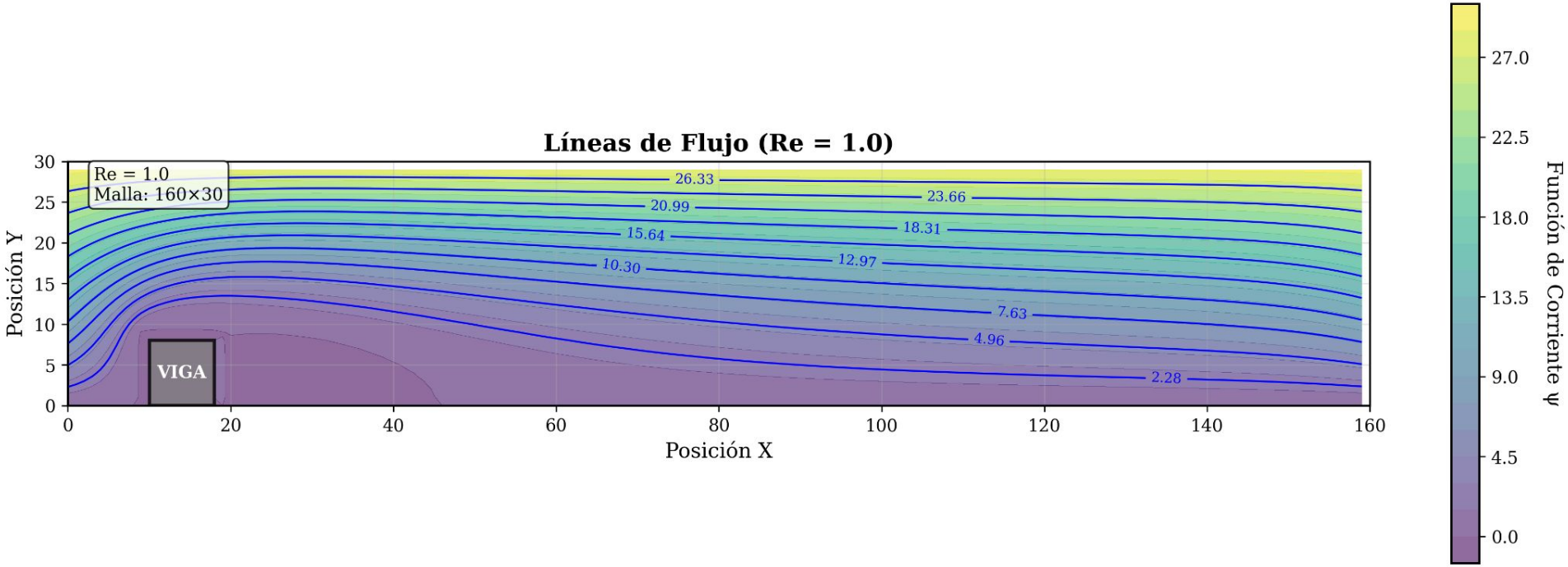
Parte Serial

Resultados



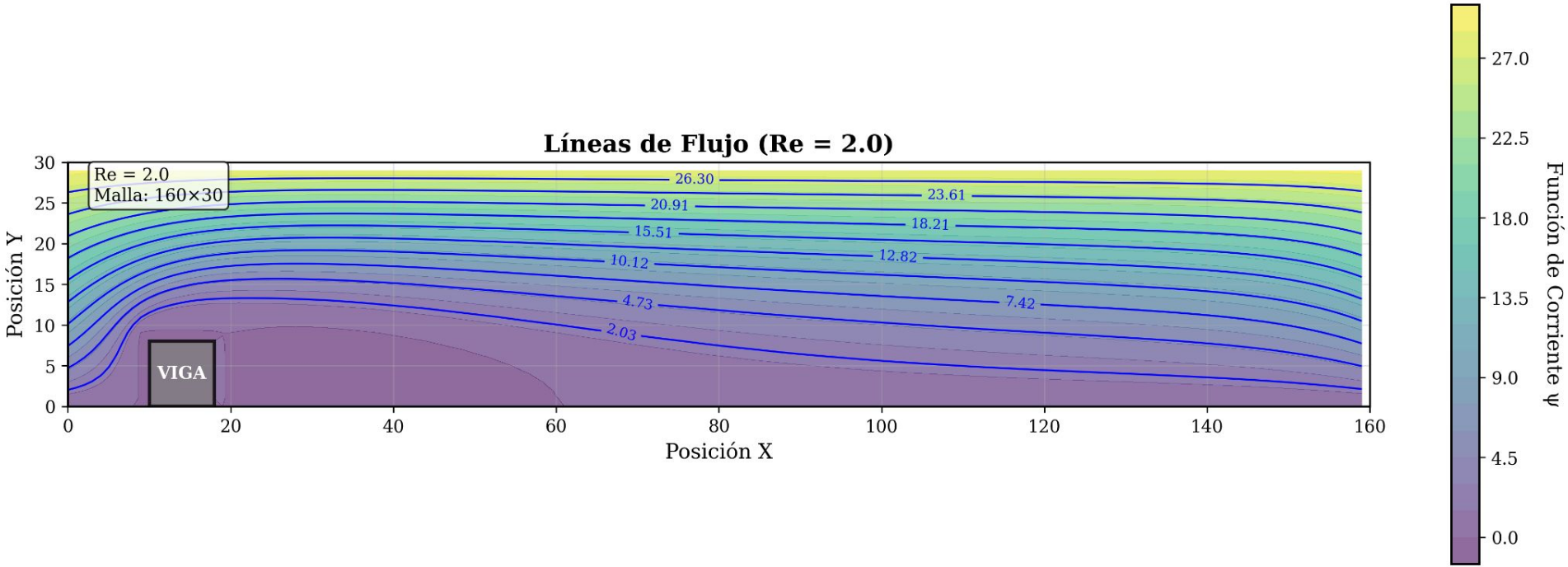
Parte Serial

Resultados



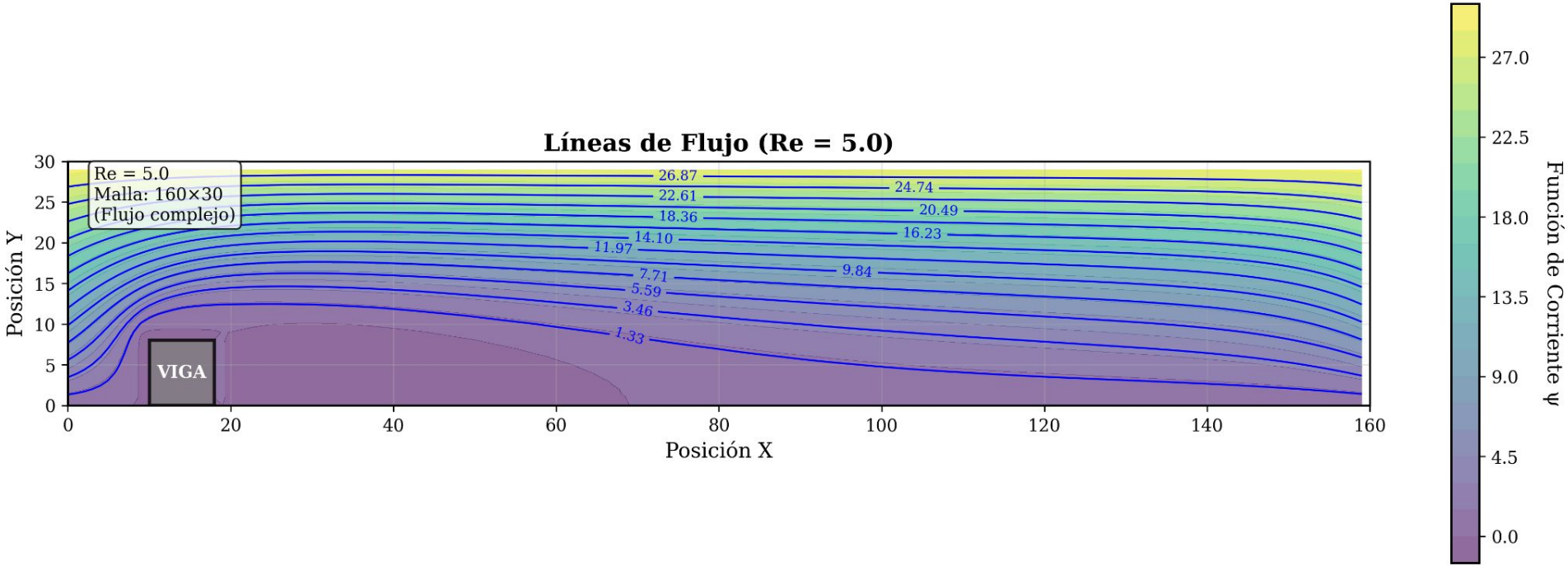
Parte Serial

Resultados



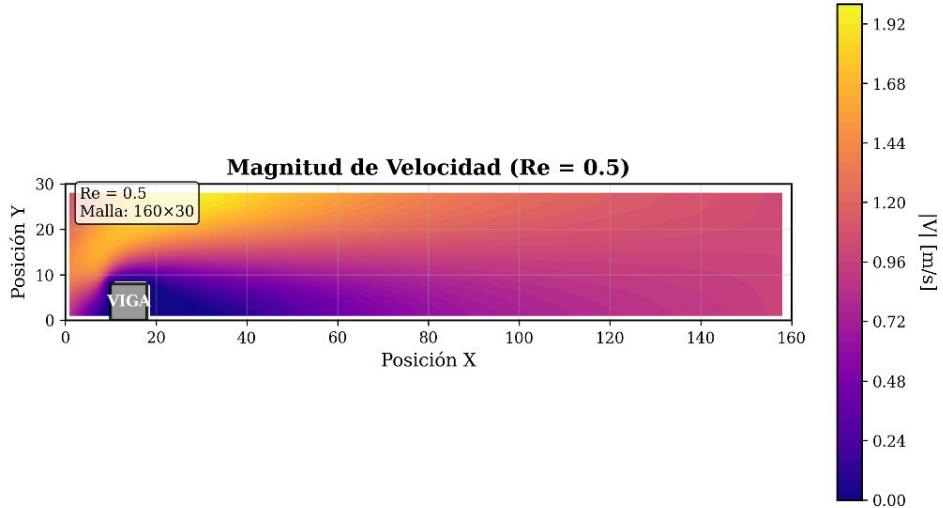
Parte Serial

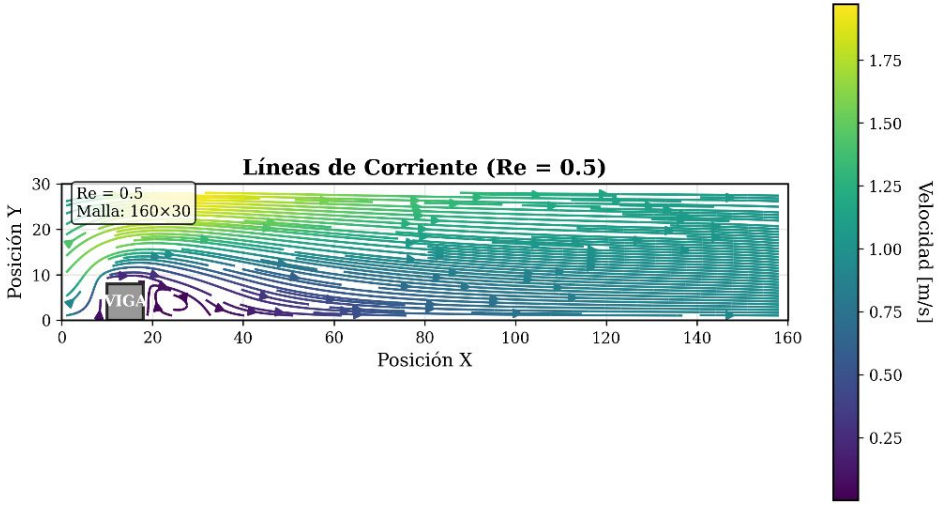
Resultados



Parte Serial

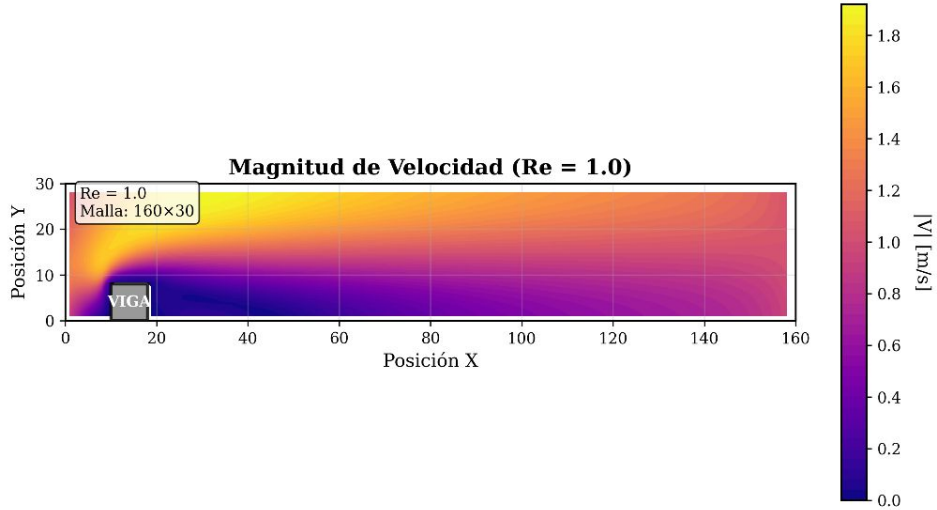
Resultados

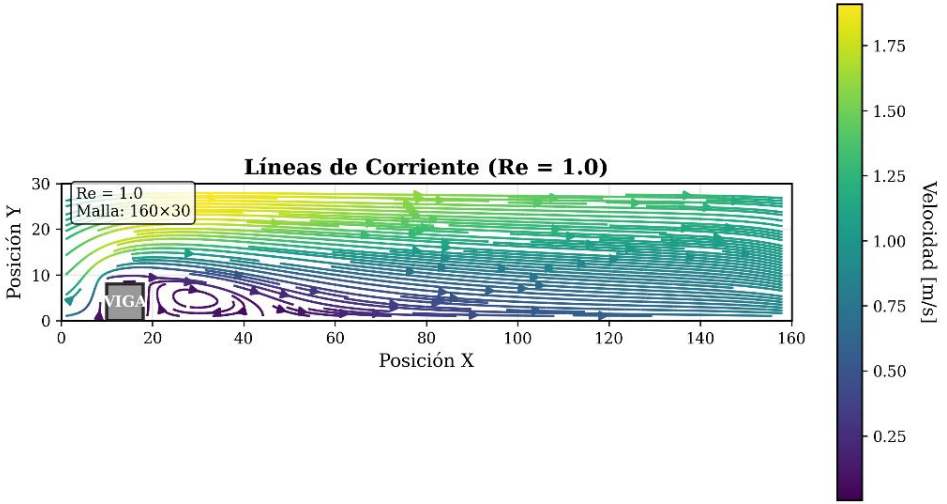




Parte Serial

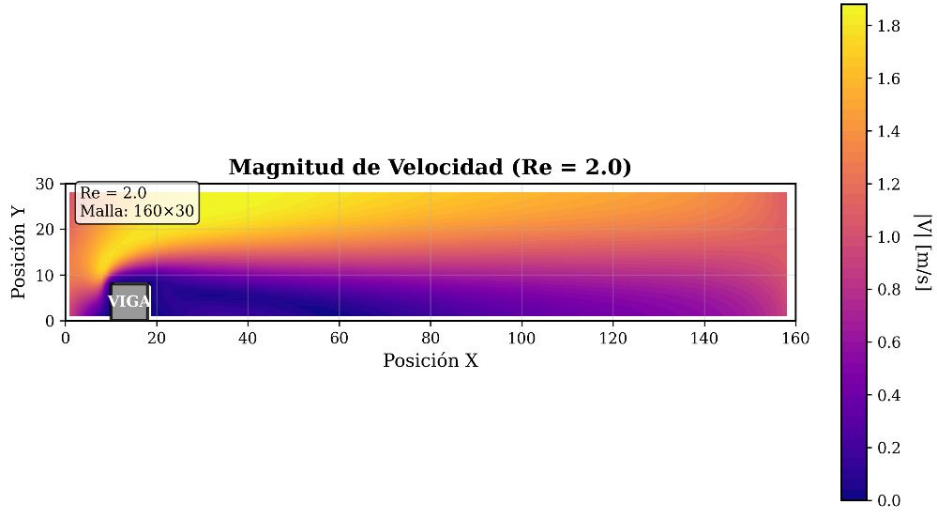
Resultados

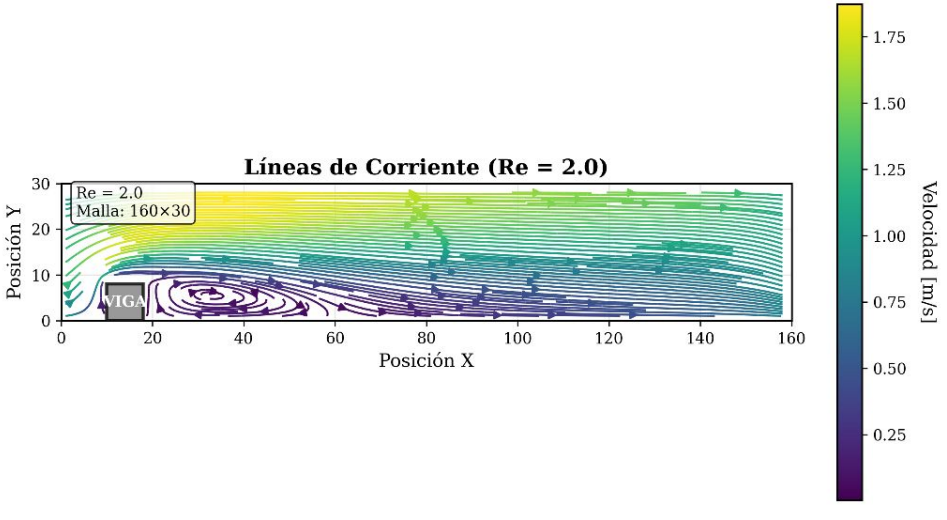




Parte Serial

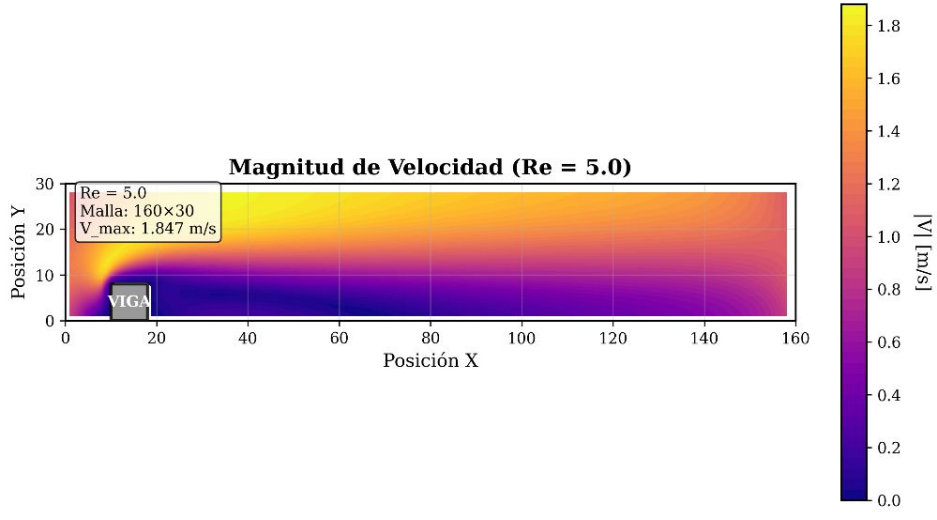
Resultados

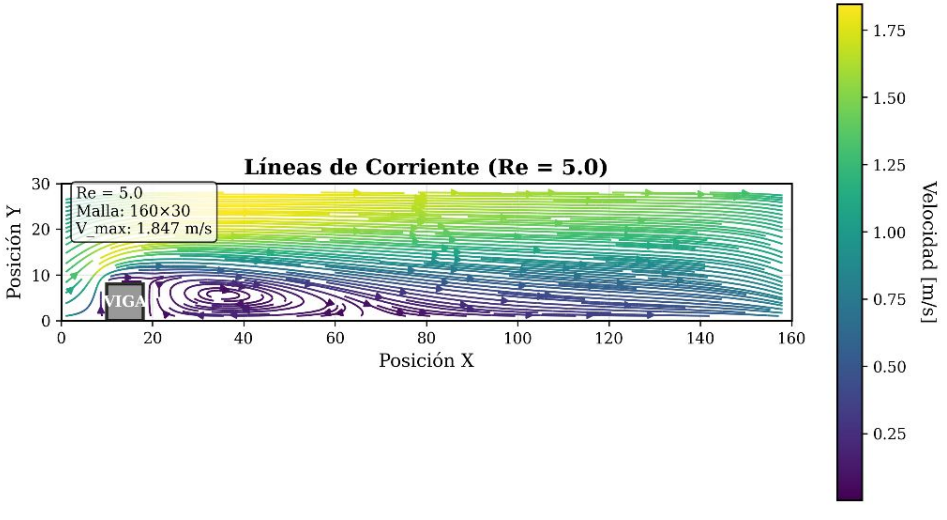




Parte Serial

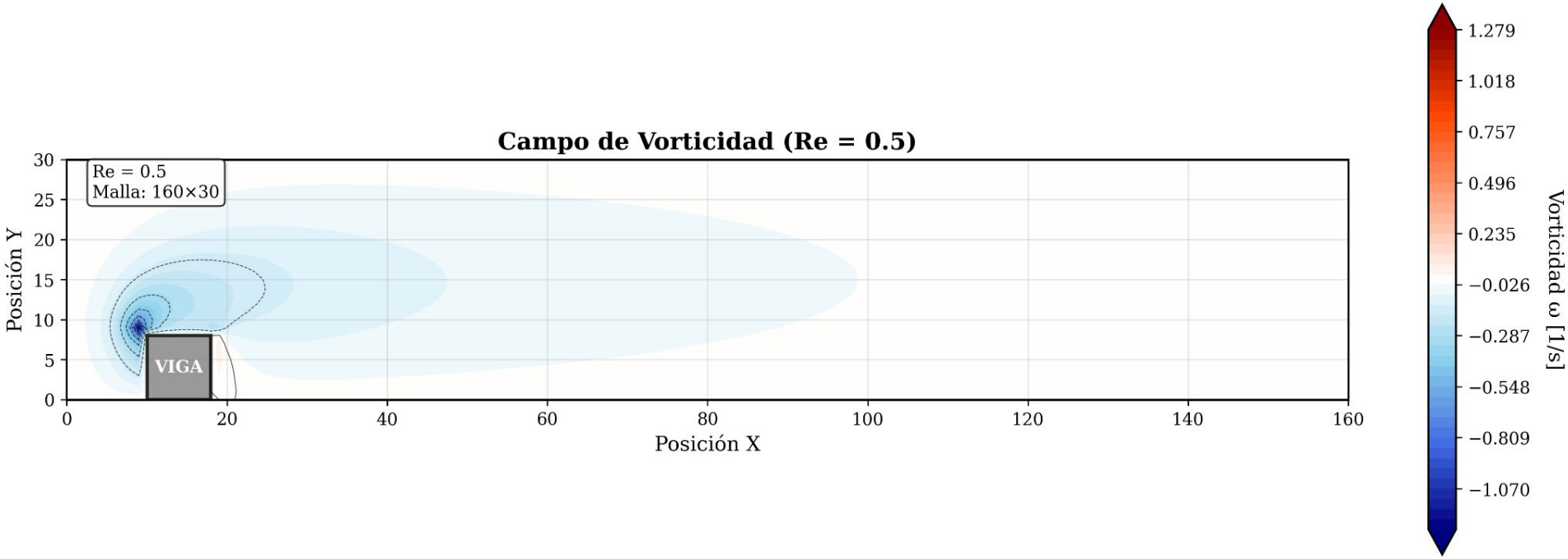
Resultados





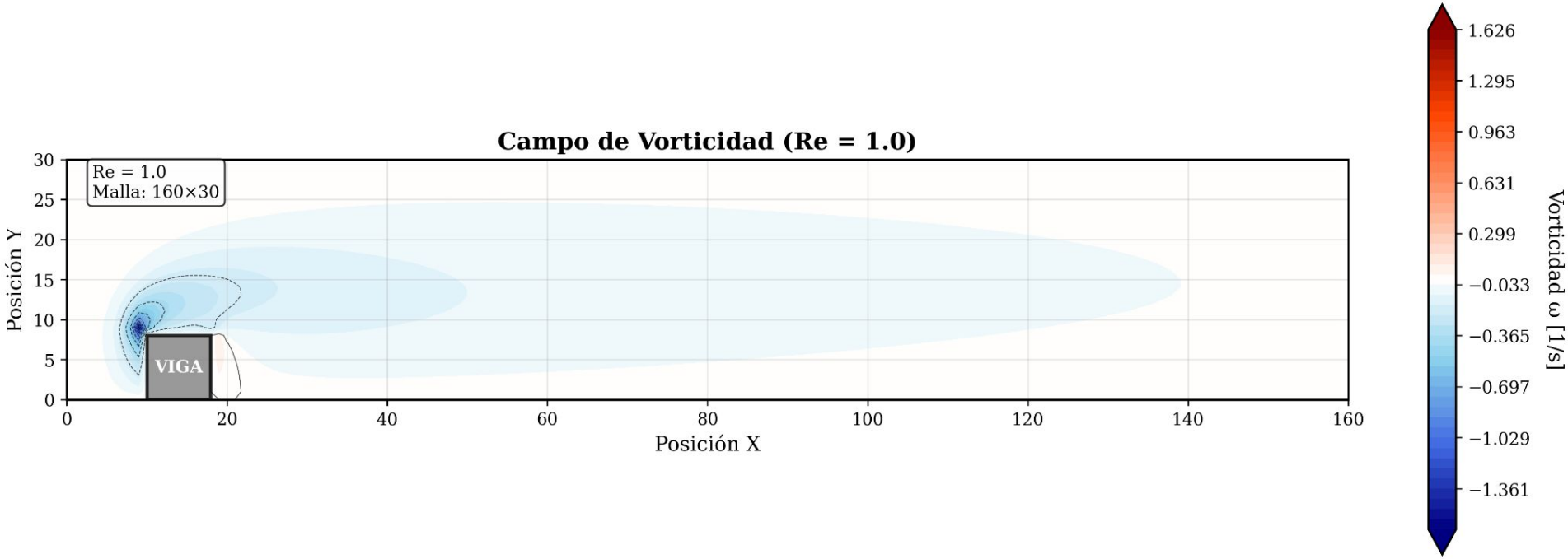
Parte Serial

Resultados



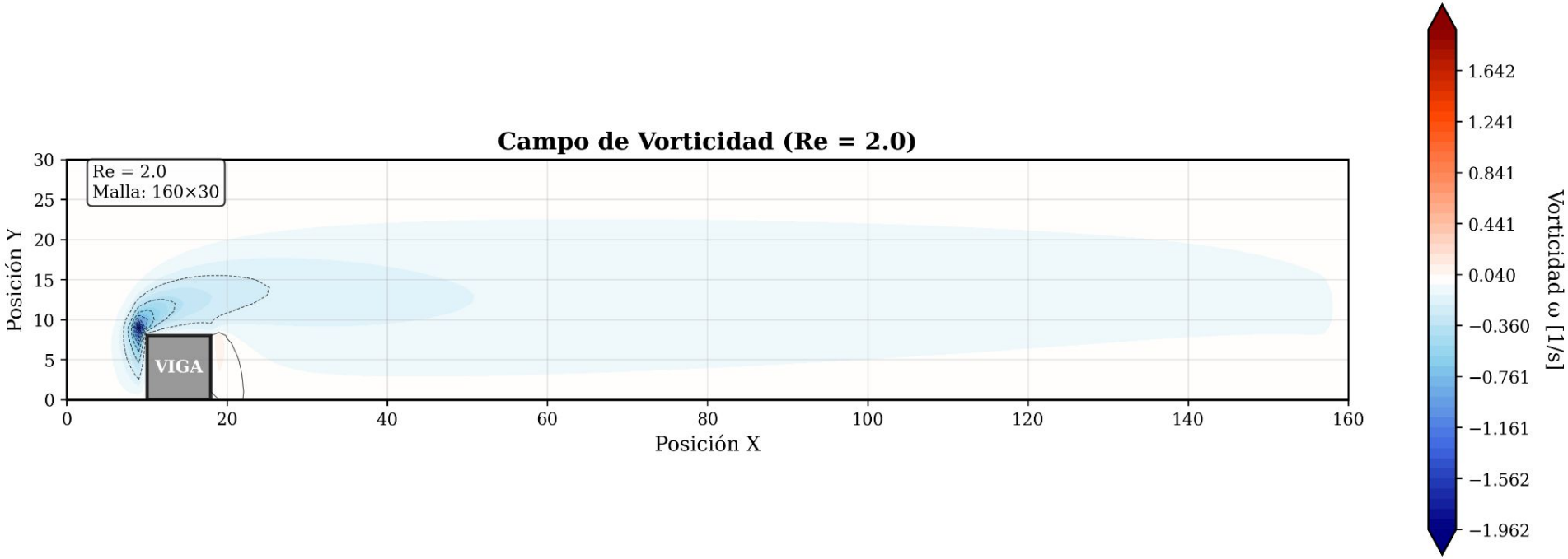
Parte Serial

Resultados



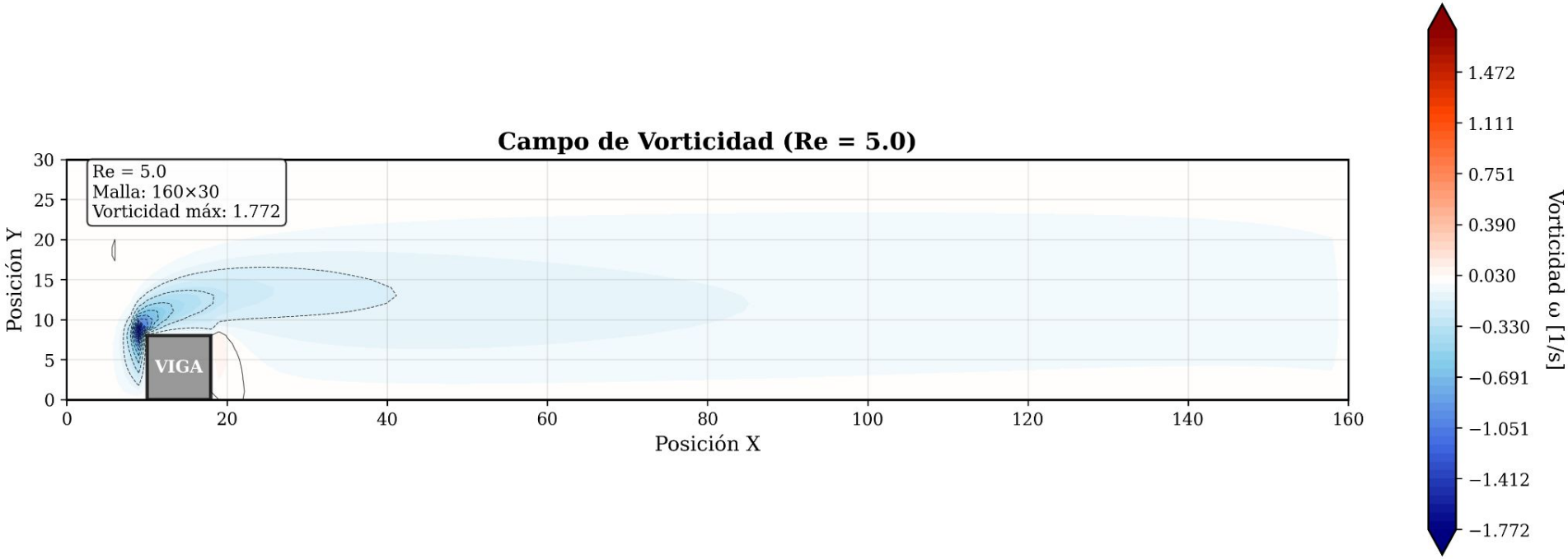
Parte Serial

Resultados



Parte Serial

Resultados



PARTE PARALELIZADA OpenMP

In an inertial frame for several days,
a physicist begins to long for absolute rest

Parte Paralelizada

Parámetros del Problema

- Velocidad característica: $V_0 = 1{,}0$
- Espaciado de malla: $h = 1{,}0$
- Tamaño de malla: $N_x = 160, N_y = 30$
- Tamaño de la viga: $L = 8h, H = 8h$

Rangos del número de Reynolds:

- $Re < 5$: flujo adherido
- $5 < Re < 40$: vórtices estacionarios
- $Re > 40$: calle de von Kármán

Códigos realizados

1. beam-parallel-for.cpp
2. beam-collapse.cpp
3. beam-static.cpp
4. beam-collapse.cpp
5. beam-parallel-for-NBS.cpp

graficador.py

Makefile

- make build
- make all (run,plot)
- make clean

```
g++ -fopenmp -O2 -o ...
```

Parte Paralelizada

Características de la máquina

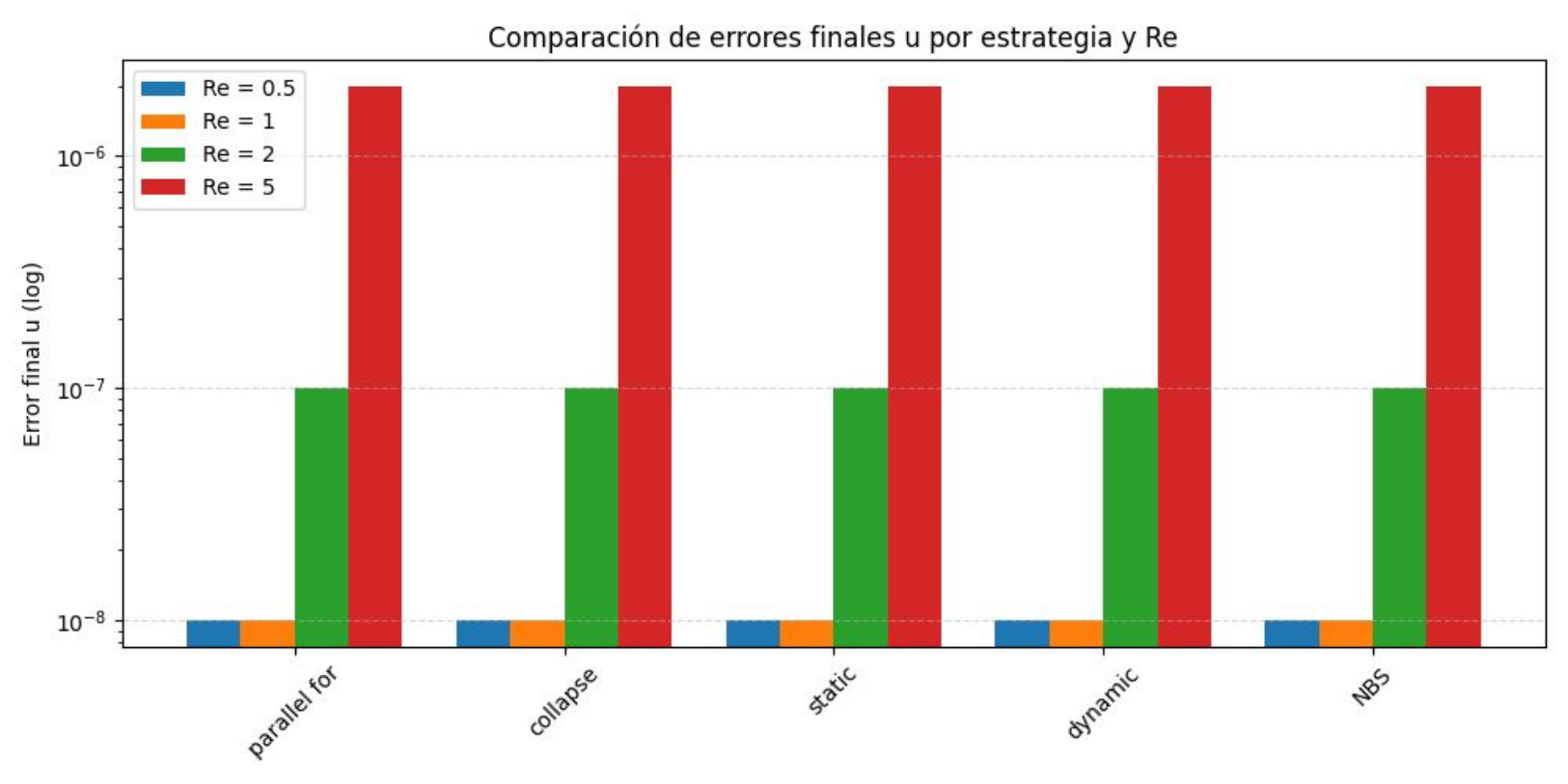
```
(base) ubuntu@ip-172-31-0-158:~$ lscpu
Architecture:                    x86_64
  CPU op-mode(s):                32-bit, 64-bit
  Address sizes:                 46 bits physical, 48 bits virtual
  Byte Order:                    Little Endian
CPU(s):                          32
  On-line CPU(s) list:           0-31
Vendor ID:                       GenuineIntel
  Model name:                    Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
    CPU family:                  6
    Model:                       106
    Thread(s) per core:          2
    Core(s) per socket:          16
    Socket(s):                   1
    Stepping:                    6
    BogoMIPS:                    5799.96
```

```
Caches (sum of all):
  L1d:                           768 KiB (16 instances)
  L1i:                           512 KiB (16 instances)
  L2:                            20 MiB (16 instances)
  L3:                            54 MiB (1 instance)
```

Versión	Directiva usada	Tiempo (s)	Iteraciones Re = 0,5	Iteraciones Re = 1	Iteraciones Re = 2	Iteraciones Re = 5
Secuencial		172,849	36,533	49,990	82,967	152,746
Paralelo básico	#pragma omp parallel reduction(max:delta)	8,424	36,593	50,063	83,057	152,827
Colapsado de bucles	#pragma omp parallel for collapse(2) reduction(max:delta)	8,635	36,593	50,063	83,057	152,828
Control explícito + schedule	#pragma omp parallel #pragma omp for schedule(dynamic) reduction(max:delta)	11,559	36,593	50,063	83,057	152,828
Control explícito + schedule	#pragma omp for schedule(static) reduction(max:delta)	8,995	36,593	50,063	83,057	152,827
Control de sincronización	#pragma omp parallel #pragma omp for nowait #pragma omp single #pragma omp barrier	8,611	36,593	50,063	83,057	152,827

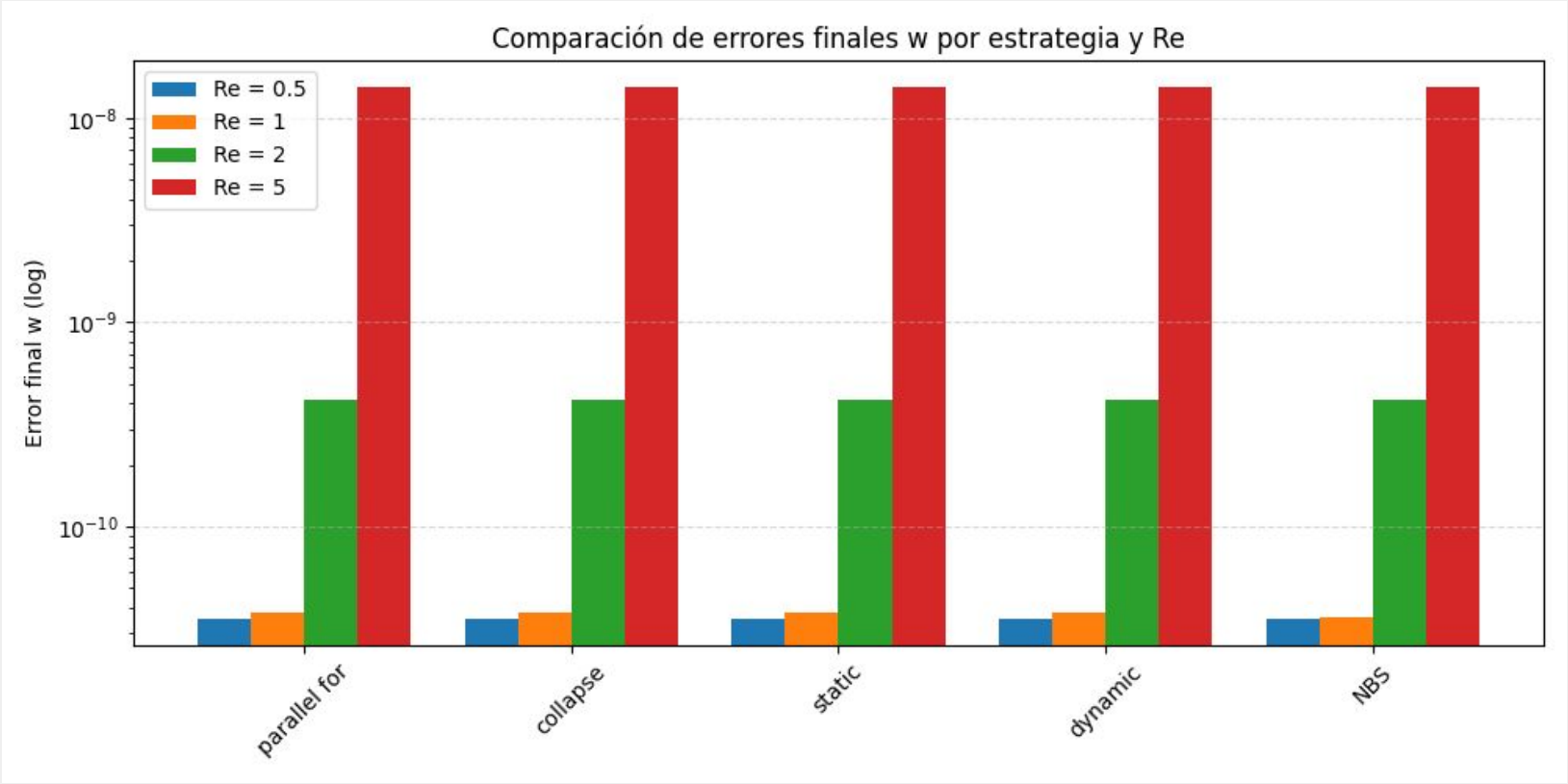
Parte Paralela

Errores Finales Hasta alcanzar convergencia



Parte Paralela

Errores Finales Hasta
alcanzar convergencia



Versión	Directiva usada	Tiempo (s)	Observaciones
Secuencial		172,849	Mayor tiempo de ejecución
Paralelo básico	#pragma omp parallel reduction(max:delta)	8,424	Versión más rápida y simple
Colapsado de bucles	#pragma omp parallel for collapse(2) reduction(max:delta)	8,635	Tercera versión más rápida y también es una implementación simple
Control explícito + schedule	#pragma omp parallel #pragma omp for schedule(dynamic) reduction(max:delta)	11,559	Segundo versión más lenta, posible overhead por dynamic
Control explícito + schedule	#pragma omp for schedule(static) reduction(max:delta)	8,995	Cuarta versión más rápida
Control de sincronización	#pragma omp parallel #pragma omp for nowait #pragma omp single #pragma omp barrier	8,611	Segunda versión más eficiente, más compleja al momento de ubicar los pragmas en el código

Conclusiones

- La implementación realizada es un muy buen primer acercamiento, porque usa la formulación de función de corriente-vorticidad, adecuada para flujos incompresibles 2D. Ajusta correctamente la viscosidad para representar el número de Reynolds, aplica condiciones de frontera físicas y modela un obstáculo rectangular (viga). Utiliza SOR, un método numérico clásico y estable en CFD (Física de Fluidos Computacional)
- Además, se hacen gráficas como la función de corriente, que representa líneas de flujo (trayectorias tangentes a la velocidad), donde su gradiente da la dirección del flujo. Se presenta el campo de velocidades (u,v) que muestra el movimiento real del fluido en cada punto y un campo de vorticidad (indica zonas de rotación o remolinos, siendo clave para analizar turbulencias o recirculaciones).
- Con las versiones de código paralelizadas es posible mejorar el código 20 veces aproximadamente, en el caso de la versión más efectiva (#parallel for). Aún así todas las versiones muestran mejoras donde se puede apreciar las ventajas de paralelizar.

Gracias…

In an inertial frame for several days,
a physicist begins to long for absolute rest