Reducing Time to Market of Differential Systems Using Cloud GPU-Accelerated CFD

Felix Pause, Malwine Fischer, Filippo Boscolo Fiore
(Dive CAE, Germany);

Ian Pegler
(NVIDIA, USA);

Daniel Derrix
(BMW Group, Germany)

Abstract

In today's competitive automotive landscape, accelerating time to market and optimizing cost efficiency are critical. BMW Group and Dive CAE have examined how advancements in computational fluid dynamics (CFD) can address these challenges, focusing on GPU acceleration, cloud parallelization and Smoothed Particle Hydrodynamics (SPH).

The study examined drivetrain design projects, particularly the development of new differential systems. Modern differential systems are becoming increasingly complex, posing significant challenges for design engineers. Additionally, evolving safety, environmental, and performance standards demand iterative redesigns and extensive testing, lengthening development cycles. Several operating points and designs were compared and assessed with respect to oil churning losses and comprehensive oil coverage of system components.

The SPH method is particularly effective for modelling problems of this type, involving free-surface or multi-phase flows. Moreover, unlike grid-based methods, its Lagrangian, particle-based framework naturally handles complex geometries and moving components without requiring re-meshing. Additionally, it reduces manual pre-processing work, paving the way for automation of large parallel simulation studies.

Dive CAE employs a Weakly Compressible SPH approach (WCSPH), incorporating a variety of measures relevant to industry-level accuracy and usability. Key methods include a semi-analytical integral boundary condition to improve near-wall flow accuracy. This paper outlines the theoretical foundations of the method and provides selected validation results.

GPU acceleration of the SPH code demonstrates a runtime reduction by a factor of 6-32 compared to CPU architectures. Cloud parallelization enabled

concurrent testing of 12 operating conditions, shortening project turnaround time (TAT) by a factor of 5 compared to an on-premise setup. Eventually, the paper also includes an analysis of the cost effect of migrating the simulations to GPUs and the cloud.

In conclusion, the study examines how GPU acceleration, cloud technologies and SPH contribute to overarching goals of accelerating time to market and reducing costs.

1. Introduction

The automotive industry faces pressure to reduce cost and shorten development cycles while meeting increasingly stringent performance, safety, and environmental standards. Accelerating time to market is critical, particularly for complex components such as differential systems. These systems, integral to modern drivetrains, involve intricate geometries and multiphase lubrication processes that challenge traditional simulation methods.

Conventional grid-based CFD techniques—such as finite volume and finite element methods—have long been the workhorses of fluid simulation. However, the requirement for detailed mesh generation and frequent re-meshing in the presence of free surfaces limits their efficiency. In contrast, Smoothed Particle Hydrodynamics (SPH) offers a mesh-free, Lagrangian approach that naturally handles moving boundaries and free-surface. Since its introduction in 1977 by Gingold and Monaghan [1], SPH has evolved into a robust tool for simulating complex fluid phenomena, with applications ranging from astrophysics to industrial processes [2]. In automotive engineering, SPH has been effectively applied to gearbox lubrication analysis, enabling detailed predictions of oil distribution, churning losses, and bearing performance [3] [4].

The advantages of SPH are further amplified by recent advances in hardware acceleration. Modern GPUs offer massively parallel processing capabilities that can drastically reduce simulation runtimes compared to traditional CPU architectures. When combined with cloud computing, these accelerators facilitate the concurrent execution of multiple simulation scenarios, significantly shortening project turnaround times and reducing costs. This integrated approach enables rapid design space exploration without the extensive pre-processing overhead typically associated with grid-based methods.

This paper presents a study on the integration of SPH with GPU acceleration and cloud parallelization to optimize CFD simulations for differential system design. We detail the theoretical foundations of SPH, validate our approach against experimental benchmarks, and demonstrate substantial runtime and cost savings relative to conventional CPU-based workflows. Our work extends

previous research in the field, contributing to the evolving body of knowledge aimed at democratizing high-fidelity simulations for automotive applications.

2. Theoretical Foundations of the SPH Method

In Computational Fluid Dynamics (CFD), fluid flows are described using basic conservation principles in the form of balance equations, the so-called Navier-Stokes equations (NSE). The NSE form a system of nonlinear partial differential equations, that assure the conservation of mass, momentum and energy.

Two paradigms are followed when representing the properties of a fluid: the Eulerian representation and the Lagrangian representation. The Eulerian formulations are the foundation of classical mesh-based methods, such as the Finite Volume Methods (FVM), Finite Differences Method (FDM) and Finite Elements Methods (FEM), where a continuum is discretised by a network of spatially fixed communicating cells. In the Lagrangian formulation, the flow is represented by moving control domains, abstracted by particles.

Figure 1: Comparison of Eulerian and Lagrangian Discretization

The SPH solver used here is based on the modelling approaches presented by Sabrowski et al. [5]. It employs a weakly compressible SPH model (WCSPH). To reduce computation time, an artificial speed of sound c_0 is introduced, chosen so that small density variations of less than 1% are possible [6] [7]. This approach is valid since compressible effects can be neglected for Mach numbers Ma < 0.1. In WCSPH, the incompressible momentum equation is used alongside the compressible continuity equation. The equation of state couples the mass density ρ to the pressure p, thus closing the system of partial differential equations [8].

Particle advection: $\frac{dx}{dt} = \boldsymbol{v}$

Compressible continuity equation: $\frac{d\rho}{dt} = -\rho\nabla \cdot \boldsymbol{v}$

Incompressible momentum balance: $\frac{d\boldsymbol{v}}{dt} = -\frac{1}{\rho}\nabla p + \nu\,\Delta\boldsymbol{v} + \boldsymbol{g}$

Cole equation of state [8]: $p = \frac{\rho_0 c_0^2}{\gamma}\left(\frac{\rho}{\rho_0} - 1\right)^{\gamma} + p_0$

with the position of the particles $\boldsymbol{x}$, velocity $\boldsymbol{v}$, time t, density ρ, pressure p, kinematic viscosity ν, gravity $\boldsymbol{g}$, initial density ρ_0, background pressure p_0 and the numerical speed of sound c_0. The exponent γ was experimentally determined to be 7 for liquid phases [8]. At the position of each particle, a weighted interpolation is applied using a kernel function. With the kernel function, the properties of any particle are calculated based on the distance to neighboring particles and their respective properties. This yields the forces that are used in the Lagrangian NSE to determine the particle's trajectory in space.

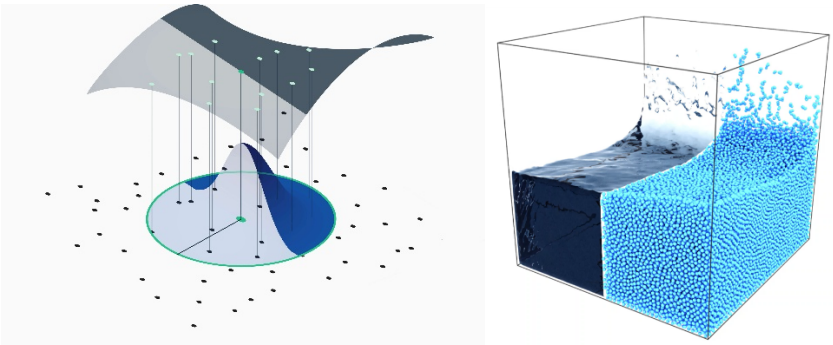


Figure 2: Left: Kernel Function, Right: Visualization of a set of particles as a representation of a continuum.

Modelling boundary conditions is one of the challenges in the SPH framework.[1] If the kernel intersects with a boundary, zeroth order consistency is no longer given. In current literature several approaches are described to determine the missing influence of the truncated kernel area. One well-established framework is to discretize the bordering geometries using buffer particles, similar to the fluid field. This prevents the kernel from being truncated [9]. This approach is straightforward but requires filling the geometry with boundary particles, thereby linking the size of the boundary particles to the resolution of the fluid particles, increasing computational effort [5]. An alternative is to represent the boundary, using a segment-based mesh. Here, a renormalization factor γ_R is introduced [10] and the volume integral is transformed into a boundary integral [11].

Other relevant modelling aspects of industry-grade SPH codes include the modelling of:

- open boundaries [12],
- surface tension effects [13] [14],
- varying particle resolution [15],

[1] „Grand Challenges" of SPH are defined by the SPHERIC steering committee, the international organization representing the SPH research community [34].

- efficient and consistent time-stepping [16] [17],
- numerical stabilization techniques,
- density diffusion [18], and
- particle shifting [19] [20].

3. Validation of the method

The accuracy of the underlying models as well as their concrete implementation within the Dive SPH solver have been extensively verified and validated against established benchmarks and experimental data. Several studies have demonstrated its effectiveness in capturing complex flow dynamics across various scenarios [21] [22] [23] [24] [3] [25] [26] [4]. Two examples will be introduced briefly.

The lubrication of transmission components, such as gearboxes, is a key application for SPH simulations, given the high-speed gear motion and the complex multiphase flow of air and lubricant. To validate the model, the no-load test rig at the Forschungsstelle für Zahnräder und Getriebe (FZG) at TU Munich has been used [4]. Specifically, the power lost because of the viscous and pressure forces exerted by the oil onto the gears is calculated from the simulation. The so called “churning losses”, determined with the SPH code of the authors show excellent agreement with the experimental values found in [27].

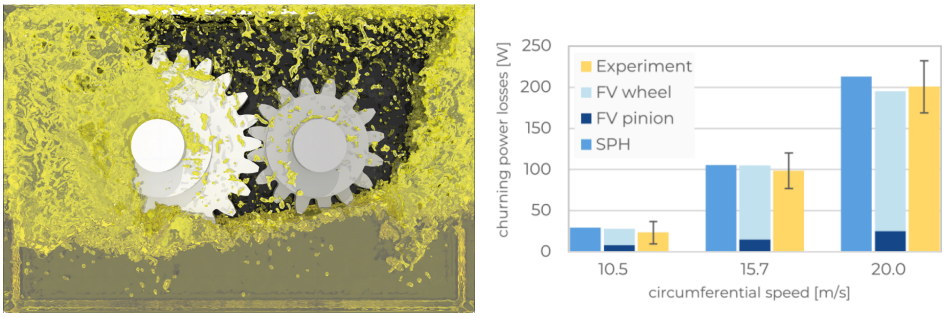
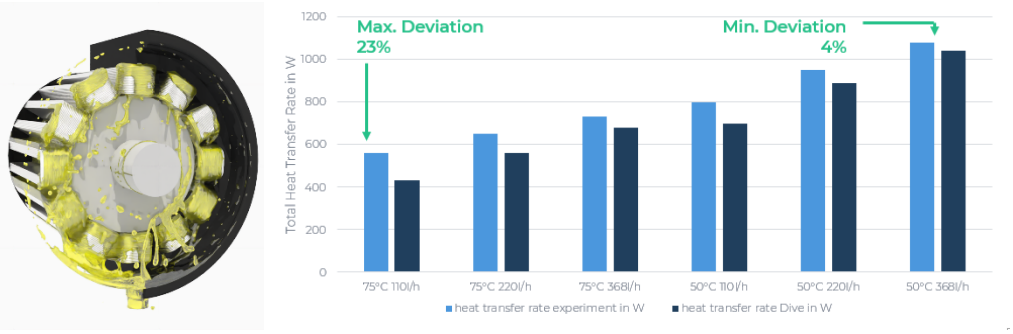


Figure 3: Results published by [MENSAH2022]. Left: Snapshot of the flow field in the FZG gearbox. Right: Comparison of SPH results with experimental values and classical CFD.

Another relevant validation is the active oil cooling system of an electric motor. The heat transfer rates of the experimental study described in [28] were closely matched for several operating points [29].



Results published by [29]. Left: Snapshot of the flow field in the electric motor. Right: Comparison of SPH results with experimental values.

4. Objective in Differential System Design

The focus of this work is an investigation of lubrication in a differential system. This use case presents multiple challenges for the CFD method used.

1. **Multiphase Flow Behavior:** Accurately capturing oil-air interactions requires a method capable of dynamically tracking free surfaces.
2. **Complex Moving Geometries:** The drivetrain consists of intricately shaped gears, bearings, and housings, making grid-based CFD methods time-consuming and difficult to use efficiently.

These factors make SPH a suitable approach for this problem.

The differential gearbox analysed in this study is shown below, with oil channels integrated into the housing to direct lubricant to the bearings efficiently.

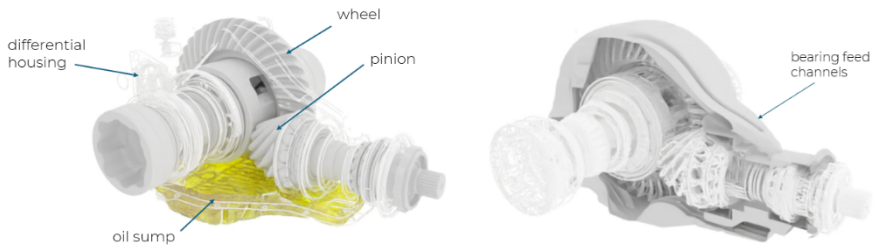


Figure 4: Left: Components of the gear train. Right: Position of feed of input bearings

The primary objective is to ensure sufficient bearing lubrication under all operating conditions, critical for preventing premature failure and costly replacement of gearboxes in the field. A secondary goal is to minimize oil churning losses to improve the efficiency of the transmission.

Design Goal	Measured By	Relevance
(Primary) Bearing Lubrication & Lifetime	Volume flow rate through bearing feed channel, oil amount in bearings, wetting levels	Proper lubrication reduces wear and heat buildup, extending component lifespan
(Secondary) Oil Churning & Efficiency	Gear churning losses	Excessive churning leads to energy losses, impacting fuel efficiency

Table 1: Design Goals

This study evaluates design variations by modifying the hypoid gear spiral angle while maintaining a constant number of teeth. The goal is to determine how these variations influence oil transport within the system and whether they can be leveraged to direct oil flow to critical lubrication points. In case both versions turn out to be equivalent, they can be interchanged without further impact on the system. The versions considered (A, B) are depicted below.

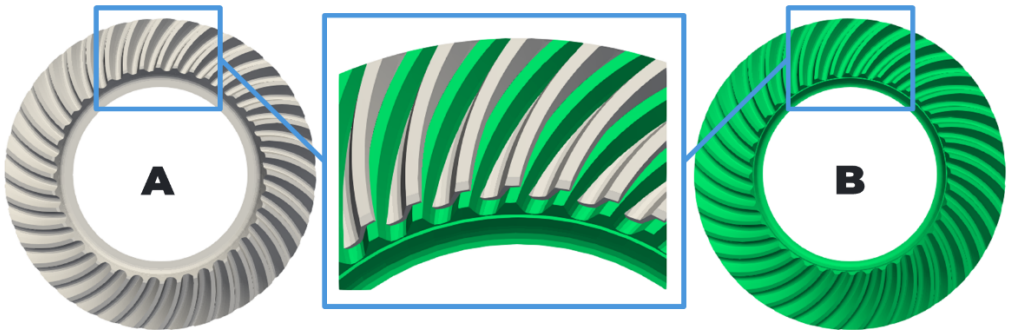


Figure 5: Comparison of gear design variants A and B.

The simulations are conducted across a range of operating conditions, varying rotational velocity and oil temperature:

Input Speed (RPM)	2000, 4000, 6000
Oil Temperature (°C)	45, 100

Table 2: Operating Conditions

Other real-world conditions, such as vehicle acceleration, recuperation, braking, cornering and hill climbing (to ensure the breather remains dry), are beyond the scope of this study but are commonly considered in similar analyses.

5. Setup and Boundary Conditions

In accordance with Table 2, a total of 12 operating points is set up. The input velocity and the rotational velocity resulting from the gear ratio are applied to the pinion and the wheel, respectively. Both sets of bearings rotate at half of the rotational velocities of their respective gear. In the first 0.1s, all rotating parts are linearly accelerated until they reach their corresponding rotational velocities. In accordance with the integral boundary condition introduced in the previous chapter, solid parts are represented in a triangulated way, using CAD generated .stl file. An example is depicted below.

Figure 6: Pinion Mesh

The oil sump is initially discretized with a particle diameter of 0.001 m, resulting in approximately 2.25 million particles. To better represent the effects in the gears and channels while saving computational cost, a total of six refinement zones is added. This includes five cylindrical zones around the wheel and pinion as well as their bearings and one cuboid zone around the channels and the pinion shaft. Note that, even though it is possible to conduct heat transfer simulations with SPH, the cooling effect of the oil is not of interest for this analysis and the different temperatures of the oil only affect the density and the viscosity of the lubricant. For 45°C, a density of 832.9 kg/m^3 and a kinematic viscosity of 0.0000344 m^2/s is utilized and for 100°C the oil has a density of 800.8 kg/m^3 and a kinematic viscosity of 0.000009 m^2/s. Surface tension is considered with a surface tension coefficient of 0.03 N/m and a contact angle of 30°.

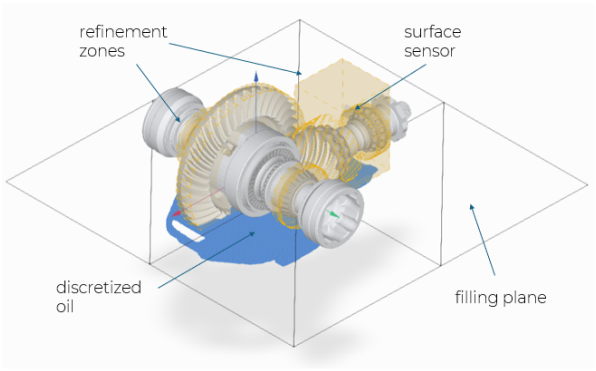


Figure 7: Simulation setup displaying the discretized oil, the refinement zones and the position of the bearing feed volume flow sensor.

6. Simulation Results

Visualizations of the flow dynamics predicted by the SPH solver are depicted below. The oil surface is rendered to highlight its trajectories. As the speed increases, the oil distributes more extensively within the housing and agglomerates in larger quantities around the bearings.

2000 RPM	

4000 RPM	
6000 RPM	

Table 3: Comparison of the oil distribution with increasing speed for 45°C in gear set A at 0.6 s. With increasing rotational speeds, more oil can be found in the channels feeding the bearing and in the region around the pinion shaft.

Volume Flow

The volume flow through the feed channels in the housing (as shown in Figure 4) serves as an indicator for the wetting of the input bearings. It is measured at a sensor placed at the end of the tubes. The differential is simulated for a duration of 0.6 s for 6000 rpm, 0.7 s for 4000 rpm and 1.1 s for 2000 rpm as lower rotational velocities cause oil to reach the sensors later.

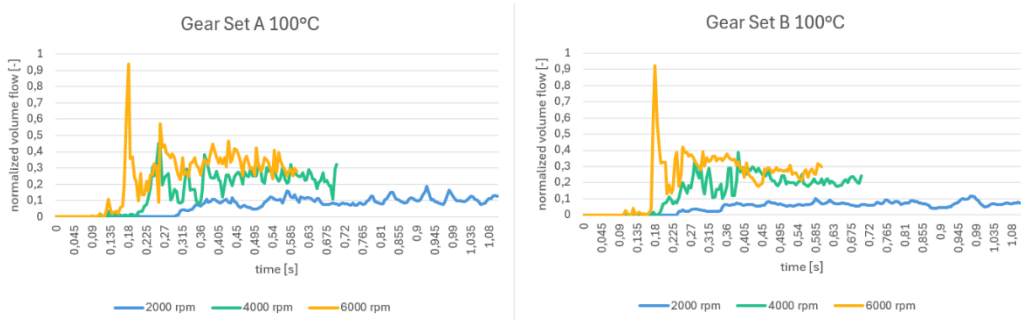


Figure 8: Volume flow of different rotational speeds at 100°C for gear set A (left) and B (right)

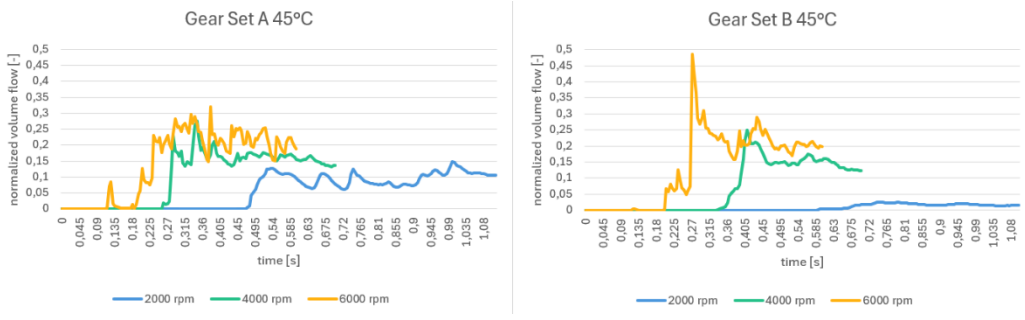


Figure 9: Volume flow of different rotational speeds at 45°C for gear set A (left) and B (right)

In both gear sets, the volume flow through the channels increases with the rotational speed of the gears (Table 3). Moreover, a higher temperature leads to an increased volume flow through the channels resulting from the lower viscosity and density of the oil. After the ramp-up phase, the oil takes longer to reach the surface sensor in the differential with gear set B compared to gear set A (Figure 8). Furthermore, the volume flow in gear set B exhibits fewer fluctuations over time. For the rotational speeds of 6000 rpm and 4000 rpm, only minor differences in the achieved average volume flow rate can be observed (Figure 8). More significant differences appear at the lowest rotational speed, especially for a temperature of 45°C, where the volume flow rate in set B stays far below the rate in set A (Figure 9). A close-up comparison (Figure 10) gives deeper insights into this effect. In gear set A, the oil feeding channel is filled, whereas in gear set B, it is only partially filled. Caused by the different spiral angles, the oil is deflected with different trajectories, leading to a significant change in the oil volume flow in the area of interest.

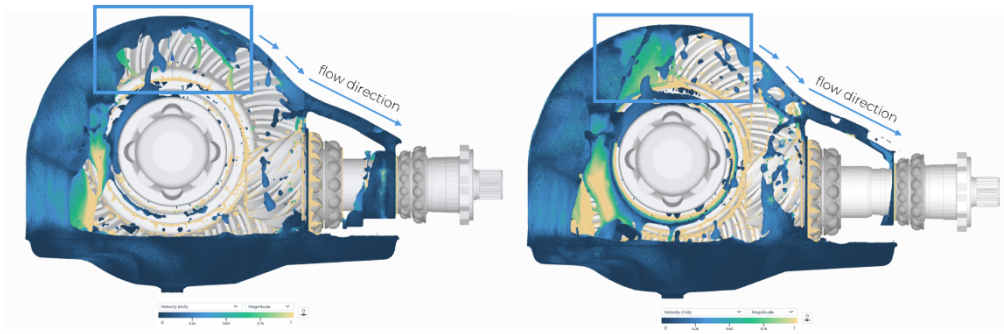


Figure 10: Oil flow at 45°C 2000 rpm for gear set A (left) and gear set B (right) at 1.1 s. In gear set B, oil descends earlier along the wheel before reaching the feeding channel.

Churning Losses

Churning losses occur when a rotating component interacts with a surrounding fluid, causing energy dissipation due to fluid resistance. The value of the power losses is measured at the boundaries of the rotating parts as the integral of the shear and normal stresses acting on the surface of the solid parts. The total loss is obtained by aggregating the losses of all parts. Due to fluctuations, the average values during the last 0.1 s of the simulation are compared.

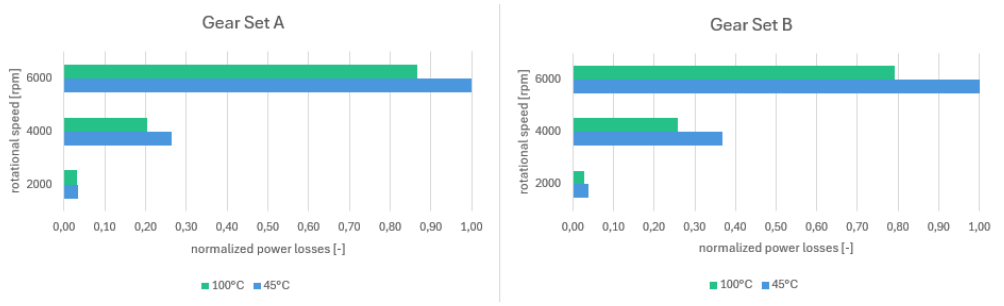


Figure 11: Comparison of the power losses of different oil temperatures at different rotational speeds for gear set A (left) and gear set B (right)

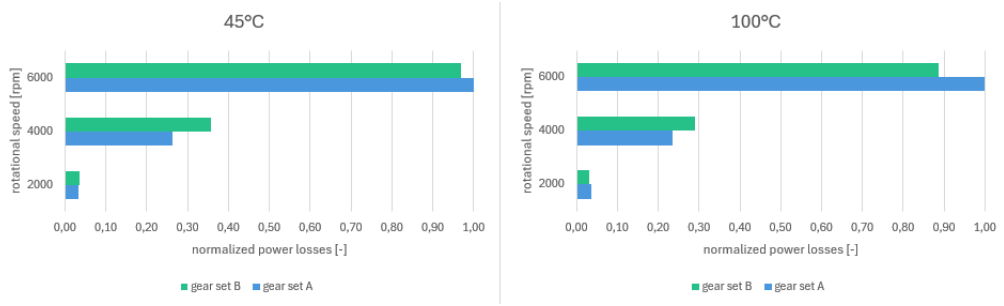


Figure 12: Comparison of the power losses of the gear sets at different rotational speeds at 45°C (left) and at 100°C (right)

As expected, the power losses show a strong dependency on the rotational speed and on the temperature for both gear sets. While the power losses increase superlinearly with increasing rotational speed, higher oil temperatures consistently decrease power losses due to the lower viscosity of the fluid and the reduced friction (Figure 11).

The analysis is more nuanced when comparing gear sets A and B (Figure 12). The difference in power losses ranges from approximately 1% to 10%. For lower temperatures, gear set B shows better efficiency at high rotational speeds, while the performance of gear set A is superior at medium and lower rotational speeds. For higher temperatures, a similar observation can be made for high and medium rotational speeds, where gear set B and gear set A show lower losses, respectively. For a lower rotational speed of 2000 rpm, gear set B performs better than gear set A.

To choose the optimal design, prioritization of the different operating conditions can now be performed, e.g. based on the expected driving profile.

7. Investigating the Impact of GPU and Cloud acceleration on Time to Market and CAE cost profile

The study presented in the previous chapter is now used as a baseline to analyse the potential of GPU acceleration and cloud workload parallelization. When evaluating the impact of R&D, particularly CAE, on time to market and project lead times, multiple factors must be considered. The entire lifecycle—from customer request to final design—plays a role, but this study focuses on the CAE-specific problem-solving level. Key time contributors include preprocessing, solving, postprocessing, and parallelization potential. As shown in the previous chapter, a typical project setup involves investigating approximately 12 operating points. This number is confirmed by an analysis of average load profiles of drivetrain design teams later in this chapter.

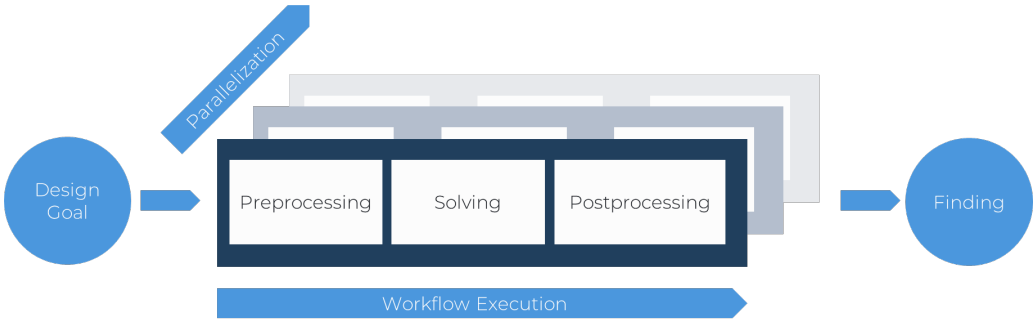


Figure 13: Simplified visualization of a typical problem-solution loop in CAE.

Effect of GPU Acceleration on Single Workflow Velocity

Using the problem described in the previous chapter, modern GPU architectures were evaluated for potential speedups. GPUs can outperform CPUs in CFD workloads due to their massively parallel architecture, which efficiently handles SPH's computational demands [30] [31]. This parallelism accelerates particle interaction processing and significantly reduces simulation time. For reasons of simplicity, the potential of multi-node parallelization is excluded from this analysis. The configurations used in the benchmark are listed in the table below. Except for one system, all tests were conducted on the Microsoft Azure cloud.

Name	Hardware	GPU	Azure Name	Used Cores
16 cores	Intel Xeon Platinum 8168 with 32 cores, 64GB RAM	-	Standard_F32s_v2	16
120 cores	2x AMD EPYC 7V73X with 64 cores, 448GB RAM	-	Standard_HB120rs_v3	120
A100	AMD EPYC 7V13 with 64 cores, 220GB RAM	NVIDIA A100 PCIe, 80 GB VRAM	Standard_NC24ads_A100_v4	1
H100	AMD EPYC 9V84 with 96 cores, 320GB RAM	NVIDIA H100 NVL, 94GB VRAM	Standard_NC40ads_H100_v5	1
B200	INTEL XEON PLATINUM 8570 with 56 cores	NVIDIA B200 SMX, 184GB VRAM	NA	1

The hardware selection was based on the following reasoning. The performance of "16 cores" Intel Xeon Platinum 8168 CPUs (32 physical cores)

is comparable to processor that have been traditionally used in HPC data centers. Therefore, it serves as a reference in this study. “120 cores” AMD EPYC 7V13 (128 physical cores) has demonstrated in previous tests of the authors to show a strong balance between cost and performance for their SPH code. Due to virtualization, not all physical cores are used for computations.

“A100” and “H100” GPUs are selected because of their high performance and availability at the time of writing. The “B200” is included to represent the capabilities of the newest GPU generation on the market.

Performance was tested for two configurations: a coarse resolution (4 million particles) and a high-resolution case (60 million particles). The high-resolution case was not run on the 16-core hardware, due to exceedingly high runtimes. The graph below shows the speedups.

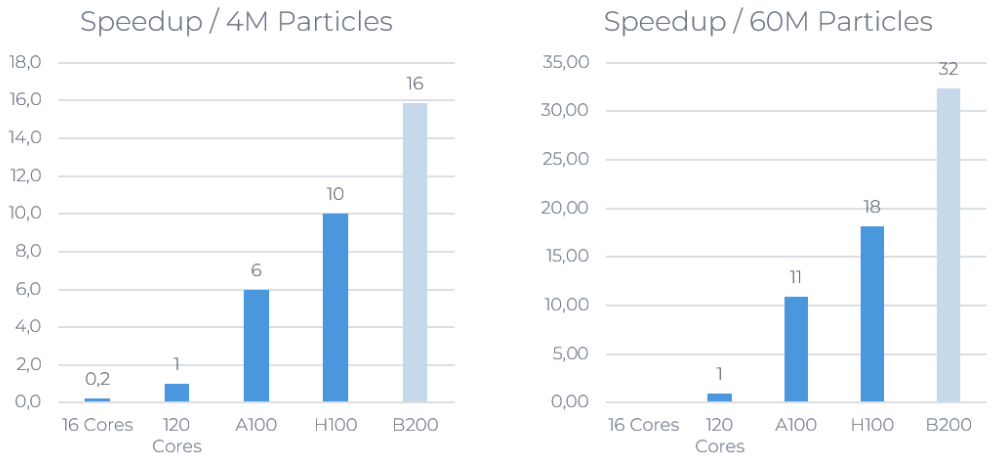


Figure 14: Speedups achieved using GPUs

Results confirm the impact of GPU acceleration, showing 6x and 11x speedups when transitioning from a 120-core AMD CPU to the A100 GPU, effectively reducing simulation runtime from days to hours. On the newest B200 GPU, a speedup of 32 is observed.

Additionally, newer GPUs enable high-fidelity simulations with tens of millions of particles on a single node, allowing for higher levels of detail in large-scale studies.

Cost Effect of GPU Hardware

To estimate the cost impact of GPU migration, we compared the cost per simulation using Microsoft Azure Cloud list prices at the time of writing [32]. Prices from other vendors were omitted, as they are comparable, and our focus is on relative differences. Comparing on-premise hardware costs is challenging

due to significant vendor price variability. The following hourly prices are used:

- 16 Cores: 1,35 €/h
- 120 Cores: 3.60 €/h
- A100: 3,67 €/h
- H100: 6,98 €/h
- B200: N/A

The 120-core CPU serves as the baseline. Using the speedup values from the previous chapter and the per-hour pricing for the various hardware configurations, cost changes are computed for both the coarse and high-resolution cases; a value below 1 indicates cost savings.

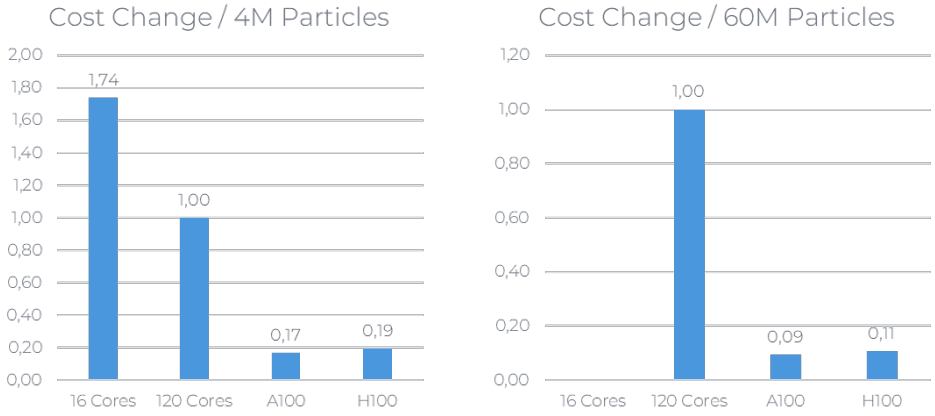


Figure 15: Cost effect of GPU migration on a single simulation

The increased performance of the GPUs translates to comparable cost savings due to similar per-hour pricing. In this example, migrating the fine simulation to the H100 yields cost savings of ~9x. This is primarily due to energy consumption being the main cost driver in cloud computing; hence, despite higher initial procurement costs, GPUs exhibit operational cost similar to CPUs.

Effect of Cloud Workflow Parallelization

Beyond accelerating individual simulations, the parallel execution of multiple workflows is crucial for reducing overall time-to-results. This study employs a workflow similar to the approach shown in [33]. STL files that represent the wall boundaries are created from a CAD tool. Via a Python SDK, preprocessing, simulation execution, and postprocessing are orchestrated and run in parallel in the cloud. It can be assumed that the maximum workflow batch size discussed here (10-20), can always be provisioned by the cloud

vendor. Therefore, there are no restrictions on peak load scaling in the cloud environment.

Estimating the potential speedup over an on-premise system is complex due to infrastructure variability and different usage patterns across organizations. On-premise systems inherently face capacity constraints. HPC resources are shared and create bottlenecks in the daily operation. Additionally, in traditional CAE environments, user groups also compete for availability of shared software licenses. For many organizations, this even becomes the main factor limiting CAE usage.

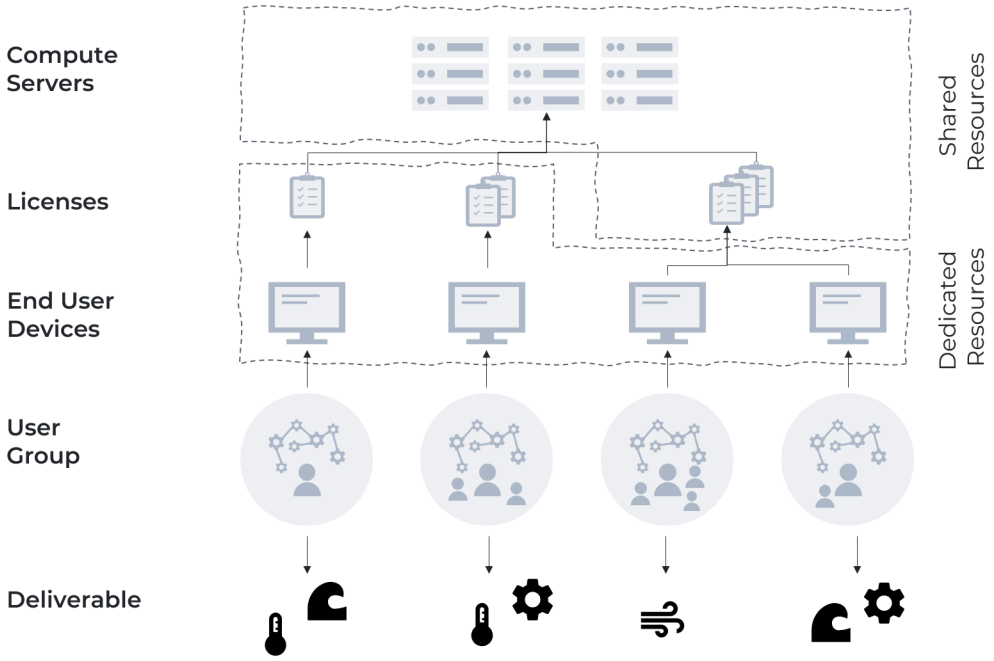


Figure 16: Traditional CAE Environment with Shared Resources (HPC, Licenses).

Any user group in this setup has the responsibility to deliver results fast in critical project situations and is aware of the system capacity and presence of all other stakeholders. In consequence, they influence each other's work and certain access patterns emerge. Incentives are created to queue strategically and assure a high utilization to secure investment in large HPC infrastructures and license stacks. Additionally, the entailing high utilization of the system discourages user groups that do not access the resources yet.

To estimate demand in an unconstrained environment—free from queue management, resource rationing, and license bottlenecks—we analysed workload patterns from transmission design teams operating entirely in the cloud. Data was collected from four independent industrial and automotive

transmission teams over one year.

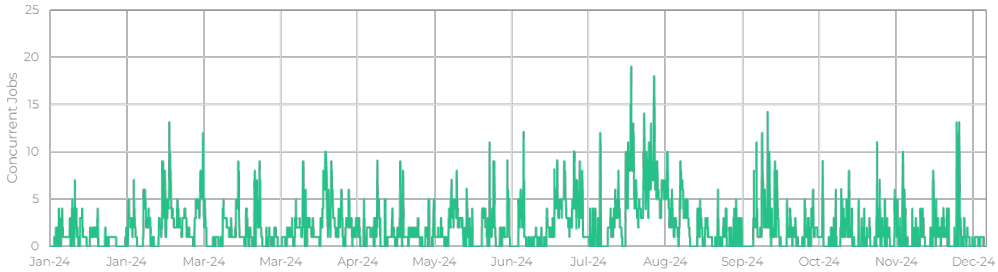


Figure 17: Concurrent simulation jobs across four transmission design teams.

The results indicate strong workload fluctuations, with peak usage reaching 15–20 concurrent jobs, aligning with our previous case study. Baseload utilization is relatively low, with extended idle periods. This reflects the project-driven nature of engineering teams, where computational demand varies based on evolving customer requirements. A breakdown of time spent in different load states provides insight into adequate infrastructure provisioning.

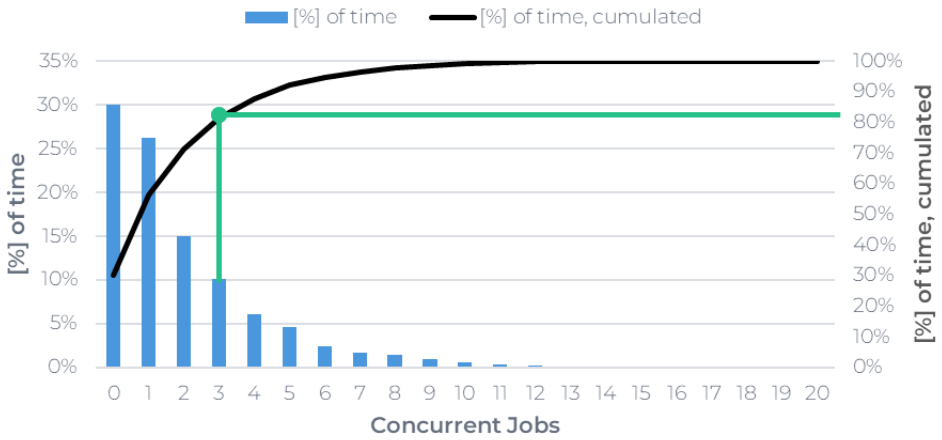


Figure 18: Time spent in different load cases. Approx. 80% of the time, 3 nodes or less are used.

Approximately 30% of the time is idle, while ~80% of the time fewer than three jobs run concurrently. This variability presents challenges for on-premise system provisioning. Optimizing for cost suggests covering baseline demand (2–3 nodes), but peak loads would then require sequential execution, extending project timelines by a factor of 3–4. Under-provisioned clusters and license stacks (<4 nodes / licenses) further discourage users from fully utilizing available resources.

To quantify this effect, we modeled three setups:

- Minimum Baseload Coverage (2 nodes)

- Medium Utilization (3 nodes)
- Peak Load Coverage (15 nodes)

Two assumptions are made to compare these setups with the cloud.

(a) The workload stays constant when transitioning to a fixed capacity system
(b) Software licenses are always available at the necessary volume.

Especially (a) is a strong assumption considering the reasoning earlier in the chapter. It would require that jobs can always be queued and that user groups do not change their behavior because of the known limitations of the system. The table below estimates utilization and slowdown effects. If these assumptions are met:

System Size	Available Nodes	Overall Utilization	Peak Load Slowdown
Minimum	2	100%	7,5
Baseload	3	66%	5
Peak Load	15	13%	1

Given a typical on-premise setup designed for "Baseload" coverage, solving CAE problems would experience an estimated 5x slowdown in peak load scenarios.

Overall Cost Comparison

After establishing comparable on-premise environments based on real-world usage data, we now compare cloud and on-premise costs. This analysis is complex due to diverse cost factors associated with CAE workloads, including:

Infrastructure Costs	1. Compute Infrastructure (CPU / GPU) 2. Storage & Networking 3. Client Devices 4. Power 5. Cooling 6. Maintenance

	7. IT Administration
Software Costs	8. Solver Licenses 9. Preprocessing & Postprocessing Licenses 10. IT Administration 11. User Training

Table 4: Costs associated with running CAE workloads.

For simplicity, we focus on (1) compute infrastructure and their electricity consumption. Other factors, such as license costs and procurement discounts, vary widely between organizations. The numbers used in this chapter are intended to give an indication and allow constitute a comprehensive economic analysis. Instead, they provide a first assessment of the economic boundary conditions, offering a rough estimation rather than a definitive cost evaluation.

We assume usage of NVIDIA A100 GPUs for their balance of performance, per simulation cost, and availability at the time of writing. Hardware refresh cycles are set at four years, and we compare the "Minimum" and "Baseload" on-premise setups from the previous chapter with cloud-based execution. A single A100 GPU is assumed to cost 18.000 €[2]. Using the Microsoft Azure Cloud list prices, cloud computing is assumed to cost 0,64 €/h [32]. For electricity cost, we assume power consumption of 400W, and electricity cost of 0,25 €/kWh.

System Setup	**On-Premise / Minimum (2 Nodes)**	**On-Premise / Baseload (3 Nodes)**	**Cloud / Spot Instance**
Investment (€)	36.000	54.000	-
Refresh Cycle (yrs)	4	4	-
Utilization (%)	100%	67%	-

[2] No official launch price is given for the A100. The number is an approximation based on different vendors' prices known to the authors.

Used Hours (p. yr)	17.520	17.520	17.520
Hardware Cost (€ p. yr)	**9.000**	**13.500**	/
Electricity Cost (€ p. yr)	**1.752**	**1.752**	/
Total Cost (€ p. yr)	**10.752**	**15.252**	**11.212**

Table 5: Comparison of cost associated with simulations in on-premise and cloud environments.

In this case, executing the workloads entirely on the cloud is at a cost level similar to the minimum on-premise setup. The key factor is the ability to leverage low-cost "spot" instances. The spot model is a special renting option cloud vendors typically offer to smoothen their data center utilization. These instances are provisioned on-demand but may be evicted during periods of high load. This eviction risk makes spot instances unsuitable for some classical workloads. Also, availability might be limited for hardware in high demand, as seen for the H100 in our case. However, they are attractive for CAE applications, where simulations can resume from checkpoints, due to their favorable pricing.

Other than neglecting important cost factors, such operations, maintenance and administration (see table 3), this analysis simplifies certain aspects. Key considerations that should be made additionally include:

1) **Procurement Differences** – On-premise infrastructure requires upfront investment, while cloud resources are rented on-demand, impacting financial models and accounting.
2) **Time-to-Market Impact** – Hardware procurement, internal decision-making, installation, and financing processes introduce significant delays, whereas cloud resources are immediately available.
3) **Hardware Refresh Cycles** – On-premise systems cannot benefit from mid-cycle hardware advancements. In contrast, cloud users can seamlessly upgrade to new hardware (e.g., moving from A100 to H100 offers a 1.7x speedup).
4) **Vendor Asset Management** – Cloud providers manage infrastructure, reducing IT administration overhead and taking over certain liabilities, and data security responsibilities.

8. Summary and Outlook

This study demonstrates that integrating modern CAE technologies, such as GPU acceleration, cloud parallelization and SPH can significantly enhance the efficiency of CFD simulations for differential system design.

The SPH method is an established and validated tool in drivetrain design because of its capability to effectively model multiphase flows with complex moving geometries. It enables rapid evaluation of the impact of different gear and housing designs on efficiency and system durability across a wide range of operating conditions without extensive manual preprocessing.

Our simulation results indicate that GPU acceleration can achieve speedups between 6x and 32x compared to CPU architectures. Furthermore, cloud-based parallel execution reduces project turnaround time in peak load scenarios by around 5x compared to a realistic on-premise alternative. Combined, these improvements yield a total speedup of 30x or greater. Furthermore, cost comparisons reveal that cloud solutions can be as economical as even the smallest on-premise setups with prices for hardware and electricity being in a similar range.

The implications for the users and decision-makers in the industry are significant. Advanced simulation tools support meeting evolving regulatory and market demands. Shortened lead times enable more iterative testing and faster decision-making, fostering a more agile R&D process. Moreover, the increased accessibility of high-performance computational resources contributes to a democratization of simulation capabilities, encouraging broader adoption across design teams.

Looking ahead, further advancements in SPH algorithms—targeting enhanced accuracy and efficiency—are anticipated. Next-generation GPU hardware, with improvements in power efficiency and computational, will drive further performance gains.

In summary, the integration of SPH with GPU and cloud technologies offers clear benefits in simulation speed, cost, and scalability. These advancements not only accelerate the design cycle for differential systems but also facilitate more complex, high-fidelity simulations across the automotive sector and beyond.

9. References

[1] R. A. Gingold and J. J. Monaghan, “Smoothed particle hydrodynamics: theory and application to non-spherical stars,” *Monthly Notices of the Royal Astronomical Society,* vol. 181, no. 3, pp. 375-389, 1977.

[2] D. Violeau, Fluid Mechanics and the SPH Method: Theory and Applications, Oxford University Press, 2012.

[3] D. C. Kunik and D. J. Kunert, “Wetting and oil flow analysis of planetary gearboxes using oil flow simulations,” in *International SPHERIC Workshop*, Berlin, 2024.

[4] G. A. Mensah, L. Braun, S. Krishna, P. Sabrowski and T. B. Wybranietz, “Industry-relevant validation cases for benchmarking SPH codes,” in *International SPHERIC Workshop*, Catania, 2022.

[5] P. Sabrowski, L. Beck and T. Wybranietz, "Modern WCSPH in industrial multiphase application considering complex moving boundaries," in *Proceedings of the 14th SPHERIC International Workshop*, Exeter, 2019.

[6] D. Violeau, Fluid Mechanics and the SPH Method: Theory and Applications, Oxford University Press, 2012.

[7] J. J. Monaghan, “Simulating Free Surface Flows with SPH,” *Journal Of Computational Physics ,* vol. 110, pp. 399-406, 1994.

[8] R. H. Cole, Underwater Explosions, Princeton, New Jersey: Princeton University Press, 1948.

[9] S. Adami, “A generalized wall boundary condition for smoothed particle hydrodynamics,” *Journal of Computational Physics,* vol. 231.21, pp. 7057-7075, 2012.

[10] S. Kulasegaram, J. Bonet, R. W. Lewis and M. Profit, “A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications,” *Computational Mechanics,* vol. 33.4, p. 316–325, 2004.

[11] J. Feldman and J. Bonet, “Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems,” *International Journal for Numerical Methods in Engineering,* vol. 72.3, p. 295–324, 2007.

[12] C. Kassiotis, D. Violeau and M. Ferrand, “Semi-Analytical Conditions for Open Boundaries in Smoothed Particle Hydrodynamics,” in *Proceedings of the 8th International SPHERIC Workshop*, Trondheim, 2013.

[13] A. Vergnaud, “Améliorations de la précision et de la modélisation de la tension de surface au sein de la méthode SPH, et simulations de cas d'amerrissage d'urgence d'helicoptères,” Ph.D. dissertation, École centrale de Nantes, Nantes, 2022.

[14] C. Bierwisch, “Consistent Thermo-Capillarity and Thermal Boundary Conditions for Single-Phase Smoothed Particle Hydrodynamics,” *Materials,* vol. 14:4530, 2021.

[15] B. Legrady, “Particle-Based CFD Study of Lubrication in Power Transmission Systems Using Local Refinement Techniques,” *Power Transmission Engineering,* vol. 2, pp. 36-49, 2024.

[16] C. Zhang, M. Rezavand and X. Hu, “Dual-criteria time stepping for weakly compressible smoothed particle hydrodynamics,” *Journal of Computational Physics,* vol. 404, pp. 109-135, 2020.

[17] T. Wybranietz, "Numerical modelling of free surface flows interacting with porous media: Development of an SPH-FEM coupling approach," Ph.D. dissertation, Technical University Berlin, Berlin, 2024.

[18] D. Molteni and A. Colagrossi, “A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH,” *Computer Physics Communications,* vol. 180.6, p. 861–872, 2009.

[19] S. J. Lind, R. Xu, P. K. Stansby and B. D. Rogers, “Incompressible smoothed particle hydrodynamics for free-surface fows: A generalised difusion-based algorithm for stability and validations for impulsive fows and propagating waves.,” *Journal of Computational Physics,* vol. 231, p. 1499–1523, 2012.

[20] P. Sun, A. Colagrossi, S. Marrone and A. M. Zhang, “The δplus-SPH model: Simple procedures for a further improvement of the SPH scheme,” *Computer Methods in Applied Mechanics and Engineering,* vol. 315, p. 25–49, 2017.

[21] F. Pause, P. Koob and K. Juckelandt, “Potenziale der Smoothed Particle Hydrodynamics Methode für die mehrphasige Simulation von Wälzlagerströmungen,” in *NAFEMS Virtual DACH Conference*, 2020.

[22] K. Häberle and B. Legrady, “Ensure IP splash- and jet-water protection with highly automated SPH simulation,” in *International SPHERIC Workshop*, Berlin, 2024.

[23] A. Bauer and G. Marcus, “Investigation of oil shielding plates using SPH,” in *International SPHERIC Workshop*, Berlin, 2024.

[24] K. Juckelandt, L. Kandari and O. Graf-Goller, “Efficient Simulation of Oil Lubrication in Rolling Bearings,” in *International SPHERIC Workshop*, Berlin, 2024.

[25] M. Gebhardt, A. Rhode and B. Legrady, “Lubrication improvement at the HS-IS spline shaft interface of a wind turbine gearbox using the smooth particle hydrodynamic method,” in *International Conference on Gears*, 2023.

[26] G. A. Mensah, P. Sabrowski and T. B. Wybranietz, “Practical guidelines on modelling electric engine cooling with SPH,” in *International SPHERIC Workshop*, Rhodes, 2023.

[27] H. Liu, T. Jurkschat, T. Lohner and K. Stahl, “Detailed Investigations on the Oil Flow in Dip-Lubricated Gearboxes by the Finite Volume CFD Method,” *Lubricants,* vol. 6, 2028.

[28] T. Davin, S. Harmand and J. Pelle, “Experimental study of oil cooling systems for electric motors,” *Applied Thermal Engineering 75,* 2015.

[29] F. Pause, M. Fischer and G. Mensah, “Simulation of Electric Engine Oil Cooling with Smoothed Particle Hydrodynamics (SPH),” in *NAFEMS DACH*, 2024.

[30] A. Hérault, G. Bilotta and R. A. Dalrymple, “SPH on GPU with CUDA,” *Journal of Hydraulic Research,* vol. 48, no. 1, pp. 74-79, 2009.

[31] A. Cavelan, R. M. Cabezón, M. Grabarczyk and F. M. Ciorba, “A Smoothed Particle Hydrodynamics Mini-App for Exascale,” in *PASC '20: Proceedings of the Platform for Advanced Scientific Computing Conference*, Geneva, 2020.

[32] Microsoft, “Linux Virtual Machines Pricing,” [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/#pricing. [Accessed 10 February 2025].

[33] B. Legrady, “Efficient Numerical Assessment of Thermal Effects in a Gearbox Using Smoothed Particle Hydrodynamics,” *AGMA Technical Paper 24FTM25,* 2024.

[34] SPHERIC, “SPHERIC Grand Challenge Working Group,” [Online]. Available: https://www.spheric-sph.org/grand-challenges. [Accessed 10.02.2025 February 2025].