

# **FLOW DYNAMICS AND SCAVENGING PHENOMENA IN EXHAUST MANIFOLDS**

Antoine BERTRAND

November 20<sup>th</sup>, 2024

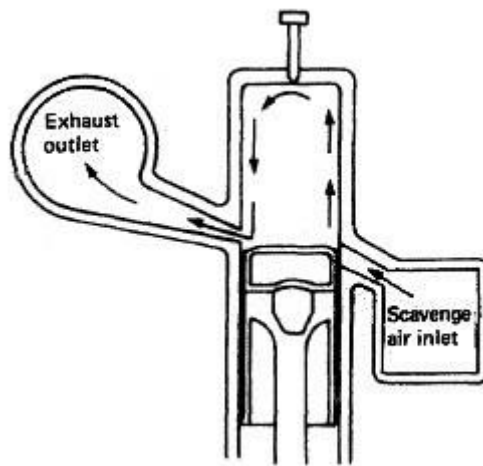
Walter Scott, Jr. College of Engineering

Colorado State University, Fort Collins, CO 80521

## I. INTRODUCTION

The exhaust manifold might seem unremarkable at first—it's just a component to expel exhaust gases. However, knowing that a well-designed exhaust manifold can add up to 20 horsepower to an engine helps us understand why sports car manufacturers strive to optimize this part. Two main concepts make an exhaust manifold efficient: **back pressure** and **scavenging**.

- Back pressure in an exhaust system refers to the resistance encountered by exhaust gases as they exit the engine through the exhaust manifold, pipes, catalytic converter, and muffler. It occurs due to restrictions, obstructions, or turbulence caused by flow transitions in the exhaust path. High back pressure makes it harder for the engine to expel exhaust gases, reducing overall power output.
- Scavenging is the process of clearing exhaust gases from the engine's combustion chamber and replacing them with a fresh air-fuel mixture. After combustion, a low-pressure shock wave travels back into the cylinder. With precise timing (just before the exhaust valve closes), this creates a vacuum in the combustion chamber. This vacuum helps pull the fresh air-fuel mixture from the intake. For naturally aspirated engines, where the intake operates at atmospheric pressure, achieving lower-than-atmospheric pressure in the combustion chamber allows more air-fuel mixture to enter, improving combustion efficiency [1].



*Figure 1 Principle of scavenging sucking the air in when the exhaust valve is still open [2]*

When reviewing available studies online, much of the research focuses on the steady-state behavior of exhaust systems, which is highly useful for thermal analysis. Similarly, the back pressure effect is already well-documented, as it primarily involves the flow of hot air through a pipe. For this reason, I chose to focus on the scavenging effect. To study it, I utilized the transient simulation capabilities of ANSYS Fluent. In the first part of the study, I will validate the model by comparing the steady-state simulation results with existing documented data. In the second part, I will model the scavenging effect using a transient simulation to address the following question: **How does the scavenging effect manifest in an exhaust manifold?**

## II. METHODS

To compare my results with existing documented data and keep the analysis straightforward, I made the following major assumptions :

- The manifold is entirely made of steel
- The exhaust gas is modeled as air (70% N<sub>2</sub>, representing both exhaust gas and air composition)
- The exhaust gases have a uniform speed and direction across the inlet area
- The back pressure at the outlet is 0 Pa (the exhaust manifold outlet leads directly outside, with no catalytic converter or muffler, and a straight pipe design)
- Thermal interaction with the surroundings is limited to convection, with no radiation effects considered (e.g., ignoring scenarios like exhaust glow)

These assumptions are common for this type of modeling.

For geometry, I will model only one side of a V8 engine because the firing order is straightforward (1 → 2 → 3 → 4). I will use an exhaust system design known as 4-1, where all the runners have equal lengths and converge at a single point. This design was chosen because it provides the strongest scavenging effect. For the dimensions, I will use those documented in existing studies [3] to ensure that my results can be accurately compared.

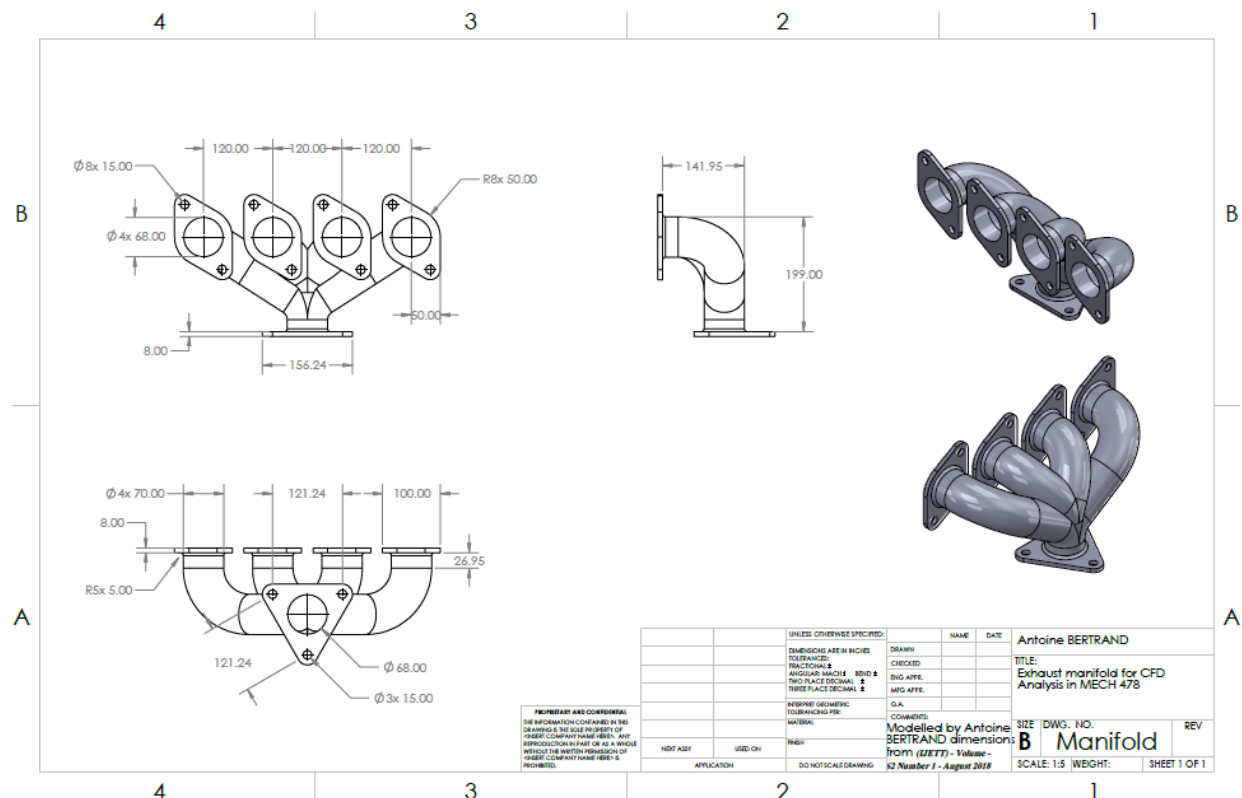
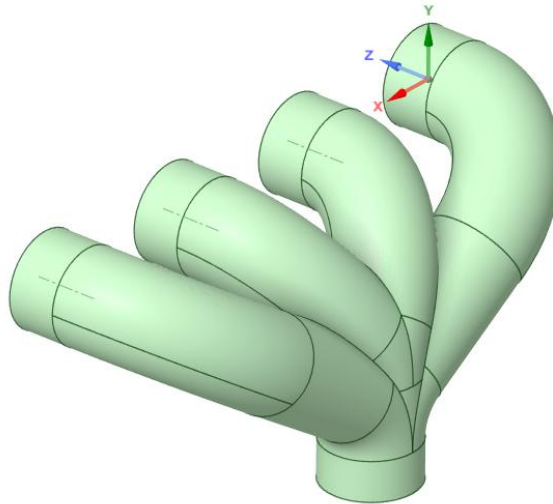


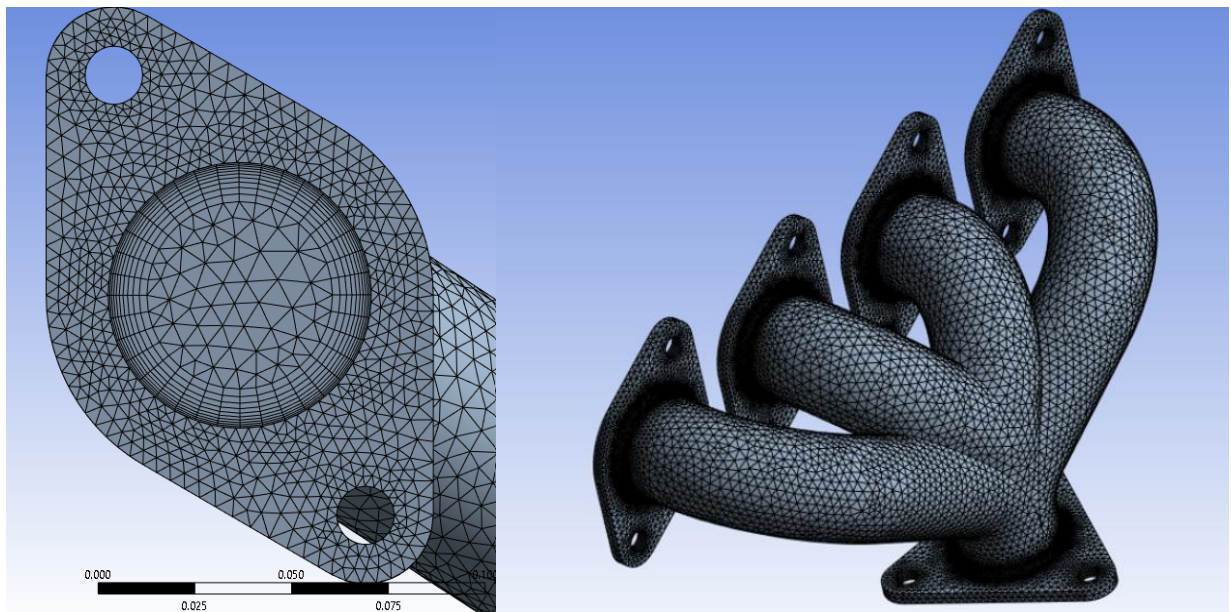
Figure 2 Drawing of the exhaust manifold I 3D design in Solidworks for this simulation

After designing the exhaust manifold as a solid body in SolidWorks (for a 3D simulation), I created the fluid domain in SpaceClaim. This was done by subtracting the solid manifold from a larger enclosing volume to define the space where the fluid flows.



*Figure 3 Fluid body created with SpaceClaim*

The meshing process is crucial for this simulation, as it involves calculating the thermal transfer between the hot air and the steel tubing. To accurately simulate the boundary interactions between the fluid and the metal piping, the mesh resolution needs to be higher near the pipe walls. For this purpose, I used the inflate tool to refine the mesh within the fluid body near the metal surfaces.



*Figure 4 Final mesh of the solid and fluid body*

The element size for both the fluid and solid bodies is set to 0.6 mm, which seems reasonable given the pipe diameter of 68 mm, similar to the setup used in this study [4]. For the inflate tool, I used the  $y^+$  method to determine the appropriate parameters, ensuring accurate boundary layer resolution :

$$Re = \frac{U \cdot D}{\nu}$$

**Density :** For air at 700 K,  $\rho = 0.52 \text{ kg/m}^3$

**Dynamic viscosity :**  $\mu = 3.72 \times 10^{-5} \text{ Pa} \cdot \text{s}$

**Cinematic viscosity :**  $\nu = \mu/\rho, \nu = 7.15 \times 10^{-5} \text{ m}^2/\text{s}$

**Average speed :**  $U = 20 \text{ m/s}$

**Diameter :**  $D = 0.07 \text{ m}$

$$Re = 20000$$

The  $\Delta y$  is the size of the first cell and this is the approximation based on the Reynolds number.

$$\Delta y = \frac{1}{\sqrt{Re}} = 0.001 \text{ m}$$

I used a growth rate of 1.2 and a target  $y^+ = 10$  resulting in a total inflation layer thickness of 0.02596 m. After conducting multiple simulations, this configuration provided the best resolution without causing Fluent to crash. This limitation made it challenging to perform a proper mesh independence study, as some combinations of mesh sizing and inflation parameters worked while others did not. Ultimately, I proceeded with the setup that proved stable and functional.

**For the steady simulation :** I used the k-omega SST model to simulate the turbulent flow, as it offers a good balance between accuracy and computational efficiency [5]. Additionally, I enabled the energy equation since temperature plays a critical role in the simulation. For the four-cylinder inlets, appropriate boundary conditions were applied to reflect the flow behavior [6] :

- Velocity of 20 m/s
- Turbulent intensity 5 % and a viscosity ratio of 0.06
- Temperature of 800 K

For the outlet, I applied a pressure outlet boundary condition with a value of 0 Pa (no back pressure). For the walls, I used convection as the thermal boundary condition, with a heat transfer coefficient of  $20 \text{ W/m}^2\text{K}$ .

To assess the simulation's convergence, I monitored the mass flow rate balance (+inlet -outlet) and the outlet temperature. As shown below, the simulation converges within 100 iterations, although the residuals remain relatively high. This is a compromise to ensure quicker

simulation times. Despite this, the simulation achieves convergence for the outlet temperature, and mass conservation is maintained.

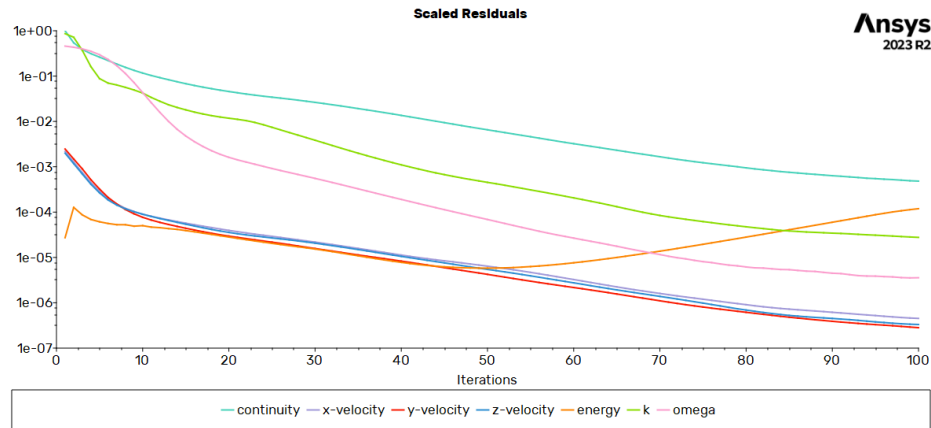


Figure 5 Residuals for the steady simulation

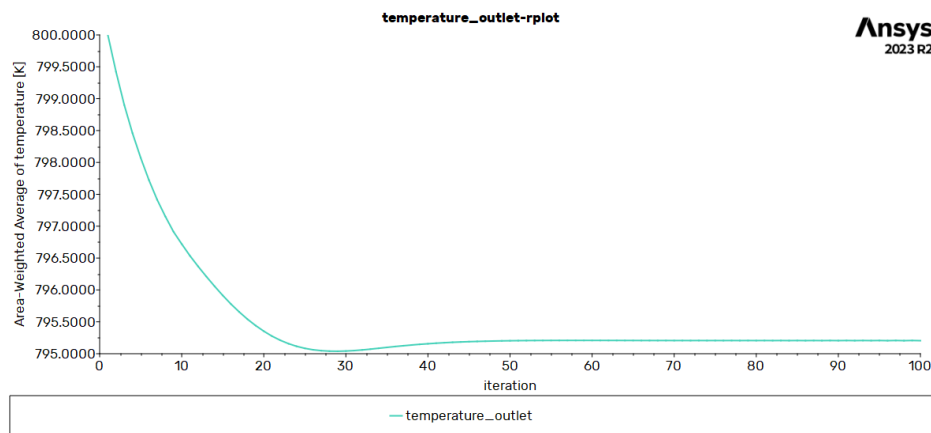


Figure 6 Temperature at the outlet for the steady simulation

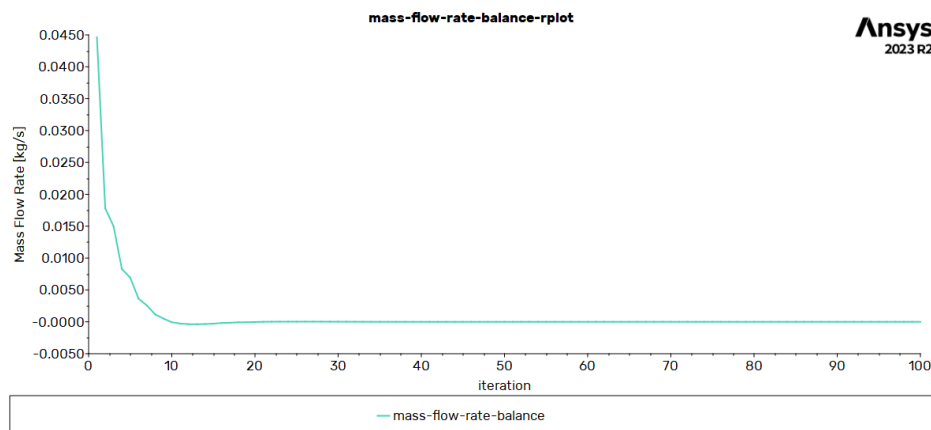


Figure 7 Mass flow rate balance for the steady simulation

**For the transient simulation :** I used the same models and methods, but I had to create User Defined Functions (UDFs) to simulate engine running. This is a simplified version, but the UDF controls the gas velocity at the inlet, which follows the curve shown below :

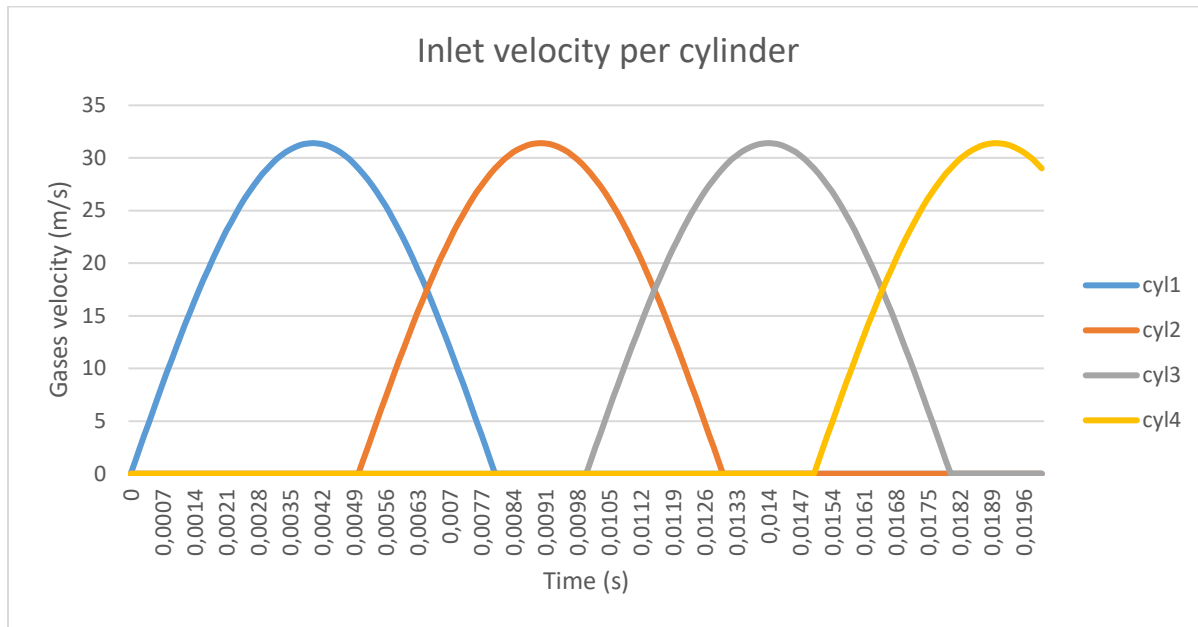


Figure 8 Exhaust gases velocity per cylinder used for the UDF's

After assigning the UDF's to each inlet, we are ready to run the calculations. To ensure a quicker simulation, I used the following parameters, knowing that one full cycle lasts 0.02 s :

- 10 time steps of 0.002s
- 10 iterations per step

To have a better simulation, I should increase the number of iterations per step but again in order to keep the computational time low I run those values.

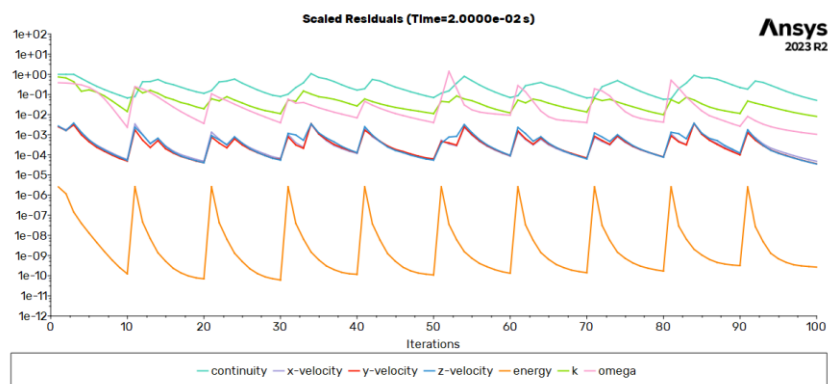


Figure 9 Residuals for the transient simulation

### III. RESULTS

For the steady simulation :

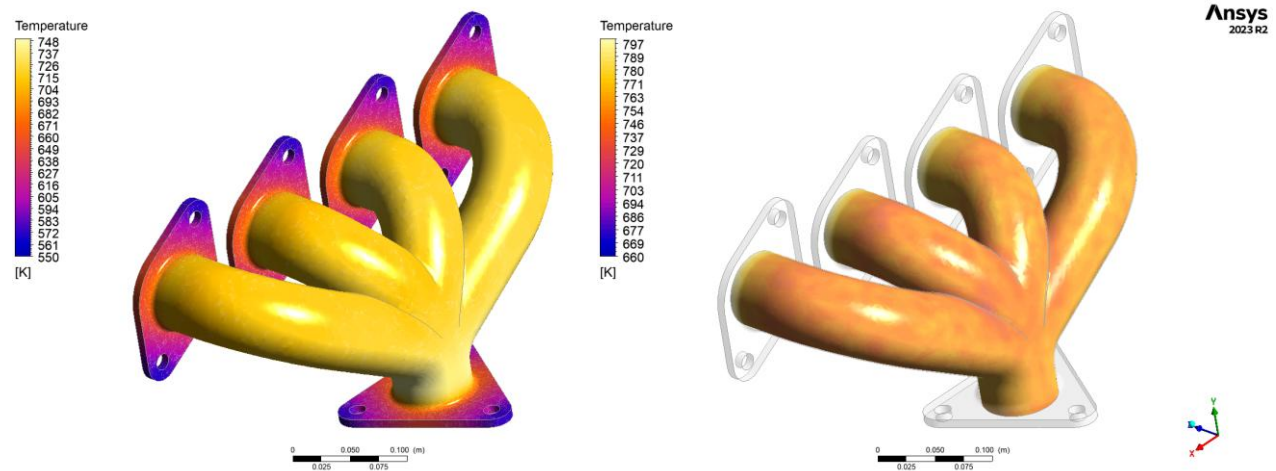


Figure 10 Temperature contour (right : solid body, left : fluid body)

The temperature contours were not surprising. In fact, the hot exhaust gas flows so quickly that the temperature inside the fluid domain remains nearly homogeneous. From the planar section view, we can also observe that the largest temperature gradient occurs within the solid body. This is why the bolt holes are placed at the coolest part of the exhaust manifold; it's a design consideration related to the flange construction.

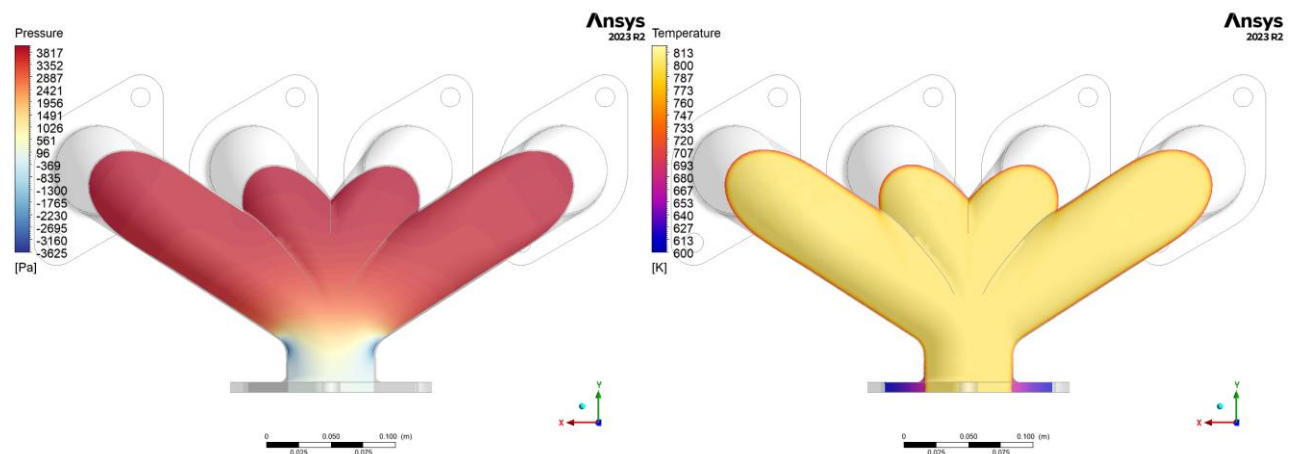


Figure 11 Planar contour : Pressure on the left, temperature on the right

The pressure builds up in the runners, but since I assumed "no back pressure", the pressure near the outlet remains very low, even lower than atmospheric pressure, where the gas flow velocity is extremely high.



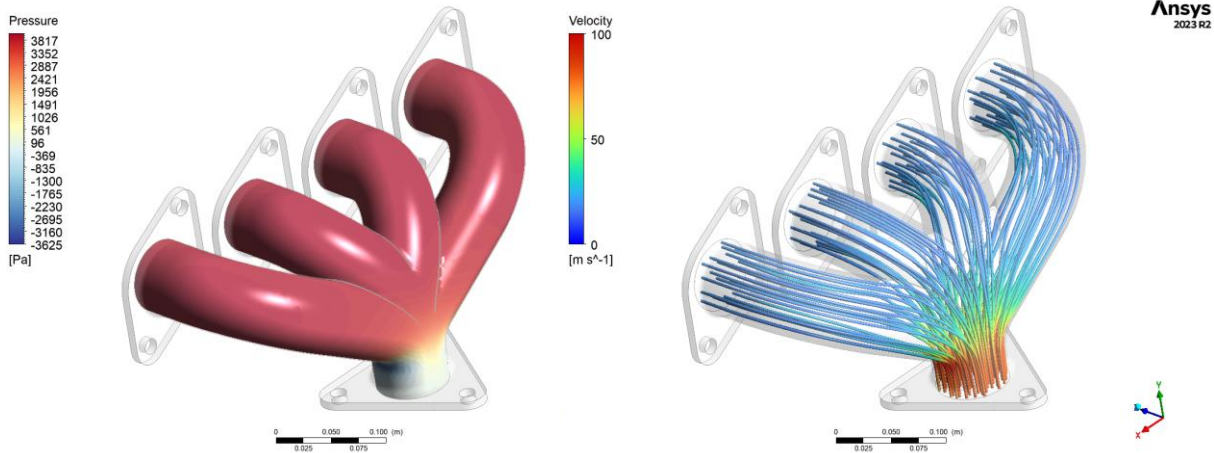


Figure 12 Pressure contour (left) velocity streamlines (right)

Same observation for the velocity, it follows our basic intuition it increases where there is a smaller section, at the bottle neck where all the runners meet.

### For the transient simulation :

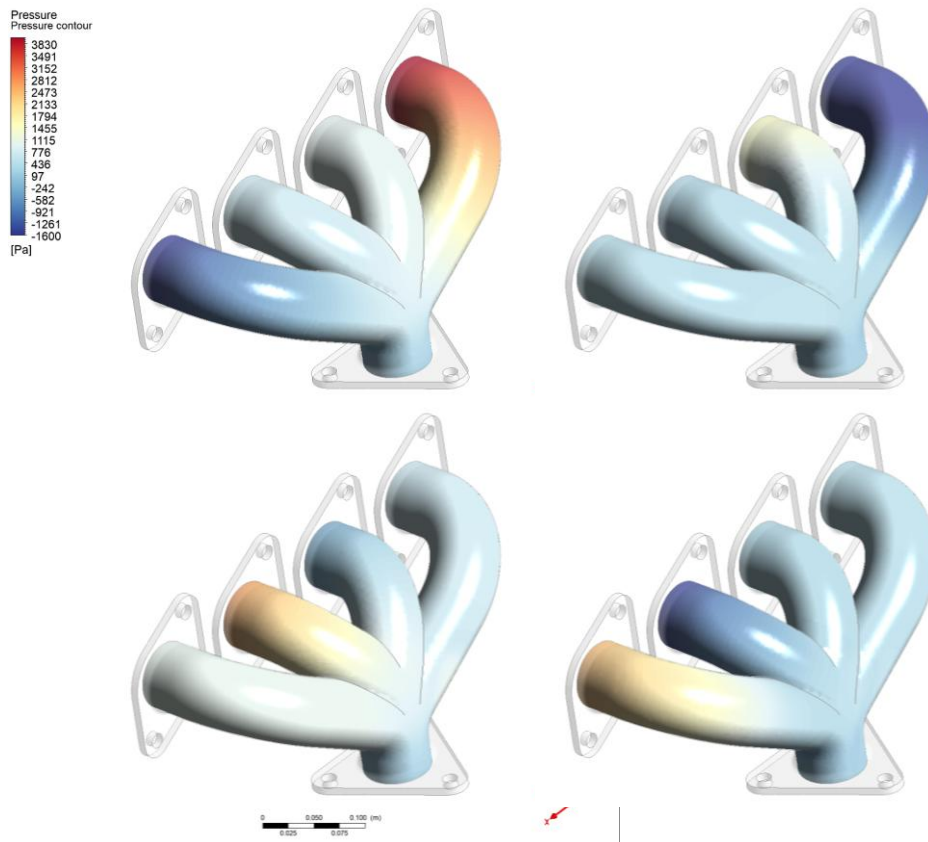
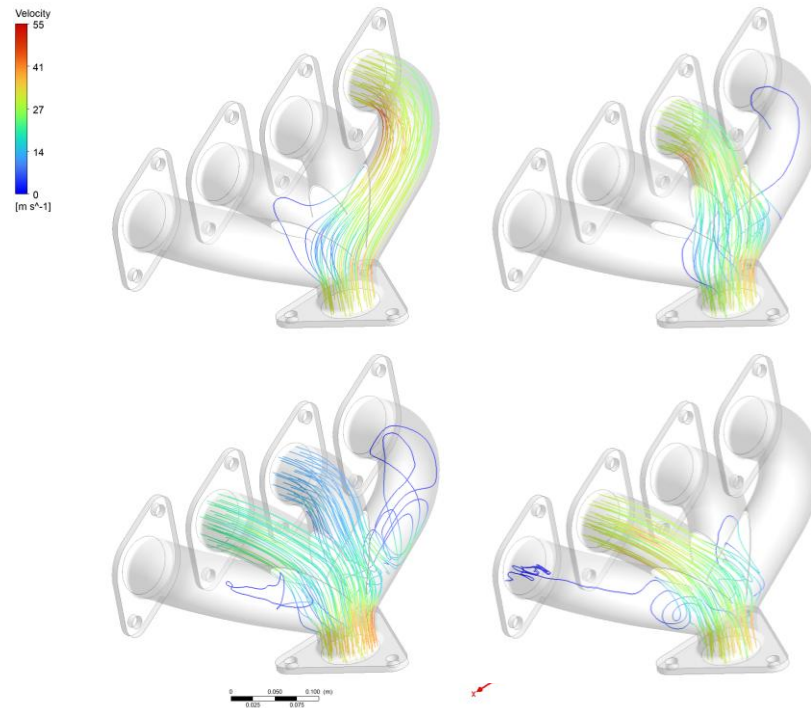


Figure 13 Pressure contour of the body fluid during the transient simulation

The simulation clearly shows the scavenging effect on this pressure contour. Specifically, when an exhaust valve opens and pressure builds up in a runner, the previously open valve experiences a pressure lower than atmospheric. For example, when the pressure in the first runner is high, the fourth runner experiences low pressure; when the second runner sees high pressure, the first runner sees low pressure, and so on.



*Figure 14 Velocity streamlines for 3 cylinders firing during the transient simulation*

We can understand why these phenomena occur by examining the velocity streamlines during the firing of three consecutive cylinders. Specifically, when an exhaust valve closes and the next one opens, it creates turbulent flow in the runner where the valve is closing. The large eddies generated in this runner may be responsible for creating very low pressure, which then leads to the scavenging effect.

#### IV. VERIFICATION & VALIDATION

To validate the mesh, I assumed that since Fluent is crashing when I put more resolution on the mesh but since I just went with the best resolution and use the  $y^+$  method, I don't think I need to perform a mesh refinement study.

To validate the steady-state model, I computed some meaningful values. While the temperature may not be highly relevant (since it depends heavily on the geometry), I included it anyway to ensure it remains consistent with the boundary conditions I set.

Activity	Average value over the surface
Average of Velocity on outlet	77.2014 [m s <sup>-1</sup> ]
Average of Pressure on inlet_cyl1	3869.65 [Pa]
Average of Pressure on inlet_cyl2	3901.68 [Pa]
Average of Pressure on inlet_cyl3	3903.77 [Pa]
Average of Pressure on inlet_cyl4	3871.21 [Pa]
Average of Temperature on fluid_body	793.422 [K]

*Figure 15 Table summing up the values for the steady simulation*

First, let's look at the back pressure. The pressure shown here represents the back pressure resulting from the exhaust manifold (and remember, we assumed that the pressure is 0 Pa at the outlet). According to a referenced document, large engines typically have a back pressure range of 6.7 to 10.2 kPa [7], so our results seem quite consistent in this regard. Additionally, since I designed the manifold to have equal-length runners, we expected the pressure to be uniform across all inlets. Finally, the gas velocity at the outlet is typically around 100 m/s [8], but since we are running this engine at a relatively low RPM (with an inlet velocity of 20 m/s), it is normal to observe a lower average outlet velocity.

## V. DISCUSSION

Now for the transient simulation, I computed the pressure at the inlet over time to see the scavenging effect, and to validate the model by comparing the value with what we are used to observed.

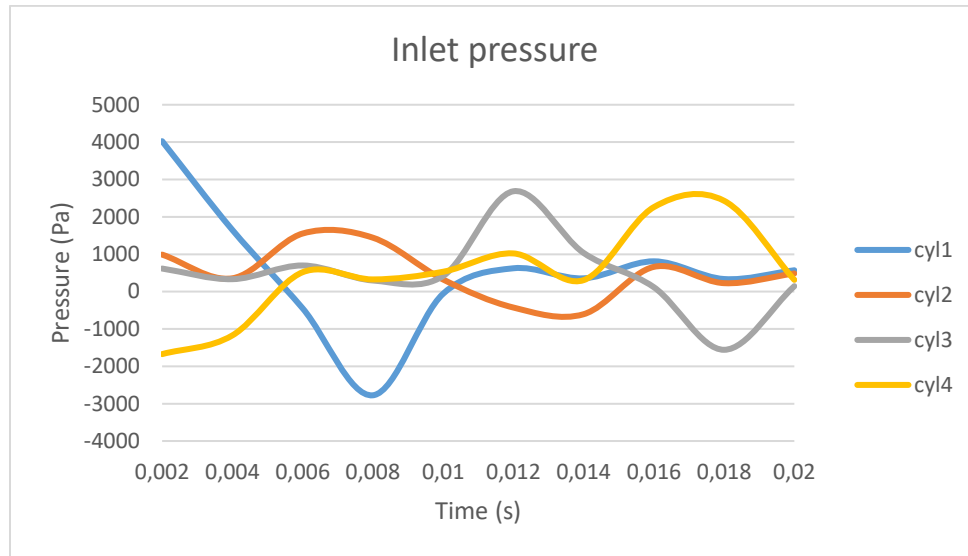


Figure 16 Inlet pressure over time for the transient simulation

This perfectly illustrates the scavenging effect. Just before the exhaust valve closes (after the high-pressure peak), there is pressure lower than atmospheric at the inlet. According to a referenced document [9], the scavenging pressure (the difference between exhaust pressure and intake pressure) can reach up to 3 bars [10]. While we are definitely lower than this value (with a pressure difference of up to 3 kPa), it's important to note that we are only running the engine for one cycle and at very low RPM.

## VI. CONCLUSIONS

In this project, I did my best to simulate the scavenging effect that occurs in the exhaust manifold. It is a complex and not very intuitive phenomenon, which is why I started with a steady-state study to validate the model, mesh, and assumptions against other studies. Once that was complete, I conducted a transient simulation over one cycle and successfully observed the scavenging effect. However, it's important to note that the results are highly sensitive to the conditions, geometry, and assumptions made, making it difficult to fully confirm whether they align with real-world behavior.

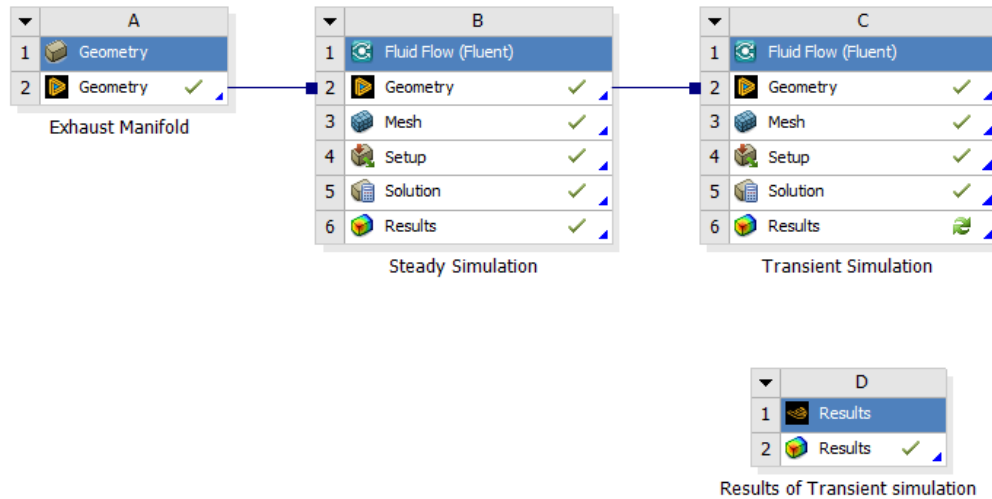
To answer the question posed in the introduction—**How does the scavenging effect appear in an exhaust manifold?**—we may not be able to provide a complete answer, but we can offer some insights. By examining the velocity streamlines, we observed the formation of large eddies and turbulent flow when one exhaust valve closes and the next one opens. This process creates pressure lower than atmospheric in the cylinder, which helps draw in the intake flow and improves combustion efficiency.

This model is not perfect and has some significant flaws. For example, the transient simulation runs for only one cycle (0.02 s), and there are only 10 iterations per time step (with residuals still relatively high, especially for the continuity equation). Additionally, I used air as the fluid instead of the actual exhaust gas mixture. All of these points can be improved to gain a better understanding of the scavenging effect.

## REFERENCES

1. Yejian Qian, Zhen Gong, Yuan Zhuang, Chunmei Wang, Peng Zhao (2018). Mechanism study of scavenging process and its effect on combustion characteristics in a boosted GDI engine. *Energy, Volume 165, Part A*, p246 – 266.
2. <http://www.machineryspaces.com/scavenging.html>
3. Gajendra Raghuwanshi, Abhay Kakirde, Dr.Suman Sharma (2018). Design and Analysis of Exhaust Manifold Comparing Different Specifications. *IJETT, Volume 62 Number 1*, p44.
4. P.Manikandan, A.Samuel Durai, S.Sathish Kumar, R.Selva Kumar, M.Navaneetha Krishnan (2017). CFD Analysis of Exhaust Manifold. *South Asian J. Eng. Technol*, 257 – 261.
5. Puspa Datta B. (2022). A Literature Review on Design of Exhaust Manifold of A Engine. *International Journal of Research Publication and Reviews, Volume 3 number 2*, p65 – p67.
6. <https://innovationspace.ansys.com/courses/courses/governing-equations-of-fluid-dynamics/lessons/simulation-examples-homework-and-quizzes-governing-equations/topic/homework-flow-through-an-exhaust-manifold/>
7. Hannu Jääskeläinen. (2007). Engine Exhaust Back Pressure. *DieselNet.com*
8. <https://www.goodfabs.com/post/how-exhaust-gas-dynamics-work#:~:text=These%20pressure%20waves%20travel%20through,about%20100%20%E2%80%93%20125m%2Fs.>
9. Hongliang Yu, Jianqun Gao, Peng Zhang, Fang Jun Han, Qizheng Yang & Bin Cui (2024). The impact of scavenging air state on the combustion and emission performance of marine two-stroke dual-fuel engine. *Scientific Reports, 14*, 15776.
10. V. Ashok Kumar, M. Madhavi, A. Krishna (2016). Manifold Optimization of an Internal Combustion Engine by using Thermal Analysis. *IARJSET, Volume 3 Issue 11*, 225 – 229.

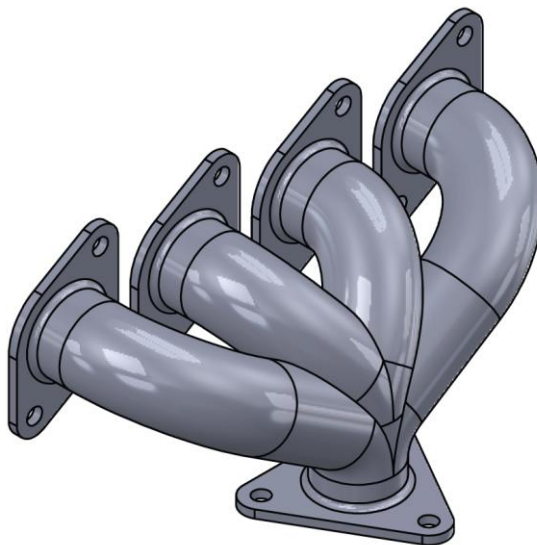
## APPENDIX 1. ANSYS WORKBENCH WORKFLOW



## APPENDIX 2. PRESSURES RESULTING FROM THE TRANSIENT SIMULATION

Inlet Pressures [Pa]										
	1	2	3	4	5	6	7	8	9	10
cyl1	4024,49	1649,51	-447,375	-2775,8	-58,445	623,538	354,121	809,549	335,914	569,831
cyl2	989,447	355,043	1553,18	1441,7	322,856	-427,236	-606,297	660,977	225,399	484,981
cyl3	616,092	331,728	698,156	296,589	417,764	2684,31	1034,44	116,354	-1558,47	144,466
cyl4	-1673,67	-1172,96	517,778	325,054	530,472	1020,98	305,785	2259,92	2429,38	313,189
Time	0,002 s	0,004 s	0,006 s	0,008 s	0,01 s	0,012 s	0,014 s	0,016 s	0,018 s	0,02 s

## APPENDIX 3. CAD OF THE MANIFOLD IN SOLIDWORKS



## APPENDIX 4. UDF C CODE TO SIMULATION THE ENGINE RUNNING

```

1  /*****
2  /* cyl1_velocity_inlet.c
3  /*
4  /* Pulsatile inlet velocity condition
5  /*
6  /* Antoine BERTRAND
7  /* Colorado State University
8  /* Department of Mechanical Engineering
9  *****/
10
11 #include "udf.h"
12
13 // Global constants
14 #define RPM 3000 // Engine speed in revolutions per minute
15 #define V_MEAN 20.0 // Desired average exhaust velocity in m/s
16 #define V_MAX (M_PI * V_MEAN / 2) // Maximum velocity calculated from the desired
    average
17 #define T_OPEN 0.008 // Duration for which the valve remains open (in seconds)
18 #define T_CYCLE (60.0 / RPM) // Time for one engine cycle for a single cylinder (in
    seconds)
19 #define M_PI 3.14159265359
20
21 // Define the UDF for velocity at the inlet of a specific cylinder
22 DEFINE_PROFILE(inlet_velocity_cyl1, thread, position)
23 {
24     face_t f;
25     real t = CURRENT_TIME; // Get the current simulation time
26     real velocity = 0.0; // Initialize the velocity to 0
27
28     // Manually compute the modulo (remainder of division)
29     t = t - 0.0; // Adjust time by the offset
30     t = t - T_CYCLE * floor(t / T_CYCLE); // Manual modulo calculation
31
32     // Calculate the sinusoidal velocity during the valve opening period
33     if (t >= 0 && t <= T_OPEN)
34     {
35         velocity = V_MAX * sin(M_PI * (t / T_OPEN)); // Sinusoidal velocity
36     }
37     else
38     {
39         velocity = 0.0; // No flow outside the valve opening period
40     }
41
42     // Apply the velocity to all cells on the inlet boundary
43     begin_f_loop(f, thread)
44     {
45         F_PROFILE(f, thread, position) = velocity;
46     }
47     end_f_loop(f, thread)
48 }
49
50 // Define the UDF for velocity at the inlet of a specific cylinder
51 DEFINE_PROFILE(inlet_velocity_cyl2, thread, position)
52 {
53     face_t f;
54     real t = CURRENT_TIME; // Get the current simulation time
55     real velocity = 0.0; // Initialize the velocity to 0
56
57     // Manually compute the modulo (remainder of division)

```



```

58     t = t - 0.005; // Adjust time by the offset
59     t = t - T_CYCLE * floor(t / T_CYCLE); // Manual modulo calculation
60
61     // Calculate the sinusoidal velocity during the valve opening period
62     if (t >= 0 && t <= T_OPEN)
63     {
64         velocity = V_MAX * sin(M_PI * (t / T_OPEN)); // Sinusoidal velocity
profile
65     }
66     else
67     {
68         velocity = 0.0; // No flow outside the valve opening period
69     }
70
71     // Apply the velocity to all cells on the inlet boundary
72     begin_f_loop(f, thread)
73     {
74         F_PROFILE(f, thread, position) = velocity;
75     }
76     end_f_loop(f, thread)
77 }
78
79 // Define the UDF for velocity at the inlet of a specific cylinder
80 DEFINE_PROFILE(inlet_velocity_cyl3, thread, position)
81 {
82     face_t f;
83     real t = CURRENT_TIME; // Get the current simulation time
84     real velocity = 0.0; // Initialize the velocity to 0
85
86     // Manually compute the modulo (remainder of division)
87     t = t - 0.010; // Adjust time by the offset
88     t = t - T_CYCLE * floor(t / T_CYCLE); // Manual modulo calculation
89
90     // Calculate the sinusoidal velocity during the valve opening period
91     if (t >= 0 && t <= T_OPEN)
92     {
93         velocity = V_MAX * sin(M_PI * (t / T_OPEN)); // Sinusoidal velocity
profile
94     }
95     else
96     {
97         velocity = 0.0; // No flow outside the valve opening period
98     }
99
100    // Apply the velocity to all cells on the inlet boundary
101    begin_f_loop(f, thread)
102    {
103        F_PROFILE(f, thread, position) = velocity;
104    }
105    end_f_loop(f, thread)
106 }
107
108 // Define the UDF for velocity at the inlet of a specific cylinder
109 DEFINE_PROFILE(inlet_velocity_cyl4, thread, position)
110 {
111     face_t f;
112     real t = CURRENT_TIME; // Get the current simulation time
113     real velocity = 0.0; // Initialize the velocity to 0
114
115     // Manually compute the modulo (remainder of division)

```

```
116     t = t - 0.015; // Adjust time by the offset
117     t = t - T_CYCLE * floor(t / T_CYCLE); // Manual modulo calculation
118
119     // Calculate the sinusoidal velocity during the valve opening period
120     if (t >= 0 && t <= T_OPEN)
121     {
122         velocity = V_MAX * sin(M_PI * (t / T_OPEN)); // Sinusoidal velocity
123     profile
124     }
125     else
126     {
127         velocity = 0.0; // No flow outside the valve opening period
128     }
129
130     // Apply the velocity to all cells on the inlet boundary
131     begin_f_loop(f, thread)
132     {
133         F_PROFILE(f, thread, position) = velocity;
134     }
135     end_f_loop(f, thread)
136 }
```