Efficient and affordable high order, high fidelity Large Eddy Simulations for industrial level problems

Yi Lu[1]
Cambridge Flow Solutions Ltd, Compass House, Vision Park, Histon, Cambridge CB24 9AD, UK

Kai Liu[2]
BoXeR Solutions KK, Chuo-ku, Kobe, Japan, 651-0087

W.N. Dawes[3]
Whittle Lab, University of Cambridge, Cambridge, CB3 0DY, UK

For several years we have been developing a fast, high fidelity, high order Large Eddy Simulation system targeted at cheap, commodity, small scale, many-core computing clusters. Our objective is to deliver efficient & affordable application of LES to complex, multi-scale, real geometry, industrial class problems. Our technical approach is based on an innovative space time extention of a high order Flux Reconstruction method (STEFR) which allows time-accurate local time stepping.

In this paper, two numerical investigations are presented: a transonic high Reynolds number NASA acoustic reference nozzle and a real geometry aircraft nose landing gear used in the BANC Workshops. These are presented together with computer resource statistics, aerodynamic and acoustic noise comparisions with experimental data and power spectrum density (PSD) plots. The Fflowcs William-Hawkings (FWH) integration is used for acoustic post-processing for the transonic nozzle case on a virtual structured surface mesh around jet flow.

Additionally, this paper presents an innovative Hierarchical Proper Orthogonal Decomposition (HPOD) method which has been developed for data analysis in parallel, on-the-fly, and which extracts the most energetic modes by performing a two-level orthogonal projection as a Reduced Order Method. This has the potential to save post-processing memory requirement significantly for analysis of massive "big data" high order transient results.

Nomenclature

CFD	Computational fluid dynamics
DOF	degree of freedoms
LES	large eddy simulation
HPC	high performance computing
HPOD	hierarchical proper orthogonal decomposition
FWH	Ffowcs-Williams and Hawkings(FW-H) integration
SPL	sound pressure level
STEFR	space time extention of the high order Flux Reconstruction method
PSD	power spectral density
dt	time step
t	physical time of simulation
T_p	flow passing (through typical length scale) time of LES
u, v, w	Cartesian fluid velocity components
X, Y, Z	Cartesian coordinates
	Subscript:

[1] CFD Engineer, Cambridge Flow Solutions Ltd., Cambridge, UK, and AIAA Associate Member.
[2] CFD Engineer, BoXeR Solutions KK, Kobe, Japan.
[3] Professor, Whittle Lab, University of Cambridge, UK, and Senior AIAA Member.

∞ free-stream quantity

I. Introduction

With continued development of computing ability and numerical schemes, it is more and more attractive to perform high fidelity flow simulations in both academia and industry. These high fidelity simulations are commonly large eddy simulations (LES) and direct numerical simulations (DNS). By resolving small turbulence scales directly, LES and, even more so DNS, have a more general ability to handle turbulence without modeling. There are three main challenges for high fidelity Large Eddy Simulations for industrial class problems: the ability to handle geometry complexity; efficient discretization in both space and time for low, affordable computer running cost; and fast data extraction and low storage requirement for overall post-processing. The simulating system, *HOTnewt*, under development at Cambridge Flow Solutions Ltd[1], for several years[2~4] attempts to address these requirements with the following features:

1. *High order coarse mesh generation*: the higher order coarse mesh generation is based on the commercial, octree level-set based, mesh generation software *BOXERMesh*[1], which is fully parallized, highly CAD-tolerant and automated for high quality mesh generation of real geometries. The functionality for higher order coarse mesh generation and smoothing are implemented as extensons of *BOXERMesh*, and enables the ability of *HOTnewt* to perform high order simulations for multi-scale, large-scale complex geometry problems with very limited cost and human resource in pre-processing.

2. *Efficient space and time discretization*: the basic numerical algorithm is based on the Flux Reconstruction (FR) method which was firstly introduced by Huynh[5] for 1D problems and extended to simplex elements by Gao and Wang[6]. The FR formulation on general 3D hybrid meshes including tetrahedrons, prisms, pyramids and hexahedrons was implemented in 2012[7]. The FR method is able to achieve arbitrarily high order accuracy, is completely local in-cell, is stable[8] and very efficient because of its simple differential form[9]. Inspired by the space-time extension of high order discontinuous galerkin (STEDG) work[10] by Gassner etc, our Space-Time Extension of Flux Reconstruction (STEFR) method was developed by separating the two parts of the FR formation, the continuous Runge-Kutta solution[11] of the dominant divergent and the time-integration of the correction flux part, as predictor and corrector, respectively, which allowes time-accurate time-stepping. This novel feature helps to achieve very high speed up ratios (10~100)[2~4,12,13] for large scale, multi-scale complex geometry simulations compared to conventional uniform time-stepping for unseady simulations.

3. *Efficient implementation on many core computing systems*: in order to achieve higher economic and energy (running) efficiency for the large scale LES simulation, the *HOTNewt* code is implemented on a system of Intel PHI co-processors in "offload mode" to make use of the very high computing ability of this modern many-core system, and combined with normal CPUs to produce a heterogeneous computing environment, to enable a very careful and optimum balance between computing ability and memory consumption.

However, hand-in-hand with advances in simulation technology goes the post-processing of the massive "big data" unsteady results. This brings the huge challenge of storage and computing cost for practical IO operation. There are two types of basic post-processing requirement for unsteady data: one is the transient data extraction and recording such as flow quantities on probe points, surface and iso-furface data-extraction; another is the analysis of unsteady volume data such as unsteady flow structures and their coherent connections. In this paper, for the post-processing analysis of unsteady volume data, a new Hierarchical Proper Orthogonal Decomposition (HPOD) method is proposed with two main objectives: one is to perform reduced order analysis for large scale high order simulations on limited computing resource, for the analysis of flow mechanisms and extraction of industrially interesting information; another is to explore the relations between different frequency parts of flow field, aiming to construct a multi-level local filter, which could help to reduce computing cost dramatically for high fidelity simulations. In HPOD, multi-level solutions in orthogonal modal space are extracted on-the-fly during the high order large eddy simulations, the memory requirement and computing cost of POD analysis on low order modal solutions are much less than the analysis of original nodal high order solutions, which makes the POD analysis affordable on modest computing resource even for very large scale simulations. These lower order modal solutions could be regarded as the filtered low frequency part of the flow field, which contains the main flow information of most interest to industry, whose connection with higher order parts of the complete modal solutions can be investigated as part of the POD analysis.

Accordingly, this paper is formatted as follows. First, the case for using higher order methods is reviewed. Then, the flow simulation system is described, followed by high order Large Eddy Simulations for two cases. These are presented and discussed, with detail comparisons and statistics for accuracy, computer efficiency and running cost. Next, the numerical algorithm of new HPOD approach is introduced, followed by statistics and detail investigation and comparison of HPOD analysis for the two LES cases. Finally, some conclusions are drawn.

II. Higher order simulation system

II.A The case for higher order methods

There is widespread misunderstanding about the role & advantages of higher order methods. A key measure of the usefulness of a simulation is the level of computer resource needed, in simple terms the power in kWh and elapsed wall-clock time needed to perform a simulation in which the appropriate physical space & time scales are successfully resolved. Clearly higher order methods deliver higher accuracy than lower order methods on the same mesh but with more floating point operations needed to deliver that accuracy– so the key question really is: *which approach uses less resource: a low order method on a finer mesh or a high order method on a coarser mesh.*

This was addressed by Leland Jameson[14] who in a pioneering paper studied the mesh resolution required to follow vortical motions over long time scales – as in LES. The key is to recognize that higher order methods can resolve a given wavelength in the flow with far fewer "Points Per Wavelength (PPW)" than can lower order approaches. Figure 1(a) illustrates this trend for P1, P2 & P3 discretisation (piecewise linear (*second order*), quadratic (*third order*) & cubic (*fourth order*) respectively). This shows that a low order P1 method might need a factor of ~10 more PPW in 1D than a P3 method – which in 3D this becomes an astonishing factor of 1000 ! Since PPW and the associated DOFs (Degrees of Freedom) translate directly to computer memory, RAM, requirement then provided the higher order method not consume too much extra memory per DOF then a well-designed algorithm could permit a better computer memory, run-time - job size trade-off. This scale of this resource utilization trade is emphasised, for example, also in a NASA[15] publication.

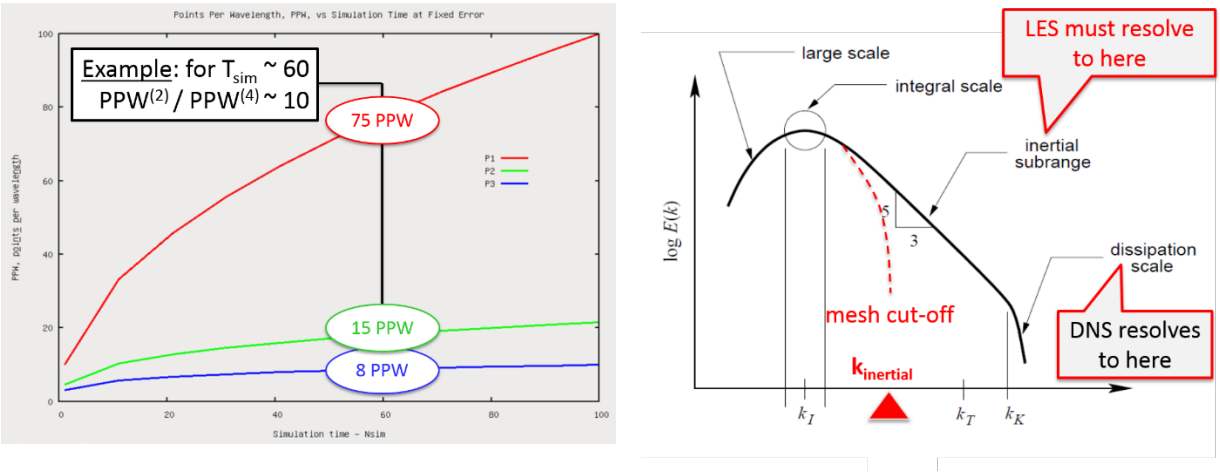


Fig.1: (a) Points per Wavelength: Simulation time T_{sim} for fixed total error E_{sim} for P1, P2 & P3 discretisation; (b) the classic turbulence energy cascade over various length scales.

As an example to make this concrete, consider an LxLxL 3D domain with L=0.1m and $Re_L=10^6$. There are several scales of interest: the Integral scale (or "outer" scale), I, commonly the scale is defined as $I \sim 0.1$ L where L is the scale of the flow domain; the Taylor microscale within the inertial subrange, λ ; and the Kolmogorov (or "inner" scale), η. These scales are related by: $\eta/I \sim Re_l^{-3/4}$ & $\lambda/I \sim Re_l^{-1/2}$ where $Re_l = |u'|l/\nu$ is the turbulent Reynolds number. Hence in terms of wavenumber: Taylor scales: $k_T \sim k_I Re^{+1/2}$ & Kolmogorov scales: $k_K \sim k_I Re^{+3/4}$. Therefore, in our example, with the turbulence length scale I =0.1L and (say) |u′|=10%U so that the turbulence Reynolds number is $Re_I = 10^{+4}$ we have the wavenumbers associated with the Integral scale $k_I \sim 314$ and the Taylor

microscale $k_T \sim 31400$. Therefore, we can select a target wavenumber within the inertial subrange, see Figure 1(b), for the simulation to resolve – here we select $k_{inertial} \sim 3140$. The following Table shows the mesh sizes that would be needed to resolve to the target wavenumber and also shown are the total Degrees of Freedom (DOF) and an estimate of the floating point operations needed (from our own code – *HOTnewt*).

Order	DOF/cell	PPW	Mesh size (B)	Relative size	DOF (B)	Floats/Δt/ mesh cell	Relative floats
P1	8	75	421.	1	16840.	3.6×10^3	1
P2	27	15	3.3	0.0078	445.	6.9×10^3	1.94
P3	64	8	0.51	0.0012	163.	14.7×10^3	4.09

Table 1: Mesh statistics needed to resolve to the target wavenumber in the example

The difference in mesh size is astonishing – but of course higher order methods consume more floats per cell to achieve the higher accuracy – so if we define the *Relative Cost = relative mesh size x relative floats* we see:

Order	Relative Cost
P1	1
P2	0.015 = 1/66.2
P3	0.0049 = 1/203.

Table 2: Relative cost of different order methods on the example

The potential reduced Relative Cost – which translates directly into reduced computer power requirements – derived from higher order methods is astonishing – and, as we'll see, the key to unlocking their potential turns out to be the ablity to develop a sufficiently coarse, higher order mesh.

II.B Our LES solver – *HOTnewt*

The flow solver code based on high order STEFR method, *HOTnewt*, under development at Cambridge Flow Solutions Ltd[1], uses general hybrid unstructured meshes and has been used to solve a wide range of problems[2~4,12,13].

An important part of the STEFR method is that every cell uses their local timestep during the unsteady simulation. The time marching for every single synchronous step, which is commonly $C * dt_{max}$, where dt_{max} is maximum timestep for all cells, and the coefficient $C > 0$ is integer. The local timestep for the i-th single cell can be expressed as $dt_i = dt_{max}/2^{L_{max}-L_i}$, where L_i is the time level and $dt_{max} = 2^{L_{max}} dt_{min}$, dt_{min} is the global minimum timestep among all cells computed before actual calculation for every single synchronous timestep. In the time-marching process for a single synchronous step, the number of "prediction" times for i-th single cell, is $C * 2^{L_{max}-L_i}$, and the number of total loops for "prediction" operation of all cells are $C * 2^{L_{max}}$, the all cells for j-th loop satisfy $mod(j * dt_{min}, dt) = 0$, which means the number of cells and computing time are variable for each loop as shown in Figure 2. Therefore, this time marching is not uniform and the data-communication is irregular. In order to achieve higher computing efficiency and lower running cost, *HOTnewt* has been implemented on many-core computing system based on Intel Xeon PHI co-processors, and it is flexible enough to be readily ported also on other many-core system such as NVIDIA Tesla GPUs and AMD GPUs.

In this paper, all simulations were ran on our own very small scale many-core heterogeneous computing cluster as shown in Figure. 3, consisting of 8 nodes, each node has 2 Intel Xeon CPUs each with 8 physical cores and 6 many-core Intel PHI cards each with in turn 57 physical cores. All components are commodity items, easily and cheaply available and occupies minimal space in a normal computer cabinet. The system provides computing capability equal to about 600~800 conventional HPC CPU cores but with a maximum power consumption is only about 12kWs. The

biggest case we can run on this cluster is the real geometry nose landing gear acoustic case which has about 19 million cells and about 1 billion DOFs. The system architecture and *HOTnewt* communication model is illustrated in Fig.4.

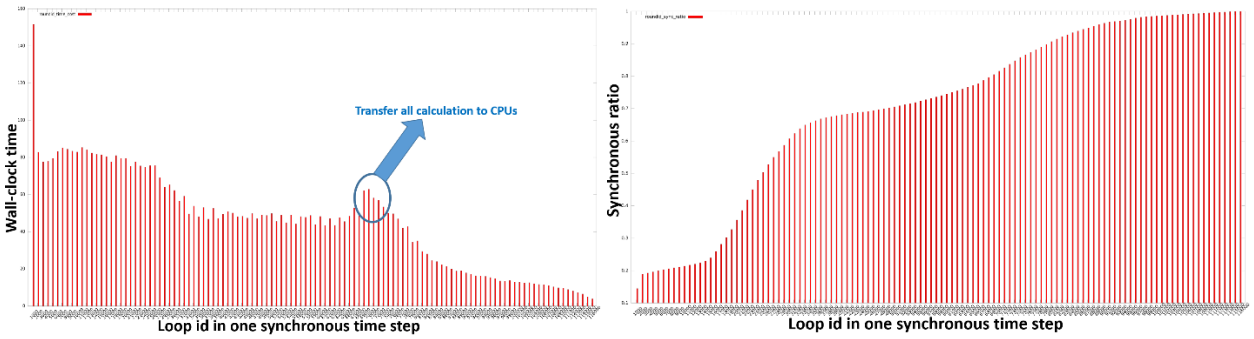


Fig.2: Snapshots during the time marching of one synchronous step: wall-clock time(left) & consumption and accumulation synchronous ratio (right) for every loop (from the nose landing gear acoustic case).

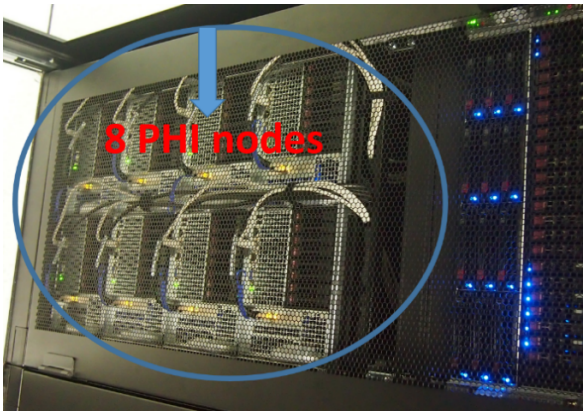


Fig.3: Our heterogeneous computing cluster

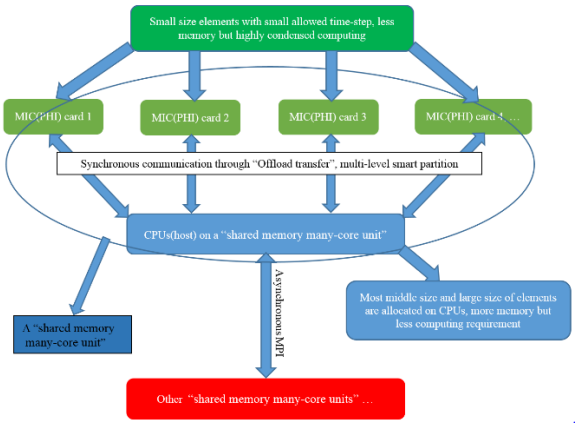


Fig.4: Communication model on many core system

III. Numerical validations

III.A NASA Acoustic Reference Nozzle

Simulation & data comparison

Our LES solver was applied to a 3D case consisting of a jet nozzle with 6 chevrons, corresponding to the NASA SMC001 nozzle tested by Bridges & Brown[16] and shown in Fig. 5. The configuration was run in the "cold" condition at M_{jet}=0.97 and a Reynolds number based on the nozzle exit diameter of 1.03×10^6. This is an aeroacoustic case that has been used widely as RANS and LES benchmarks (for example Uzun *et al.*[17], Palaith *et al.*[18], Bres *et al.*[19]). Results from the *HOTnewt* simulation are compared to data from both experiment and from previous numerical results: time averaged centreline axial velocity, turbulence spectrum, acoustic predictions at 40D from the jet exit plane. *HOTnewt*'s computational efficiency is also compared to that of previous simulations by these various groups, and shown to offer a significant improvement over these approaches. A comparison of estimated computational cost is put forward, which

suggest that the cost of a *HOTnewt* simulation is also considerably lower, offering much more affordable LES simulation on cases of industrial relevance. The memory requirements are also found to be low.

Geometry	Nozzle inlet total pressure p_0(Pa)	Nozzle inlet total temperature T_0(K)	Free stream static pressure p_∞(Pa)	Free stream temperature T_∞(K)
SMC001 nozzle	1.78×10^5	286.4	9.7×10^4	280.2

Table 3: Jet flow conditions

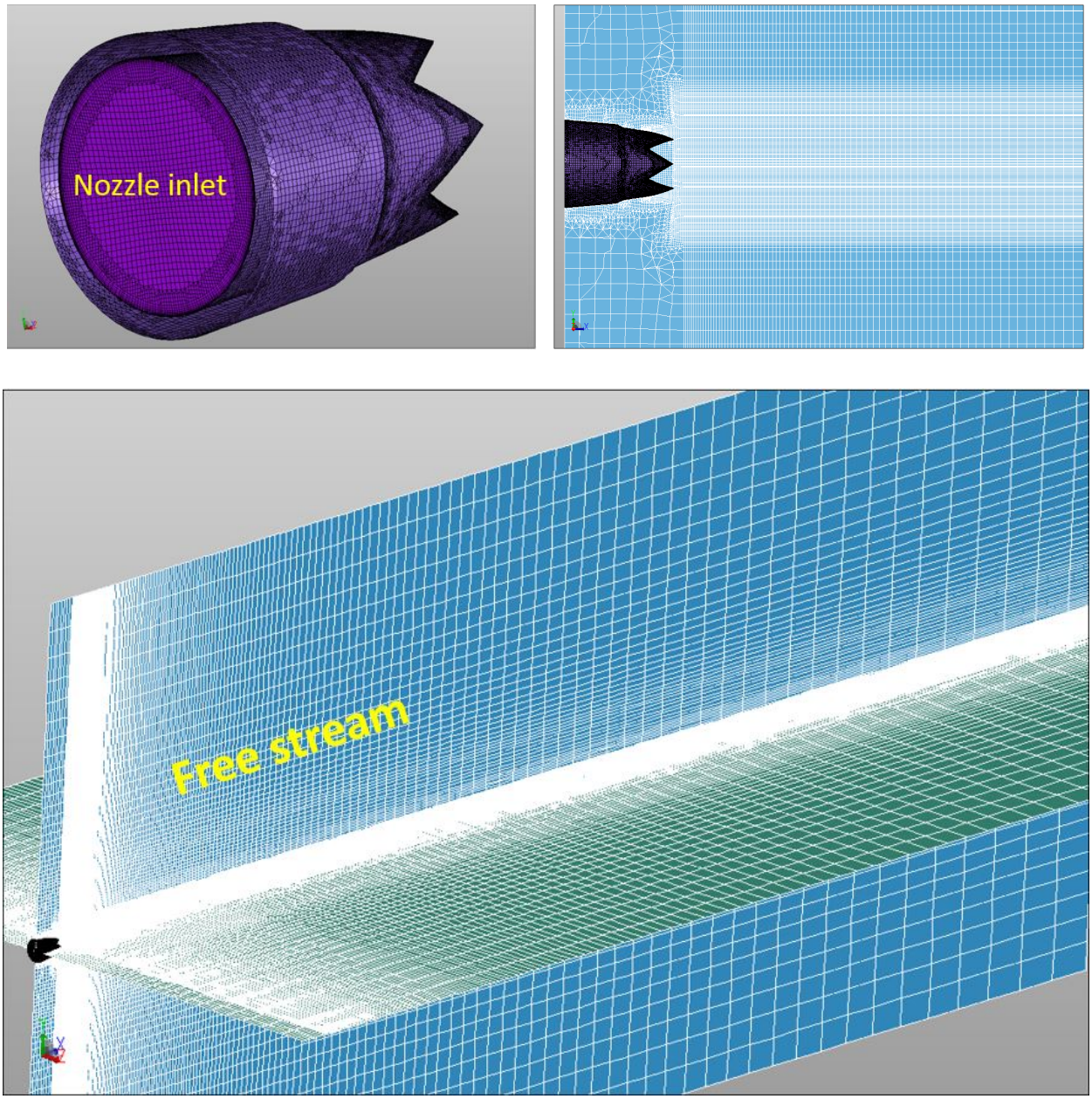


Fig.5: Views of mesh generated for the Chevron Nozzle; this used the *BOXERmesh* Bounding Box Extension to extend the mesh local to nozzle efficiently to the farfield.

Some results from the simulation are presented in Figure 6-8. Figure 6 shows a comparision of time-averaged centreline u-velocity with experimental measurement. The LES was run in p-adaptive mode with third order accuracy in the region occupied by the jet and second order further out into the far-field. These third order predictions agree very well with experimental data and are much better than the globally second order LES. Figure 7(a) shows a snapshot of instantaneous Mach number and Fig.7(b) the time-averaged Mach number. Finally, Figure 8 shows snapshots of instantaneous Q-criterion illustrating the jet structure. The following Section will describe the acoustic post-processing.

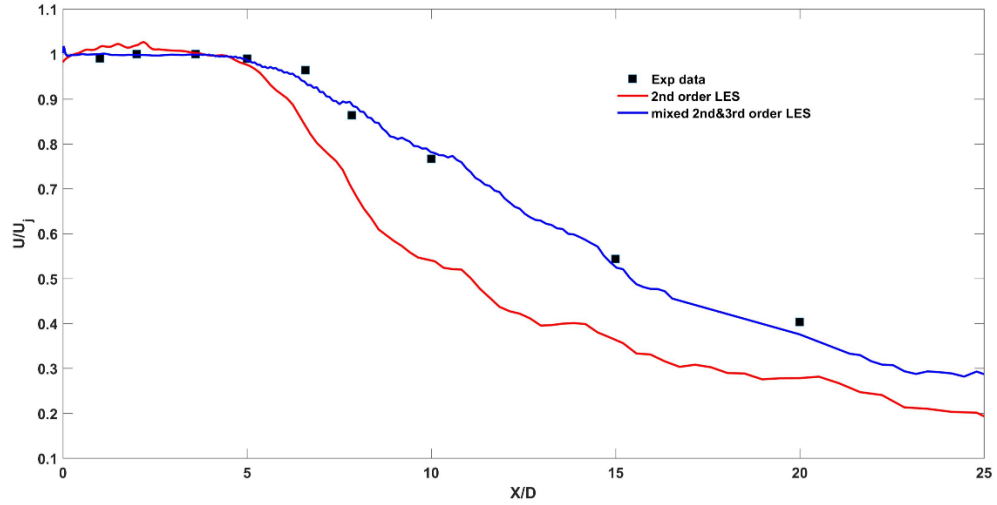


Fig.6: Comparision of time-averaged velocity for centreline u-velocity.

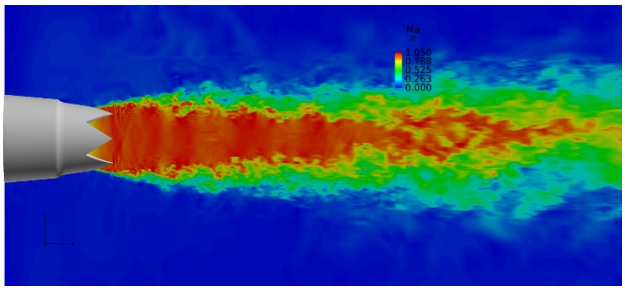


Fig.7(a) Instantaneous Mach number

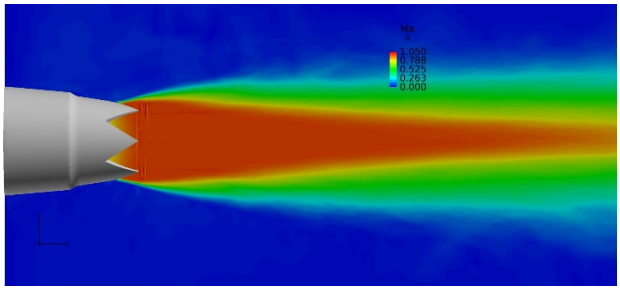


Fig.7(b) Time-averaged Mach number

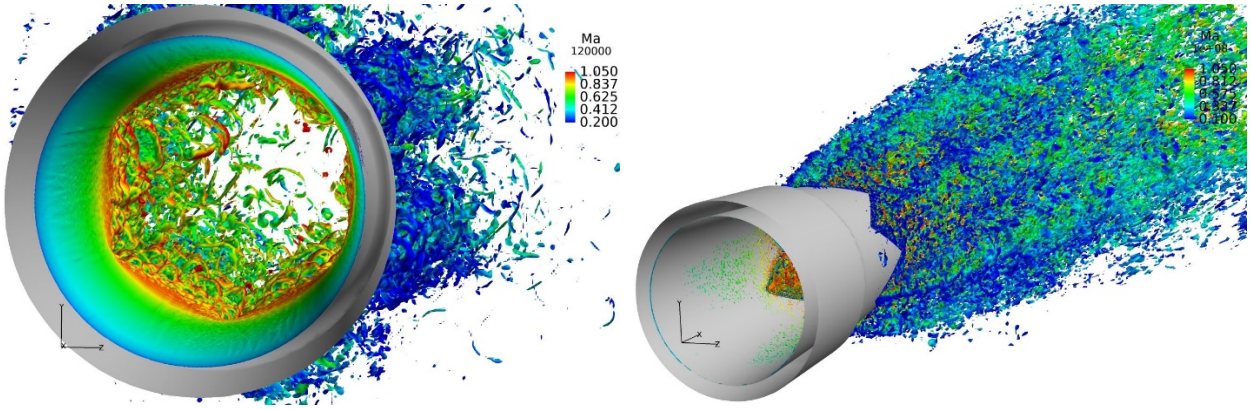


Fig.8: Snapshots of instantaneous Q-criterion illustrating the jet structure, colored by Mach number

Computer Resources

Case ID	Order of accuracy	Number of cells	Number of DOFs	Speed Up Ratio	Number of nodes on our cluster	Equivalent number of CPU cores	Memory consumption (GB)	Wall-clock time for $1T_p$ (hours)
SMC001_1	2^{nd}	9.04M	338.4M	16.3	1 (8 PHI cards)	80	167	5.09
SMC001_2	p-adaptive 2^{nd} /3^{rd}	7.66M	671.1M	29.2	4 (3 PHI cards each)	160	317.4	9.72

Table 4: Statistics for the transonic SMC001 nozzle case.

The flow passing time is defind as $T_p = U_j/D$, where U_j is the exit mean jet flow from Table 3 and $D = 0.0522m$ the diameter of the nozzle. It is observed that using only 160 equivalent typical HPC CPU cores as computing resource, the wall-clock time is only 9.72 hours cost for 1 T_p for the larger scale simulation with about 670M DOFs.

Comparison with other published simulations is always difficult but, for example, Xia *at al*[20] presented second order FVM simulations for the SMC001 on a mesh with 12.5M DOFs needing about 2 hours wall-clock for every T_p on 128 HPC CPU cores. Our second order run, SMC001_1, used 9.04M cells with 8 FR DOFs per cell (see P1, Table 1) – so, with 5 flow variables, a total of 338.4M DOFs; the Xia *et al*[20] case had 5 flow variables x12.5M FVM mesh cells, ie. 62.5M DOFs. So, scaling our *HOTnewt* second order run to match the number of DOFs and cpu cores used by Xia *et al*[20] case shows our method to be about a factor 3.4 faster in wall clock terms. Taking into account the much improved energy efficiency of the Intel PHI cards, Jaeggi[21], means our simulation consumes over a factor 30 less electrical power!

Acoustic post-processing

The classic fflowcs Williams-Hawkings (FWH) integration[22] is used as acoustic post-processor, to compute the far-field sound. Firstly, a surface S is defined which is expected to be large enough and far enough from the jet exit, to include the noise sources; see Figure 9 below. The far-field acoustic pressure fluctuation $\hat{p}'(\boldsymbol{X}, t)$ at the observation point, is calculated from the following integral equation

$$4\pi\hat{p}' = \frac{\partial}{\partial t}\int_S \left[\frac{\rho u_n}{|\vec{r}|}\right] dS + \frac{1}{c_\infty}\frac{\partial}{\partial t}\int_S \left[\frac{p'_{nr}+\rho u_n u_r}{|\vec{r}|}\right] dS + \int_S \left[\frac{p'_{nr}+\rho u_n u_r}{|\vec{r}|^2}\right] dS \quad (1)$$

In the above, $\vec{r}$ is the vector from integration point location to the observer location, defining $\vec{n}$ is the unit surface outer normal vector, $p'_{nr} = p'\vec{n}\cdot\vec{r}/|\vec{r}|$, $u_n = \vec{u}\cdot\vec{n}$, $u_r = \vec{u}\cdot\vec{r}/|\vec{r}|$ and the fluctuating pressure pressure $p' = p - p_\infty$, where p_∞ is the far-field free stream pressure. Respectively, the far-field overall sound pressure level is defined as:

$$OASPL = 20\log_{10}\left(\frac{\hat{p}'_{rms}}{p_{ref}}\right) \quad (2)$$

where $p_{ref} = 2\times10^{-5}$ Pa.

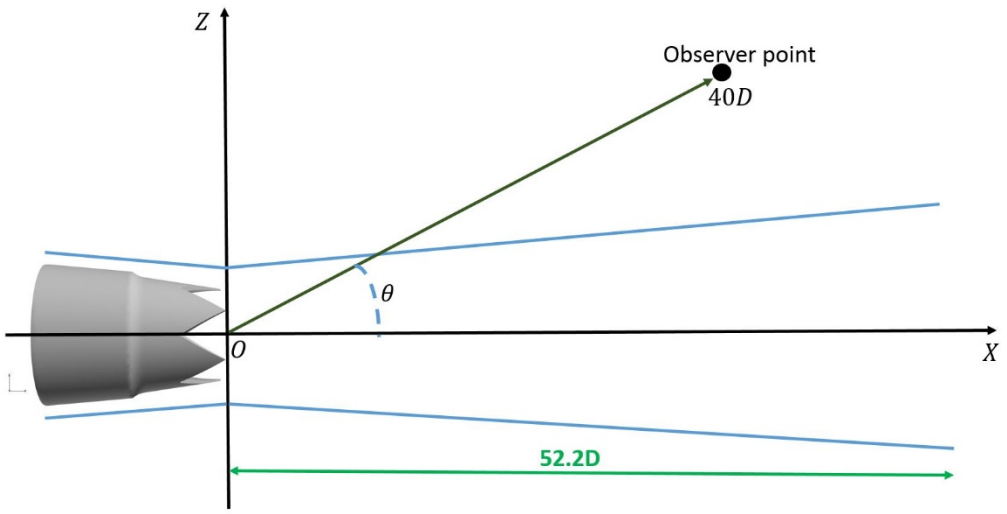


Fig.9: Representation of the FW-H surface (blue line) and the far-field observer position.

The current flow solver HOTnewt, is based on the time-accurate local-time stepping method STEFR, which intended to permit the synchronous time step $\Delta t_{sync} = C\Delta t_{max} (C > 0$ is integer) scale to the size of largest cell of the computing mesh; however, Δt_{sync} is obviously too big a time scale for acoustic integration. Therefore, in this work, in order to perform a high fidelity FW-H integration, a simple structured mesh, which is rotated to form the FW-H surface in the shape of two connected cones (as shown in blue in Figure 9) is embedded within the main solution mesh. A total of 114160 quadrilaterals were generated for this embedded structured mesh and the 4-node Gauss quadrature points of each quadrilateral were used as "Probe points", to record the locally instantaneous result with the local time step of the 3D element where the "Probe point" is located.

The right hand side integral of Equation (1) is performed on each quadrature point ("Probe point") on each assigned physical time t_i^{FW-H}, and the timestep dt^{FW-H} is set close to Δt_{min} for all of these "Probe points".

Define functions F1 and F2 as:

$$F1 = \left[\frac{\rho u_n}{|\vec{r}|}\right] + \frac{1}{c_\infty}\left[\frac{p'_{nr}+\rho u_n u_r}{|\vec{r}|}\right] \tag{3}$$

$$F2 = \frac{p'_{nr}+\rho u_n u_r}{|\vec{r}|^2} \tag{4}$$

Then substitute Eq.(3) and Eq.(4) to Eq.(1), the pressure fluctuation far-field acoustic $\hat{p}'(\vec{r}_i, t)$ could be simply calculated as follow:

$$\hat{p}'(\vec{r}_i, t) = \sum_{s=1}^{N_s}\sum_{q=1}^{4}|Jac|_{s,q} w_q \left[\frac{dF1(t)}{dt} + F2(t)\right]_{s,q}(\vec{r}_i) \tag{5}$$

where $|Jac|_{s,q}$ is the determinant of the Jacobian matrix of the coordinate transformation for q −th quadrature point on s −th 2D quadrilaterals, w_q is the quadrature weight for q −th classic Gauss quadrature point.

Figure 10(a) and 10(b) show the results in the form of the SPL directivity at the observer location 40D predicted by the present simulation compared with experimental data[16]; and (b) a Power Spectra Density at the observation point with $\theta = 40°$. The level of agreement is considered to be satisfactory and expected to be improved with longer physical time marching.

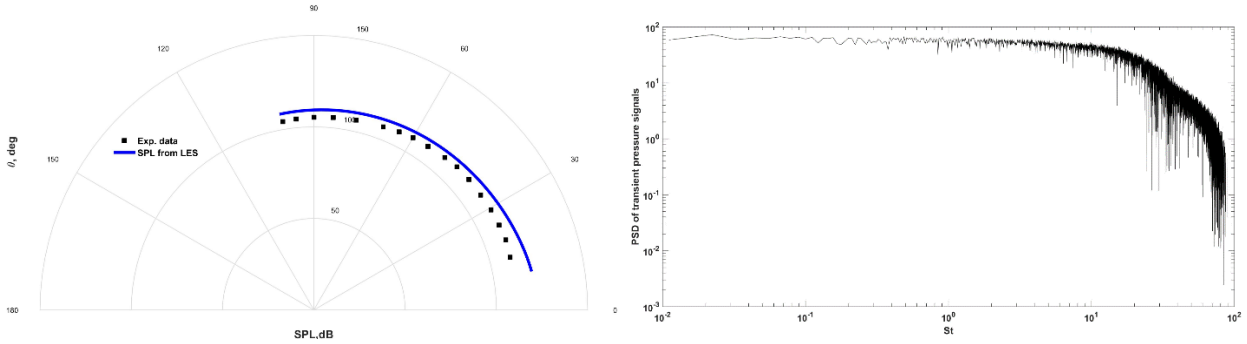


Fig.10: (left) The SPL directivity at observation points with $R = 40D$ predicted by the present simulation compared with experiment[16]; and (right) a Power Spectra Density of density at observation point with $R = 40D$, $\theta = 40°$ as shown in Fig.9.

III.B NASA BANCII Landing Gear acoustic case

Simulation & data comparison

The test model is a ¼-scale, high-fidelity replica of a Gulfstream G550 nose landing gear which includes part of the lower fuselage section. A series of wind tunnel experiments has been performed (Khorrami[23]) in the Basic Aerodynamic Research Tunnel (BART) at NASA Langley Research Center for extensive aerodynamic measurements, and in the open-jet University of Florida Aeroacoustic Flow Facility (UFAFF) mainly for corresponding acoustic measurements. The current simulations were performed at a freestream Mach number of 0.166, which is identical as the experiments which consist of both aerodynamics and acoustic measurements. The detail farfield flow parameters are set as $u_\infty = 56.3\text{m/s}, T_\infty = 286℃, P_\infty = 99241\text{Pa}$, which results in a Reynolds number of 7.3×10^4 based on the main strut (piston) diameter.

We have presented preliminary simulations for this case in LuYi *et al*[2]; this paper presents refined results and more data comparions and computer resource discussions. Figure 11 shows the background P1 mesh (left) generated by *BOXERmesh*, and the high order curved surface (right) rendered by gmsh[24]. Three simulations were performed and for the most resolved we used 14.65M mesh cells with correspondingly nearly 1B DOFs. Figure 12 shows iso-surface (visualized by wireframe) of Q-criterion (Q=800000) for a snapshot of a transient result for the testcase, coloured by Mach number, for our most refined case: case *Landing-Gear-3*.

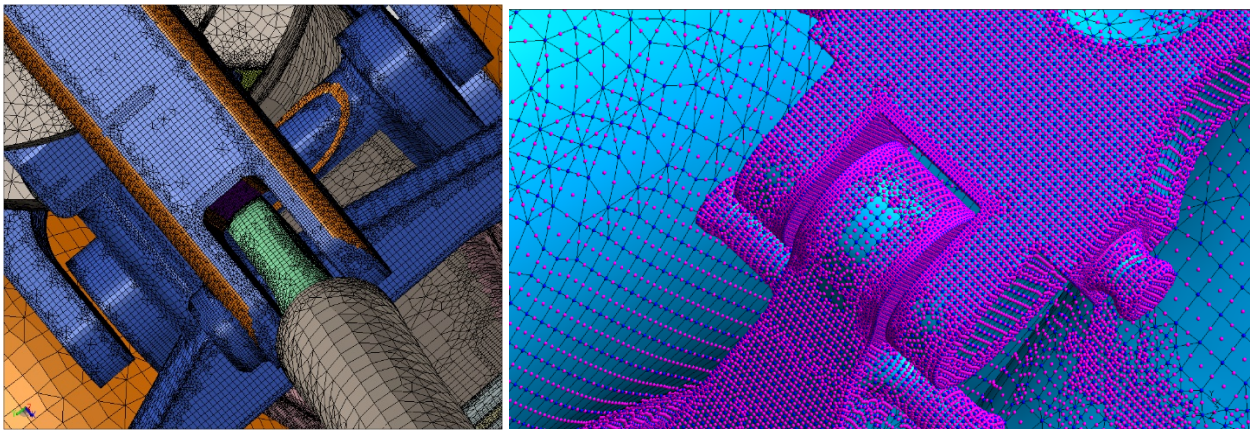


Fig.11: Background P1 mesh (left) generated by *BOXERmesh*, and the high order curved surface (right) rendered by gmsh[14], for NASA/Gulfstream Landing Gear Case.

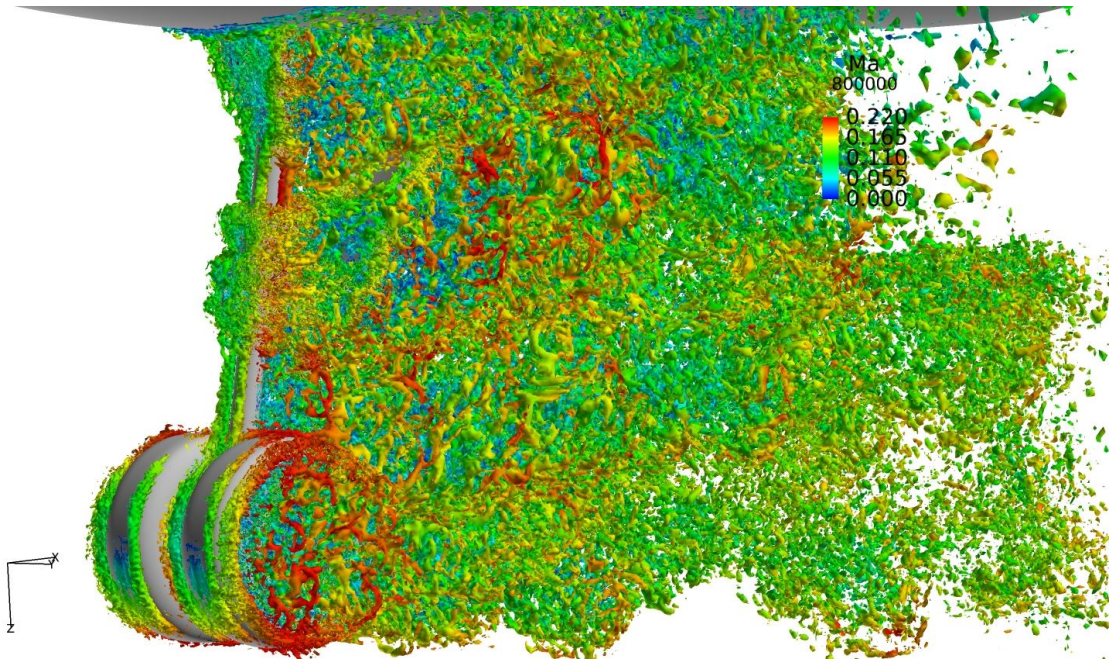


Fig.12: Iso-surface (visualized by wireframe) of Q-criterion (Q=800000) for a snapshot of a transient result for NASA/Gulfstream Landing Gear Case, coloured by Mach number, for case *Landing-Gear-3*.

Starting with basic near-field aerodynamics, a comparison of predicted time-averaged and measured pressure distribution around one wheel is shown in Figure 13; the third order simulation is in good agreement with the experimental data[23] and much better than the second order simulation. Figure 14 compares spanwise vorticity contours on the mid-wheel plane, left for experiment PIV data and right for the simulated case *Landing-Gear-3*; it is difficult, however, to conclude much from this as the PIV is very weakly resolved.

In terms of acoustics, the final result is shown in Figure 15 with a farfield Power Spectral Density extracted at a farfield probe point – taken directly from the simulation, not using the FW-H analogy.

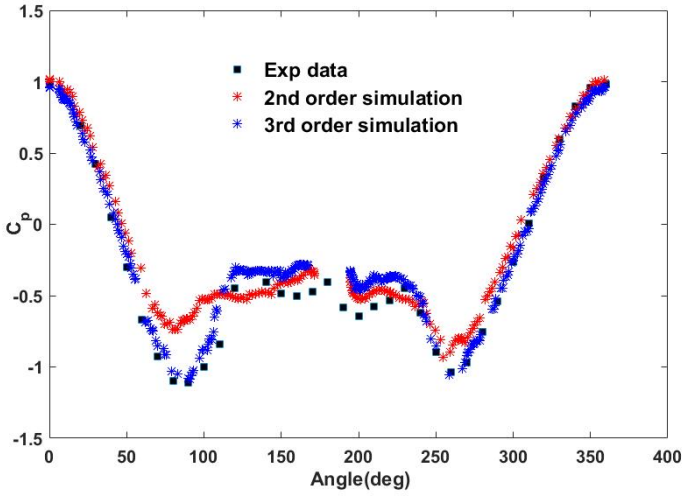


Fig.13: Comparison of predicted time-averaged and measured pressure distribution around one wheel

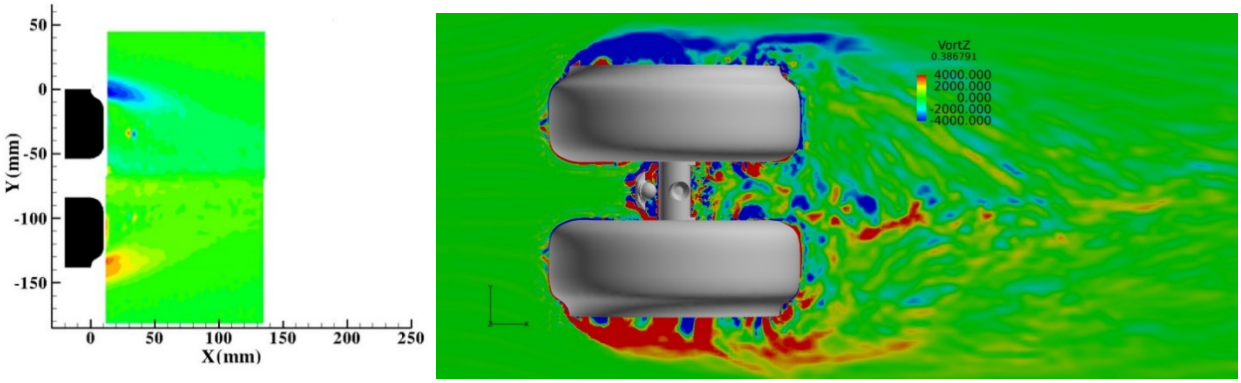


Fig.14: Spanwise vorticity contours on mid-wheel plane, left for experiment PIV data and right for our LES case *Landing-Gear-3*

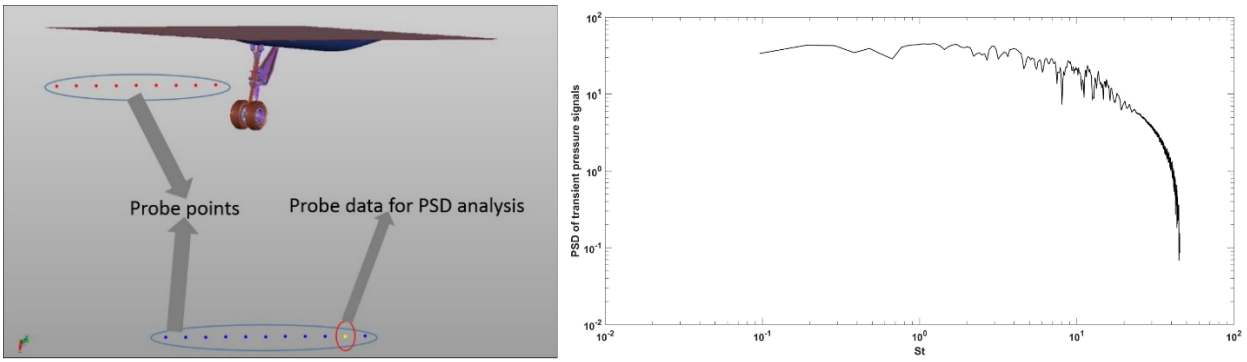


Fig.15: A farfield PSD(right) for NASA/Gulfstream Landing Gear Case at farfield proble point shown in left picture – taken directly from the simulation, not using the FW-H analogy

Computer Resources

The statistics for the present simulations are shown below in Table 5. Three simulations were performed, basic second order, fully third order and a p-adaptive second/third order. Reference Run #2 was fully 3rd order, wall-modelled and with 11.7M cells (862 M DoFs) consuming 292 Gb memory and running on all 2,736 cores of our 12kW Intel PHI cluster, equivalent to about 480 conventional cpu cores. Wall-clock 18.7 hours were needed for one flow passing period Δt_D (based on the main strut (piston) diameter). Run #3, the p-adaptive case, is very much more efficient in terms of computer resource and allows the simulation visualised in Figure 12 above to be run with nearly a billion DOFs but only on the equivalent of 160 conventional cpu cores and about 20% faster in terms of wall-clock time!

Case ID	Order of accuracy	Number of mesh cells	Number of DOFs	Speed Up Ratio	Number of nodes on cluster	Equivalent number of CPU cores	Memory comsumption (GB)	Wall-clock time for $1T_p$ (hours)
Landing -Gear-1	2^{nd}	8.1M	208.1M	80.5	1(8 PHI cards)	80	90.6	10.2
Landing -Gear-2	3^{rd}	11.76M	862.6M	34.85	8(6 PHI card each)	480	292	18.7
Landing -Gear-3	p-adaptive 2^{nd} /3^{rd}	14.65M	963.3M	144.5	4(3 PHI card each)	160	358	15.7

Table 5: Statistics for the present simulations of the NASA/Gulfstream Landing Gear Case

It is difficult to find comparative computer resource data in the open literature – and comparisons are made doubly difficult by the different mesh sizes and algorithms - but Khorrami[23] does give data for the ONERA code CEDRE [see Vuillot et al[25]]. For the BANC test case CEDRE was run as a second order solver with a 70M cell mesh on 480 conventional cpu cores and needed 1.44 hours per $\Delta t_{D.}$ and so we take this is as our reference. Our Intel PHI cluster is equivalent to 600-800 conventional cpu cores depending on loading so taking the average of 700 means the *HOTnewt* simulations scaled to 480 conventional cores would use 27.3 hours per $\Delta t_{D.}$ On the face of it *HOTnewt* is therefore much slower than CEDRE but we need to try and correct somehow for the differences between mesh size and algorithm order of accuracy.

So, 70M cells in a second order solver is equivalent to $(15/75)^3$x70M=0.55M cells in a third order solver using the PPW data from Section II.A. Scaling the computer work between meshes is partly the ratio of mesh sizes (this is simply the basic floating point work) but also the (ratio of mesh sizes)$^{1/3}$ to approximately scale the time step change assumed limited by a CFL number criterion. Hence scaling the *HOTnewt* run time for both mesh size and time step from 11.1M to 0.55M 3rd order mesh cells suggests a run time of 480/0.49 (conventional cpu cores/ hours per $\Delta t_{D.}$). This would be a factor 2.9 faster than ONERA/CEDRE.

An alternative approximate scaling is to go the other way – from 3rd order to 2nd order. The *HOTnewt* 11.1M cell 3rd order mesh is equivalent to a $(75/15)^3$x11.1M=1,423M cell 2nd order mesh. The run time for this would scale as (mesh size)$^{4/3}$ as described above implying a second order solver would consume 480/79.8 (conventional cpu cores/ hours per $\Delta t_{D.}$). This is also a factor 2.9 slower than the equivalent *HOTnewt* simulation.

This factor 2.9 run time advantage of *HOTnewt* estimated above is comparable with the factor 3.4 derived in the earlier chevron nozzle test case when comparing *HOTnewt* to a completely different second order LES code (Xia *et al*[20]). Therefore, when combined with the factor ~10 reduced energy consumption of our Intel PHI system (Jaeggi[21]) *HOTnewt* LES would appear to be around a factor ~30 cheaper than other LES methods for comparable resolution.

IV. Hierarchical Proper Orthognal Decomposition (HOP) for data-extraction and analysis

Reduced order models (ROMs) are widely used for post-processing of turbulence flow and analysis, shape optimization, flow mechanism analysis, accelerating simulations and constructing closure model for LES (see for example references[26-32]). The POD (Proper Othogonal Decomposition) method is one of the most popular ROMs, which puts solutions along different dimensions (space and time for unsteady simulations, space and shape for optimization) together and constructs snapshots as coefficients of a global orthogonal modal basis. So far, most efforts for POD analysis in publicitons focus on low Reynolds number flows, or low resolution simulations, and the scale of these problems are quite limited. The main bottleneck for large scale case is the very big size of the "global matrix" consisting of a set of snapshots of unsteady solutions, which cause both memory and speed problems for analysis of large scale simulations. For high fidelity simulations, as shown in Fig. 16, more modes are required to reconstruct the flow in POD analysis because of the resolution of small turbulence scale, which represents high frequency part of the space-time flow field. Overall, the more complex of the flow field, the higher resolution of the simulation, the size of both spatial discretization DOFs and time snapshots are larger, which requires more memory and higher computing power for POD analysis.

Generally speaking, most information of industrial interest is extracted from the low frequency part of the flowfield, the main problem is how to extract the low frequency part accurately and fast. In the same sense, one of the main purposes of high fidelity simulations is to resolve small turbulent scales flow so as to provide more accurate low frequency part of the flow field. From another perspective, all models, like RANS models, for small under-resolved turbulence scales are a sort of filter to dump aliasing error due to inadequate resolution from higher frequencies to the lower modes.

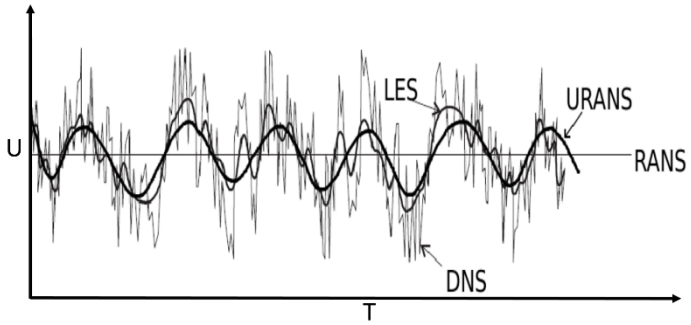


Fig.16: Examples of time signals from DNS, LES, URANS and RANS simulations at one point in the flow[33]

In this paper, an innovative hierarchical POD (HPOD) method is introduced for analysis of high order LES. For local reconstruction type high order numerical schemes, like our present method, the local in-cell solutions could be projected into orthogonal, hierarchical modal space. The different order part of the modal solution represents different scales of turbulence kinetic energy. The time evolution of POD coefficients for the low order part which contains the main flow energy, is low frequency and capable of using a very few number of energetic modes to reconstruct the dominant flow mechanisms and additionally build the connection between fine scale statistical flow quantities of interest in research and the longer scales useful for industrial design. A POD analysis for only the lower order part of the modal solutions could potentially dramatically reduce both memory consumption and computational cost. The relation between low order parts (low frequency) and high order parts (high frequency) could also be investigated by HPOD.

In our work the Karhunen-Loeve Decomposition (KLD) method is adopted under the generalised term of POD, which was first introduced by Sirovich[28] for the study of coherent structures in flows. As a method of snapshots, let $\boldsymbol{A}$ denote an $M\times N$ matrix of real data, where $M \geq N$, for analysis of LES results in this work, m is number of the DOFs for space discretisation, and n is transient time steps. Any snapshot can be expanded in terms of spatial KLD modes $\Phi_i(x)$ and temporal KLD eigenfunctions $c_i(t)$:

$$U(x,t) = U_m + \sum_{i=i}^{N_t} c_i(t)\Phi_i(x) \tag{6}$$

where $U_m(x)$ is the mean flow field, $N_t \leq N$ is the selected number of snapshots for reconstruction. To generate this decomposition, the first step is to calculate the correlation matrix.

$$\boldsymbol{B} = (1/N)(\boldsymbol{A}^T\boldsymbol{A}) \tag{7}$$

The resulting matrix is symmetric and has dimensions of $N\times N$, the eigenvalues and eigenvectors are then computed by solving the following eigenvalue problem

$$\boldsymbol{B}v_i = \lambda_i v_i, \ for\ i \in [1,N] \tag{8}$$

which has eigenvalue $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$. The matrix containing the eigenvectors are temporal KLD eigenfunctions $c_i(t)$. The snapshot (POD basis) of rank $d \leq N$ is given by

$$\Phi_i = \frac{1}{\sqrt{\lambda_i}}\sum_{j=1}^{N}(v_i)_j U_j^h, \quad for\ i \in [1,d] \tag{9}$$

It can be observed from the above that although the size of eigenvalue problem is only relevant to the number of time steps, which is commonly quite small even on modest computers, the memory requirement is still quite a challenge to store matrix $\boldsymbol{A}$ if excessive DOFs used for spatial discretization. The $j-$th orthogonal modal solution (coefficient) in $i-$th element, could be projected from nodal-based in-cell solutions as

$$u_{i,j}^m = \int_E \phi_j^m U_i dE, \quad j \in [1, N_{dofs}^m] \tag{10}$$

In order to perform HPOD analysis, the orthogonal, hierarchical modal coefficients were recorded on-the-fly during time marching process itself and calculated by Equation (10), which is exactly conservative. The reconstruction of flow variables on a nodal point $\vec{p}$ on i −th element is

$$u_i^n(\vec{p}) = \sum_j^{N_{dofs}^m} \phi_j^m(\vec{p}) u_{i,j}^m \tag{11}$$

The following Sections show applications of this to the two test cases: the chevron nozzle and the landing gear.

IV.A HPOD analysis for SMC001 transonic nozzle case.

The statistics of the HPOD analysis for different flow variables are listed in Table 6 for the transonic SMC001 chevron nozzle test case. The details and comparisons of different energy modes and reconstructed flow fields from modal snapshots are then analyzed and presented.

Job ID	Modal order	No. of variables per cell	Total DOFs	Size of instanenous results(GB)	No. of snapshots	Size of matrix ***A***(single precesion)	Actual memory consuming for HPOD(GB)	Size of modal result for each snapshot(GB)
SMC001-K0-Pressure	1st	1	7665463	1543.5	250	7.14	8.35	0.1591
SMC001-K1-Pressure	2nd	1	30661852	1543.5	250	28.56	28.37	0.6364
SMC001-K2-Pressure	3rd	1	76654630	308.71	100	28.56		1.591
SMC001-K0-Velocity	1st	3	22996389	1543.5	250	21.42	21.3	0.4773
SMC001-K1-Velocity	2st	3	91985556	1543.5	250	85.68	85.2	1.9092
SMC001-K2-Velocity	3rd	3	2.3×10^8	308.71	100	85.68		4.773

Table 6: Summary statistics of HPOD analysis for SMC001 transonic nozzle case *SMC001_2*.

Figure 17 shows the varation of the Eigenvalues with mode number (both 1st and 2nd order); it can be seen that only a small fraction of the modes contains the majority of the energy in the flow. In Figure 18 a global flowfield result, the jet centre-line time-averaged u-velocity, is reconstructed via Equation (11) and using only the *first* POD mode – the agreement with that derived from the full LES result is as expected! Exporting this single mode, derived on-the-fly from the simulation, for external use is very much more efficient than exporting the whole solution for external time-averaging. These results prove the advantages of the HPOD algorithm in two propectives: one for exacting two-level filtered, conservative, low order but contain most energy mode on-the-fly and only cost very limited memory and computing resource, which is useful for main industrial interest; another for exploring the connections between low frequency part and high frequency by not only in POD space but also in local orthogonal modal space.

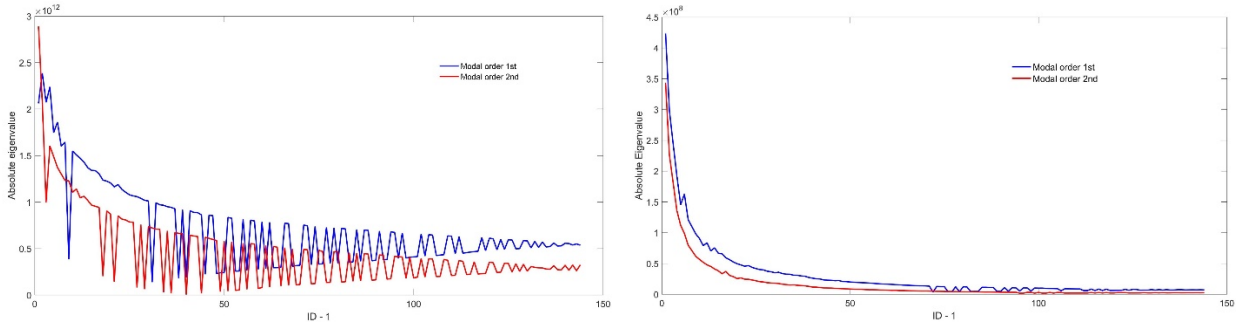


Fig.17: Plot of Eigenvalues(not including the dominant 1st mode) from the HPOD analysis: (left) for pressure; and (right) for velocity

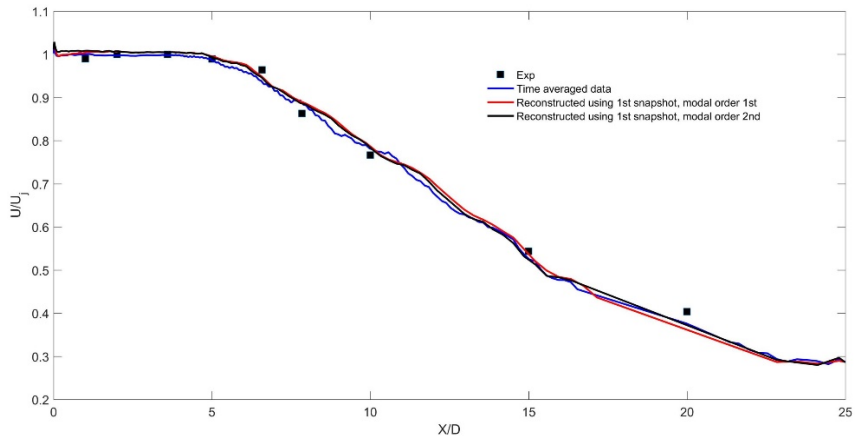
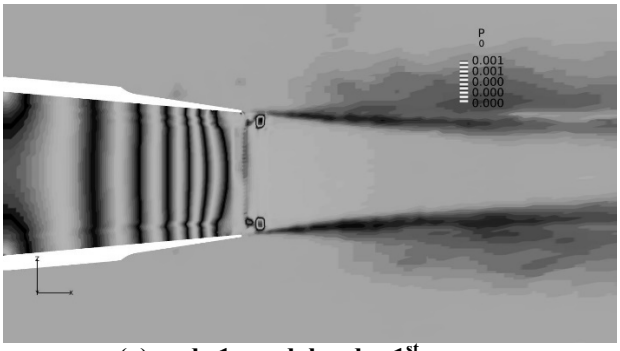
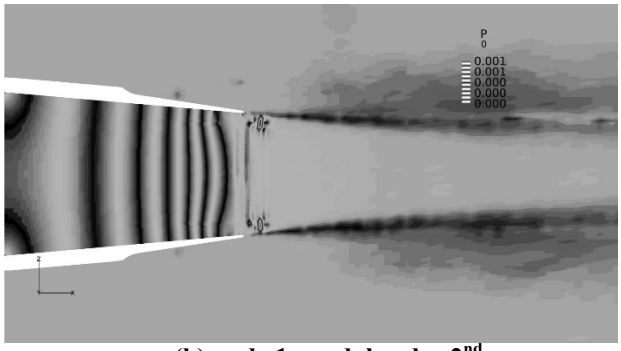


Fig.18: Comparison of reconstructed, time-averaged centreline u-velocity using the 1st POD mode

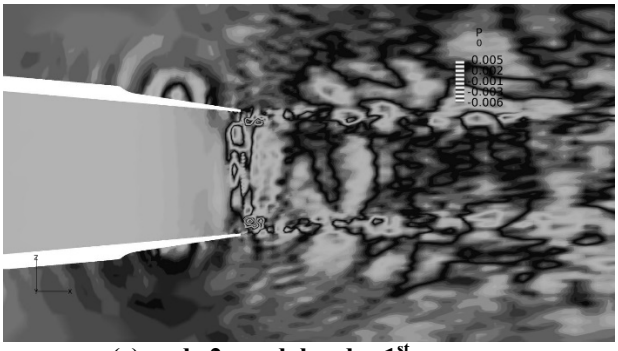
Turning now to the structure of the flowfield itself, Figure 19 shows POD modes of pressure field in the X-Z plane (Y=0) for LES case *SMC001_2* with HPOD analysis job id *SMC001-K0-Pressure* and *SMC001-K0-Pressure*. Modes higher than 21have less than ~10% of the energy(not including the first dominant mode) of the lower order modes as seen from Figure 17 above. It is fascinating to observe the fine scale structures contained within this flow captured with such efficiency with so few modes.



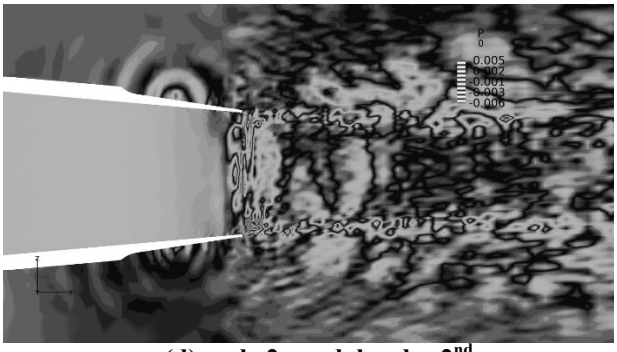
(a)mode 1, modal order 1st



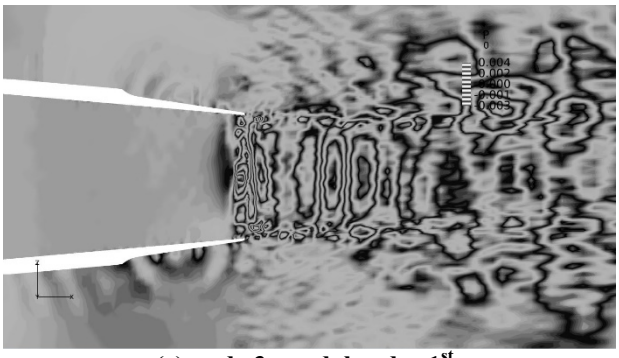
(b)mode 1, modal order 2nd



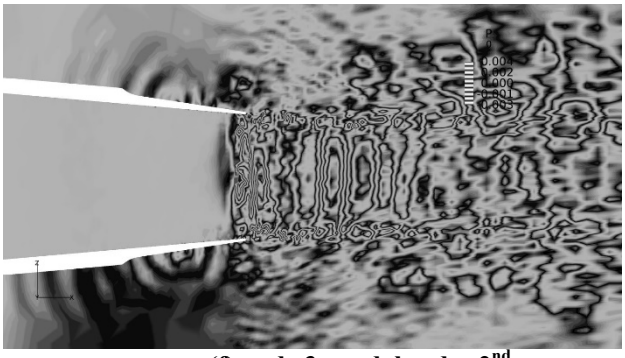
(c)mode 2, modal order 1st



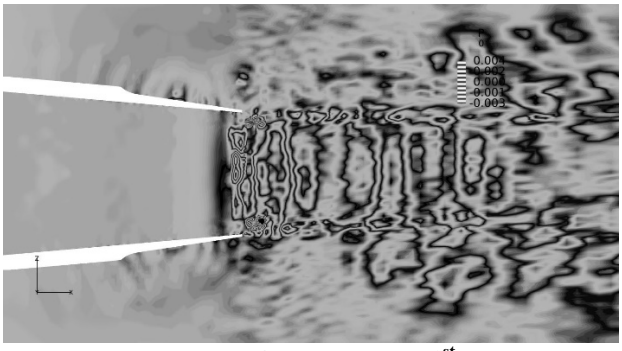
(d)mode 2, modal order 2nd



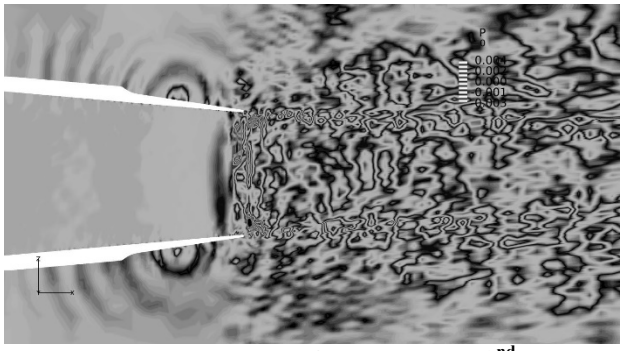
(e)mode 3, modal order 1st



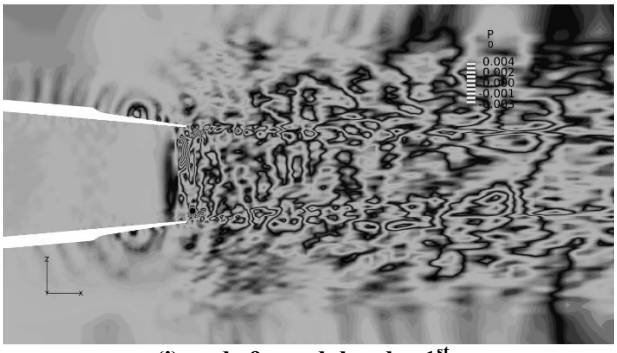
(f)mode 3, modal order 2nd



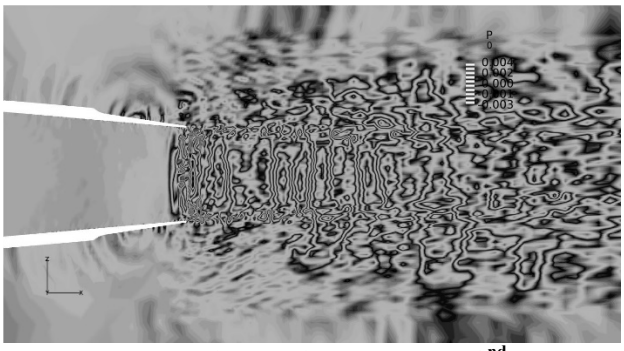
(g)mode 4, modal order 1st



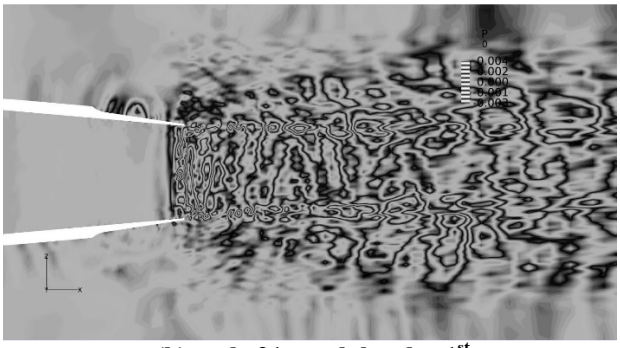
(h)mode 4, modal order 2nd



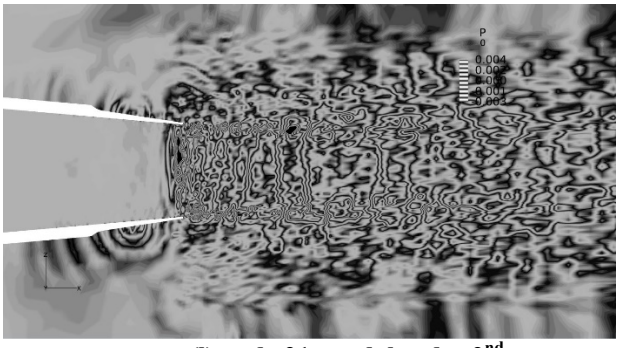
(i)mode 9, modal order 1st



(j)mode 9, modal order 2nd



(k)mode 21, modal order 1st

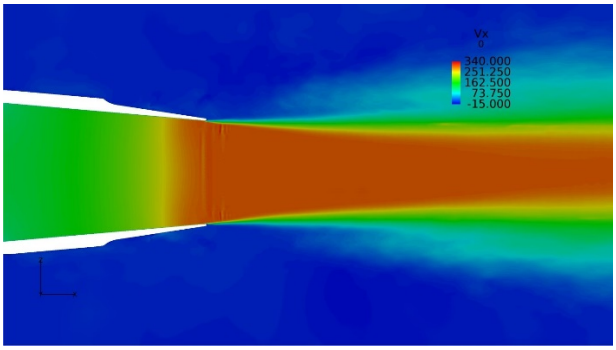


(l)mode 21, modal order 2nd

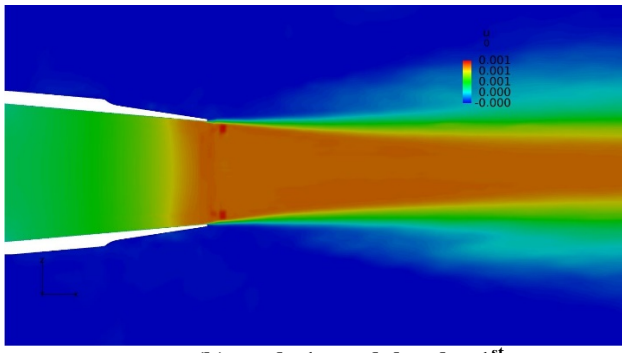
Fig.19: POD modes of pressure field in the X-Z plane (Y=0) for LES case *SMC001_2* with HPOD analysis job id *SMC001-K0-Pressure* and *SMC001-K0-Pressure*

Turning to reconstruction, Figure 20 shows *time-averaged* flow in terms of u-velocity in the X-Y plane (Z=0); (a) full time-averaged flow field; (b) the 1st POD mode of velocity x-component u; and then reconstructed using (c) mode 1 order 1 and (d) mode 1 order 2; all for case *SMC001_2* with HPOD analysis job id *SMC001-K0-Velocity* and *SMC001-K1-Velocity*. It is remarkable how closely the flow can be reconstructed with just one mode from the POD.

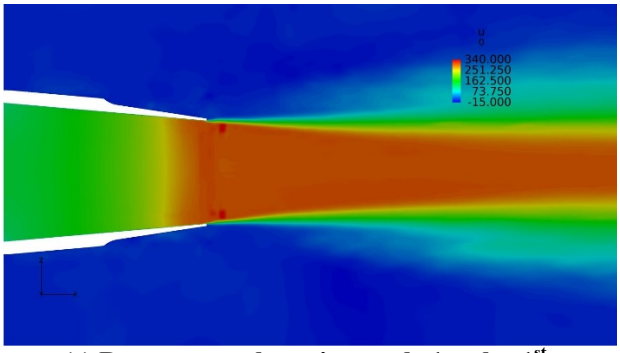
In terms of reconstructing the *instantaneous unsteady* flow Figure 21 shows the flow in the X-Y plane (Z=0) with comparisons of the reconstructed u-velocity using different numbers of snapshots at the same transient physical time, for case *SMC001_2* with HPOD analysis job id *SMC001-K0-Velocity* and *SMC001-K1-Velocity*. Again the efficiency of the POD analysis is remarkable.



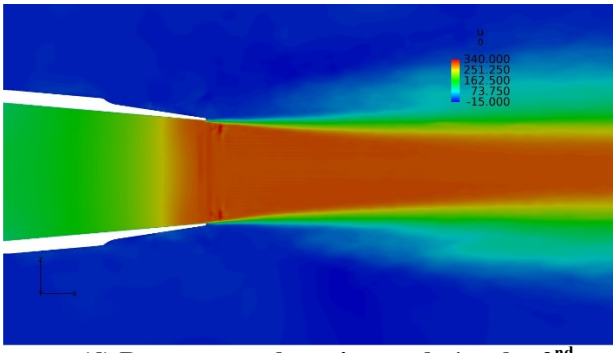
(a)Time-averaged u



(b) mode 1, modal order 1st

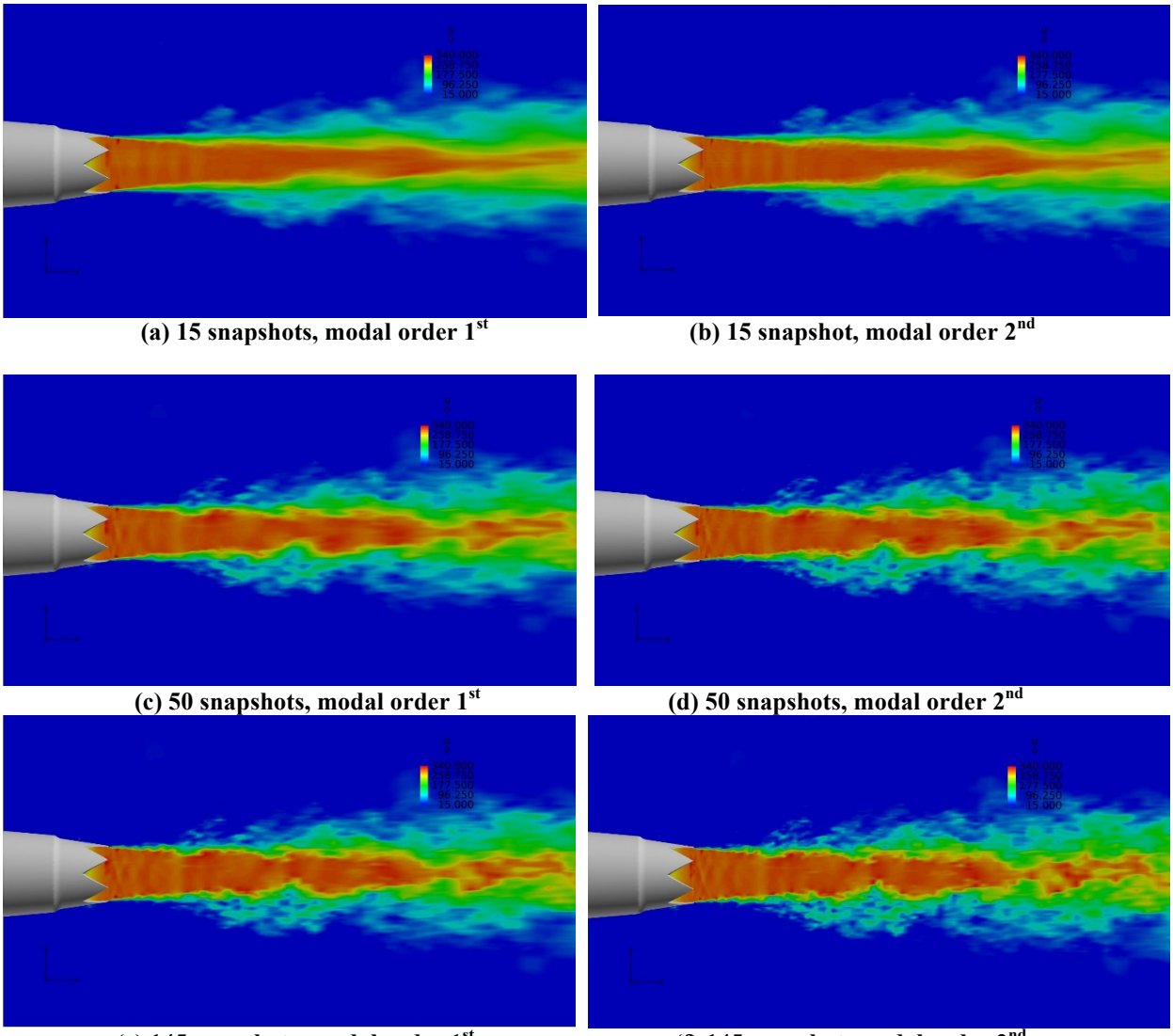


(c) Reconstructed u using mode 1 order 1st



(d) Reconstructed u using mode 1 order 2nd

Fig.20: Time-averaged flow in terms of u-velocity in the X-Y plane (Z=0); (a) full time-averaged flow field; (b) the 1st POD mode of u-velocity; and reconstructed using (c) mode 1 order 1 and (d) mode 1 order 2; all for case *SMC001_2* with HPOD analysis job id *SMC001-K0-Velocity* and *SMC001-K1-Velocity*



(a) 15 snapshots, modal order 1st

(b) 15 snapshot, modal order 2nd

(c) 50 snapshots, modal order 1st

(d) 50 snapshots, modal order 2nd

(e) 145 snapshots, modal order 1st

(f) 145 snapshot, modal order 2nd

Fig.21: Unsteady flow in the X-Y plane (Z=0): comparison of reconstructed u-velocity using different number of snapshots at the same transient physical time, for case *SMC001_2* with HPOD analysis job id *SMC001-K0-Velocity* and *SMC001-K1-Velocity*

IV.B HPOD analysis for Landing Gear case

The statistics of the HPOD analysis for different flow variables are listed in Table 7 for the Landing Gear test case. The details and comparisons of different energy modes and reconstructed flow fields from modal snapshots are then analyzed and presented.

Job ID	Modal order	No. of variables per cell	Total DOFs	Size of instanenous results(GB)	No. of snapshops	Size of matrix ***A***(single precesion)	Actual memory consuming for HPOD(GB)	Size of modal result for each snapshot(GB)
Landing-gear-K0-Pressure	1st	1	14649682	2286.86	250	13.65	20.1	0.304
Landing-gear - K1-Pressure	2nd	1	58598728	2286.86	250	54.6	59.86	1.216
Landing-gear - K2-Pressure	3rd	1	146496820	457.37	100	54.6		3.04
Landing-gear - K0-Velocity	1st	3	43949046	2286.86	250	40.95	45.8	0.912
Landing-gear - K1-Velocity	2st	3	1.76×10^8	2286.86	250	163.8	169.1	3.648
Landing-gear - K2-Velocity	3rd	3	4.4×10^8	457.37	100	163.8		9.12

Table 7: Memory consuming of HPOD analysis for Landing gear case *Landing-Gear-3*.

Figure 22 shows the variation of the Eigenvalues with mode number (both 1st and 2nd order); it can be seen that as in the earler, simpler, chevron nozzle case, only a small fraction of the modes contains the majority of the energy in the flow. Figure 23 shows POD modes in the X-Z plane (Y=0) for velocity u-component, for case *Landing-Gear-3* with HPOD analysis job id *Landing-gear-K0-Velocity* and *Landing-gear-K1-Velocity*. The range of turbulent scales in this flow are captured with relatively few modes leading to very efficient reconstruction for subsequent post-processing as shown in the next Figure. Figure 24 shows econstructed velocity u-component in the X-Z plane (Y=0): comparison using different numbers of snapshots at the same transient physical time; for case *Landing-Gear-3* with HPOD analysis job id *Landing-gear-K0-Velocity* and *Landing-gear-K1-Velocity*. It needs only around 80 modes to convincingly reproduce the fine scales of the simulation. Exporting these modes, harvested on-the-fly during the LES suggests that great efficiency in post-processing these "big data" problems is indeed achievable.

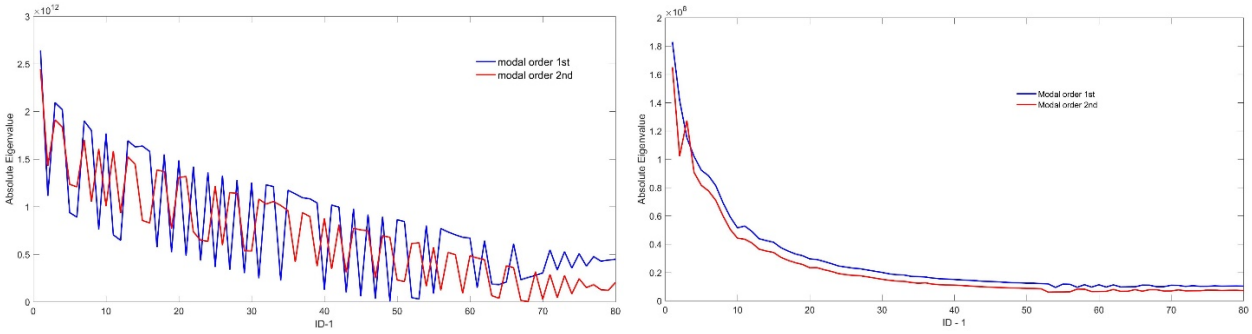
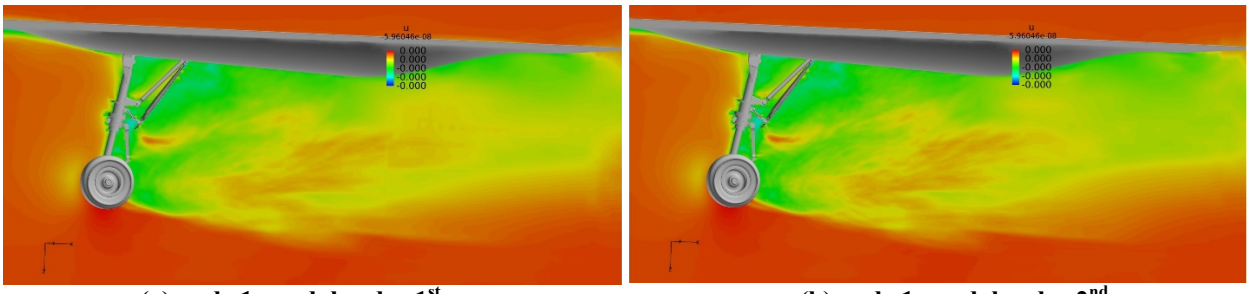
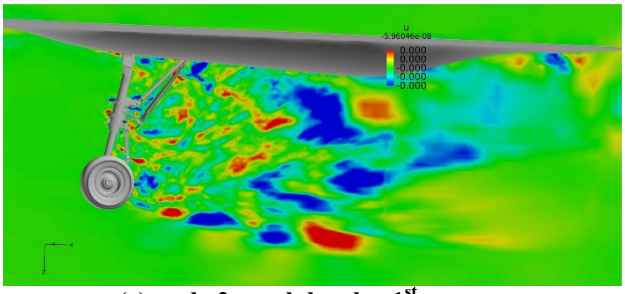


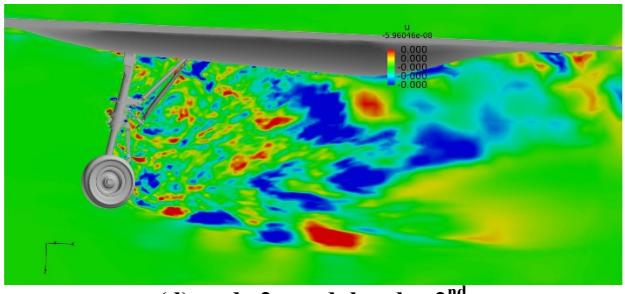
Fig.22: Plot of Eigenvalues from the HPOD analysis: (left) for pressure and (right) for velocity



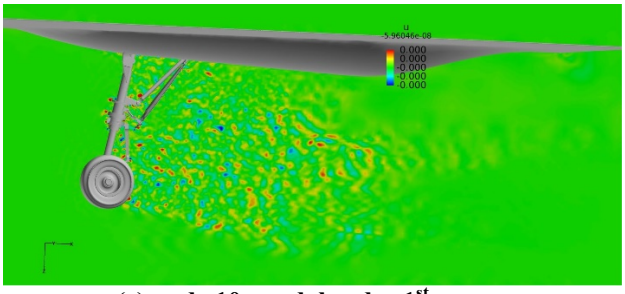
(a)mode 1, modal order 1st **(b)mode 1, modal order 2nd**



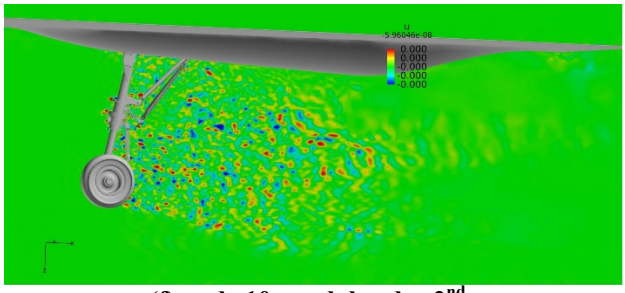
(c)mode 2, modal order 1st



(d)mode 2, modal order 2nd

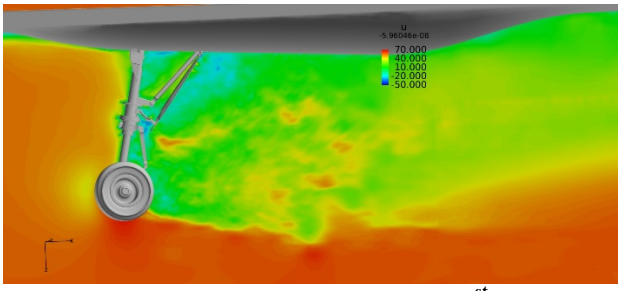


(e)mode 10, modal order 1st

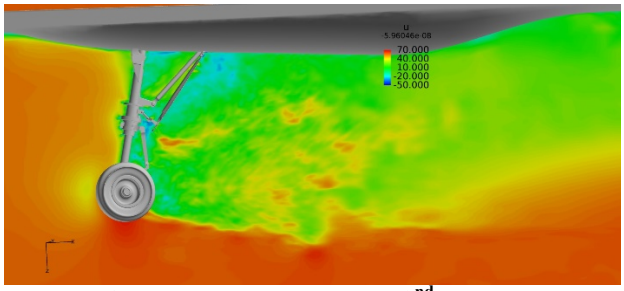


(f)mode 10, modal order 2nd

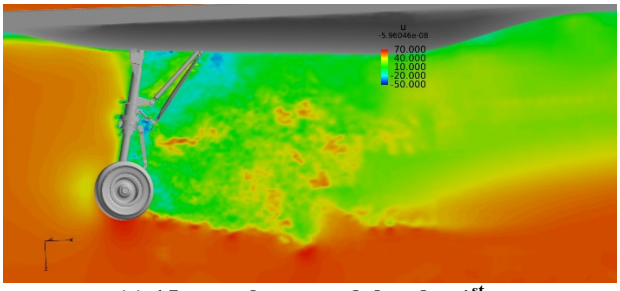
Fig.23: POD modes in the X-Z plane (Y=0) for velocity u-component, for case *Landing-Gear-3* with HPOD analysis job id *Landing-gear-K0-Velocity* and *Landing-gear-K1-Velocity*



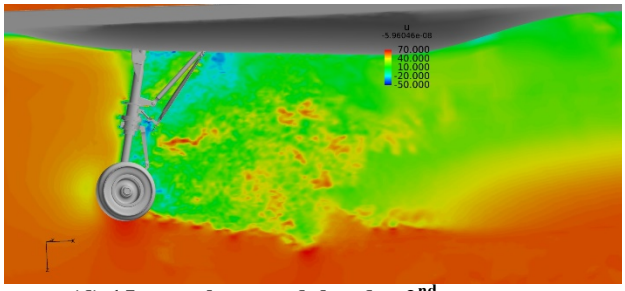
(a) 15 snapshots, modal order 1st



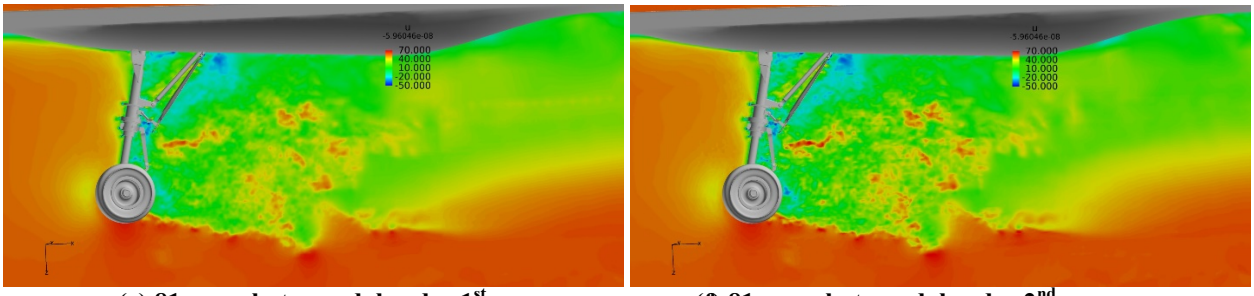
(b) 15 snapshot, modal order 2nd



(c) 15 snapshots, modal order 1st



(d) 15 snapshot, modal order 2nd



(e) 81 snapshots, modal order 1st **(f) 81 snapshot, modal order 2nd**

Fig.24: Reconstructed velocity u-component in the X-Z plane (Y=0): comparison using different numbers of snapshots at the same transient physical time; for case *Landing-Gear-3* with HPOD analysis job id *Landing-gear-K0-Velocity* and *Landing-gear-K1-Velocity*

V. Concluding Remarks

In this, perhaps over-long, paper we have attempted to demonstrate the capability of our higher order LES code *HOTnewt*. By comparisons with data we have attempted to show good matches with both experimental data and other simulations. Our goal is not really LES itself but *affordable* LES for industrial problems of realistic complexity – and we believe we have demonstrated this capability.

The real challenge is post-processing in the true sense of "big data". We have described post-processing on-the-fly and believe that in the near future that dynamic, parallel HPOD analysis will be implemented during the simulation for large-scale, complex geometry industrial problems. We demonstrated that this needs only a few of the modes to convincingly reproduce the fine scales of the simulation. Exporting these modes, harvested on-the-fly during the LES, suggests that great efficiency in post-processing these "big data" problems is indeed achievable.

Acknowledgments

The authors are grateful to Cambridge Flow Solutions Ltd (*www.cambridgeflowsolutions.com*). For permission to publish this paper. We are grateful also to Innovate UK for the SMART award which helped fund the research to build our Intel PHI cluster.

References

[1]http://www.cambridgeflowsolutions.com

[2]Y. Lu, A. Demargne, K. Liu, and W.N.Dawes, "High Fidelity, High Order, Large Eddy Simulations of a Real Geometry Aircraft Nose Landing Gear on Hybrid Unstructured Meshes & Small-scale Many-core Computing System", AIAA Paper 2016-0555, Jan 2016

[3]Y. Lu, K. Liu, and W.N.Dawes, Fast high order Large eddy simulations on many core computing systems for turbomachinery, ASME paper 2016-57468, Jun 2016

[4]Y. Lu, K. Liu, and W. N. Dawes. "Flow simulation system based on high order space-time extension of flux reconstruction method". In AIAA Science and Technology Forum and Exposition(SciTech2015), AIAA2015-0833, Jan 2015.

[5] Huynh, H. T. "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods", 18th AIAA Computational Fluid Dynamics Conference, 2007, AIAA 2007-4079

[6]Z. J. Wang and Haiyang Gao, "A unifying lifting col- location penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids," Journal of Computational Physics, Vol. 228, No. 21, 2009, pp. 8161, 8186.

[7]Y.Lu. "Local Reconstruction High Order Method and Experimental Research for Internal Flow of Turbomachinery". *phD thesis*, Tsinghua University, China.

[8]P.Vincent, P. Castonguay, and A.Jameson. "A new class of high-order energy stable flux reconstruction schemes". *Journal of Scientific Computing, Vol.47, No. 1, 2010, pp. 50,72.*

[9] C. Liang, C. Cox, and M. Plesniak. "A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction". Journal of Computational Physics, Vol.239, , 2013, pp. 244, 261.

[10] G. Gassner, M. Dumnser, F. Hindenlang, and C.D Munz. Explicit one-step time discretization for discontinuous galerkin and finite volume schemes based on local predictors. Journal of Computational Physics, Vol.230, 2011, pp. 4232, 4247.

[11]B. Owren and M Zennaro. Derivative of efficient continuous explicit runge-kutta methods. Journal of Science Computing, Vol 239, 2013, pp.138, 146.

[12]Y. Lu, K. Liu, and W. N. Dawes. "Large eddy simulations using high order flux reconstruction method on hybrid un- structuredmeshes". In AIAA Science and Technology Forum and Exposition(SciTech2014), AIAA2014-0424, Jan 2014.

[13]Y.Lu, W.N.Dawes, "High order Large Eddy Simulations for a transonic turbine blade using hybrid unstructured meshes", ASME Paper, GT2015-42283.

[14]Jameson L "Numerical errors: reliable numerical simulations" Aerospace Numerical Simulation Symposium 2001, Chofu, Japan, June 2001

[15]NASA TM-2009-215616, 2006

[16]Bridges, J & Brown, C.A., "Parametric Testing of Chevrons on Single Flow Hot Jets" NASA TM 22004-213107 also AIAA Paper 2004-2824, 2004

[17]Uzun et al. "Simulation of Noise Generation in Near-Nozzle Region of a Chevron Nozzle Jet", AIAA-2009-3596 & AIAA Journal

[18]Paliath et al. "Large Eddy Simulation for Jets from Chevron and Dual Flow Nozzles" AIAA 2011-2881 (from GE GRC)

[19]Bres et al. "Large eddy simulation of a Mach 0.9 turbulent jet" Centre for Turbulence Research, Proceedings of the Summer Programme 2014

[20]Xia H, Tucker PG & Eastwood S "Large Eddy Simulations of chevron jet flow with noise prediction" International Journal of Heat and Fluid Flow, 30, pp.1067-1079, 2009

[21]Jaeggi DM - private communication- report to Innovate UK SMART Award, 2016

[22]Shur M., Spalart P., Strelets M., and Garbaruk A., "Further Steps in LES-Based Noise Prediction for Complex Jets", *44th AIAA Aerospace Science Meeting and Exhibit, 9-12 January 2006, Reno, Nevada, Paper No. AIAA 2006-485*

[23]Khorrami MR "An overview of contributions for nose landing gear configuration – BANC-III Workshop" 21st AIAA/CEAS Aeroacoustics Conference, Dallas TX, June 22-26 2015.

[24]http://geuz.org/gmsh/

[25]Vuillot F, Houssen F, Manohsa E, Redonnet S & Jacob J "Applications of the CEDRE unstructured flow solver to landing gear unsteady flow and noise predictions" AIAA/CEAS Aeroacoustics Confrence (32nd AIAA Aeroacoustics Conference), Portland, Oregon, June 5-8, 2011

[26]D. Alonso, F. Terragni, A. Velazquez and J. M. Vega, Reduced order modeling of some fluid flows of industrial interest. Fluid Dynamics Research, Volume 44, Number 3, 2012

[27]T. Régert, P. Rambaud, M. L. Riethmuller. Investigation of the link between physics and POD modes. ADA471498, 2005

[28]L. Sirovich, Turbulence and the Dynamics of Coherent Structures, Quarterly of Applied Mathematics, 3:561–590, 1987.

[29] K. Willcox and J. Peraire. Balanced Model Reduction via the Proper Orthogonal Decomposition, AIAA JOURNAL, Vol. 40, No. 11, 2002

[30]Kevin Carlberg and Charbel Farhat, A Compact Proper Orthogonal Decomposition Basis for Optimization-Oriented Reduced-Order Models, AIAA 2008-5964, 2008

[31]T. Bui-Thanh, M. Damodaran, K. Willcox. Proper Orthogonal Decomposition Extensions For Parametric Applications in Transonic Aerodynamics. AIAA Paper 2003-4213, 2003

[32]Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper Orthogonal Decomposition Closure Models For Turbulent Flows: A Numerical Comparison. Computer Methods in Applied Mechanics and Engineering, Volumes 237–240, Pages 10–26, 2012

[33]K. Hanjalic, Turbulence and Transport Phenomena: Modelling and Simulation. , 2005