AFRL-RQ-WP-TR-2022-0024

RAPID CFD: A FLEXIBLE RAPID COMPUTATIONAL FLUIDS DYNAMIC PLATFORM

Rachel Clipp
Sorin Mitran

Kitware, Inc.
University of North Carolina – Chapel Hill

NOVEMBER 2021
Final Report

THIS IS A SMALL BUSINESS TECHNOLOGY TRANSFER (STTR) PHASE 1 REPORT.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited; AFRL/PA-2024-1883; Cleared 04/08/2024.

AIR FORCE RESEARCH LABORATORY
WRIGHT RESEARCH SITE
2130 EIGHTH STREET, BUILDING 45
WRIGHT -PATTERSON AFB OH 45433-7541
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RQ-WP-TR-2022-0024 has been reviewed and is approved for publication in accordance with assigned distribution statement.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YY)*	**2. REPORT TYPE**	**3. DATES COVERED *(From - To*)**
22-11-2021	Final	23 April 2021 – 22 November 2021

4. TITLE AND SUBTITLE
Rapid CFD: A flexible rapid computational fluid dynamics platform

5a. CONTRACT NUMBER
FA8650-21-P-2315

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Rachel Clipp
Sorin Mitran

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
Q2CW

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Kitware, Inc.
1712 Route 9, Suite 300
Clifton Park, NY 12065

8. PERFORMING ORGANIZATION REPORT NUMBER
K003054-00

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Aerospace Systems Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RQKPB

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RQ-WP-TR-2022-0024

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited; AFRL/PA-XXXX-XXXX; Cleared XX-XX-XXXX.

13. SUPPLEMENTARY NOTES
This is a Small Business Technology Transfer (STTR) Phase I Report. Report contains color.

14. ABSTRACT *(Maximum 200 words)*

Report developed under SBIR contract for topic AF19B-013 Modern, Rapid, Usable, Lower Order Computational Fluid Dynamics (CFD) Development for Aerodynamic Analysis. The Air Force Research Lab (AFRL) is tasked with maintaining air superiority for the warfighter through research and development of aerodynamic technologies. This research includes configuration aerodynamics, flow control device development, wind tunnel testing, and high-fidelity computational fluid dynamics (CFD). However, current methods of CFD analysis are user time intensive and require extensive computational resources over hours or even days to complete a simulation. We completed a proof-of-concept platform, RapidCFD, RapidCFD includes a reduced-order CFD model and a custom ParaView plugin with for problem definition and setup and results analysis and visualization. Our RapidCFD Solver executes simulations in less than one second and less than five minutes for two-dimensional three-dimensional geometries, respectively. Our ParaView plugin successfully demonstrates the potential for leveraging ParaView to expand on these results in Phase II to create an end-to-end simulation workflow and improve the RapidCFD Solver for aerospace applications.

15. SUBJECT TERMS
STTR Report, Lattice Boltzmann Method, Computational Fluid Dynamics, Residual Distribution Schemes, Aerodynamic Performance, Open Source, ParaView, Simulation, Scientific Visualization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			David Weston
Unclassified	Unclassified	Unclassified	UU	29	**19b. TELEPHONE NUMBER *(Include Area Code)***

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

1. Summary

RapidCFD is a software platform for efficient aerodynamic analysis during initial exploration and/or optimization of the design parameter space. The principal objective of RapidCFD is to markedly reduce the time and complexity associated with the design and analysis of aerodynamic performance. The computation time associated with traditional computational fluid dynamics (CFD) methods can limit the number of designs and analyses completed. The complex workflow is attributed to number of third party tools that may require costly licenses and numerous input and output requirements that are not compatible. RapidCFD addresses this objective with a combination of software engineering and fundamental research through collaboration between Kitware, Inc. (Kitware) and the University of North Carolina at Chapel Hill (UNC).

Kitware has created a proof-of-concept platform for geometry input, CFD analysis, result visualization, and scripting for post-processing of CFD results. Kitware has created a plugin compatible with ParaView, our open source, multi-platform data analysis and visualization application. By extending the ParaView interface to include the custom workflow required to load geometry, execute a rapid CFD analysis, and post-process the data, we have leveraged the extensive functionality already available in ParaView. The prototype design exhibits the potential for an end-to-end simulation workflow specifically for the aerospace industry. In Phase II, we will expand on this design by combining Kitware's Computational Model Builder (CMB), an open-source platform for full life-cycle simulation workflows, and Kitware's Catalyst, an open source extension of ParaView for the *in situ* use case, to provide a workflow informed by task analysis, enable simulation both locally and remotely, create ensemble simulations and single simulation for real-time analysis, and access to extensive data visualization and analysis features.

UNC has carried out fundamental research to extend the applicability of the Lattice Boltzmann Method (LBM) to aerodynamic analysis and design. LBM is a reduced-order representation of the motion of the atoms that form a fluid, with a reduced set of allowed fluid particle velocities. A statistical approach is adopted, and the Boltzmann equation furnishes a procedure for time evolution of fluid particle positions and velocities. Averaging of this data leads to the Navier-Stokes equations of fluid dynamics. UNC fundamental research extended the applicability of this approach to compressible and high Reynolds number flow of interest in aerodynamic design by: (1) use of the mathematical theory of information geometry (IG) to model particle collision processes; and (2) development of a reduced-order Euler residual distribution scheme (RDS). Both methods leverage large-scale parallelism of graphics processing cards. The overall approach is a viscid/inviscid interaction approach that carries out typical two-dimensional aerodynamic analysis in seconds, and three-dimensional isolated wing analysis in a less than five minutes.

2. Introduction

The Air Force Research Laboratory is tasked with maintaining air superiority for the warfighter through research and development of aerodynamic technologies. This research includes configuration aerodynamics, flow control device development, wind tunnel testing, and high-fidelity CFD. However, current methods of CFD analysis are user time intensive and require extensive computational resources over hours or even days to complete a simulation. In addition to the extensive computational resources, there are limited tools available that provide the ability to visualize and refine the mesh geometry and structure, define the CFD parameters, and analyze and visualize simulation results. Disparate solutions and tools are required for each step in the process, which requires user time to transition between tools and incompatibility between inputs and outputs of different third party tools. Without an end-to-end solution, the user spends hours importing geometry, defining, and refining mesh, using command line interfaces to define solver parameters and regimes, and analyzing and visualizing streamlines, velocity vectors, and pressure profiles post simulation. We proposed RapidCFD, a customized end-to-end solution for problem definition, rapid, reduced-order CFD simulation, and results analysis and visualization (Figure 1). Our Phase I proposal included a proof-of-concept custom workflow and a reduced-order CFD solution. The reduced order CFD solution reduces computation time by several orders of magnitude and our end-to-end solution allows problem definition to occur in minutes and enable real-time visualization of simulation results within a single interface.

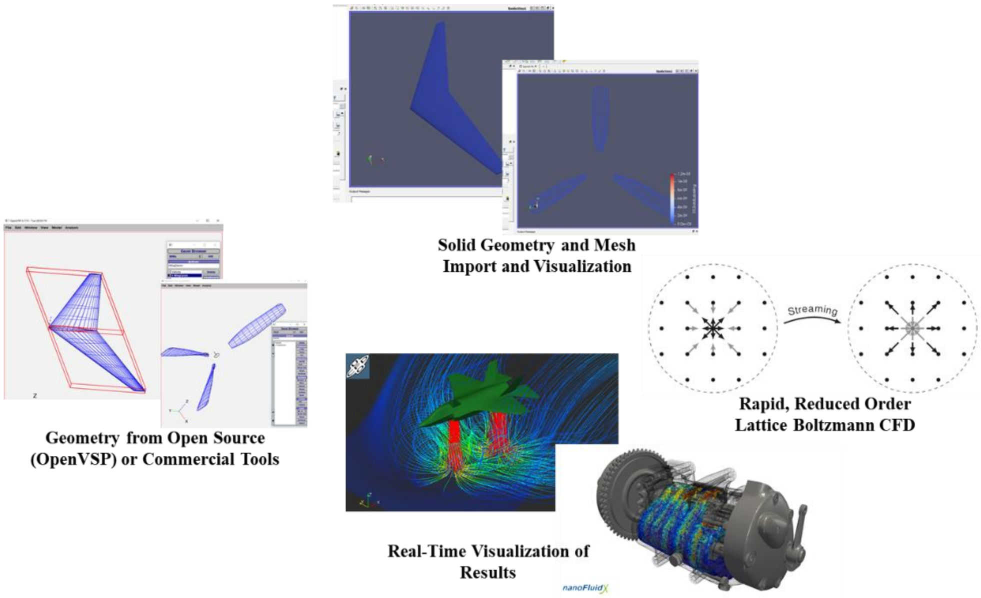


Figure 1. Proposed Components of the RapidCFD platform.

Reduced-Order CFD Background

Current high-fidelity CFD methods are routinely used to establish the aerodynamic characteristics of new aircraft designs and component configurations. Aircraft flow physics are accurately described by the Navier-Stokes (NS) equations; however, full direct numerical simulation of this method is prohibitively computationally expensive. Therefore, simplified models have been developed, such as the Reynold-Averaged NS (RANS)[1], in which the effects of small-scale dynamics are evaluated through a turbulence model, the Euler equations that assume viscous effects can be ignored, and the potential flow (e.g., panel method) that assume negligible compressibility. All of these methods require construction of a computational mesh or grid (typically unstructured for aerodynamics) and apply the fluid dynamics equations via relationships between the mesh values for density, pressure, and velocity. Even with the use of multiple processors these simulations, particularly for complex geometries and components, can take days to simulate using these methods. Therefore, further order reduction is required for rapid initial evaluation of aerodynamic designs, for example sub-structuring, most notably by identification of coherent modes [2] or principal components [3].

A markedly different approach to order reduction is to consider a fluid not as a continuum modeled by the NS or Euler equations, but to revert to a statistical description of the medium as a collection of particles (i.e., gas molecules) and consider the probability distribution function $f(t,x,v)$ that at given coordinates in time and space (t,x) a particle would attain some velocity value v. The evolution of the probability distribution function f is governed by the Boltzmann equation [4], and kinetic theory has shown that the continuum Euler and NS equations are obtained by taking moments of the Boltzmann equation. In this sense, the Boltzmann equation can be considered a more fundamental representation of the flow. The Lattice Boltzmann Method (LBM) is a discretization of the Boltzmann equation that allows for arbitrarily complex geometric configurations [5] by simply marking which lattice nodes are inside or outside of the flow domain. This approach has been used for biological flight [5], where multiple, moving aerodynamic surfaces over a wide range of Re numbers have highlighted the utility of this fundamentally different approach to flow modeling. This approach has found its way from biological flight back to engineering design of aircraft [6].

The LBM has proven to be an effective fluid dynamics algorithm for aerodynamics. The underlying algorithm, known as the "stream and collide" technique, includes two steps: (i) in the streaming step, particles at a lattice node advance one lattice step along a finite set of allowed directions, and (ii) particles at each lattice node relax towards an equilibrium state due to molecular collisions, Figure 2. The algorithm is particularly suited for massive parallelization on graphical processor units (GPUs). This technique has been demonstrated to reduce simulation time from days to minutes or a three to four orders of magnitude reduction [1]. However, this work focused on the human airways, which operate at lower Mach and Reynolds numbers (Ma < 0.2 and Re $< 10^4$). Higher Mach and Reynolds number regimes, relevant in aircraft design, are known to exhibit instabilities in the solution. Therefore, we have leveraged recent research from information geometry [7], the study of statistical manifolds in which each point is a probability distribution function. The resulting Information Geometry accelerated Lattice Boltzmann (IGLBM) algorithm is further detailed in the Methods section.

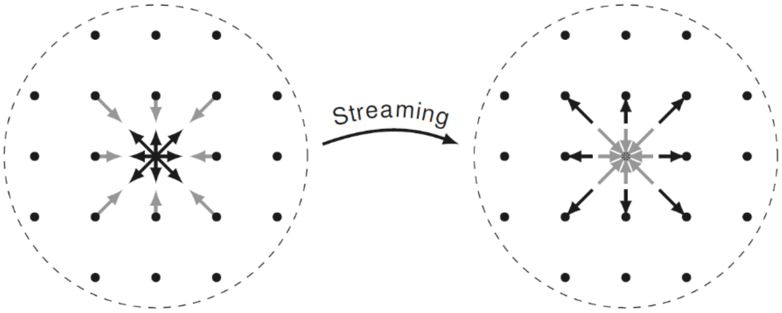


Figure 2. The Lattice Boltzmann Method stream and collide algorithm.

Workflow Background

Leveraging algorithmic advances into effective software is another important challenge. Current design codes (e.g., the Zeus TLNS3D in use at Boeing [8]) incorporate many legacy assumptions on grid generation, sub-structuring, and connectivity that significantly slows down the design process. Even using reduced-order models can be unwieldly for aerodynamic optimization or initial investigation of possible shapes due to cumbersome and inflexible procedures for importing geometry, specifying boundary conditions, and extraction of aerodynamic figures of merit. Without a structured, thoughtful workflow design, these software packages do not allow for transparent parameter identification and specification, efficient CFD execution, or analysis of CFD results. This is reflected in long user set up times and the use of multiple software packages. Kitware, Inc. is a leader in the development of productive workflows for physics-based simulations and computational analysis that are implemented as efficient codes in worldwide use, including ParaView [9]–[11], CMB [12]–[14], and Catalyst[15]. Kitware's experience incorporating the varied moving pieces, such as models, meshes, and information that describes material properties and boundary conditions have created a strong proof-of-concept for efficient and less error prone methods for initial CFD analysis.

RapidCFD Phase I

In this report, Kitware and UNC-CH share the results of our Phase I efforts and highlight the need for our Phase II advancements to our customer AFRL. We will share the advancements to the reduced -order CFD algorithm developed as part of this effort, including the combination of Euler and LBM. We will share the construction of our prototype workflow, including build instructions for each component, detailed instructions for execution of CFD analysis via command line interface and user-interface via our ParaView plugin, and examples of visualization techniques. Our results show the feasibility and potential of the reduced-order CFD method and the ability to leverage our open source software platforms to create an optimized workflow for the aerospace industry. We will close the report with a summary of the advancements we plan to propose for an end-to-end simulation workflow for the aerospace industry.

2. Methods, Assumptions, and Procedures

Overview

Kitware teamed with the UNC to develop a proof-of-concept reduced order simulation platform to address the critical needs of AFRL and the aerospace industry. The RapidCFD platform combines a solver (RapidCFD Solver) that uses the Euler reduced-order method and the LBM with a custom ParaView plugin, to generate a quick, yet accurate analysis of the behavior of aerodynamic components. Our Phase I work tested the RapidCFD Solver for two-dimensional and three-dimensional simulations and demonstrated the ability to execute these solutions in an easy-to-use workflow set up in ParaView. Using ParaView allows us to leverage the open source visualization tools developed for scientific visualization, including aerospace and automotive applications. The detailed methods of the solver and the plugin are discussed in the following sections.

Lattice Boltzmann Method

The Boltzmann equation [4] for time evolution of the probability distribution function (PDF) is

$$\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} + b \cdot \frac{\partial f}{\partial v} = I(f) \tag{1}$$

where *f(t,x,v)* is the probability that at given coordinates in time and space *(t,x)* a particle would attain some velocity value *v*. The PDF at position *(t,x)* changes due to streaming from adjacent locations ($v \cdot \partial f / \partial x$), effect of external body forces ($b \cdot \partial f / \partial v$) such as gravity, and collisions between particles *I(f)*. Averaging over the many particles that constitute a fluid element as in

$$\int_0^\infty v^k \, f(v) dv \tag{2}$$

giving the fluid density ρ for $k=0$, the fluid velocity u for $k=1$, and the fluid kinetic energy for $k=2$. Applying the averaging procedure leads to the conservation laws of fluid dynamics. The collision term average is approximated by expansion of f around the Maxwell-Boltzmann equilibrium value f_0 in terms of the Knudsen number, a small parameter,

$$\int_0^\infty I(f) f(v) dv = \int_0^\infty I(f_0 + \kappa\, f_1 + \kappa^2 f_2 + ..) f(v) dv \tag{3}$$

Truncation of the above expansion at successive powers of the Knudsen number leads to the Euler, Navier-Stokes, Burnett equations of fluid dynamics.

In its original form, the Boltzmann equation is extremely costly to solve since f depends on seven variables: time t, three space position components x, and three velocity components v. In a lattice Boltzmann method (LBM), the velocity is limited to taking only a small number of values (reduced-order description), for example in 2D, nine velocity values are allowed (D2Q9 model):

$$c\_i \in \{(0,0),(\pm 1,0),(0,\pm 1),(\pm 1,\pm 1)\} \tag{4}$$

The restriction of the *f(t,x,v)* probability density function (PDF) to the allowed discrete velocity values $f_i = f(t,x,c_i)$ allows the definition of a vector of single-velocity PDFs

$$\boldsymbol{f} = [f_0\, f_1\ \dots\ f_9] \tag{5}$$

at each position of a regular lattice covering the flow domain. The time evolution is then modeled in two steps (Figure 2):

1) Streaming

$$f^s{}_i(t,x) = f_i(t-1,x-c_i) \tag{6}$$

2) Collision

$$f_i(t,x) = f^s{}_i(t,x) + \int_0^1 \quad I(f)dt \tag{7}$$

where units are chosen such that the time step is equal to one, $\delta t = 1$. Both steps are inherently parallel, allowing efficient implementation on graphics processing units. The principal challenge in constructing a viable LBM is evaluation of the integral of the collision term, which corresponds to modeling molecular-scale relaxation processes. The earliest, and still most widely used, model is the single time scale (τ) Bhatnagar-Gross-Krook (BGK) process

$$f_i(t,x) = f^s{}_i(t,x) - \frac{f^s{}_i(t,x) - f^{eq}{}_i(t,x)}{\tau} \tag{8}$$

where $f^{eq}{}_i(t,x)$ is the local Maxwell-Boltzmann equilibrium.

The above BGK relaxation corresponds to a straight-line path from the non-equilibrated state obtained from the streaming step towards thermodynamic equilibrium. This approximation is sufficiently accurate for slow flows (incompressible, low Reynolds number), but in fast flows the fluid molecules do not fully relax to equilibrium, and the path towards equilibrium is not a straight line.

Information geometry is the study of families of probability distributions organized as a statistical manifold

$$S = \{f(t, x, v; \lambda)\} \tag{9}$$

in which each individual PDF is labeled by parameter vector λ. Of interest to LBM, geodesic paths (of minimal information distance) can be defined that lead to alternative relaxation models for the LBM. Incorporating this fundamental research result leads to the definition of the IGLBM method used in RapidCFD.

RapidCFD Solver: Euler and LBM

Exterior boundary conditions for the LBM method are furnished by an inviscid reduced-order model, based on solving the Euler equations using a Residual Distribution Scheme (RDS). The Euler equations are obtained by the lowest order truncation of the averaged Boltzmann equation, neglecting viscous effects, and are stated in two dimensions as

$$\partial_t q + \partial \frac{F(q)}{\partial} x + \partial \frac{G(q)}{\partial} x = 0 \tag{10}$$

With

$$q = (\rho \; \rho u \; \rho v \; \rho E) \tag{11}$$

containing the conservation variables, densities of mass, momentum, and energy. The fluxes are

$$F(q) = (\rho u \; \rho u^2 + p \; \rho uv \; \rho uH), G(q) = (\rho v \; \rho vu \;\; \rho v^2 + p \; \rho vH) \tag{12}$$

and the system is closed through an equation of state between thermodynamic variables, e.g., the perfect gas law $p = \rho R\, T$.

In a residual distribution scheme, the Euler equations are integrated over a control volume T (triangle in 2D, tetrahedron in 3D)

$$Q = \int_T^t q dV \tag{13}$$

$$\partial_t Q = -\int_T \left[\partial \frac{F(q)}{\partial} x + \partial \frac{G(q)}{\partial} x\right] dV = R \qquad (14)$$

At steady state the residual is null, while at stage *n* of an iterative process in which the current approximation of the flow field is Q^n , the residual is $R(Q^n)$. In a residual distribution scheme (RDS) the imbalance ($R(Q^n) \neq 0$) is distributed to the cell vertices in order to attain local equilibrium. The great computational advantage of an RDS is inherent parallelism: all cells can be processed simultaneously, followed by a step in which all cell vertices (nodes) are processed simultaneously.

In most aerodynamic analysis, the body of interest is placed into a steady free stream, and unsteady effects are typically concentrated in a small region in the immediate vicinity of the aerodynamic body. The Euler RDS efficiently transmits far field boundary conditions to the vicinity of an aerodynamic body, a region in which the IGLBM is subsequently applied to capture viscous and unsteady flow.

RapidCFD Solver Build Instructions

The core implementation of both the Euler and LBM algorithms are contained in a repository that provides a C++/CUDA library that can be integrated into various applications. A simple command line test driver is also provided to demonstrate execution of the algorithms on a provided mesh file. This repository is located within a gitlab git repository hosted by Kitware. The RapidCFD repository is currently set to private and only viewable by users controlled by the Kitware team. To gain access to the repository, users must create a profile at https://gitlab.kitware.com/ and provide it to the Kitware team by emailing the principal investigator (PI). Once the user has been added to the repository, users may go to https://gitlab.kitware.com/LBM/rapidcfd to clone the repository.

Example of cloning the repository using a bash prompt:

```
$ mkdir rcfd
$ cd rcfd
# Pull the source, Note the period in the following line
$ git clone https://gitlab.kitware.com/rapidcfd/rapidcfd.git.
```

The implementation utilizes C++17 and CUDA11, this requires the minimum C++ compiler provided by either GCC 7 or Visual Studio 2017, along with CUDA 11.0. If you have multiple CUDA versions and/or GCC versions, it may be necessary to specify Cmake and or CUDA configuration variables to properly build. The provided project is also a Cmake project and will require Cmake. Go to the cmake website, https://cmake.org/download, and download the appropriate distribution. Ensure that the Cmake bin is on your PATH and available in your cmd/bash shell.

```
# This assumes you are still in your source directory from the steps above
# Create a separate build directory
$ cd..
$ mkdir rcfd_build
$ cd rcfd_build
# Run Cmake and provide it the source directory
$ cmake../rcfd
```

The RapidCFD library supports outputting data in simple ASCII based TecPlot files and VTK based files for inspection by the user. The RapidCFD library provides an optional build parameter to generate VTK based files or not. By default this VTK extension is disabled. If enabled, the user will need to pull, build and provide Cmake with a prebuilt VTK directory. To build VTK, follow the steps provided here:

https://gitlab.kitware.com/vtk/vtk/blob/master/Documentation/dev/build.md

Enabling the Cmake variable RapidCFD_VTK_EXTENSION will enable VTK support and allow users to generate native vtk formats along with the generic TecPlot formats. To change cmake options (ex. Enable a Cmake switch), you can do this several different ways. (NOTE: ON is interchangeable with YES, and OFF is interchangeable with NO.)

- Via Command Line: Add -DRapidCFD_VTK_EXTENSION=ON
- Via Cmake GUI: Click to check the value field box associated with the RapidCFD_VTK_EXTENSION line to change it
- Via ccmake: Use arrows to highlight the RapidCFD_VTK_EXTENSION option and use the spacebar to toggle the option state (ON or OFF)

To learn more about Cmake visit the Running Cmake Help Page (https://cmake.org/runningcmake/). If you are using the Cmake GUI, it is suggested to enable Grouped and Advanced in the Cmake GUI. If you are using Cmake via the command like, it is suggested to run Cmake and use the cmake-curses-gui if you would like to modify any configuration options.

If you are enabling RapidCFD_VTK_EXTENSION, you will need to build VTK and set the Cmake Property VTK_DIR to your VTK build directory. This variable will be created for you to set when you configure Cmake for RapidCFD with RapidCFD_VTK_EXTENSION enabled, but you may provide it when running via command line:

```
# Run Cmake and provide it the source directory
$ cmake../rcfd -DRapidCFD_VTK_EXTENSION=ON -DVTK_DIR=<path/to/vtk/build/dir>
```

If you are building the RapidCFD Plugin, you should not build VTK, instead build ParaView first and set the VTK_DIR parameter to the ParaView build directory. To build ParaView, please see these instructions.

Once Cmake configure and generate has been run, you can build the library and example executable. Microsoft Visual Studio users can simply open and build the RapidCFD.sln found in your specified build directory. Linux users only need to call make within the build directory

```
# Build the project
$ make-j6
```

Running RapidCFD Example Driver

The RapidCFD repository also contains a simple executable to exercise the provided algorithms via a command line interface. This driver takes a folder or a single configuration file (*.cfg) as the input. If you provide a directory, rcfd will recursively search it for all configuration files. A single rcfd run consists of a folder containing an input configuration file along with the associated mesh/geometry file (.msh). When executed, the driver will:

1. Create a new folder in its working directory (install/bin) named results
    a. If given a directory, create the same folder hierarchy found under the input (or default) search folder.
2. Write all results and data files associated with the execution and analysis of each run to its analogous folder under a results folder
    a. If only running a single configuration file, all generated files will be placed directly in the results folder

Inputs

A configuration file must be supplied. A set of examples are provided in the repository under the data directory. In line comments within the example configuration files describe each property and its use, see Table 1.

Table 1. RapidCFD Parameter Definitions

Data Value	Variable Name	Units	Description
0.15	mach		Mach Number
0.0	alpha	deg	Angle Of Incidence
1.4	gammaGas		Ratio Of Specific Heats
1.0e-6	epsilon		Residual per node for Euler convergence criteria
101300.	pRef	Pa	Reference pressure
340.00	cSound	m/s	Sound speed
0.1	visc	Pa s	Viscosity
1.0	scale		Reference length of aerodynamic body
0.5	delta		Fraction of scale to add to surrounding body
0.002	dx	m	Grid Spacing
6000	nEulerCalls		Maximum allowed iterations for Euler solver
1000	nEulerIter		Euler iterations to write intermediate results to Euler data file
medium.msh	nLBMsteps		Number of LBM time steps, set to 0 to use internal default

<u>Outputs</u>

At the root of the results directory will be several files containing data calculated at various stages of RapidCFD. All files will have the same name as the input configuration file. The files defined in Table 2 are generation during the simulation.

Table 2. RapidCFD Output File Descriptions

File name	Description
*.euler.dat	A TechPlot file containing the values generated by the Euler solver
*.euler.vtk	A VTK file containing the values generated by the Euler solver (Only if VTK extensions is enabled)
*.lbm.dat	A TechPlot file containing the values generated by the LBM solver
*.lbm.vtk	A VTK file containing the values generated by the LBM solver (Only if VTK extensions is enabled)
*.log	A log file of all information/errors/warnings provided by RapidCFD
*.msh	A mesh file of the uniform grid provided to the LBM code

Example usage

```
# The rcfd executable will be placed in your install/bin directory
$ cd <build_dir>/install/bin/
# Run the rcfd Driver for one test case (should a few seconds)
$ ./rcfd <source_dir>/data/2D/airfoils/NACA0012/Mach0p15Alpha00p0.cfg
# 6 files will be written to <build_dir>/install/bin/results/
```

All files generated can be viewed using ParaView.

RapidCFD ParaView Plug In Build Instructions

A separate repository contains the RapidCFD ParaView plugin source code. Building the RapidCFD ParaView plugin requires a build of ParaView. To build ParaView, please see these instructions. Once ParaView has been built, follow the above build instructions to build the RapidCFD library with VTK extensions. This repository is located within a gitlab git repository hosted by Kitware. The RapidCFD ParaView Plugin repository is currently set to private and only viewable by users controlled by the Kitware team. To gain access to the repository, users must create a profile at https://gitlab.kitware.com and provide it to the Kitware team via email to the PI. Once the user has been added to the repository, users may go to https://gitlab.kitware.com/LBM/rapidcfd-paraview to clone the repository.

Example of cloning the repository using a bash prompt:

```
$ mkdir rcfd_paraview
$ cd rcfd_paraview
# Pull the source, Note the period in the following line
$ git clone https://gitlab.kitware.com/rapidcfd/rapidcfd-paraview.git.
```

The provided project is also a Cmake project and will require Cmake. Go to the cmake website, https://cmake.org/download, and download the appropriate distribution. Ensure that the Cmake bin is on your PATH and available in your cmd/bash shell.

```
# This assumes you are still in your source directory from the steps above
# Create a separate build directory
$ cd..
$ mkdir rcfd_paraview_build
$ cd rcfd_paraview_build
# Run Cmake and provide it the source directory
$  cmake  -DparaView_DIR=/path/to/paraview/build  -DRapidCFD_DIR=../rcfd_build
../rcfd_paraview
```

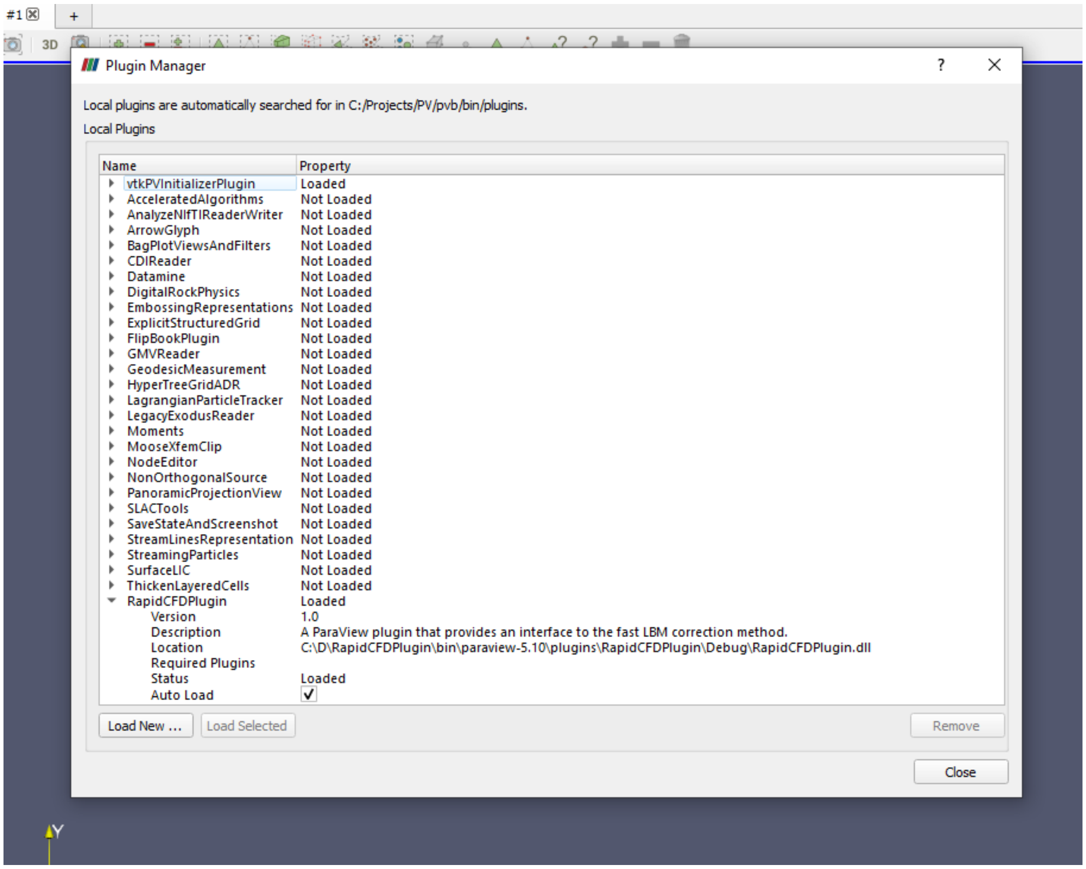
Once Cmake configure and generate has been run, you can build the library and example executable. Microsoft Visual Studio users can simply open and build the RapidCFD.sln found in your specified build directory. Linux users only need to call make within the build directory

```
# Build the project
$ make -j6
```

When the build completes successfully, the ParaView plugin will be located under the rcfd_paraview_build/bin/paraview-5.9/plugins/RapidCFDPlugin directory. Depending on the operating system, the plugin file have a *.dll or *.so extension.

RapidCFD ParaView Plugin Use

We have set up the generic architecture for the plugin, which can be built and loaded into an installed version of ParaView per the previous section's instructions. To load the RapidCFD plugin, the user should click on Tools->Manage Plugins. From the Plugin Manager, click Load New and navigate to the RapidCFD plugin. By clicking the Autoload checkbox, the plugin will load with ParaView start up in the future, see Figure 3. If you choose not to Auto Load, then you will need to load the plugin each time you start ParaView.



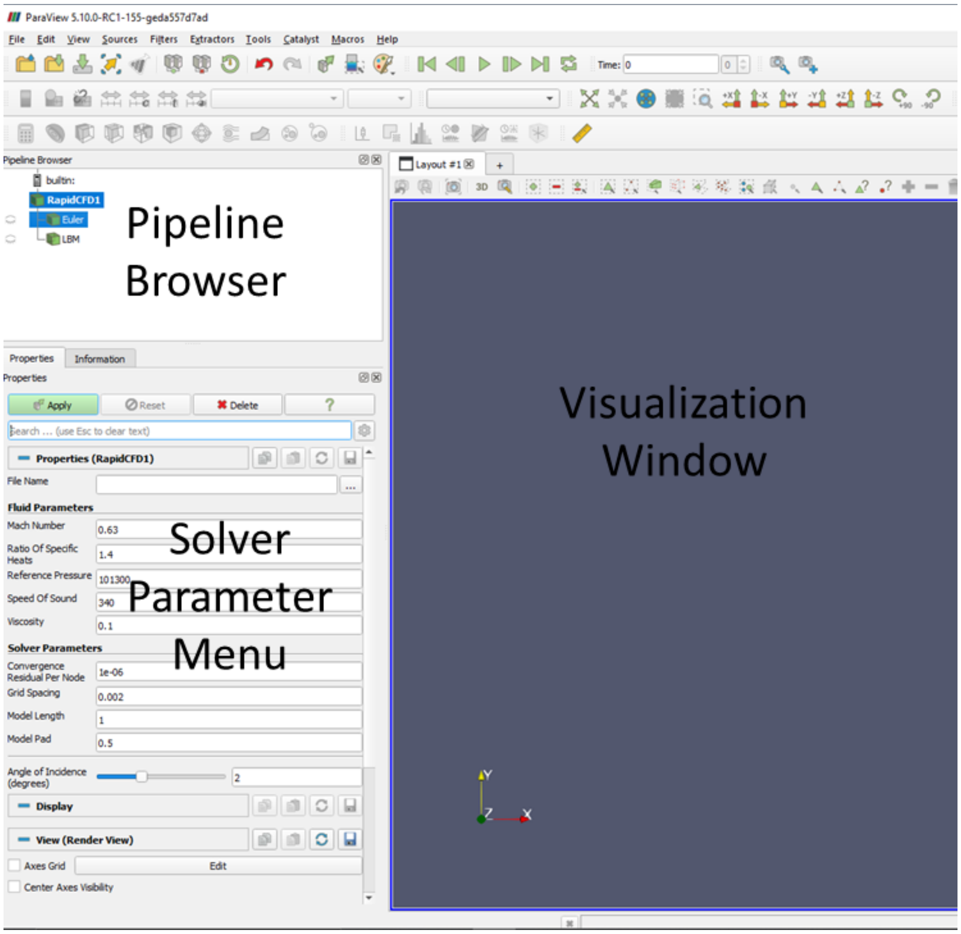
Local plugins are automatically searched for in C:/Projects/PV/pvb/bin/plugins.

Local Plugins

Name	Property
vtkPVInitializerPlugin	Loaded
AcceleratedAlgorithms	Not Loaded
AnalyzeNIfTIReaderWriter	Not Loaded
ArrowGlyph	Not Loaded
BagPlotViewsAndFilters	Not Loaded
CDIReader	Not Loaded
Datamine	Not Loaded
DigitalRockPhysics	Not Loaded
EmbossingRepresentations	Not Loaded
ExplicitStructuredGrid	Not Loaded
FlipBookPlugin	Not Loaded
GMVReader	Not Loaded
GeodesicMeasurement	Not Loaded
HyperTreeGridADR	Not Loaded
LagrangianParticleTracker	Not Loaded
LegacyExodusReader	Not Loaded
Moments	Not Loaded
MooseXfemClip	Not Loaded
NodeEditor	Not Loaded
NonOrthogonalSource	Not Loaded
PanoramicProjectionView	Not Loaded
SLACTools	Not Loaded
SaveStateAndScreenshot	Not Loaded
StreamLinesRepresentation	Not Loaded
StreamingParticles	Not Loaded
SurfaceLIC	Not Loaded
ThickenLayeredCells	Not Loaded
RapidCFDPlugin	Loaded
Version	1.0
Description	A ParaView plugin that provides an interface to the fast LBM correction method.
Location	C:\D\RapidCFDPlugin\bin\paraview-5.10\plugins\RapidCFDPlugin\Debug\RapidCFDPlugin.dll
Required Plugins	
Status	Loaded
Auto Load	☑

Figure 3. From the Plugin Manager, the user can click Load New…and navigate to the RapidCFDPlugin.dll. Clicking the Autoload checkbox will load the plugin at ParaView start up.

To use the loaded plugin, click on the Sources menu and select the Search option. Type RapidCFD into the search box, select RapidCFD from the list, and click Enter. The RapidCFD plugin will open in the Pipeline Browser and the Solver Parameter menu will be visible, as shown in **Figure *4***.



File Name

Fluid Parameters

Mach Number: 0.63
Ratio Of Specific Heats: 1.4
Reference Pressure: 101300
Speed Of Sound: 340
Viscosity: 0.1

Solver Parameters

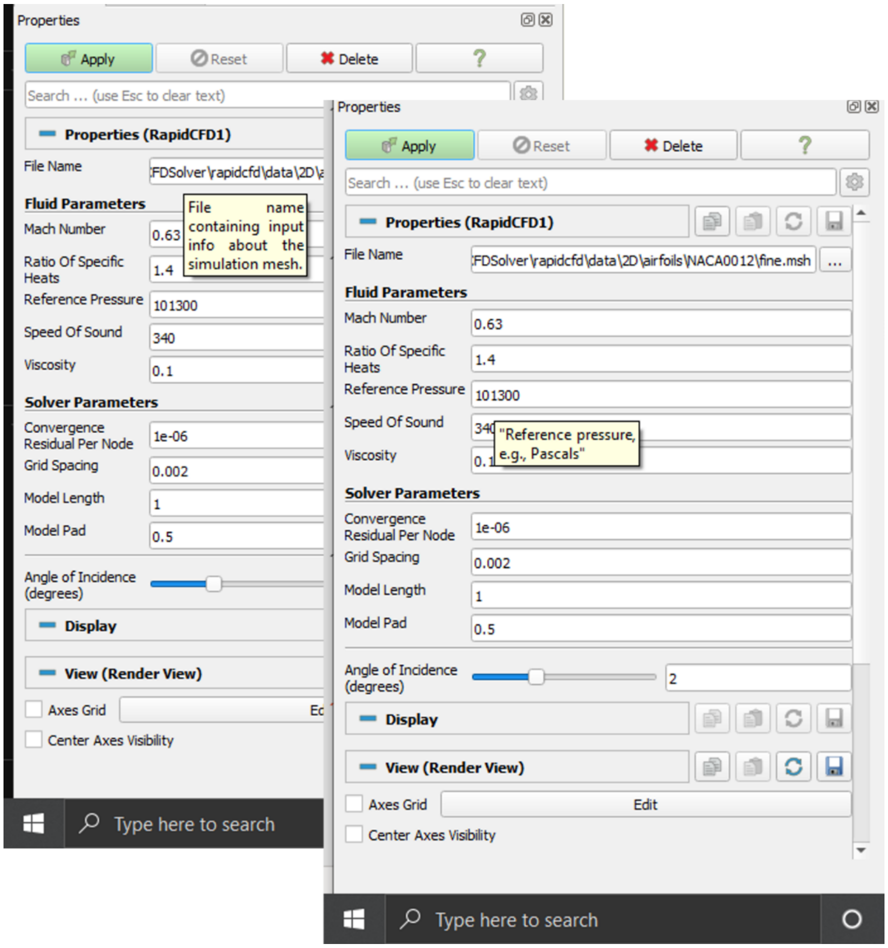
Convergence Residual Per Node: 1e-06
Grid Spacing: 0.002
Model Length: 1
Model Pad: 0.5

Angle of Incidence (degrees): 2

☐ Axes Grid
☐ Center Axes Visibility

Figure 4. The RapidCFD plugin is shown in the Pipeline Browser with a Solver Parameter menu in the lower portion of the menu and the visualization is shown in the right-hand portion of ParaView.

These parameters are divided into Fluid Parameters, Solver Parameters, and Angle of Incidence. A tool tip appears when hovering over the parameter name with tips about its use and/or units, as shown in **Figure *5***. A volumetric mesh can be loaded for analysis, by entering the address of the mesh into the File Name box or navigating to its location by clicking on the … button. Once the mesh is loaded, the remaining parameters can be specified. To execute the Euler and LBM solver, the user clicks the Apply button. This will execute one iteration of the Euler reduced order model, then the LBM, which improves on the Euler solution.



Properties (RapidCFD1)

File Name FDSolver\rapidcfd\data\2D\a

File name containing input info about the simulation mesh.

Fluid Parameters

Mach Number 0.63

Ratio Of Specific Heats 1.4

Reference Pressure 101300

Speed Of Sound 340

Viscosity 0.1

Solver Parameters

Convergence Residual Per Node 1e-06

Grid Spacing 0.002

Model Length 1

Model Pad 0.5

Angle of Incidence (degrees)

☐ Axes Grid

☐ Center Axes Visibility

Properties (RapidCFD1)

File Name :FDSolver\rapidcfd\data\2D\airfoils\NACA0012\fine.msh

Fluid Parameters

Mach Number 0.63

Ratio Of Specific Heats 1.4

Reference Pressure 101300

Speed Of Sound 340

"Reference pressure, e.g., Pascals"

Viscosity 0.1

Solver Parameters

Convergence Residual Per Node 1e-06

Grid Spacing 0.002

Model Length 1

Model Pad 0.5

Angle of Incidence (degrees) 2

☐ Axes Grid

☐ Center Axes Visibility

Figure 5. The parameters are divided into solver and fluid parameters and the angle of incidence. Tool tips appear to provide guidance and unit hints for populating the parameters.

The simulation will calculate in seconds and appear in the visualization window for further analysis. We are also working to automate the calculation of life coefficients and pressure distributions. The results and visualization will be discussed further in the following section.

6. Results and Discussion

Our research showed promising results for the Euler-LBM solution. The Euler method was rewritten in CUDA C to be executed on GPUs, exploiting the inherent parallelism of the residual distribution approach in which the main computational workload is done in each individual cell (triangle in two-dimensional (2D), tetrahedron in three-dimensional (3D)). The resulting optimized code executes approximately 20 times faster than the previous CPU implementation. Two-dimensional airfoil inviscid flows are determined in 0.3-0.5 seconds. Further optimization has been carried on the two-dimensional IGLBM code, and the overall 2D viscid-inviscid procedure is now complete. The preliminary computation time for the 3D geometry is less than five minutes.

Two-Dimensional (2D) Results

A first validation has been carried out for incompressible (Mach M=0.15, standard atmospheric conditions) flow around the NACA0012 airfoil at various angles of incidence for which there are available wind tunnel results [16]. Figure 6, Figure 7, and Figure 8 show the inviscid prediction and viscid correction at angles of incidence of 0, 10, and 14 degrees, respectively. The viscid IGLBM correction in general carries out slight modification of the pressure distribution, with significant modification of the inviscid prediction only at the trailing edge where vortex shedding is captured.

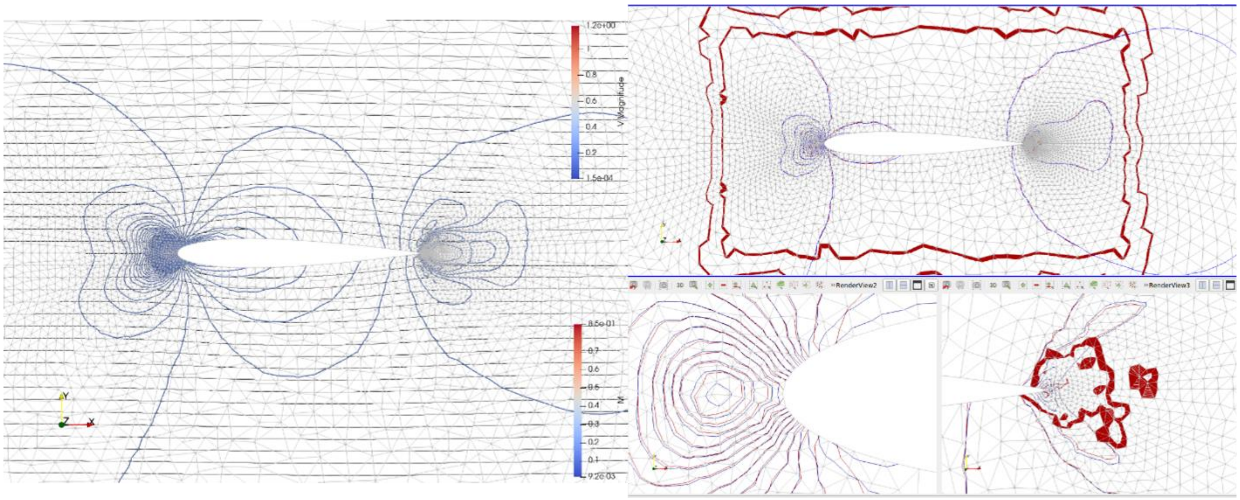


Figure 6. NACA0012 airfoil at 0 degrees incidence. Left: Isocontours of pressure and streamlines overlaid on triangular grid from inviscid Euler procedure. Right: Viscid IGLBM correction carried out in the airfoil nearfield region. Insets show viscid correction of pressure contours (red) compared to inviscid prediction (blue) in leading and trailing edge regions. The trailing edge exhibits vortex shedding.

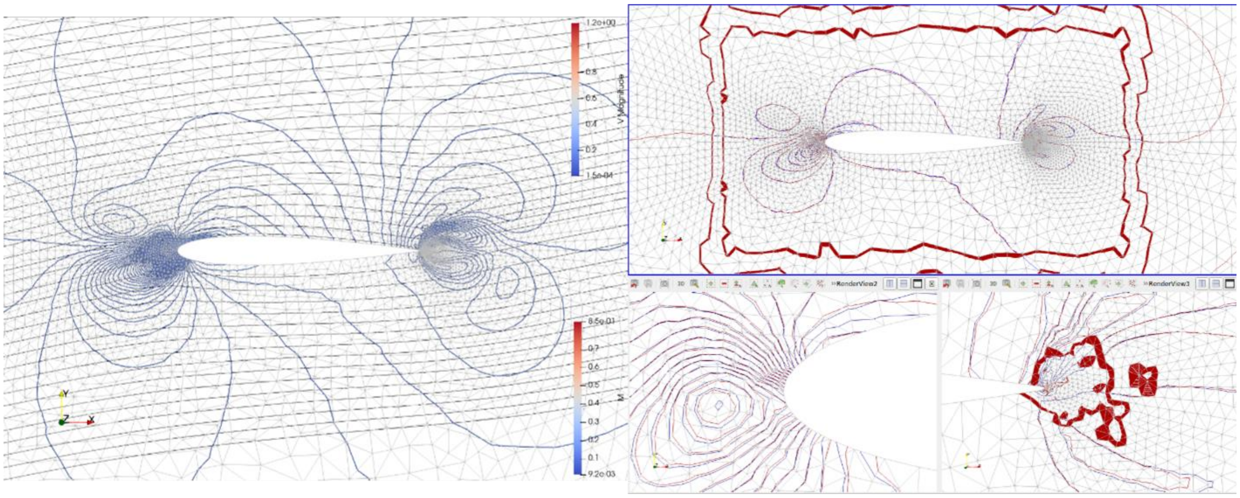


Figure 7. NACA0012 airfoil at 10 degrees incidence. Left: Isocontours of pressure and streamlines overlaid on triangular grid from inviscid Euler procedure. Right: Viscid IGLBM correction carried out in the airfoil nearfield region. Insets show viscid correction of pressure contours (red) compared to inviscid prediction (blue) in leading and trailing edge regions. The trailing edge exhibits vortex shedding.

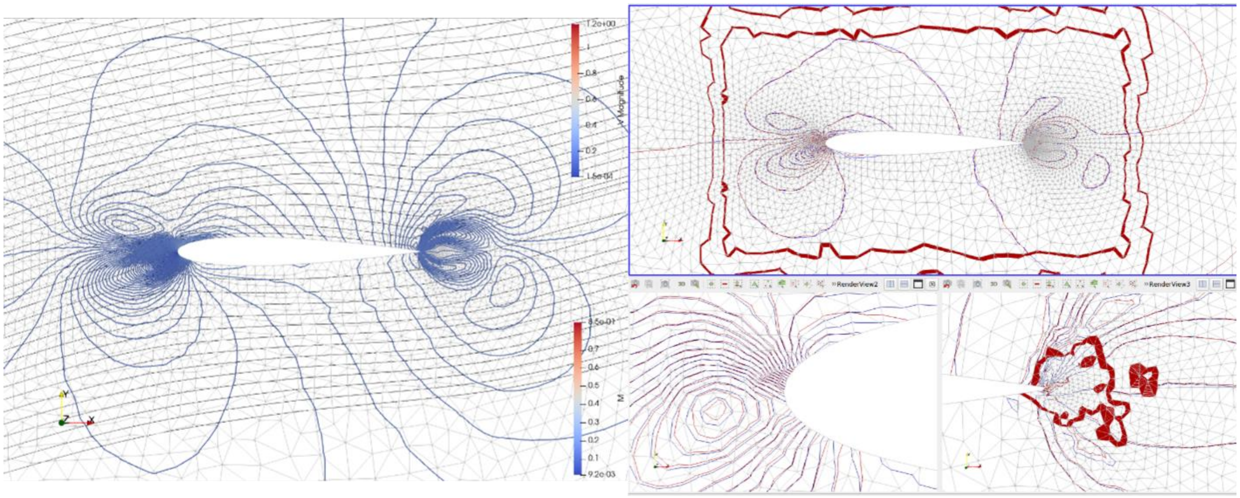


Figure 8. NACA0012 airfoil at 14 degrees incidence. Left: Isocontours of pressure and streamlines overlaid on triangular grid from inviscid Euler procedure. Right: Viscid IGLBM correction carried out in the airfoil near field region. Insets show viscid correction of pressure contours (red) compared to inviscid prediction (blue) in leading and trailing edge regions. The trailing edge exhibits vortex shedding.

The calculated pressure distribution on the airfoil is compared to experimental results in Figure 9, and the polar lift coefficient versus incidence angle is shown in Figure 10. The two-dimensional solution will be refined in Phase II for higher Mach and Reynolds numbers.

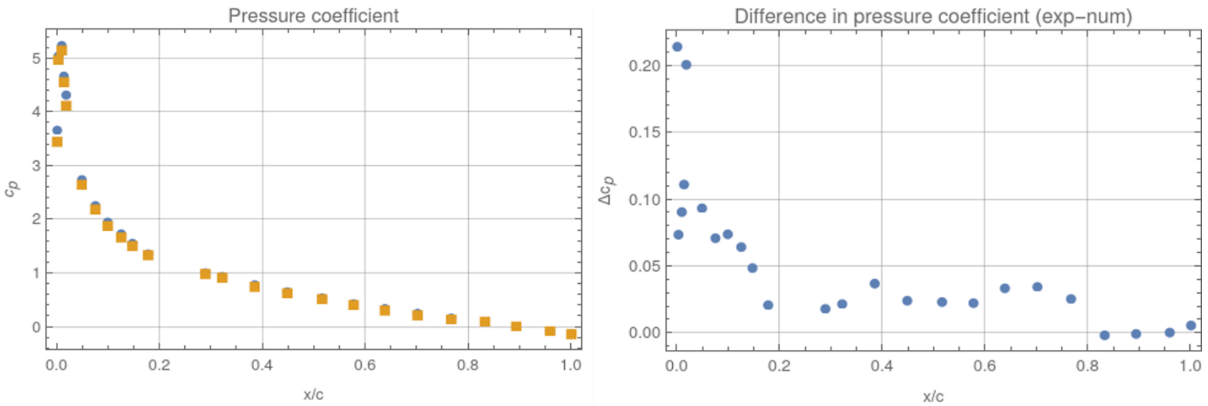


Figure 9. NACA0012 airfoil at 10 degrees incidence, Mach = 0.15, chord Reynolds = 106. Left: Comparison of experimental (blue circles) with numerical prediction (orange squares). Numerical procedure slightly under predicts observed pressure distribution, with maximum error observed at leading edge of 4%. Right: difference in experimental minus computational result.

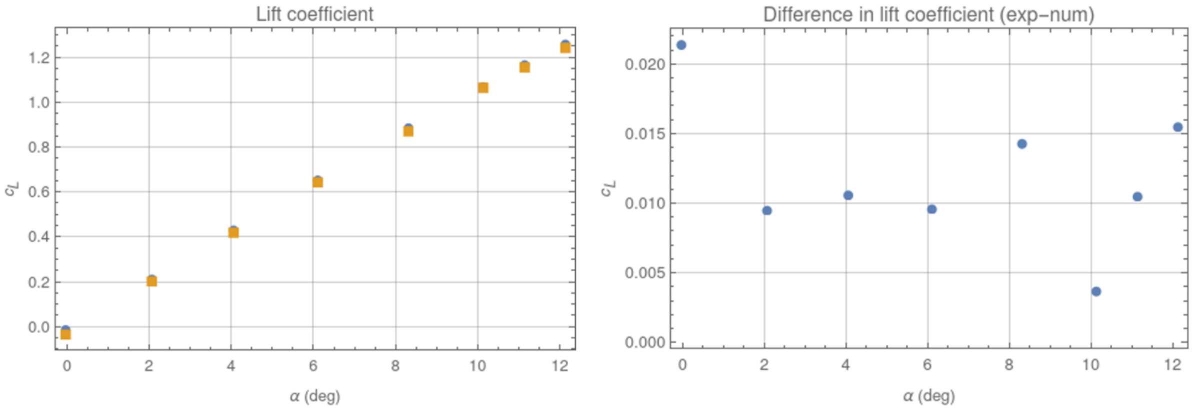


Figure 10. Left: Comparison of experimental lift coefficient (blue circles) to computational result (orange squares) shows slight underprediction of lift coefficient with maximum error of 4%. Right: difference in experimental minus computed lift coefficient.

3D Results

Three-dimensional versions of both the Euler RDS and IGLBM have been implemented and initial tests have been carried out on the ONERA M6 wing. The principal goal of the Phase I project to demonstrate efficiency has been attained. A low-resolution computation with $N = 34 \times 10^3$ nodes and $E = 194 \times 10^3$ cells was computed in approximately one minute by the Euler RDS scheme, with a viscous correction with a lattice step size of $5\, y^+$ (in wall units) is found under one minute (Fig.). A high-resolution computation with $N = 124 \times 10^3$ nodes and $E = 918 \times 10^3$ cells is computed in approximately ten minutes by the Euler RDS scheme, with a viscous correction again with a lattice step size of $5\, y^+$ (in wall units) is found under five minutes (Fig.). At the time of the first draft of this report, comparison to alternative computational approaches and experimental results is still ongoing. However, preliminary results are shown in **Figure *11*** and **Figure *12***.

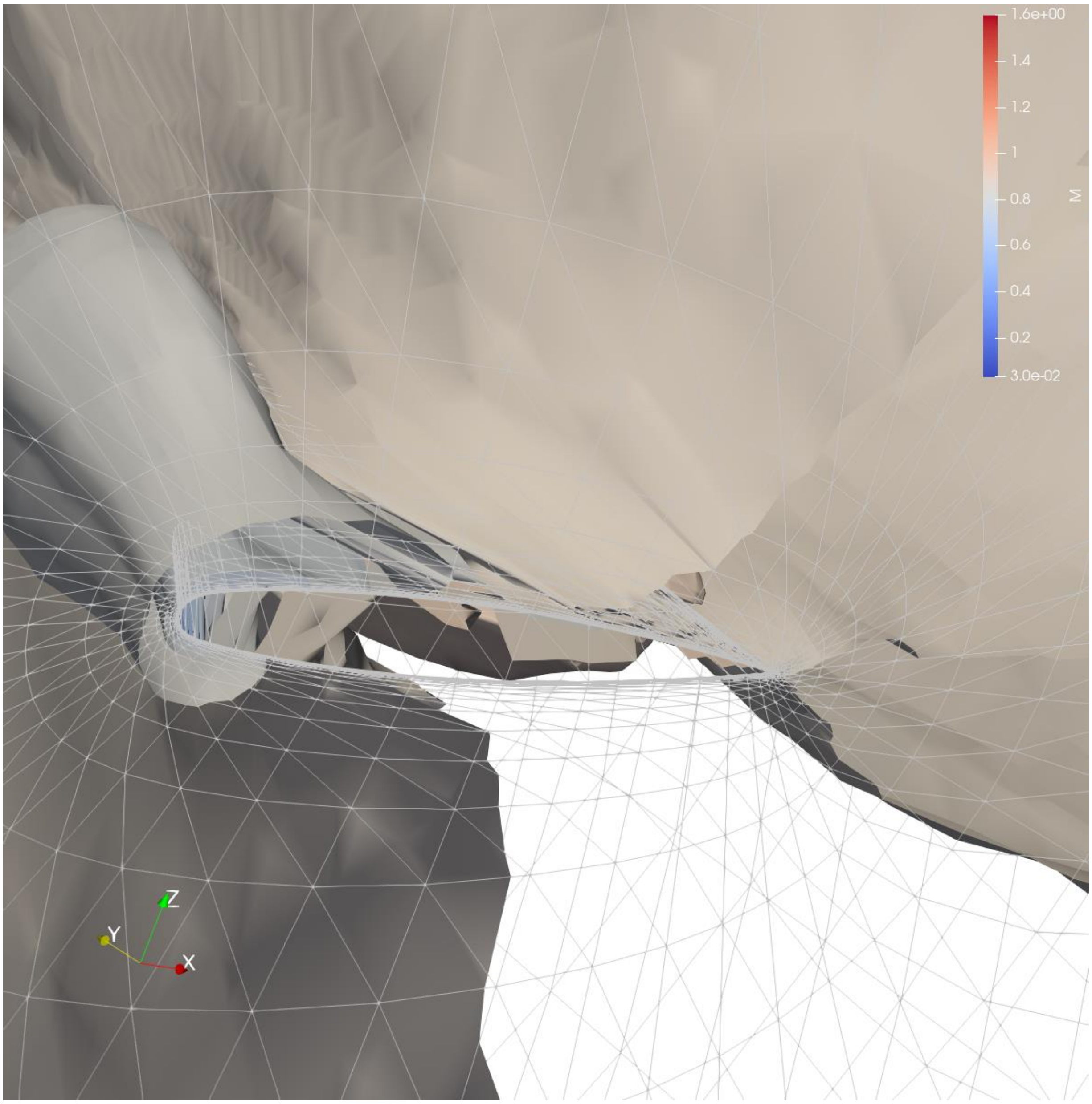


Figure 11. ONERA M6 wing at upstream Ma=0.85 and zero incidence. Mach number contours using low-resolution grid.

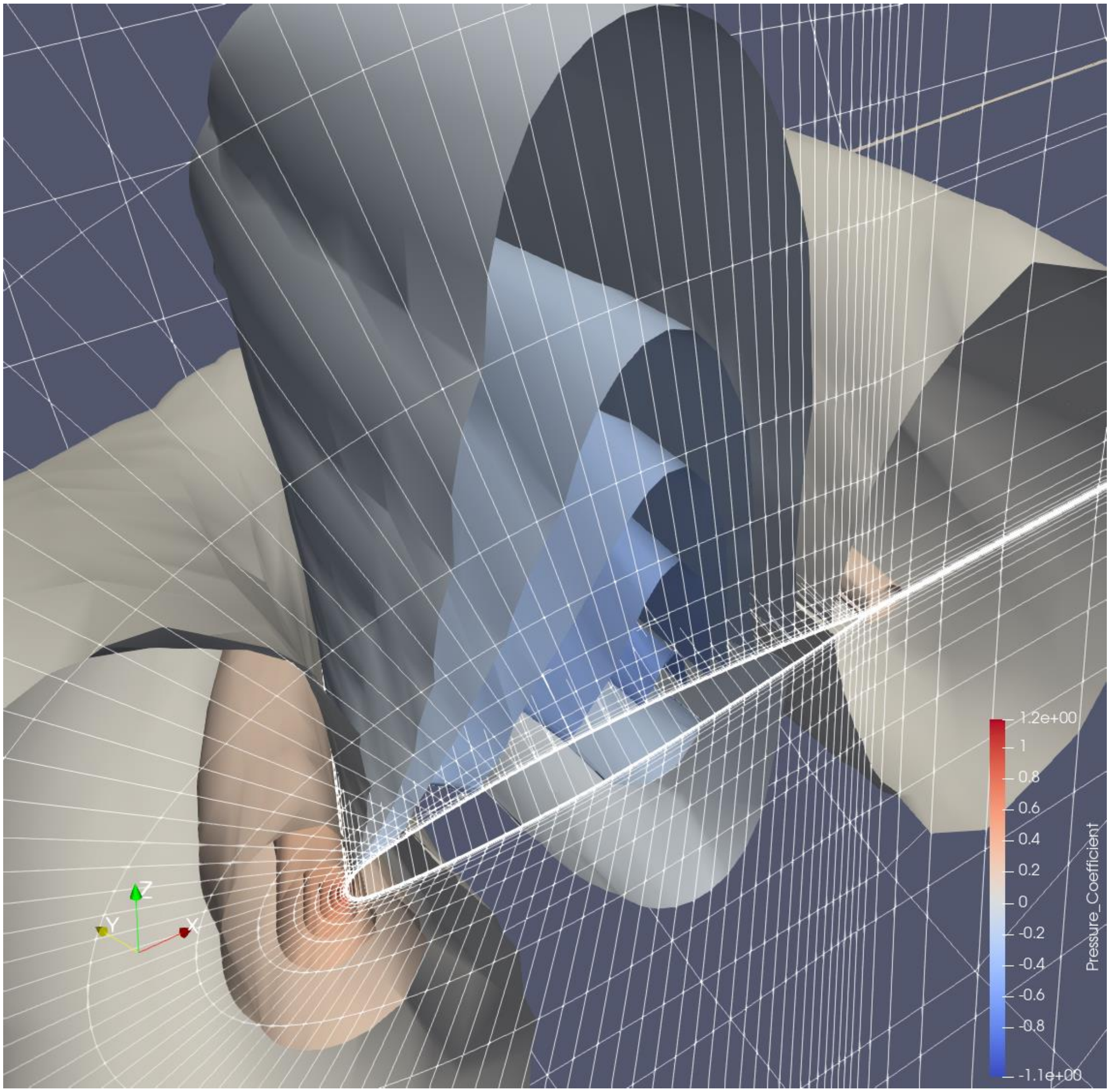


Figure 12. ONERA M6 wing at upstream Ma=0.85 and zero incidence. Pressure coefficient using high-resolution grid.

Plug In Visualizations

All of the visualization in this report were generated in the ParaView visualization window. These results show the ability to leverage the ParaView visualization platform to create a custom workflow for the aerospace government and industry stakeholder.

5. Conclusion

Reduced-Order CFD Simulation

The goals of the Phase I STTR were to present the feasibility of the reduced-order model, comprised of the Euler and LBM algorithms, to estimate the aerodynamic profiles of common geometries accurately and efficiently. The results show a significant improvement in computation time with basic geometries calculating in less than one second for 2D and less than one minute for 3D. In Phase II, we will propose an immediate validation stage to perform a more complete analysis on the accuracy of the results and identify areas for improvement in the calculation. Kitware has extensive experience creating validation test suites that automate the validation process, creating error tables and plots, assessing computational efficiency, and generating validation reports. We will develop this for the RapidCFD Solver that will be used throughout Phase II to constantly evaluation the performance and efficiency and track improvement. Also in Phase II, UNC will complete fundamental research to fully realize the inviscid-viscid interaction, starting with boundary layer thickening and continuing to model the separation bubbles and wakes. In Phase I, a uniform lattice was applied throughout a flow domain centered on the aerodynamic body of interest. In Phase II, an adaptive grid will be developed and applied in the LBM simulation, both for improved results and for optimization of computational efficiency. In Phase II, fundamental research on the relaxation parameter will also be completed. Instead of relaxation towards thermal equilibrium as is typical in most current LBM approaches, a modification of the probability distribution functions based upon geodesic transport on information manifolds will be employed. This allows partial relaxation, as is typically the case in high-speed flows, particularly near shocks. Finally, an additional feature to appropriately model turbulence will be developed as part of this fundamental research. Though capable of full resolution direct numerical simulation (DNS), the LBM is more economically executed in conjunction with a sub grid turbulence model. During Phase II, eddy-viscosity and one-equation turbulence models shall be adapted to execution within the kinetic formulation of LBMs, including Smagorinsky and

Spalart-Allmaras models [17] – [19].

Customized Workflow and Plugin Development

The plugin results show the ability to leverage the ParaView visualization platform to create a custom workflow for the aerospace government and industry stakeholder. The resultant interface is easy to use to specify parameters and execute simulations. The easy availability of data analysis tools in ParaView makes it quick to evaluate the results of the RapidCFD Solver. While this was a solid prototype for a proof-of-concept, in Phase II we plan to create a seamless custom workflow that addresses the needs of the aerospace industry. In Phase II, we will improve on our prototype by performing a task analysis with industry experts to define the requirements for a user interface, an improved user interface with additional simulation and visualization features, integration of volumetric meshing tools to allow users to begin from a surface mesh, automation of additional data analysis components, and ensemble simulation setup. Kitware will create this custom application using our Computational Model Builder Software built on ParaView in conjunction with Catalyst which is designed to enable remote simulation and ensemble simulation. By connecting these two powerful tools, we will create a custom application for an optimal workflow.

6. References

[1] R. Agarwal, "COMPUTATIONAL FLUID DYNAMICS OF WHOLE-BODY AIRCRAFT," *Annual Review of Fluid Mechanics*, vol. 31, no. 1, pp. 125–169, Jan. 1999, doi: 10.1146/annurev.fluid.31.1.125.

[2] F. Hussain and A. K. M. Fazle Hussain, "Coherent structures and turbulence Microfluidic biomechanical differentiation of tumor metastatic cells View project Turbulence View project Coherent structures and turbulence," *Article in Journal of Fluid Mechanics*, vol. 173, pp. 303–356, 1987, doi: 10.1017/S0022112086001192.

[3] J. Borggaard, T. Iliescu, and Z. Wang, "Artificial viscosity proper orthogonal decomposition," *Mathematical and Computer Modelling*, vol. 53, no. 1–2, pp. 269–279, Jan. 2011, doi: 10.1016/J.MCM.2010.08.015.

[4] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The Lattice Boltzmann Method*. Cham: Springer International Publishing, 2017. Doi: 10.1007/978-3-319-44649-3.

[5] K. Suzuki, K. Minami, and T. Inamuro, "Lift and thrust generation by a butterfly-like flapping wing-body model: immersed boundary-lattice Boltzmann simulations." Accessed: Jun. 25, 2019. [Online]. Available: https://core.ac.uk/download/pdf/148783466.pdf

[6] R. Satti, Y. Li, R. Shock, and S. Noelting, "Unsteady Flow Analysis of a Multi-Element Airfoil Using Lattice Boltzmann Method," *AIAA Journal*, vol. 50, no. 9, pp. 1805–1816, Sep. 2012, doi: 10.2514/1.J050906.

[7] S. Amari and H. Nagaoka, *Methods of Information Geometry*. Tokyo: Iwanami Shoten, 1993. Accessed: Jun. 25, 2019. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=vc2FWSo7wLUC&oi=fnd&pg=PR7&ots=4HsqHz45I-&sig=p5KnRbRW2jYEWz-yi9IpdIZVJuo#v=onepage&q&f=false

[8] F. T. Johnson, E. N. Tinoco, and N. J. Yu, "Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle," 2005, doi: 10.1016/j.compfluid.2004.06.005.

[9] "ParaView." https://www.paraview.org/ (accessed Nov. 23, 2021).

[10] J. Ahrens, "ParaView: An End-User Tool for Large Data Visualization." Accessed: Jun. 18, 2019. [Online]. Available: https://www.researchgate.net/publication/247111133

[11] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. USA: Kitware, Inc., 2015. Accessed: Jun. 26, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=2789330

[12] "Computational Model Builder." https://www.computationalmodelbuilder.org/ (accessed Nov. 23, 2021).

[13] A. Hines *et al.*, "Computational Model Builder (CMB): A Cross-Platform Suite of Tools for Model Creation and Setup," in *2009 DoD High Performance Computing Modernization Program Users Group Conference*, Jun. 2009, pp. 370–373. Doi: 10.1109/HPCMP-UGC.2009.60.

[14] B. Obara, “Developing Casting Simulations in CMB – Kitware Blog,” *Kitware Blog*, 2017. https://blog.kitware.com/developing-casting-simulations-in-cmb/ (accessed Jun. 26, 2019).

[15] “In situ | ParaView.” https://www.paraview.org/in-situ/ (accessed Nov. 23, 2021).

[16] C. D. Harris, “Two-dimensional aerodynamic characteristics of the NACA 0012 airfoil in the Langley 8 foot transonic pressure tunnel.” 1981.

[17] X.-P. Chen, “Engineering Applications of Computational Fluid Mechanics Applications of Lattice Boltzmann Method to Turbulent Flow Around Two-Dimensional Airfoil APPLICATIONS OF LATTICE BOLTZMANN METHOD TO TURBULENT FLOW AROUND TWO-DIMENSIONAL AIRFOIL,” *Engineering Applications of Computational Fluid Mechanics*, vol. 6, no. 4, pp. 572–580, 2012, doi: 10.1080/19942060.2012.11015443.

[18] N. Pellerin, S. Leclaire, and M. Reggio, “An implementation of the Spalart–Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows,” *Computers & Mathematics with Applications*, vol. 70, no. 12, pp. 3001–3018, Dec. 2015, doi: 10.1016/J.CAMWA.2015.10.006.

[19] P. Sagaut, “Toward advanced subgrid models for Lattice-Boltzmann-based Large-eddy simulation: Theoretical formulations,” *Computers & Mathematics with Applications*, vol. 59, no. 7, pp. 2194–2199, Apr. 2010, doi: 10.1016/J.CAMWA.2009.08.051.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

2D	Two Dimensional
3D	Three Dimensional
AFRL	Air Force Research Laboratory
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
LBM	Lattice Boltzmann Method
SBIR	Small Business Innovation Research.
UNC	University of North Carolina – Chapel Hill
CMB	Computational Model Builder
IG	Information Geometry
RDS	Residual Distribution Scheme
NS	Navier-Stokes
RANS	Reynold’s Averaged Navier-Stokes
PDF	Probability Density Function
BGK	Bhatnagar-Gross-Krook
STTR	Small Business Technology Transfer
IGLBM	Information Geometry accelerated Lattice Boltzmann Method
GPU	Graphical Processing Unit
PI	Principal Investigator