Effects of Mesh Loop Modes on Performance of Unstructured Finite Volume GPU Simulations

Yue Weng
Sun Yat-Sen University
Xi Zhang (zhangx299@mail.sysu.edu.cn)
Sun Yat-Sen University https://orcid.org/0000-0002-1196-0176
Xiaohu Guo
Hartree Center
Xianwei Zhang
Sun Yat-Sen University
Yutong Lu
Sun Yat-Sen University
Yang Liu
Sun Yat-Sen University

Research

Keywords: GPU, CFD, finite volume, unstructured mesh, mesh loop modes, data locality, data dependence

Posted Date: April 20th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-408298/v1

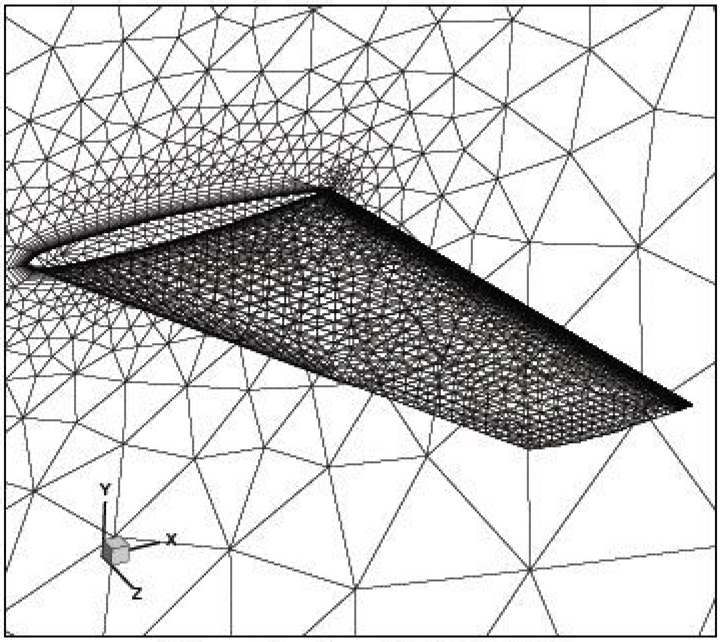
Version of Record: A version of this preprint was published at Advances in Aerodynamics on July 22nd, 2021. See the published version at https://doi.org/10.1186/s42774-021-00073-y.

Abstract

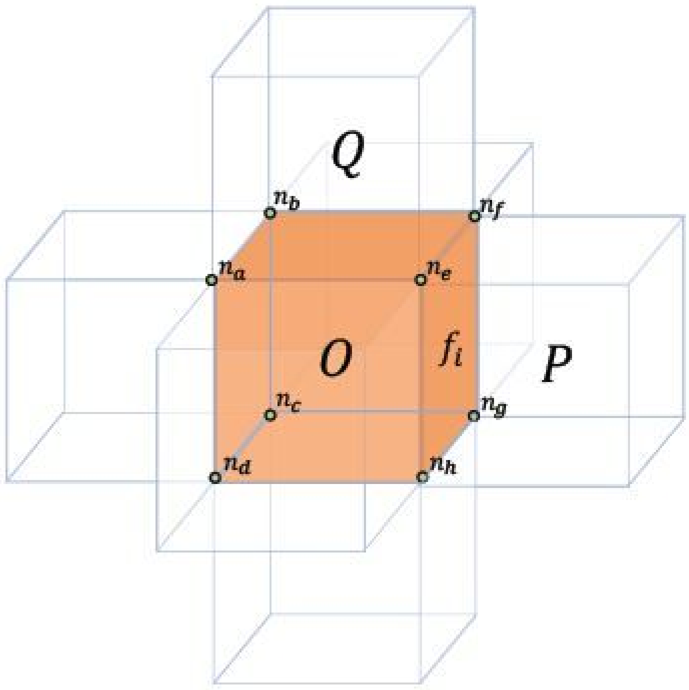
In unstructured finite volume method, loop on different mesh components such as cells, faces, nodes, etc is used widely for the traversal of data. Mesh loop results in direct or indirect data access that affects data locality significantly. By loop on mesh, many threads accessing the same data lead to data dependence. Both data locality and data dependence play an important part in the performance of GPU simulations. For optimizing a GPU-accelerated unstructured finite volume Computational Fluid Dynamics (CFD) program, the performance of hot spots under different loops on cells, faces, and nodes is evaluated on Nvidia Tesla V100 and K80. Numerical tests under different mesh scales show that the effects of mesh loop modes are different on data locality and data dependence. Specifically, face loop makes the best data locality, so long as access to face data exists in kernels. Cell loop brings the smallest overheads due to non-coalescing data access, when both cell and node data are used in computing without face data. Cell loop owns the best performance in the condition that only indirect access of cell data exists in kernels. Atomic operations reduced the performance of kernels largely in K80, which is not obvious on V100. With the suitable mesh loop mode in all kernels, the overall performance of GPU simulations can be increased by 15%-20%. Finally, the program on a single GPU V100 can achieve 4.8 speed up comparing with 28 MPI tasks on two Intel CPUs Xeon Gold 6132.

Full Text

Figures



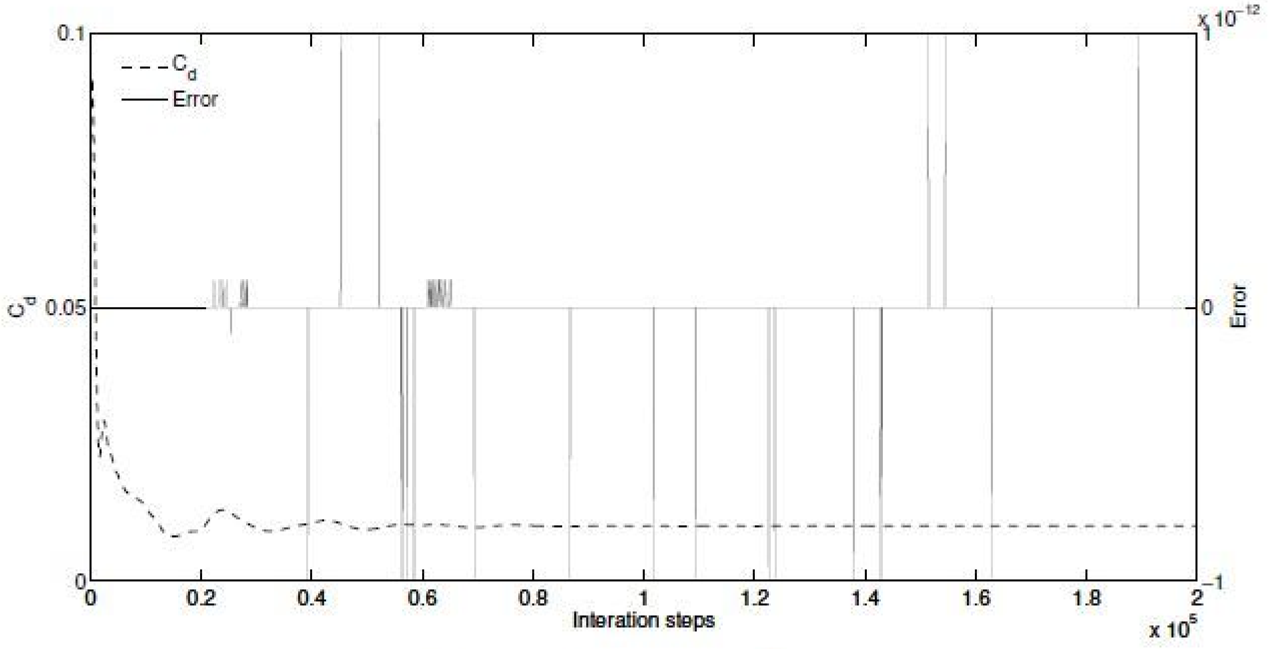
(a) mesh of an ONERA M6



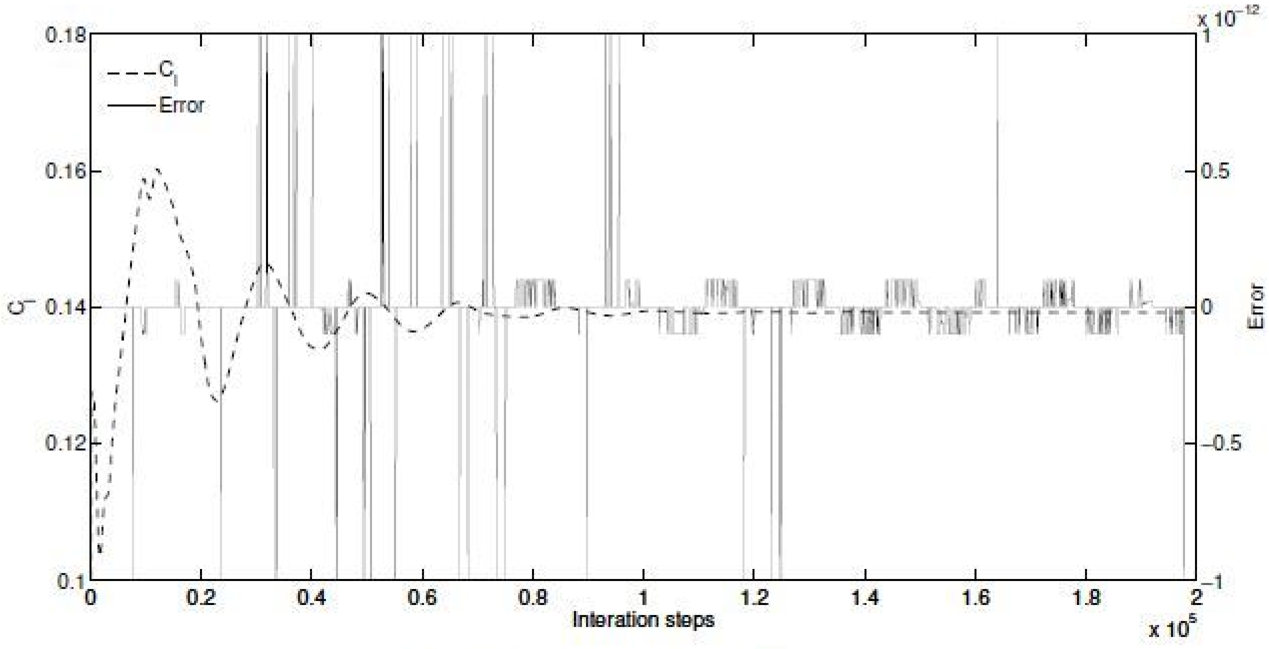
(b) cells, faces, and nodes

Figure 1

Cells, faces, nodes of mesh in cell-centered FV discretization



(a) Drag coefficient and Error



(b) Lift coefficient and Error

Figure 2

Error between CPU and GPU results

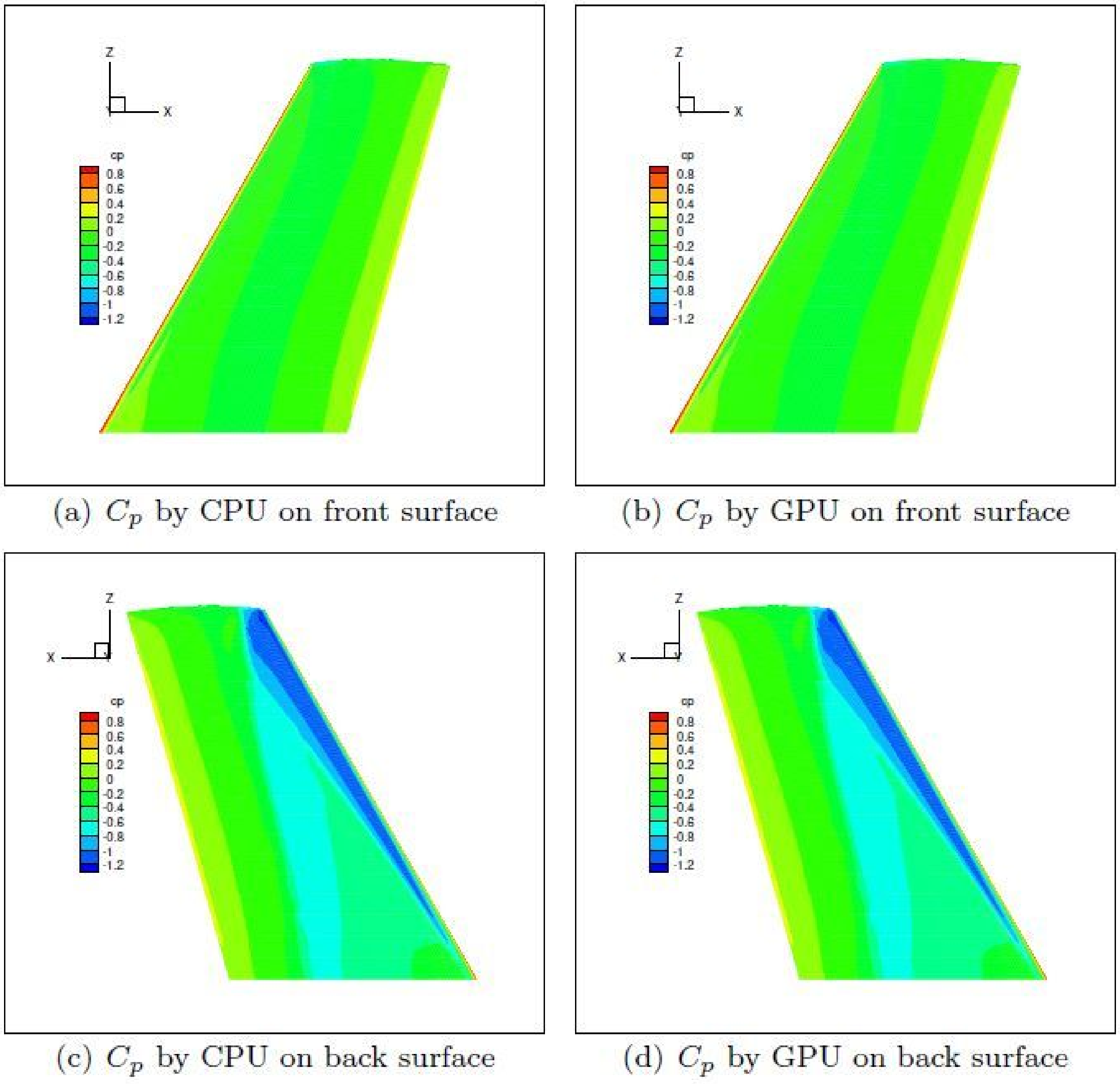
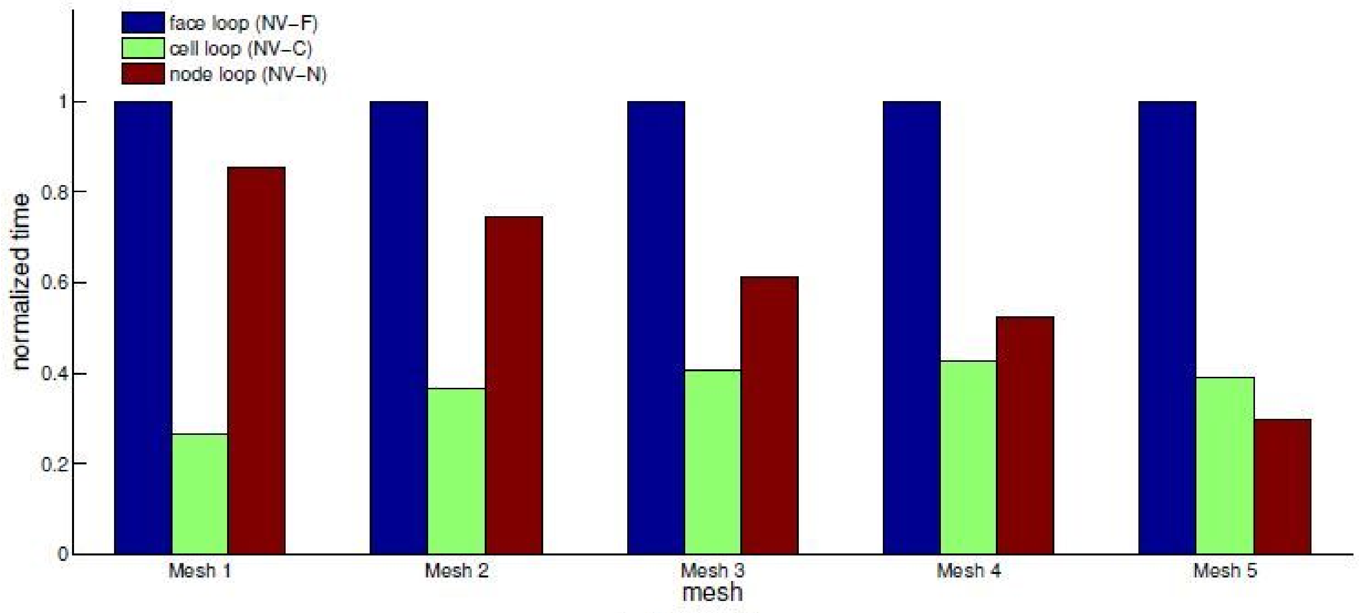
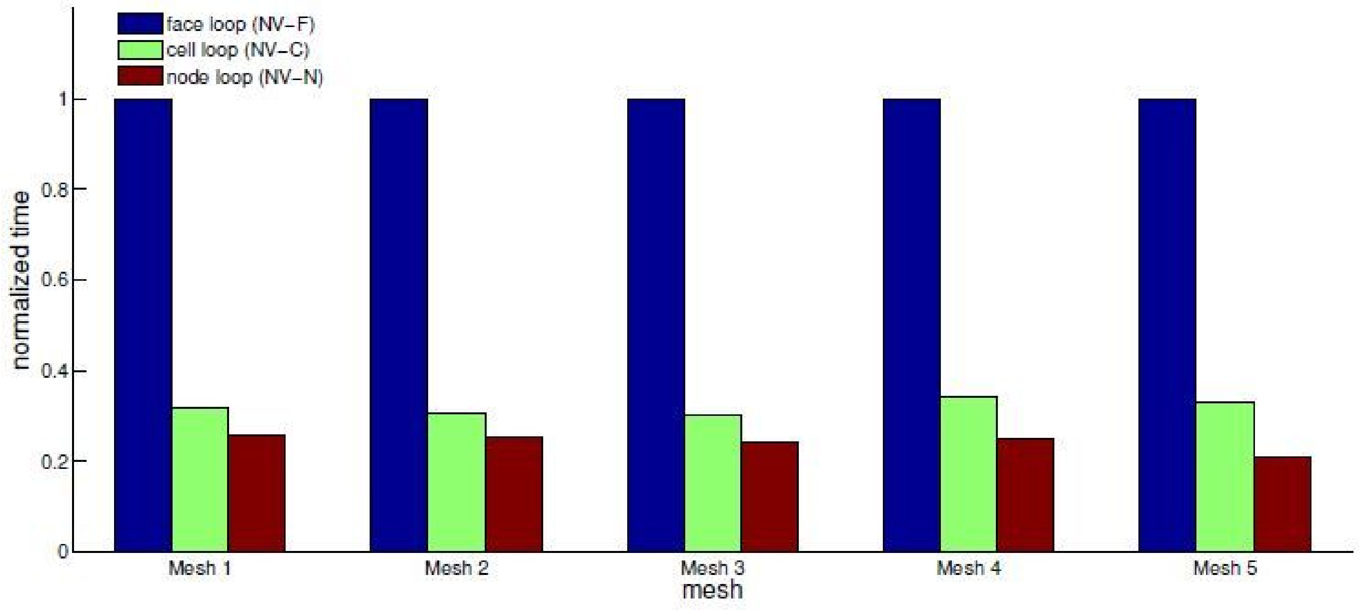


Figure 3

Comparison of pressure coeficient between CPU and GPU



(a) V100



(b) K80

Figure 4

Performance of GPU kernels for interpolating q by face loop, cell loop, and node loop

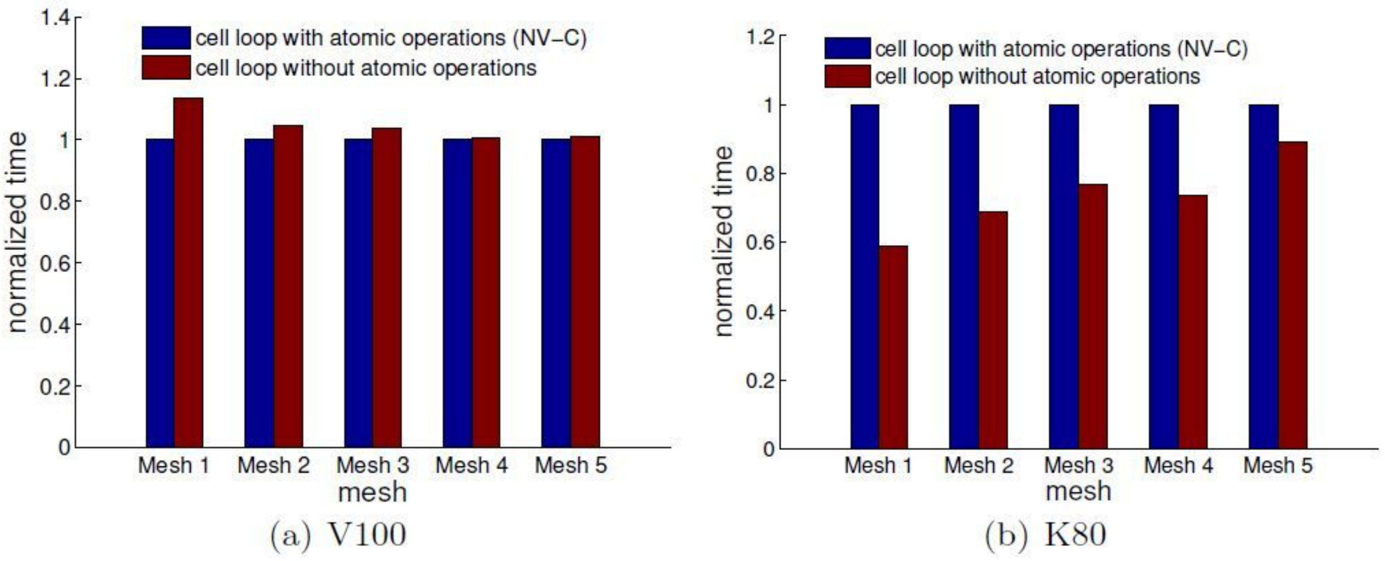
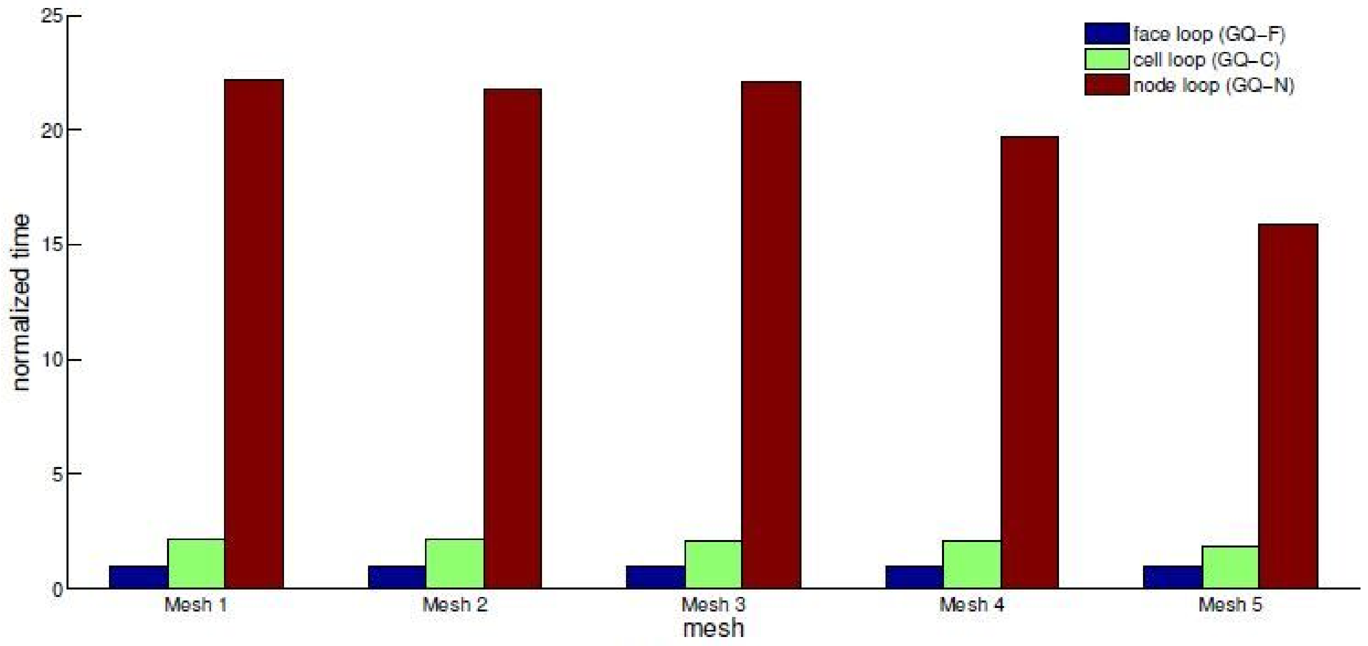
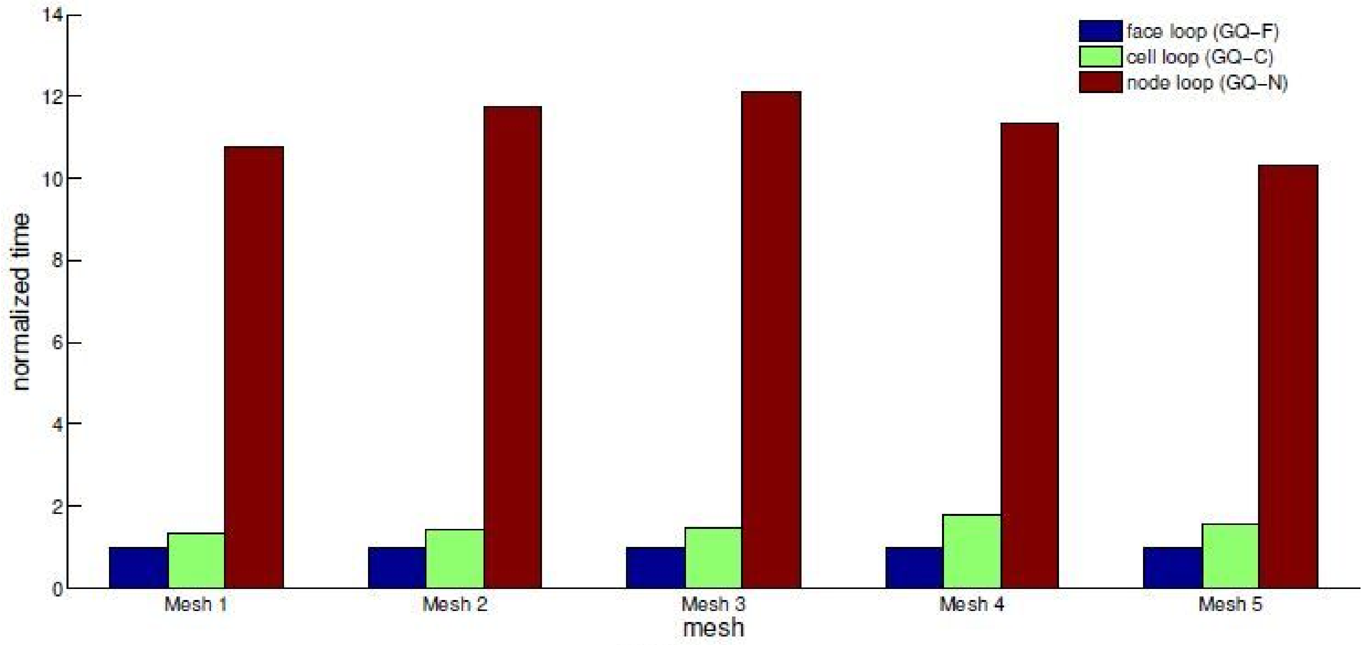


Figure 5

Performance comparison of cell loop with and without atomic operations



(a) V100



(b) K80

Figure 6

Performance of GPU kernels for gradient of q by face loop, cell loop, and node loop

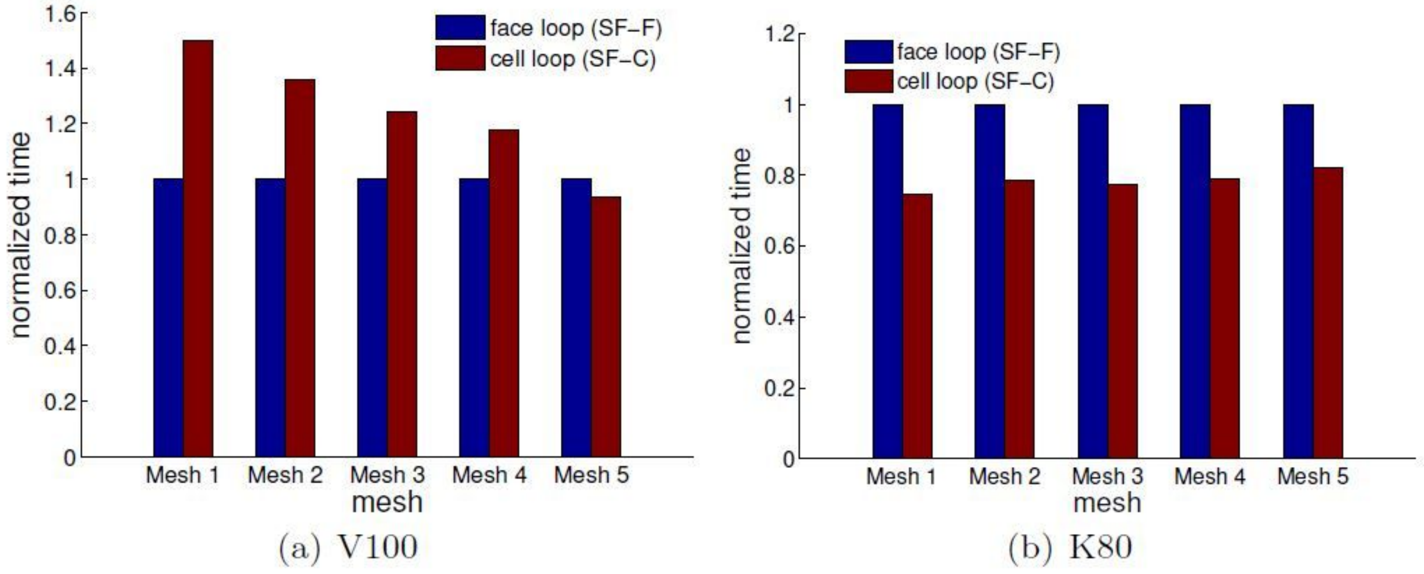


Figure 7

Performance of GPU kernels for summation of flux by face loop and cell loop

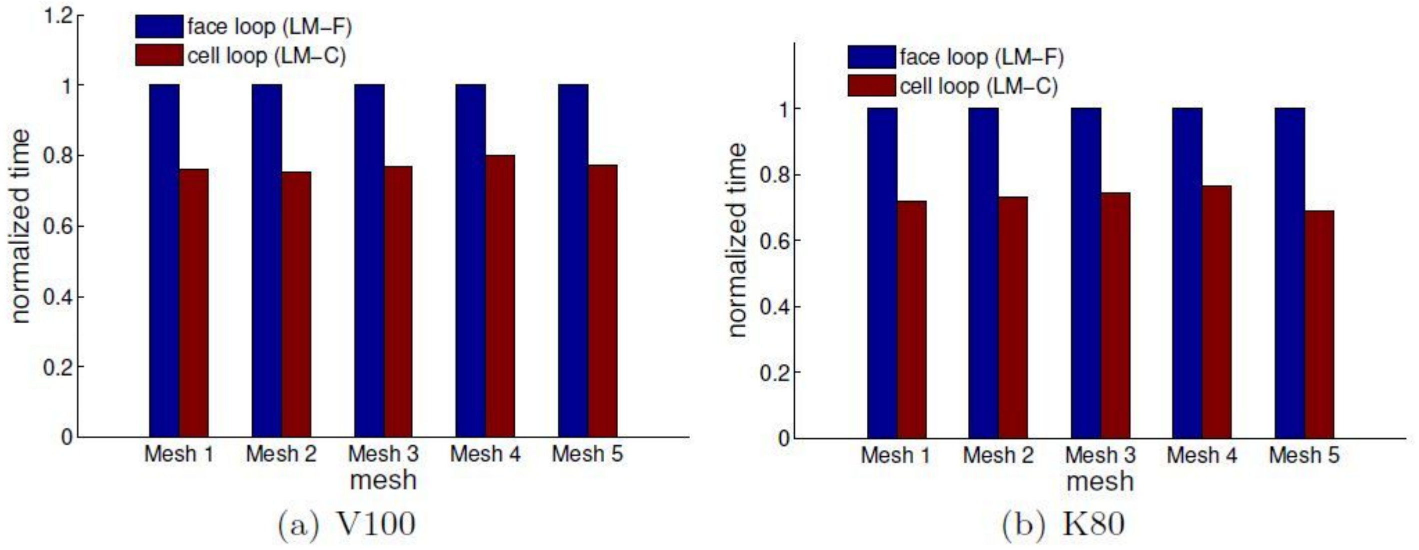


Figure 8

Performance of GPU kernels for determining local maximum & minimum pressure by face loop and cell loop