On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools

B.C. Vermeire *, F.D. Witherden, P.E. Vincent

Department of Aeronautics, Imperial College London, SW7 2AZ, United Kingdom

Abstract

First- and second-order accurate numerical methods, implemented for CPUs, underpin the majority of industrial CFD solvers. Whilst this technology has proven very successful at solving steady-state problems via a Reynolds Averaged Navier–Stokes approach, its utility for undertaking scale-resolving simulations of unsteady flows is less clear. High-order methods for unstructured grids and GPU accelerators have been proposed as an enabling technology for unsteady scale-resolving simulations of flow over complex geometries. In this study we systematically compare accuracy and cost of the high-order Flux Reconstruction solver PyFR running on GPUs and the industry-standard solver STAR-CCM+ running on CPUs when applied to a range of unsteady flow problems. Specifically, we perform comparisons of accuracy and cost for isentropic vortex advection (EV), decay of the Taylor–Green vortex (TGV), turbulent flow over a circular cylinder, and turbulent flow over an SD7003 aerofoil. We consider two configurations of STAR-CCM+: a second-order configuration, and a third-order configuration, where the latter was recommended by CD-adapco for more effective computation of unsteady flow problems. Results from both PyFR and STAR-CCM+ demonstrate that third-order schemes can be more accurate than second-order schemes for a given cost e.g. going from second- to third-order, the PyFR simulations of the EV and TGV achieve 75× and 3× error reduction respectively for the same or reduced cost, and STAR-CCM+ simulations of the cylinder recovered wake statistics significantly more accurately for only twice the cost. Moreover, advancing to higher-order schemes on GPUs with PyFR was found to offer even further accuracy vs. cost benefits relative to industry-standard tools.

Keywords: GPU, High-order, Flux reconstruction, Turbulent Flows, Comparison

1. Introduction

Industrial computational fluid dynamics (CFD) applications require numerical methods that are concurrently accurate and low-cost for a wide range of applications. These methods must be flexible enough to handle complex geometries, which is usually achieved via unstructured mixed element meshes. Conventional unstructured CFD solvers typically employ second-order accurate spatial discretizations. These second-order schemes were developed primarily in the 1970s to 1990s to improve upon the observed accuracy limitations of first-order methods [1]. While second-order schemes have been
successful for steady state solutions, such as using the Reynolds Averaged Navier–Stokes (RANS) approach, there is evidence that higher-order schemes can be more accurate for scale-resolving simulations of unsteady flows [1]. Recently, there has been a surge in the development of high-order unstructured schemes that are at least third-order accurate in space. Such methods have been the focus of ongoing research, since there is evidence they can provide improved accuracy at reduced computational cost for a range of applications, when compared to conventional second-order schemes [1]. Such high-order unstructured schemes include the discontinuous Galerkin (DG) [2,3], spectral volume (SV) [4], and spectral difference (SD) [5,6] methods, amongst others. One particular high-order unstructured method is the flux reconstruction (FR), or correction procedure via reconstruction (CPR), scheme first introduced by Huynh [7]. This scheme is particularly appealing as it unifies several high-order unstructured numerical methods within a common framework. Depending on the choice of correction function one can recover the collocation based nodal DG, SV, or SD methods, at least for the case of linear equations [7,8]. In fact, a wide range of schemes can be generated that are provably stable for all orders of accuracy [9]. The FR scheme was subsequently extended to mixed element types by Wang and Gao [8], three-dimensional problems by Haga and Wang [10], and tetrahedra by Williams and Jameson [11]. These extensions have allowed the FR scheme to be used successfully for the simulation of transitional and turbulent flows via scale resolving simulations, such as large eddy simulation (LES) and direct numerical simulation (DNS) [12–14].

Along with recent advancements in numerical methods, there have been significant changes in the types of hardware available for scientific computing. Conventional CFD solvers have been written to run on large-scale shared and distributed memory clusters of central processing units (CPUs), each with a small number of scalar computing cores per device. However, the introduction of accelerator hardware, such as graphical processing units (GPUs), has led to extreme levels of parallelism with several thousand compute “cores” per device. One advantage of GPU computing is that, due to such high levels of parallelism, GPUs are typically capable of achieving much higher theoretical peak performance than CPUs at similar price points. This makes GPUs appealing for performing CFD simulations, which often require large financial investments in computing hardware and associated infrastructure.

The objective of the current work is to quantify the cost and accuracy benefits that can be expected from using high-order unstructured schemes deployed on GPUs for scale-resolving simulations of unsteady flows. This will be performed via a comparison of the high-order accurate open-source solver PyFR [15] running on GPUs with the industry-standard solver STAR-CCM+ [16] running on CPUs for four relevant unsteady flow problems. PyFR was developed to leverage synergies between high-order accurate FR schemes and GPU hardware [15]. We consider two configurations of STAR-CCM+: a second-order configuration, and a third-order configuration, where the latter was recommended by CD-adapco for more effective computation of unsteady flow problems. Full configurations for all STAR-CCM+ simulations are provided as electronic supplementary material. We will compare these configurations on a set of test cases including a benchmark isentropic vortex problem and three cases designed to test the solvers for scale resolving simulations of turbulent flows. These are the types of problems that current industry-standard tools are known to find challenging [17], and for which high-order schemes have shown particular promise [1]. The utility of high-order methods in other flow-regimes, such as those involving shocks or discontinuities, is still an open research topic. In this study we are interested in quantifying the relative cost of each solver in terms of total resource utilization on equivalent era hardware, as well as quantitative accuracy measurements based on suitable error metrics, for the types of problems that high-order methods have shown promise.

The paper is structured as follows. In section 2 we will briefly discuss the software packages being compared. In section 3 we will discuss the hardware configurations each solver is being run on, including a comparison of monetary cost and theoretical performance statistics. In section 4 we will discuss possible performance metrics for comparison and, in particular, the resource utilization metric used in this study. In section 5 we will present several test cases and results obtained with both PyFR and STAR-CCM+. In particular, we are interested in isentropic vortex advection, Taylor–Green vortex breakdown, turbulent flow over a circular cylinder, and turbulent flow over an SD7003 aerofoil. Finally, in section 6 we will present conclusions based on these comparisons and discuss implications for the adoption of high-order unstructured schemes on GPUs for industrial CFD.

2. Solvers

2.1. PyFR

PyFR [15] (http://www.pyfr.org/) is an open-source Python-based framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction (FR) scheme of Huynh [7]. PyFR is platform portable via the use of a domain specific language based on Mako templates. This means PyFR can run on AMD or NVIDIA GPUs, as well as traditional CPUs. A brief summary of the functionality of PyFR is given in Table 1, which includes mixed-element unstructured meshes with arbitrary order schemes. Since PyFR is platform portable, it can run on CPUs using OpenCL or C/OpenMP, NVIDIA GPUs using CUDA or OpenCL, AMD GPUs using OpenCL, or heterogeneous systems consisting of a mixture of these hardware types [18]. For the current study we are running PyFR version 0.3.0 on NVIDIA GPUs using the CUDA backend, which utilizes cuBLAS for matrix multiplications. We will also use an experimental version of PyFR 0.3.0 that utilizes the open source linear algebra package GiMMiK [19]. A patch to go from PyFR v0.3.0 to this experimental version has been provided as electronic supplementary material. GiMMiK generates bespoke kernels, i.e. written specifically for each particular
operator matrix, at compile time to accelerate matrix multiplication routines. The cost of PyFR 0.3.0 with GiMMiK will be compared against the release version of PyFR 0.3.0 to evaluate its advantages for sparse operator matrices.

### 2.2. STAR-CCM+

STAR-CCM+ [16] is a CFD and multiphysics solution package based on the finite volume method. It includes a CAD package for generating geometry, meshing routines for generating various mesh types including tetrahedral and polyhedral, and a multiphysics flow solver. A short summary of the functionality of STAR-CCM+ is given in Table 2. It supports first, second, and third-order schemes in space. In addition to an explicit method, STAR-CCM+ includes support for implicit temporal schemes. Implicit schemes allow for larger global time-steps at the expense of additional inner sweeps to converge the unsteady residual. For the current study we use the double precision version STAR-CCM+9.06.011-R8. This version is used since PyFR also runs in full double precision, unlike the mixed precision version of STAR-CCM+.

### 3. Hardware

PyFR is run on either a single or multi-GPU configuration of the NVIDIA Tesla K20c. For running STAR-CCM+ we use either a single Intel Xeon E5-2697 v2 CPU, or a cluster consisting of InfiniBand interconnected Intel Xeon X5650 CPUs. The specifications for these various pieces of hardware are provided in Table 3. The purchase price of the Tesla K20c and Xeon E5-2697 v2 are similar, however, the Tesla K20c has a significantly higher peak double precision floating point arithmetic rate and memory bandwidth. The Xeon X5650, while significantly cheaper than the Xeon E5-2697 v2, has a similar price to performance ratio when considering both the theoretical peak arithmetic rate and memory bandwidth.

### 4. Cost metrics

Several different cost metrics could be considered for comparing PyFR and STAR-CCM+ including hardware price, simulation wall-clock time, and energy consumption. In the recent high-order workshop TauBench was used as a normalization metric for total simulation runtime [1]. However, there is no GPU version of TauBench available for normalizing the PyFR simulations. Also, this approach does not take into account the price of different types of hardware. While energy consumption is a relevant performance metric, it relies heavily on system architecture, peripherals, cooling systems, and other design choices that are beyond the scope of the current study.
In the current study we introduce a cost metric referred to as resource utilization. This is measured as the product of the cost of the hardware being used for a simulation in £, and the amount of time that hardware has been utilized in seconds. This gives a cost metric with the units £ \times \text{seconds}. Therefore, resource utilization incorporates both the price to performance ratio of a given piece of hardware, and the ability of the solver to use it efficiently to complete a simulation in a given amount of time. This effectively normalizes the computational cost by the price of the hardware used.

Two fundamental constraints for CFD applications are the available budget for purchasing computer hardware and the maximum allowable time for a simulation to be completed. Depending on application requirements, most groups are limited by one of these two constraints. When the proposed resource utilization metric is constrained with a fixed capital expenditure budget it becomes directly correlated to total simulation time. If constrained by a maximum allowable simulation time, resource utilization becomes directly correlated to the required capital expenditure. Therefore, resource utilization is a useful measurement for two of the dominant constraints for CFD simulations, total upfront cost and total simulation time.

Any solver and hardware combination that completes a simulation with a comparatively lower resource utilization can be considered faster, if constrained by a hardware acquisition budget, or cheaper, if constrained by simulation time.

5. Test cases

5.1. Isentropic vortex advection

5.1.1. Background

Isentropic vortex advection is a commonly used test case for assessing the accuracy of flow solvers for unsteady inviscid flows using the Euler equations [1]. This problem has an exact analytical solution at all times, which is simply the advection of the steady vortex with the mean flow. This allows us to easily assess error introduced by the numerical scheme over long advection periods. The initial flow field for isentropic vortex advection is specified as [1,15]

\[
\begin{align*}
\rho &= \left[1 - \frac{S^2 M^2 (\gamma - 1) e^{2 f}}{8 \pi^2}\right]^{\frac{1}{\gamma - 1}}, \\
u &= \frac{S x e^f}{2 \pi R}, \\
v &= \frac{S y e^f}{2 \pi R}, \\
p &= \frac{\rho u^\gamma}{\gamma M^2},
\end{align*}
\]

where \( \rho \) is the density, \( u \) and \( v \) are the velocity components, \( p \) is the pressure, \( f = (1 - x^2 - y^2)/2R^2 \), \( S = 13.5 \) is the strength of the vortex, \( M = 0.4 \) is the free-stream Mach number, \( R = 1.5 \) is the radius, and \( \gamma = 1.4 \).

For PyFR we use a K20c GPU running a single partition. We use a 40 \times 40 two-dimensional domain with periodic boundary conditions on the upper and lower edges and Riemann invariant free stream boundaries on the left and right edges. This allows the vortex to advect indefinitely through the domain, while spurious waves are able to exit through the lateral boundaries. The simulations are run in total to \( t = 2000 \), which corresponds to 50\( t_c \) where \( t_c \) is a domain flow through time. A five-stage fourth-order adaptive Runge–Kutta scheme [20–22] is used for time stepping with maximum and relative error tolerances of \( 10^{-8} \). We consider \( P_1 \) to \( P_5 \) quadrilateral elements with a nominal 480\(^2\) solution points. The number of elements and solution points for each scheme are shown in Table 4. All but the \( P_4 \) simulation have the nominal number of degrees of freedom, while the \( P_2 \) simulation has slightly more due to constraints on the number of solution points per element. Solution and flux points are located at Gauss–Legendre points and Rusanov [15] fluxes are used at the interface between elements.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time</th>
<th>Elements</th>
<th>Solution points</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAR 2nd-order Explicit</td>
<td>480(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
<tr>
<td>STAR 2nd-order Implicit</td>
<td>480(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
<tr>
<td>STAR 2nd-order Explicit</td>
<td>960(^2)</td>
<td>960(^2)</td>
<td></td>
</tr>
<tr>
<td>STAR 2nd-order Implicit</td>
<td>960(^2)</td>
<td>960(^2)</td>
<td></td>
</tr>
<tr>
<td>STAR 3rd-order Implicit</td>
<td>480(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
<tr>
<td>PyFR ( P_1 ) Explicit</td>
<td>240(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
<tr>
<td>PyFR ( P_2 ) Explicit</td>
<td>160(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
<tr>
<td>PyFR ( P_3 ) Explicit</td>
<td>120(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
<tr>
<td>PyFR ( P_4 ) Explicit</td>
<td>100(^2)</td>
<td>500(^2)</td>
<td></td>
</tr>
<tr>
<td>PyFR ( P_5 ) Explicit</td>
<td>80(^2)</td>
<td>480(^2)</td>
<td></td>
</tr>
</tbody>
</table>
With STAR-CCM+ we use all 12 cores of the Intel Xeon E5-2697 v2 CPU with default partitioning. We also use a 40 \times 40 two-dimensional domain with periodic boundary conditions on the upper and lower edges. The left and right boundaries are specified as free stream, again to let spurious waves exit the domain. For the second-order configuration we use the coupled energy and flow solver settings. We use an explicit temporal scheme with an adaptive time step based on a fixed Courant number of 1.0. We also test the second-order implicit solver using a fixed time-step ten times greater than the average explicit step size. The ideal gas law is used as the equation of state with inviscid flow and a second-order spatial discretization. All other solver settings are left at their default values. For the third-order configuration a Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL) scheme is used with coupled energy and flow equations, the ideal gas law, and implicit time-stepping with a fixed time-step $\Delta t = 0.025$. Once again, the number of elements and solution points are given in Table 4. We perform one set of STAR-CCM+ simulations with the same total number of degrees of freedom as the PyFR results. A second set of simulations were also performed using the second-order configuration on a grid that was uniformly refined by a factor of two in each direction.

To evaluate the accuracy of each method, we consider the $L_2$ norm of the density error in a 4 \times 4 region at the center of the domain. This error is calculated each time the vortex returns to the origin as per Witherden et al. [15]. Therefore, the $L_2$ error is defined as:

$$
\sigma(t) = \sqrt{\frac{\int_{-2}^{2} \int_{-2}^{2} (\rho_t(x, t) - \rho_e(x, t))^2 \, dx}{2 \times 2}},
$$

where $\rho_t(x, t)$ is the numerical solution, $\rho_e(x, t)$ is the exact analytical solution, and $\sigma(t)$ is the error as a function of time. For PyFR these errors are extracted after each advection period. STAR-CCM+ does not allow for the solution to be exported at an exact time with the explicit flow solver, so the closest point in time is used instead and the exact solution is shifted to a corresponding spatial location to match. To get a good approximation of the true $L_2$ error we use a 196 point quadrature rule within each element.

5.1.2. Results

Contours of density for the PyFR $P_2$ and the 480$^2$ degree of freedom STAR-CCM+ simulations are shown in Fig. 1 at $t = t_c$, $t = 5t_c$, $t = 10t_c$, and $t = 50t_c$. It is evident that all three simulations start with the same initial condition at $t = 0$. Some small stepping is apparent in both STAR-CCM+ initial conditions due to the projection of the smooth initial solution onto the piecewise constant basis used by the finite volume scheme. For PyFR $P_2$ all results are qualitatively consistent with the exact initial condition, even after 50 flow through times. The results using the second-order STAR-CCM+ configuration at $t = t_c$ already show some diffusion, which is more pronounced by $t = 5t_c$ and asymmetrical in nature. By $t = 50t_c$ the second-order STAR-CCM+ results are not consistent with the exact solution. The low density vortex core has broken up and been dispersed to the left hand side of the domain, suggesting a non-linear build up of error at the later stages of the simulation. The third-order STAR-CCM+ configuration has significantly less dissipation than the second-order configuration. However, by $t = 50t_c$, the vortex has moved up and to the left of the origin.

Plots of the $L_2$ norm of the density error against resource utilization are shown in Fig. 2 to Fig. 4 for $t = t_c$, $t = 5t_c$, and $t = 50t_c$, respectively, for all simulations. After one flow through of the domain, as shown in Fig. 2, all of the PyFR simulations outperform all of the STAR-CCM+ simulations in terms of resource utilization by approximately an order of magnitude. The simulations with GIMMIIK outperform them by an even greater margin. The PyFR simulations are all more accurate, with the $P_2$ scheme $\approx 5$ orders of magnitude more accurate than STAR-CCM+. This trend persists at $t = 5t_c$ and $t = 50t_c$, the PyFR simulations are approximately an order of magnitude cheaper than the 480$^2$ degree of freedom STAR-CCM+ simulations and are significantly more accurate. Interestingly, the PyFR $P_1$ to $P_3$ simulations require approximately the same resource utilization, suggesting greater accuracy can be achieved for no additional computational cost. Also, we find that the PyFR simulations using GIMMIIK are between 20% and 35% less costly than the simulations without it, depending on the order of accuracy.

We also observe that simulations using the second-order STAR-CCM+ configuration with implicit time-stepping have significantly more numerical error than the explicit schemes, but are less expensive due to the increased allowable time-step size. However, this increase in error is large enough that by $t = 5t_c$ the implicit schemes have saturated to the maximum error level at $\sigma \approx 1E0$. Increasing the mesh resolution using the implicit scheme has little to no effect on the overall accuracy of the solver, suggesting that it is dominated by temporal error. Increasing the resolution for the explicit solver does improve the accuracy at all times in the simulation, however, this incurs at least an order of magnitude increase in total computational cost. By extrapolating the convergence study using the explicit scheme, we can conclude that an infeasibly high resource utilization would be required to achieve the same level of accuracy with the second-order STAR-CCM+ configuration as the higher-order PyFR simulations.

5.2. DNS of the Taylor–Green vortex

5.2.1. Background

Simulation of the Taylor–Green vortex breakdown using the compressible Navier–Stokes equations has been undertaken for the comparison of high-order numerical schemes. It has been a test case for the first, second, and third high-order work-
Fig. 1. Contours of density at $t = 0, t = \tau_c, t = 5\tau_c, t = 50\tau_c$ for isentropic vortex advection with explicit PyFR $P_5$ and the second-order explicit and third-order implicit STAR-CCM+ configurations.

shops [1]. It is an appealing test case for comparing numerical methods due to its simple initial and boundary conditions, as well as the availability of spectral DNS results for comparison from van Rees et al. [23].

The initial flow field for the Taylor–Green vortex is specified as [1]

$$
\begin{align*}
  u &= +U_0 \sin(x/L) \cos(y/L) \cos(z/L), \\
  v &= -U_0 \cos(x/L) \sin(y/L) \cos(z/L), \\
  w &= 0, \\
  p &= P_0 + \frac{\rho_0 U_0^2}{16} (\cos(2x/L) + \cos(2y/L)) (\cos(2z/L) + 2), \\
  \rho &= \frac{p}{R T_0},
\end{align*}
$$

where $T_0$ and $U_0$ are constants specified such that the flow Mach number based on $U_0$ is $Ma = 0.1$, effectively incompressible. The domain is a periodic cube with the dimensions $-\pi L \leq x, y, z \leq +\pi L$. For the current study we consider a Reynolds number $Re = 1600$ based on the length scale $L$ and velocity scale $U_0$. The test case is run to a final non-dimensional time of $t = 20\tau_c$ where $\tau_c = L/U_0$.

We are interested in the temporal evolution of kinetic energy integrated over the domain

$$
E_k = \frac{1}{\rho_0 \Omega} \int_{\Omega} \rho \frac{\mathbf{v} \cdot \mathbf{v}}{2} d\Omega,
$$

and the dissipation rate of this energy defined as $\epsilon = -\frac{dE_k}{dt}$. We are also interested in the temporal evolution of enstrophy

$$
\epsilon = \frac{1}{\rho_0 \Omega} \int_{\Omega} \rho \frac{\mathbf{\omega} \cdot \mathbf{\omega}}{2} d\Omega.
$$
where $\omega$ is the vorticity. For incompressible flows the dissipation rate can be related to the enstrophy by $\epsilon = 2 \frac{\mu}{\rho_0} \varepsilon$ [1,23].

We can also define three different $L_\infty$ error norms. First the error in the observed dissipation rate

$$\|\tilde{\varepsilon}_1\|_\infty = \frac{\left\| \frac{dE_k}{dt} - \left( \frac{dE_k}{dt} \right)^* \right\|_1}{\max\left( \frac{dE_k}{dt} \right)} _{t, \infty},$$

(6)
Table 5
Configuration and results for Taylor–Green vortex simulations.

| Degree         | Elements | DOF | $|\epsilon_1|_\infty$ | $|\epsilon_2|_\infty$ | $|\epsilon_3|_\infty$ |
|---------------|---------|-----|----------------|----------------|----------------|
| STAR-CCM+ 2nd order | 256$^3$ | 256$^3$ | 1.97E–01 | 6.41E–01 | 5.85E–01 |
| STAR-CCM+ 3rd order | 256$^3$ | 256$^3$ | 4.27E–02 | 2.35E–01 | 1.94E–01 |
| PyFR $P_1$    | 128$^3$ | 256$^3$ | 1.43E–01 | 4.38E–01 | 1.35E–01 |
| PyFR $P_2$    | 86$^3$  | 258$^3$ | 4.17E–02 | 1.36E–01 | 1.06E–01 |
| PyFR $P_3$    | 64$^3$  | 256$^3$ | 3.00E–02 | 3.80E–02 | 3.49E–02 |
| PyFR $P_4$    | 52$^3$  | 260$^3$ | 1.94E–02 | 3.42E–02 | 1.61E–02 |
| PyFR $P_5$    | 43$^3$  | 258$^3$ | 1.99E–02 | 2.96E–02 | 1.90E–02 |
| PyFR $P_6$    | 37$^3$  | 259$^3$ | 1.34E–02 | 1.93E–02 | 8.45E–03 |
| PyFR $P_7$    | 32$^3$  | 256$^3$ | 1.68E–02 | 1.98E–02 | 6.18E–03 |
| PyFR $P_8$    | 29$^3$  | 261$^3$ | 1.60E–02 | 1.68E–02 | 5.38E–03 |

where $(\frac{dE}{dt})^\star$ is the reference spectral DNS dissipation rate \[23\]. We also consider the error in the dissipation rate predicted from enstrophy

$$\left\|\epsilon_2\right\|_\infty = \frac{\left\|2\nu \epsilon - 2\nu \epsilon^\star\right\|_{t,\infty}}{\max \left(\frac{dE}{dt}\right)^\star},$$

(7)

and the difference between the measured dissipation and that predicted from enstrophy during a particular simulation

$$\left\|\epsilon_3\right\|_\infty = \frac{\left\|\frac{dE}{dt} - 2\nu \epsilon\right\|_{t,\infty}}{\max \left(\frac{dE}{dt}\right)^\star},$$

(8)

where

$$\left\|a\right\|_{t,\infty} = \max_{\epsilon(0,1)} |a|.$$  

These definitions are consistent with the error calculations performed in the high-order workshops \[1\] and allow us to assess relative errors in the resolved dissipation mechanisms and actual dissipation.

For PyFR we use $P_1$ to $P_8$ schemes with structured hexahedral elements. Each mesh is generated to provide $\sim256^3$ degrees of freedom, as shown in Table 5, based on the number of degrees of freedom per element. The interface fluxes are LDG \[24\] and Rusanov type \[15\]. Gauss–Legendre points are used for the solution point locations within the elements and as flux point locations on the faces of the elements. A five-stage fourth-order adaptive Runge–Kutta scheme \[20–22\] is used with maximum and relative error tolerances of $10^{-6}$. The simulations are run on three NVIDIA K20c GPUs with the exception of the $P_3$ simulation, which was run on six GPUs due to the available memory per card. We perform two sets of simulations, the first with the release version of PyFR 0.3.0 and the second with the experimental version of PyFR 0.3.0 including GiMmiK.

For STAR-CCM+ we generate a structured mesh of $256^3$ hexahedral elements via the directed meshing algorithm. This gives a total of $256^3$ degrees of freedom as shown in Table 5, consistent with the number required for DNS \[23\]. For the second-order configuration we use the explicit time-stepping scheme provided with STAR-CCM+ with a constant CFL number of unity. We use a second-order spatial discretization with coupled energy and flow equations. We use an ideal gas formulation and laminar viscosity, since we expect to resolve all length and time scales in the flow. Periodic boundary conditions are used on all faces and all other settings are left at their default values. The third-order configuration is similar to the second-order configuration, however, we use the third-order MUSCL scheme for spatial discretization and second-order implicit time-stepping with $\Delta t = 0.01t_c$. The second-order configuration is run using all 12 cores of the Intel Xeon E5-2697 v2 CPU and the built in domain partitioning provided with STAR-CCM+. Due to increased memory requirements, the third-order configuration of STAR-CCM+ is run on five nodes of an Infiniband interconnected cluster of Intel Xeon X5650 CPUs.

### 5.2.2. Results

Isosurfaces of Q-criterion are shown in Fig. 5 to Fig. 8 at various instants from simulations using the PyFR $P_8$ scheme and the second-order and third-order STAR-CCM+ configurations. At the beginning of each simulation up to $t = 5t_c$, the flow is dominated by large scale vortical structures, with length scales proportional to the wavelength of the initial sinusoidal velocity field. In Fig. 6 at $t = 10t_c$ we see that the flow has undergone turbulent transition and contains a large number of small scale vortical structures. Significant differences are apparent between PyFR and the results from the second-order STAR-CCM+ configuration at this time. The PyFR simulation has a much broader range of turbulent scales than the STAR-CCM+ simulation. Also, nearly all of the smallest scale structures have been dissipated by the second-order STAR-CCM+...
Fig. 5. Isosurfaces of Q-criterion for the Taylor–Green vortex at $t = 5t_c$ PyFR $P_8$ (left), STAR-CCM+ second-order (middle), and STAR-CCM+ third-order (right).

Fig. 6. Isosurfaces of Q-criterion for the Taylor–Green vortex at $t = 10t_c$ PyFR $P_8$ (left), STAR-CCM+ second-order (middle), and STAR-CCM+ third-order (right).

Fig. 7. Isosurfaces of Q-criterion for the Taylor–Green vortex at $t = 15t_c$ PyFR $P_8$ (left), STAR-CCM+ second-order (middle), and STAR-CCM+ third-order (right).

Fig. 8. Isosurfaces of Q-criterion for the Taylor–Green vortex at $t = 20t_c$ PyFR $P_8$ (left), STAR-CCM+ second-order configuration (middle), and STAR-CCM+ third-order configuration (right).
configuration. In Fig. 7 at $t = 15t_c$, we see that the PyFR simulation has an increasing number of very small turbulent structures, while the second-order STAR-CCM+ configuration only has a few intermediate scale structures. Finally, by $t = 20t_c$, the turbulent structures predicted by the second-order STAR-CCM+ configuration have nearly completely dissipated, while PyFR has preserved them even until the end of the simulation. However, we see that increasing the order of accuracy of STAR-CCM+ with the third-order configuration significantly reduces the amount of numerical dissipation. These third-order results are qualitatively consistent with the high-order PyFR results, although some over-dissipation of small scale structures is still apparent at $t = 15t_c$ and $20t_c$.

Plots of the temporal evolution of the kinetic energy dissipation rate are shown in Fig. 9 for both STAR-CCM+ simulations and in Fig. 10 for the PyFR $P_1$ to $P_3$ simulations. The second-order STAR-CCM+ configuration is overly dissipative, over-predicting the kinetic energy dissipation rate up to $t_c \approx 8$ when compared to the spectral DNS results. After the peak dissipation rate the second-order STAR-CCM+ configuration then under-predicts the kinetic energy dissipation rate up until the end of the simulation. This is consistent with our qualitative observations of the type and size of turbulent structures in the domain. The second-order STAR-CCM+ configuration quickly dissipates energy from the domain and, as a consequence, little energy is left to be dissipated during the later stages of decay. By increasing the order of accuracy with the third-order configuration of STAR-CCM+ we observe a significant improvement in the predicted dissipation rate. However, there are still some inaccuracies, particularly around the time of peak dissipation. For PyFR, it is clear that the kinetic energy dissipation rate rapidly approaches the spectral DNS results with increasing order of accuracy from $P_1$ through $P_3$. By $P_3$ there is little difference between the current results and those of the reference spectral simulation, and it is significantly more accurate than either of the STAR-CCM+ simulations.

Plots of the temporal evolution of enstrophy are shown in Fig. 9 for the STAR-CCM+ simulations and in Fig. 11 for the PyFR simulations. The second-order STAR-CCM+ configuration under-predicts enstrophy throughout the simulation. Since enstrophy gives a measure of dissipation due to physical flow structures, we can conclude that a significant portion of the dissipation associated with the second-order STAR-CCM+ configuration is numerical. We see a significant improvement in the prediction of the temporal evolution of enstrophy with the third-order configuration of STAR-CCM+. However, there are still significant differences when compared to the reference spectral DNS data. We also observe that the PyFR simulations rapidly converge to the spectral DNS results with increasing order of accuracy. By $P_3$ the results are nearly indistinguishable from the reference solution. This demonstrates that the higher-order PyFR simulations can accurately predict the turbulent structures present during the simulation, and that the majority of the observed kinetic energy dissipation is physical, rather than numerical, in nature.

To quantify the relative accuracy and cost of the STAR-CCM+ and various PyFR simulations we can compare the three proposed error norms $\epsilon_1$, $\epsilon_2$, and $\epsilon_3$ against total resource utilization required for each simulation. The error in the observed dissipation rate is shown in Fig. 12 for all of the simulations plotted against the resource utilization measured in $\mathcal{E} \times$ seconds. Our first observation is that all of the PyFR simulations, from $P_1$ through $P_3$, are cheaper than simulations using the second-order STAR-CCM+ configuration. In fact, the $P_1$ to $P_3$ simulations are nearly an order of magnitude cheaper than the second-order STAR-CCM+ configuration. The third-order STAR-CCM+ configuration also costs significantly less than the second-order configuration, since it uses an implicit time-stepping approach. Also, we find that GiMMiK can reduce the cost of the PyFR simulations by between 20% and 45%, depending on the order of accuracy. Interestingly, the computational cost of the $P_1$ to $P_3$ schemes are comparable, demonstrating that PyFR can produce fourth-order accurate results for the same cost as a second-order scheme. Secondly, we observe that all of the PyFR simulations are more accurate than the second-order STAR-CCM+ simulations for this, and all other metrics including the temporal evolution of enstrophy in Fig. 13 and the difference between the observed dissipation rate and that predicted from enstrophy as shown in Fig. 14. When compared to the third-order STAR-CCM+ configuration, PyFR results with similar error levels are less expensive. Or, conversely, PyFR simulations of the same computational cost are up to an order of magnitude more accurate.
Fig. 10. Dissipation rate from DNS of the Taylor–Green vortex using PyFR.
Fig. 11. Temporal evolution of enstrophy from DNS of the Taylor–Green vortex using PyFR.
5.3. Turbulent flow over a circular cylinder

5.3.1. Background

Flow over a circular cylinder has been the focus of several previous experimental and numerical studies. Its characteristics are known to be highly dependent on the Reynolds number \( Re \), defined as

\[
Re = \frac{\rho U D}{\mu},
\]
where $U$ is the free-stream velocity, $\rho$ is the fluid density, $D$ is the cylinder diameter, and $\mu$ is the fluid viscosity. In the current study we consider flow over a circular cylinder at $Re = 3900$, and an effectively incompressible Mach number of 0.2. This case sits in the shear-layer transition regime identified by Williamson [25] and contains several complex flow features including separated shear layers, turbulent transition, and a fully turbulent wake. Recently Lehmkuhl et al. [26] and Witherden et al. [18] have shown that at this Reynolds number the flow field oscillates at a low frequency between a low energy mode, referred to as Mode-L, and a high energy mode, referred to as Mode-H. Previous studies [27–32] had only observed one, the other, or some intermediate values between the two in this Reynolds number regime, since their averaging periods were not of sufficient length to capture such a low frequency phenomenon [26]. The objective of the current study is to perform long-period averaging using both PyFR and STAR-CCM+ to compare with the DNS results of Lehmkuhl et al. [26].

We use a computational domain with dimensions [$-9D, 25D$]; [$-9D, 9D$]; and [0, $\pi D$] in the stream-wise, cross-wise, and span-wise directions, respectively. The cylinder is centered at (0, 0, 0). The span-wise extent was chosen based on the results of Norberg [30], who found no significant influence on statistical data when the span-wise dimension was doubled from $\pi D$ to $2\pi D$. Indeed, a span of $\pi D$ has been used in the majority of previous numerical studies [27–30], including the recent DNS study of Lehmkuhl et al. [26]. The stream-wise and cross-wise dimensions are also comparable to the experimental and numerical values used by Parnaudeau et al. [33] and those used for the DNS study of Lehmkuhl et al. [26]. The domain is periodic in the span-wise direction with a no-slip isothermal wall boundary condition applied at the surface of the cylinder. Both solvers are run using the compressible Navier–Stokes equations. For PyFR we use Riemann invariant boundary conditions at the far-field, while for STAR-CCM+ we use a free-stream condition for the inlet and pressure outlet for the upper, lower, and rear faces of the domain. We use meshes of similar topology for the PyFR and second-order STAR-CCM+ configurations, with a well resolved near-wall region composed of prismatic elements and a refined wake region composed of unstructured tetrahedral elements. However, we use fewer elements in the PyFR mesh since the FR scheme has multiple solution points per element. The PyFR mesh, shown in Fig. 15, has a total of 79,344 prismatic elements and 227,258 tetrahedral elements with a total of $\sim13.9$ million solution points when using $P_4$ elements. The second-order STAR-CCM+
mesh, shown in Fig. 16, has a ~6.0 million prismatic elements and ~7.2 million tetrahedral elements for a total of ~13.2 million solution points. For the third-order STAR-CCM+ simulation a structured mesh was used, as shown in Fig. 17, with a similar ~13.5 million solution points.

The PyFR and second-order STAR-CCM+ configurations were started by running an initial 100$t_c$ lead in time, where $t_c = D/U$, until the wake became fully developed. The third-order configuration data, provided by CD-adapco, was initialized from an inviscid solution, then run until it developed into a fully turbulent flow. All simulations were then run for an additional 1000$t_c$ to perform long-period time-averaging and statistical analysis for comparison with the results of Lehmkuhl et al. [26]. PyFR was run with a $P_4$ degree polynomial representation of the solution, a 5-stage 4th-order explicit Runge–Kutta time stepping scheme with $\Delta t \approx (2.4E-4)t_c$, Rusanov and LDG interface fluxes [15], and as an ILES simulation [12,13]. The PyFR simulation was run on an Infiniband interconnected cluster of 12 NVIDIA K20c GPUs, with three cards per node. The second-order STAR-CCM+ configuration was run with a second-order implicit time stepping scheme with $\Delta t \approx (5.0E-3)t_c$, the coupled implicit solver, second-order spatial accuracy, and the WALE subgrid scale model. The third-order STAR-CCM+ configuration was run using the third-order MUSCL scheme, second-order implicit time-stepping with $\Delta t \approx (8.5E-2)t_c$, and the WALE subgrid scale model. The computational cost for both STAR-CCM+ simulations was assessed on five nodes of an Infiniband interconnected cluster of Intel Xeon X5650 CPUs. The implicit solver was chosen for two reasons for both the second- and third-order STAR-CCM+ configurations. First, there is no available SGS model for the explicit scheme in STAR-CCM+, which is generally required for performing LES using finite-volume schemes. Secondly, due to mesh induced stiffness the explicit time-step size to achieve the recommended CFL $\approx 1.0$ was impractically small. This would have resulted in a total resource utilization of approximately 3.87E12 $E \times$ seconds for the second-order configuration, which is infeasible due to computational cost.

5.3.2. Results

The PyFR and STAR-CCM+ simulations had similar resource utilizations, PyFR with an explicit scheme at approximately 3.02E10 $E \times$ seconds, the second-order STAR-CCM+ configuration at 1.69E10 $E \times$ seconds, and the third-order configuration at 3.46E10 $E \times$ seconds. These are shown in Fig. 18, which also includes the estimated cost of an explicit STAR-CCM+ simulation. While the PyFR and STAR-CCM+ implicit time-stepping cases have similar computational costs, it is clear that explicit time-stepping using STAR-CCM+ is prohibitively expensive for this case.

Iso-surfaces of density colored by velocity magnitude are shown in Fig. 19 for PyFR and STAR-CCM+ at similar instants in their shedding cycles, respectively. The PyFR results exhibit more small and intermediate scale turbulent structures when compared to the second-order STAR-CCM+ configuration. This includes in the near wake directly behind the cylinder, as well as far downstream towards the end of the refined wake region of the mesh. These results corroborate earlier observations, in particular the dissipative behavior of the second-order STAR-CCM+ configuration for the previous Taylor–Green vortex test case. As the turbulent structures are advected downstream they are rapidly dissipated, and by $x/D \approx 5$ only the largest scale vortices remain in the flow. The third-order STAR-CCM+ configuration is able to capture more of the turbulent wake features behind the cylinder. However, the third-order configuration still appears to be qualitatively more dissipative than the PyFR simulation, particularly in the coarse grid beyond $x/D = 5$.

Time-averaged wake profiles in the stream-wise direction for both simulations are shown in Fig. 20 for the PyFR and STAR-CCM+ simulations alongside the DNS results of Lehmkuhl et al. [26]. A first observation is that the current PyFR results show excellent agreement with the reference long-period averaged DNS results of Lehmkuhl et al. [26]. The two simulations predict nearly identical separation bubble lengths and the peak recirculation bubble strength and corresponding location. The second-order STAR-CCM+ configuration predicts a much larger separation bubble that extends well past $x/D \approx 1.5$.  

![Fig. 18. Resource utilization for circular cylinder simulations.](image-url)
Both Lehmkuhl et al. [26] and Witherden et al. [15] demonstrated that this test case should oscillate between Mode-H and Mode-L type wakes, with the long-period average over both of these models being somewhere between the two. The second-order STAR-CCM+ configuration was only able to capture the characteristic Mode-L described by Lehmkuhl et al. [26], and failed to predict any transitions to Mode-H over the 1000t, simulation time. Therefore, the long-period average from the second-order STAR-CCM+ configuration is not consistent with previous observations for this test case. However, the third-order configuration shows significant improvement in the prediction of the mean wake profile. It has a profile consistent with the expected oscillation between the Mode-H and Mode-L type wakes, although it predicts a slightly stronger separation bubble.

Time-averaged wake profiles in the cross-stream direction are shown in Fig. 21, Fig. 22, and Fig. 23 at $x/D = 1.06$, $x/D = 1.54$, and $x/D = 2.02$, respectively. These plots show both the stream-wise and cross-stream velocity components.
normalized by the free-stream velocity, alongside the DNS results of Lehmkuhl et al. [26]. All plots show that there is excellent agreement between the current PyFR simulation and the reference DNS data [26]. This includes both the stream-wise and cross-stream velocity components at all measurement locations in the wake. The second-order STAR-CCM+ configuration produces a U-shaped velocity profile at $x/D = 1.06$, which is characteristic of only the Mode-L shedding process. It also predicts an increasingly large separation bubble size when considering the $x/D = 1.54$ and $x/D = 2.02$ measurement locations. Both the stream-wise and cross-stream velocity profiles do not show agreement between the reference DNS results [26] and results from the second-order STAR-CCM+ configuration. However, the third-order STAR-CCM+ configuration shows good agreement with the reference dataset. The only significant discrepancy is observed in the stream-wise velocity component at $x/D = 1.54$, where STAR-CCM+ predicts a stronger recirculation bubble.

Time-averaged velocity fluctuations for the stream-wise velocity component are shown in Fig. 20 for both the PyFR and STAR-CCM+ simulations alongside the DNS results of Lehmkuhl et al. [26]. PyFR predicts a double peak shape profile, with large velocity fluctuations in the wake directly behind the cylinder. Farther downstream the velocity fluctuations decrease and then plateau beyond $x/D \approx 3$, consistent with the reference data. The second-order STAR-CCM+ configuration under-predicts the stream-wise velocity fluctuations throughout the wake profile. It also predicts less velocity fluctuation than reported by Lehmkuhl et al. [26] for the Model-L shedding process. This is consistent with qualitative observations based on the turbulent structures observed in Fig. 19. The third-order STAR-CCM+ configuration shows a significant improvement. It successfully predicts the stream-wise velocity fluctuations at all measurement locations, although it generally predicts less fluctuation than both the PyFR and reference DNS data sets.

Time-averaged fluctuations of the stream-wise and cross-stream velocity components are also shown in Fig. 24, Fig. 25, and Fig. 26 at downstream locations $x/D = 1.06$, $x/D = 1.54$, and $x/D = 2.02$, respectively. The second-order STAR-CCM+ configuration consistently under-predicts both velocity fluctuations at all measurement locations throughout the profiles. However, the results from PyFR are consistent with the reference data [26], with only small differences apparent from the plots. This trend continues for the cross-stream velocity fluctuations. PyFR shows good agreement with the reference DNS dataset [26], while the second-order STAR-CCM+ configuration under-predicts the turbulent velocity fluctuations. In contrast, the third-order configuration shows much better agreement with the reference DNS data set. The only significant discrepancy is observed at $x/D = 2.02$, where the STAR-CCM+ simulation shows less fluctuation of the stream-wise velocity component along the wake centerline.

Power spectral density plots of the stream-wise and cross-stream velocity components taken at the point P1 [$x/D = 0.71$, $y/D = 0.66$] in the shear layer, as marked in Fig. 27, are shown in Fig. 28. The power spectral density is plotted against the Strouhal number $St = f D / U$ where $f$ is frequency. PyFR predicts a dominant vortex shedding Strouhal number $St_{sv} = 0.213$ for both the stream-wise and cross-stream velocity components. The second-order STAR-CCM+ configuration predicts a similar value, $St_{sv} = 0.212$, and so does the third-order configuration with $St_{sv} = 0.210$. All three simulations show a second large peak at a Strouhal number $St_{sp} = 0.419$ for PyFR, $St_{sp} = 0.428$ for the second-order STAR-CCM+ configuration, and $St_{sp} = 0.419$ for the third-order STAR-CCM+ configuration. PyFR also predicts a wide-banded Kelvin–Helmholtz instability around $St_{th} = 1.56$, while the second-order STAR-CCM+ configuration predicts it as a significantly slower phenomenon with $St_{th} = 1.02$, and the third-order STAR-CCM+ configuration predicts it around $St_{th} = 1.52$. A general observation is that the second-order STAR-CCM+ configuration predicts significantly smaller power peaks, and less power throughout the spectra particularly in the inertial subrange for the stream-wise and cross-stream velocity components. Moving further downstream into the mixing-layer at P2 [$x/D = 2.00$, $y/D = 0.59$] and the wake at P3 [$x/D = 4.00$, $y/D = 0.00$] the power deficit becomes even greater between PyFR and the second-order STAR-CCM+ configuration. (See Figs. 29 and 30.) Also, the shape of the inertial subrange predicted by the second-order configuration changes from the curved type profile predicted by the DNS of Lehmkuhl et al. [26] and current PyFR simulations to a linear power-law type profile. However, the third-order STAR-CCM+ configuration shows good agreement with the results from PyFR at all three of the measurement locations.
5.4. SD7003 aerofoil

5.4.1. Background

Finally, we investigate transitional and turbulent flow over an SD7003 aerofoil [34] using a $P_4$ degree scheme with PyFR and a second-order configuration with STAR-CCM+. A recommended third-order configuration was not available from CD-adapco for this case. Both simulations are run at a Reynolds number $Re = 60,000$, Mach number $Ma = 0.2$, and angle of attack $\alpha = 8^\circ$. The ratio of specific heats is $\gamma = 1.4$, the Prandtl number is $Pr = 0.72$, and constant viscosity is used due to the relatively low Mach number. This test case is commonly used to examine the suitability of numerical schemes for...
Fig. 24. Time-averaged velocity fluctuation profiles for turbulent flow over a circular cylinder at \( x/D = 1.06 \) from PyFR, STAR-CCM+, and Lehmkuhl et al. [26].

Fig. 25. Time-averaged velocity fluctuation profiles for turbulent flow over a circular cylinder at \( x/D = 1.54 \) from PyFR, STAR-CCM+, and Lehmkuhl et al. [26].

Fig. 26. Time-averaged velocity fluctuation profiles for turbulent flow over a circular cylinder at \( x/D = 2.02 \) from PyFR, STAR-CCM+, and Lehmkuhl et al. [26].

predicting separation, transition, and turbulent flow. It has been studied previously by, for example, Visbal and collaborators including Visbal et al. [35] and Garmann et al. [36] using finite-difference methods and Beck et al. [37] using a DG spectral element method (DGSEM). The characteristic features of the flow include laminar separation on the upper surface of the aerofoil, which then reattaches further downstream forming a laminar separation bubble. The flow transitions to turbulence part-way along this separation bubble, creating a turbulent wake behind the aerofoil.

For both simulations we use unstructured hexahedral meshes with similar topologies as shown in Fig. 31. The domain extends to 10c above and below the aerofoil, 20c downstream, and 0.2c in the span-wise direction, where c is the aerofoil chord length. A structured mesh is used in the boundary layer region, with a fully unstructured and refined wake region behind the aerofoil to capture the turbulent wake. The PyFR mesh uses quadratically curved elements at the boundaries.
to match the aerofoil geometry. The boundary layer resolution gives $y^+ \approx 0.4$ and $y'^+ \approx 0.45$ at the first solution point off the surface for PyFR and STAR-CCM+, respectively, where $y^+ = u^+_r y/v$, $u^+_r = \sqrt{f T/2 \nu}$, $U_\infty$ is the free-stream velocity magnitude, and $C_f \approx 8.5 \times 10^{-3}$ is the maximum skin friction coefficient in the turbulent region reported by Garmann et al. [36]. The PyFR mesh has a total of 138,024 hexahedral elements yielding 17,253,000 solution points and has 12 elements in the span-wise direction, while the STAR-CCM+ mesh has a total of 18,509,652 hexahedral elements with 60 elements in the span-wise direction. Both solvers are run using the compressible Navier–Stokes equations and adiabatic no-slip wall boundary conditions for the surface of the aerofoil. PyFR used Riemann invariant boundary conditions at the far-field, while STAR-CCM+ used a free-stream condition. PyFR was run using GiMMyK due to the performance improvements observed in the Taylor–Green vortex test case for hexahedral elements. For PyFR, an adaptive RK45 temporal scheme was used with relative and absolute error tolerances of $10^{-6}$ [20–22], LDG [24] and Rusanov type [15] interface fluxes, and Gauss points for both solution and flux points in each element. The PyFR simulation was run on an Infiniband interconnected cluster of 12 NVIDIA K20c GPUs, with three cards per node. The second-order STAR-CCM+ configuration was run with a second-order implicit time stepping scheme with $\Delta t \approx (5.0 \times 10^{-3})t_c$, the coupled implicit solver, second-order spatial accuracy, and the WALE subgrid scale model. The computational cost for the STAR-CCM+ simulation was assessed on five nodes of an Infiniband interconnected cluster of Intel Xeon X5650 CPUs. All simulations were run to $t = 20 t_c$, where $t_c = c/U_\infty$, to allow the flow to develop, separate, and transition. Statistics were then extracted over an additional 20$t_c$, including span-wise averaging where appropriate.

5.4.2. Results

The resource utilization of PyFR over the 20$t_c$ averaging period was 10.01E9 £ × seconds, while for STAR-CCM+ it was lower by a factor of $6.4 \times 1.56E9 £ × seconds$. Isosurfaces of q-criterion colored by velocity magnitude are shown in Fig. 32 for PyFR and STAR-CCM+ in the fully-developed regime after 20$t_c$. The PyFR simulation appears to resolve more intermediate and small scale turbulent structures when compared to the STAR-CCM+ simulation, even though the spatial resolution of both simulations is equivalent. This is consistent with previous results from the Taylor–Green vortex and circular cylinder test cases. Qualitatively, the transition point in the PyFR simulation also appears nearer the leading edge, while for STAR-CCM+ it appears further downstream.

Contours of time and span-averaged streamwise velocity are shown in Fig. 33 for PyFR and STAR-CCM+. Both simulations exhibit a laminar separation bubble near the leading edge on the suction side of the aerofoil, which is characteristic of this test case [36,37]. The relative separation ($x_{sep}$) and reattachment ($x_{rea}$) points are shown in Table 6 alongside reference values from previous studies [36,37]. The separation points for both simulations are in good agreement with previous studies. The reattachment point of the PyFR simulation is in agreement with previous studies, whereas the STAR-CCM+ result appears to be further downstream.

A plot of time and span-averaged pressure coefficient is shown in Fig. 34 for both PyFR and STAR-CCM+ alongside a collection of reference results [36,37]. The PyFR results appear to be in agreement with the reference data sets, including the location of the transition region. The STAR-CCM+ results show relatively less suction on the upper surface and also predict turbulent transition further downstream when compared to the reference data sets.

6. Conclusions

We have investigated the accuracy and computational cost of PyFR and STAR-CCM+ for a range of test cases including scale resolving simulations of turbulent flow. Results from isentropic vortex advection show that all of the PyFR simulations on GPUs were approximately an order of magnitude cheaper than both the second-order and third order STAR-CCM+ configurations on CPUs. In addition, the PyFR $P5$ simulation was approximately five orders of magnitude more accurate.
PyFR simulations of the Taylor–Green vortex test case on GPUs were consistently cheaper and more accurate than the second-order STAR-CCM+ configuration on CPUs. This was across all three accuracy metrics including the kinetic energy dissipation rate, temporal evolution of enstrophy, and the difference between the observed and expected dissipation rates based on enstrophy. Qualitatively, the high-order PyFR results were found to resolve more of the fine, small scale features expected for the Taylor–Green vortex test case at this Reynolds number. In contrast, the second-order STAR-CCM+ configuration was found to rapidly dissipate these turbulent structures of interest. The third-order STAR-CCM+ configuration was found to provide more accurate results than the second-order configuration, and with lower computational cost, by employ-
ing implicit time-stepping. However, PyFR was still over an order of magnitude more accurate for equivalent computational cost with higher-order schemes.

Turbulent flow over a circular cylinder at \( Re = 3900 \) was then considered. PyFR was run using a \( P_4 \) scheme on GPUs, and STAR-CCM+ was run using both second-order and third-order configurations on CPUs. Both STAR-CCM+ simulations were run using implicit time-stepping, due to the high computational cost of the available explicit scheme and its lack of a suitable SGS model. Qualitative results showed that the second-order STAR-CCM+ configuration rapidly dissipated turbulent structures in the wake behind the cylinder. The third-order STAR-CCM+ and the \( P_4 \) PyFR configurations were found to be less dissipative. Both PyFR and the third-order STAR-CCM+ configuration showed good agreement with the reference DNS dataset in terms of time-averaged velocity and fluctuations, while the second-order STAR-CCM+ configuration failed to predict low frequency wake oscillations and also under-predicted velocity fluctuations [26]. Velocity power spectra from PyFR and the third-order STAR-CCM+ configuration also showed good agreement, while the second-order STAR-CCM+ configuration was found to be overly dissipative.

Finally, we considered turbulent flow over an SD7003 aerofoil at \( Re = 60,000 \). PyFR was run using a \( P_4 \) scheme on GPUs and STAR-CCM+ was run using a second-order configuration on CPUs. PyFR resolved a wide range of turbulent length scales and showed good agreement with reference data sets in terms of lift and drag coefficients, separation and reattachment points, and the mean pressure coefficient. STAR-CCM+ predicted a longer separation bubble compared to the reference datasets, with both the transition and reattachment points further downstream. However, we note that the STAR-CCM+ simulation required approximately \( 6.4 \times \) less resource utilization by employing implicit time-stepping.

In summary, results from both PyFR and STAR-CCM+ demonstrate that third-order schemes can be more accurate than second-order schemes for a given cost. Moreover, advancing to higher-order schemes on GPUs with PyFR was found to offer even further accuracy vs. cost benefits relative to industry-standard tools. These results demonstrate the potential utility of high-order methods on GPUs for scale-resolving simulations of unsteady turbulent flows.
Fig. 32. Isosurfaces of $q$-criterion colored by instantaneous streamwise velocity magnitude for the SD7003 test case using $P_4$ PyFR (top) and second-order STAR-CCM+ (bottom).

Fig. 33. Time and span-averaged stream-wise streamwise velocity contours for the SD7003 test case using $P_4$ PyFR (top) and second-order STAR-CCM+ (bottom).

Acknowledgements

The authors would like to thank the Engineering and Physical Sciences Research Council for their support via a Doctoral Training Grant, an Early Career Fellowship (EP/K027379/1), and the Hyper Flux project (EP/M50676X/1), and NVIDIA for hardware donations. The authors would also like to thank CD-adapco, and in particular Alastair West and Doru Caraeni, for
providing the third-order STAR-CCM+ configurations, and data for the third-order STAR-CCM+ cylinder simulation. Finally, we note that since undertaking the simulations presented here, a newer version of STAR-CCM+ (v11) has become available.

Appendix A. Supplementary material

Data Statement: Data relating to the results in this manuscript can be downloaded as Electronic Supplementary Material under a CC-BY-NC-ND 4.0 license.

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.jcp.2016.12.049.

References


