A systematic study on Ansys Forming® performance

Dr.-Ing Kang Shen[1]

[1]Ansys Germany GmbH

1 Abstract

Mesh adaptivity refines the blank mesh as needed in stamping simulations. Users do not need to anticipate where a dense mesh will be required. Despite its universal use, it demands significant effort due to serialization and the need to carry a dense mesh through subsequent iterations. In-Core adaptivity and Mesh fusion assist the solver in conserving effort, thereby enhancing performance. This paper will demonstrate best practices for utilizing In-Core adaptivity and Mesh fusion in Ansys Forming through practical cases. In addition, for different model, we should find an optimum number of CPUs to run the job. Beyond this number, the scalability will not see any obvious improvements.

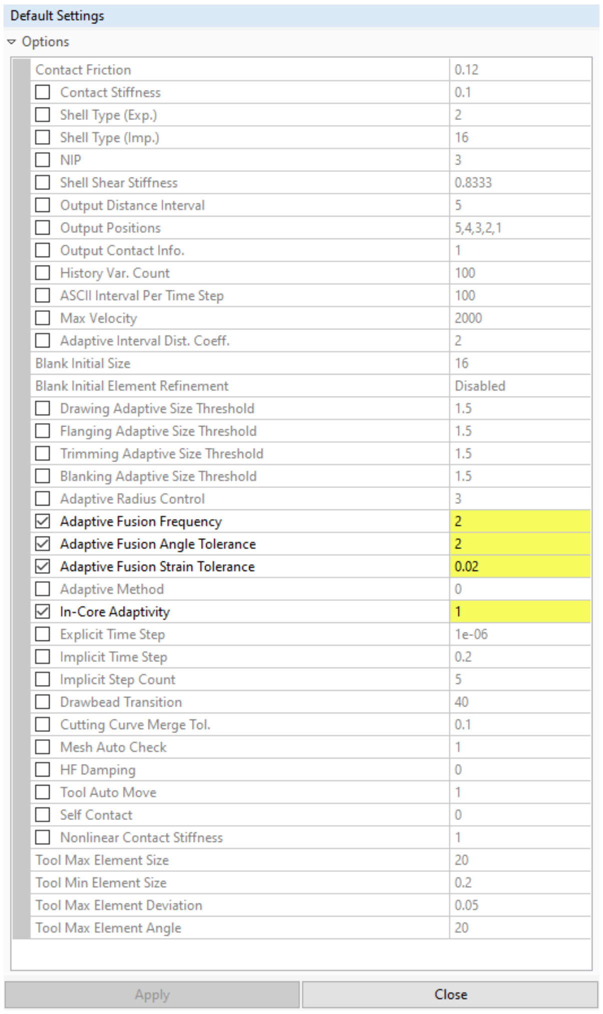
2 Background

In-core adaptivity aims to dynamically create new nodes and elements during the solution loop, avoiding the significant disk I/O time and serialization that limit scalability. This is achieved by moving data from a single large array to dynamically allocated arrays, allowing their size to change while the problem runs. Implementing this requires substantial modifications, as every data array dependent on the number of nodes must be moved and referenced differently [1, 2].

To capture detailed physical behaviors near drastic changes, a finer finite element mesh is required. This is achieved by in-core adaptivity. However, using a finer FEA mesh for the entire model is sometimes impractical due to the long computation time and large computational resources needed. Incremental metal forming is a slow process and simulating it can take many hours with finer mesh. Improved methods for time-marching simulations are needed to obtain accurate numerical physical behaviors of sheet metal during forming processes with predefined load paths. Mesh fusion helps in this case by combining low sensitive finer element together. It will further improve the solver performance [3, 4].

3 Option in Ansys Forming

To use in-core adaptivity and mesh fusion, the options need to be defined before the simulation started. Adaptive Fusion Frequency, Angle Tolerance, Strain tolerance need to be set for fusion. In-Core Adaptivity option needs to be activated.



Contact Friction	0.12
☐ Contact Stiffness	0.1
☐ Shell Type (Exp.)	2
☐ Shell Type (Imp.)	16
☐ NIP	3
☐ Shell Shear Stiffness	0.8333
☐ Output Distance Interval	5
☐ Output Positions	5,4,3,2,1
☐ Output Contact Info.	1
☐ History Var. Count	100
☐ ASCII Interval Per Time Step	100
☐ Max Velocity	2000
☐ Adaptive Interval Dist. Coeff.	2
Blank Initial Size	16
Blank Initial Element Refinement	Disabled
☐ Drawing Adaptive Size Threshold	1.5
☐ Flanging Adaptive Size Threshold	1.5
☐ Trimming Adaptive Size Threshold	1.5
☐ Blanking Adaptive Size Threshold	1.5
☐ Adaptive Radius Control	3
☑ Adaptive Fusion Frequency	2
☑ Adaptive Fusion Angle Tolerance	2
☑ Adaptive Fusion Strain Tolerance	0.02
☐ Adaptive Method	0
☑ In-Core Adaptivity	1
☐ Explicit Time Step	1e-06
☐ Implicit Time Step	0.2
☐ Implicit Step Count	5
☐ Drawbead Transition	40
☐ Cutting Curve Merge Tol.	0.1
☐ Mesh Auto Check	1
☐ HF Damping	0
☐ Tool Auto Move	1
☐ Self Contact	0
☐ Nonlinear Contact Stiffness	1
Tool Max Element Size	20
Tool Min Element Size	0.2
Tool Max Element Deviation	0.05
Tool Max Element Angle	20

4 Test cases and results

In this paper, 6 cases are tested. The first four cases are s-rail, bipolar plate, water sink and car hood forming. The last two cases are submitted by industrial partner (due to the confidentiality, only performance study will be shown).

Following table provides the information about the forming operations in each test cases, including operation plan (D for drawing, T for trimming, F for flanging), element number at the end of simulation and testing plan.

	Operation plan	Elem. Num.	Testing plan	
s-rail	D	26,509	2, 4, 8, 16 CPUs	In-core & fusion
bipolar plate	D	56,180	2, 4, 8, 16 CPUs	In-core
water sink	D-T-T-F-F	93,750	2, 4, 8, 16 CPUs	In-core
car hood	D-T-T-F	147,122	2, 4, 8, 16 CPUs	In-core & fusion
Industrial case 1	D-T	1,359,128	28, 56 CPUs	In-core
Industrial case 2	F-F-T-F-T	74,196	28, 56 CPUs	In-core

For all test, LS-DYNA solver with MPP single precision is used. The operations system is Linux CentOS and Intel MPI is chosen for this case.

S-rail is the classic case with Ansys Forming. It has a coarse blank mesh initially and undergoes many mesh adaptivity steps. Increasing the CPU count from 2 to 16 accelerates the simulation by 3.6 times. In-core adaptivity reduces another 37% with 16 CPUs. In this case, mesh fusion is added to the in-core adaptivity. Mesh fusion decreases the number of elements on the side wall and saves an additional 4.5% of time.

A bipolar plate is a crucial component of fuel cells and is often requested by fuel cell manufacturers. Because it often contains fine features in the tool, a fine mesh is used to start the stamping simulation, and a low number of adaptivity steps is expected. In this test, a speed-up of 2.1 times is shown when 16 CPUs are used. The in-core adaptivity requires more time than a normal run because it takes CPU capability to run adaptivity, which is not needed in this case. If a similar case like the bipolar plate is used, it is recommended not to use in-core adaptivity for better performance.

The water sink case contains more operations than the first two cases. As shown, the trimming operation does not show a significant difference with in-core adaptivity and mesh fusion. The in-core adaptivity has more effect on the Drawing and Flanging operations, especially with 16 CPUs. With 2 CPUs, in-core only saves 2% of computation time. With 16 CPUs, it saves 24.6% of time in the first drawing operation. The two flanging operations show the same tendency.

The car hood case has the largest number of elements among the first four cases. It is very challenging for in-core adaptivity with a low CPU count. In this case, only when the number of CPUs reaches 16 does it save 17.4% of time with in-core. Using 16 cores makes more sense compared to a lower core count like 2 CPUs, as it saves 4.5 times the effort.

Industrial cases 1 and 2 come from a tool maker. Case 1 was tested with 28 CPUs because it was in the sweet spot of performance and the number of CPUs used. As shown, when doubling the CPU count, it only saved 5.5% of time. However, it is a perfect case for in-core adaptivity because a high number of CPUs are available, and the adaptivity could be processed well in-core. With in-core, it saves about 39% of time with 56 CPUs. The same observation is shown for case 2.

5 Summary

In this paper, a systematic study is conducted with normal, in-core, and mesh fusion adaptivity. When the number of elements is larger than 100,000 and the available CPU count is fewer than 16, normal mesh adaptivity performs better. In-core adaptivity shows better performance when the number of elements is low (<50,000) or when more CPUs (>16) are available. Mesh fusion contributes to performance based on in-core adaptivity.

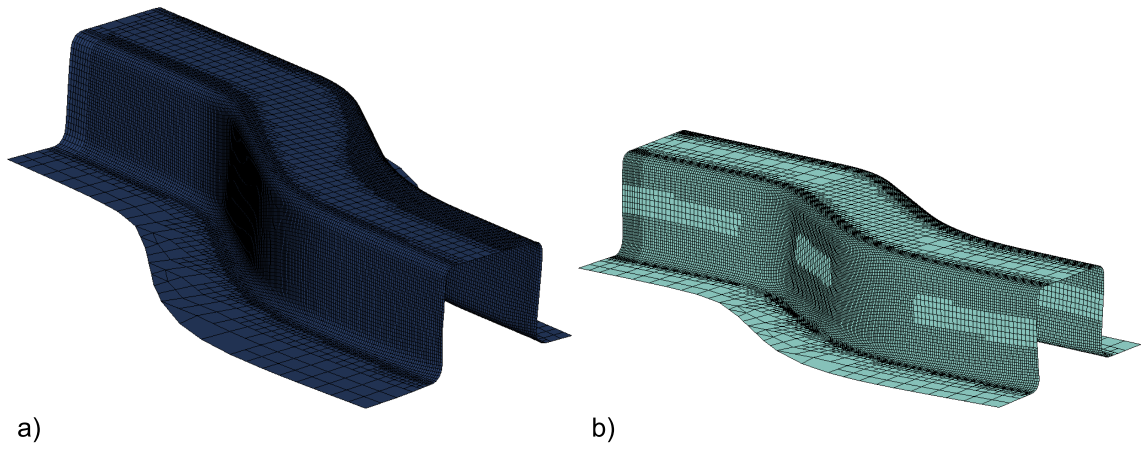


Figure 1 S-rail case a) with adaptivity, b) with mesh fusion

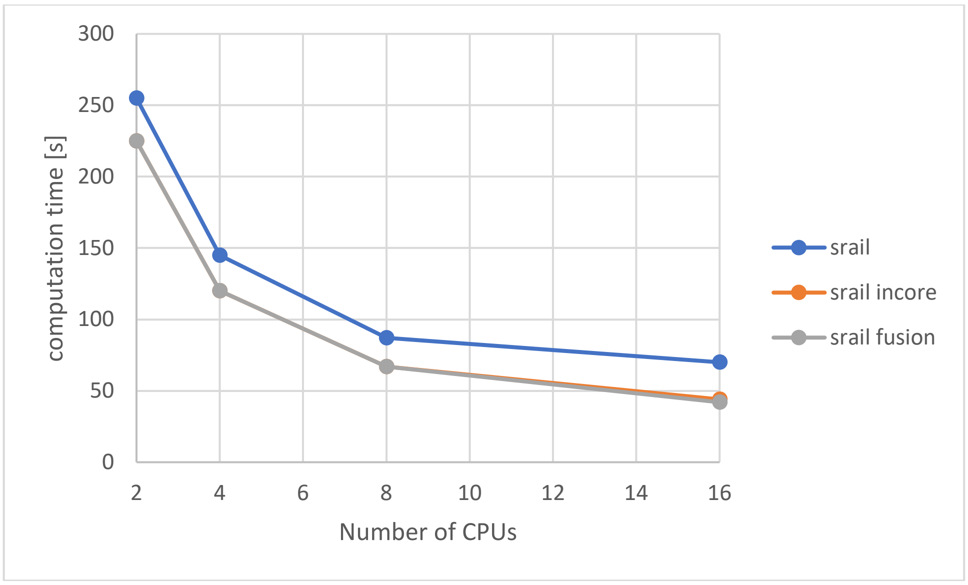


Figure 2 performance of s-rail case

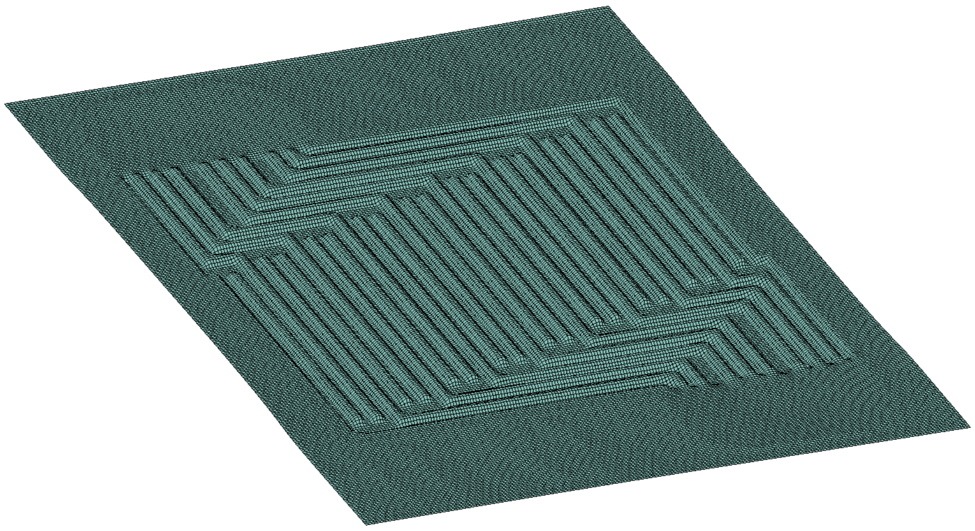


Figure 3 Bipolar plate case

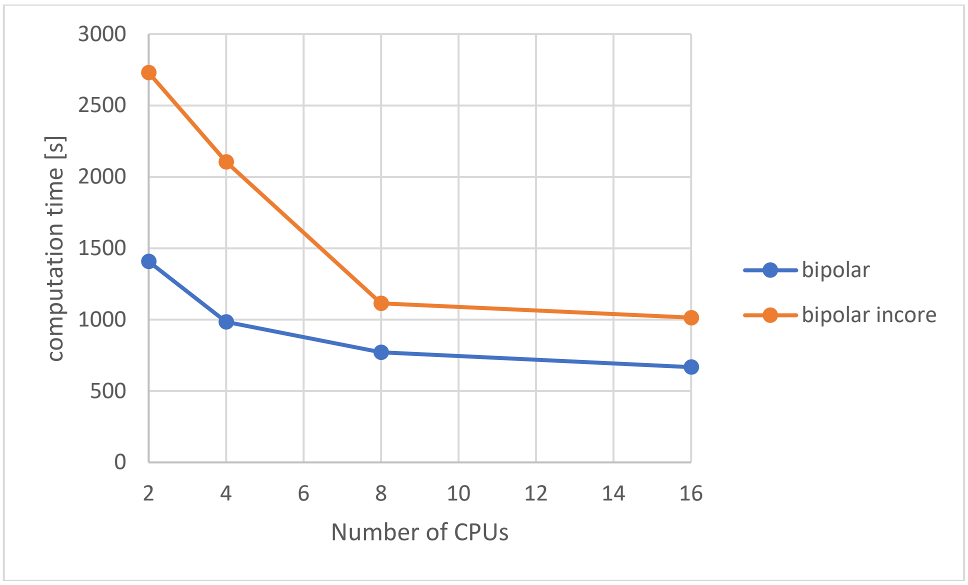


Figure 4 performance of Bipolar plate

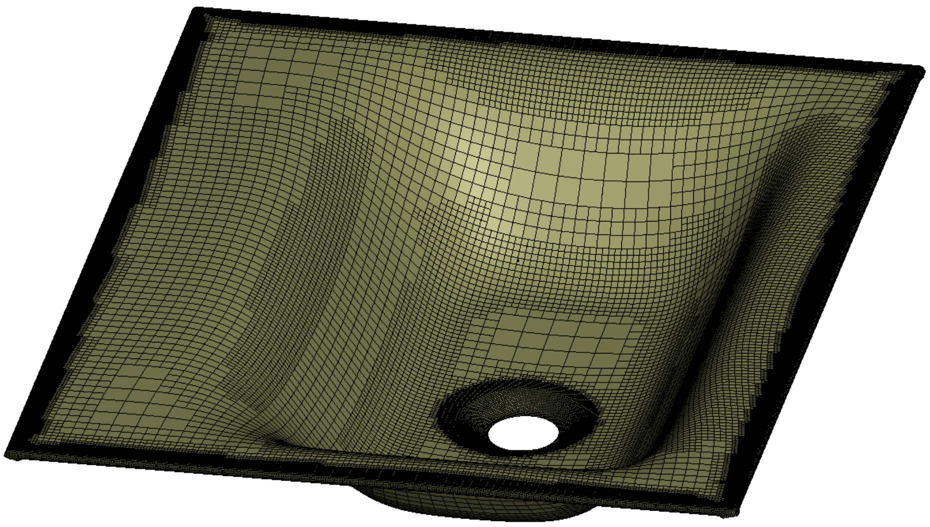
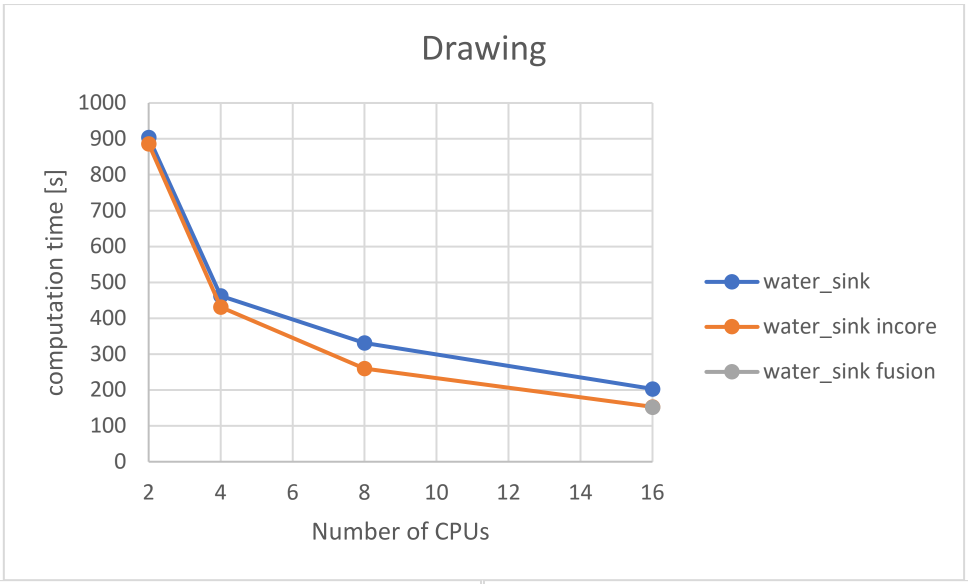
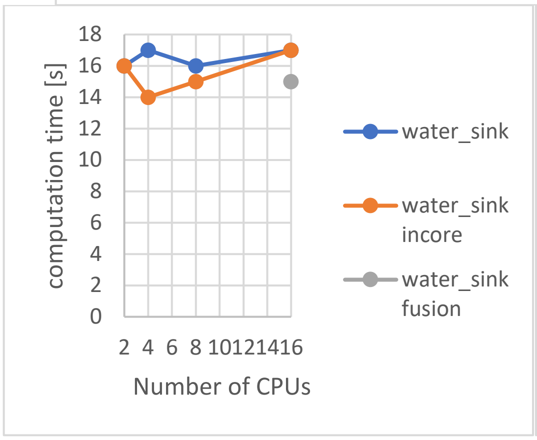
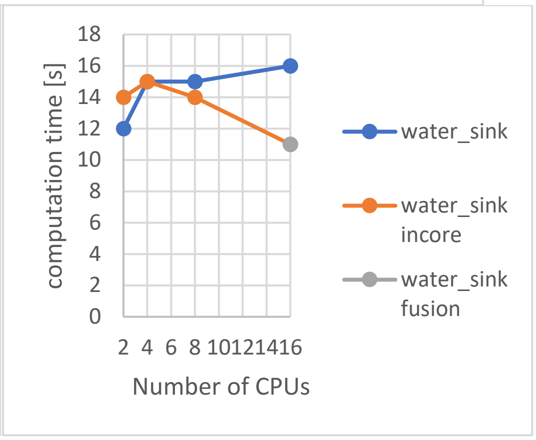


Figure 5 Water sink case







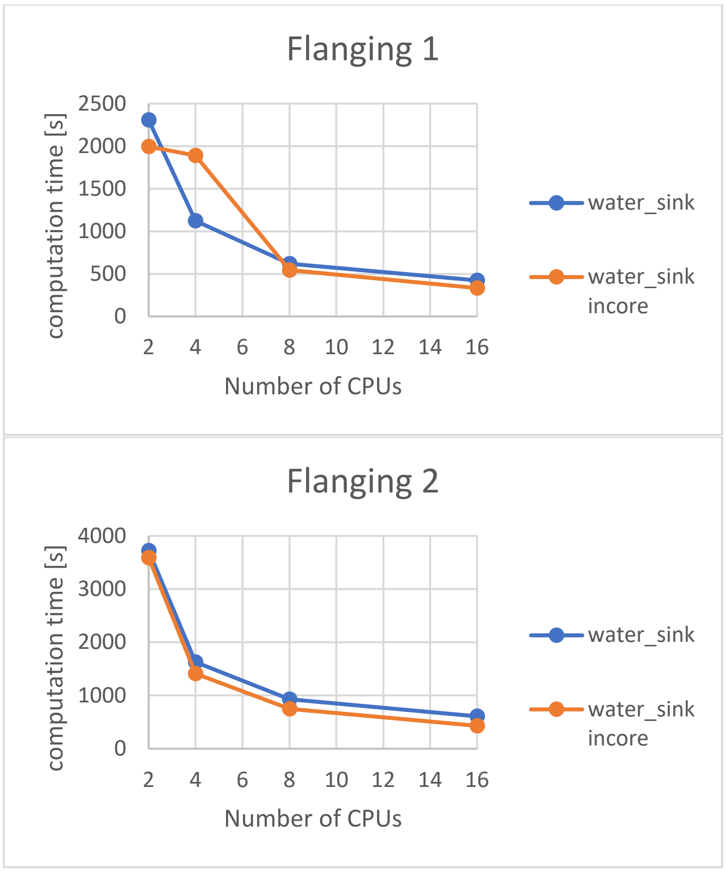


Figure 6 performance of Water sink case

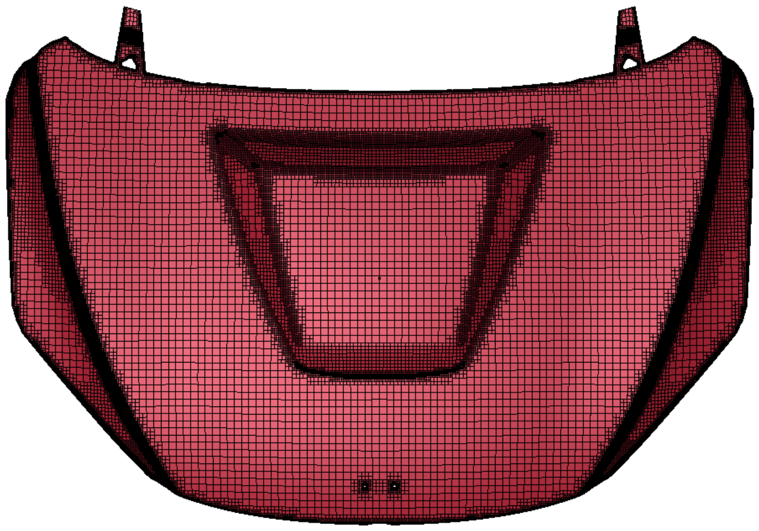


Figure 7 car hood case

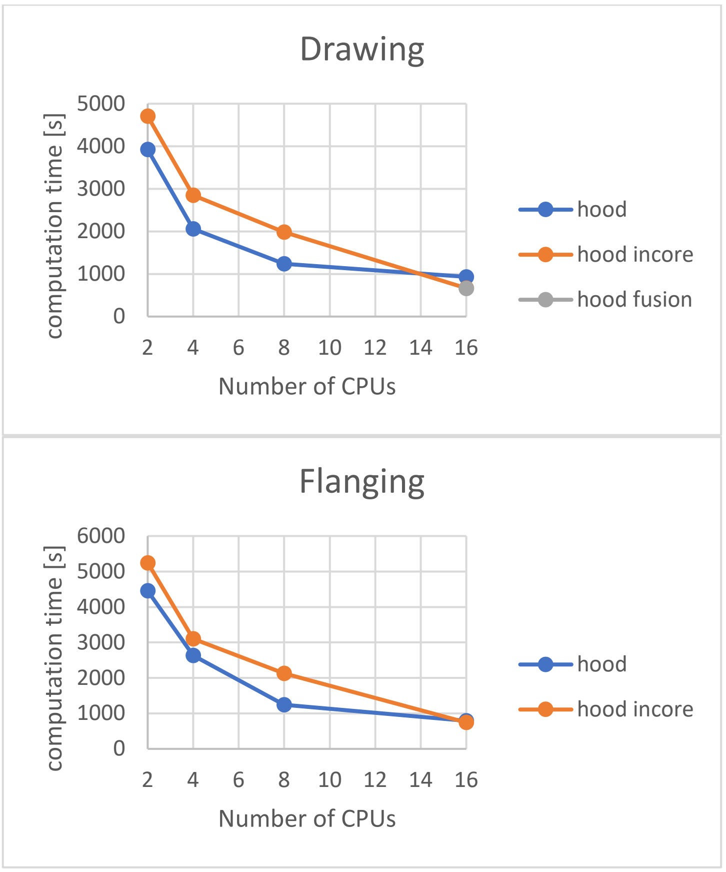


Figure 8 performance of car hood case

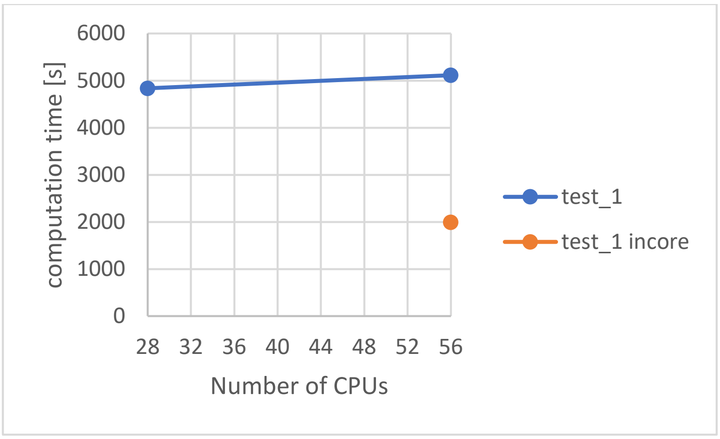


Figure 9 performance of industrial case 1

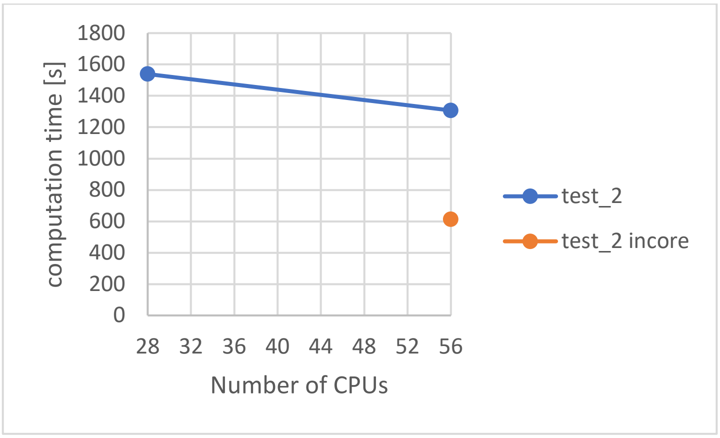


Figure 10 performance of industrial case 2

6 Literature

[1] Brian Wainscott and Houfu Fan, LSTC, In Core Adaptivity, 15th international LS-DYNA User Conference
[2] Houfu Fan, Brian Wainscott, Li Zhang and Xinhai Zhu, Performance Study of In Core Adaptivity in LS-DYNA, 16th international LS-DYNA User Conference
[3] Houfu Fan, Li Zhang, Xinhai Zhu and Yuzhong Xiao, Improvement of Mesh Fusion in LS-DYNA, 15th international LS-DYNA User Conference
[4] Xinhai Zhu, Houfu Fan, Li Zhang, Yuzhong Xiao and Ninshu Ma, Tube Adaptivity for Mesh Fission/Fusion in LS-DYNA, 15th international LS-DYNA User Conference