HIGH-PERFORMANCE IMPLEMENTATION OF MATRIX-FREE RUNGE-KUTTA DISCONTINUOUS GALERKIN METHOD FOR EULER EQUATIONS BASED ON OPENFOAM

LIYANG XU1*, YONGQUAN FENG1, SHUAI YE1, YANZHANG CHEN2, YUNRUI GUO1, XIAOGUANG REN3, XINHAI XU3

[1] *State key Labratory of High Performance Computing, NUDT, Changsha, China*

[2] *Advanced Institute of Engineering Science for Intelligent Manufacturing, Guangzhou University, China*

[3] *National Innovation Institute of Defense Technology*

**xuliyang08@nudt.edu.cn*

Keywords: *HopeFOAM, RKDG, Euler Equations, Sum-factorization, SIMD Vectorization*

1. Introduction

The high-order methods have received considerable attention from research communities during the past several decades in that they offer the potential to significantly improve solution accuracy and efficiency[1, 2]. The Discontinuous Galerkin(DG) method is one of the high-order finite element methods using completely discontinuous basis functions[3]. It is locally conservative and stable, and can easily handle complex geometries and irregular meshes. These properties bring it into the mainstream of the CFD and prompt its application to a wide variety of problems. Our research group have developed DG discretization framework on HopeFOAM[4] which is based on OpenFOAM-4.0[5].

[6] has pointed that relying on the coefficient matrix limits the efficiency of high-order finite element discretization and the combination of the matrix-free and the Continuous Galerkin(CG) method can effectively improve the performance of its numerical simulation. Recently, the idea of matrix-free has been used to improve the performance of the DG implicit numerical simulations and achieved significant optimization[7]. However, DG explicit numerical simulations have not received proper attention that they deserve. Our research group have combined the matrix-free with Runge-Kutta Discontinuous Galerkin(RKDG) method[8] on HopeFOAM and dramatically speed-up the numerical simulation of Euler equations.

2. Major Work

This paper focuses on the matrix-free RKDG implementation for the compressible Euler equations on HopeFOAM. The inviscid compressible Euler equations in conservative form can be written in Eqn.1

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0 \tag{1}$$

The RKDG numerical solution of the above equations consists of two parts: temporal discretization and spatial discretization. For temporal discretization, we use Runge-Kutta explicit time stepping method and in this paper, we use 2nd order Runge-Kutta which can be detailed as Eqn.2, where L is the spatial discretization operator for $\left(-\nabla \cdot F(u)\right)$ in Eqn. 1

$$\begin{cases} K_1 = L\left(U^n\right) \\ K_2 = L\left(U^n + \Delta t K_1\right) \\ U^{n+1} = U^n + \Delta t \dfrac{\left(K_1 + K_2\right)}{2} \end{cases} \tag{2}$$

For spacial discretization, we use DG discretization and Eqn.3 can be get from Eqn.1.

$$M\frac{dU}{dt} + R\left(U\right) = 0 \tag{3}$$

The matrix-free implementation includes two stages. First, the data structures of two- and three-dimensional space construct from tensor products of one-dimensional objects. In this stage, there is no coefficient matrix, so both the arithmetic operations and the memory cost during numerical simulation decrease. In the second stage, we implement vectorization explicitly to improve the gain by Single Instruction Multiple Data(SIMD) instruction. As we all know, most modern CPU designs include SIMD instructions to improve performance and offers a third level of parallelism. However, SIMD vectorization is left to compilers to take optimization in most CFD simulation code, which brings the cost of data alignment and decreases the gain of vectorization. Therefore, we design and implement the data SIMD vectorization to make the most of CPU's multiple processing elements.

3. Results and Conclusions

We verify the correctness and efficiency of matrix-free implementation on Euler equations with several benchmark test cases. First, we consider a two-dimensional isentropic vortex whose exact solution given by Eqn.4.

$$\begin{cases} u = 1 - \beta e^{\left(1-r^2\right)} \dfrac{y - y_0}{2\pi} \\ v = \beta e^{\left(1-r^2\right)} \dfrac{x - x_0}{2\pi} \\ \rho = \left(1 - \dfrac{\gamma - 1}{16\gamma\pi^2} \beta^2 e^{2\left(1-r^2\right)}\right)^{\frac{1}{\gamma-1}} \\ p = \rho^\gamma \end{cases} \tag{4}$$

where, x0 = 5 y0 = 0, β = 5, and γ = 1.4. To verify the correctness and high-order effectiveness of matrix=free implementation, the case are run on a sequence of meshes with different scale. Table.1 shows convergence results, where matrix-free implementation obtains nearly theoretically optimal convergence rate.

Table 1: 2D isentropic vortex pressure field error and convergence rate

Order	h	0.5h	0.25h	Rate
1	7.5534E-03	2.0602E-03	5.2704E-04	1.92
2	2.3543E-03	2.6309E-04	3.1887E-05	3.10
3	4.2271E-04	3.0372E-05	1.7986E-06	3.94
4	1.0954E-04	3.7670E-06	1.0757E-07	4.99
5	2.7787E-05	4.3644E-07	7.2526E-09	5.95
6	6.1963E-06	7.0940E-08	5.3634E-10	6.85
7	2.1034E-06	7.2805E-09	1.2982E-10	7.00

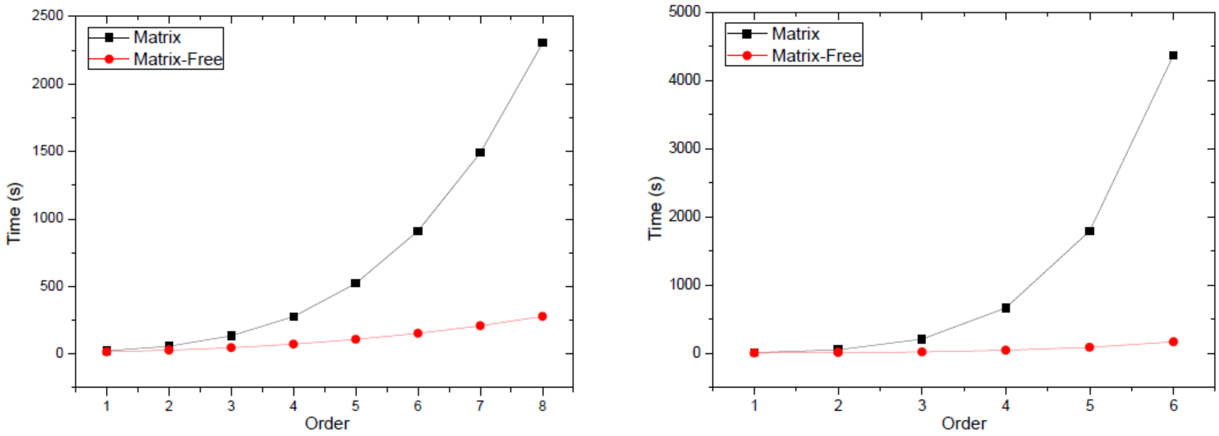
Based on the correctness and high-order effectiveness of our implementation, we have tested the performance by measuring the wall-clock time needed for different orders. Fig.1(a) shows the wall time of numerical simulation(1000 time steps) for various orders of DG, comparing the traditional matrix-vector multiplication implementation in serial with the matrix-free. Fig.1(b) details the wall time of simulation for three-dimensional isentropic vortex whose exact solution can be listed as Eqn.5.

$$\begin{cases} u = 1 - \beta e^{\left(1-r^2\right)} \dfrac{y - y_0}{2\pi} \\ v = \beta e^{\left(1-r^2\right)} \dfrac{x - x_0}{2\pi} \\ w = 0 \\ \rho = \left(1 - \dfrac{\gamma - 1}{16\gamma\pi^2} \beta^2 e^{2\left(1-r^2\right)}\right)^{\frac{1}{\gamma-1}} \\ p = \rho^\gamma \end{cases} \tag{5}$$

Compared the result of original matrix-vector multiplication with the matrix-free's, we can get that the matrix-free RKDG effectively speed up the simulation. Typically, when the polynomial order is 8th, the performance speed-up ratioof matrix-free is 8.278 higher than the original matrix-vector multiplication in two-dimensional space, and 26.287 times higher than for the 6th polynomial order in three-dimensions.

Moreover, we test the parallel scalability of matrix-free RKDG implementation on two-dimensional isentropic vortex problem with 4 million DoFs. Fig.2 describes the corresponding parallel speed-up ratio on different number of cores. From 1 to 192 cores, the parallel speed-up ratio of matrix-free implementation is approximately the same as original, which means the matrix-free does not decrease the parallel speed-up ratio of HopeFOAM.

Finally, we test the complex double mach problem with reflective boundary conditions using the detector and the limiter. The temporal integration is conducted by RKDG, coupled with matrix free. The numerical results reveal excellent accuracy and efficiency. Fig.3 shows the density of double mach problem.



(a) 2D case result (b) 3D case result

Figure 1: Wall-clock time

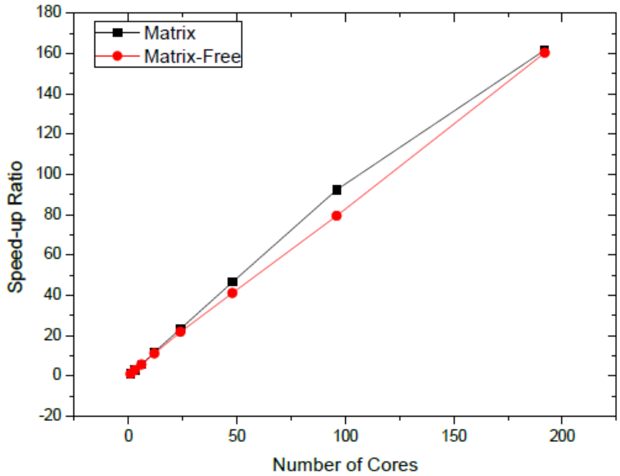


Figure 2: The parallel speed-up ratio of 2D isentropic vortex test on HopeFOAM

Figure 3: The contour plot for density of double mach

Acknowledgments

The authors would like to thank the Science Challenge Project (No. TZ2016002).

References

[1] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh et al., "High-order cfd methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, vol. 72, no. 8, pp. 811–845, 2013.

[2] Z. Wang, "A perspective on high-order methods in computational fluid dynamics," *Science China Physics, Mechanics & Astronomy*, vol. 59, no. 1, p. 614701, 2016.

[3] C.-W. Shu, “Discontinuous galerkin methods: general approach and stability,” *Numerical solutions of partial differential equations*, vol. 201, 2009.

[4] T. E. Group, “Hopefoam: High order parallel extensible cfd software,” https://github.com/HopeFOAM/HopeFOAM, accessed January 2, 2018.

[5] H. Jasak, A. Jemcov, Z. Tukovic et al., “Openfoam: A c++ library for complex physics simulations,” in *International workshop on coupled methods in numerical dynamics*, vol. 1000. IUC Dubrovnik, Croatia, 2007, pp. 1–20.

[6] M. Kronbichler and K. Kormann, “A generic interface for parallel cell-based finite element operator application,” *Computers & Fluids*, vol. 63, pp. 135–147, 2012.

[7] M. Kronbichler, B. Krank, N. Fehn, S. Legat, and W. A. Wall, “A new high-order discontinuous galerkin solver for dns and les of turbulent incompressible flow,” in *New Results in Numerical and Experimental Fluid Mechanics XI. Springer*, 2018, pp. 467–477.

[8] B. Cockburn and C.-W. Shu, “Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework,” *Mathematics of computation*, vol. 52, no. 186, pp. 411–435, 1989.