The Journal of Open Source Software

FEST-3D: Finite-volume Explicit STructured 3-Dimensional solver

Jatinder Pal Singh Sandhu[1], **Anant Girdhar**[1], **Rakesh Ramakrishnan**[1], **R. D. Teja**[1], **and Santanu Ghosh**[1]

1 Department of Aerospace engineering, IIT Madras, Chennai

DOI: 10.21105/joss.01555

Software

- Review
- Repository
- Archive

Editor: Lorena A Barba

Reviewers:

- @hariradh
- @rouson

Submitted: 04 July 2019
Published: 10 February 2020

Summary

Academic research in the mechanical- and aerospace-engineering communities has been aided in the last couple of decades by the development of open-source software packagess like OpenFoam (Weller, Tabor, Jasak, & Fureby, 1998) and SU2 (Economon, Palacios, Copeland, Lukaczyk, & Alonso, 2016).

`FEST-3D` is a modular CFD solver written in FORTRAN 90, developed with a similar motivation: to help solve problems of academic and engineering interest. This software is designed to solve the compressible Favre-averaged Navier-Stokes equations using the finite-volume method on block-structured grids using MPI-based parallelization. The modularity of the code makes it easy to implement a new method for flux reconstruction, or a turbulence model. It provides a large number of options for higher-order spatial and temporal discretization, along with the latest turbulence and transition models, which are not all available in other open-source CFD software. To illustrate, FEST-3D provides the latest one-equation γ transition model (Menter, Smirnov, Liu, & Avancha, 2015) and zero-equation BC transition model (Cakmakcioglu, Bas, & Kaynak, 2018). It also provides standard turbulence models: SA (Spalart & Allmaras, 1992) and SST (Menter, 1994), and the k-kL (Abdol-Hamid, Carlson, & Rumsey, 2016) turbulence model. As `FEST-3D` uses structured grids to solve fluid flow problems, higher-order methods of 3rd (Van Leer, 1979), 4th (Colella & Woodward, 1984), and 5th (Shu, 2003) order accuracy in space —- for uniform grids —- can be employed; this is difficult to achieve with solvers designed for unstructured grids and data-structures.

A Python script is provided to simplify the user interface with the main FEST-3D code. Most of the user inputs can easily be specified in the first few lines of the `edit-automaton.py` script, as listed in the table below.

Variable	Expected Input	Description
RunDir	String	Name of the run directory
GridDir	String (path)	Directory name having only grid files
NumberOfBlocks	Integer	Total number of blocks
CFL	Real Number greater than zero	Courant–Friedrichs–Lewy number
LoadLevel	Integer	Restart folder number in the time_directories/ directory
MaxIterations	Integer greater than zero	Maximum number of iteration
SaveIterations	Integer lesser than MaxIterations	Solution is written after every these many iterations
OutputFileFormat	'vtk' or 'tecplot'	Format of the solution output file
OutputDataFormat	'ASCII'	Type of the data in the output files. Only ASCII is supported for now

Variable	Expected Input	Description
InputFileFormat	'vtk' or 'tecplot'	Format of the solution file from which solution will be restarted
InputDataFormat	'ASCII'	Type of the data in the restart file. Only ASCII is supported for now
Precision	Integer, lesser than 14 and greater than 1	Data precision for residual output; not used for solution output
Purge	Integer	Number of recent solution folders to keep and delete others. **0** input will keep all the folders
ResidualWriteInterval	Integer greater than zero	Residual is written after every these many iterations
Tolerance	Real number and ["Mass_abs", "Continuity_abs", "Viscous_abs", "Resnorm_abs", "TKE_abs", "Tv_abs", "Dissipation_abs", "Omega_abs", "Kl_abs", "Turbulent_abs", "Resnorm_rel", "Viscous_rel", "Turublent_rel"]	Tolerance value and variable. The solver will stop once this value is achieved. List of tolerance variables that can be used is given in the expected input column. You can specify only one tolerance variable. The variable with *rel* suffix is normalized with first iteration residual
DebugLevel	1, 2, 3, 4, or 5	5-Only important information is logged, 1-All the information is logged, which helps in debugging. Will be removed in later release
InviscidFlux	'ausm', 'slau', 'ausmUP', or 'ldfss0'	Scheme to calculate inviscid fluxes through cell faces
FaceState	'none', 'muscl', 'ppm', 'weno', or 'wenoNM'	Scheme for higher-order face-state reconstruction
Limiter	'1 1 1 0 0 0' or '0 0 0 0 0 0'	Switch for limiters and pressure based switching when using higher order face-state reconstruction. Three values for I,J, and K directions; 1->on and 0-> off
TurbulenceLimiter	'1 1 1' or '0 0 0'	Switch for limiters when used for higher-order face-state reconstruction of turbulent variables; 1->on and 0-> off
TurbulenceModel	'none', 'sa', 'sst', or 'sst2003'	Turbulence model
TransitionModel	'none', 'bc', 'lctm2015'	Transition model
TimeStep	'l' or 'g '	Time-step for time-integration. 'l' for local and 'g' for global. In case of using a global method, you can provide the exact value of time step here
TimeIntegration	'none', 'RK2', 'RK4', 'TVDRK2', 'TVDRK3', 'implicit', or 'plusgs'	Method for time-integration
HigherOrderBC	0 or 1	Higher-order symmetry boundary condition. 1->on and 0-> off

Variable	Expected Input	Description
NumberOfVariables	5	Total number of variables to solve. This number is not used in current version of solver
DensityInf	Real Number	Free-stream density
UInf	Real Number	Free-stream x-component of velocity
VInf	Real Number	Free-stream y-component of velocity
WInf	Real Number	Free-stream z-component of velocity
PressureInf	Real Number	Free-stream pressure
TurbulenceIntensity	Real Number	Free-stream turbulence intensity in percentage
ViscosityRatio	Real Number	Free-stream turbulent viscosity to laminar viscosity ratio
Intermitency	Real Number	Free-stream turbulence intermittency
ReferenceViscosity	Real Number	Reference laminar viscosity
ViscosityLaw	'sutherland_law' or 'constant'	Method used for viscosity calculation
ReferenceTemp	Real Number	Reference temperature for viscosity calculation usiing Sutherland's law
SutherlandTemp	Real Number	Sutherland temperature
PrandtlNumbers	Two real numbers	Prandtl number and turbulent Prandtl number
SpecificHeatRatio	Real number	Specific heat ratio
GasConstant	Real	Specific gas Constant
OutputControl['Out']	["Velocity" , "Density" , "Pressure" , "Mu" , "Mu_t" , "TKE" , "Omega" , "kL" , "tv" , "Wall_distance" , "Resnorm"]	Variables to write in the output file. Specify the only the ones required. You do not need to specify the entire list
OutputControl['In']	["Velocity" ,"Density" ,"Pressure" ,"viscosity" ,"TKE" ,"Omega" ,"kL" ,"tv"]	Variables to read in case of restrart. Specify all the variable in the restart file
ResidualControl['Out']	Expected inputs are from the list of "Tolerance" variables	Residual to write in the resnorm file. Specify only the ones you want to write and you do not need to specify the entire list
BoundaryConditions	[-3, -4, -5, -8, -6, -6] where <-1:'SUPERSONIC INFLOW (DIRICHLET)', -2:'SUPERSONIC OUTFLOW (EXTRAPOLATION)', -3:'SUBSONIC INFLOW (MASS-FLOW RATE FIXED)', -4:'SUBSONIC OUTFLOW (PRESSURE FIXED)', -5:'WALL (NO SLIP)', -6:'SYMMETRY', -7:'POLE', -8:'FAR-FIELD', -11:'TOTAL INLET'>	Boundary conditions used for the six face of a block

Higher-order methods

Most modern CFD software is based on unstructured-grid data structures and are limited to a maximum of 3rd order of accuracy in space (Check OpenFoam v6 User Guide: 4.4), as it is computationally expensive and difficult to implement higher-order methods in this case. FEST-3D uses structured-grid data structures and provides higher than second-order methods like MUSCL (3rd-order accurate in space), PPM (4th-order accurate in space) and WENO (5th-order accurate in space), at least for uniform grid spacing. Such higher-order methods can especially be useful in academic research.

Past and current applications

FEST-3D is suitable for academic research and can also be used in industrial research. It has been used for obtaining simulations to investigate the effect of slope limiters on the convergence of the solution of smooth turbulent flows while using higher-order methods(Singh Sandhu, Girdhar, Ramakrishnan, Teja, & Ghosh, 2018). Currently, FEST-3D is being used for the development of a new local-correlation-based transition model and a 3D immersed-boundary method for compressible flows. FEST-3D is also being used for teaching in the department of Aerospace Engineering, IIT Madras.

Acknowledgement

We acknowledge the open source projects which we have used in the FEST-3D code. Seth Mortron's template project, available on GitHub as cmake_fortran_template, has been used to build the code. To automatically generate documentation, Fortran FOSS Programmers Group's document generator, FORD, has been used.

References

Abdol-Hamid, K. S., Carlson, J.-R., & Rumsey, C. L. (2016). Verification and Validation of the k-kL Turbulence Model in FUN3D and CFL3D Codes. In *46th AIAA Fluid Dynamics Conference* (pp. AIAA 2016–3941). doi:10.2514/6.2016-3941

Cakmakcioglu, S. C., Bas, O., & Kaynak, U. (2018). A correlation-based algebraic transition model. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *232*(21), 3915–3929. doi:10.1177/0954406217743537

Colella, P., & Woodward, P. R. (1984). The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, *54*(1), 174–201. doi:10.1016/0021-9991(84)90143-8

Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., & Alonso, J. J. (2016). SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, *54*(3), 828–846. doi:10.2514/1.J053813

Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, *32*(8), 1598–1605. doi:10.2514/3.12149

Menter, F. R., Smirnov, P. E., Liu, T., & Avancha, R. (2015). A one-equation local correlation-based transition model. *Flow, Turbulence and Combustion*, *95*(4), 583–619. doi:10.1007/s10494-015-9622-4

Shu, C.-W. (2003). High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. *International Journal of Computational Fluid Dynamics*, *17*(2), 107–118. doi:10.1080/1061856031000104851

Singh Sandhu, J. P., Girdhar, A., Ramakrishnan, R., Teja, R., & Ghosh, S. (2018). A convergence study of solutions using two two-equation rans turbulence models on a finite volume solver for structured grids. In *2018 AIAA Fluid Dynamics Conference* (pp. AIAA 2018–3859). doi:10.2514/6.2018-3859

Spalart, P., & Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*. doi:10.2514/6.1992-439

Van Leer, B. (1979). Towards the ultimate conservative difference scheme. V. A second-order sequel to godunov's method. *Journal of Computational Physics*, *32*(1), 101–136. doi:10.1016/0021-9991(79)90145-1

Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, *12*(6), 620–631. doi:10.1063/1.168744