High-fidelity simulations of transonic buffet on wide-span airfoils in the OpenSBLI automatic code-generation framework on GPUs

David J. Lusher*, Andrea Sansica, Markus Zauner, Atsushi Hashimoto (JAXA)

JAPAN SOCIETY FOR THE PROMOTION OF SCIENCE
日本学術振興会

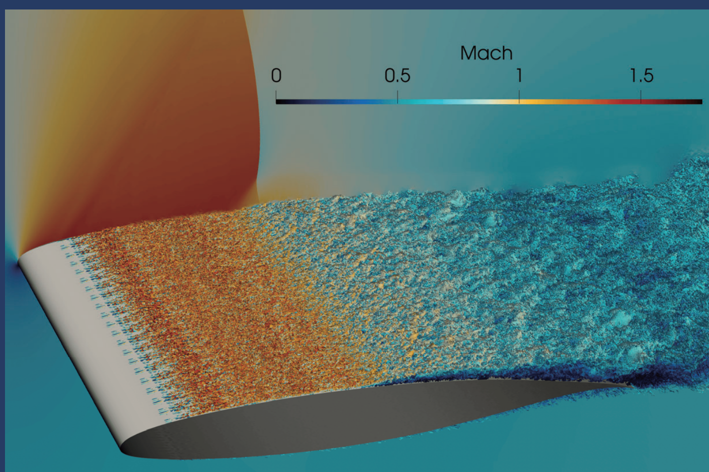
ANSS 55 Grand Numerical Prize Award Lecture - JAXA Chofu Aerospace Centre - November 21st 2023
第55回流体力学講演会／第41回航空宇宙数値シミュレーション技術シンポジウム 最優秀賞
*JSPS Postdoctoral Research Fellow & JSPS KAKENHI Grant: 22F2205

JAXA

Acknowledgements

The authors would like to thank the following people for their support:

- 溝渕さん, 橋本さん, 中北さん and other LC Hub members.
- 中村さん, 上野さん, 松山さん, and all ANSS organising committee.
- JSS3 management and support team staff.



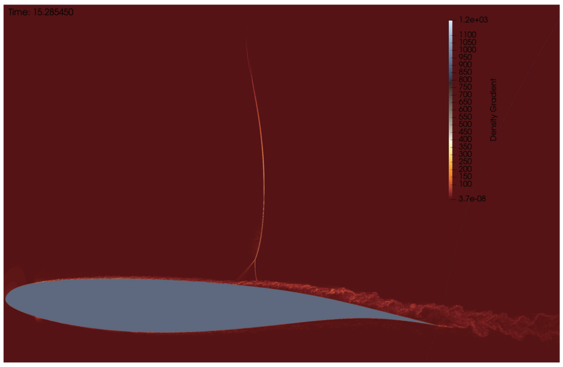
Wide-span turbulent transonic buffet simulation in OpenSBLI on the JAXA JSS3 GPU nodes

JAPAN SOCIETY FOR THE PROMOTION OF SCIENCE
日本学術振興会

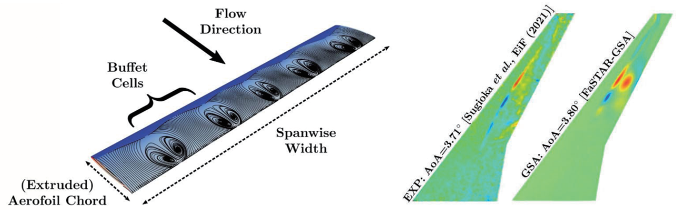
JAXA

High-fidelity transonic buffet instabilities

- Transonic buffet is an aerodynamic instability that limits the flight envelope of aircraft.
- Depending on configuration, known to consist of:
  - a) A 2D **shock-oscillation** mode.
  - b) A 3D 'buffet cell' **span wise modulation** mode.
- Most computational studies are limited to **low-fidelity (RANS)-based methods.**
- High-fidelity (ILES/DNS) are very expensive, typically limited to narrow domains (Span ~ 5% chord length ~ 0.05 aspect ratio).
- Therefore, **only suitable for studying the 2D shock oscillations.**
- In this work, we apply ILES to **wider spans up to AR=3**, to search for 3D effects.



a) 2D shock oscillation instability: OpenSBLI multi-block simulation of transonic buffet on infinite (span-periodic) OAT15A wing segment, on 100 V100 GPUs on the JSS3 supercomputer.



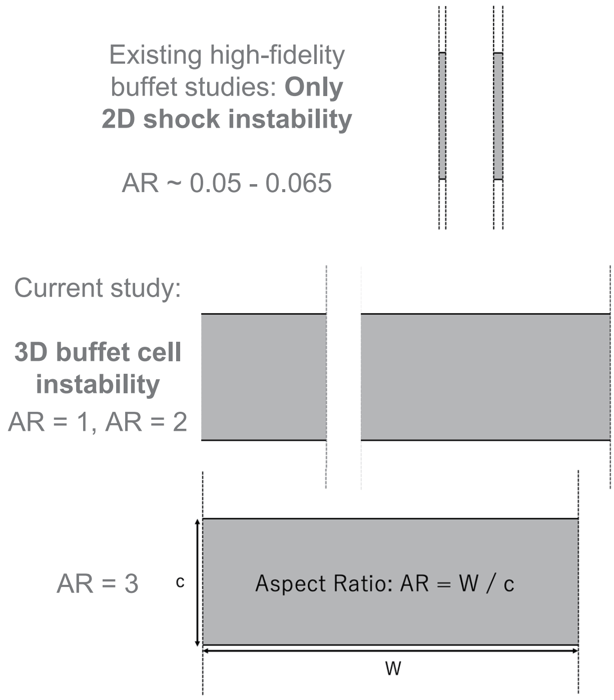
b) 3D buffet cell instability: [CFD: **Sansica A., Hashimoto A. AIAA-J, 61(10) 2023**]; EXP: **Koike et al,** AIAA Aviation, 2016; **Sugioka et al**, Exp. In Fluids, 2018]

3

High-fidelity transonic buffet instabilities

- Transonic buffet is an aerodynamic instability that limits the flight envelope of aircraft.
- Depending on configuration, known to consist of:
  - a) A 2D **shock-oscillation** mode.
  - b) A 3D 'buffet cell' **span wise modulation** mode.
- Most computational studies are limited to **low-fidelity (RANS)-based methods.**
- High-fidelity (ILES/DNS) are very expensive, typically limited to narrow domains (Span ~ 5% chord length ~ 0.05 aspect ratio).
- Therefore, **only suitable for studying the 2D shock oscillations.**
- In this work, we apply ILES to **wider spans up to AR=3**, to search for 3D effects.

Extruded wings comparison

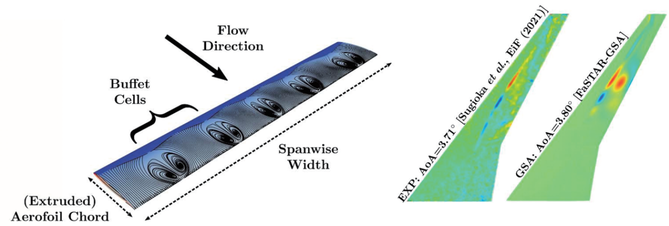


4

Open questions and previous studies

3D ‘buffet cell’ span wise modulation

- Previous studies:
  - 3D instability is superimposed over the 2D shock-oscillation.
  - **3D instability is dominant** for realistic configurations (finite-wings).
  - Buffet cells convect with sweep angle.
  - Q: Can we identify **3D effects in the absence of sweep (+ infinite wing)**?
  - **Lack of high-fidelity literature** for 3D buffet.
  - Buffet cells / stall cells share similar features - Are they the **same phenomena** at different flow conditions?



3D buffet cell instability: [CFD: **Sansica A., Hashimoto A. AIAA-J, 61(10) 2023**]; EXP: **Koike et al,** AIAA Aviation, 2016; **Sugioka et al**, Exp. In Fluids, 2018]

Iovnovich and Raveh, AIAA-J (2015)

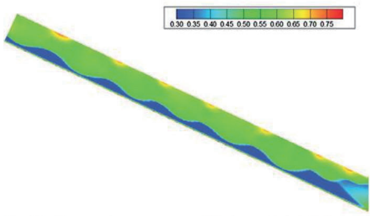


Fig. 26 Upper-surface-pressure coefficient snapshot: infinite-swept configuration, $b/c = 12$, and $\Lambda = 25$ deg nominal buffet conditions.

Plante et al, JFM (2021)

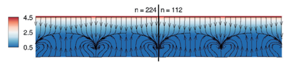
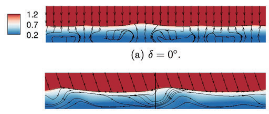


Figure 4.3 – Surface pressure coefficient and skin friction lines of 3-D steady solution with 224 (left) and 112 (right) spanwise grid cells (NACA4412, $M = 0.2$, $Re = 350\,000$, $\alpha = 15°$, $\delta = 0°$, $L_z = 6$).



(a) $\delta = 0°$.

(b) $\delta = 30°$ (right 112 spanwise grid cells, left 168 spanwise grid cells).

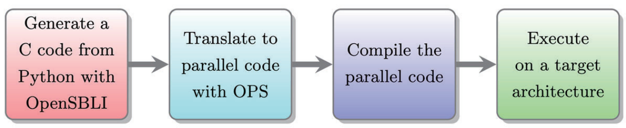
Figure 4.7 – Instantaneous surface pressure coefficient and skin friction lines for URANS simulations of infinite swept wings with two sweep angles (OALT25, $M = 0.7352$, $Re = 3 \times 10^6$, $\alpha = 4°$, $L_z = 6$).

OpenSBLI Python-based code-generation

High-level problem specification via symbolic algebra

- OpenSBLI is a Python-based **code-generation** framework for compressible CFD using finite-differences [1].
- **Users specify the equations to solve** and simulation options in Python.
- **Symbolic algebra (SymPy)** is converted into a complete ILES/DNS CFD solver in **C/C++ code**.
- The OPS DSL enables parallel execution on many platforms (MPI/OpenMP/CUDA/…).
- Developed **2016-2021 (University of Southampton)**, during my PhD [1].
- New development at **JAXA, JSPS project (2022-2023) - next release & paper**.

```
ndim = 3
sc1 = "**{\'scheme\':\'Teno\'}"
# Define the compresible Navier-Stokes equations in Einstein notation.
mass = "Eq(Der(rho,t), - Conservative(rhou_j,x_j,%s))" % sc1
momentum = "Eq(Der(rhou_i,t) , -Conservative(rhou_i*u_j + KD(_i,_j)*p,x_j , %s) + Der(tau_i_j,x_j) )" % sc1
energy = "Eq(Der(rhoE,t), - Conservative((p+rhoE)*u_j,x_j, %s) - Der(q_j,x_j) + Der(u_i*tau_i_j ,x_j) )" % sc1
stress_tensor = "Eq(tau_i_j, (mu/Re)*(Der(u_i,x_j)+ Der(u_j,x_i) - (2/3)* KD(_i,_j)* Der(u_k,x_k)))"
heat_flux = "Eq(q_j, (-mu/((gama-1)*Minf*Minf*Pr*Re))*Der(T,x_j))"
# Numerical scheme selection
Avg = RoeAverage([0, 1])
LLF = LLFTeno(teno_order, averaging=Avg)
cent = Central(4)
rk = RungeKuttaLS(3, formulation='SSP')
# Specifying boundary conditions
boundaries[direction][side] = IsothermalWallBC(direction, 0, wall_eqns)
# Generate a C code
alg = TraditionalAlgorithmRK(block)
OPSC(alg)
```



[1] D.J. Lusher, S.P. Jammy, N.D. Sandham. OpenSBLI: Automated code-generation for heterogeneous computing architectures applied to compressible fluid dynamics on structured grids. **Computer Physics Communications** (2021).

Oxford Parallel Structured (OPS) DSL

OP-DSL

Automatic parallelisation for multi-block structured mesh applications

- OpenSBLI/OPS collaboration.
- The OPS library uses **source-to-source code translation** for structured mesh applications.
- OpenSBLI code written in the OPS API can be **automatically parallelised.**
- Supports C/C++/Fortran, to generate: MPI/OpenMP/CUDA/OpenCL/OpenACC/HIP
  .
- **JSS3 TOKI-RURI & NASA Cluster** (Nvidia GPUs) -> CUDA+MPI+GPUDIRECT.
- JSS3 **TOKI-SORA & Fugaku** (A64FX CPUs) -> Hybrid MPI+OpenMP.

G. Mudalige, I. Reguly, M.Giles.

https://op-dsl.github.io/

I. Z. Reguly, G. R. Mudalige and M. B. Giles, Loop Tiling in Large-Scale Stencil Codes at Run-Time with OPS, in IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 4, pp. 873-886, 1 April 2018, doi: 10.1109/TPDS.2017.2778161.

Parallel loop structure

```
int range64[] = {-3, b0np0 + 4, -3, b0np1 + 4, -3, b0np2 + 4};
ops_par_loop(opensbliblock00Kernel064,
"Constituent Relations evaluation", opensbliblock00, 3, range64,
ops_arg_dat(Et_B0, 1, stencil_0_00, "double", OPS_READ),
ops_arg_dat(rho_B0, 1, stencil_0_00, "double", OPS_READ),
ops_arg_dat(u0_B0, 1, stencil_0_00, "double", OPS_READ),
ops_arg_dat(u1_B0, 1, stencil_0_00, "double", OPS_READ),
ops_arg_dat(u2_B0, 1, stencil_0_00, "double", OPS_READ),
ops_arg_dat(a_B0, 1, stencil_0_00, "double", OPS_WRITE),
ops_arg_dat(p_B0, 1, stencil_0_00, "double", OPS_RW));
```

Body of the function being called

```
void opensbliblock00Kernel064(const ACC<double> &Et_B0, const ACC<double> &rho_B0,
 const ACC<double> &u0_B0, const ACC<double> &u1_B0, const ACC<double> &u2_B0,
 ACC<double> &a_B0, ACC<double> &p_B0)
{
   double inv_rho = 0.0;
   inv_rho = 1.0/rho_B0(0,0,0);

    p_B0(0,0,0) = (gama - 1)*(Et_B0(0,0,0) - 0.5*pow(u0_B0(0,0,0), 2)
    - 0.5*pow(u1_B0(0,0,0), 2) - 0.5*pow(u2_B0(0,0,0),2))*rho_B0(0,0,0);

   a_B0(0,0,0) = sqrt(gama*inv_rho*p_B0(0,0,0));
}
```

Automatically generated CUDA GPU code

```
__global__ void ops_opensbliblock00Kernel064(
double* __restrict arg0,
double* __restrict arg1,
double* __restrict arg2,
double* __restrict arg3,
double* __restrict arg4,
double* __restrict arg5,
double* __restrict arg6,
int size0,
int size1,
int size2 ){


  int idx_z = blockDim.z * blockIdx.z + threadIdx.z;
  int idx_y = blockDim.y * blockIdx.y + threadIdx.y;
  int idx_x = blockDim.x * blockIdx.x + threadIdx.x;

  arg0 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[0][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[0][0] * dims_opensbliblock00Kernel064[0][1];
  arg1 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[1][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[1][0] * dims_opensbliblock00Kernel064[1][1];
  arg2 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[2][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[2][0] * dims_opensbliblock00Kernel064[2][1];
  arg3 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[3][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[3][0] * dims_opensbliblock00Kernel064[3][1];
  arg4 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[4][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[4][0] * dims_opensbliblock00Kernel064[4][1];
  arg5 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[5][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[5][0] * dims_opensbliblock00Kernel064[5][1];
  arg6 += idx_x * 1*1 + idx_y * 1*1 * dims_opensbliblock00Kernel064[6][0] + idx_z * 1*1 * dims_opensbliblock00Kernel064[6][0] * dims_opensbliblock00Kernel064[6][1];

  if (idx_x < size0 && idx_y < size1 && idx_z < size2) {
    const ACC<double> argp0(dims_opensbliblock00Kernel064[0][0], dims_opensbliblock00Kernel064[0][1], arg0);
    const ACC<double> argp1(dims_opensbliblock00Kernel064[1][0], dims_opensbliblock00Kernel064[1][1], arg1);
    const ACC<double> argp2(dims_opensbliblock00Kernel064[2][0], dims_opensbliblock00Kernel064[2][1], arg2);
    const ACC<double> argp3(dims_opensbliblock00Kernel064[3][0], dims_opensbliblock00Kernel064[3][1], arg3);
    const ACC<double> argp4(dims_opensbliblock00Kernel064[4][0], dims_opensbliblock00Kernel064[4][1], arg4);
    ACC<double> argp5(dims_opensbliblock00Kernel064[5][0], dims_opensbliblock00Kernel064[5][1], arg5);
    ACC<double> argp6(dims_opensbliblock00Kernel064[6][0], dims_opensbliblock00Kernel064[6][1], arg6);
    opensbliblock00Kernel064_gpu(argp0, argp1, argp2, argp3,
                  argp4, argp5, argp6);
```

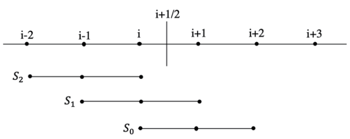
Summary of OpenSBLI numerical methods

Central + WENO shock-capturing

- **Non-dissipative (central) schemes** for high-resolution of turbulence and wave propagation.
- Convective terms written in cubic-split form to **improve numerical stability.**
- 5th order WENO-based shock-capturing is **applied only at shockwaves.**
- Dispersion-Relation-Preserving (DRP) filters are used in the freestream.
- Explicit 4th order low-storage Runge-Kutta time-stepping.

$$\frac{\partial \rho u \varphi}{\partial x} = \alpha \frac{\partial \rho u \varphi}{\partial x} + \beta \left(u \frac{\partial \rho \varphi}{\partial x} + \rho \frac{\partial u \varphi}{\partial x} + \varphi \frac{\partial \rho u}{\partial x} \right)$$
$$+ (1 - \alpha - 2\beta) \left(\rho u \frac{\partial \varphi}{\partial x} + \rho \varphi \frac{\partial u}{\partial x} + u \varphi \frac{\partial \rho}{\partial x} \right)$$

Cubic split-form of convective derivative operators for the compressible Naiver-Stokes equations [1]



Staggered 5th order WENO shock-capturing stencil

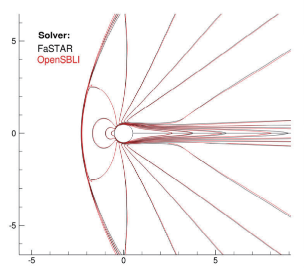
$$U^{n+1} = \widehat{U}^{n+1} - \Delta t L_f (F)_{i,j,k}$$
$$L_f (F)_{i,j,k} = \frac{1}{\Delta x} \left[\tilde{F}^*_{j+1/2,k} - \tilde{F}^*_{j-1/2,k} \right] + \ldots$$

Application of non-linear shock-capturing at the end of a full time-step [2]

Cross-validation of OpenSBLI to the FaSTAR code for supersonic cylinder flows.

Sansica, A, Lusher, D., Hashimoto, A. Mach Evolution of the Cylinder Wake Flow Bifurcations.

The 34th International Symposium on Shockwaves (ISSW34, South Korea 2023)



[1] G. Coppola et al. Journal of Computational Physics (2019).

[2] H. Yee et al. Computers & Fluids (2018).

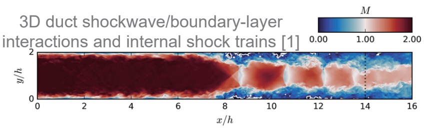
8

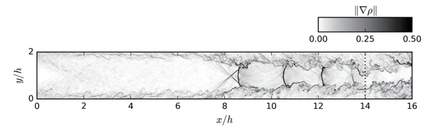
OpenSBLI: Applications

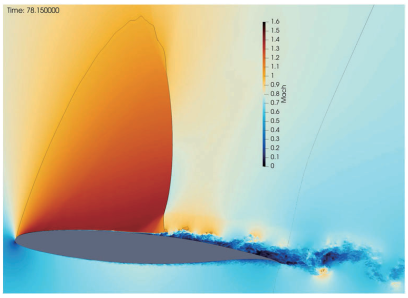
Compressible turbulence and shockwave/boundary-layer interactions

- Previous applications include **compressible turbulence[2]**, 3D shockwave/boundary-layer interactions, shock-trains.
- Now extended to **multi-block for aerofoil and cylinder problems** (bottom, right).
- Successful **Fugaku FY2023 Junior Researcher project** completed.

3D duct shockwave/boundary-layer interactions and internal shock trains [1]

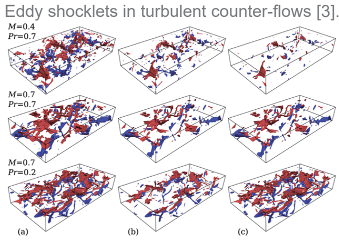




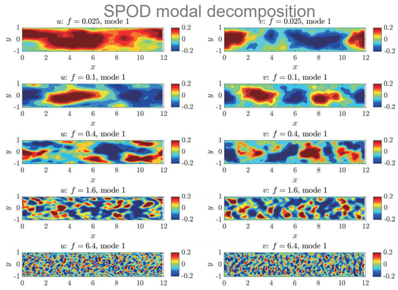


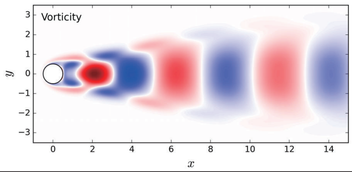
Turbulent transonic shock buffet (NASA CRM)

Eddy shocklets in turbulent counter-flows [3].



SPOD modal decomposition





[1] A. Gillespie, N.D. Sandham. Shock Train Response to High-Frequency Backpressure Forcing. AIAA Journal 60 (6), 3736-3748

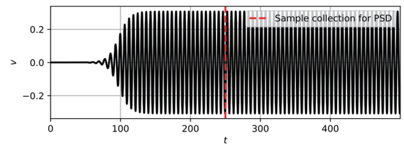
[2] D.J. Lusher, N.D. Sandham. Assessment of Low-Dissipative Shock-Capturing Schemes for the Compressible Taylor-Green Vortex. AIAA Journal (2021).

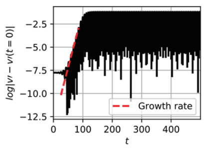
[3] A. Hamzehloo, D.J. Lusher, S. Laizet, N.D. Sandham. Direct numerical simulation of compressible turbulence in a counter-flow channel configuration. Physical Review Fluids 6 (9), 094603

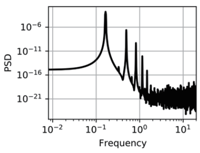
OpenSBLI: Recent validations

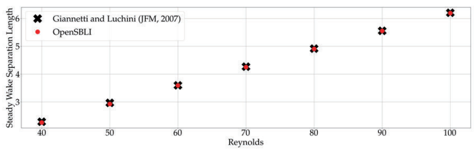
Cylinder and Airfoil validations in new OpenSBLI release

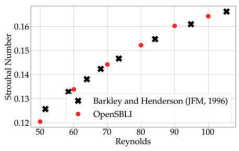
- Low Reynolds number (Re=100) validations against literature [1,2].
- Circular cylinder flows with Selective Frequency Damping (SFD).
- SFD demonstration with code and mesh in the release.

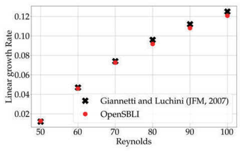


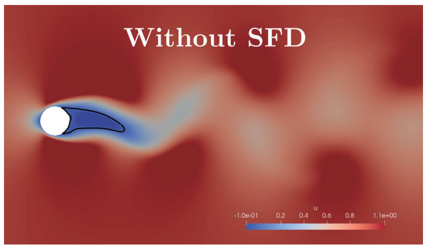


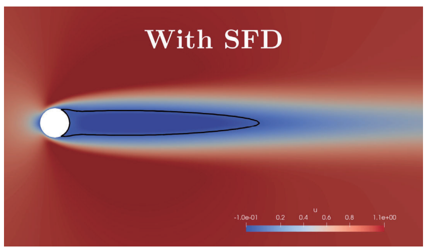












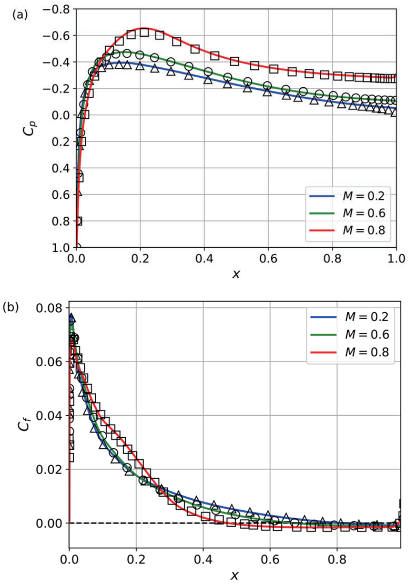
[1] F. Giannetti, et al. Journal of Fluid Mechanics 581 (2007).

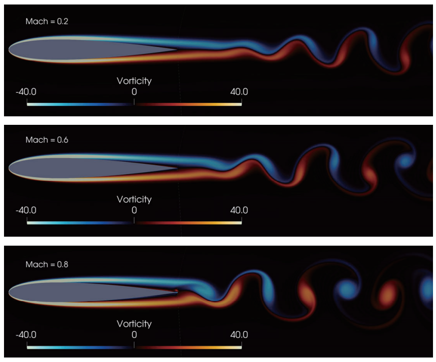
[2] D. Barkley, et al, Journal of Fluid Mechanics (1996).

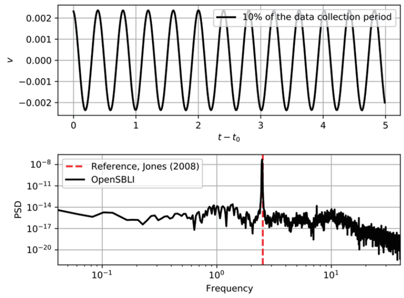
OpenSBLI: Recent validations

Cylinder and Airfoil validations in new OpenSBLI release

- Low Reynolds number (Re=10,000) validations against DNS data [1,2].
- NACA0012 airfoil.
- Open geometry can be contained in the code release.
- Airfoil demonstration with code and mesh







[1] L. E. Jones, Ph.D. thesis, University of Southampton (2008).

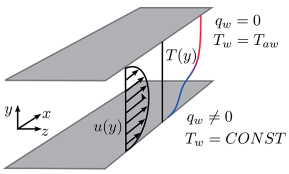
[2] L. E. Jones, Journal of Fluid Mechanics (2008).

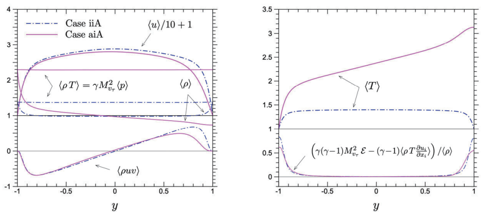
11

OpenSBLI: Large-scale HPC on JAXA JSS3 GPUs

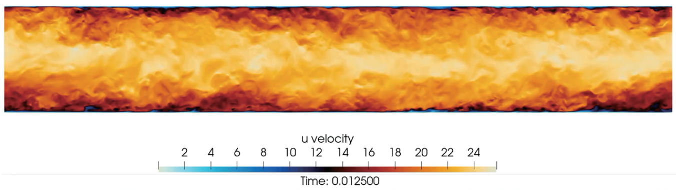
Using JSS3/Fugaku to generate high-quality DNS databases to improve RANS models at JAXA

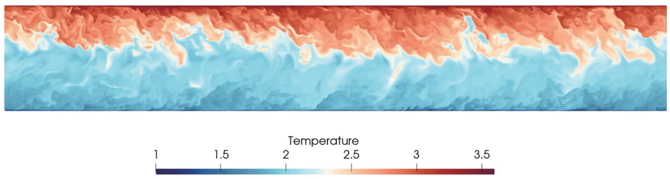
- DNS of supersonic turbulence, **mixed isothermal/adiabatic** wall conditions within the same problem configuration [1].
- (JAXA): Dr. David J. Lusher, Dr. Andrea Sansica, Dr. Atsushi Hashimoto, Dr. Hiroyuki Abe.
- (NASA): Dr. Gary Coleman, (Boeing/Retired) Dr. Philippe Spalart.
- Generation of high-quality **(thermally balanced)** DNS databases of turbulent quantities.
- OpenSBLI DNS on JSS3 GPUs allows us to:
  - Validate lower-fidelity methods and boundary-conditions used in FaSTAR.
  - Improved understanding of terms such as the turbulent Prandtl number.
- Applied for **FY2024 Fugaku Junior Researcher Project.**





OpenSBLI mixed thermal wall conditions DNS: aiF2s (2201x787x1237)





[1] DJ Lusher, GN Coleman. Numerical Study of Compressible Wall-Bounded Turbulence–the Effect of Thermal Wall Conditions on the Turbulent Prandtl Number in the Low-Supersonic Regime. **International Journal of Computational Fluid Dynamics**, 1-19, (2023).

[2] D.J. Lusher, G.N. Coleman. Numerical Study of the Turbulent Prandtl Number in Supersonic Plane-Channel Flow – the Effect of Thermal Boundary Conditions. **NASA Technical Memorandum,** 10483 (2022).

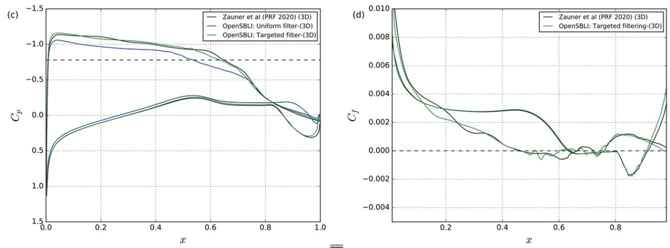
12

Computational setup

Multi-block mesh and flow parameters

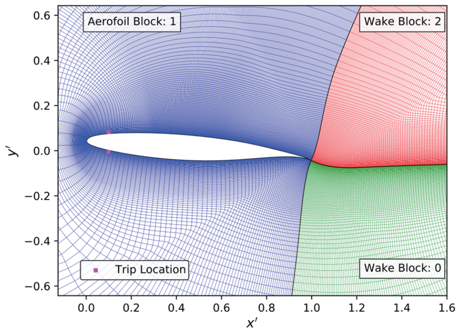
- OpenSBLI was **validated for buffet** against literature for **Dassault Aviation V2C airfoil** [1].
- Using 3-block structured C-Mesh (Nx, Ny) plane points: (701,681), (2249,681), (701,681).
- Uniformly extruded in the span: 50-75 points per 0.05c width ~ **2.5 - 7.5 billion mesh points**.
- Narrow (0.05c) cases are extruded to generate **restart conditions for wide-span** (1c - 2c) cases.
- Flow conditions:
  - Moderate Reynolds number: 500,000.
  - Baseline Freestream Mach: 0.72.
  - AoA: 4 to 8 degrees.
  - **Zero sweep angle:** Q: Can we observe 3D effects without any imposed cross-flow?
  - **Turbulent buffet**: wall tripped at 0.1c chord.

$$\rho u_\eta\big|_{\eta=0} = \sum_{i=1}^{3} A \exp\left(-\frac{(x-x_t)^2}{2\sigma^2}\right)\sin\left(\frac{k_i z}{0.05c}\right)\sin\left(\omega_i t + \Phi_i\right)$$



	C_L / C_D	$\overline{\overline{C_L}}$	C_D	$C_{D,f}$	$C_{D,p}$
Zauner et al [46]	20.97	0.76	0.036	0.0037	0.032
OpenSBLI (Uniform filter)	17.66 (-15.8%)	0.61 (-16.0%)	0.034 (-5.5%)	0.0034 (-8.1%)	0.031 (-3.1%)
OpenSBLI (Targeted filter)	21.36 (+1.8%)	0.76 (± 0.0%)	0.035 (-2.8%)	0.0037 (± 0.0%)	0.032 (± 0.0%)

[1] Lusher D., et al. Automatic Code-Generation to Enable High-Fidelity Simulations of Multi-Block Airfoils on GPUs. AIAA SciTech 2023.



Example NASA-CRM segment - 2D mesh configuration with 3 blocks.

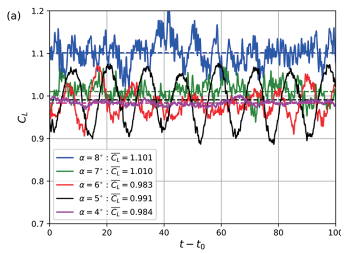
(Plotting every 7th line for clarity)

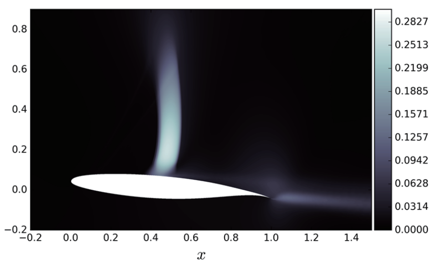
13

Narrow Domain: AoA Study

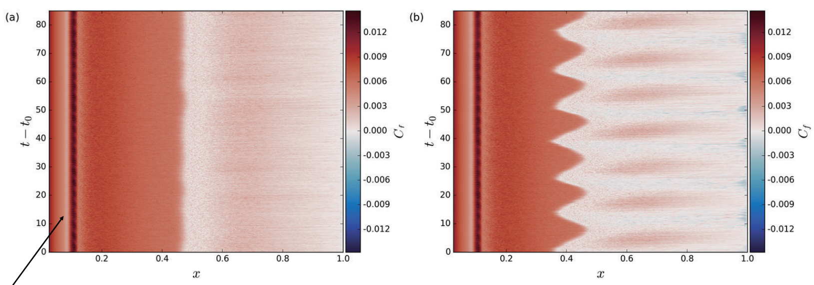
Time histories of surface skin-friction

- 2D (shock oscillation) buffet **onset conditions** are investigated with 3D simulations at AR=0.05.
- Range of: 4-8 degrees AoA.
- Time history (x-t) plot of surface **skin-friction**.
- AoA = 4 (pre-onset): **fixed shock location.**
- AoA = 5 (buffet onset): **shock oscillates** on the suction side





Time history of aerodynamic lift coefficient and RMS of pressure.



Trip location

AoA = 4 (pre-onset)

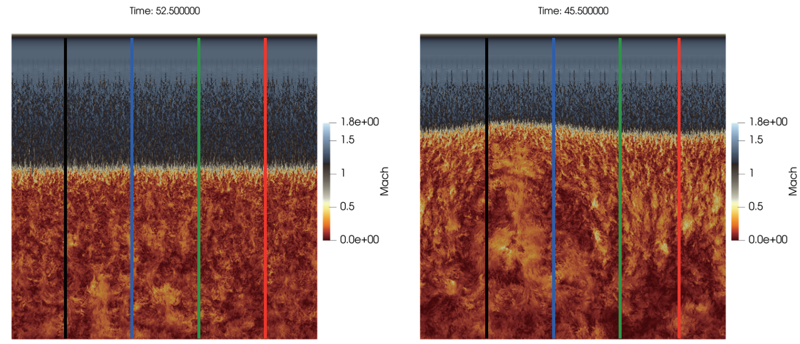
AoA = 5 (buffet post-onset)

x-t diagram of surface skin friction

Extrusion to wide-span: AR1

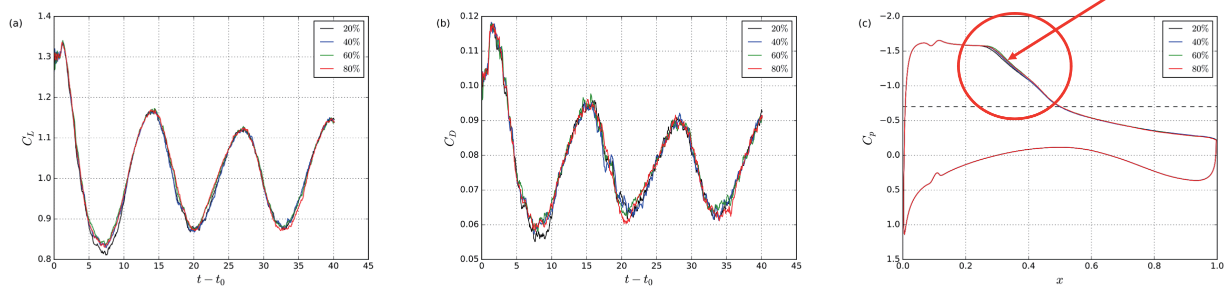
Wide-span buffet effects no longer purely 2D

- AR=1 case, M=0.72, **AoA: 6 degrees**.
- **2.5 billion mesh points at AR=1**
- At AR=1: the flow is no longer strictly 2D across the span.
- Large **3D separation bubbles** occur, during low-lift phases.
- Sectional plots across the span (bottom) show the **lack of span homogeneity**.
- Seem to be linked to the **point of maximum flow separation** (low-lift).



Near surface: High-lift buffet phase (2D)

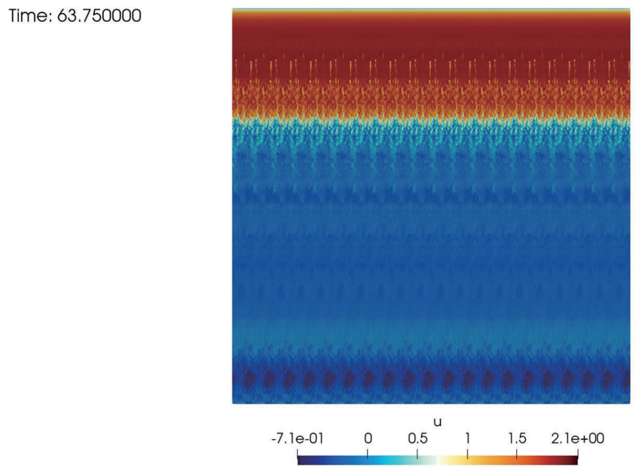
Near surface: Low-lift buffet phase (3D effects)

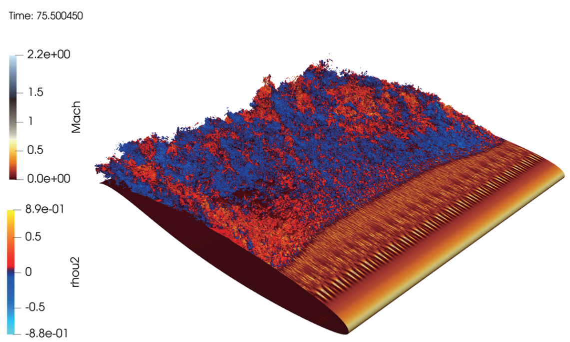


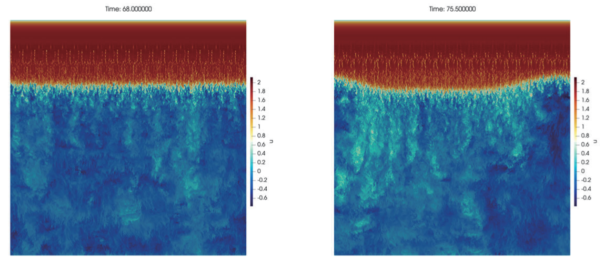
Extrusion to wide-span: AR1, AoA: 8

Wide-span buffet effects no longer purely 2D

- AR=1 case, M=0.72, **AoA: 8 degrees**.
- At AR=1: the flow is no longer strictly 2D across the span.
- Large **3D separation bubbles** occur, during low-lift phases.
- Seem to be linked to the point of **maximum flow separation** (low-lift).
- **Curvature of the shock front (no longer 2D planar shock)**



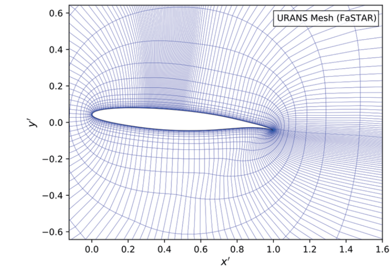




16

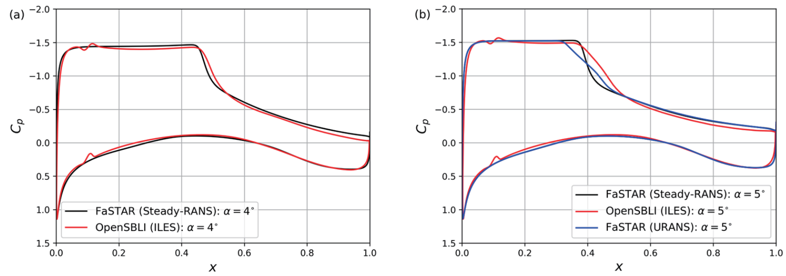
OpenSBLI (ILES) vs FaSTAR (URANS)

Validation of (2D) buffet frequency



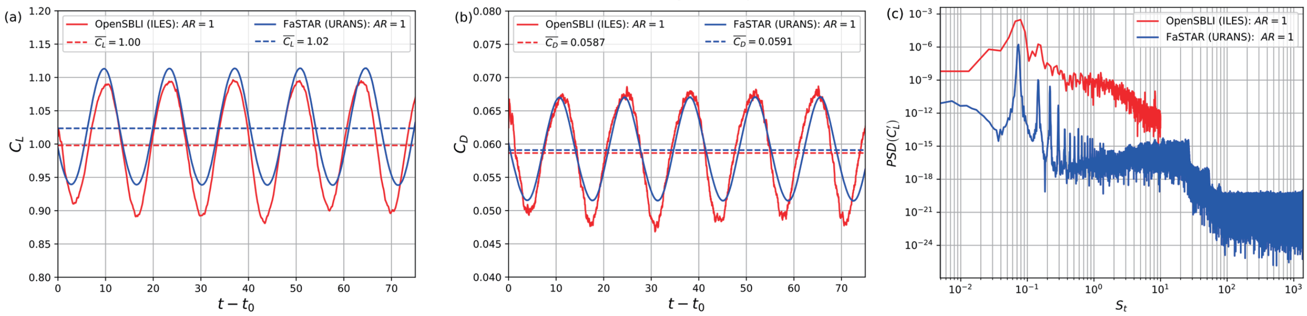
Example NASA-CRM mesh. (Plotting every 5th line for clarity)

Steady RANS + Unsteady RANS (Cp distributions)



Very good agreement at moderate AoA - Despite many differences between ILES/URANS, grid etc

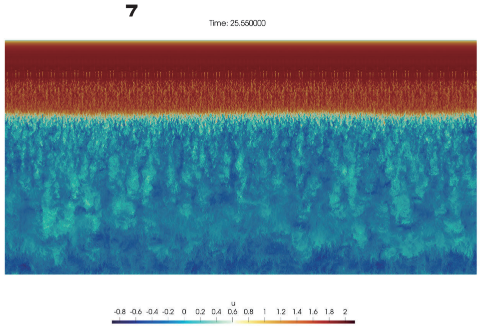
Unsteady RANS (CL, CD histories and PSD comparison) - AoA = 5

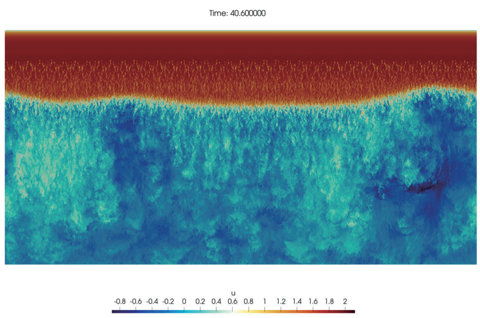


3D buffet effects at AR=2 & URANS comparison

- Similar behaviour observed at **AR=2** and AoA = 7 degrees.
- However, now **multiple large intermittent 3D separation bubbles** develop across the span.
- Persistent across multiple buffet cycles and **strongest during low-lift buffet phases**.

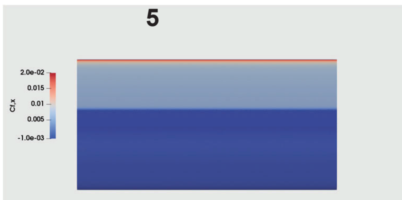
OpenSBLI: AoA = 7



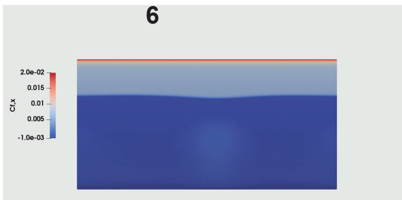


FaSTAR (URANS) - Andrea-san

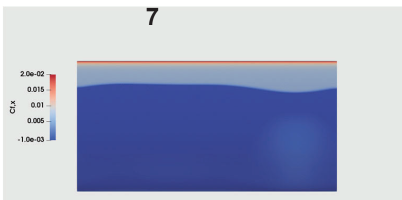
AoA = 5

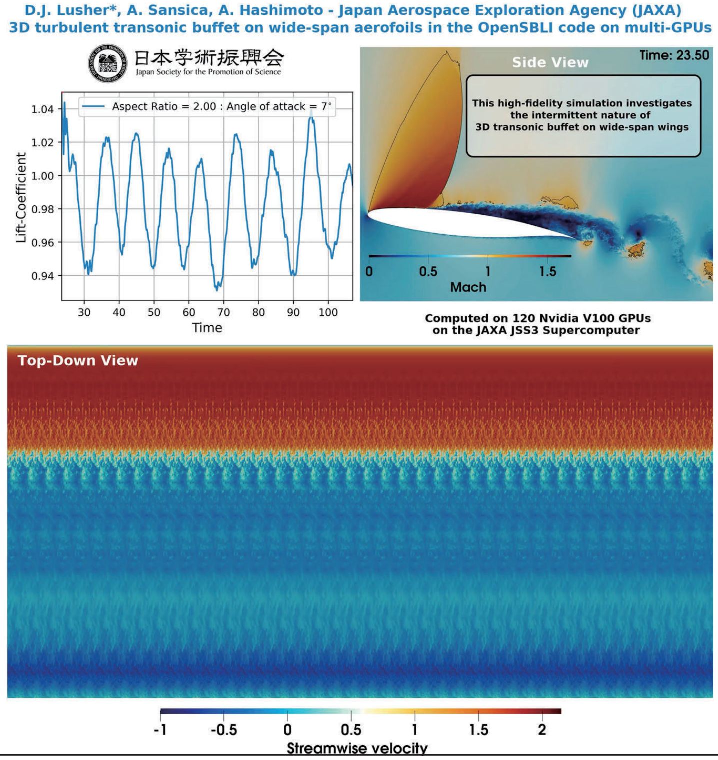


AoA = 6



AoA = 7





Conclusions & Summary

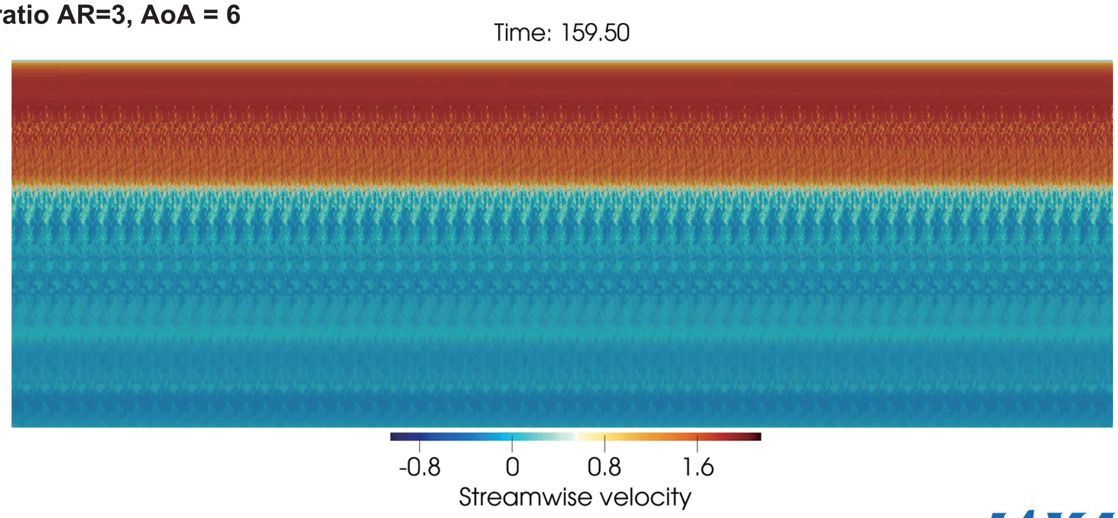
- Q. Can we observe 3D buffet effects for infinite (periodic) wings with zero sweep? - Yes, **at sufficiently wide aspect ratios.**
- World first high-fidelity **3D wide-span turbulent transonic buffet** simulations were performed on JSS3 GPU nodes, for NASA-CRM extruded wings at Re = 500,000 and Mach 0.72.
- **Large simulations on N > 10^9 mesh points.**
- Parametric study showed **2D shock oscillation buffet onset** occurs between 4-5 degrees AoA at **AR=0.05**.
- At **wide-span (AR=1,2,3)**, large **3D separation bubbles** form during low-lift phases, these **cannot be captured by narrow-span simulations**.
- They lead to **span-wise inhomogeneous curvature of the main shockwave.**
- We show that **narrow span simulations are not sufficient** to fully capture buffet phenomenon.
- Outlook: **Applying Modal Decomposition Methods (SPOD/DMD)**, publishing results.

Contact: lusher.david@jaxa.jp. More information on the OpenSBLI code (https://github.com/opensbli/opensbli): D.J. Lusher, S.P. Jammy, N.D. Sandham. **OpenSBLI: Automated code-generation for heterogeneous computing architectures applied to compressible fluid dynamics on structured grids**. Computer Physics Communications (2021).

This work is funded by a JSPS postdoctoral fellowship and JSPS KAKENHI Grant: 22F2205. Computational time was provided by the JAXA JSS3 supercomputing facility and associated support staff, and the Fugaku supercomputer at RIKEN on projects hp220195, hp220226.

Thank you - Questions?

Aspect ratio AR=3, AoA = 6



Contact: lusher.david@jaxa.jp. More information on the OpenSBLI code (https://github.com/opensbli/opensbli): D.J. Lusher, S.P. Jammy, N.D. Sandham. **OpenSBLI: Automated code-generation for heterogeneous computing architectures applied to compressible fluid dynamics on structured grids**. Computer Physics Communications (2021).

This work is funded by a JSPS postdoctoral fellowship and JSPS KAKENHI Grant: 22F2205. Computational time was provided by the JAXA JSS3 supercomputing facility and associated support staff, and the Fugaku supercomputer at RIKEN on projects hp220195, hp220226.

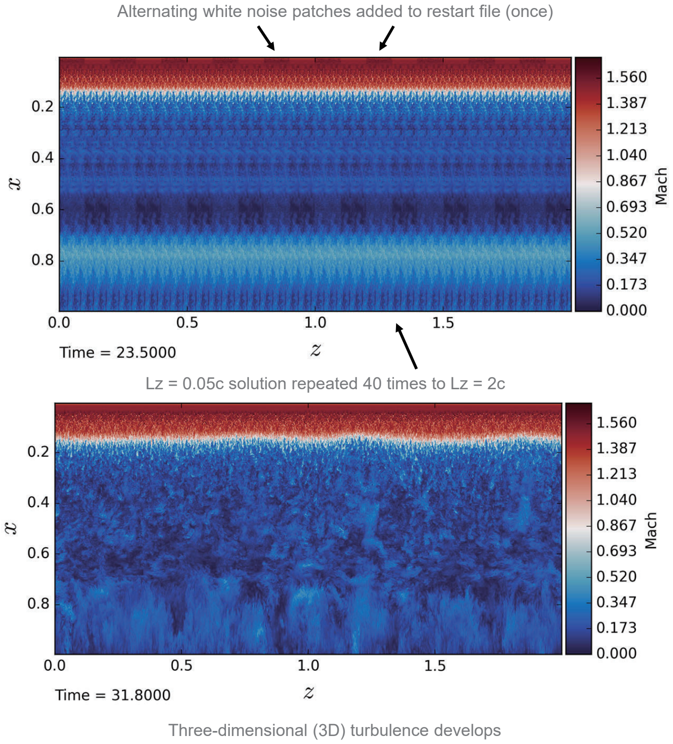
JAPAN SOCIETY FOR THE PROMOTION OF SCIENCE
日本学術振興会

Extra

Extrusion to wide-span: initial condition

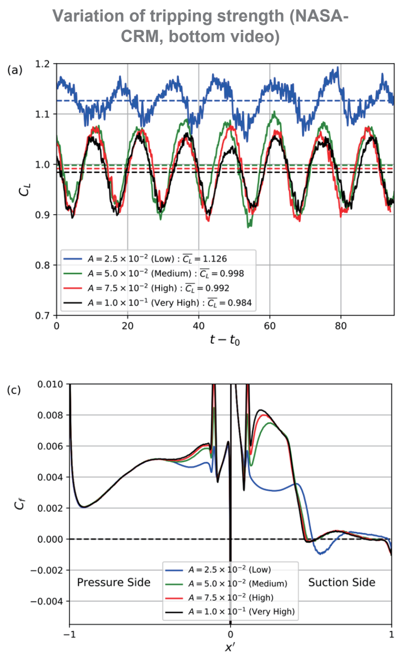
Avoiding long expensive transients

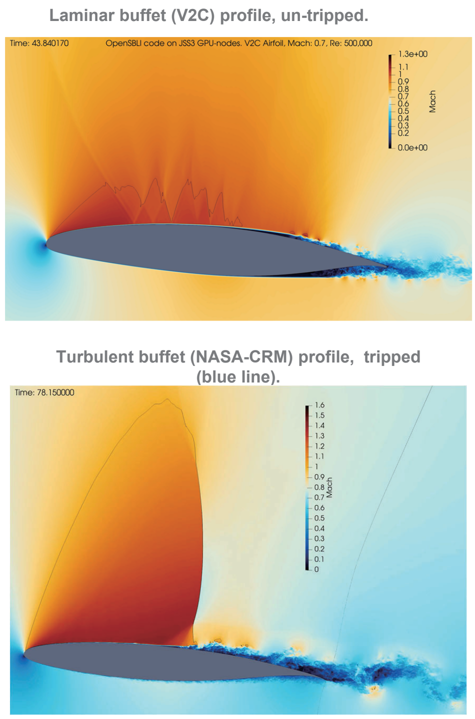
- Due to span wise periodic boundary condition, wide-span simulations can be **initialised with the fully developed narrow-domain flow-fields**.
- 0.05c profile is **repeated across the span 20-60** (AR1-AR3) times.
- White noise is added to the boundary-layer once into the restart file to help **break symmetry quickly**.
- No long wavelengths are forced, large 3D effects develop naturally.



Influence of forcing amplitude - IUTAM 2024

- Plan to submit work to **IUTAM 2024 conference**.
- Varying the **strength of the tripping** on buffet cases.
- For weaker cases, the flow becomes transitional.
- Would like to understand **the point of switching** between laminar and turbulent buffet.





Influence of forcing amplitude