



Towards Exascale Computation for Turbomachinery Flows

Yuhang Fu*
Weiqi Shen*
121fuyh@zju.edu.cn
weiqishen1994@gmail.com
Zhejiang University
Hangzhou, China

Jifa Zhang
Tingwei Ji
jfzhang@zju.edu.cn
zjttw@zju.edu.cn
Zhejiang University
Hangzhou, China

Hanyue Liu
Xu Liu
1424039642@qq.com
aj_qlss@163.com
National Supercomputing Center in
Wuxi
Wuxi, China

Yan Yan
Yan.Yan@xjtlu.edu.cn
Xi'an Jiaotong-Liverpool University
Suzhou, China

Shirui Luo
shirui@illinois.edu
University of Illinois
Urbana, United States

Jiahuan Cui†
Yao Zheng†
jiahuancui@intl.zju.edu.cn
yao.zheng@zju.edu.cn
Zhejiang University
Hangzhou, China

Fangfang Xie
fangfang_xie@zju.edu.cn
Zhejiang University
Hangzhou, China

Xiyang Liu
Xiaoyu Song
17367765682@163.com
songxiaoyu33@163.com
Taiyuan University of Technology
Taiyuan, China

Paul Tucker
pgt23@cam.ac.uk
University of Cambridge
Cambridge, United Kingdom

Seid Koric
koric@illinois.edu
University of Illinois
Urbana, United States

Guangwen Yang†
Zhao Liu
ygw@tsinghua.edu.cn
liuz18@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Xiaojing Lv
jing3704@126.com
National Supercomputing Center in
Wuxi
Wuxi, China

Guocheng Tao
guochengtao@intl.zju.edu.cn
Zhejiang University
Hangzhou, China

Steven Miller
saem@ufl.edu
University of Florida
Gainesville, United States

Weimin Zheng
zwm-dcs@tsinghua.edu.cn
Tsinghua University
Beijing, China

ABSTRACT

A state-of-the-art large eddy simulation code has been developed to solve compressible flows in turbomachinery. The code has been engineered with a high degree of scalability, enabling it to effectively leverage the many-core architecture of the new Sunway system. A consistent performance of 115.8 DP-PFLOPs has been achieved on a high-pressure turbine cascade consisting of over 1.69 billion mesh elements and 865 billion Degree of Freedoms (DOFs). By leveraging a high-order unstructured solver and its portability to large heterogeneous parallel systems, we have progressed towards solving the

grand challenge problem outlined by NASA [1], which involves a time-dependent simulation of a complete engine, incorporating all the aerodynamic and heat transfer components.

KEYWORDS

Exascale computing, turbomachinery, heterogeneous many-core system, Sunway supercomputer, large eddy simulation, flux reconstruction method

*Both authors contributed equally to this research.

† Corresponding authors



This work is licensed under a Creative Commons Attribution International 4.0 License.

SC '23, November 12–17, 2023, Denver, CO, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0109-2/23/11.
<https://doi.org/10.1145/3581784.3627040>

ACM Reference Format:

Yuhang Fu, Weiqi Shen, Jiahuan Cui, Yao Zheng, Guangwen Yang, Zhao Liu, Jifa Zhang, Tingwei Ji, Fangfang Xie, Xiaojing Lv, Hanyue Liu, Xu Liu, Xiyang Liu, Xiaoyu Song, Guocheng Tao, Yan Yan, Paul Tucker, Steven Miller, Shirui Luo, Seid Koric, and Weimin Zheng. 2023. Towards Exascale Computation for Turbomachinery Flows. In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '23)*, November 12–17, 2023, Denver, CO, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3581784.3627040>

1 JUSTIFICATION FOR GORDON BELL PRIZE

- High complexity: Aero-thermodynamics of high pressure turbine blades with geometrical features across multiple orders of magnitude are modelled using an unstructured Computational Fluid Dynamics (CFD) code.
- High precision: The unstructured CFD code solves Navier-Stokes equations with numerical order of accuracy upto 8^{th} order.
- High fidelity: The numerical results are validated with detailed experimental results.
- High efficiency: A sustained peak performance of 115.8 DP-PFLOPs was achieved with strong scaling parallel efficiency measured at 89%.

2 PERFORMANCE ATTRIBUTES

Table 1: Performance attributes on Sunway supercomputer.

Attributes	Values
Category of achievement	Scalability and peak performance
Performance	115.8 DP-PFLOPs
Maximum problem size	865 billion DOFs
Type of method used	Explicit flux reconstruction
Results reported on basis of	Application with and without I/O
Precision reported	Double
System scale	350K MPEs with 22.4M CPEs
Measurement mechanism	FLOP count and timer

3 OVERVIEW OF THE PROBLEM

Turbomachinery remains a significant consumer of fossil fuels worldwide. At current fossil fuel prices, any small gas turbine performance improvement can result in a multi-billion-dollar economic impact and a major CO_2 emission reduction to mitigate the effects of global climate change. The bladed components of gas turbines, commonly referred as Turbomachines, are typically exposed to extremely high temperature and pressure, especially for the turbines located downstream of the combustor (i.e. high pressure turbine as illustrated in Fig. 1). Given that the thermodynamic efficiency of a gas turbine is proportional to both the turbine entry temperature and compressor pressure ratio, there is a strong motivation to increase both parameters. However, this results in an elevated operating temperature (often above the material melting point) for high-pressure turbines. To mitigate the problem associated with high temperatures and pressures, intricate flow paths within blades are designed to channel cooling gas directly from the compressor outlet (as depicted in Fig. 1(e)). Reliable prediction of the blade temperature field is critical, as an error of just 2% in predicting the blade metal temperature can halve its operating lifespan [2]. Currently, a conservative design approach is typically employed to maintain the material and structural integrity from thermal damage. A higher fidelity prediction method could yield more accurate results. For instance, a 1% reduction in cooling flow via a better thermal design could increase the overall efficiency of a gas turbine by 0.4% [2]. This could result in significant fuel savings worth billions of dollars, considering the number of gas turbines currently in operation [3].

The high pressure turbine blade is widely considered to be one of the most challenging components to simulate and design in gas turbines. Due to the highly unstable flow and temperature at the combustor exit, particularly under off-design conditions, heat transfer coefficient prediction errors can exceed 100% [4]. The richness and complexity of the flow behavior within high pressure turbines contribute to the difficulty of prediction. The internal cooling passages are long, narrow channels with widths on the millimeter scale and lengths on the centimeter scale. This wide range of scales, coupled with the complex geometry, presents a significant challenge in meshing the computational domain [5, 6].

Advanced CFD methods, such as Large Eddy Simulation (LES) or Direct Numerical Simulation (DNS), have been proposed to predict the flow and temperature of high pressure turbines with great accuracy [7]. Although these methods offer accurate and reliable simulation, they are computationally demanding. The cost of applying such high-fidelity methods is significantly higher compared to Reynolds-Averaged based methods. However, with the continuous advancement of high-performance computing power over the last decades, exascale computing is now within reach. By leveraging powerful computers and advanced numerical methods, more complex physics can be accurately captured and ultimately revealed to the designer.

To take full advantage of the computing power of exascale clusters, a novel code called ZJU-FR (ZFR) has been developed from scratch to ensure portability across heterogeneous systems. Developed initially at the University of Florida [8] and subsequently at Zhejiang University in China, ZFR is equipped with advanced turbomachinery flow capabilities, including rotating periodic boundary, sliding plane method, etc. The present study investigates the flow surrounding representative high pressure turbine blades from von Kármán institute [9] and NASA Lewis Research Center [10].

4 CURRENT STATE OF THE ART

4.1 Current State of Large Eddy Simulation

Reynolds-Averaged Navier-Stokes (RANS) simulations, which are widely used in industrial and academic applications, possess several drawbacks, including the loss of temporal information and the lack of universality of the models. Despite the inherent complexity and unsteadiness of flows in gas turbines, turbomachinery designers still rely heavily on RANS simulation for day-to-day analysis. The circumferential averaging performed by RANS simulation between each blade row results in the smoothing out many important flow behaviors, such as wake-blade interaction. It has been demonstrated that these flow features are important in determining compressor and turbine efficiency, particularly for multi-stage components with many blade rows [3, 11].

As supercomputers are more accessible to research communities now, especially with the development of computation-focused devices like Graphics Processing Unit (GPU) or many-core devices, previously unaffordable DNS and LES are becoming more feasible for industrial applications. Researchers have used AVBP, a compressible Navier-Stokes solver developed by CERFACS, to conduct a full engine LES and investigate the complex flow features within gas turbines [12, 13]. The finite element method (FEM) based AVBP

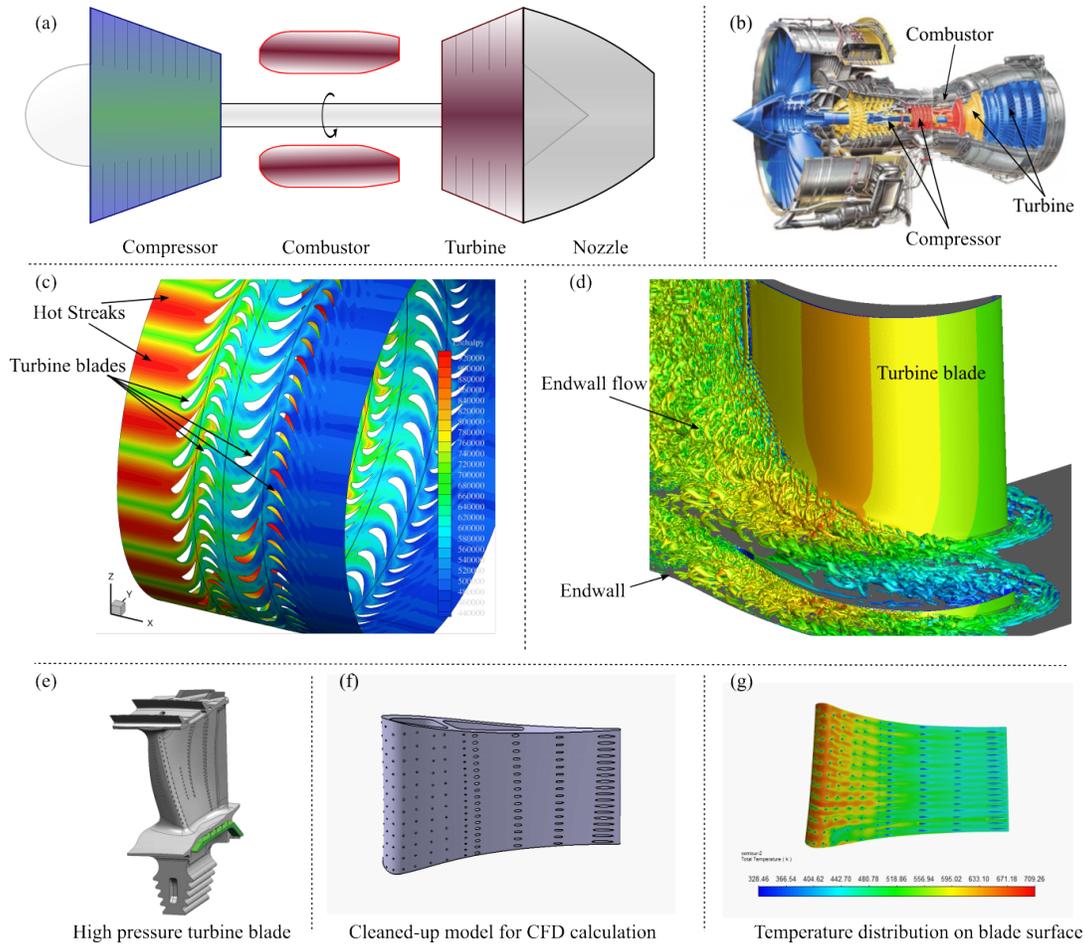


Figure 1: Main components of gas turbine and turbomachinery flow features: (a) schematic of core components of a gas turbine; (b) illustration of a turbo-fan engine; (c) enthalpy field of hot streaks incoming from combustor into the high pressure turbine; (d) turbulent flow structures near the turbine endwall; (e) CAD model of a real turbine blade; (f) clean-up CAD model of a turbine blade with cooling holes for CFD calculation; (g) temperature field on high pressure turbine blade surface with cooling holes.

solver utilizes a Taylor-Galerkin discretization method to overcome the numerical instability of traditional FEM for convection equations. As part of the European Programming Environment for Heterogeneous Supercomputers (EPEEC) project, more advanced multi-layer parallel programming model, such as GASPI+MPSs and ArgoDSM+MPSs [14], have been integrated with AVBP to take advantage of the upcoming heterogeneous supercomputers.

4.2 Current State of Numerical Methods and Related Solvers

Computational mesh must be fine enough for LES to resolve the majority of turbulent scales and the time step size must be small enough to capture time-accurate flow dynamics, resulting in a computational cost orders of magnitude higher compared with RANS simulation [15]. To address this challenge, it is essential to take advantage of the large modern massive parallel supercomputers.

Almost all flagship systems feature a high FLOP/s-per byte ratio, meaning that memory speed is increasingly becoming the bottleneck of the computation. To keep up with the fast advancement of computing facilities, new series of finite-element-based high-order numerical schemes with higher data locality have been proposed, such as Discontinuous Galerkin (DG) method [16], and the Flux Reconstruction (FR) method [17]. These methods are hybrid structured/unstructured, with structurally aligned solution/flux points within each element while data associated with elements is unstructurally stored. This hybrid method offers flexibility when meshing complex domains, and fully utilizes the computing power of modern architectures due to its compactness and structuredness.

A few finite-element-based high-order CFD solvers have been developed over the past few decades. NEK5000 [18] and Alya [19] served as a pioneering work that inspired and initiated the effort to push finite element methods for more practical industrial flows. Later, solvers based on DG method, such as SU2 [20] and Flexi [21],

were developed to enable the solution of high-speed compressible flows on unstructured grids. A more unified framework called FR method or Correction Procedure via Reconstruction (CPR), was introduced later by Huynh in 2007 [17]. The FR method can recover DG and spectral Difference (SD) methods [22] and is simpler and more computationally efficient than its predecessors [23].

PyFR is an open-source framework based on the FR approach to solve advection diffusion type problems. It was developed at Imperial College London [24], and is primarily written in Python. Mako template language is embedded in the code to generate numerical scheme kernel code for different hardware platforms at runtime, enabling developers to achieve high performance while still maintain a clean code base. The performance of FR methods is heavily limited by memory bandwidth on modern hardware. To alleviate this issue, a data structure facilitating cache blocking has been implemented, leading to a speedup of up to 2.8X for the Tylor Green vortex test case [25]. A petascale DNS simulation of compressible flow over a low pressure turbine (LPT) at Reynolds number of 200,000 was conducted. Meshes with up to 180 million hexahedron elements with fifth order accuracy were used, resulting in over 11 billion degree of freedoms per equation. The result was selected as one of the finalists for the 2016 Gordon Bell prize.

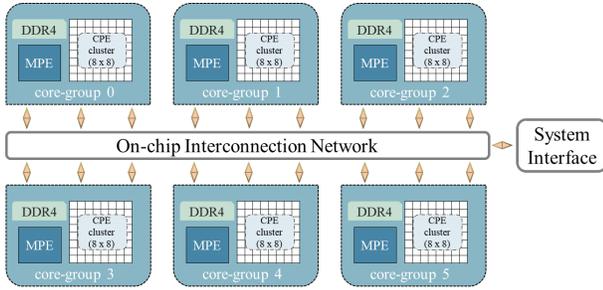


Figure 2: The architecture of SW26010Pro many-core processor on the new Sunway supercomputer.

4.3 Current State of New Sunway Supercomputer

4.3.1 System Architecture. The latest generation of the Sunway supercomputer has far more computing power than its predecessor (Sunway TaihuLight) thanks to a larger number of chips and the improved high-performance heterogeneous many-core processor. The system consists of over 100,000 SW26010 pro chips. Compared to the previous generation chip used in the Sunway TaihuLight, the upgraded processor includes six core-groups (CGs), each containing one MPE and one 8x8 CPE cluster with 16 GB DDR4 sharing memory (see Fig. 2). Each CPE has a local data memory (LDM) extended to 256 KB and supports wider 512 bit SIMD operation. More details of this new Sunway system can be found in [26, 27]

Sunway Accelerate Computing Architecture (SACA) is an essential component of the latest Sunway supercomputer. The SACA contains a parallel accelerated computing platform and programming model tailored to Sunway many-core architecture. SACA provides three levels of parallel acceleration, which are task (MPI),

thread (Athread) and data (SIMD), that maximize system performance. By utilizing Direct Memory Access (DMA) and register communication, programmers can explicitly and effectively control data exchange on the LDM, which is crucial to minimize the communication cost of CPEs.

4.3.2 Parallelization Challenge on Sunway for ZFR. The ZFR solver faced several challenges to be efficiently ported on the Sunway system. To fully utilise the computing processor on the heterogeneous system, a hybrid "MPI + Athread" strategy incorporating multi-level parallelism needs to be adopted. Additionally, as an unstructured solver, it is impossible to perform regular partition like stencil computations on CPE cores. Though FR method can significantly reduce data communication with high orders, bandwidth can still be the bottleneck, as Sunway has a relative lower byte-to-flop ratio compared to other heterogeneous systems. Thus, attention needs to be paid to data locality. As the mesh scales to billions of elements and the solver runs with tens of millions of cores, an efficient parallel mesh pre-processing is also required. With these challenges in mind, many innovative methods have been implemented on the pre-processor and the solver.

5 INNOVATIONS REALISED

5.1 ZFR Framework

This section provides an overview of the structure of the inhouse code – ZFR. A schematic of the solver is illustrated in Fig. 3. The code contains two main components: a low-level supporting framework and a high-level FR solver. The low-level framework incorporates algorithms and data structures that are independent of numerical schemes, offering a unified interface for high-performance physics simulation. The parallel partitioner is used to decompose the computational domain into non-overlapping sub-domains for parallel computation, with either ParMETIS [28] or PT-Scotch [29] working as the backend.

The FR solver comprises several components that define the geometrical data structures, numerical algorithms, and equation systems. Only the equation system encapsulates the physical problem while other components are problem independent. The current implementation supports the compressible flow equation system, with physical models including boundary conditions, Riemann solvers, body forcing, LES sub-grid scale models, and shock capturing methods.

5.1.1 FR Algorithm. The governing equations are rewritten into the form of conservation law and transformed to reference domain within each element as

$$\frac{\partial \hat{Q}}{\partial t} + \nabla_{\xi} \cdot (\hat{F} + \hat{G}) = \hat{S}, \quad (1)$$

where $\hat{Q}(\xi, t) = |J|Q(x, t)$ is the transformed solution variable, $\hat{F} = |J|J^{-1} \cdot F$ and $\hat{G} = |J|J^{-1} \cdot G$ are the transformed inviscid and viscous flux, $\nabla_{\xi} \cdot \square$ is divergence operator in the reference domain, and $\hat{S} = |J|S$ is the transformed source term. Here, $J_{ij} = \frac{\partial x_i}{\partial \xi_j}$ is the transformation Jacobian, where x represents coordinates in physics domain, and ξ represents the coordinates in reference domain. The transformed solution and flux functions are approximated by piecewise discontinuous polynomials defined in the reference

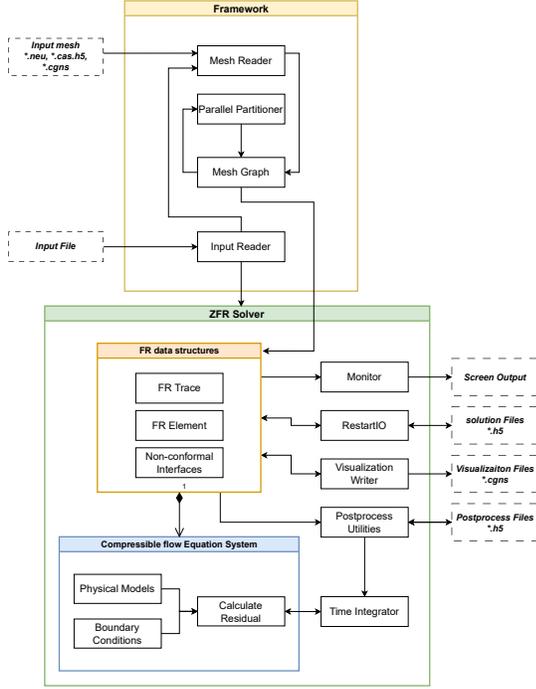


Figure 3: Overview of the code structure.

element as

$$\hat{Q}^\delta(\xi) = \sum_{i=1}^{N_s} \hat{Q}^\delta(\xi_i) l_i(\xi), \quad (2a)$$

$$\hat{F}^\delta(\xi) = \sum_{i=1}^{N_s} \hat{F}^\delta(\xi_i) l_i(\xi), \quad (2b)$$

$$\hat{G}^\delta(\xi) = \sum_{i=1}^{N_s} \hat{G}^\delta(\xi_i) l_i(\xi), \quad (2c)$$

where N_s is number of solution points inside each element. On the interfaces, the normal common numerical flux is calculated using an approximate Riemann solver with solutions interpolated to both sides of the interface. Then, a correction function g is used to correct the divergence of flux with the normal common numerical flux on the interface to recover a global solution. In this work, g is chosen so that the resulting scheme is equivalent to a nodal collocated DG method [17]. The viscous flux is handled using the local discontinuous Galerkin (LDG) method, which is described in detail by Huynh [30]. The semi-discretized form of the scheme

takes the form

$$\frac{dQ}{dt} = -|J|^{-1} \left[\nabla_\xi \cdot \hat{F}^\delta + \sum_{i=1}^{N_I} \sum_{j=1}^{N_{F_i}} \nabla_\xi \cdot g_{ij}(\xi) \left(\hat{F}_{ij}^{\delta I \perp} - \hat{F}_{ij}^{\delta \perp} \right) \right] + S, \quad (3)$$

where N_I and N_{F_i} are the number of interfaces on each element and number of flux points on the interface, respectively. The transformed numerical flux, and the normal transformed discontinuous flux on the interface are represented by $\hat{F}^{\delta I \perp}$ and $\hat{F}^{\delta \perp}$, respectively. Finally, the solution is advanced in time using an ODE solver such as strong-stability-preserving Runge-Kutta method [31].

5.1.2 Pre-processor Parallelization. The preprocess module of the code is carefully designed to efficiently read and process large scale computational mesh in parallel. Mesh entities including cells and vertices to be read by each rank are determined using a cumulative storage format, so that each rank reads a non-overlapping chunk of data. For CGNS file format, this operation is performed using the parallel I/O support of HDF5 built on MPI-IO, either collectively or independently.

After reading the cell and vertex information, the mesh partitioner is called to construct a dual-graph, which includes cell-cell connectivity information. The face list is constructed on each rank using local cell connectivity, and the vertex list of each face is sorted in ascending order to create a unique numbering. The face list is then sorted in ascending order by vertex IDs to make sure that coupled faces are next to each other. Then, local internal faces are coupled and repeating entries are eliminated, so that the remaining uncoupled faces are either MPI internal faces or boundary faces.

To read the boundary faces, algorithm 1 presents the pseudo code that uses the cumulative storage format to read each boundary. According to the CGNS specification, each boundary condition corresponds to one or more face element sections.

MPI internal faces in each rank need to find their remote counterparts. The face matching information is used later in the solver to exchange data within each time step. The algorithm to find MPI matching faces is presented in Algorithm 2. Uncoupled MPI internal faces are redistributed linearly throughout the ranks by their leading vertex ID. This method makes sure that coupled faces will be sent to the same rank. Similar to the local internal face matching algorithm, the received faces are sorted and their counterparts are found.

5.2 Three-level Parallelization Design

To optimize computational performance on the Sunway system, we have implemented a three-level parallelization strategy using the SACA programming model. This approach, as illustrated in Fig. 4, involves decomposing the unstructured grid into different partitions for each MPI process, distributing computing tasks evenly across elements and interfaces for each CPE thread, and conducting vectorization on the data dimension of sequential points per element or interface. The success of the last two levels of parallelism relies on maximizing utilization of the limited LDM space on CPEs. To

Algorithm 1 Method to read boundary faces with CGNS.

```

1: procedure READBOUNDARY(boundaryList,faceList)
2:   boundaryList  $\leftarrow$  read boundary condition info
3:   glob_bface  $\leftarrow$  get global number of boundary faces
4:   read_range  $\leftarrow$  DistributeEntity(glob_bface)
5:   for ibnd  $\leftarrow$  1 to N do
6:     boco_sect  $\leftarrow$  get section list
7:     for isec  $\in$  boco_sect do
8:       section_range  $\leftarrow$  get section range
9:       if read_range  $\cap$  section_range  $\neq \emptyset$  then
10:        vlist  $\leftarrow$  read connectivity
11:        update bface_list with vlist and ibnd
12:      end if
13:    end for
14:  end for
15:  distribute uncoupled faces with leading vertex ID
16:  sbuffer  $\leftarrow$  uncoupled faces
17:  np  $\leftarrow$  number of processes
18:  MPI_Issend(sbuffer,p) for p  $\leftarrow$  1, np
19:  repeat
20:    if MPI_Iprobe then
21:      MPI_Recv(rbuffer)
22:      update face_recv with rbuffer
23:    end if
24:    if MPI_Issend finished then
25:      call MPI_Ibarrier
26:    end if
27:  until MPI_Ibarrier completed
28:  sort(face_recv)
29:  remove repeating entries in face_recv
30:  distribute boundary faces with leading vertex ID
31:  sbuffer  $\leftarrow$  boundary faces
32:  MPI_Issend(sbuffer,p) for p  $\leftarrow$  1, np
33:  repeat
34:    if MPI_Iprobe then
35:      MPI_Recv(rbuffer)
36:      update bface_recv with rbuffer
37:    end if
38:    if MPI_Issend finished then
39:      call MPI_Ibarrier
40:    end if
41:  until MPI_Ibarrier completed
42:  sort(bface_recv)
43:  match bface_recv with face_recv
44:  sbuffer  $\leftarrow$  matched faces
45:  MPI_Issend(sbuffer,p) for p  $\leftarrow$  1, np
46:  repeat
47:    if MPI_Iprobe then
48:      MPI_Recv(rbuffer)
49:      update faceList with rbuffer
50:      update boundaryList with rbuffer
51:    end if
52:    if MPI_Issend finished then
53:      call MPI_Ibarrier
54:    end if
55:  until MPI_Ibarrier completed
56: end procedure

```

Algorithm 2 Method to match MPI internal faces.

```

1: procedure MATCHMPIFACES(MPIFaceRank,faceList)
2:   distribute uncoupled faces with leading vertex ID
3:   sbuffer  $\leftarrow$  uncoupled faces
4:   np  $\leftarrow$  number of processes
5:   MPI_Issend(sbuffer,p) for p  $\leftarrow$  1, np
6:   repeat
7:     if MPI_Iprobe then
8:       MPI_Recv(rbuffer)
9:       update face_recv with rbuffer
10:    end if
11:    if MPI_Issend finished then
12:      call MPI_Ibarrier
13:    end if
14:  until MPI_Ibarrier completed
15:  sort(face_recv)
16:  match face_recv
17:  sbuffer  $\leftarrow$  coupled faces
18:  MPI_Issend(sbuffer,p) for p  $\leftarrow$  1, np
19:  repeat
20:    if MPI_Iprobe then
21:      MPI_Recv(rbuffer)
22:    end if
23:    if MPI_Issend finished then
24:      call MPI_Ibarrier
25:    end if
26:  until MPI_Ibarrier completed
27:  update faceList with rbuffer
28:  update MPIFaceRank with rbuffer
29: end procedure

```

achieve these, we have implemented the following three levels of parallelization.

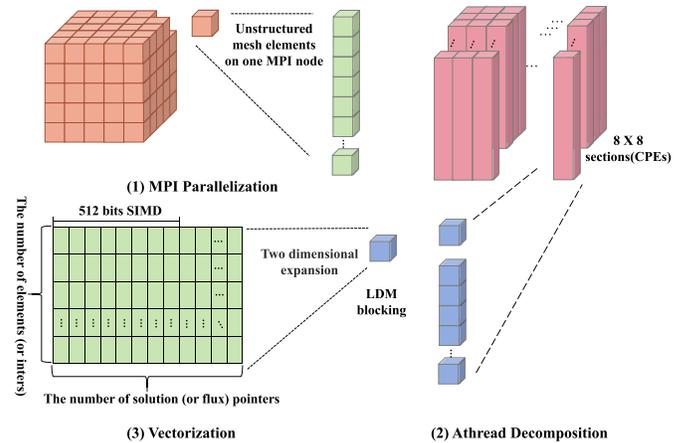


Figure 4: The multi-level parallelization design.

5.2.1 MPI Parallelization. The ZFR solver utilizes a decomposition strategy to partition the entirety of an unstructured grid into multiple sub-computation domains. Each domain is assigned a partition

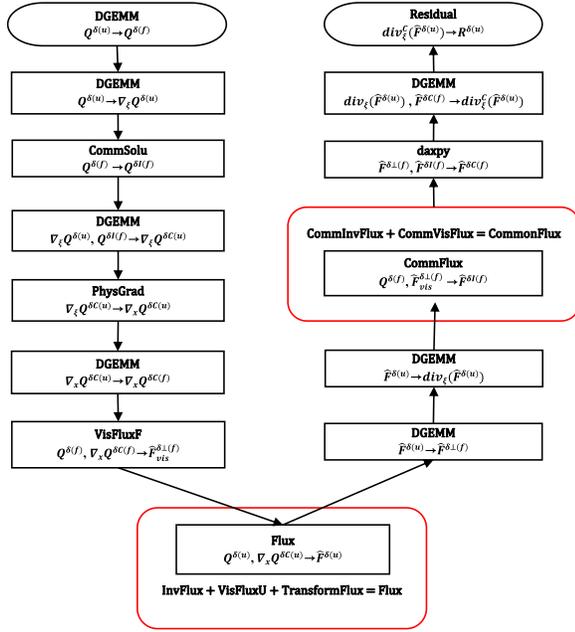


Figure 6: Point-wise kernels fusion scheme.

6.1.1 *Test Case 1.* Performance measurement and accuracy validation are conducted by solving the flow of the LS89 cascade. The test conditions and geometry are based on MUR129 experimental data obtained at the von Kármán Institute [9]. The case specifics are summarized in Table 2. The computational domain with the adopted boundary conditions are shown in Fig. 7. It is noted that two sponge zones are imposed near the inlet and outlet boundaries to damp out any fluctuations and prevent reflections from boundaries.

Table 2: Test Case 1 specifics.

Parameter	Value
Chord (C)	67.647mm
Pitch	0.85C
Stagger angle	55.0°
$Mach_{exit}$	0.84
$Mach_{inlet}$	0.15
Reynolds Number	0.57×10^6

A total of five sets of unstructured meshes were generated, with the element counts given in Table 3. Mesh1 was employed to validate the framework and case setup. This validation run enabled the time averaging, sponge zones, statistics collection, and parallel data output with fourth-order solution polynomials interpolation within each element. Mesh2 to Mesh5 were designed to run the weak scaling test, Mesh2 and Mesh3 were also used for the strong scaling test. The peak sustained performance was measured using the Mesh5.

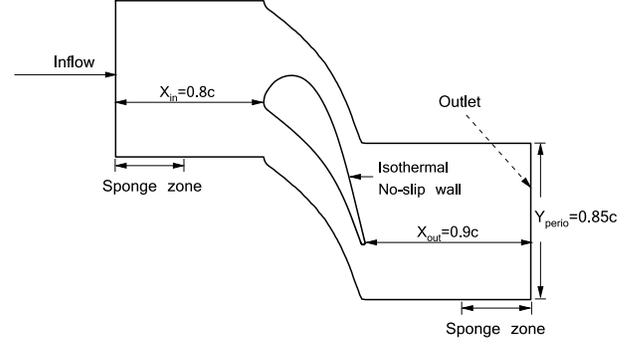


Figure 7: Computational domain of the high pressure turbine test case.

Table 3: Mesh element counts of Test Case 1.

Mesh	Mesh elements counts (million)
Mesh1	1.7
Mesh2	211
Mesh3	422
Mesh4	844
Mesh5	1,689

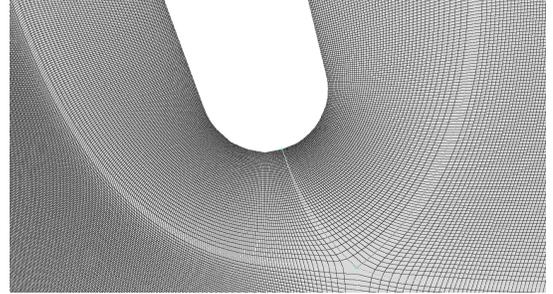


Figure 8: Cross-sectional view of the mesh around the blade trailing edge.

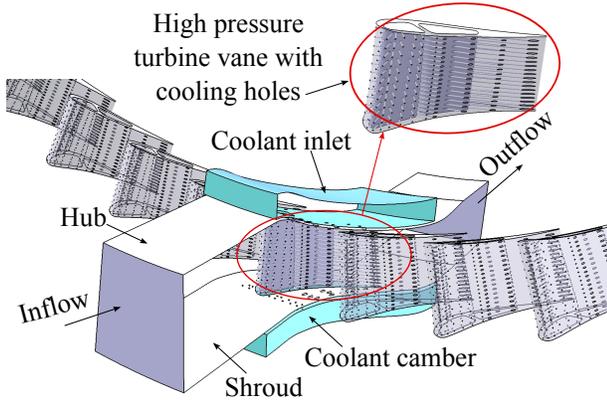
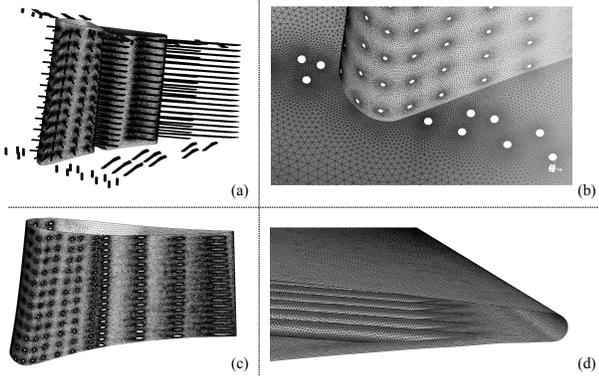
6.1.2 *Test Case 2.* To demonstrate the current solver can simulate complex practical cases, flow within the vane of the first stage high pressure turbine of GE-E³ [10] is calculated. Fig. 9 shows the three-dimensional model of the turbine vane and the boundary condition implemented in the current study. With periodic boundary condition applied in the circumferential direction, one flow passage is solved. Relatively low temperature coolant is injected into the computational domain near the hub. The coolant channels into the internal passage of the turbine vane, and then ejects to the mainstream from the small cooling holes. The detailed boundary condition and mesh element counts can be found in Table 4.

6.2 FLOPs Measurement

All numerical tests were performed on the new Sunway supercomputer using double precision. The performance tests were designed

Table 4: Test Case 2 boundary condition and mesh element counts.

Parameter	Value
Total temperature (main inlet)	709K
Total temperature (coolant inlet)	399K
Total pressure (main inlet)	3.4474×10^5 Pa
Total pressure (coolant inlet)	3.5095×10^5 Pa
Pressure ratio (inlet total to outlet static)	1.63
$Mach_{inlet}$	0.1
Mesh element counts	10 million
Degree of freedom	640 million

**Figure 9: Geometrical model and boundary condition of high pressure turbine vane with cooling holes.****Figure 10: Surface mesh of the high pressure turbine vane on (a) internal channels and cooling holes and (b) the junction of leading edge and endwall and (c) the pressure surface and (d) the trailing edge.**

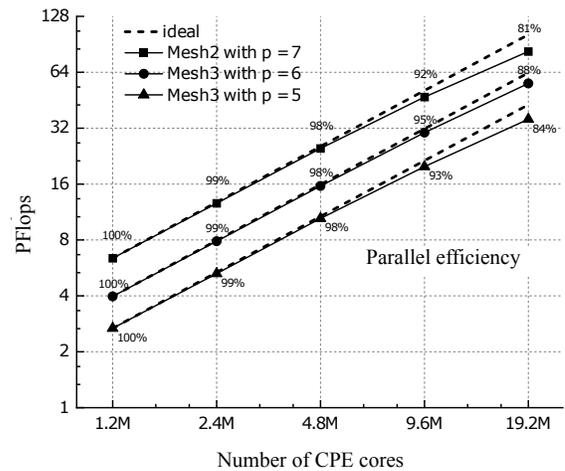
to evaluate the scaling and efficiency of the solver with no post-processing module enabled (i.e. no I/O or statistical data collection). The performance is measured by averaging time taken for one time step, after running a test for 50 time steps. The total number of floating point operations (FLOPs) of one time step is estimated in a

manner similar to that in [33]. The CPE kernels were divided into two groups: matrix multiplications and point-wise calculation. In matrix-multiplication kernels, the floating point operations was counted as $2mnk$, with m representing the number of rows in A and C matrix, n representing the number of columns in B and C matrix, k representing the common dimension of A and B matrix. This estimate was less than the actual FLOPs executed in the ZFR solver, resulting in a conservative estimation of the overall FLOPs. For point-wise kernels, we used two methods to count the number of operations. The first method counted all double-precision instructions in the assembly code, while the other counted the specific algorithm and operation of the numerical scheme. Both methods produced similar operation counts. Here, we chose the latter for the final number.

7 PERFORMANCE RESULTS

7.1 Scaling tests for Test Case 1

7.1.1 Strong scaling. Fig. 11 shows the strong scaling test results on the latest generation Sunway supercomputer. The 211 million and 422 million unstructured meshes of LS89 case were benchmarked and achieved the excellent parallel efficiency from 1.2 million to 19.2 million CPE cores. At 19.2 million CPE cores, the parallel efficiency of three configurations all dropped to below 90%. This is due to the fact that each CPE core is only allocated with 11 elements or 22 elements for Mesh2 and Mesh3 respectively, at which the CPE core cannot be fully loaded. Considering the configuration of Mesh3 with different polynomial orders, it is observed that the efficiency of order $p = 6$ is higher than that with $p = 5$. The strong scaling test result suggests that the element size on each CPE plays a vital role in the parallel performance.

**Figure 11: Strong scaling of ZFR on the latest generation Sunway supercomputer for different meshes and polynomial orders.**

7.1.2 Weak scaling. For weak scaling, four sets of meshes (see Table 3) have been generated, with element counts ranging from 211 million to 1.69 billion. Fig. 12 shows the weak scaling test result,

where 2.4 million to 19.2 million CPE cores are used in the test. Excellent weak scaling performance is observed up to 19.2 million CPE cores with various orders.

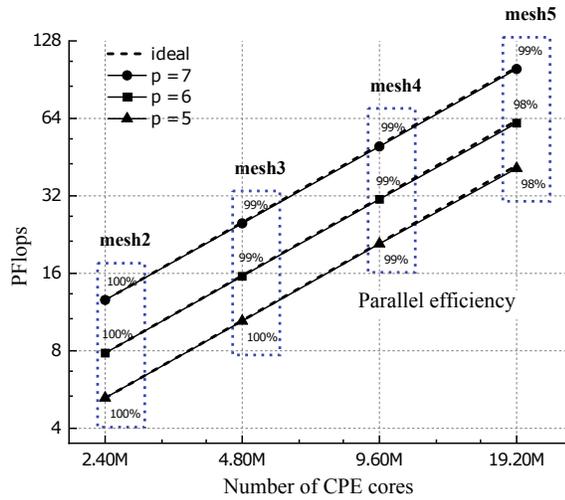


Figure 12: Weak scaling of ZFR on the latest generation Sunway supercomputer for different polynomial orders.

7.2 Peak sustained performance for Test Case 1

The peak performance was benchmarked using Mesh5 with a seventh order of polynomial, resulting in 1.69 billion elements and 865 billion DOFs. The case was ported to 60% of the Sunway machine with 350 thousand MPE cores and 22.4 million CPE cores, achieving a sustained performance of 115.8 DP-PFLOPs.

7.3 Physics revealed through simulation with high complexity, high fidelity, high precision, and high efficiency

7.3.1 Test Case 1. The physics run was performed on Mesh1 with a third order polynomial. The chosen Courant-Friedrichs-Lewy (CFL) value was 1, requiring approximately 100,000 time steps to complete one flow pass over the blade chord. The case commenced with a 1st order of accuracy to flush out transients, and then switched to the fourth order for a high fidelity run. Following the low order run, a total of 300,000 time steps was initiated to achieve a fully developed state, after which time averaging was performed over another 300,000 time steps. The entire simulation was performed on 2000 nodes with 768,000 CPEs for 40 wall clock hours.

Fig. 13 compares the simulation results with experimental data. It is noted that numerical results compare favourably with experimental and numerical results in literature for both heat transfer coefficient and isentropic Mach number over the blade surface. The convective heat transfer coefficient is notoriously difficult to predict with errors of over 50% for RANS simulations [7]. The fact that the current high fidelity simulation can accurately predict the heat transfer coefficient is that the rich flow physics is captured. Fig. 14 shows a snapshot of the instantaneous flow with turbulent flow structures visualized by Q-criterion [34] and shock wave and wake

propagation by pressure gradient in the background. The vortex shedded from the blade trailing edge generates strong pressure waves, which impinge on the neighbouring blade suction surface, creating a region with oscillating heat transfer coefficient (see fig. 13). On the suction surface near trailing edge, boundary layer transition triggered by the upstream moving pressure wave is also evident, which is consistent with the sharp increase of heat transfer coefficient near the trailing edge.

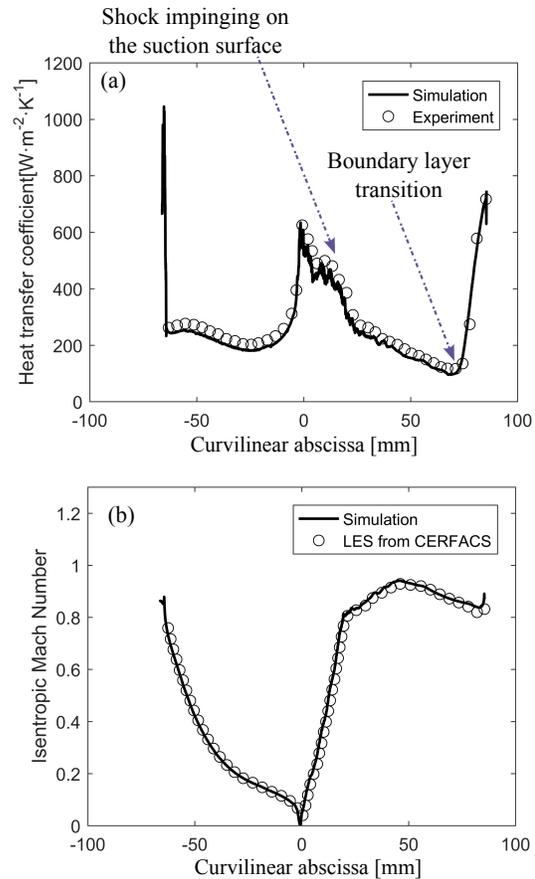


Figure 13: Comparison between numerical results and measurements.

7.3.2 Test Case 2. Only physics runs were performed for Test Case 2. Similar case running strategy as Test Case 1 was used. The case started with 1st order of accuracy and then switched to the fourth order to resolve smaller turbulent scales. Fig. 15 shows the instantaneous temperature field over the blade suction surface with main and coolant flow streamlines embedded. It is evident that the temperature of the blade surface is lowered significantly by the coolant ejecting from the cooling holes. Different to the Test Cases 1, hub and shroud are included in Test Case 2, resulting in the evident endwall flow downstream of the trailing edge. Though there is no surface blade temperature and heat transfer coefficient measurement for this test case, it is believed that this high fidelity

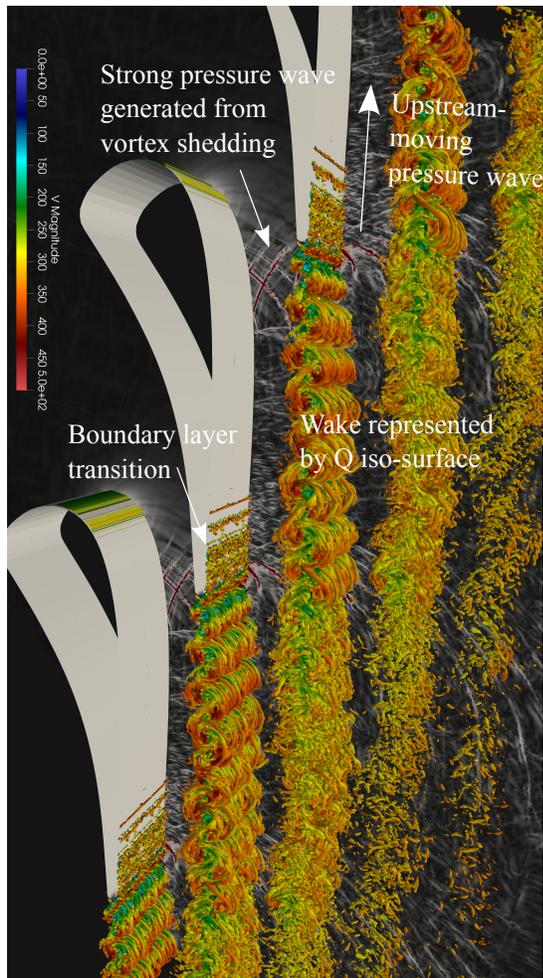


Figure 14: The screenshot of Q-criteria iso-surface colored with velocity magnitude, with background contoured by pressure gradient to visualize the shock wave and pressure propagation.

simulation can confidently predict the blade surface temperature as more physics are captured, such as jet ejecting to the mainstream, endwall flows, wake shedding from the trailing edge, etc.

8 IMPLICATIONS

The present study details the development of a LES code, which is fully portable to the new Sunway architecture. The code includes a parallel pre-processor that can handle large meshes with billions of elements and efficiently find matching faces for MPI communication. The code was validated and profiled using high-pressure turbine testcases, and numerical results compared favorably with experimental data, particularly for the challenging prediction of heat transfer coefficient. The ability to accurately predict heat transfer is crucial for predicting blade temperature and material integrity in industrial turbomachinery applications, particularly for transitional and separated flows. With higher confidence in simulation,

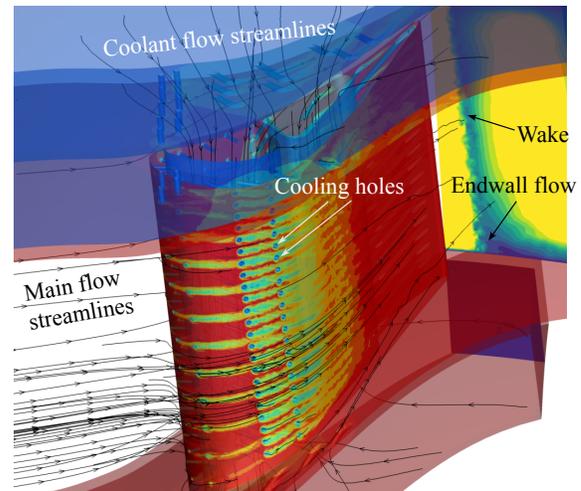


Figure 15: Temperature over the suction surface of the high pressure turbine vane with coolant ejecting from cooling holes.

more boundary-pushing and groundbreaking designs can be made to increase overall engine efficiency and eventually reduce CO_2 . Our work demonstrates the potential of high order FR methods and heterogeneous computing power for LES of turbomachinery flows.

8.1 Parallelization Optimization for FR method on Heterogeneous Computing Architecture

We have demonstrated in this work that the Sunway architecture can be efficiently utilized for high-order CFD simulations on unstructured meshes, through the implementation of multilevel parallelism technique. Our results demonstrate excellent scalability up to 300,000 MPI ranks with 19.2 million CPE cores. Furthermore, while the Sunway architecture differs from other heterogeneous platforms in terms of its memory model, we anticipate that such multi-level parallelism technique can be employed to optimize algorithms for other platforms as well. With support from other technologies, such as adaptive mesh generation with high-order elements, the flux reconstruction method can be a competitive candidate for the future whole engine simulation.

8.2 Large Eddy Simulation for Turbomachinery Flows

Nowadays, CFD plays an increasingly important role in turbomachinery's aerodynamic, thermal, and aeromechanic design, resulting in shorter design cycles, safer and more efficient performance, and reduced weight and costs. However, RANS based turbulence model often underestimates the diffusion-driven turbulent mixing. This affects the prediction of almost every turbomachinery component. LES, on the other hand, offers substantially more accurate and detailed information about the flow field than RANS, but requires drastically more computational resources. In our research, we demonstrated how LES can accurately model sizable sections of practical turbomachinery blades on the emerging exascale computing technology. Since the scale of computing used in cutting-edge

HPC systems, like Sunway, is about 10-15 years ahead of that used by the leading industrial adopter [35], this pioneering research will open the door for future high-fidelity and complex numerical studies for turbomachinery flows, encouraging more innovative designs with short turnaround time to achieve safer and more efficient gas turbines with lower emissions.

ACKNOWLEDGMENTS

We acknowledge the support from the National Nature Science Foundation of China with grant number 52106060 and 92152202. The open-sourced HiFiLES and SU2 project at Stanford University and the PyFR project at Imperial College London inspired some of our algorithm development in this work, which are also acknowledged here. We also thank Will Trojak at IBM for useful technical discussions.

REFERENCES

- [1] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis. Mar. 2014. CFD vision 2030 study: A Path to Revolutionary Computational Aerosciences. Tech. rep. NASA CR-2014-218178. NASA Langley Research Center.
- [2] J.-C. Han, S. Dutta, and S. Ekkad. 2012. *Gas Turbine Heat Transfer and Cooling Technology*. CRC Press, (Nov. 2012).
- [3] R. D. Sandberg and V. Michelassi. 2022. Fluid dynamics of axial turbomachinery: blade- and stage-level simulations and models. *Annual Review of Fluid Mechanics*, 54, 1, (Jan. 2022), 255–285.
- [4] N. Gourdain, L. Y. M. Gicquel, and E. Collado. 2012. RANS and LES for the heat transfer prediction in turbine guide vane. *Journal of Propulsion and Power*, 28, 2, (Mar. 2012), 423–433.
- [5] Y. Zheng and M.-S. Liou. 2003. A novel approach of three-dimensional hybrid grid methodology: part 1. grid generation. *Computer Methods in Applied Mechanics and Engineering*, 192, 37–38, 4147–4171.
- [6] M.-S. Liou and Y. Zheng. 2003. A novel approach of three-dimensional hybrid grid methodology: part 2. flow solution. *Computer Methods in Applied Mechanics and Engineering*, 192, 37–38, 4173–4193.
- [7] J. C. Tyacke and P. G. Tucker. 2015. Future use of large eddy simulation in aero-engines. *Journal of Turbomachinery*, 137, 8, 081005.
- [8] W. Shen and S. A. E. Miller. 2020. Validation of a high-order large eddy simulation solver for acoustic prediction of supersonic jet flow. *Journal of Theoretical and Computational Acoustics*, 28, 03, (Sept. 2020), 1950023.
- [9] T. Arts, M. Lambert de Rouvroit, and A. Rutherford. Sep. 1990. Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade. A test case for inviscid and viscous flow computations. Tech. rep. VKI Technical Note 174. von Karman Institute for Fluid Dynamics, Belgium.
- [10] L. P. Timko. Jan. 1984. Energy efficient engine high pressure turbine component test performance report. Tech. rep. NASA CR-168289. NASA Lewis Research Center.
- [11] J. Cui, V. N. Rao, and P. Tucker. 2016. Numerical investigation of contrasting flow physics in different zones of a high-lift low-pressure turbine blade. *Journal of Turbomachinery*, 138, 1, 011003.
- [12] C. P. Arroyo, J. Dombard, F. Duchaine, L. Gicquel, B. Martin, N. Odier, and G. Staffelbach. May 2021. Towards the large-eddy simulation of a full engine: integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part I: methodology and initialisation. *Journal of the Global Power and Propulsion Society*, Special Issue: Data-Driven Modelling and High-Fidelity Simulations.
- [13] C. P. Arroyo, J. Dombard, F. Duchaine, L. Gicquel, B. Martin, N. Odier, and G. Staffelbach. May 2021. Towards the large-eddy simulation of a full engine: integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part II: comparison against stand-alone simulations. *Journal of the Global Power and Propulsion Society*, Special Issue: Data-Driven Modelling and High-Fidelity Simulations, 1–16.
- [14] S. Lanteri. 2010. Specification report on requirements of application codes. In *EPEEC Project Reports*.
- [15] H. Gopalan, S. Heinz, and M. K. Stölinger. 2013. A unified RANS–LES model: computational development, accuracy and cost. *Journal of Computational Physics*, 249, 249–274.
- [16] J. S. Hesthaven and T. Warburton. 2008. *Nodal Discontinuous Galerkin Methods*. Springer New York.
- [17] H. T. Huynh. 2007. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*. AIAA, (June 2007).
- [18] H. M. Tufo, P. F. Fischer, M. E. Papka, and K. Blom. 1999. Numerical simulation and immersive visualization of hairpin vortices. In *Proceedings of the 1999 ACM/IEEE Conference on Supercomputing*. ACM, (Jan. 1999).
- [19] M. Vázquez et al. 2016. Alya: multiphysics engineering simulation toward exascale. *Journal of Computational Science*, 14, 15–27.
- [20] T. D. Economou, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. 2016. SU2: an open-source suite for multiphysics simulation and design. *AIAA Journal*, 54, 3, (Mar. 2016), 828–846.
- [21] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz. 2012. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61, (May 2012), 86–93.
- [22] D. A. Kopriva and J. H. Koliass. 1996. A conservative staggered-grid Chebyshev multidomain method for compressible flows. *Journal of Computational Physics*, 125, 1, (Apr. 1996), 244–261.
- [23] H. Huynh, Z. Wang, and P. Vincent. 2014. High-order methods for computational fluid dynamics: a brief review of compact differential formulations on unstructured grids. *Computers & Fluids*, 98, (July 2014), 209–220.
- [24] F. D. Witherden, A. Farrington, and P. E. Vincent. 2014. PyFR: an open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach. *Computer Physics Communications*, 185, 11, (Nov. 2014), 3028–3040.
- [25] S. Akkurt, F. Witherden, and P. Vincent. 2022. Cache blocking strategies applied to flux reconstruction. *Computer Physics Communications*, 271, (Feb. 2022), 108193.
- [26] W. Hu et al. 2022. 2.5 million-atom ab initio electronic-structure simulation of complex metallic heterostructures with dgdf. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–13.
- [27] H. Shang et al. 2021. Extreme-scale ab initio quantum raman spectra simulations on the leadership hpc system in china. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–13.
- [28] D. Padua et al. 2011. METIS and ParMETIS. In *Encyclopedia of Parallel Computing*. Springer US, 1117–1124.
- [29] C. Chevalier and F. Pellegrini. 2008. PT-Scotch: a tool for efficient parallel graph ordering. *Parallel Computing*, 34, 6, 318–331.
- [30] H. T. Huynh. 2009. A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. AIAA, (Jan. 2009).
- [31] D. I. Ketcheson. 2008. Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations. *SIAM Journal on Scientific Computing*, 30, 4, (Jan. 2008), 2113–2136.
- [32] F. Liu et al. 2022. xMath2.0: a high-performance extended math library for SW26010-pro many-core processor. *CCF Transactions on High Performance Computing*, 5, 1, (Oct. 2022), 56–71. doi: 10.1007/s42514-022-00126-8.
- [33] P. Vincent, F. Witherden, B. Vermeire, J. S. Park, and A. Iyer. 2016. Towards green aviation with Python at petascale. In *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, (Nov. 2016).
- [34] Y. Dubief and F. Delcayre. 2000. On coherent-vortex identification in turbulence. *Journal of Turbulence*, 1, (Jan. 2000), N11.
- [35] E. Strohmaier, J. Dongarra, H. Simon, and M. Meuer. 2023. The top500 list. <https://top500.org>.