Recent Developments in the CHApel Multi-Physics Simulation Software

CHIUW 2022, The 9th Annual Chapel Implementers and Users Workshop

Frédéric Plante, Post-doc researcher
Michael Gagnon, M. Sc. A.
Éric Laurendeau, Professor

CRIAQ
CONSORTIUM FOR RESEARCH AND INNOVATION IN AEROSPACE IN QUÉBEC

NSERC CRSNG

POLYTECHNIQUE MONTRÉAL
TECHNOLOGICAL UNIVERSITY

Table of contents

Aerospace industry

The aerospace industry uses several design tool :

- Flight test ($\approx 100\,000\$/hour$)
- Wind tunnel ($\approx 10\,000\$/hour$)
- Numerical Simulation ($\approx 1\,000\$/hour$)

Hence, the industry relies more and more on numerical simulations. However, the software needs to be :

- Robust : run thousand of simulations without failing ;
- Fast : run problems over a billion degrees of freedom ;
- Multi-physics : the optimization of aircraft by taking account multiple discipline simultaneously is now the state-of-the-art ;
- Precise.

Professor Laurendeau research team

Professor Laurendeau holds a Canada Research Chair and an Industrial Research Chair aimed at tackling :

- Aero-icing simulation (Computing the accretion of ice on wings) ;
- Aero-elastics (Computing structural deformations caused by aerodynamic forces) ;
- Predict edges of the flight envelope ;
- Apply multi-physic methods to aircraft design.

The challenges of the laboratory are :

- Developing software for applications close to the industrial ones ;
- The students come from a mechanical engineering background but must do High Performance Computing ;
- The projects must be done in 2 to 4 years.

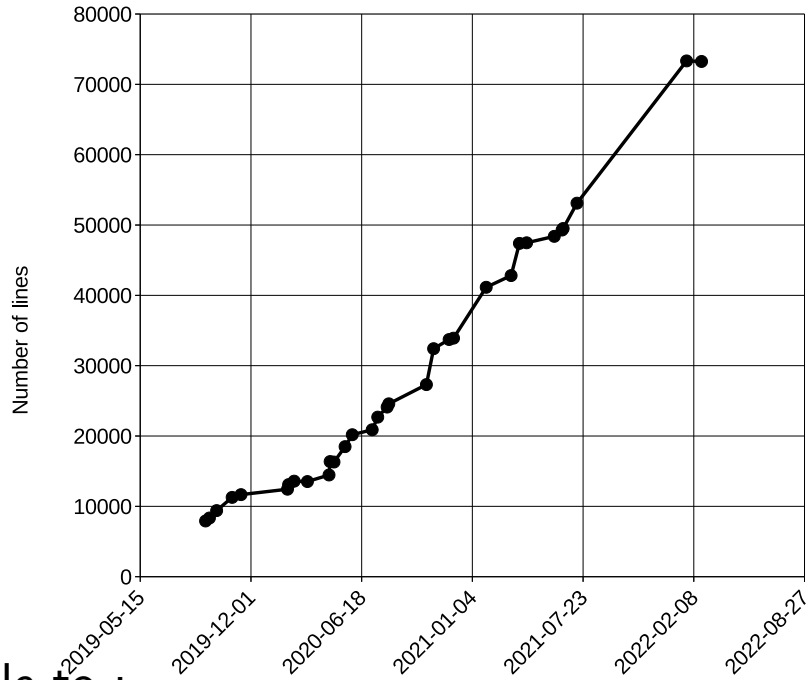
Why Chapel ?

- Chapel is an alternative to the classic C/C++ or Fortran with MPI paradigm used in CFD ;
- Use a global namespace for direct access to local or remote variables (allow to rapidly do distributed memory computations) ;
- Feature rich language (memory management, domain, list, set, etc.) ;
- Object-oriented, allowing us to reuse structures for several models ;
- Chapel is a productive language :
  - Reduces the barrier of entry to HPC. Well suited in our academic laboratory ;
  - CHAMPS was developed by 4 Master students, a Ph.D student (part time), a research assistant and a postdoctoral researcher.

For these reasons we chose to use the Chapel Language to develop our new 3D Computational Fluid Dynamics Software (CHAMPS) in 2019.

CHapel Multi-Physics Simulation

Number of lines of the CHAMPS software

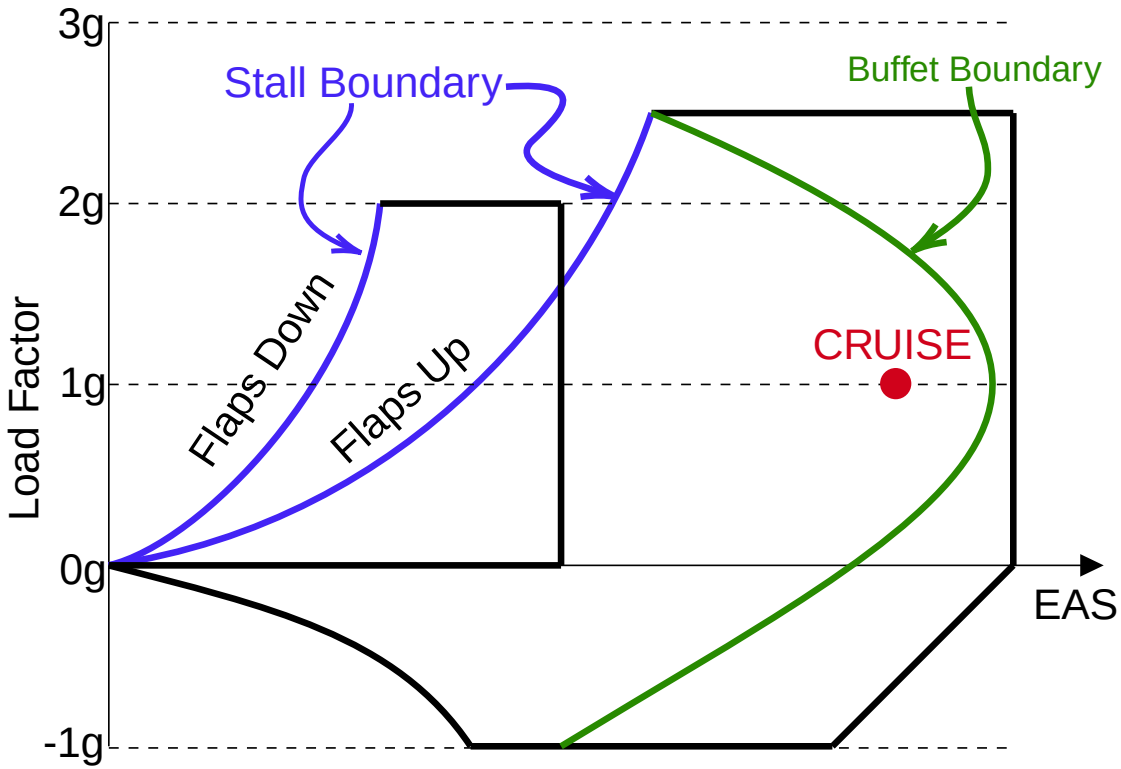


By 2020 we were able to :

- Reproduce the 5th AIAA CFD Drag Prediction Workshop ;
- Participate in the 1st AIAA CFD Icing Prediction Workshop ;
- Participate in the 4th AIAA CFD High Lift Prediction Workshop.

All of which are the state of the art for external aerodynamics CFD aircraft application.

Context - Flight Envelope



- Minimum takeoff safety speed is $V_{2min} = 1.13 V_{S1g}$;
- Minimum landing approach speed is $V_{REF} = 1.23 V_{S1g}$;
- Maximum cruise speed is limited by buffeting (vibration of $\pm 0.050g$).

Context - Flight Envelope

Buffet and stall involve unsteady phenomena. The prediction of such phenomena can be done by several methods :

- Unsteady simulations (URANS, DES, DNS) ;
  - Solves the non-linear equations ;
  - Very costly - not ready to be used in design loops.
- Empirical criteria (break in the lift/moment curves, critical Mach number, etc.) ;
  - Very fast to evaluate ;
  - Decoupled from the physical model.
- Global stability analysis [1, 2, 3, 4]
  - Relies on a linearized version of the physical model ;
  - Lower cost than unsteady simulations.

Governing equations

The Reynolds-Averaged Navier-Stokes (RANS) equations in integral form are :

$$\boldsymbol{\Omega}\frac{\partial \boldsymbol{W}}{\partial t} + \boldsymbol{R}(\boldsymbol{W}) = 0 \tag{1}$$

Assuming a small perturbation $\epsilon \boldsymbol{W}'$ around $\boldsymbol{W_0}$ so that $\boldsymbol{\Omega}\frac{\partial \boldsymbol{W_0}}{\partial t} + \boldsymbol{R}(\boldsymbol{W_0}) = 0$

$$\boldsymbol{\Omega}\frac{\partial \boldsymbol{W_0}}{\partial t} + \boldsymbol{\Omega}\epsilon\frac{\partial \boldsymbol{W}'}{\partial t} + \boldsymbol{R}(\boldsymbol{W_0} + \epsilon\boldsymbol{W}') = \boldsymbol{\Omega}\frac{\partial \boldsymbol{W}'}{\partial t} + \left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}\right|_{\boldsymbol{W_0}} \boldsymbol{W}' = 0 \tag{2}$$

Assuming the solution takes the form of a normal mode : $\boldsymbol{W}' = \hat{\boldsymbol{W}} e^{\lambda t}$

$$\lambda \hat{\boldsymbol{W}} = A\hat{\boldsymbol{W}} \tag{3}$$

To solve this problem we use an Arnoldi iteration with shift-and-invert spectral transformation :

$$(\boldsymbol{A} - \eta \boldsymbol{I})^{-1}\boldsymbol{x} = \boldsymbol{x}\kappa \tag{4}$$

with $\kappa = 1/(\lambda - \eta)$. This allows to find the eigenvalue λ closest to η. Here $\lambda = \sigma + \omega j$ with σ **the Growth rate** and ω **the Frequency**.

Arnoldi method

The Arnoldi iteration is (for $(\boldsymbol{A} - \eta \boldsymbol{I})^{-1}\boldsymbol{x} = \boldsymbol{x}\kappa$) :

1: Set q_1 to an arbitrary unitary vector
2: **for** $k = 2, 3, ..n$ **do**
3: $\quad \boldsymbol{q}_k \leftarrow (\boldsymbol{A} - \eta \boldsymbol{I})^{-1} \boldsymbol{q}_{k-1}$
4: $\quad$ **for** $j = 1..k-1$ **do**
5: $\quad\quad \boldsymbol{h}_{j,k-1} \leftarrow \boldsymbol{q}_j^* \boldsymbol{q}_k$
6: $\quad\quad \boldsymbol{q}_k \leftarrow \boldsymbol{q}_k - \boldsymbol{h}_{j,k-1} \boldsymbol{q}_j$
7: $\quad$ **end for**
8: $\quad \boldsymbol{h}_{k,k-1} \leftarrow ||\boldsymbol{q}_k||$
9: $\quad \boldsymbol{q}_k \leftarrow \frac{\boldsymbol{q}_k}{\boldsymbol{h}_{k,k-1}}$
10: **end for**

Line 3 $\rightarrow$ Need to solve large linear systems of equations.
Solution : We use a iterative GMRES method

Implementation - SolverHandle$_c$

In CHAMPS we have a class to define a linear system of equation (Right-Hand Side, Left-Hand Side and solver). For the stability, we have no RHS, but the LHS is the same as in the flow solver. So we add an attribute for the eigensolver (and we define either the solver_ or the eigenSolver_).

```
1
2  class SolverHandle_c
3  {
4     ...
5     var solver_ :owned LinearSolver_c=new owned LinearSolver_c(0);
6     var eigenSolver_ :owned eigenValuesSolver_c=new owned eigenValuesSolver_c();
7     var lhs_  : [lhsDom_] real_t;
8     var rhs_  : [rhsDom_] real_t;
9     ...
10    proc init(inputs : SolverInputs_r, ...)
11    {
12       ...
13       when SolverType_t.GMRES
14       {
15          ...
16          solver_=new owned GMRESsolver_c(nVariables_, ...);
17       }
18       when SolverType_t.EIGENVALUE
19       {
20          ...
21          eigenSolver_=new owned arnoldiShiftAndInvert_c(nVariables_, ...);
22       }
23    }
24 }
```

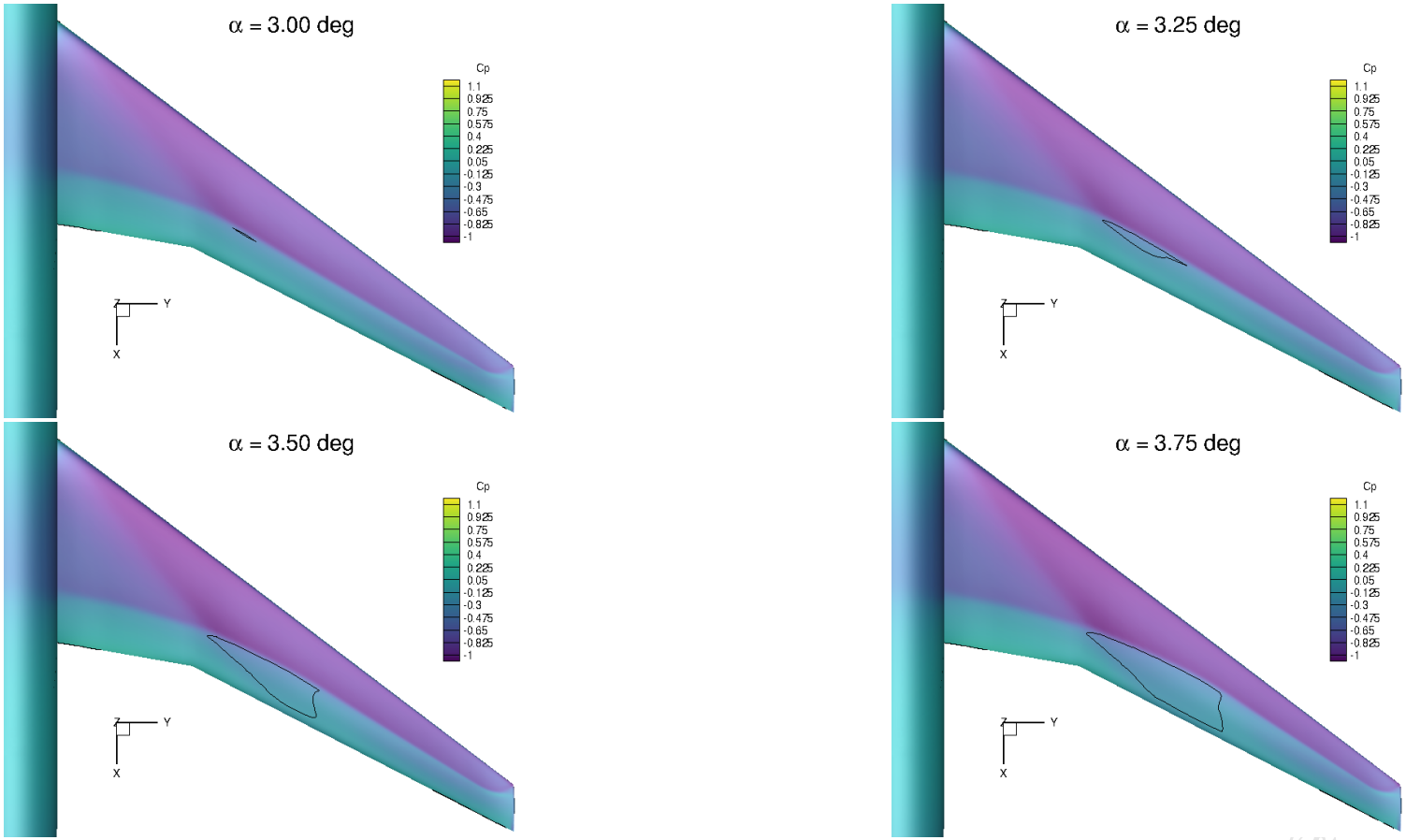
Implementation - GMRESsolver_c

```
1  class GMRESsolver_c : LinearSolver_c
2  {
3      ...
4      var gmresLinearSolver_ : owned GMRES_c = new owned GMRES_c();
5      override proc linearSolve(...)
6      {
7          ...
8          var it = gmresLinearSolver_.solve(lhs_, solverHandle.rhs_,
       buildPreconditioner = updateLHS);
9          ...
10     }
11 }
```

```
1  class arnoldiShiftAndInvert_c : eigenValuesSolver_c
2  {
3      ...
4      var krylovSize_ : int = 1;
5      const maxKrylovSize_ : int = 1;
6      var gmresLinearSolver_ : owned GMRES_c = new owned GMRES_c();
7      override proc eigenSolve(solverHandle, zone)
8      {
9          ...
10         fillLHS(solverHandle, zone);
11         for iteration in 0..#maxIteration_
12         {
13             ...
14             for kk in 0..#krylovSize_
15             {
16                 ...
17                 gmresLinearSolver_.solve(matShift_, tempVec1, true);
18             }
19         }
20     }
21 }
```

3D Buffet

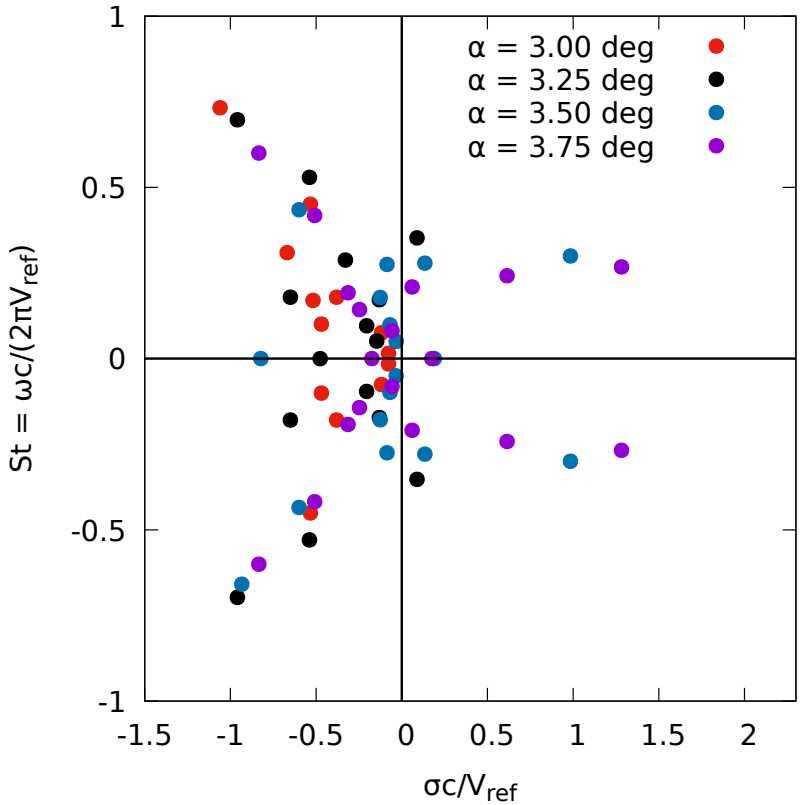
Case 2a of the AIAA DPW7 : NASA CRM, Reynolds 20×10^6, Mach 0.85 Grids provided by J. Vassberg are the medium level (41 million cells), de-refined with *ONERA – Cassiopee* to have the tiny level (5 million cells)



Contour shows the line of $Cf_x = 0.0$

3D Buffet

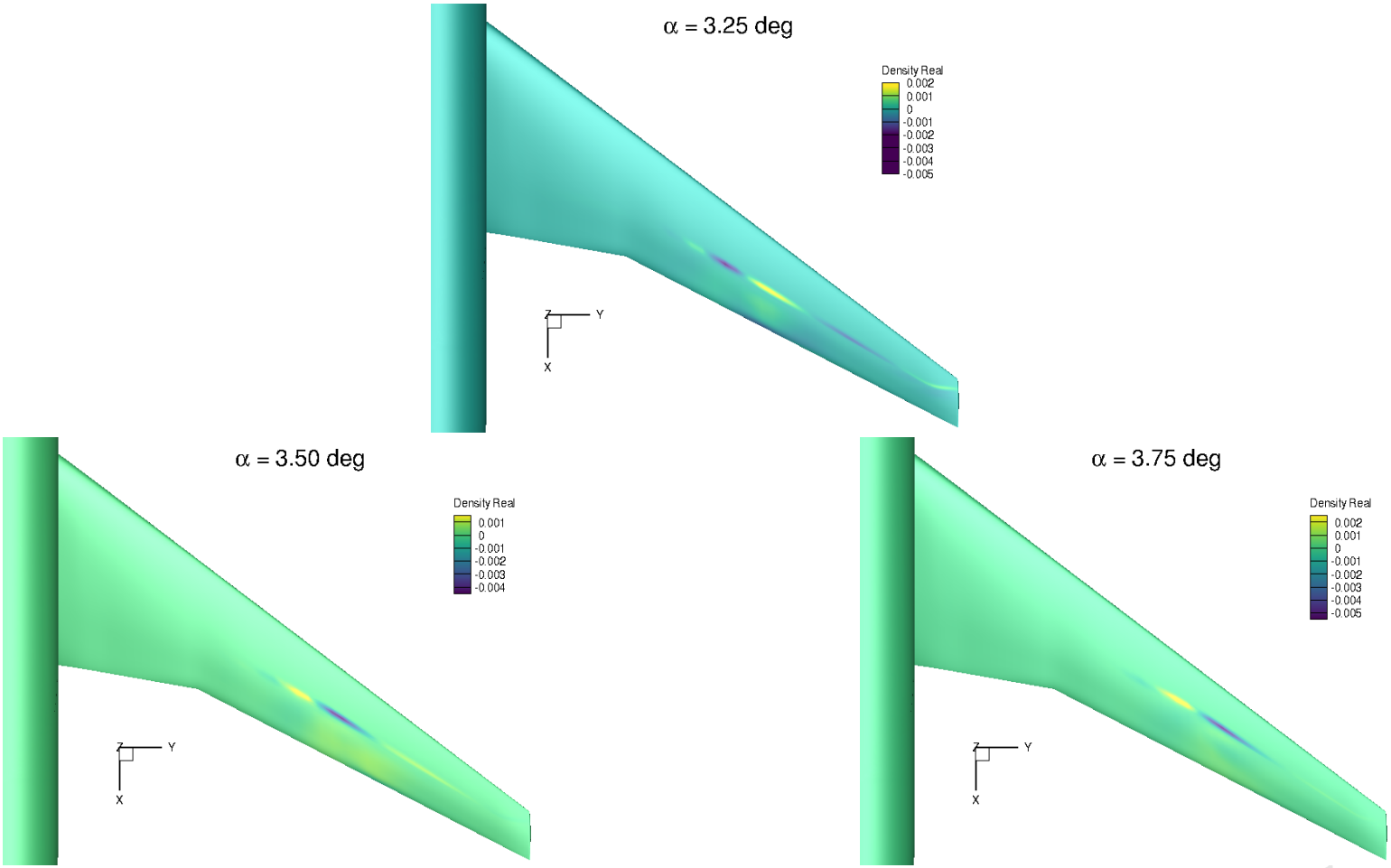
Case 2a of the AIAA DPW7 : NASA CRM, Reynolds 20×10^6, Mach 0.85



- A mode becomes unstable for α between 3.00° and 3.25° ;
- Multiple unstable modes when α increases.

3D Buffet

Case 2a of the AIAA DPW7 : NASA CRM, Reynolds 20×10^6, Mach 0.85



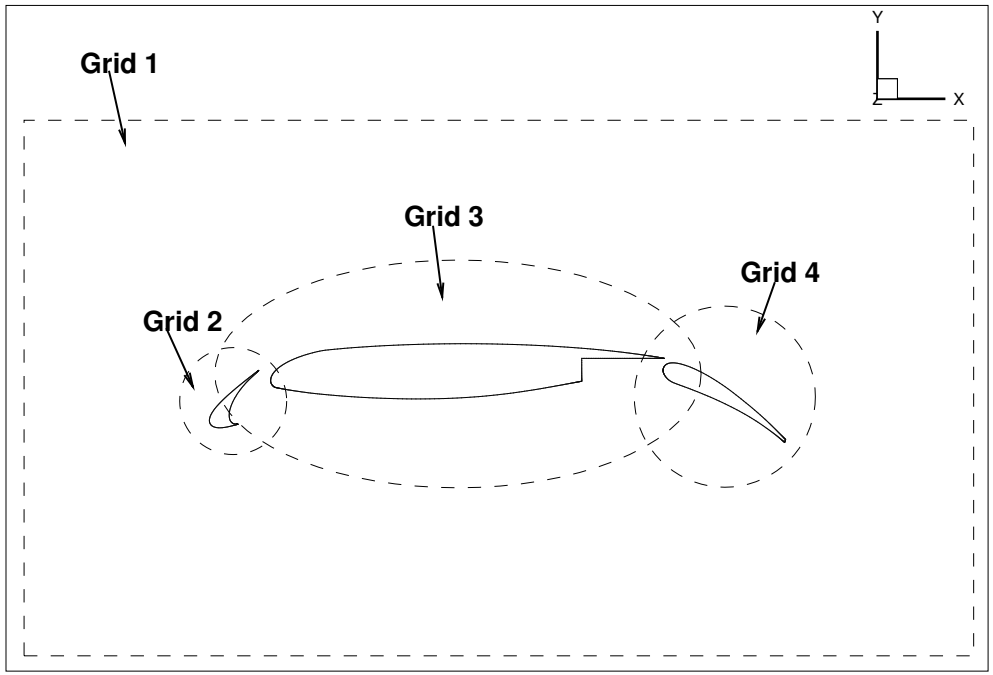
Contour shows the line of $Cf_x = 0.0$
Similar results to those of Timme [5]

Spatial discretization

- Simulations are highly sensitive to grid quality.
- Two different discretizations of the same domain can lead to substantially different results. [6].
- Up to 50% of the total time of CFD process.

The *overset* approach

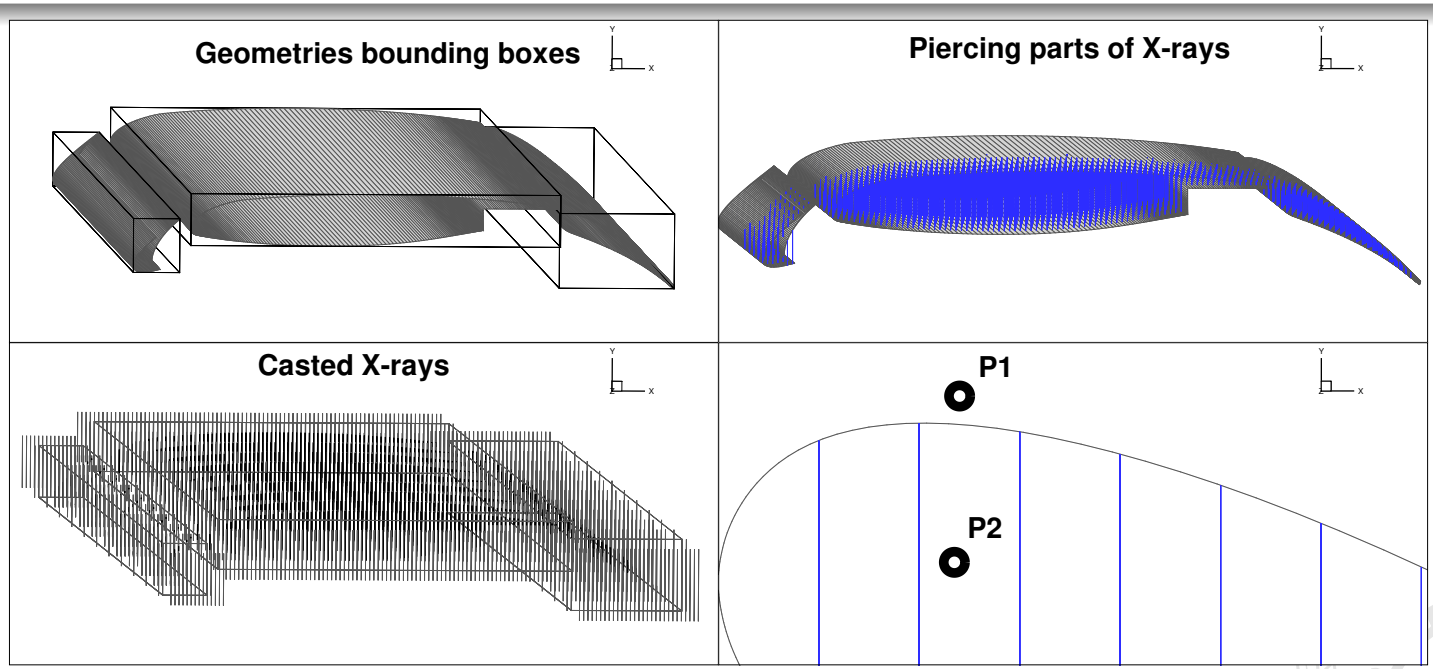
- Subdivision of the computational domain.
- Arbitrary overlap between the oversetting grids.
- Creation of a connectivity using a preprocessor.



X-ray hole cutting method

X-ray method [7]

1. Create a bounding box around each solid components.
2. Cast rays within the bounding boxes.
3. Locate collision points between the rays and the facets of the solid components.
4. Verify the inside/outside status of each grid cell.



Connectivity

if $\vec{f_i} \cdot \vec{n_i} > 0$, cell center is outside.

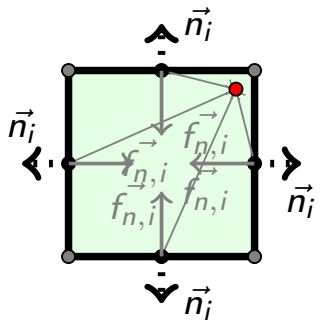
else $\vec{f_i} \cdot \vec{n_i} =< 0$, cell center is inside.

$\vec{f_{n,i}}$= Middle facet - interpolated point vector

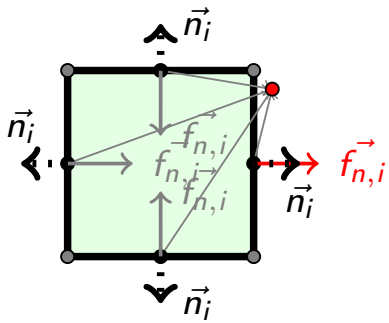
$\vec{n_i}$ = Facet normal vector

• = Current potential interpolated cell center

Point is contained



Point is not contained



Oct-tree data structure

Speed and automation of the overset approach are two very important criteria for a good overset preprocessor. The oct-tree data structure is an approximation of a dimensional object by a set of cubes divided recursively [8]. The donor search step and the hole cutting step both require many searches through the domain. To speed up the search process, the cubes are hierarchically organized as illustrated in Figure 2.8 to reach the information faster.

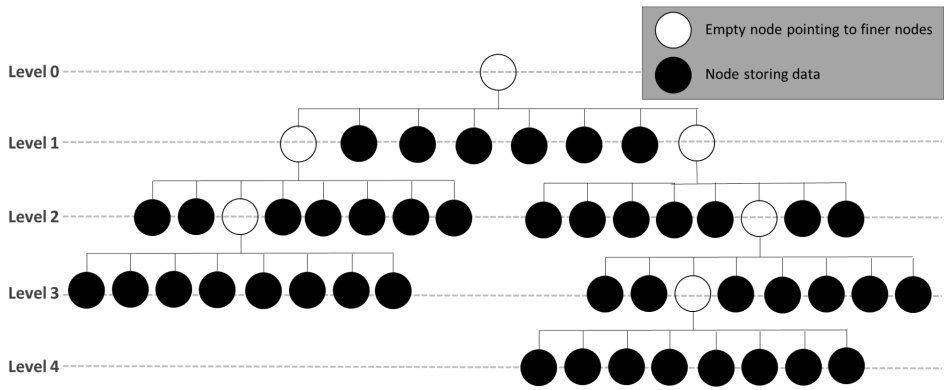
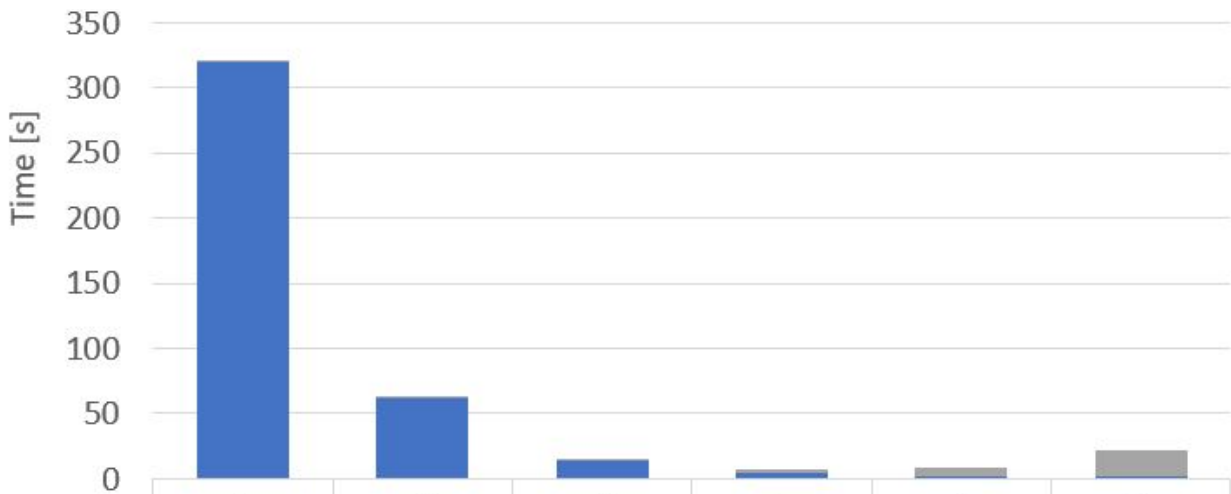


Illustration of the Oct-tree data structure

Oct-tree data structure

Donor search for 100'000 randomly generated grid points within a grid with 1'000'000 grid cells.

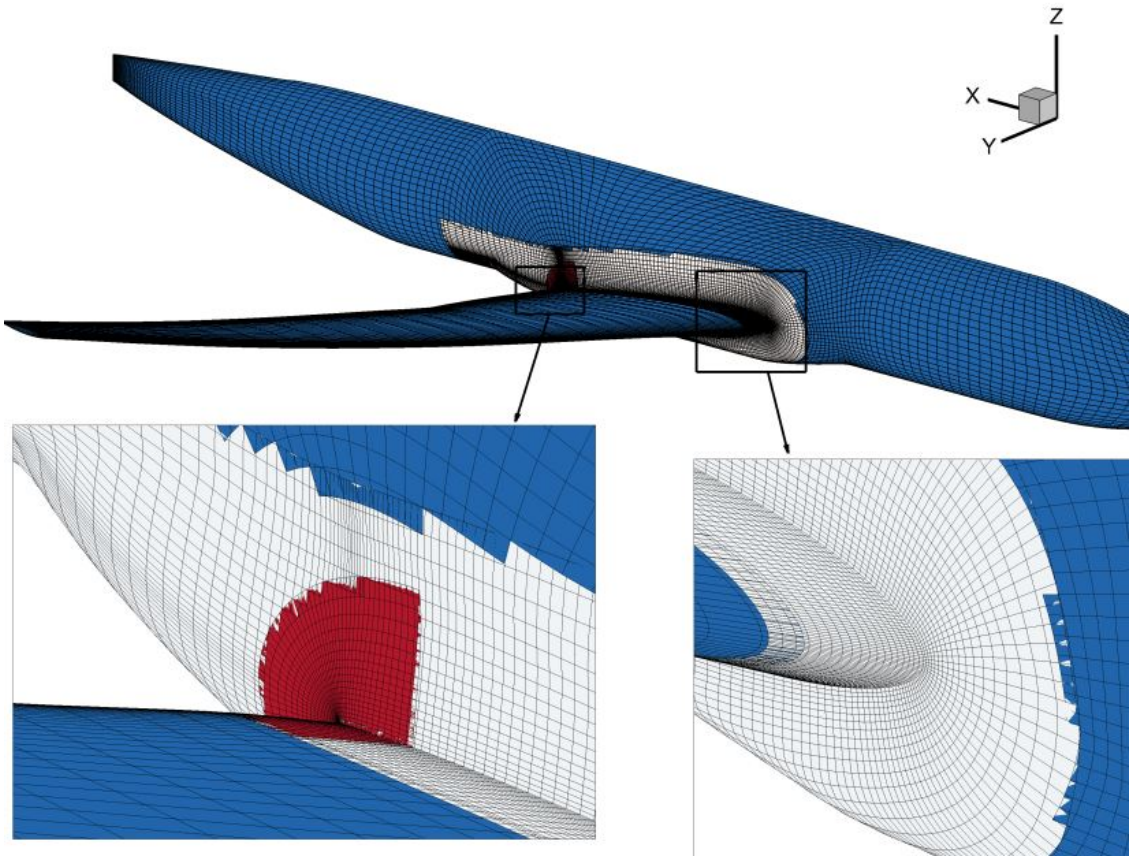
Time in seconds to create the structure (grey) and compute the donor search (blue) in relation to the deepness of the oct-tree structure :



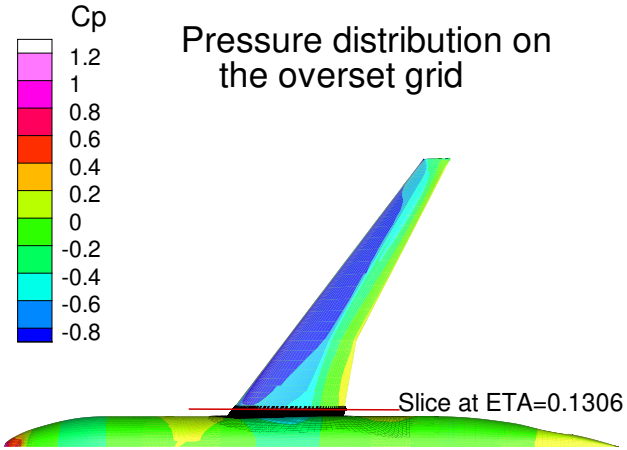
	1	2	3	4	5	6
Build Octree Structure	0.45414	0.95928	1.59908	2.76917	6.70799	20.18529
Compute Donor Search	319.168	61.3029	13.8711	4.01603	2.11207	1.46236

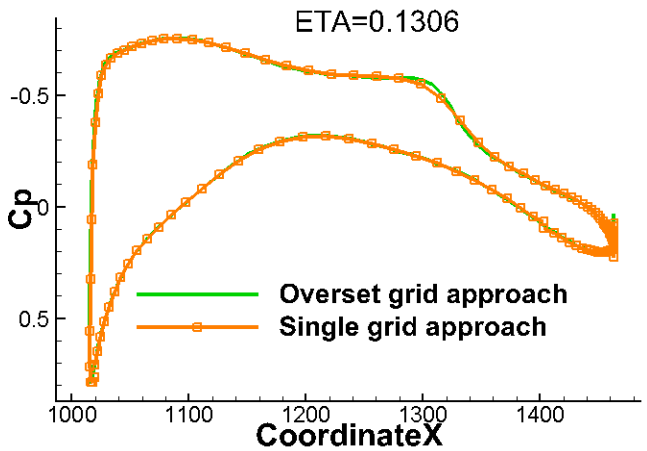
Overset approach to improve grid regions

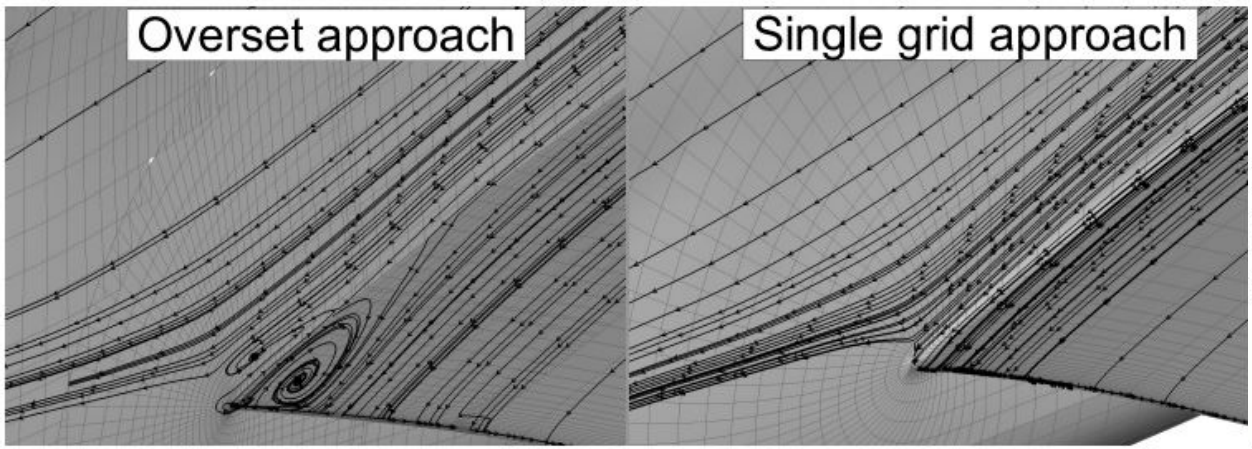
Here we used the overset to mesh the region connecting the wing to the fuselage of an airplane

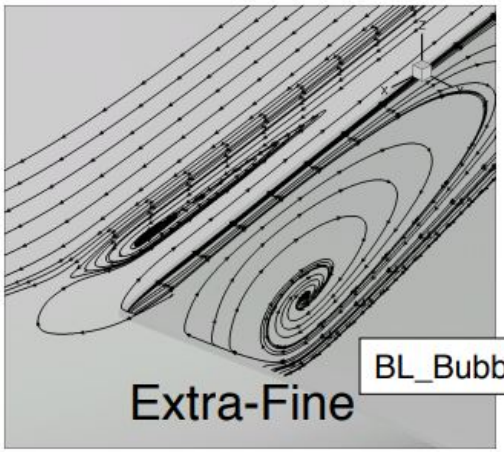


Overset approach to improve grid regions









Results from CHAMPS

Results from FUN3D[9]

Conclusion

In conclusion, the capability of Champs continues to increase :

- A matrix-free global stability analysis method relying on a GMRES solver is added to CHAMPS ;
- Application to a half wing-body configuration allows to predict buffet ;
- Overset capability is functional for complex industrial level application.

What is next ?

- Participation to the 7th Drag Prediction Workshop on June 25-26 2022 ;
- 6 papers at the AIAA Aviation conference in Chicago ;
- New students all working on/with CHAMPS ;
- Up coming second Icing prediction workshop.

Reference I

[1] Sipp, D., Marquet, O., Meliga, P., and Barbagallo, A., "Dynamics and Control of Global Instabilities in Open-Flows : A Linearized Approach," *Applied Mechanics Reviews*, Vol. 63, No. 3, 2010, p. 030801. doi: 10.1115/1.4001478.

[2] Theofilis, V., "Advances in Global Linear Instability Analysis of Nonparallel and Three-Dimensional Flows," *Progress in Aerospace Sciences*, Vol. 39, No. 4, 2003, pp. 249–315. doi: 10.1016/S0376-0421(02)00030-1.

[3] Theofilis, V., "Global Linear Instability," *Annual Review of Fluid Mechanics*, Vol. 43, No. 1, 2011, pp. 319–352. doi: 10.1146/annurev-fluid-122109-160705.

[4] Taira, K., Brunton, S. L., Dawson, S. T. M., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S., "Modal Analysis of Fluid Flows : An Overview," *AIAA Journal*, Vol. 55, No. 12, 2017, pp. 4013–4041. doi: 10.2514/1.J056060.

[5] Timme, S., "Global Instability of Wing Shock-Buffet Onset," *Journal of Fluid Mechanics*, Vol. 885, 2020, p. A37. doi: 10.1017/jfm.2019.1001.

[6] Mavriplis, D., "Grid Resolution Study of a Drag Prediction Workshop Configuration Using the NSU3D Unstructured Mesh Solver," *23rd AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, 2005. doi: 10.2514/6.2005-4729, URL https://doi.org/10.2514/6.2005-4729.

[7] Meakin, R., "Object X-rays for cutting holes in composite overset structured grids," *15th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2001. doi: 10.2514/6.2001-2537, URL https://doi.org/10.2514/6.2001-2537.

[8] Yamaguchi, K., Kunii, T., Fujimura, K., and Toriya, H., "Octree-Related Data Structures and Algorithms," *IEEE Computer Graphics and Applications*, Vol. 4, No. 1, 1984, pp. 53–59. doi: 10.1109/mcg.1984.275901, URL https://doi.org/10.1109/mcg.1984.275901.

[9] "16_FUN3D_DPW_IV_2.ppt," https://aiaa-dpw.larc.nasa.gov/Workshop4/presentations/DPW4_Presentations_files/D1-16_FUN3D_DPW_IV_2.pdf, ???? (Accessed on 04/22/2022).

Acknowledgements

This project is supported in part by NSERC and CRIAQ.
Computations were performed on Compute Canada/Calcul Québec Beluga and Narval clusters.