Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Efficient WENO library for OpenFOAM

Jan Wilhelm Gärtner[a,*], Andreas Kronenburg[a], Tobias Martin[b]

[a] Institute for Combustion Technology, University of Stuttgart, Herdweg 51, 70174 Stuttgart, Germany
[b] Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway

ARTICLE INFO

Article history:
Received 31 July 2020
Received in revised form 6 October 2020
Accepted 7 October 2020

Keywords:
OpenFOAM
WENO
Higher order schemes
Numerical validation
CFD

ABSTRACT

The weighted essentially non oscillating (WENO) concept is well established in research and its advantages are known, however, implementation details such as memory demand hindered the usage for general applications and general purpose libraries for many open source CFD tools do not yet exist. This paper introduces a WENO library for OpenFOAM and describes an efficient implementation. The large memory demand of the scheme is solved be reusing already calculated stencil sets and storing them in a data bank, giving a possible memory reduction of over 90%. A new class is implemented to allow the reconstruction of the stencil list on highly decomposed meshes by reconstructing a regional mesh around each processor. This avoids accuracy deficiency and stability problems at the processor boundary. Lastly, the performance of the implemented scheme is demonstrated by a standard Taylor Green Vortex test case.

Code metadata

Current code version	1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-20-00017
Code Ocean compute capsule	
Legal Code License	GNU General Public License v3.0
Code versioning system used	git
Software code languages, tools, and services used	C++, MPI, OpenFOAM, Catch2
Compilation requirements, operating environments & dependencies	OpenFOAM v5.x or OpenFOAM v7. Tested on Ubuntu 18.04 LTS
If available Link to developer documentation/manual	
Support email for questions	jan-wilhelm.gaertner@itv.uni-stuttgart.de

Software metadata

Current software version	1.0
Permanent link to executables of this version	For example: https://github.com/WENO-OF/WENOEXT
Legal Software License	GNU General Public License v3.0
Computing platforms/Operating Systems	Ubuntu 18.04 LTS
Installation requirements & dependencies	OpenFOAM v5.x or OpenFOAM v7
If available, link to user manual - if formally published include a reference to the publication in the reference list	
Support email for questions	jan-wilhelm.gaertner@itv.uni-stuttgart.de

1. Motivation and significance

Computational fluid dynamics (CFD) have become a standard tool in research and industrial development. Commercial tools

[*] Corresponding author.
E-mail address: jan-wilhelm.gaertner@itv.uni-stuttgart.de (J.W. Gärtner).

https://doi.org/10.1016/j.softx.2020.100611

provide a wide range of applications but licensing costs may be high which provokes an ever increasing interest in open source and free to use software such as OpenFOAM. OpenFOAM provides a versatile and easy to use tool set to develop and write CFD software for structured and unstructured grids [1]. It already ships with a wide range of solvers and discretization schemes. Yet, one class of discretization schemes, derived to handle flows with discontinuities and a high order of accuracy, is still missing.

While linear schemes are most suitable to resolve the complete wave number spectrum, they suffer from Gibbs instabilities around large discontinuities which make them unsuitable for simulations containing shocks. This problem is overcome by using hybrid schemes that fall back to a more stable first order, often upwind biased, solution for regions around discontinuities. One class of these kind of schemes are the total variation diminishing (TVD) and normalized variation diminishing (NVD) schemes with an appropriate limiter. However, these schemes are too diffusive, even in smooth regions, and their use in LES simulations is questionable [2–5]. Essentially non oscillating schemes (ENO) have been developed to achieve high-order discretization in smooth regions and to retain the TVD property, required to resolve discontinuities [2]. This type of scheme uses a reconstructed polynomial to modify the flux to achieve TVD properties. The stability and accuracy of the scheme has then been greatly advanced by using a weighted set of the polynomial stencils instead of choosing the smoothest stencil [6,7]. These schemes are commonly referred to as WENO schemes.

Although WENO schemes have originally been developed for structured grids, they have been extended to be applicable for any cell shape as well as unstructured grids [3,8] and tested OpenFOAM implementations exist [9,10]. However, the straightforward implementation of the WENO scheme raised performance issues concerning parallelization, CPU costs, and memory management. These issues rendered WENO unfeasible for general purpose, large scale simulations. The work presented here improves some major drawbacks of the WENO scheme in OpenFOAM, especially memory requirement. Further, parallelization is now improved and uses a full reconstruction instead of a 0-halo approach to represent the multi-stencil algorithm of the WENO scheme.

2. Software description

2.1. Theoretical background of the WENO scheme

The principal idea of the WENO scheme is to replace the cell average value $\overline{\Phi}_i$ of cell i and volume V_i, by a polynomial representation p_i. To prevent scaling effects due to mesh stretching, e.g. in unstructured mesh, the polynomial is constructed in a reference space $\vec{\xi} = \vec{\xi}(x, y, z)$ which can be transformed into the physical space by its affine transformation $\vec{x} = \vec{x}(\xi, \eta, \zeta)$,

$$\overline{\Phi}_i = \frac{1}{|V_i|}\int_{V_i} \Phi(\vec{x})\mathrm{d}x\mathrm{d}y\mathrm{d}z = \frac{1}{|V_i'|}\int_{V_i'} \Phi(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta = \frac{1}{|V_i'|}\int_{V_i'} p(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta . \tag{1}$$

Here, V_i' is the mapped cell V_i in the reference space and p is the representing polynomial which can be expressed by an expansion over local basis functions Ω_k with the degrees of freedom K as,

$$p_i(\vec{\xi}) = \overline{\Phi}_i + \sum_{k=1}^{K} a_k \Omega_k(\vec{\xi}). \tag{2}$$

The basis functions are summed up over the degrees of freedom which depend on the order of the polynomial r [8]. The central idea of the WENO scheme is to use a weighted convex combination of the polynomial of each stencil [6],

$$p_{i,\mathrm{weno}}(\vec{\xi}) = \sum_{m=0}^{N_{Si}} \omega_m p_i^{(m)}(\vec{\xi}), \tag{3}$$

where N_{Si} is the number of stencils. The stencil list includes one central stencil and n sectorial stencils, where n is the number of faces of the cell V_i. The weighting of the stencil follows the approach presented in Pringuey [9] and Henrick et al. [11] and it is referred to these references for further detail.

The basis function Ω_k of Eq. (2) satisfy the constrain of Eq. (1) for

$$\Omega_k(\vec{\xi}) = \Psi_k(\vec{\xi}) - \frac{1}{|V_i'|}\int_{|V_i'|} \Psi_k(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \tag{4}$$

and Ψ_k as an arbitrary orthogonal basis function. A suitable basis function is found using a Taylor series expansion around the center of V_i [12,13].

The final equation system to calculate the coefficients a_k is derived by calculating the average value $\overline{\Phi}_j$ of all cells in stencil m by replacing $p^{(m)}$ in,

$$\overline{\Phi}_j = \frac{1}{|V_j'|}\int_{V_j'} p^{(m)}(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \tag{5}$$

with Eq. (2) of the central cell i which leads to,

$$\overline{\Phi}_j = \frac{1}{|V_j'|}\int_{V_j'} \left(\overline{\Phi}_i + \sum_{k=1}^{K} a_k^{(m)} \Omega_k(\vec{\xi})\right)\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \tag{6}$$

$$\overline{\Phi}_j - \overline{\Phi}_i = \sum_{k=1}^{K} a_k^{(m)} \left(\frac{1}{|V_j'|}\int_{V_j'} \Omega_k(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta\right) \tag{7}$$

$$\overline{\Phi}_j - \overline{\Phi}_i = \sum_{k=1}^{K} a_k^{(m)} \mathcal{A}_{jk} \tag{8}$$

The matrix $\mathcal{A}_{jk}$ is calculated under consideration of the basis functions of Eq. (4) with,

$$\mathcal{A}_{jk} = \frac{1}{|V_j'|}\int_{V_j'} \Psi_k\text{ł}(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta - \frac{1}{|V_i'|}\int_{|V_i'|} \Psi_k(\vec{\xi})\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta . \tag{9}$$

The solution of the volume integrals through a transformation and using surface integrals is described in detail in the work of Martin and Shevchuk [10].

2.1.1. Solving the equation system

To solve the equation system of Eq. (8), at least K cells have to be considered for each stencil, e.g. with the matrix dimension $\mathcal{A}^{K\times K}$. Using this minimum number of cells often leads to ill conditioned matrix systems which is why a surplus of cells is considered, $\mathcal{A}^{N\times K}$, where N is typically chosen to be two times the degree of freedom [3,8].

It is obvious that the matrix $\mathcal{A}$ only depends on geometrical parameters and thus can be computed and stored before run time. Typically a QR decomposition is used to efficiently compute a_k at run time. However, as mentioned above, the memory requirement of this scheme is one main concern and the QR decomposition requires storage of the $Q^{N\times N}$ and the $R^{N\times K}$ matrices. Therefore, it is more feasible to use a singular value decomposition (SVD) to calculate and then store the Moore–Penrose pseudo inverse $\mathcal{A}^{+}$ of $\mathcal{A}$.

2.2. Challenges of WENO schemes in OpenFOAM

The stencil collection and calculation of the Moore–Penrose inverse is the computationally most expensive part. From the presented implementation it is obvious, that most of the work required for the WENO scheme can be done in a pre-processing step. The solution is then stored in memory and written to the hard disk for future use. Pre-processing and storage of the data, however, leads to a large memory demand. This becomes especially important when the scheme is used in an LES context, in which grids with more than 1 million cells are not uncommon. In addition, standard OpenFOAM routines cannot be used for parallelization as they only consider the next direct neighbor of cells and do not represent the multi stencil approach. Therefore, two main concerns have been identified that hinder the use of the WENO scheme in a general purpose OpenFOAM code:

(1) Memory demand for the pseudo inverse matrix $\mathcal{A}^+$
(2) Stencil collection on parallel decomposed meshes

Both of these concerns are addressed in the following section.

2.2.1. Memory demand

The huge memory demand required for a straight forward implementation is illustrated by this example. Assuming a grid size of 1 million cubic cells, a WENO scheme of third order has to store 7 matrices with 19 × 38 coefficients. This leads to a memory demand of 40 GB for the matrices alone. For 5th order scheme 338 GB of memory would be required.

As the pseudo inverse only depends on the geometry of the mesh it can be potentially different for each stencil and cell. In most cases, however, we deal with somewhat structured grids. Therefore, most of the matrices are not unique and it would be desirable to store only the unique matrices and pointing to them where they are needed. This can be achieved by replacing the used list with a new data structure, in the following called the matrix data bank.

Matrix data bank system. The underlying data structure for this data bank is an STL standard multimap which has a lookup time of $\log(n)$. During the calculation of the inverse matrices $\mathcal{A}^+$ for each stencil and cell it is checked if a similar matrix is already stored within the data bank. To check if this matrix already exists the key of the multimap is calculated by summation over the matrix coefficients. It is favorable to convert the floating point value of the coefficients to a representing integer, so that similar matrices are grouped together. To do so, the matrix is normalized by the largest element and then multiplied by 1×10^6. This hashing algorithm is preferred over standard algorithms, like `arrays.hashCode()` of java, as it is not sensitive to round off errors. After the key k is calculated all matrices for the key below k^- up to the key above k^+ are looked up and compared to the new matrix. Hereby, a matrix $\mathcal{A}^*$ is defined as similar if none of the coefficients differs by more than 1×10^{-9} to the reference matrix $\mathcal{A}^k$ stored in the data bank,

$$\frac{\sum(\mathcal{A}^* - \mathcal{A}^k)}{\max(\mathcal{A}^*)} < 1 \times 10^{-9}. \qquad (10)$$

If such a matrix is found the new matrix $\mathcal{A}^*$ is not added and the iterator to the previously stored matrix is returned. Here, it shall be stressed that a writing precision for ASCII files below 9 significant digits will lead to many matrices that should in theory be equal but are due to the inaccuracy of the read in mesh data not. Thus, it is advised to use a high writing precision to generate the mesh, or best using a binary format. If no similar matrix is found, $\mathcal{A}^*$ is added to the data bank and the iterator is returned as well. Now, only the iterators to the matrices within the data bank have to be stored. Storing the iterators instead of the map key allows constant lookup times during the run time of the simulation.

Table 1
Memory demand for a case on a regular cubic and a cylinder mesh using a WENO 2nd order scheme.

Case	Memory matrix databank [GB]	Relative reduction
No matrix DB	8	–
With matrix DB	0.1	98%
Best conditioned	0.1	98%
Grading & matrix DB	0.1	98%
Cylinder Mesh & no matrix DB	3.0	–%
Cylinder Mesh & matrix DB	0.2	93%

Best conditioned matrix system. Tsoutsanis et al. [8] recommended to use twice the number of cells to achieve convergence. However, this leads to a significantly increased memory demand for higher orders. Further, the accuracy of the matrix system,

$$\mathcal{A}x = b, \qquad (11)$$

depends on the square of the condition of $\mathcal{A}$ [14]. As the condition of the matrix $\mathcal{A}$ does not necessarily improve with more cells, a new algorithm is implemented that calculates the best conditioned matrix of the matrix list $\mathcal{A}^{K\times K}$ to $\mathcal{A}^{2K\times K}$. Thus, giving the potential to decrease the memory demand if the best conditioned matrix has a dimension less than $2K$ cells.

2.3. Stencil collection for parallel runs

The parallelization of the WENO scheme is not as straight forward as for any other OpenFOAM scheme. In difference to other schemes, such as the linear or TVD schemes, not only the direct neighbor of a cell is taken into account but a wider stencil. This means that for the collection of the stencil list the neighbor cells of a processor patch are not sufficient. Yet, OpenFOAM only provides information about the direct neighbor cells. To solve this issue each processor reconstructs a regional mesh including its direct neighboring processors and their neighbors to get a fully closed mesh around the processors own mesh. This process allows to collect the stencil list with all the information of the mesh present, while, at the same time, only a part of the mesh needs to be reconstructed for each processor.

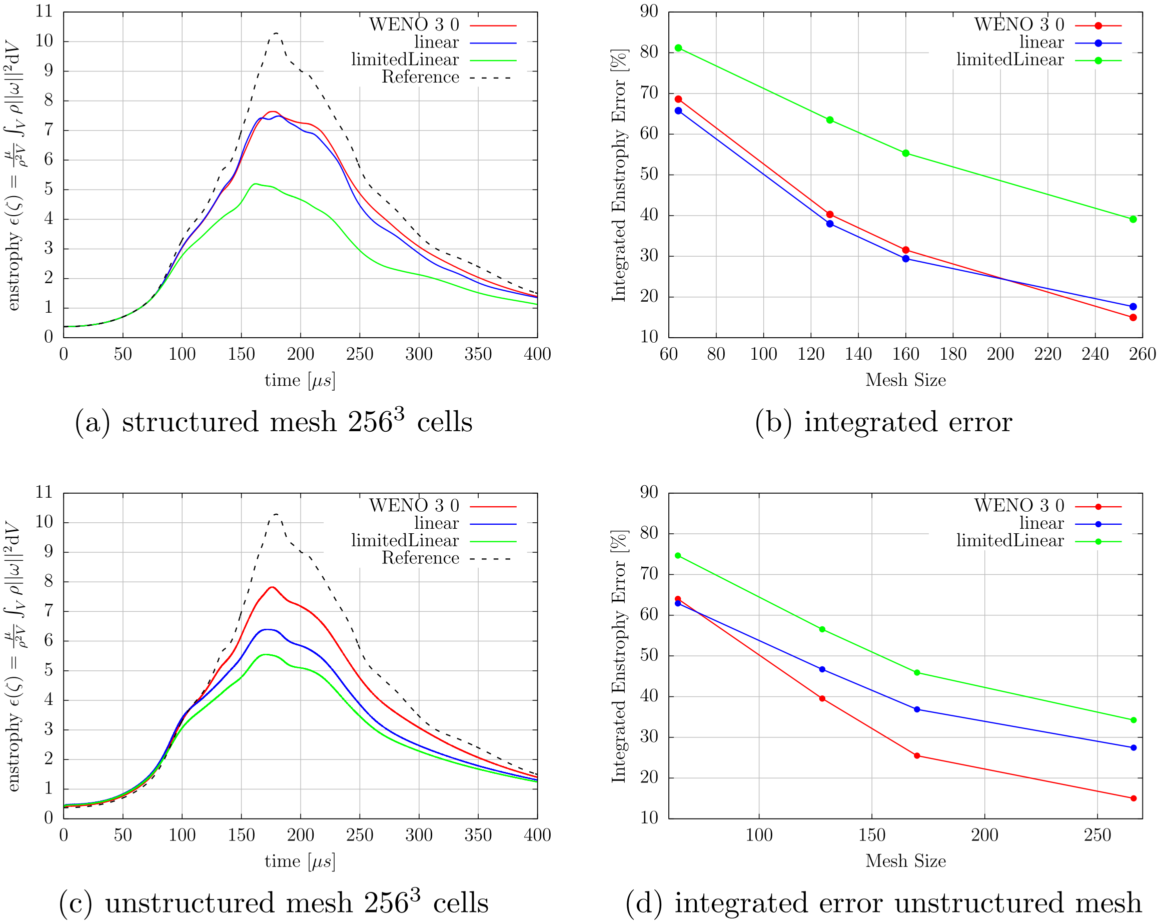
3. Illustrative examples

3.1. Memory reduction

To test the efficiency of the matrix data bank, three meshes one cube without and with mesh grading containing 100 × 100 × 100 cells and one cylinder (320k cells) are used. The memory demand is listed in Table 1 showing clearly that for such regular grids large memory reduction is possible. Further, the results show that the mesh grading has no effect on the pseudo inverse matrices. In fact, the most important factor for a high memory reduction is the accuracy of the mesh data. If the writing precision is too low, the calculated cell centers for theoretically equal cells are off and thus the reference space changes and as a consequence with it the matrices for each stencil. For the chosen test cases a writing precision of 9 was sufficient.

3.2. Achieved accuracy

To show the potential accuracy of the scheme the standard Taylor Green vortex (TGV) test case with a Reynolds number of 1600 is chosen. The test case is run on structured meshes with 64^3 up to 256^3 cells and unstructured meshes with a similar cell



(a) structured mesh 256^3 cells

(b) integrated error

(c) unstructured mesh 256^3 cells

(d) integrated error unstructured mesh

Fig. 1. Enstrophy of the Taylor Green vortex test case for a structured mesh with 128^3 cells (a) and the integrated error to the reference solution (b). Results for the unstructured mesh is shown in (c) and (d).

number. The structured meshes are generated with OpenFOAM blockMesh and the unstructured with the software SALOME using the Netgen-3D algorithm. A reference solution generated with a dealiased pseudo spectral code on a mesh with 512^3 cells is used to compare to the finite volume OpenFOAM results [15]. As the validating parameter the enstrophy is chosen [15,16]. Note, that the reference solution uses a finer mesh than used for this study. For comparing the implemented WENO scheme, the built in linear scheme and limited linear TVD scheme are selected for the velocity divergence. The TGV is then solved with the incompressible pimpleFoam solver of OpenFOAM.

Fig. 1(a) shows the development of the enstrophy over time for the structured grid and 256^3 cells. The results show, that the WENO scheme, with a third order polynomial, gives results in the range of the linear scheme and performs significantly better then the limitedLinear scheme. To compare the results for varying mesh size the difference to the reference solution is calculated as the relation of the integrated area under the curve. Figs. 1(b) and 1(d) display the integrated error over the mesh size. This shows that the WENO scheme is at least as accurate as the linear scheme and shows significant improvements for unstructured grids.

3.3. Resolving shocks in two phase flows

In fully flashing sprays the evaporating liquid exits the injector with a pressure higher than the ambient value which leads to an under expanded jet [17–20]. The resulting complex shock system is difficult to simulate as most multi phase solvers are using a pressure based approach while commonly transonic flows are solved by a density based approach. Whereas TVD schemes can aid to solve the discontinuity at the shock front, they fail to deliver the required accuracy, and are not stable enough. The developed WENO scheme can overcome this problem and allows for a stable and accurate simulation.

This is shown in Fig. 2 which depicts a 2D simulation of a fully flashing cryogenic liquid nitrogen jet representing the case 'LN2-2' of Gärtner et al. [20] with a superheat ratio of $R_p = 48.1$. In this test case liquid nitrogen at 89.7 K is injected through an injector with 1 mm diameter and an L/D ratio of 2.9 into a near vacuum chamber with the same temperature. The jet exits the injector with a pressure 37 times higher than the ambient pressure, leading to the shock system typical in under expanded jets. The case is simulated with a compressible multiphase solver developed in the same work using the Homogeneous Relaxation model (HRM) for the phase change.

With increasing mesh resolution, required for LES simulations, the shock front can no longer be resolved by the limitedLinear TVD scheme, see Fig. 2. In difference to this, the WENO scheme captures the shock accurately and allows for a correct resolution of the shock front while maintaining a high order accuracy in smooth regions. This shows the applicability of the WENO scheme for such problems especially in a multi phase context.

4. Impact and conclusion

In this work a high order WENO scheme library is introduced to OpenFOAM and made for the first time publicly available. The

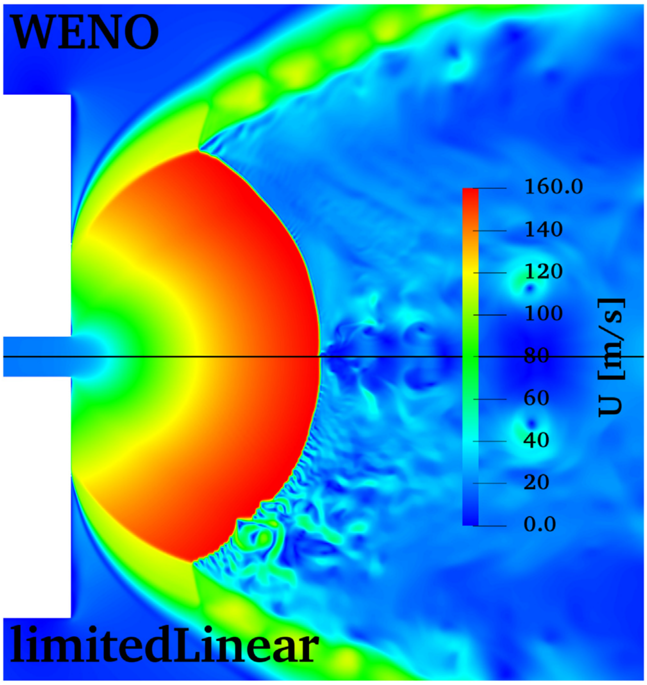


Fig. 2. Velocity magnitude at a shock front for fully flashing sprays comparing WENO scheme (top) and limitedLinear scheme (bottom) for the discretization of the momentum equation.

inherent problems of memory demand of a WENO scheme for unstructured grids is overcome by using an efficient data structure. Further, a new stencil collection strategy is implemented to solve the problem of processor boundaries for decomposed domains. The performance of the new data management is proven and makes the WENO scheme now applicable for large scale research or engineering applications. The accuracy of the scheme is investigated with the standard Taylor Green Vortex test case and has shown for structured grids equal performance as the central difference linear scheme. For unstructured grids the implemented WENO scheme gives far better results than the linear scheme and reduces the error by over 40%. Lastly, an example is given in which the WENO scheme is used to solve transonic flows within a multiphase context. While the formally stable TVD scheme fails to resolve the shock front accurately the WENO scheme delivers smooth and accurate results. Giving now the possibility to solve transonic multiphase problems in OpenFOAM with a high accuracy and stability.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank the German Research Foundation (DFG) for financial support of the project within the collaborative research center SFB-TRR 75, Project number 84292822. The authors also thank for the access to the supercomputer ForHLR funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

References

[1] The OpenFOAM Foundation. OpenFOAM v5 user guide. 2020, URL https://cfd.direct/openfoam/user-guide.

[2] Harten A, Osher S. Uniformly high-order accurate nonoscillatory schemes. I. SIAM J Numer Anal 1987;24(2):279–309.

[3] Dumbser M, Käser M. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. J Comput Phys 2007;221:693–723. http://dx.doi.org/10.1016/j.jcp.2006.06.043.

[4] Tsang C-W, Rutland C. Effects of numerical schemes on large eddy simulation of turbulent planar gas jet and diesel spray. SAE Int J Fuels Lubricants 2016;9(1):149–64. http://dx.doi.org/10.4271/2016-01-0866.

[5] Cao Y, Tamura T. Assessment of unstructured les with both structured les and experimental database: Flow past a square cylinder at Re = 2.2E+4. In: The 29th computational fluid dynamics symposium. 2015.

[6] Liu XD. Weighted essentially non-oscillatory schemes. J Comput Phys 1994;115:200–12. http://dx.doi.org/10.1006/jcph.1994.1187.

[7] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126(1):202–28. http://dx.doi.org/10.1006/jcph.1996.0130.

[8] Tsoutsanis P, Antoniadis AF, Drikakis D. WENO schemes on arbitrary unstructured meshes for laminar, transitional and turbulent flows. J Comput Phys 2014;256:254–76. http://dx.doi.org/10.1016/j.jcp.2013.09.002.

[9] Pringuey TRCM. Large eddy simulation of primary liquid-sheet breakup [Ph.D. thesis], University of Cambridge; 2012.

[10] Martin T, Shevchuk I. Implementation and validation of semi-implicit WENO schemes using OpenFOAM®. Computation 2018;6(1):6. http://dx.doi.org/10.3390/computation6010006.

[11] Henrick AK, Aslam TD, Powers JM. Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points. J Comput Phys 2005;207(2):542–67. http://dx.doi.org/10.1016/j.jcp.2005.01.023.

[12] Ollivier-Gooch CF. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. J Comput Phys 1997;133:6–17. http://dx.doi.org/10.1006/jcph.1996.5584.

[13] Friedrich O. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. J Comput Phys 1998;144:194–212. http://dx.doi.org/10.1006/jcph.1998.5988.

[14] Golub GH, Van Loan CF. Matrix computations, vol. 3. JHU Press; 2012.

[15] Wang Z, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, et al. High-order CFD methods: current status and perspective. Internat J Numer Methods Fluids 2013;72(8):811–45. http://dx.doi.org/10.1002/fld.3767.

[16] van Rees WM, Leonard A, Pullin DI, Koumoutsakos P. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. J Comput Phys 2011;230(8):2794–805. http://dx.doi.org/10.1016/j.jcp.2010.11.031.

[17] Lamanna G, Kamoun H, Weigand B, Steelant J. Towards a unified treatment of fully flashing sprays. Int J Multiph Flow 2014;58:168–84. http://dx.doi.org/10.1016/j.ijmultiphaseflow.2013.08.010.

[18] Vieira MM, Simoes-Moriera JR. Low-pressure flashing mechanisms in iso-octane liquid jets. J Fluid Mech 2007;572:121–44. http://dx.doi.org/10.1017/S0022112006003430.

[19] Kurschat T, Chaves H, Meier G. Complete adiabatic evaporation of highly superheated liquid jets. J Fluid Mech 1992;236:43–59.

[20] Gärtner JW, Kronenburg A, Rees A, Sender J, Oschwald M, Lamanna G. Numerical and experimental analysis of flashing cryogenic nitrogen. Int J Multiph Flow 2020;130:103360. http://dx.doi.org/10.1016/j.ijmultiphaseflow.2020.103360.